Abstract—In this paper, a novel transform-based, blind and robust 3D mesh watermarking scheme is presented. The 3D surface of the mesh is firstly divided into a number of discrete continuous regions, each of which is successively sampled and mapped onto oblate spheroids, using a novel surface parameterization scheme. The embedding is performed in the spheroidal harmonic coefficients of the spheroids, using a novel embedding scheme. Changes made to the transform domain are then reversed back to the spatial domain, thus forming the watermarked 3D mesh. The embedding scheme presented herein resembles, in principal, the ones using the multiplicative embedding rule (inherently providing high imperceptibility). The watermark detection is blind and by far more powerful than the various correlators typically incorporated by multiplicative schemes. Experimental results have shown that the proposed blind watermarking scheme is competitively robust against similarity transformations, connectivity attacks, mesh simplification and refinement, unbalanced re-sampling, smoothing and noise addition, even when juxtaposed to the informed ones.

Index Terms—3D watermarking, blind detection, copyright protection, mesh watermarking, spheroidal harmonics.

I. INTRODUCTION

In recent years, as an enormous number of multimedia data becomes publicly available (such as digital photographs, video, audio, etc.), the dire necessity for an effective protection of the ownership rights of the developers as well as of the owners of multimedia content emerges. 3D virtual models do not constitute an exception, inasmuch as they are widely utilized in applications such as gaming, simulation, industrial design, advertising, and the like. Watermarking is a concerted approach to address this need.

Although the application of watermarking for copyright protection of images, video and audio is already reaching maturity level, yet the 3D mesh watermarking continues to remain a challenging issue due to particularities so much to the representation as to the rendering of 3D models. In this way, initially there are numerous alternative representation structures for 3D models, such as point-sampled surfaces, polygonal meshes, parametric surfaces (e.g. through Bezier Splines, Non-Uniform Rational B-Splines (NURBS), etc.).

Regarding the rendering of 3D models, depending on the application area, 3D models are frequently combined with attributes that carry additional information such as texture, color, reflective characteristics, etc., yet reducing, in an indirect way, the significance of the level of details in the 3D structure of the model (as part of this information can be conveyed by these attributes).

In a typical watermarking scheme, a signal (the watermark) is cast into the original content being protected. The requirements which must be met by a watermarking scheme depend on the area of application. Typical application areas of watermarking include:

- **Data hiding**, where watermarks are used to carry hidden information within the original content.
- **Authentication and integrity checking**, where watermarks are used to verify the originality of the protected content or even to locate potentially altered parts of the content.
- **Copyright protection**, where watermarks are used to carry information about content ownership, in a way that is robust against malicious modifications of the protected content (called watermark attacks).

Furthermore, if, in a watermarking scheme, the detection of the presence of a watermark requires the mere knowledge of a private key, then the scheme is said to have a blind detection. Should further information be required, such as specificities of the watermark embedding procedure or even the knowledge of the original content, then this scheme is said to have an informed detection. Blind detection, obviously, has advantages over informed detection, since in the former there is no necessity of additional information to hold the detection, the knowledge of which constitutes a security issue. For more details on the advantages of blind detection, the interested reader can refer to [1].

Given that the present work focuses on copyright protection applications, henceforth we will be solely occupied with this application area, when referring to watermarking schemes. In the sequel, there will be a concise presentation of the published work on 3D watermarking. A detailed presentation of most existing 3D watermarking algorithms and watermark attacks can be found in [2] and [3].

II. RELATED WORK

Since the first work published on 3D watermarking by Ohbuchi et al. [4], a considerable number of 3D watermarking schemes have been proposed. From those performing the watermark embedding in the spatial domain, [5], [6], [7], and
[8] use informed detection, while [9], [10], [11], [12], [13], and [14] use blind detection. Likewise, from those performing the embedding in the transform domain, informed detection is the one used in [15], [16], [17], [18], and [19], while blind detection is used in [20], [21], [22], and [23].

Informed schemes robust against most known watermark attacks can be found in both spatial and transform-based algorithms. For example, in the spatial domain, Benedens [8] embeds a watermark in properly selected feature points of the 3D mesh, using Free Form Deformations (FFDs), in a scheme robust against affine transformations, smoothing, cropping, noise addition and re-sampling. In the transform domain, Praun et al. [18] use wavelet coefficients and radial basis functions (RBFs) to embed the watermark, resulting in a scheme robust against similarity and connectivity attacks, noise addition, compression, cropping and intense re-sampling.

On the contrary, the existing blind schemes to date fall appreciably short, in terms of robustness, from their counterpart informed ones. Nevertheless, blind detection constitutes an appealing property for any copyright protection system. In the spatial domain embedding, Wagner [9] proposes a blind scheme which embeds the watermark in the point normals, estimated by the application of the Laplacian operator in the neighborhood of any watermarked point. This scheme fails under connectivity attacks. Harte et al. [10] embed the watermark via alterations of the position of the mesh points. The watermarked points are chosen based on their distance to their neighboring points center, while the watermark bit embedded to them depends on their relative position to a bounding volume of their neighboring points. Though the scheme is robust against similarity attacks, smoothing and noise addition, it fails under connectivity attacks.

Zafeiriou et al. [14] use vertex deformations along the radius \( r \) of the spherical coordinates \((r, \theta, \phi)\) computed on the surface of the 3D object, parameterized through continuous NURBS patches. The proposed scheme is robust against similarity and mesh-simplification attacks. Spherical coordinates are also used in the blind scheme of Cho et al. [11], which embeds the watermark by modifying the mean or variance of the distribution of vertex norms, trough histogram mapping functions. Both schemes fail under cropping. In a recent publication by Rondao Alface et al. [13], a method to withstand cropping attacks is proposed. They employ geodesics and protrusion functions for the robust feature points extraction on the surface of the mesh, and propose a modification of the method presented in [11] to better withstand re-sampling attacks.

Concluding with the blind schemes in the spatial domain, A. G. Bors [12] uses controlled nonlinear perturbations on properly selected and ordered vertices of the mesh. The selection and ordering of the vertices to be watermarked is based on a visibility criterion, applied to their neighborhood. The algorithm resists similarity attacks, noise addition and cropping.

In the transform-based blind schemes, Cayre et al. [20] propose a method, which, based on spectral decomposition, embeds watermarks in the middle and high pseudo-frequencies by flipping spectral coefficient triplets. The scheme is robust against smoothing and noise addition, but fails under connectivity attacks. An attempt to decrease the vulnerability of this scheme to connectivity attacks was performed by Rondao Alface et al. [21], by using feature points to regain synchronization.

Finally, Uccheddu et al. [22] propose a blind wavelet-based watermarking scheme, applicable to meshes with semi-regular subdivision connectivity. This limitation is vanished in the scheme of Valette et al. [23], by employing lazy wavelets. Still, both schemes suffer from fragility against connectivity attacks.

Typically, the minimum requirements that any robust 3D watermarking scheme must fulfill are robustness against similarity transformations and connectivity attacks, as both leave the geometry of the 3D mesh unaffected. The most robust blind schemes presently available fail under connectivity attacks. The rest of the blind schemes are only capable of handling a limited number of attacks each.

Motivated by the need for a blind 3D watermarking method robust against combined attacks, in this paper, a novel approach to blind 3D mesh watermarking, suitable for copyright protection applications, is proposed. The mesh is firstly divided into a number of locally selected continuous regions (patches), each of which is sampled, based on an intensively smoothed version of the mesh. The watermark is embedded in the transform domain, by incorporating spheroidal harmonics, using a novel robust multiplicative embedding scheme. A compatible detector that comes along with the proposed embedding scheme is capable of robustly performing even under a very limited number of input coefficients. In this way, the detection can be separately held for the high and low pseudo-frequency coefficients respectively.

The proposed watermarking scheme presents some similarities with the work of Zafeiriou et al. [14], Cho et al. [11] and Rondao Alface et al. [13]. In this way, the proposed method partitions the surface of the mesh into patches of properly selected geodesic length, just as the method presented in [13] does. Likewise [14] and [11], which use the radius \( r \) of spherical coordinates to embed the watermark, the proposed method uses the spheroidal height \( u \) of the Jacobi ellipsoidal coordinates.

The key differences between our method and the above three lie in the introduction of a robust sampling technique over a smoothed version of the original mesh, the execution of the embedding in a transformed domain, and the use of a novel embedding and detection scheme. The new sampling technique presents high tolerance against noise and smoothing attacks, thus significantly enhancing the overall robustness of the watermarking scheme against these attacks. The invertibility of the transform guarantees that all the available data will be used to embed the watermark, while the orthonormality (and hence statistical independence) of the spheroidal coefficients makes it highly unlikely for the energy of a certain watermark attack to be properly distributed with equal severity into all watermarked coefficients. The less affected coefficients can then be used for a reliable decision extraction from the side of the detector, in regard to the presence of the watermark under inspection.
The resulting scheme presents high embedding capacity and robustness against similarity transformations, connectivity attacks, mesh simplification and refinement, unbalanced resampling, smoothing and noise addition, or a combination of these very attacks. The scheme fails under cropping.

The rest of the paper is organized as follows: In Section III the spheroidal harmonics series expansion is presented. The details of the embedding and detection procedures of the proposed watermarking scheme are described in Section IV, while the experimental results verifying the robustness of the watermarking algorithm against various watermark attacks are given in Section V. Finally, conclusions are drawn in closing Section VI.

III. SPHEROIDAL HARMONICS

The watermark embedding and detection procedures of the proposed scheme are both held in a transform domain, based on the use of one of the many variants of oblate spheroidal harmonics; namely the Jacobi ellipsoidal coordinates. A detailed presentation of oblate spheroidal harmonics and Jacobi ellipsoidal coordinates can be found in [24] and [25]. A brief overview of both follows, for the sake of completeness.

A. Oblate Spheroidal Coordinates

An oblate spheroid created by the revolution of an ellipse around the z-axis in 3D space is defined by

$$\frac{x^2 + y^2}{\alpha^2} + \frac{z^2}{b^2} = 1, \quad \alpha^2 > b^2$$

where $\alpha$ and $b$ refer to the semi-major and semi-minor axis of the spheroid respectively.

There exists a number of variants of the oblate spheroidal coordinates. The most commonly used variant, called Jacobi ellipsoidal coordinates $\{\lambda, \phi, u\}$, is chosen, by the use of which, a point in $\mathbb{R}^3$ is uniquely described by the intersection of families of (Fig.1)

- **Half planes**

\[ \mathbb{P}^2_{\cos \lambda, \sin \lambda} := \left\{ x \in \mathbb{R}^3 \mid y = x \tan \lambda \text{ for } \lambda \in [0, 2\pi), \lambda \neq \pi \pm \frac{\pi}{2}, x = 0, y \in [0, \text{sgn} \lambda \cdot \infty) \right\} \]

- **Confocal oblate spheroids**

\[ \mathbb{E}^2_{\sqrt{u^2 + \varepsilon^2}, u} := \left\{ x \in \mathbb{R}^3 \mid \frac{x^2 + y^2}{\varepsilon^2} + \frac{z^2}{u^2} = 1, \quad u \in (0, +\infty) \right\} \]

- **Confocal half hyperboloids of revolution**

\[ \mathbb{H}^2_{\cos \phi, \varepsilon \sin \phi} := \left\{ x \in \mathbb{R}^3 \mid \frac{x^2 + y^2}{\varepsilon^2 \cos^2 \phi} - \frac{z^2}{\varepsilon^2 \sin^2 \phi} = 1, \phi \in (-\frac{\pi}{2}, \frac{\pi}{2}), \phi \neq 0, \text{sgn} z = \text{sgn} \phi \right\} \]

where $\varepsilon := \sqrt{\alpha^2 - b^2}$ is the absolute eccentricity.

By intersecting the aforementioned families of 3D surfaces, the forward transformation of spheroidal coordinates $\{\lambda, \phi, u\}$ into Cartesian coordinates $\{x, y, z\}$ can be derived:

\[ x = \sqrt{u^2 + \varepsilon^2 \cos \phi \cos \lambda} \]

\[ y = \sqrt{u^2 + \varepsilon^2 \cos \phi \sin \lambda} \]

\[ z = u \sin \phi \]

and hence the backward (inverse) transformation of Cartesian coordinates $\{x, y, z\}$ into spheroidal coordinates $\{\lambda, \phi, u\}$ becomes

\[ \lambda = \arctan(y, x) \]

\[ \phi = \frac{z}{|z|} \arcsin \left( \sqrt{\frac{1}{2\varepsilon^2} (C - A)} \right) \]

\[ u = \sqrt{\frac{1}{2} (A + C)} \] where $A = x^2 + y^2 + z^2 - \varepsilon^2$ and $C = \sqrt{4\varepsilon^2 z^2 + A^2}$.

B. Spheroidal Harmonic Expansion

Thong and Grafarend [24] have shown that Jacobi ellipsoidal coordinates decompose the three dimensional Laplace differential equation into separable ordinary differential equations. Thus, a function $U(\lambda, \phi, u)$ defined in the orthogonal curvilinear coordinates $\{\lambda, \phi, u\}$, and such that it satisfies the three-dimensional Laplacian, can be expressed through

\[ U(\lambda, \phi, u) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} u_{nm} Q_{nm}(\frac{\phi}{\varepsilon}) e_{nm}(\lambda, \phi) \]

where

\[ e_{nm}(\lambda, \phi) = \left\{ \begin{array}{ll} \mathcal{P}_{nm}^+ (\sin \phi \cos (\lambda m)) & \forall m \geq 0 \\ \mathcal{P}_{nm}^- (\sin \phi \cos (\lambda m)) & \forall m < 0 \end{array} \right. \]

are the surface spheroidal harmonics, $\mathcal{P}_{nm}^+ (\cdot)$ and $\mathcal{P}_{nm}^- (\cdot)$ are the normalized associated Legendre functions of first and second kind respectively (both orthonormal functions), and $u_{nm}$ are the spheroidal harmonics coefficients of degree $n$ and order $m$.

The values of $U(\cdot, \cdot, \cdot)$ onto the surface of the reference spheroid are given, as depicted in (11), through

\[ U(\lambda, \phi, u = b) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} u_{nm} e_{nm}(\lambda, \phi) \]

The weight function

\[ w(\phi) := \frac{\alpha}{\sqrt{b^2 + \varepsilon^2 \sin^2 \phi} \left( \frac{1}{2} + \frac{b^2}{4\alpha \varepsilon} \ln \frac{\alpha + \varepsilon}{\alpha - \varepsilon} \right)} \]

sets the surface harmonic functions orthonormal with respect to their weighted scalar product, namely

\[ \langle e_{nm}(\lambda, \phi)|e_{kl}(\lambda, \phi) \rangle_w := \frac{1}{S} \int_{\mathbb{E}^2_{\sqrt{b^2 + \varepsilon^2}, b} \setminus \mathbb{E}^2_{\sqrt{b^2 + \varepsilon^2}, \infty}} w(\phi) e_{nm}(\lambda, \phi) e_{kl}(\lambda, \phi) dS \]

\[ = \delta_{kn} \delta_{lm} \]
where we employed the standard Kronecker delta $\delta_{ij}$, the global area of the reference spheroid $\mathbf{E}_r^{2\alpha,b}$

$$S = 4\pi\alpha \cdot \left\{ \frac{1}{2} + \frac{b^2}{4\alpha \varepsilon} \ln \frac{\alpha + \varepsilon}{\alpha - \varepsilon} \right\}$$

and the infinitesimal surface element

$$dS = \alpha \cdot \sqrt{b^2 + \varepsilon^2 \sin^2 \phi} \cos \phi \, d\lambda \, d\phi$$

Using the weighted orthonormality property depicted in (15), the spheroidal harmonic coefficients can be computed as follows:

$$u_{nm} = \sum_{k=0}^{\infty} \sum_{l=-m}^{m} \sum_{k=0}^{\infty} \sum_{l=-m}^{m} u_{kl} \delta_{kn} \delta_{lm}$$

$$= \sum_{k=0}^{\infty} \sum_{l=-m}^{m} \sum_{k=0}^{\infty} \sum_{l=-m}^{m} u_{kl} \frac{Q_{kl}^*(\frac{\phi}{2})}{Q_{kl}^2(\frac{\phi}{2})} \langle e_{nm}(\lambda, \phi) | e_{kl}(\lambda, \phi) \rangle_w$$

$$= \langle U(\lambda, \phi, u=b) | e_{kl}(\lambda, \phi) \rangle_w$$

$$= \frac{\alpha}{4\pi} \int_0^{2\pi} d\lambda \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \Lambda(\lambda, \phi, u=b) \, d\phi$$

where

$$\Lambda(\lambda, \phi, u) := \cos \phi \, e_{nm}(\lambda, \phi) \, U(\lambda, \phi, u)$$

(18) simplifies into

$$u_{nm} = \left\{ \begin{array}{ll} \frac{\alpha}{4\pi} \int_0^{2\pi} d\lambda \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \Lambda(\lambda, \phi, b) \, d\phi \, (-1)^{(n+m)} = 1 \\ (-1)^{(n+m)} = -1 \end{array} \right.$$

In this paper, all formed $U(\cdot, \cdot, \cdot)$ functions are symmetric with respect to $\phi$. Furthermore, the values that those functions take outside the surface of the reference spheroid are not of interest. In this context, (22) can be used to go into the transform domain of spheroidal harmonics, while (13) can be used to return back into the spatial domain.

IV. 3D WATERMARKING VIA SPHEROIDAL HARMONICS

In this section, the process of watermark embedding and detection in the transform domain is fully analyzed. For convenience, the presentation of the proposed watermarking scheme is broken down into the following discrete steps:

1) **Preprocessing:** The purpose of this step is to provide a unique normalization of the targeted 3D object (i.e. rotation, translation and uniform scaling). The robustness of the normalization process against various watermark attacks is of high significance, as it drastically affects the overall robustness of the entire watermarking scheme.

2) **Patch Generation And Surface Sampling:** The normalized mesh is divided into a number of continuous regions (patches), each of which is properly sampled, thus leading to a number of 2D functions (equal in number with that of the patches).

3) **Transform Domain Analysis:** Each of the 2D functions produced in the preceding step is mapped onto the surface of the reference spheroid $\mathbf{E}_r^{2\alpha,b}$ and subsequently analyzed into spheroidal coefficients, up to a certain degree and order.

4) **Watermark Embedding:** The watermark is embedded in the spheroidal coefficients of each patch, using a novel embedding scheme, which, in principle, resembles the ones using the multiplicative embedding rule. The watermark embedding strengths are different for the coefficients in low and middle pseudo-frequencies than are those in high pseudo-frequencies. The imperceptibility of the watermark is inherent, due both to the continuity of the spheroidal harmonic functions as well as to the multiplicative embedding rule adopted herein.
5) **Watermark Detection**: The input of the detection step is the 3D mesh under suspicion of copyright violation. The first three aforementioned steps are repeated here for this mesh, prior to holding the detection. The spheroidal coefficients of the produced patches are grouped into two sets: the set of low and middle pseudo-frequencies and the set of high pseudo-frequencies. The detection is held separately for each set, by forming a weighted sum from the elements of each set and by comparing it with a properly chosen threshold. Essentially, the two sets are treated as independent watermarks.

A detailed analysis of each of these steps is presented in the following subsections. Both embedding and detection require the knowledge of the owner’s private key. This private key is used to seed (find a starting point) a random number generator, in order to produce all needed pseudo-random entities.

### A. Preprocessing

One fundamental type of watermark attack, that any robust watermarking scheme should be able to withstand, is a basic affine (or similarity) transform comprised of 3D mesh translation, uniform scaling, rotation or a combination of the above. Thus, any scheme that does not embed and detect the watermark in an affine-invariant manner, including the one proposed herein, has to be provided with a unique orientation and scaling of the input mesh. This process is called normalization. A typical normalization procedure is the following [14]:

- **Translation**: The 3D mesh is translated in such a way that its center of mass becomes the origin of the coordinate system. A common mistake in this, rather trivial, process is to use the points of the 3D mesh, instead of the area of its faces, in order to compute the object’s center of mass.
- **Uniform Scaling**: The 3D mesh is uniformly scaled so as to fit, in its entirety, within a bounding volume. A typical bounding volume used is the unit sphere, centered around the mass center of the mesh.
- **Rotation**: The 3D mesh is reoriented by means of Principal Component Analysis (PCA). Again, the computation of the covariance matrix needed for the PCA is performed upon the barycenters of the faces of the 3D mesh, each weighted by the surface area of its corresponding face.

The robustness of the normalization scheme against various attacks defines, to a considerable degree, the corresponding robustness of the overall watermarking scheme against these attacks. The reason for this is that an incorrect normalization at the detection stage would de-synchronize the entire process, by forcing the detector to search for the watermark in different regions than the ones actually used at the embedding stage.

The normalization method previously described is indeed robust against basic affine transformations. Its robustness, however, significantly decreases if affine transformations are combined with attacks that offset the center of mass or de-synchronize the PCA (e.g. unbalanced re-sampling, smoothing, noise addition, etc.). In order to improve the robustness of the normalization procedure against this kind of combined attacks, the mesh is properly processed prior to holding the normalization.

The preprocessing stage we propose is based on the simple observation that an intensively smoothed version of the original mesh adequately converges to the corresponding smoothed version of the attacked mesh. The latter holds for attacks like noise addition, smoothing, connectivity attacks, unbalanced re-sampling or mesh simplification, that mainly alter high-frequency attributes of the mesh. Thus, since smoothing tends to eliminate high-frequency attributes, the smoothed versions of the former and the latter mesh converge to a rough representation of the bulk of the 3D object under consideration.

In this context, let us first assume that the polyhedral surface of the original 3D object \(O^{or}\) is expressed through a set of vertices \(V^{or}\) and a list of polygonal faces \(L^{or}\). At the beginning of the preprocessing stage (Fig.2), the original mesh \(O^{or}\) is intensively smoothed. The normalization procedure already described is subsequently applied to the smoothed mesh, leading to the normalized smoothed version of the mesh \(O^{sn}\). The exact same translation, rotation and scaling that the smoothed mesh has undergone is finally applied to the original mesh, resulting in the normalized original mesh \(O^{on}\). That ends the preprocessing stage.

With regard to the mesh smoothing, it can be performed by repetitive applications of one or more linear smoothing filters. A fairly general form of such filters follows [27]:

\[
x_{j}^{new} = x_{j}^{old} + \lambda \sum_{i \in N_{j}} w_{ij} (x_{i} - x_{j})
\]

where \(N_{j}\) is the neighborhood of point \(x_{j}\) and \(w_{ij}\) positive weights adding up to one. In case of connectivity or re-meshing attacks, the rate of convergence of repetitive smoothing iterations towards the desired smoothed mesh \(O^{sn}\) is strongly affected. An effective solution to this problem is to choose neighborhoods \(N_{j}\) of constant area.

In really sparse meshes, a mesh refinement by means of mesh subdivision is necessitated (surface lengths are going to be computed upon it). Classical subdivision schemes are the
Catmull-Clark refinement [28], [29], creating refined polyhedral surfaces with quadrilateral faces, and the Loop refinement process [30], [31], which divides every triangular face of the initial model into four new triangles in the refined model. The latter is preferred, as it preserves the initial structure of the mesh.

It should be stressed here that linear smoothing algorithms, including the most commonly used Laplacian and Taubin smoothing schemes [27], exclusively alter the vertices of the mesh onto which they apply. The connectivity information of the faces comprising the mesh is left intact, and hence the entire smoothing process can be viewed as a mapping $f : \mathbf{O}^{on} \rightarrow \mathbf{O}^{sn}$ of a point lying onto the surface of the original mesh $x_{on} \in \mathbf{O}^{on}$, into a new point lying onto the surface of the smoothed and normalized mesh $x_{sn} \in \mathbf{O}^{sn}$, namely

$$f(x_{on}) = x_{sn} \quad \forall x_{on} \in \mathbf{O}^{on} \quad (24)$$

Using this mapping, the sampling points can be chosen from the smoothed mesh and afterwards, by mapping those sampling points back to the original mesh, the sampled values can be extracted.

**B. Patch Generation And Surface Sampling**

In this step, the surface of the smoothed 3D mesh $\mathbf{O}^{sn}$ is divided into a number of $K \in \mathbb{N}$ continuous surface regions (patches). Each of these patches is sampled according to a distance metric (from the center of each patch). The distance metric and the sampling points are extracted from the smoothed version of the mesh $\mathbf{O}^{sn}$, while the sampled values are taken from the original mesh $\mathbf{O}^{on}$, by mapping the sampling points from $\mathbf{O}^{sn}$ back to $\mathbf{O}^{on}$. This choice is done to avoid de-synchronization of the sampling process at the detection stage, due to surface length and curvature changes caused by watermark attacks like noise addition or excessive mesh simplification. Any such type of attack severely alters the surface of the original mesh $\mathbf{O}^{on}$, but leaves relatively unaffected the intensively smoothed version of the mesh $\mathbf{O}^{sn}$.

The sampled values of each patch are going to be mapped onto the surface of an oblate spheroid in the following subsection. In this way, the sampling must be performed in two dimensions and hence each sampled point on the patch must be associated with two attributes uniquely characterizing it. Let $\ell$ and $\xi$ be the two sampling dimensions. Using this notation, a point belonging to the smoothed patch number $i$, assigned with an $\{\ell, \xi\}$ pair of attributes, will be denoted as $s x_{\ell, \xi}^i$. Note that the $i$ superscript consistently denotes, for the rest of the paper, values referring to patch number $i$, where $i \in [1, K]$. To simplify notation, the sampling process shall be described for one patch, by temporarily dropping the $i$ superscript in what follows.

At the start of the patch generation process, a number $L_M \in \mathbb{R}$ defining the desirable length of the patch is pseudo-randomly chosen. For each patch, a center point $s x_{0,0}$ is also pseudo-randomly chosen. For example, this point can be the first of all intersections with the smoothed mesh $\mathbf{O}^{sn}$ (the one closest to the center of mass) of a pseudo-randomly chosen ray $r^i$, starting from the center of mass of $\mathbf{O}^{sn}$. More complex schemes can be used for the selection of the patch centers, such as the one proposed in [13]. It simply suffices to preserve a consistent convention in regard to the choice of the centers, so as to preserve the blindness of the watermarking scheme.

The goal is to create a sampling rule providing a natural parameterization of the surface of the mesh, by minimizing distortions in lengths and angles between points and preserving inherent properties of the surface. Surface parameterizations fulfilling the aforementioned requirements are typically used in applications such as texture mapping and remeshing [32], [33], by employing the geodesic (shortest) paths between points on the surface. Additionally, the sampling rule must be robust against de-synchronization caused by watermark attacks. In this context, geodesics have recently been used in [13] for robust feature points selection on a 3D mesh.

Likewise, in the proposed scheme, a patch is built by progressively forming the isocurves of increasing geodesic length, on the $0$-genus two-manifold built around the center of the patch $s x_{0,0}$ (Fig.3(a)). The process is stopped when the maximum desired geodesic length $L_M$ is reached. Note that a $0$-genus representation of the surface is only locally required, i.e. around the patch centers (typically true), and is not a globally applicable constraint for the model being watermarked. Surazhsky et al. [34] recently presented accurate and computationally efficient ways to compute the isocurves and geodesic paths on a 3D mesh.

Clearly, the first sampling dimension $\ell$ can be the geodesic length that corresponds to any sampled point. In order to find a second sampling dimension, each point on an isocurve (i.e. with fixed geodesic length $\ell$) has to be labeled with a unique attribute.

Let $c_{\ell}$ be the positively oriented simple closed curve corresponding to the isocurve with geodesic length equal to $\ell$ units (Fig.3(b)), and $c_{\ell}(t)$ denote a parametric form of this curve. A novel sampling dimension shall be introduced, called integral angle $\xi$. In order to be able to define this second dimension, a reference point in each isocurve is needed, that corresponds to zero integral angle. For this reason, a point lying onto the outmost isocurve $c_{L_M,0}$ of the patch is pseudo-randomly chosen, and labeled with zero integral angle (i.e. $s x_{L_M,0}$ with the randomly selected point $s x_{L_M,0}$ (Fig.3(b)) is subsequently computed. This geodesic path intersects with each isocurve $c_{\ell}$ at a unique point, which is also labeled with zero integral angle, namely $s x_{\ell,0}$. The definition of the integral angle (Fig.3,(c)) can now be expressed through

$$\xi := 2\pi \frac{\int_{t_i}^{t_{i+1}} ||c_{\ell}(t)||dt}{l_{tot}} \quad (25)$$

where

$$l_{tot} := \int_{\ell} ||c_{\ell}(t)||dt \quad (26)$$

From (25), the boundedness of the integral angle readily becomes

$$0 \leq \xi < 2\pi \quad (27)$$

Thus, every point of the patch has been associated with a pair of $\{\ell, \xi\}$ values that uniquely describe it. In order to
proceed with the presentation of the sampling scheme, the i superscript that helps discriminate between values corresponding to different patches is reintroduced. As previously mentioned, the sampling values shall be taken from the original mesh \( O^{on} \). Up to now, the patch generation and the definition of the two sampling dimensions where done onto the surface of the smoothed mesh \( O^{on} \). Recall that, as depicted in (24), there is a one-to-one correspondence between a point in the original \( O^{on} \) and the smoothed \( O^{sn} \) mesh. Thus, for every point \( x_{\ell,\xi} \in O^{sn} \) there exists a corresponding point \( x_{\ell,\xi} \in O^{on} \).

The surface sampling can now be expressed through the following sampling rule:

\[
F'_u(\ell, \xi) = u(x'_{\ell,\xi}) \quad \forall \ell \in [0, L_M^2], \xi \in [0, 2\pi), i \in [1, K] 
\]

(28)

\[
F'_\lambda(\ell, \xi) = \lambda(x'_{\ell,\xi}) \quad \forall \ell \in [0, L_M^2], \xi \in [0, 2\pi), i \in [1, K] 
\]

(29)

\[
F'_\phi(\ell, \xi) = \phi(x'_{\ell,\xi}) \quad \forall \ell \in [0, L_M^2], \xi \in [0, 2\pi), i \in [1, K] 
\]

(30)

To sum up, at the end of this step, \( K \) new \( F'_u(\cdot, \cdot) \) functions have been created, by sampling the spheroidal height coordinate \( u \) along the surface of the patches. At the embedding stage, those functions shall be watermarked (i.e. new values will be assigned to the sampled spheroidal heights \( u \)). In order to be able to apply these changes to the spatial domain and form the watermarked mesh, the \( F'_\lambda(\cdot, \cdot) \) and the \( F'_\phi(\cdot, \cdot) \) functions are also created, by sampling the spheroidal longitude \( \lambda \) and the spheroidal latitude \( \phi \) that corresponds to any certain sampled point.

C. Transform Domain Analysis

In this section, the sampled functions \( F'_u(\cdot, \cdot) \) of the previous step are properly mapped onto the surface of the reference spheroid \( E_{\alpha,b}^2 \), for the purpose of continuing the embedding into the transform domain of spheroidal harmonics. Thus, a new function \( U^t(\cdot, \cdot, \cdot) \), constrained to the surface of \( E_{\alpha,b}^2 \), is created for every \( F'_u(\cdot, \cdot) \), using the following mapping rule:

\[
U^t(\lambda, \phi, u=b) = \begin{cases} 
F'_u(k^t(\phi), \lambda) & \forall \phi \in (0, \frac{\pi}{2}), \lambda \in [0, 2\pi) \\
F'_u(k^t(-\phi), \lambda) & \forall \phi \in (-\frac{\pi}{2}, 0), \lambda \in [0, 2\pi) 
\end{cases} 
\]

(31)

where, by definition,

\[
k^t(\phi) := (1 - \frac{2\phi}{\pi})L_M^2 \quad \forall i \in [1, K] 
\]

(32)

Given that \( U^t(\cdot, \cdot, \cdot) \) is known, the inverse transformation that provides the corresponding \( F'_u(\cdot, \cdot) \) function takes the following form

\[
F'_u(\ell, \xi) = U^t(\xi, h^t(\ell), u=b) \quad \forall \xi \in [0, 2\pi), \ell \in (0, L_M^2) 
\]

(33)

where, again by definition,

\[
h^t(\ell) = \frac{\pi}{2}(1 - \frac{\ell}{L_M^2}) \quad \forall i \in [1, K] 
\]

(34)

From the linear transformations in (32) and (34), it can be easily shown that the following inequalities hold:

\[
0 < k^t(\phi) < L_M^2, \quad \forall \phi \in (0, \frac{\pi}{2}), i \in [1, K] 
\]

(35)

\[
0 < k^t(-\phi) < L_M^2, \quad \forall \phi \in (-\frac{\pi}{2}, 0), i \in [1, K] 
\]

(36)

\[
0 < h^t(\ell) < \frac{\pi}{2}, \quad \forall \ell \in (0, L_M^2), i \in [1, K] 
\]

(37)

Clearly, now, every \( U^t(\cdot, \cdot, \cdot) \) function constrained onto the surface of \( E_{\alpha,b}^2 \) is symmetric around the \( xo\gamma \) plane of \( E_{\alpha,b}^2 \):

\[
U^t(\lambda, \phi, u) := U^t(\lambda, -\phi, u), \quad \forall (\lambda, \phi, u) \in E_{\alpha,b}^2 
\]

(38)

Intuitively, by the use of (31), each sampled function \( F'_u(\cdot, \cdot) \) is mapped onto the part of the reference spheroid \( E_{\alpha,b}^2 \) that lies above the \( xo\gamma \) plane and, subsequently, the values of the upper part are mirrored to those of the lower part. Not only does that provide \( C^0 \) continuity for each function \( U^t(\cdot, \cdot, \cdot) \), but it also simplifies the application of the forward spheroidal harmonics transformation (approximately half of the spheroidal coefficients become zero due to the symmetry).

Due to this property, by applying the forward harmonics transformation depicted in (22) for every \( U^t(\cdot, \cdot, \cdot) \) function, and by analyzing up to a finite degree and order \( N_H \), new sets of spheroidal coefficients \( u^t = \{u^t_{nm} \in [0, N_H] \}, n \in [-n, n] \} \) are formed. The coefficients of each \( u^t \) set can be separated into two independent subsets, one for those corresponding to low and middle pseudo-frequencies and one for those corresponding to high pseudo-frequencies.
The discrimination between coefficients of different subsets can simply be done through their degree and order indices. These indices can be grouped into the following sets:

\[
L_0 := \{(n,m) | n \in [1,N_H], m \in [-n,n]\} \quad (39)
\]

\[
L_L := \{(n,m) | n \in [1,N_L], m \in [-n,n]\} \quad (40)
\]

\[
L_H := \{(n,m) | n \in [N_L + 1, N_H], m \in [-n,n]\} \quad (41)
\]

where \(N_L\) is the maximum degree for a coefficient belonging to the set of low and middle pseudo-frequencies. Note that the pair \((0,0)\) that corresponds to the indices of the DC coefficient does not belong to any of the previous sets. That is so because no alterations are allowed to the DC coefficients of the various patches, during the watermark embedding stage, since any such change would offset the entire watermarked patch from the surface of the 3D mesh.

### D. Watermark Embedding

The input of the embedding process is the 3D object to be watermarked and the owner’s private key, which ensures the secrecy of the watermark. After the completion of the first 3 steps previously analyzed, the spheroidal coefficient sets \(u^i\) are formed. In this step, the watermark data are generated and embedded within the \(u^i\) sets. After that, the changes enforced to these coefficients are inverted back to the spatial domain and, by altering the original mesh data, the watermarked 3D object is formed (Fig.4).

As a first step in this direction, for every coefficient \(u_{nm}^i \in u^i\), the owner’s private key is used in order to generate a pseudo-random number \(w_{nm}^i\), uniformly distributed in the interval \([-1,1]\). These numbers constitute the watermark data that will be embedded in the chosen patches. Now, using the same private key, a sequence of real numbers \(c_{nm} \in \mathbb{R}\) is also generated. These numbers constitute the embedding strengths that correspond to any certain coefficient of degree \(n\) and order \(m\). Using the formerly generated embedding strengths, we define a set of positive real numbers \(d_{nm}^j\), \(\forall (n,m) \in L_0\) through the following iteration:

\[
d_{nm}^j := \begin{cases} t_{min} \cdot \frac{1+c_{nm}}{1-c_{nm}} & \text{if } j = 0 \\ t_{min} \cdot \frac{1-c_{nm}}{1+c_{nm}} & \text{if } j \in [1, +\infty) \end{cases}
\]

(42)

where the embedding threshold \(t_{min}\) is a positive real number. Any coefficient with absolute value below that threshold (i.e. \(|u_{nm}^i| \leq t_{min}\)) will be excluded from both watermark embedding and detection. Care must be taken so as not to set the \(t_{min}\) threshold too high, since that will decrease the overall capacity of the watermarking scheme, by automatically excluding most of the coefficients from the embedding process (especially those with high degree and order, that typically have small values).

The afore defined numbers can now be used to create a corresponding partition of the positive real line \(P_{nm} = \{P_{nm}^0, P_{nm}^1, P_{nm}^2, \ldots\}\), with the subspaces \(P_{nm}^j \subseteq \mathbb{R}\) defined through

\[
P_{nm}^j := \begin{cases} \{0, d_{nm}^0, d_{nm}^{j-1}, d_{nm}^j\}, & \text{if } j = 0 \\ \{d_{nm}^{j-1}, d_{nm}^j\}, & \text{if } j \in [1, +\infty) \end{cases}
\]

(43)

Next, for every \(d_{nm}^j\) value, a corresponding \(q_{nm}^j\) value is defined by use of

\[
q_{nm}^j := d_{nm}^j(1-c_{nm}) \quad \forall j \in [0, +\infty) \quad (44)
\]

From the definition of \(q_{nm}^j\) and \(d_{nm}^j\), the following equality can be derived

\[
d_{nm}^j - q_{nm}^j = q_{nm}^{j+1} - d_{nm}^j \quad \forall j \in [0, +\infty) \quad (45)
\]

from which it is inferred that \(d_{nm}^j\) is the center of the subspace \([q_{nm}^j, q_{nm}^{j+1}]\).

We can now define an operator \(Q_{nm}[\cdot]\) applicable to any real number and such that

\[
Q_{nm}[z] := \left\{ \begin{array}{ll} z & \text{if } z \in [0, t_{min}], \\
\frac{z}{|z|} \cdot q_{nm}^1 & \text{if } |z| > t_{min} \end{array} \right. \quad \forall |z| \in P_{nm}^0, j \neq 0 \quad (46)
\]

The idea behind this operator is that it quantizes any input value \(z\) to the nearest \(q_{nm}^j\) value (since, as (45) depicts, both edges of every subspace \(P_{nm}^j\) are lying in the center of two successive \(q_{nm}^j\) values), but leaves unaffected any value that lies in the first subspace of the partition \(P_{nm}^0\), i.e. any value below the embedding threshold \(|z| \leq t_{min}\). Note that the \(n\) and \(m\) subscripts in the \(Q_{nm}[\cdot]\) operator are just there to clarify the partition \(P_{nm}\) upon which it applies.

Finally, by taking into account the fact that

\[
q_{nm}^j \in P_{nm}^j, \quad \forall j \in [0, +\infty), \quad (n,m) \in L_0 \quad (47)
\]

it can be shown that

\[
Q_{nm}[q_{nm}^j] = q_{nm}^j, \quad \forall j \in [1, +\infty), \quad (n,m) \in L_0 \quad (48)
\]

Now that the mathematical formulation of the tools used in the embedding process has concluded, we can proceed with the presentation of the watermark embedding scheme. The watermarked coefficients \(w_{nm}^i\) are computed through the following equation:

\[
w_{nm}^i = \frac{Q_{nm}[w_{nm}^i]}{1-\frac{c_{nm}w_{nm}^i}{1+c_{nm}w_{nm}^i}} \quad \forall i \in [1,K], \quad (n,m) \in L_0 \quad (49)
\]

It can be shown (see Appendix B) that the aforementioned embedding rule is multiplicative. This property, in combination with the natural parameterization of the mesh at the sampling stage, provides high imperceptibility of the watermark, as the scheme automatically adapts to the local attributes of each mesh (curvature variations, texture, etc.).

From (49) and (48) an important property of the watermarked coefficients can be derived

\[
w_{nm}^i(1-c_{nm}w_{nm}^i) = Q_{nm}[w_{nm}^i(1-c_{nm}w_{nm}^i)] = Q_{nm}[w_{nm}^i] \quad (50)
\]

The changes that the coefficients have undergone must now be inverted back to the spatial domain. Recall that each set of coefficients \(u^i\) was formed by the application of the forward spheroidal harmonics transformation upon the function \(U^i(\cdot, \cdot, \cdot)\). One way to produce the new watermarked functions \(wU^i(\cdot, \cdot, \cdot)\) is to apply inverse spheroidal harmonics transformation to the watermarked coefficients \(w_{nm}^i\). But that would lead to a poor reconstruction and increased perceptibility of the watermark, since the forward transform was
performed up to a finite degree and order. To avoid this kind of artifacts, by making use of the linearity of the transformation, the inversion can be held as follows:

\[
u U^i(\lambda, \phi, u=b) = \sum_{(n,m) \in L_0} \left( u'_{nm} - u_{nm} \right) e_{nm}(\lambda, \phi) + \sum_{n=0}^{\infty} \sum_{m=-n}^{n} u_{nm} e_{nm}(\lambda, \phi) = \sum_{(n,m) \in L_0} \left( u'_{nm} - u_{nm} \right) e_{nm}(\lambda, \phi) + U^i(\lambda, \phi, u=b)\] (51)

Therefore, only the differences between the watermarked and the original coefficients are inverted instead, and the results are added to the original \(U^i(\cdot, \cdot, \cdot)\) functions.

Given that the new watermarked functions \(wU^i(\cdot, \cdot, \cdot)\) are computed, the watermarked sampled functions \(wF_u^i(\cdot, \cdot, \cdot)\) can be formed by applying (33). The \(F_u^i(\cdot, \cdot, \cdot)\) functions remain unchanged, as the watermark is exclusively casted into the spheroidal height coordinate \(u\). The watermarked object can now be readily reconstituted from these functions. Thus, while at the sampling stage a sampled point could be expressed, in terms of Jacobi ellipsoidal coordinates, as

\[
x_{i,\xi} = \{ F_u^i(\ell, \xi), F_\lambda^i(\ell, \xi), F_\phi^i(\ell, \xi) \} \] (52)

the corresponding point belonging to the watermarked mesh now becomes

\[
wx_{i,\xi} = \{ wF_u^i(\ell, \xi), F_\lambda^i(\ell, \xi), F_\phi^i(\ell, \xi) \} \] (53)

\[E. \text{ Watermark Detection}\]

The objective, at the side of the detection process, is the estimation, with the highest possible probability of success, of whether a certain input 3D object has or has not been watermarked using a specific private key. A visual representation of the various stages of the detection process is presented in Fig. 5. A detailed presentation follows.

To condense notation, in what follows, \(H_1\) and \(H_0\) denote the hypotheses that the watermark under investigation does or does not exist in the suspect object respectively. Similarly, \(A_1\) and \(A_0\) denote the events that the detector estimates that the watermark under investigation does or does not exist in the suspect object respectively. To evaluate the performance of the detector, the probabilities of the following events are usually incorporated:

- \(The \ false \ alarm \ probability\), which is the probability of hypothesizing, during the detection, the existence of a watermark in an object that does not contain it, namely \(Pr(A_1|H_0)\).
- \(The \ false \ rejection \ probability\), which is the probability of hypothesizing the absence of a watermark in an object that does indeed contain it, namely \(Pr(A_0|H_1)\).

It is true that the detector used in the proposed watermarking scheme allows a deterministic approach for the verification of this hypothesis. In this context, if we denote as \(u'_{nm}\) the spheroidal coefficients of the 3D mesh under suspicion, according to (50), the 3D mesh is indeed watermarked with the use of this private key, if and only if

\[
u u'_{nm} = q u_{nm}, \quad \forall i \in [1, K], \quad (n,m) \in L_0 \] (54)

where

\[
q u_{nm}^i := \frac{Q_{nm}[u_{nm}^i (1 - c_{nm} w_{nm}^i)]}{1 - c_{nm} w_{nm}^i} \] (55)

For a detector which hypothesizes based on whether (54) holds true or not, the false alarm and false rejection probabilities are both zero:

\[
Pr(A_1|H_0) = 1 - Pr(A_0|H_0) = 0 \] (56)
\[
Pr(A_0|H_1) = 1 - Pr(A_1|H_1) = 0 \] (57)

This detector can only work when no watermark attack exists. Any watermark attack alters the watermarked coefficients, and the equality in (50) does no longer hold. Still, if the noise added by the attack is relatively low, the coefficients of the attacked 3D mesh should be close, in value, to the ones given by (50). Based on this observation, the coefficients \(u'_{nm}\) of the suspect mesh are decomposed into two parts:

\[
u u'_{nm} = q u_{nm} + z_{nm}, \quad \forall i \in [0, +\infty), \quad (n,m) \in L_0 \] (58)

Ideally, the noise part \(z_{nm}\) of every coefficient \(u'_{nm}\) must be zero (in absence of attacks and presence of the watermark...
under search). In case of any attack, the noise parts $z_{nm}^i$ are generally non-zero. But the $z_{nm}^i$ parts can also be non-zero simply when the watermark being searched for is absent (event $H_0$). The detection can only be held, in this case, by means of statistical hypothesis testing, since the distribution of the noise is generally unknown to the side of a blind detector.

A typical approach to address this problem is to firstly apply a functional over the coefficients $u_{nm}^i$ and then make the decision, as to whether the 3D object is watermarked or not, by comparing the output of the functional (detection ratio) with that of a properly chosen threshold $T$. Two independent detection ratios are formed here, one for the set $L_L$ of low and middle pseudo-frequencies, and another for the set $L_H$ of high pseudo-frequencies. In both cases, the detection ratio is computed according to

$$D_F := \frac{1}{J_K} \sum_{(n,m,i) \in S_F} |u_{nm}^i| \cdot v_{nm}^i \quad (59)$$

where

$$v_{nm}^i = 1 - \frac{|z_{nm}^i (1 - c_{nm})|}{|q u_{nm}^i \cdot c_{nm}|} \quad (60)$$

$$J_K = \sum_{(n,m,i) \in S_F} |u_{nm}^i| \quad (61)$$

and

$$S_F = \{(n, m, i): (n, m) \in L_F, i \in [1, K], q u_{nm}^i \notin I_{nm} \} \quad (62)$$

The subscript in $D_F$ and $S_F$ can be either $L$, for the case of low and middle pseudo-frequencies, or $H$, for the case of high pseudo-frequencies. This convention, with regard to the $F$ subscript, will be maintained consistently throughout this paper.

In Appendix C, it is proven that the normalized quantization residuals $v_{nm}^i$ are uniformly distributed within the interval $[0,1]$. This is true when event $H_0$ holds (i.e. absence of watermark). On the contrary, when event $H_1$ holds, the distribution of the normalized quantization residuals $v_{nm}^i$ changes. If no watermark attack is present, they all assume maximum value (i.e. $v_{nm}^i = 1$). In this case, if a certain watermark attack is afterwards applied to the watermarked mesh, it will distribute its energy along the various spheroidal coefficients, and hence the absolute values of the affected $v_{nm}^i$ residuals will, in general, be smaller. The intuition behind the conception of the new multiplicative embedding framework is based on the fact that it is highly unlikely to have the energy of a certain watermark attack properly distributed so as to equally affect all $v_{nm}^i$ values. Thus, in order to effectively attack a watermark, significant distortions must be imposed to the geometry of the mesh. The embedding strengths $c_{nm}$ simply increase the quantization intervals. Increasing the $c_{nm}$ strengths maintains the assumption of uniform distribution in the absence of watermark, but this time requires increased energy from a watermark attack, in order to have the same impact on the detector.

It can be shown (see Appendix A) that the detection ratio $D_F$ belongs to the interval

$$0 < D_F \leq 1$$

Thus, for fixed $u_{nm}^i$ values, the sum in (59) maximizes when every $z_{nm}^i$ is zero

$$D_F^{MAX} = \frac{1}{J_K} \sum_{(n,m,i) \in S_F} |u_{nm}^i| = 1$$

From the latter it is inferred that the probability of $D_F$ being less than $D_F^{MAX}$ is

$$Pr(D_F < D_F^{MAX}) = 1$$

The detection ratio $D_F$ defined in (59) bears the following properties:

- The contribution of a certain coefficient $u_{nm}^i$ to the detection ratio is directly proportional to its absolute value, and hence to its reliability (as higher values tend to be more insensitive to attacks than lower ones).
- The smaller the quantization residual $z_{nm}^i$ that corresponds to a $u_{nm}^i$ coefficient, the higher is its contribution to the detection ratio.

The decision, at the side of the detector, as to whether the watermark does or does not exist within the suspect
mesh, relies on the mere comparison of each $D_F$ with a corresponding threshold $T_F$. The definition of the mutually exclusive events $A_0$ and $A_1$ henceforth becomes

\begin{align*}
A_0 & := \{(D_F < T_F|H_0) \cup (D_F < T_F|H_1)\} \quad (66) \\
A_1 & := \{(D_F \geq T_F|H_0) \cup (D_F \geq T_F|H_1)\} \quad (67)
\end{align*}

Apparently, both false alarm and false rejection probabilities are now functions of the chosen thresholds:

\begin{align*}
P_{fa}^F(T_F) & := Pr(A_1|H_0) = Pr(D_F \geq T_F|H_0) \\
& = \int_{T_F}^{\infty} f_{D_F|H_0}(y) \, dy \quad (68) \\
P_{fr}^F(T_F) & := Pr(A_0|H_1) = Pr(D_F < T_F|H_1) \\
& = \int_{T_F}^{\infty} f_{D_F|H_1}(y) \, dy \quad (69)
\end{align*}

where $f_{D_F|H_0}$ and $f_{D_F|H_1}$ are the probability distribution functions of the detection ratio $D_F$. Note that the knowledge of these distributions is a function of the noise elements $z_{nm}$, which in turn are, generally speaking, unknown to the side of the blind detector. Thus, the performance of the detector under various watermark attacks (i.e. for various noise patterns) can only be experimentally tested.

It should be stressed here that, if hypothesis $H_1$ holds and no watermark attack is present, all noise elements $z_{nm}$ assume zero value and consequently, according to (64), all detection ratios are maximized

\begin{equation}
D_F = D_F^{MAX}, \quad \forall F \in \{L, H\} \quad (70)
\end{equation}

And since, in this case, $T_F$ is always less or equal to $D_F^{MAX}$, it follows that

\begin{equation}
P_{fr}^F(T_F) = Pr(D_F^{MAX} < T_F|H_1) = 0 \quad (71)
\end{equation}

independently of the choice of the $T_F$ threshold. That is, there is zero probability of failing to detect an existing watermark, when no watermark attack is present. This is so because the detector presented herein exclusively admits certain input values as valid, which can be extracted under the mere knowledge of the correct private key (see (50)). The contribution of any coefficient deviating from the valid values, to the formation of the detection ratio, is properly weighted according to its distance from the closest valid value.

V. EXPERIMENTAL RESULTS

In this section, the robustness of the proposed watermarking scheme against various watermark attacks is presented. For the evaluation of the performance of the algorithm, the Receiver Operating Characteristic (ROC) curves have been employed, a standard technique for evaluating statistical detection schemes.

A ROC curve is produced by plotting the false alarm probability $P_{fa}^F(T_F)$ against the false rejection probability $P_{fr}(T_F)$, for varying threshold $T_F$. The point of a ROC curve where $P_{fr} = P_{fa}$ is called Equal Error Rate (EER), and will be used as a quantitative figure for the succinct review of the experimental results.

The various parameters involved in the proposed scheme where experimentally chosen as a good compromise between speed and robustness. In detail, all the experiments presented in this section are obtained for $K = 7$ patches per watermarking session, a low frequency bound set to $N_L = 5$, a high frequency bound equal to $N_H = 12$ and fixed embedding strengths $c_{nm} = 0.01$, \forall (n,m) \in L_0$. For the derivation of a ROC curve that corresponds to a certain 3D model and watermark attack, 1000 groups of $K$ patches have been pseudo-randomly produced from the attacked model, with the embedding and detection being held for a minimum of 1000 different watermark keys per group. For the estimation of extremely low EER values, a Gaussian distribution is assumed both for the probability density functions of $P_{fa}$ and $P_{fr}$.

In Fig.8 the results of fitting a Gaussian in the $P_{fa}$ data corresponding to the Horse [35] model are presented.

In Table I, the EER points of the ROC curves that correspond to 3D models Horse [35], Igea [35], Bunny [35] and Fandisk [36] (CAD model), for various watermark attacks, are presented. More specifically, attack A corresponds to similarity attacks. Attacks B through D correspond to white noise addition. For example, 1% noise corresponds to random displacement of all vertices of the 3D mesh, with maximum displacement equal to 0.01 times the maximum distance of a point of the original 3D mesh from its center of mass. Attacks E through G correspond to Taubin smoothing attacks, with parameters $\lambda = 0.6|k_{PB} = 0.1$ (see [27]). The number of iterative applications of the Taubin smoothing is explicitly mentioned in each row. Attacks H and I correspond to mesh simplification. For example, 0.2 mesh simplification corresponds to the removal of 20% of the vertices of the 3D mesh. Finally, attacks J and K correspond to a combination of some of the previous attacks. Note that, for the derivation of the results presented in Table I, it is implied that similarity transformations are combined with any of the aforementioned attacks.

The ability of the detector to work even with a limited number of input samples (i.e. small value for $N_L$) significantly increases the robustness of the proposed scheme. The EER
TABLE I

ROBUSTNESS OF THE PROPOSED SCHEME AGAINST VARIOUS ATTACKS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A. No Attack</td>
<td>$10^{-31}$</td>
<td>$10^{-35}$</td>
<td>$10^{-29}$</td>
<td>$10^{-21}$</td>
</tr>
<tr>
<td>B. Noise 0.3%</td>
<td>$2 \times 10^{-15}$</td>
<td>$2 \times 10^{-13}$</td>
<td>$3 \times 10^{-13}$</td>
<td>$5.1 \times 10^{-5}$</td>
</tr>
<tr>
<td>C. Noise 0.7%</td>
<td>$3.0 \times 10^{-9}$</td>
<td>$1.4 \times 10^{-6}$</td>
<td>$2.3 \times 10^{-5}$</td>
<td>$2.07 \times 10^{-2}$</td>
</tr>
<tr>
<td>D. Noise 0.9%</td>
<td>$1.1 \times 10^{-5}$</td>
<td>$1.2 \times 10^{-3}$</td>
<td>$8.0 \times 10^{-4}$</td>
<td>$3.64 \times 10^{-2}$</td>
</tr>
<tr>
<td>E. Taubin 2 it.</td>
<td>$7.9 \times 10^{-8}$</td>
<td>$1.7 \times 10^{-7}$</td>
<td>$3.8 \times 10^{-5}$</td>
<td>$9.73 \times 10^{-2}$</td>
</tr>
<tr>
<td>F. Taubin 5 it.</td>
<td>$1.2 \times 10^{-5}$</td>
<td>$7.1 \times 10^{-3}$</td>
<td>$2.67 \times 10^{-2}$</td>
<td>$4.39 \times 10^{-1}$</td>
</tr>
<tr>
<td>G. Taubin 12 it.</td>
<td>$4.21 \times 10^{-2}$</td>
<td>$4.26 \times 10^{-2}$</td>
<td>$1.01 \times 10^{-1}$</td>
<td>$4.92 \times 10^{-1}$</td>
</tr>
<tr>
<td>H. Simplif. 0.2</td>
<td>$2 \times 10^{-19}$</td>
<td>$4 \times 10^{-22}$</td>
<td>$7 \times 10^{-16}$</td>
<td>$8 \times 10^{-17}$</td>
</tr>
<tr>
<td>I. Simplif. 0.6</td>
<td>$7 \times 10^{-17}$</td>
<td>$6 \times 10^{-13}$</td>
<td>$3 \times 10^{-10}$</td>
<td>$7.1 \times 10^{-7}$</td>
</tr>
<tr>
<td>J. Attacks H+B</td>
<td>$3 \times 10^{-13}$</td>
<td>$2 \times 10^{-12}$</td>
<td>$8.1 \times 10^{-9}$</td>
<td>$4.2 \times 10^{-3}$</td>
</tr>
<tr>
<td>K. Attacks H+E</td>
<td>$5.2 \times 10^{-8}$</td>
<td>$9.9 \times 10^{-6}$</td>
<td>$2.77 \times 10^{-5}$</td>
<td>$3.1 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Fig. 6. Watermark attacks. Column Description: (a) original model, (b) model after 0.6 simplification, (c) model after 0.7% uniform noise addition and (d) model after 5 iterations of Taubin filter. Row description: First row corresponds to Horse model. Models Igea, Bunny and Fandisk (CAD model) follow.

values of Table I are all taken from the ROC curves that correspond to the low frequency detection ratio $D_L$. The embedding in the low frequency set has the following advantages:

- Low frequency coefficients typically have higher value and, consequently, higher robustness against most common attacks, that typically alter high frequency attributes of the mesh.
- It provides higher imperceptibility, since the human eye is more sensible to high frequency variations.
- The detection in the low frequency coefficients is the least sensitive to translation and rotation of the patches, caused by de-synchronization of the mesh registration stage, due to severe watermark attacks.

The EER values for the high frequency detection ratio $D_H$ where comparable (yet inferior) with the corresponding EER values of $D_L$ only for attacks A and H. Still, the high frequency set was not discarded from the watermark embedding and detection process. Its presence has the following advantages:

- It provides a reliable, yet computationally expensive
way to compensate for small de-synchronizations at the registration stage. This can be done by slightly moving the patch center around its vicinity, until the detection for the high frequency set is maximized.

- It allows the detector to make more complex decisions. For example, if the detection ratio for the low frequency set is high, while the detection ratio for the high frequency set is relatively low, the detector can hypothesize that the watermark does exist within the suspect mesh, which, at the same time, has undergone a high-frequency type of attack (e.g. noise addition).

By setting the number of patches per session $K$ to a higher value, the capacity of the embedding scheme is increased. Different patches are allowed to overlap, further increasing maximum attainable capacity, but in this case the detection must be performed in the exact reverse order than that of the embedding (with respect to the analysis of each patch).

Let us further clarify this through an example. Suppose only two patches are chosen on an embedding session, namely patch $A$ and patch $B$, that partly intersect in $C = A \cap B$. Patch $A$ is firstly formed and watermarked, leading to watermarked patch $A^*$ and part $C^*$. Secondly, patch $B$ is formed (now consisting of $C^*$) and watermarked, leading to $B^*$ and $C^{**}$. At the detector side, using the private key, the exact reverse order is followed by firstly forming patch $B'$, consisting in part by the overlapping section $C''$. The detection is held for this patch and the (potentially) embedded sequence that corresponds to the private key is reproduced and subtracted from $C''$, leading to $C'$. Secondly, patch $A'$ is formed, now consisting of $C'$. The detection is held for this patch and the decision is made, regarding the existence of the watermark under inspection.

The robustness of the watermarking scheme is chiefly due to the watermark embedded in the low pseudo-frequency coefficients of the set $L$, thus rendering the choice of the low $N_L$ and high $N_H$ frequency bounds highly crucial. These parameters must be kept as low as possible, while still providing an adequate number of coefficients in each set. The higher the degree and order of a coefficient, the lesser is, in general, its value and the higher its sensitivity to watermark attacks like smoothing and noise addition (that mainly alter the high frequencies of the mesh).

The values used for the watermarking strengths $c_{nm} = 0.01$ were experimentally chosen so as to ensure watermark imperceptibility, while still providing adequate robustness, even under severe watermark attacks. The watermark remains imperceptible for embedding strengths up to $c_{nm} = 0.02$ (Fig.7). Further increasing the embedding strengths for the coefficients of the low pseudo-frequencies set $L_L$, requires visual masking, since, in this case, the border of the patches tends to separate from the surface of the mesh. For a quantitative estimation of the watermark imperceptibility, the median MRMS [11] value of 20 tests is presented in Table II, for various 3D models and embedding strengths. The Metro tool [37] has been used to

Fig. 7. Watermarked models. Column Description: (a) original model, (b) model watermarked with embedding strengths $c_{nm} = 0.01 = 1\%$, and (c) model watermarked with embedding strengths $c_{nm} = 0.02 = 2\%$. 

(a) (b) (c)
TABLE II  
QUANTITATIVE ESTIMATION OF THE WATERMARK IMPERCEPTIBILITY  
USING THE MRMS MEASURE

<table>
<thead>
<tr>
<th>Strength</th>
<th>MRMS for Horse</th>
<th>MRMS for Igea</th>
<th>MRMS for Bunny</th>
<th>MRMS for Fandisk</th>
</tr>
</thead>
<tbody>
<tr>
<td>c_{nm}</td>
<td>6.5 x 10^{-5}</td>
<td>1.8 x 10^{-5}</td>
<td>3.2 x 10^{-5}</td>
<td>5.7 x 10^{-5}</td>
</tr>
<tr>
<td>0.006</td>
<td>1.2 x 10^{-4}</td>
<td>2.7 x 10^{-5}</td>
<td>5.9 x 10^{-5}</td>
<td>1.1 x 10^{-4}</td>
</tr>
<tr>
<td>0.01</td>
<td>2.2 x 10^{-4}</td>
<td>5 x 10^{-5}</td>
<td>1.3 x 10^{-4}</td>
<td>1.9 x 10^{-4}</td>
</tr>
</tbody>
</table>

TABLE III  
COMPARISON OF THE PROPOSED WATERMARKING SCHEME WITH THE TWO METHODS IN [11], FOR THE BUNNY MODEL [35]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Simpl. 0.3</td>
<td>1 x 10^{-7}</td>
<td>9 x 10^{-13}</td>
<td>2 x 10^{-15}</td>
</tr>
<tr>
<td>Simpl. 0.5</td>
<td>7 x 10^{-5}</td>
<td>3 x 10^{-12}</td>
<td>5 x 10^{-13}</td>
</tr>
<tr>
<td>Simpl. 0.7</td>
<td>3 x 10^{-2}</td>
<td>4 x 10^{-8}</td>
<td>7 x 10^{-8}</td>
</tr>
<tr>
<td>Noise 0.1%</td>
<td>9 x 10^{-6}</td>
<td>1 x 10^{-10}</td>
<td>3 x 10^{-12}</td>
</tr>
<tr>
<td>Noise 0.2%</td>
<td>4 x 10^{-4}</td>
<td>2 x 10^{-7}</td>
<td>1 x 10^{-8}</td>
</tr>
<tr>
<td>Noise 0.3%</td>
<td>4 x 10^{-2}</td>
<td>2 x 10^{-2}</td>
<td>8.5 x 10^{-4}</td>
</tr>
</tbody>
</table>

Fig. 9. Plot of the false alarm probability $P_{fa}(T_F)$ against the detection threshold $T_F$, for the 3D models used in our experiments.

A comparison between the proposed watermarking scheme and the two methods presented in [11], in terms of robustness against simplification attacks and binary noise addition, is presented in Table III (the results are taken straight from [11]). The comparison results correspond to the Bunny model, while the MRMS value has been kept constant and equal to $0.41 \times 10^{-4}$ for all tests, by iteratively varying the embedding strengths $c_{nm}$. The elevated performance of the proposed scheme can be effectively attributed to the robust sampling of relatively large regions in the smoothed mesh, in combination with the use of a novel embedding scheme upon orthonormal (statistically independent) coefficients.

Typical running times of our implementation of the algorithm were between 4 and 11 seconds, for a watermark embedding session, and between 3 and 9 seconds for a watermark detection session, depending on the size of the input 3D mesh. The algorithm was time-tested on a personal computer running on Windows XP® equipped with an Intel Centrino Duo® T2050 1.6GHz processor.

VI. Conclusions

We have presented a new approach to blind 3D watermarking, with high imperceptibility and robustness against similarity transformations, connectivity attacks, mesh simplification, unbalanced re-sampling, smoothing and noise addition. We introduced a novel surface parameterization scheme that preserves the natural characteristics of the surface. We also presented a novel embedding scheme and a compatible detecting scheme along with it.

The natural surface parameterization in combination with the multiplicative embedding rule provide high watermark imperceptibility, as the embedding scheme automatically adapts to the local attributes of the various patches. The proposed detector can work even with a very limited number of input coefficients, thus allowing the detection to be held independently for the set of low and high frequency coefficients respectively. In this way, the detection for the low frequency set is not hampered by the inclusion of the highly sensitive high frequency coefficients.

Regrettably, the scheme fails under cropping attacks, due to strong dependence both on the global registration of the mesh, for the proper selection of the patch centers, and on the 3D object’s mass center, for the computation of the u-coordinate that corresponds to any sampled point. A possible solution to this problem could be the use of a method similar with that presented in [13], applied onto the smoothed version of the mesh, to preserve robustness against noise and smoothing.

Appendix A

DETECTION RATIO UPPER AND LOWER BOUND

Let $q_{nm}^k$ be the quantized value that corresponds to $q_{nm}^iu_i$, namely $q_{nm}^i = q_{nm}^k$, $k \in [0, +\infty)$. Using (42), (43), and (44), the distance of $q_{nm}^k$ from the upper and lower bound of the $P_{nm}$ subspace becomes

$$|q_{nm}^k - d_{nm}^k| = d_{nm}^k c_{nm}$$  \hspace{1cm} (72)

and

$$|q_{nm}^k - d_{nm}^{k-1}| = d_{nm}^k \frac{1 - c_{nm}}{1 + c_{nm}} c_{nm} < d_{nm}^k c_{nm}$$  \hspace{1cm} (73)

respectively. From (58), (72), and (73), and by taking into account that $u_{nm}^i \in P_{nm}$, the following inequality can be derived:

$$0 \leq |z_{nm}^i| < |q_{nm}^k c_{nm}| = \frac{|q_{nm}^k c_{nm}|}{|1 - c_{nm}|} < \frac{|q_{nm}^i c_{nm}|}{|1 - c_{nm}|}$$  \hspace{1cm} (74)

which in turn is equivalent to

$$0 < \left(1 - \frac{|z_{nm}^i (1 - c_{nm})|}{|q_{nm}^i c_{nm}|}ight) \leq 1$$  \hspace{1cm} (75)

From the definition of the detection ratio in (59) and the inequality in (75), the upper and lower bounds of the detection ratio readily become

$$0 < D_F \leq 1$$  \hspace{1cm} (76)
**APPENDIX B**

**PSEUDO-MULTIPlicative EMBEDDING RULE**

The most typically used multiplicative embedding rule is

\[ x_w = x(1 + \gamma w) \]  

(77)

or equivalently

\[ x_w - x = x\gamma w = x_w \frac{\gamma w}{1 + \gamma w} \]  

(78)

where \( x_w \) and \( x \) are the watermarked and original values respectively, \( \gamma \) the embedding strength, and \( w \) the watermark data. By assuming that \( w \) is uniformly distributed in the interval \([-1, 1]\) and that \( \gamma \) is positive, we can express the multiplicative embedding rule as

\[ -\frac{|x_w\gamma|}{|1 - \gamma|} \leq (x - x_w) \leq \frac{|x_w\gamma|}{|1 - \gamma|} \]  

(79)

In the proposed scheme, \( x \) corresponds to \( u_{nm}^i \) and \( x_w \) corresponds to \( w_u u_{nm}^i \). From (72), (73) and (74) it can be shown that

\[ -\frac{|w u_{nm} c_{nm}|}{|1 - c_{nm}|} < (u_{nm} - w u_{nm}^i) < \frac{|w u_{nm} c_{nm}|}{|1 - c_{nm}|} \]  

(80)

Note that (80) has the same form with (79). That is, the embedding scheme is multiplicative with embedding strength equal to

\[ \gamma = c_{nm} \]  

(81)

**APPENDIX C**

**DISTRIBUTION OF NORMALIZED QUANTIZATION RESIDUALS**

Let once again \( q_{nm}^k \) be the quantized value that corresponds to \( q_{nm}^i \). From (43), (47) and (58) it follows that

\[ z_{nm}^i \in \left[ q_{nm}^{k-1} - q_{nm}^i, q_{nm}^{k-1} - q_{nm}^i \right] \]  

For adequately dense quantization intervals (i.e. small embedding strengths \( c_{nm} \)), it is fairly reasonable to assume that each quantization residual is uniformly distributed within its interval

\[ z_{nm}^i \sim U(d_{nm}^{k-1} - q_{nm}^i, d_{nm}^{k-1} - q_{nm}^i) \]  

(82)

In (75) it has been shown that \( 0 < v_{nm}^i \leq 1 \). From (60) and the assumption in (82), since \( v_{nm}^i \) is a linear transformation of \( z_{nm}^i \), it follows that the normalized quantization residuals are also uniformly distributed

\[ v_{nm}^i \sim U(0, 1) \]  

(83)

with expectation \( E(v_{nm}^i) = \frac{1}{2} \). Finally, from (59) and (83) the expectation of the detection ratio \( D_F \) can be readily computed

\[ E(D_F) = \frac{\sum E(v_{nm}^i) v_{nm}^i}{\sum E(v_{nm}^i)} = \frac{\sum E(v_{nm}^i)}{\sum E(v_{nm}^i)} = \frac{1}{2} \]  

(84)

The Central Limit Theorem justifies that the sum of random variables forming the detection ratio \( D_F \) converge to a Gaussian distribution (see Fig.8). As previously shown in (84), the mean value of this Gaussian is \( E(D_F) = \frac{1}{2} \). Thus, for any threshold \( T_F \geq 0.5 \) the performance of the detector, in terms of false alarm probability, is better than random (i.e. \( P_{fa}(T_F) \leq \frac{1}{2} \) \( \forall T_F \geq 0.5 \)).

**REFERENCES**


