Wavelet-Based Blind Watermarking of 3D Models

F. Uccheddu
Department of Electronic and Telecommunications
University of Florence
Via S. Marta 3
50139, Firenze, Italy
uccheddu@lci.det.unifi.it

M. Corsini
Department of Electronic and Telecommunications
University of Florence
Via S. Marta 3
50139, Firenze, Italy
corsini@lci.det.unifi.it

M. Barni
Department of Information Engineering
University of Siena
Via Roma 56
53100, Siena, Italy
barni@dii.unisi.it

ABSTRACT

Watermarking of 3D meshes has received a limited attention due to the difficulties encountered in extending the algorithms developed for 1D (audio) and 2D (images and video) signals to topological complex objects such as meshes. Other difficulties arise from the wide variety of attacks and manipulations 3D watermarks should be robust to. For this reason, most of the 3D watermarking algorithms proposed so far adopt a non-blind detection. In this paper we present a new blind watermarking algorithm for 3D meshes. In order to simultaneously achieve watermark imperceptibility and robustness a multiresolution framework is adopted. To do so we assume that host meshes are semi-regular ones, a property that permits to first perform a wavelet decomposition and then to embed the watermark at a suitable resolution level. Watermark detection is accomplished by computing the correlation between the watermark signal and the to-be-inspected mesh. Robustness against geometric transformations such as rotation, translation and uniform scaling is achieved by embedding the watermark in a normalized version of the host mesh, obtained by means of Principal Component Analysis. Experimental results show the validity of the proposed algorithm both in terms of imperceptibility and robustness against a wide class of attacks including noise addition, smoothing and cropping.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; K.5.1 [**Legal Aspects of Computing**]: Software Protection—*Copyrights*

General Terms

Algorithms

Keywords

3D watermarking, mesh watermarking, blind detection, copyright protection, 3D wavelets

MM&Sec'04 September 20-21, 2004, Magdeburgo, Germany
Copyright 2004 ACM 1-58113-854-7/04/0009 ... $5.00.

1. INTRODUCTION

In the last decade, watermarking has been one of the most active research topics, attracting the interest of researchers with different backgrounds, such as signal processing, communication and information theory, cryptograhy, and computational vision [4, 2]. However a great deal of this research effort has focused on digital watermarking of audio, images and video data. The result is that watermarking technology for this kind of media has now reached a good maturity. On the contrary, watermarking of 3D objects is far from this level of maturity even if 3D models are diffused in several applications such as virtual prototyping, Cultural Heritage, and entertainment industry (movies and video-games). One of the reasons for this gap is that it is difficult to extend common processing algorithms used in signal processing to 3D data. This is the case, for example of basic tools such as filtering and frequency analysis.

Even if 3D objects can be represented in several different ways (e.g. NURBS, voxels, implicit surface, polygonal meshes) most of the existing 3D watermarking algorithms work on polygonal meshes since this representation is the lowest common denominator of the other ones (i.e. it is easy to convert the other representations to meshes). For example the watermark may be inserted by altering mesh attributes such as vertex coordinates or vertex connectivity. Here we follow the same approach, i.e. we embed the watermark in the mesh describing the shape of the 3D object.

A strategy which is successfully adopted by many watermarking algorithms designed to deal with still images and video sequences, consists in first describing the host document by means of a multiresolution framework and then inserting the watermark at a resolution level presenting a satisfactory trade-off between perceptibility of the watermark and robustness against attacks.

The extension of this multiresolution approach to the 3D case, however, is not straightforward. The main reason for this difficulty, as well as for the difficulties encountered when trying to extend 2D processing tools to the 3D mesh case, is that the essentially 2-manifold structure of 3D surfaces has to be taken in account. In particular it is not possible to define equi-spaced sampling patterns on general 2-manifolds thus making the extensions of Fourier and other multiresolution analysis and synthesis tools very difficult. So, in order to extend common signal processing algorithms to geometry data we need to use sampling patterns which are as regular as possible.

Subdivision surfaces [20] have recently attracted the attention of computer graphics researchers since by providing a semi-regular sampling of surfaces, they are likely to become a fundamental block of many multiresolution algorithms for mesh processing [19]. This is the case, for example, of 3D wavelet decomposition [14]. In the following, a polygonal mesh obtaining by regularly subdividing an irregular coarse one will be referred to as a *semi-regular* mesh. In this paper, we present a novel multiresolution mesh watermarking algorithm particularly designed to work with semi-regular meshes with subdivision connectivity.

The proposed algorithm embeds the watermark by modifying the wavelet coefficients of 3D models obtained by decomposing the host mesh by means of the algorithm proposed by Lounsbery et al. [14]. Particular attention is paid to ensure that the embedding algorithm preserves the visual integrity of the models. The watermark is recovered by means of a correlation detector designed according to statistical detection theory [12].

Another distinguishing feature of the watermarking algorithm proposed in this paper, regards watermark detection. In fact, whereas most 3D watermarking systems proposed so far adopt non-blind detection, our system does not require that the original non-marked mesh is available at the detector, thus resulting in a much more flexible system easily adaptable to practical applications.

This paper is organized as follows. In section 2 a brief overview of 3D models representation is given, and the peculiarities of 3D watermarking reviewed. In section 3, the state of the art of 3D watermarking is briefly sketched. Section 4 describes the new watermarking algorithm, with regard to the embedding phase, whereas section 5 is devoted to watermark detection. Experimental results are presented in section 6. Finally some conclusions are drawn in section 7.

2. BACKGROUND

In this section we give some background material on 3D models representation and outline the main peculiarities of 3D watermarking.

2.1 3D models representation

Different representations are commonly used for 3D models; for example a 3D model can be described as a collection of parametric curves (e.g. Non-Uniform Rational B-Splines, NURBS) or as a set of implicit surfaces[1]. More usually, a 3D model is represented by polygonal meshes.

A mesh M can be seen as a $t-$uple (K, V) where $V = \{v_i \in \mathbb{R}^3 | i = 1 \dots N_v\}$ is the set of the vertices of the model (points in $\mathbb{R}^3$) and K is a set encoding adjacency information for vertices, edges and faces of the mesh. In particular K is formed by subsets of $I = \{1, \dots, N_v\}$ called *simplices.* We have three types of simplices: vertices $\mathcal{V} = \{\{i\} | i \in I\}$, edges $\mathcal{E} = \{\{i, j\} | i, j \in I, \{i, j\} \text{ is an edge}\}$ and faces $\mathcal{F} = \{\{i, j, k\} | i, j, k \in I, \{i, j, k\} \text{ is a face}\}$. The set K is called *simplicial complex* and is defined as $K = \mathcal{V} \bigcup \mathcal{E} \bigcup \mathcal{F}$. A vertex v_i is a neighbor of another vertex v_j if an edge exists that connects v_i and v_j. The set of all the neighbors of a vertex v_i is called *1-ring* of the vertex and is defined as $v_1(i) = \{j | \{i, j\} \in \mathcal{E}\}$. The cardinality of $v_1(i)$ is called *degree* or *valence* of the vertex v_i. The *geometric realization* of a simplex $s \in K$, denoted with $\varphi(s)$, is the strictly convex hull of the vertices v_i with $i \in s$. For example the geometric realization of an edge $\{i, j\} \in \mathcal{E}$ is the segment connecting the vertex v_i with the vertex v_j, the realization of a face $\{i, j, k\} \in \mathcal{F}$ is the triangle defined by the vertices v_i, v_j and v_k, and so on. The 3D model is the geometric realization of the mesh $\varphi(K)$ defined as $\bigcup_{s \in K} \varphi(s)$. Usually the vertices are characterized not only by theirs coordinates but even by other attributes such as texture coordinates, color and so on. Here we are interested only in the geometry of the mesh so we do not take in account these attributes. In the following when we refer to a mesh we intend a *triangular mesh*, i.e. a mesh composed by triangles only. This assumption implies no loss of generality since every polygon of a non-triangular mesh can be triangulated to obtain a triangular mesh. A mesh is called *irregular* if its vertices can have any valence, *completely-regular* if all vertices have the same valence and *semi-regular* if most of its vertices have the same degree except a small number that can have any valence. This last definition arises because a semi-regular mesh is obtained by repeatedly and regularly subdividing [20] an irregular mesh. During the subdivision process the irregular vertices of the initial mesh remain irregular while most of the newly inserted vertices converge to valence six (for triangular semi-regular mesh). This classification is very important because for semi-regular meshes (and for completely regular ones) a lot of geometric processing tools exist. For example wavelet decomposition is defined only for semi-regular and completely-regular meshes[14]. It is important to underline that an irregular mesh can be always converted to a semi-regular one by an operation called *remeshing* [1, 21, 13]. As noted in the introduction, our algorithm is expressly designed to work with semi-regular meshes so to take advantage of the multiresolution framework provided by 3D wavelet analysis [14].

2.2 Main peculiarities of 3D watermarking

Any watermarking system must cope with three basic, yet conflicting, requirements: payload, robustness and imperceptibility. In 3D watermarking, each of the above aspects assume a particular characterization.

2.2.1 Watermark payload

The payload of a watermarking systems is the amount of information bits that the watermark is able to convey. In general, watermark payload depends on the particular watermarking algorithm and it is related to the characteristic of the host data. This is also true for 3D models: the achievable payload is tightly related to the complexity of the mesh. Because it is not simple to define mesh complexity from an information theoretic viewpoint, we will refer to *mesh complexity* as the number of faces and vertices it contains, assuming that mesh vertices do not over-sample the shape they represent.

2.2.2 Imperceptibility

Watermark *imperceptibility* is a crucial point for 3D watermarking since in many cases the user is allowed to manipulate the 3D mesh in a variety of ways. For instance, it is easily guessed that watermark visibility will depend on the

[1] The implicit method uses a function depending on axis variables, usually equal to 0, to describe a shape. For example the equation $x^2 + y^2 + z^2 = 1$ represents the sphere of radius 1.

particular rendering algorithm used to visualize the mesh, e.g. it will depend on the shading algorithm, the reflectance properties of the object and so on. So far, only a few studies on 3D watermark perceptibility have been done, all the more that a 3D model can be viewed in an interactive way thus making the perceptual analysis of 3D models more complex than for image or video data. Our approach to guarantee the visual quality of the watermarked model is to perform an interactive visual comparison between the watermarked and the original mesh to establish the maximum watermark power that can be injected within the host model without introducing significant distortions in the geometry of the model. Fixed rendering conditions have been also assumed. The possibility of using an automatic process indicating the maximum watermark power still ensuring invisibility is a matter of an on-going research.

2.2.3 Robustness

A final important requirement for any watermarking algorithms, especially for those used in Intellectual Property Right (IPR) protection applications, is *robustness* against manipulations. One of the main peculiarities of 3D watermarking is that a lot of sophisticated and difficult-to-prevent attacks are possible on polygonal meshes. Just to mention some, the watermark should be robust against noise addition, translation, rotation, scaling, simplification, retriangulation, smoothing, cutting, remeshing and many others. A detailed description of some of these attacks follows:

- **Translation/Rotation/Uniform Scaling.** These geometric transformations are very used in computer graphics to position a 3D model inside a scene. Other geometric transformations less used than these, are affine and projective ones may be used, even if they are less common than .

- **Noise.** By noise attack we intend the addition of random vectors to mesh vertices. The modulus of these vectors have to be small compared with the mesh dimension to preserve the overall shape of the model.

- **Re-triangulation.** This attack changes the connections between mesh vertices leaving their position unaltered. (see figure 1).

- **Mesh smoothing.** A smoothing of the surface represented by a polygonal mesh can be obtained by mesh filtering such as Taubin filtering [22]. This kind of filter acts on the mesh as a low-pass filter attenuating the roughness of the surface.

- **Polygonal simplification.** Polygonal simplification is often used to transmit a low-level version of the model or to optimize a model eliminating most of the non-salient faces.

- **Cropping.** Cropping concerns the disjunction of a part of the model. Users can discard the pieces of the model that they do not need (e.g. the hand of a statue).

- **Remeshing** Remeshing is used to regularize a mesh converting an irregular mesh into a semi-regular [13, 1, 21] or a completely-regular [6] one. This operation can be seen as a geometric resampling of the shape of the model followed by a re-definition of the vertices connections (a re-triangulation).

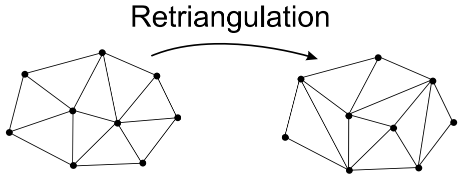

Figure 1: Re-triangulation attack.

2.2.4 Embedding domain

The first step towards the definition of a watermarking algorithms, consists in the choice of the host features, i.e. the selection of a set of properties of the host model that will bear the watermark information. For a 3D objects many possibilities exist, vertices positions, texture coordinates, shape-related features and so on. Here, we are interested in the geometric properties of the mesh, so we focus only on geometric and topological features.

- **Geometric Features.** The main geometric features of a mesh are its vertices. One possible way to embed the watermark is to modify the position of the vertices. Another way to embed the watermark is to modify the normals of vertices (vertex normals are related to the curvature of the mesh, hence to the shape of the 3D objects). Both these entities are altered by perturbing the coordinates of mesh vertices.

- **Topological Features.** These features are related to the connectivity of the mesh vertices. Usually, a set of connected vertices is selected by using some geometric features. Then, the topology of these vertices is redefined to encode one or more bits. Usually topological features are used when the robustness constraint of the watermarking algorithm is relaxed, in fact these features are particularly vulnerable to the re-triangulation attack, that it is straightforward to implement.

Our algorithm embeds the watermark by modifying the position of the mesh vertices in a transformed domain, i.e. wavelet domain.

2.2.5 Watermark retrieval

The way the system extracts the embedded information from the host mesh has a strong impact on practical applications. For example, it is known that non-blind techniques are less useful in practical applications than blind ones. Another important distinction is between *readable* and *detectable* watermarks. In the first case, the algorithm embeds a code that can be re-extract (read) without knowing it in advance. In the second case the algorithm is only capable of *verifying* whether a given code is contained in the data or not. The latter kind of techniques are sometimes referred to as *1-bit watermarking* because the output of the detector is just yes or not. The algorithm presented in this paper belongs to the category of detectable watermarks.

3. PREVIOUS WORKS

Watermarking of 3D models is a relatively new research topic. One of the earliest works on 3D watermarking was presented by Ohbuchi et al. in [16, 17]. They presented several techniques embedding the watermark in the geometrical domain: the watermark is embedded into the polygon data by modifying either the vertex coordinates, the vertex connectivity , or both. They also discussed the ordering schemes into a set of geometrical or topological elements for embedding and extracting the watermark. The techniques they presented are non blind and robust to 3D affine transformations and cropping.

Kanai et al. [10] propose to decompose the host mesh into a multiresolution representation by applying the lazy wavelet transform proposed by Lounsbery et al. [14]. Then they modify the wavelet coefficients to embed the watermark and detect it in a non-blind way. Our method is based on this work and extend it providing a novel blind technique for 3D watermarking.

Praun and Hoppe [18], also presented a detail-preserving non-blind method driven by multiresolution theory. Due to the construction of filters applied to the mesh, the watermark affects the overall sweep of surfaces, so that the algorithm is resistant against many local transformation such as noise addition and smoothing. Moreover robustness against complex attacks such as remeshing are achieved thanks to a regisration - resampling phase performed before the extraction phase.

Benedens [3] developed two methods, namely Normal Bin Encoding and Affine Invariant Embedding, both of which are based on the alteration of surface normals. He maps surface normals onto the unit sphere, and then subtly alters groups of similar normals in order to embed the individual bits of the watermark sequence. However he demonstrates robustness only with simplification attacks. These methods are not completely blind (semi-blind), in that some information directly derived from the original mesh must be provided to the detector.

Thomas Harte and Adrian G. Bors[8] presented a blind technique to embed information in a 3D objects by altering vertices position in geometric domain. They demonstrated robustness only against scaling and rotation and combination of geometrical transformation.

In [15], Ohbuchi et al. proposed a frequency domain approach to the watermarking of 3D shapes. The mesh is first segmented into patches, and then the watermark is embedded by modulating the amplitude of the mesh spectral coefficients with a spread-spectrum approach. Spectral coefficients are computed by means of a spectral analysis based on the Kirchhoff matrix of each patch. Nevertheless, the patch generation step cannot be performed automatically, and manual interaction is needed.

Yin et al. reported an informed (non blind) detection, robust mesh watermarking algorithm that works in a transformed domain [23]. It is based on a multiresolution decomposition of polygonal mesh shapes developed by Guskov [7] that separates a mesh into detail and coarse feature sequences essentially by repeatedly applying local smoothing combined with shape difference. This non-blind watermarking algorithm has shown good robustness properties, somewhat similar to the method proposed by Praun et al. [18].

As some of the methods discussed above, our scheme embeds the watermark in a transformed domain, namely the wavelet domain. As opposed to these methods, though, we do not need the original mesh to recover the watermark, since blind detection is accomplished.

4. WATERMARK EMBEDDING

The 3D watermarking system presented in this paper embeds a numeric code into a semi-regular mesh with subdivision connectivity using a multiresolution framework. An advantage of the multiresolution analysis is that it permits to extend the geometrical deformations introduce by the watermark from low to high resolution in a smooth way, so that deformations result distributed around the perturbed vertices avoiding the ”noise” effect usually caused by most watermarking algorithm that works at full resolution. In this way, the visual quality of the final watermarked model is greatly improved. Moreover robustness improvements are expected from this approach since the watermark is tied to the very basic shape of the mesh, and hence it is very difficult to remove the watermark without that, at the same time, the mesh quality is severely degraded. This also explains why we chose to work with semi-regular meshes, since this is the basic assumption multiresolution analysis relies on.

Another important feature of the new algorithm is detector blindness. To obtain this result, and yet preserve robustness against geometrical transformations, it is necessary that the watermark embedding and detection phases work on a normalized model, i.e. on a model re-oriented and scaled as the original one. This normalization allows to ignore translation, rotation and uniform scaling modifications. It goes without saying that model normalization at the detector must be accomplished without the use of the original model, not to compromise the blindness of the overall watermarking system. For this reason, a normalization phase expressly designed to cope with geometric manipulations of the host model is inserted before the embedding and detection stages (see section 4.2).

In the following, after a brief excursus on subdivision surface and multiresolution analysis, we describe in detail the normalization phase and the watermark embedding algorithm. Watermark recovery will be described in section 5.

4.1 3D multiresolution framework

Wavelet theory, and multiresolution analysis in general, is a powerful tool for representing signals at different levels of detail. In our system we implemented a class of 3D wavelets based on subdivision surface [14]. According to this theory, it is assumed that the mesh to be decomposed by wavelet analysis is a semi-regular one, i.e. a triangular mesh obtained by regularly subdividing an irregular one [20]. Note that it is always possible to transform an irregular mesh into a semi-regular one by a remeshing operation [1, 21, 13]. Usually remeshing operation is performed by extracting a base mesh, parameterizing the model, and choosing a rule of subdivision [20].

Figure 2 illustrates the decomposition and multiresolution representation of a polygonal model by using the wavelet transform. The high-resolution polygons are decomposed into a low-resolution part and a detail part. The detail parts are expressed as the wavelet coefficient vectors, while the low resolution part is the coarse model.

In the following we indicate by M the semi-regular mesh. The mesh M can be seen as a tuple (V, H) where V is the

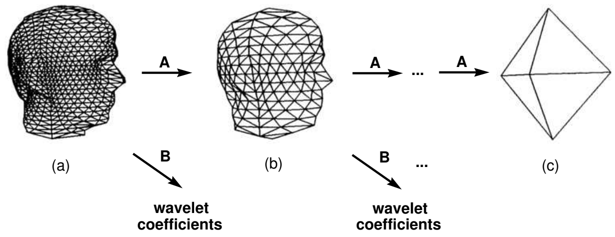

Figure 2: Decomposition of a polyhedral surface.

set of vertices and H encodes adjacency information (to represent edges and faces). M^j indicates the mesh at level of resolution j, in particular for a mesh with n levels of resolution, M^0 is the base mesh and $M^{n-1} = M$ is the mesh at full resolution. With this notation we can formulate the wavelet transformation by:

$$\begin{aligned} V^{j-1} &= A^{j-1}V^j \\ W^{j-1} &= B^{j-1}V^j \end{aligned} \tag{1}$$

where $V^j = [v_1^j v_2^j ... v_{m^j}^j]^T$ is a matrix whose rows correspond to vertex coordinates of the model M^j, V^{j-1} corresponds to the next lower resolution level $j-1$ and m^j is the number of vertices at resolution j. $W^{j-1} = [w_1^{j-1} w_2^{j-1} ... w_{n^{j-1}}^{j-1}]^T$ denotes a matrix whose rows correspond to the wavelet coefficient vectors at resolution level $j-1$ where $n^{j-1} = m^j - m^{j-1}$. The matrices A^j and B^j are called *analysis filters*. The process defined in equation (1) is recursively applied to the high resolution part until the coarsest representation of the mesh V^0 is obtained. The whole process can be formulated as:

$$\begin{aligned} V^0 &= A^0A^1...A^{l-2}A^{l-1}V^l \\ W^0 &= B^0B^1...B^{l-2}B^{l-1}V^l \end{aligned} \tag{2}$$

On the contrary, the inverse wavelet transform is expressed as:

$$V^j = P^{j-1}V^{j-1} + Q^{j-1}W^{j-1} \tag{3}$$

where P and Q are called *synthesis filters*. The relationship between these filters and the analysis ones is:

$$\left[\begin{array}{cc} P^{j-1} & Q^{j-1} \end{array} \right] = \left[\begin{array}{c} A^{j-1} \\ B^{j-1} \end{array} \right]^{-1} \tag{4}$$

The synthesis can be viewed more concretely as consisting of two steps: each triangle of the low resolution mesh (V^{j-1}) is split into four sub-triangles by introducing the new vertices at the midpoints of edges by using P^{j-1} (polyhedral subdivision), and then the position of these new vertices is modified according to the wavelet coefficients by using Q^{j-1}. Figure 3 shows this process. It is important to remark that in this implementation the wavelet coefficients are vectors.

4.2 Mesh normalization

Mesh normalization works in two stages. In the first one, model orientation is normalized by means of Principal Component Analysis (PCA). Then the model is fitted to a bounding box consisting of a cube of dimensions 1.0 × 1.0 × 1.0 centered in the barycenter of the model. Fitting is obtained by applying a translation and a uniform scaling. While translation and scaling are trivial operations, the orientation phase needs some comments.

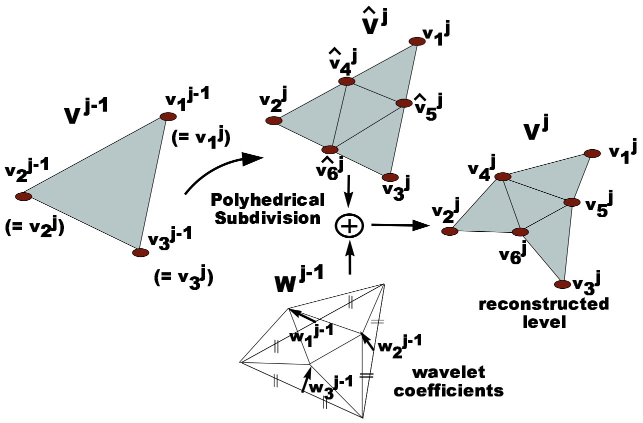

Figure 3: Inverse wavelet transform. In the first step each triangle of the low resolution mesh is split into four sub-triangles by introducing new vertices at the midpoints and then the position of these new vertices is modified according to the wavelet coefficients vectors.

As we already said, the basic idea is to re-orient the model using principal component analysis. As opposed to previous works (see for example [9]), we compute the covariance matrix and the center of mass of the model by considering the centers of the faces of the mesh, instead of its vertices. Each face center is associate to a mass m_i which is equal to the area of the face (see [5]). This strategy is more reliable than the one based on vertices positions because re-orienting the model by means of its surface area distributions provides more robustness against vertices positions changes. For example, if we imagine to add randomly vertices around a specific location on the mesh, in the first approach the center of the mass changes, while in our approach mass center remains the same. Specifically, we calculate the center of mass O of the model as:

$$O = \frac{1}{n_f} \sum_{i=1}^{n_f} p_i m_i, \tag{5}$$

where p_i is the center of the i-th face of the model and the mass m_i associated to p_i is the area of the face p_i belongs to. Then, the PCA is applied to the following covariance matrix:

$$I = \left[\begin{array}{ccc} \sum_{i=1}^n x_i^2 m_i & \sum_{i=1}^n y_i x_i m_i & \sum_{i=1}^n z_i x_i m_i \\ \sum_{i=1}^n x_i y_i m_i & \sum_{i=1}^n y_i^2 m_i & \sum_{i=1}^n z_i y_i m_i \\ \sum_{i=1}^n x_i z_i m_i & \sum_{i=1}^n y_i z_i m_i & \sum_{i=1}^n z_i^2 m_i \end{array} \right] \tag{6}$$

where (x_i, y_i, z_i) are the coordinates of the points p_i's relative to the center of mass O. Finally, the eigenvectors of I are computed and used to align the principal axis of I to the coordinate axis. This is done by simply multiplying the position of each vertex by the PCA matrix.

4.3 Embedding rule

The watermarking embedding algorithm works according to three parameters: i) a secret key K; ii) the resolution level l that will host the watermark, and iii) a coefficient γ determining the strength of the watermark.

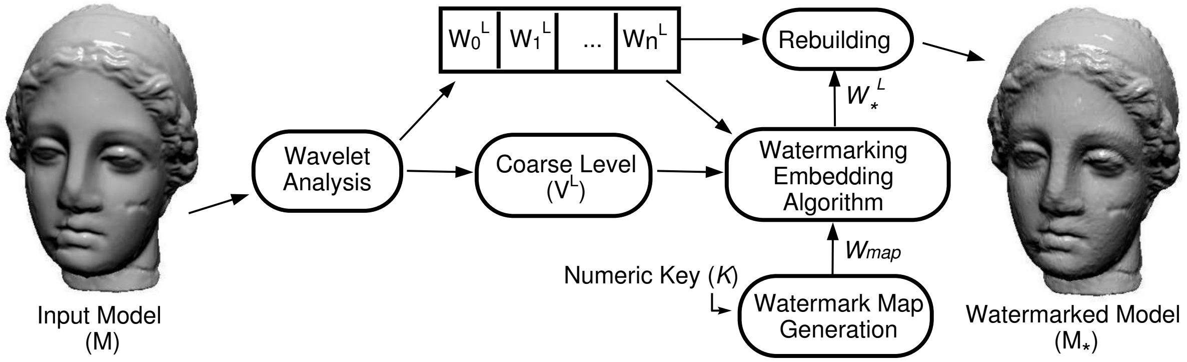

Figure 4: Scheme of the watermark embedding algorithm.

An overall picture of the watermark insertion process is shown in figure 4. In particular the following steps are performed:

1. A particular watermark-dependent structure called *watermarking map* is generated according to K.
2. The input model M is decomposed into a set of wavelet coefficients $W^0, W^1, \ldots, W^{n-1}$ and a base domain M^0.
3. The vertices of the model at level l (V^l) and the wavelet coefficients at the same level (W^l) are used by the embedding algorithm to embed the watermark by altering the modulus of a subset of the wavelets coefficients of W^l.
4. The watermarked wavelet coefficients W^l_* and the other sets of wavelets $W^{l+1}, \ldots, W^{n-1}$ are used to reconstruct the watermarked full resolution model.

In the next sections we give a detailed description of each of the above step.

4.3.1 Watermarking map generation

Given the key K, a matrix called *watermarking map* (W_{MAP}) is built. This map is used in the embedding phase to insert K within the host model. To build W_{MAP}, the key K is used as a seed to generate a pseudo-random sequence of displacement values $D = \{d_1, d_2, \ldots, d_t\}$ uniformly distributed in the $[-1, 1]$ interval. Then, this sequence is arranged into a matrix with n rows and m columns ($t = n \times m$). The basic meaning of W_{MAP}, is that of a mapping of the watermarking signal into polar coordinates, where the row index of W_{MAP} samples the azimuth angle ($\theta \in [0, 2\pi]$) and the column index samples the elevation angle ($\phi \in [0, \pi]$). The number of rows and columns of W_{MAP} determines the accuracy with which the polar representation of the unitary sphere is sampled. Here we adopted a W_{MAP} of 360×180 elements, i.e. 1 degree of resolution.

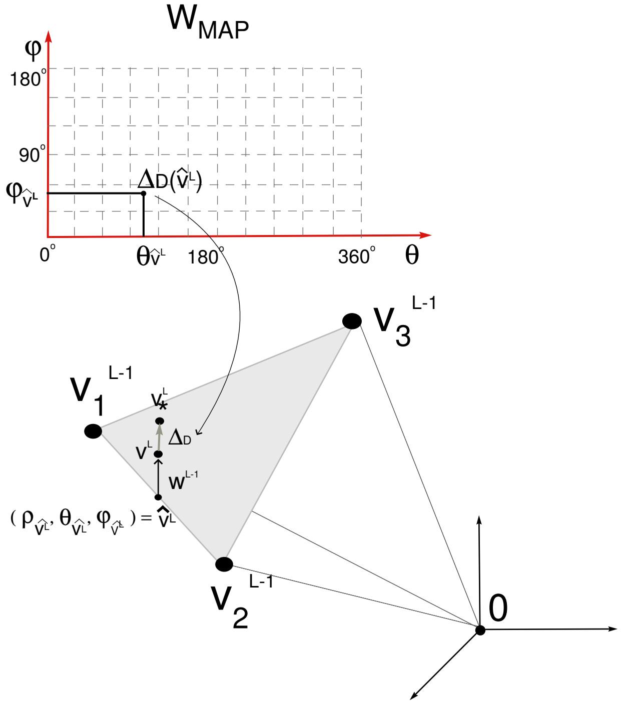

Figure 5: Watermark embedding. The modulus of the wavelet coefficient $|w^{l-1}|$ is perturbed according to the value of the watermarking map (W_{MAP}) indexed by the polar coordinates $(\theta_{\hat{v}}, \phi_{\hat{v}})$ of the application point $\hat{v}^l$ of the wavelet coefficient.

4.3.2 Watermark casting

As we already said, the set of wavelet coefficients W^l and the vertices V^l are used to embed the watermark. In particular the modulus of the each wavelet coefficients w_i at level l (W^l) is altered as follows:

$$|w_{i,*}| = |w_i| + \gamma\ \Delta_D(\hat{v}(w_i), W_{MAP})\ ,\ \forall w_i \in \overline{W}^l, \qquad (7)$$

where the exact meaning of $\overline{W}^l$ will be explained below. The exact amount of perturbation the modulus of the wavelet coefficient undergoes, namely $\gamma\ \Delta_D(\hat{v}(w_i), W_{MAP})$, is determined by the parameter γ, and by $\Delta_D(\hat{v}(w_i), W_{MAP})$, a quantity which depends on the application point $\hat{v}(w_i)$ of the wavelet coefficient w_i and by the watermarking map W_{MAP}.

Specifically, the polar coordinates $(\theta_{\hat{v}}, \phi_{\hat{v}})$ of $\hat{v}(w_i)$ are used to index the watermarking map and to obtain a value of perturbation specific for that position and that watermark. In the following we assume that $(\theta_{\hat{v}}, \phi_{\hat{v}})$ are expressed in degrees and not in radians. Of course no guarantee exists that the polar coordinates of $\hat{v}(w_i)$ will exactly correspond to

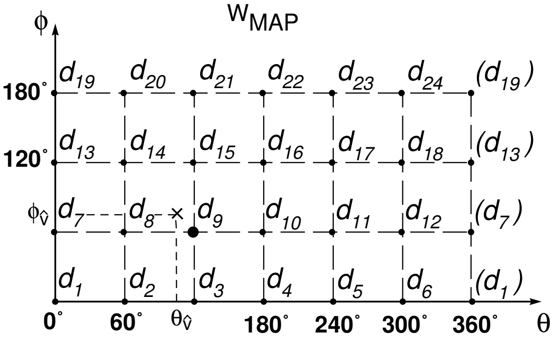
ϕ
W_MAP
180°
120°
ϕ_v̂
d_19 d_20 d_21 d_22 d_23 d_24 (d_19)
d_13 d_14 d_15 d_16 d_17 d_18 (d_13)
d_7 d_8 d_9 d_10 d_11 d_12 (d_7)
d_1 d_2 d_3 d_4 d_5 d_6 (d_1)
0°
60°
θ_v̂
180°
240°
300°
360°
θ

Figure 6: Example of computation of $\Delta_D(\hat{v}(w_i), W_{MAP})$. As an example, a 24-sample long sequence $D = \{d_1, \ldots, d_{24}\}$ is arranged into a watermarking map of 4×6 elements. The point $\hat{v}(w_i)$ of polar coordinates ($\theta_{\hat{v}} = 110.2°, \phi_{\hat{v}} = 73.5°$) is mapped to $\Delta_D(\hat{v}(w_i), W_{MAP}) = d_9$.

a position expressed by an integer index of W_{MAP}, hence the following simple interpolation rule is adopted to calculate the exact value of $\Delta_D(\hat{v}(w_i), W_{MAP})$:

$$\Delta_D(\hat{v}(w_i), W_{MAP}) = W_{MAP}(\theta_i, \phi_i) \tag{8}$$

$$\begin{aligned} \theta_i &= \mathsf{round}(\theta_{\hat{v}}(m/360.0)) \\ \phi_i &= \mathsf{round}(\phi_{\hat{v}}((n-1)/180.0)) \end{aligned} \tag{9}$$

where $\mathsf{round}(x)$ is the integer nearest to the value of x. Note that if two or more points $\hat{v}(w_i)$'s are mapped to the same index of W_{MAP}, then some correlation is introduced within the watermarking sequence $\Delta_D(\hat{v}(w_i), W_{MAP})$, hence it is desirable that this happens as rarely as possible. In our system the resolution of 1 degree has been found to be a good trade-off between independence of w_i's and robustness. This question will be better clarify in the experimental results section. Figure 6 shows graphically how $\Delta_D(\hat{v}(w_i), W_{MAP})$ is calculated.

In order to minimize the geometrical distortion introduced by the use of spherical coordinates, only those wavelets for which $\phi_{\hat{v}}$ is far from the poles are marked. Thus, only wavelets with $\Delta\phi \leq \phi_{\hat{v}} \leq 180.0 - \Delta\phi$ are marked. In our implementation $\Delta\phi$ is set to twenty degrees. We indicate this subset with $\overline{W}^l$.

A particular attention must be paid to avoid that the watermarking process produces negative modules. To do so we replace negative modules with zero, that is:

$$|w_{i,*}| = \min(0, |w_i| + \gamma\Delta_D(\hat{v}(w_i), W_{MAP})) \ , \ \forall w_i \in \overline{W}^l \tag{10}$$

5. WATERMARK DETECTION

The detection procedure works as follows: the user specifies the numeric key K and the level of resolution l where he wants to verify the presence of K and the detector provides a positive or negative answer.

To do so the watermarking map W_{MAP} is generated by starting from K, V_l and W_l are obtained by wavelet analysis, and then the correlation between the watermarking signal and the wavelet coefficients is calculated as follows:

$$\rho = \frac{1}{n} \sum_{w_i \in \overline{W}^l} |w_i| \Delta_D(\hat{v}(w_i), W_{MAP}) \tag{11}$$

where n is the cardinality of $\overline{W}^l$.

If ρ is greater than a certain threshold T_ρ the watermark is present, else the model is declared non-marked. The value of T_ρ is obtained by means of statistical considerations. In the next section we provide the theoretical analysis used to determine this threshold and a theoretical evaluation of the performance of the proposed algorithm. In the experimental results section we validate this theoretical performance.

5.1 Choice of T_ρ

The most popular approach to the choice of T_ρ consists in resorting to statistical detection theory. Specifically the Neyman-Pearson criterion is usually adopted, which consists in maximizing the missed detection probability for a given probability of falsely revealing the watermark in a non-marked host mode [2]. We use the same approach here. To go on, we need to compute the probability of falsely detecting the watermark presence, i.e.:

$$P_f = P\{\rho > T_\rho | H_0\}. \tag{12}$$

where H_0 indicates the hypothesis that $|w_i|$ are not marked. Similarly we can define the probability of missing the watermark as:

$$P_m = P\{\rho < T_\rho | H_1\}. \tag{13}$$

where H_1 indicates the hypothesis that $|w_i|$ are marked with the watermark K whose presence is under verification.

In both cases, the error probabilities are obtained by fixing the host model and averaging over different watermarks K. In the sequel we will assume that watermark samples are zero mean i.i.d. random variables. As we anticipated when describing the role of the watermarking map, this is true only if the application points of wavelet coefficients are mapped into different elements of W_{MAP}. In turn, this is true only if the step used to sample the polar coordinates of $\hat{v}(w_i)$ is small enough. We found experimentally that by letting such a sample step be equal to 1 degree a reasonable trade-off is reached between independence of watermark samples and robustness (see section 6).

In order to go on, let us indicate by w_i' the i-th coefficient under inspection. We use a different symbol w_i' to explicitly indicate that the detector does not know whether w_i' is marked or not. We start by noting that according to our model $\Delta_D(w_i)$'s are independent random variables and $|w_i'|$'s are fixed parameters which are known to the detector[2], hence, by invoking the central limit theorem, we can conclude that ρ follows a normal distribution. To completely characterize ρ, then, it is sufficient to estimate its mean and variance. Let us assume that H_0 holds. In this case we have $|w_i'| = |w_i|$, hence:

$$\begin{aligned} \mu_{\rho|H_0} = E[\rho|H_0] = \frac{1}{n} E\left[\sum_{w_i \in \overline{W}^l} |w_i| \Delta_D(\hat{v}(w_i))\right] = \\ = \mu_{\Delta_D} \overline{w} = 0, \end{aligned} \tag{14}$$

[2] From now on we avoid to explicitly indicate the dependence of Δ_D upon W_{MAP}.

where $\overline{w} = \sum |w_i|/n$, n is the number of elements in $\overline{W}^l$ and μ_{Δ_D} is the mean of the sequence $\Delta_D(\hat{v}(w_i))$. As to the variance of ρ under hypothesis H_0, we can write:

$$\sigma^2_{\rho|H_0} = \text{var}\left(\frac{1}{n}\sum_{w_i \in \overline{W}^l} |w_i|\Delta_D(\hat{v}(w_i))\right) = \frac{1}{n}\sigma^2_{\Delta_D}\overline{w^2}, \quad (15)$$

with

$$\overline{w^2} = \frac{1}{n}\sum_{w_i \in \overline{W}^l} |w_i|^2, \quad (16)$$

denoting the sample mean square value of $|w_i| \in \overline{W}^l$, and $\sigma^2_{\Delta_D}$ denoting the variance of the sequence $\Delta_D(\hat{v}(w_i))$. We can now calculate the false detection probability, obtaining:

$$P_f = \int_{T_\rho}^{\infty} p(\rho|H_0)d\rho = \frac{1}{2}\ \text{erfc}\left(\sqrt{\frac{(T_\rho - \mu_{\rho|H_0})^2}{2\sigma^2_{\rho|H_0}}}\right). \quad (17)$$

This expression of P_f can be inverted to calculate the detection threshold, yielding:

$$T_\rho = \sqrt{2}\sigma_{\rho|H_0}\text{erfc}^{-1}(2P_f) + \mu_{\rho|H_0}. \quad (18)$$

By inserting equations (14) and (15) in the above expression, we finally have:

$$T_\rho = \sqrt{\frac{2\sigma^2_{\Delta_D}\overline{w^2}}{n}}\text{erfc}^{-1}(2P_f). \quad (19)$$

In a similar way we can evaluate the probability of missing the watermark presence [2], obtaining:

$$P_m = \frac{1}{2}\ \text{erfc}\left(\sqrt{\frac{(\mu_{\rho|H_1} - T_\rho)^2}{2\sigma^2_\rho}}\right). \quad (20)$$

By using the result in (19), it is easy to obtain the following expression for P_m, in which the missed detection probability is expressed as a function of P_f:

$$P_m = \frac{1}{2}\ \text{erfc}\left(\sqrt{\frac{(\gamma\sigma^2_{\Delta_D} - \sqrt{\frac{2}{n}\sigma^2_{\Delta_D}\overline{w^2}}\text{erfc}^{-1}(2P_f))^2}{\frac{2}{n}\sigma^2_{\Delta_D}\overline{w^2} + \gamma^2[E[{\Delta_D}^4] - E[{\Delta_D}^2]^2]}}\right). \quad (21)$$

Equation (21) completely characterizes the detector performance. Such performance are usually summarized through ROC curves where the missed detection probability is plotted against P_f.

6. EXPERIMENTAL RESULTS

In this section we report a selection of the results that we obtained while testing the validity of our watermarking algorithm. Specifically three aspects where considered: i) watermark invisibility, ii) actual false detection probability, and iii) robustness.

The models used in the experiments are the semi-regular meshes made available to the public by the Multiresolution Modeling Group of Caltech University. These models are the "bunny", the "feline", the "Venus" head, the "rabbit" and the "horse" model. Here we present some of the results obtained, in particular, for the bunny and the Venus models but similar results, from a qualitative viewpoint, are obtained even for the other models.

bunny		venus	
l	$\gamma_{\max}$	l	$\gamma_{\max}$
3	0.0005	3	0.0003
2	0.0008	2	0.0005
1	0.0030	1	0.0009

Table 1: Maximum value of γ before the watermark becomes perceivable ($\gamma_{\max}$) for different watermarking levels (l).

Model	l	$P^*_f = 10^{-2}$	$P^*_f = 10^{-3}$
bunny	3	3.43×10^{-2}	8.4×10^{-3}
bunny	2	1.35×10^{-2}	1.8×10^{-3}
bunny	1	1.11×10^{-2}	1.1×10^{-3}
venus	3	3.14×10^{-2}	6.4×10^{-3}
venus	2	1.07×10^{-2}	1.2×10^{-3}
venus	1	0.97×10^{-2}	1.0×10^{-3}

Table 2: Actual false alarm rate for different watermarking levels (l). P^*_f is the target value of P_f.

6.1 Watermark Perceptibility

In order to achieve high visual quality of the watermarked model we have carefully considered the problem of watermark perceptibility after its insertion in the model. As previously noticed, it is very difficult to evaluate in a objective way the perceptibility of the geometric distortion introduced by the watermarking process. Hence we evaluated this distortion by visual inspection; an human user compares the original and the watermarked model using a software expressly designed for this purpose and find the maximum watermark strength (γ) that results in a non-visible watermark. We used this analysis to determine a value of γ that ensured invisibility for all the models used. All the experimental results we provide in this section guarantee a imperceptibility of the watermark, i.e. a high visual quality of the watermarked model. An example of the influence of γ on the visibility of the watermark is shown in figure 7, where the 3D model "Venus" has been watermarked with different values of γ.

In table 1 it is shown how the maximum perceptually tolerable value of γ changes while varying the model and the resolution level l. In particular $\gamma_{\max}$ increases at lower resolution levels.

6.2 False alarm rate

Equation 19 permits to determine a threshold T_ρ that guarantees the performance of the detector in terms of false detection probability. To validate this theoretical result, we tried to detect 10,000 different watermarks on the same nonmarked model for a fixed value of $P_f = P^*_f$; we expect that the actual false alarm rate equals P^*_f. The results we obtained are summarized in table 2.

As it is readily seen, the agreement between theory and practice is rather good. The slight differences that can be observed can be explained by considering how watermark coefficients are mapped onto the watermarking map W_{MAP}. In fact, as previously noticed, if the model has a great density of vertices it may happen that a certain number of different points $\hat{v}(w_i)$ are mapped on the same index of W_{MAP}. The effect is that $\Delta(\hat{v}(w_i), W_{MAP})$ values are no more independent, thus increasing the actual variance of ρ and causing

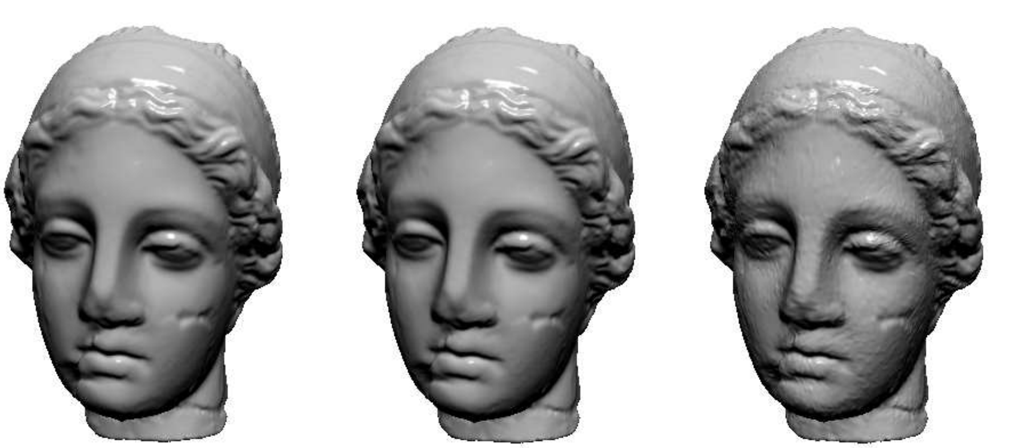

Figure 7: Watermark Perceptibility. (Left) Original Venus model. (Center) Venus model watermarked with $l = 3$, $\gamma = 0.0002$, the watermark is imperceptible. (Right) Venus model watermark with $l = 3$, $\gamma = 0.002$, the geometric manipulations introduced by the watermarking process become visible.

an actual value of P_f greater than expected. This problem can be alleviated either by embedding the watermark at a lower resolution level or by augmenting the resolution of W_{MAP}, for example by using a map with 720×360 elements (0.5 degrees of resolution).

6.3 Robustness

One of the main problems of 3D watermarking is the wide variety of different attacks possible on a polygonal mesh. This is one of the main reason why many 3D watermarking algorithms use the original model in the extraction phase.

Since our technique is specific for semi-regular meshes with subdivision connectivity, we do not take in account those attacks that alter this properties of the mesh such as re-triangulation, simplifications or re-meshing. Instead we tested the robustness of the algorithm against additive noise, low pass filtering, geometric manipulations, cropping and a combination of the above.

6.3.1 Additive noise

Additive noise is a standard attack to evaluate the performance of a watermarking system. In our case the noise is added to the watermarked model by perturbing its vertices at full resolution in a random way. More specifically, for each vertex a different displacement vector $\Delta_{noise} = (\Delta_x, \Delta_y, \Delta_z)$ is applied. The vector components Δ_x,Δ_y and Δ_z are random variables with uniform distribution in the interval $[-\Delta, \Delta]$. In figure 8 the value of ρ and T_ρ for increasing values of Δ is given. In particular the plot is given as a function of the quantity l_{med}/Δ, where l_{med} is the average length of the edges of the model. The models used in this test are the bunny and Venus both watermarked at level of resolution $l = 3$ with $\gamma = 0.0004$ and $P_f = 10^{-8}$. To give a visual idea of the maximum amount of noise the watermark can survive, in figure 9, a noisy version of the bunny model is shown, where noise strength is set to the maximum level for which the watermark can be recovered. As it can be seen, though the mesh is significantly deteriorated, the watermark is still detectable.

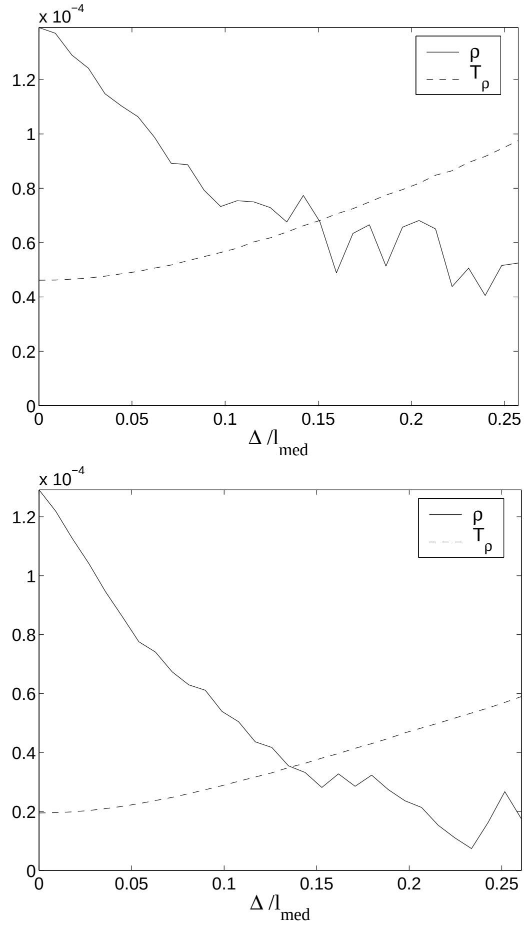

Figure 8: Robustness against Additive Noise attack. (Top) Bunny model watermarked with $l = 3$, $\gamma = 0.0004$ and $P_f = 10^{-8}$. (Bottom) Venus model watermarked with the same watermarking parameters.

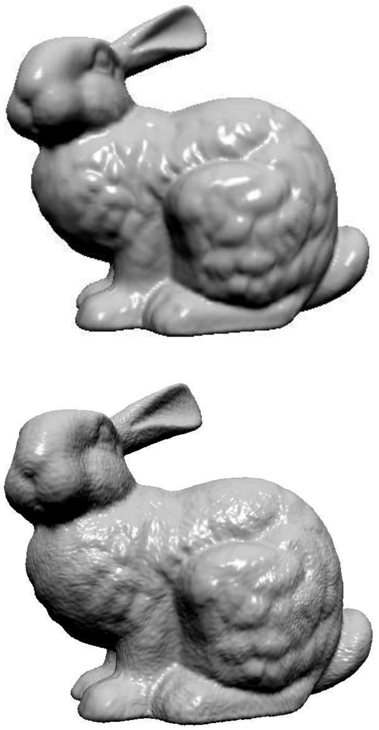

Figure 9: Robustness against Additive Noise attack. Original (up) and attacked (bottom) models. The model is attacked with the maximum amount of noise for which the watermark is detected.

6.3.2 *Low pass filtering*

Mesh filtering is used by a number of diverse applications, e.g. to eliminate the surface imperfections in an 3D object acquired by laser scanner or for editing purposes. Usually mesh filtering is a local operations, in other words it affects only the high-resolution part of the model. So, we expect that by embedding the watermark at a low-medium resolution level a good resistance against this kind of attacks is obtained. This is indeed the case. In particular we evaluated the performance of the watermarking system against the Taubin filter [22]. Several coefficients of the filter have been tried to obtain different smoothing effects and these battery of filters has been applied to the watermarked model a certain numbers of times. An example of the results that we obtained is given in Figure 10 and 11.

6.3.3 *Geometric transformations*

In the absence of further attacks, robustness against geometric manipulations such as translation, rotation and uniform scaling is guaranteed by the normalization phase preceding both watermark insertion and detection. Some problems may arise when a geometric manipulation is accompanied by other attacks that may cause an error in the normalization phase. Hence we measured the robustness of the watermark when the marked mesh is filtered, degraded by noise addition and rotated. Specifically, in figure 12, the mesh obtained after a 22 degree rotation around the x axis, 11 degree rotation around y axis, 5 applications of Taubin filtering with $\lambda = 0.6307$ and $\mu = -0.6352$

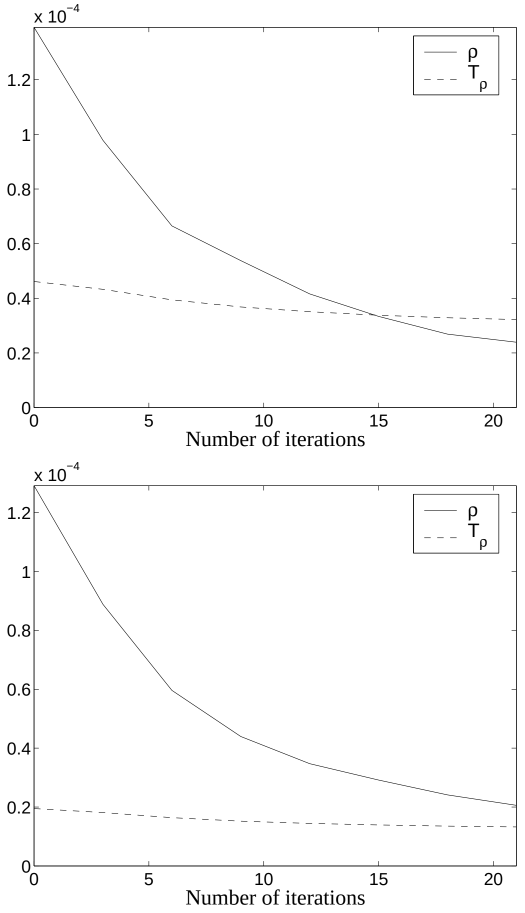

Figure 10: Robustness against Taubin filtering (the coefficients of the filter are $\lambda = 0.6307$ $\mu = -0.6352$ producing a strong smoothing effect). (Top) Bunny model. (Bottom) Venus model. For both the models watermarking parameters are the same used for the noise attack ($l = 3$, $\gamma = 0.0004$, $P_f = 10^{-8}$).

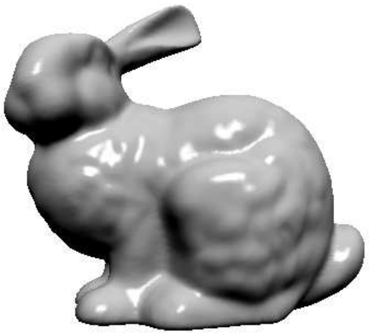

Figure 11: Visual effect of Taubin filtering (15 iterations). The watermark can be recovered even if the host mesh is severely degraded by the filter.

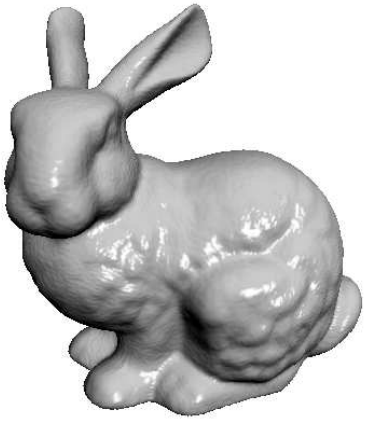

Figure 12: Robustness against combined attacks. The model has been watermarked with $l = 3$,$\gamma = 0.0004$ and $P_f = 10^{-5}$, then it has been smoothed by Taubin filter ($\lambda = 0.6307$, $\mu = -0.6352$, 5 applications), attacked with noise ($\Delta\ l_{med} = 0.1$) and rotated by 22 around x axis and 11 around y axis. The detector is still able to recover the watermark.

parameters and noise addition ($\Delta/l_{med} = 0.1$) is shown. Even in this case the watermark was successfully recovered ($\rho = 3.02 \times 10^{-5}, T_\rho = 3.06 \times 10^{-5}$).

6.3.4 Cropping

Finally we verified whether the watermarked can be recovered even when a part of the mesh is removed. This may cause a synchronization problem, since the mapping on W_{MAP} depends by the center of mass of the mesh. In fact, if a part of the mesh is cut, the position of the center of mass changes desynchronizing the watermarking map used by the embedder and that available at the detector. Hence in order to verify whether the watermark could be recovered on a subpart of the mesh, we assumed that absolute coordinates are used. It goes without saying that a proper synchronization algorithm must be developed in order to ensure full robustness against cropping (for a possible way to achieve this goal see section 7). Apart from the above considerations, the watermark exhibited an excellent robustness against cropping, as it is shown in Figure 13.

7. CONCLUSIONS AND FUTURE WORKS

In this paper we presented a new watermarking algorithm for 3D models. In order to cast the watermarking problem in a multiresolution framework, the algorithm is expressly designed to work with semi-regular meshes, thus making 3D wavelet analysis feasible. A particular mapping strategy is proposed to take into account the non-regular sampling of the 3D mesh. Correlation-based and geometric normalization allow the blind detection of the watermark and a good robustness against several attacks.

Several directions for future work remain open. First of all, we are planning to apply the watermark to subparts of the mesh [11]. By applying geometric normalizing to each subpart, robustness against combined cropping and geometric manipulations should be achieved. We are also going to evaluate the robustness of the watermark when the semi-regular mesh is converted to a irregular one, edited, and brought back to a semi-regular format by a remeshing operation. Finally, to further diminish watermark visibility, the possibility of modulating the watermark strength according to perceptual considerations will be investigated.

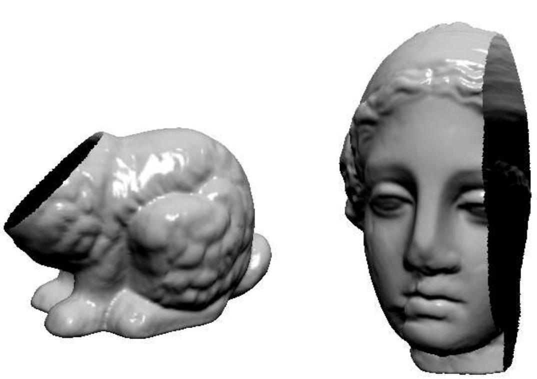

Figure 13: Cut Attacks. (Left) Bunny model watermarked with $l = 3$, $\gamma = 0.0009$ and $P_f = 10^{-8}$ and then cut by a plane. (Right) Venus model watermarked with $l = 3$, $\gamma = 0.0004$ and $P_f = 10^{-8}$ and then cut by a plane. In both cases the system is able to recover the watermark.

8. ACKNOWLEDGEMENTS

This work was partially supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this paper is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The semi-regular meshes used for the experimental results are made public available by the Multiresolution Modeling Group of Caltech University.
(http://www.multires.caltech.edu/software/pgc)

9. REFERENCES

[1] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, and M. Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3):485–493, 2003.

[2] M. Barni and F. Bartolini. *Watermarking Systems Engineering: Enabling Digital Assets Security and other Applications.* Marcel Dekker, 2004.

[3] O. Benedens and C. Busch. A frequency-domain approach to watermarking 3d shapes. *EUROGRAPHICS 2002*, 21(3), 2002.

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

[5] S. Gottschalk. Collision queries using oriented bounding box. *PhD Thesis, Department of Computer Science, University of North Carolina at Chapel Hill*, 1999.

[6] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 355–361. ACM Press, 2002.

[7] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. *SIGGRAPH '99*, pages 49–56, 1999.

[8] T. Harte and A. Bors. Watermarking 3d models. In *International Conference on Image Processing*, volume 3, pages 661 – 664, 2002.

[9] A. Kalivas, A. Tefas, and I. Pitas. Watermarking of 3d models using principal component analysis. In *Proceedings of Acoustics, Speech and Signal Processing (ICASSP'03)*, volume 5, pages 676–679. ACM Press, 2003.

[10] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3d polygons using multiresolution wavelet decomposition. In *Sixth IFIP WG 5.2 GEO-6*, 1998.

[11] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph.*, 22(3):954–961, 2003.

[12] S. M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*, volume II. Prentice Hall, 1998.

[13] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. Maps: multiresolution adaptive parameterization of surfaces. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 95–104. ACM Press, 1998.

[14] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, 1997.

[15] R. Ohbuchi, M. Akio, and T. Shigeo. A frequency-domain approach to watermarking 3d shapes. *EUROGRAPHICS 2002*, 21(3), 2002.

[16] R. Ohbuchi, H. Masuda, and M.Aono. Watermarking three-dimensional polygonal models. In *ACM Multimedia 97*, 1997.

[17] R. Ohbuchi, H. Masuda, and M.Aono. Watermarking three-dimensional polygonal models through geometric and topological modifications. *IEEE Journal on selected areas in communications*, 16(4):551–559, 1998.

[18] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. *SIGGRAPH '99*, pages 49–56, 1999.

[19] P. Schröder. Subdivision as a fundamental building block of digital geometry processing algorithms. *Journal of Computational and Applied Mathematics*, 149(1):207–219, Dec. 2002.

[20] P. Schröder and D. Zonin. Course notes: Subdivision for modeling and animation. In *Proc. SIGGRAPH '99*, 1999.

[21] V. Surazhsky, P. Alliez, and C. Gotsman. Isotropic remeshing of surfaces: a local parameterization approach. In *Proceedings of 12th International Meshing Roundtable*, 2003.

[22] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM Press, 1995.

[23] K. Yin, Z. Pan, J. Shi, and D. Zhang. Robust mesh watermarking based on multiresolution processing. *Computer and Graphics*, 25:409–420, 2001.