3D Watermarking Design Evaluation

Oliver Benedens[1] Jana Dittmann[2] Fabien A. P. Petitcolas[3]
[1] Freelancer Researcher, Germany, [2]Otto-von-Guericke University Magdeburg, Germany,
[3]Microsoft Research, Cambridge, UK

ABSTRACT

Due to the inherent combination of image, video, audio and 3D-models in new MPEG standards like MPEG-4, robust 3D-watermarking is becoming more and more popular to ensure data authenticity and integrity. Beside the design of 3D-watermarking schemes, the evaluation is necessary. In the paper we analyze currently proposed 3D watermarking algorithms for weaknesses that may possibly be exploited in watermarking attacks and derive according design principles for improving algorithms. The paper makes contributions in three respects: First we analyze properties with respect to preservation of surface continuity and symmetries. Second, we analyze how algorithms proposed by Ohbuchi et al. and Praun et al. compensate for errors introduced through imperfect mesh resampling. For Ohbuchi et al. algorithm, we explain how spectral decomposition can be applied in directions other than canonical coordinate system axes in order to minimize errors introduced to resampling. With an experimental prototyped algorithm based on spectral decomposition, we demonstrate a significant increase of robustness of features with respect to errors introduced in resampling. Third, we analyzed the general effects caused by polygon simplification on watermarking algorithms, which, from the results from Praun et al. can be considered as a "critical" operation. As a first result of our analysis, we present a modification of Praun et al. watermarking scheme detector trying to improve compensation for these effects.

Keywords: Watermarking algorithms, spectral decomposition, resampling, simplification, continuity, symmetries.

1. MOTIVATION

Beside very active research activities on image, video and audio watermarking we observed an increasing interest in watermarking of 3D-models, which is closely related to the development and usage of MPEG-4 data streams, which contain also 3D objects. Motivated by these developments in this paper we analyze a selection of currently proposed 3D watermarking schemes from the literature, with emphasis on [1, 2], for certain weaknesses which may be exploited in intentional attacks or cause limited detector performance in effect of every-day mesh processing tasks. The goal is to raise fundamental issues regarding the design and of today's robust 3D-watermarking algorithms. The insights and derived design principles given in this paper are very simple in nature, and our intention is to summarize these basic observations as design issues.

Based on the general knowledge about transparency, robustness and security as well as capacity as shown in [17] or in [18] and [19][1], the identified and discussed problems in our paper can be categorized into three main categories: First, we discuss transparency aspects: surface continuity and symmetry. If the watermarking algorithm does not preserve these, embedding strength has to be lowered to preserve visual quality. Second, we investigate robustness aspects: errors related to the mesh resampling process in watermark retrieval, in particular when meshes undergo operations such as retesselations. In general, we propose that the design of the core watermarking algorithm should reflect and compensate for errors related to the pre-processing of meshes, in particular the process of resampling. Third, we investigated another robustness issue: the effects of polygon simplification on watermark detectors. We draw particular attention to this operation, because from benchmarks by Praun et al. [1] it causes degradation of watermark although differences in terms of e.g. Hausdorff distances (see [24] for measuring) of original and watermarked copy are small. As it turns out, effects caused by polygon simplification are not captured by the uniform or gaussian noise type model and require to be reflected by the design of the mesh pre-processing (registration, resampling) or the detector.

The rest of this paper is organized as follows: We describe the underlying general problems, exemplify these problems for proposed watermarking algorithms and derive associated design principles.

As we see in image, video and audio watermarking discussed in several publications for example [17], [19] or [20] the watermarking positions and watermarking strength influence the transparency and also the robustness and security. The embedding process should not introduce any perceptible artifacts, that is, the watermark should not affect the quality of

[1] There are several publications about that topic and the mentioned should be seen as example publications from the variety.

the original signal. However, for robustness, the watermark energy should be maximized under the constraint of keeping perceptual artifacts as low as possible. Thus, there must be a trade-off between perceptual transparency and robustness. Similarly in the field of 3D-watermarking we recognized the following aspects which are important for removal and estimation attacks: If watermark locations and embedding energy are not carefully chosen, several kinds of artifacts may be identified by using flat shading or by the help of curvature and difference diagrams as proposed by Zhou and Pang [4] which use an discrete curvature approximation proposed by Desbrun et al. [3] and utilized by Benedens in [6] for highlighting artifacts caused by watermarking CAD dataset (fandisk) using his deformation based algorithm (targeting for smooth meshes [5]).

In the following we list a selection of possible traces left by a watermark embedding algorithm signaling presence of watermarks and possibly identifying modified regions for watermark localization and removal:

- Surface discontinuities: Not preserving surface continuity causes e.g. visible ridges in curvature diagrams.
- Violated global or local symmetries: Not preserving symmetries may cause a strange overall appearance of a 3D object.
- Degradation of surface tessellation quality or change in tessellation density: For a watermarking algorithm preserving the original topology, tessellation density *must* change in order of e.g. deformations. We note that small geometry changes of highly and equally tessellated surfaces not visible to the human eye become amplified and visible when applying e.g. polygon simplification. Either a modified region is subject to larger or lower extent of decimations with respect to its surroundings.
- Violation of constrains applying to single object components: Individual components of an object may no longer stitch/connect together as intended by the designer.
- Violated inter-object or scene constrains: Parts of scene are no longer assembled in a "convincing" way, e.g. furniture intersecting walls or a character standing inside the floor.

In this context we made experiments regarding the first three mentioned topics, documented in sections 2 and 3.

2. SURFACE CONTINUITY

Meshes or parts of them may represent smooth objects. These objects are often constructed using a modeling technique and object representation which preserves surface continuity up to a certain degree, e.g. through rational BSpline surfaces. The final result is converted to a mesh, which is only a discrete approximation of the underlying continuous surfaces. Everyday processing of meshes involves operations degrading surface continuity properties, most noticeably polygon simplification. In practice this does not necessary result in visible artifacts, because discontinuities may be smoothed out by applying interpolative shading (Gouraud, Phong, [11]) in the rendering pipeline. Additionally, artifacts may not be visible due to large viewing distance or meshes may be refined based on viewing distance utilizing subdivision surfaces techniques, a process which also may take place in the rendering pipeline.

In general, artifacts constituting high frequency noise can be removed using low pass filtering utilizing e.g. Laplacian smoothing [7].

Discontinuities can be visualized using flat shading or through the application of curvature diagrams as exemplified in figure 1. 3D watermarking schemes should take care not to harm surface continuity, in particular they should apply smooth deformations to smooth surfaces, otherwise visual quality will be degraded if embedding strength is not adequately limited and traces of watermark will become visible to attackers (e.g., in curvature diagrams, see references above).

One group of proposed watermarking schemes applies embedding strengths so small, they do not need to worry about continuity issues. This mainly involves algorithms for labeling or integrity verification, e.g. [8, 25]. The watermarking method proposed by Ohbuchi et al. for NURBS surfaces [26] embeds information by utilizing a reparameterization of surfaces and does not change geometry at all. [9] embeds information information by adding vertices and triangles and slightly perturbs a small number of original vertices only to remove "false traces" of watermarks.

The other group of schemes target applications like proof of copyright and fingerprinting and take following precautions to cope with the continuity problem: The scheme of Yin et al. [10] utilizes a multiresolution editing framework proposed by Guskov et al. [13]. In this framework a certain relaxation operator suitable for fairing irregular meshes is applied. Guskov et al. point out, that this operator achieves "nearly" C^1 continuity in practice.

Praun et al. [1] try to maintain surface continuity by applying deformations in respective normal or reverse normal direction of vertices. The amount and direction (normal/reverse normal) of a displacement of a vertex is determined by

a smooth scaling function (degree 3 or higher polynomial), which is evaluated with the shortest path of the vertex to a certain center of the deformed region as argument. The shortest path is determined by applying Dijkstra's algorithm to the original topology. We note, that using geodesic distances instead of shortest paths along edges or applying Dijkstra's algorithm to an over-sampled version of the original geometry would improve smoothness of deformations. Although no surface smoothness property is guaranteed, we could verify that the scheme achieves smooth deformations in practice (on smooth surfaces) in our experiments (see examples in figure 1 and figure 3).
However, embedding regions must be sufficiently large and embedding strength adequately limited. As we observed in our experiments in figure 3, the derby scaling function is visually more unsuspicious than the sombrero function. This is not surprising since the sombrero function causes more undulations near the border and therefore requires more "space" in order to reach the same level of "inconspicuousness". As Praun et al. documented in [1], the sombrero function performs best with respect to informational theoretic properties of the detector (low false positive probability).

Continuity properties of the watermarking scheme proposed by Ohbuchi [2], which is based on spectral decomposition [14], are difficult to evaluate, since there is no direct link from adding pseudo random noise to spectral coefficients to effects related to surface continuity in the spatial domain. Our experiments performed for spectral decomposition are summarized in figure 2. From our results we infer that changes to low frequency coefficients result in smooth deformations, however deformations are difficult to control because they depend entirely on mesh connectivity. We will further elaborate this point in section 4.1.

We note that none of the mentioned schemes of the second group is able to preserve planarity of areas. Praun et al. try to avoid planar regions by selecting regions with high frequency components. They select regions with largest approximation error in their respective edge-collapse history. This does not prevent an embedding region from containing planar regions or edges and therefore embedding strength must be adequately limited. In the scheme of Yin et al. [10] planar areas or edges are not excluded from deformations. Vertices of these areas may be either part of the edge collapse hierarchy of a vertex displaced at the coarse level, or they may be adjacent to a vertex in this hierarchy and are therefore repositioned in the relaxation process. Both schemes may be improved by restricting embedding to suitable regions (curved, non-planar, non-regular, high frequency components, no local symmetries).
We note a further important observation in experiments of figure 1: In the curvature diagram we see a regular pattern in the embedding region. This is because the original region has regular topology (in terms vertices are connected and spaced in a uniform way), tessellation density is rather low and measuring inter mesh vertex distances along existing edges and deriving weights for displacements from these distances amplifies this regular topology. This problem may be resolved using an oversampling of the mesh for embedding. The watermarking method based on spectral decomposition proposed in section 4 utilizes such an oversampling. The main disadvantage is, that projecting original topology on oversampled topology causes degradation of embedded information.
There is a related problem: All schemes previously discussed do not alter the topology of the mesh (they do not add vertices or faces). On one hand this is desirable because we don't want to raise complexity of a mesh (and therefore space requirements) through the process watermarking and scalar attributes as vertex/face colors or texture coordinates need not to be recalculated. On the other hand, discontinuities may arise because the deformed mesh represents a non-optimal subsampling of a smooth mesh (over-sampled representation).

In summary, maintaining surface continuity, surface planarity or symmetries, the latter ones are discussed in detail in the next section, pose problems for currently proposed watermarking schemes. Adequately limiting embedding strengths currently solves these.

3. SYMMETRIES

Transparency issues are not only concerned with local artifacts manifesting as high frequency noise. Watermarking may falsify content properties and degrade visual quality on a more a global scale, in a more subtle way: Yu et al. in [21] observe techniques for content-based graph authentications and determine, that graphs are difficult to watermark because of their binary nature. A minimal alteration of bits in a binary graph can results in a substantial change in the graph's appearance and content properties. A similar problems arises with respect to symmetries: In 2D vector based data sets, e.g. of digital circuits (VLSI), as well as for most 3D meshes. Preservation of (local) symmetries in topology was a design issue in construction of surface subdivision schemes [30].
In the 3D case, preservation of symmetries is not stated as a design goal by most authors. In [5], Benedens proposed to preserve a global reflective symmetry, which is allowed to be "slightly broken", by utilizing symmetrical derformations for watermark embedding and pointed out, preserving symmetry halves the available capacity. In our analysis we differ

between global and local symmetries. For example, the teapot in figure 1 exhibits a global reflective symmetry while the lid exhibits a local rotational symmetry.

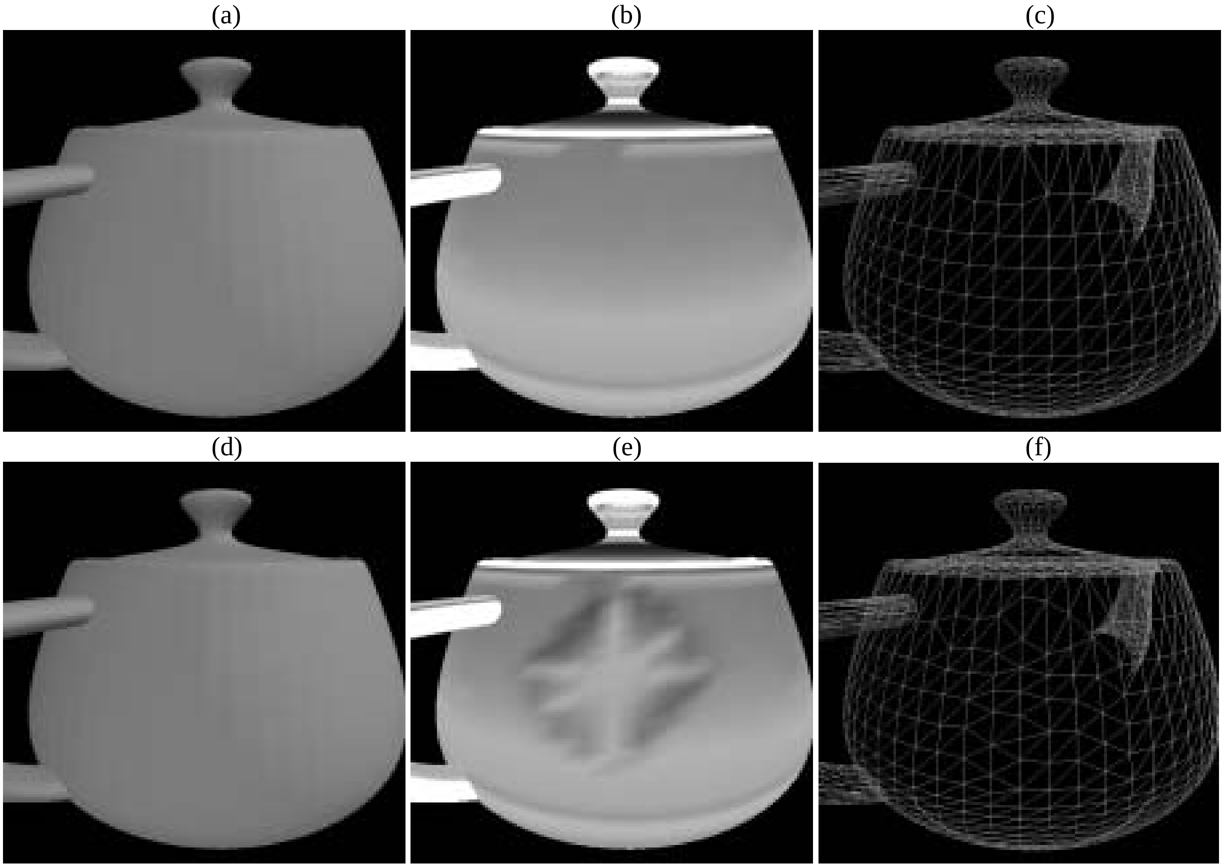

Figure 1: Amplification of watermarks and degradation of tessellation quality of simplified meshes: We apply a deformation as described by Praun et al [1] using the derby filter and embedding strength of 0.2 % (of bounding box diameter) (watermark value *w* set to 1). First row, form left to right: (a) original teapot flat shaded, (b) mean curvature diagram as proposed by Zhou and Pang [4] for mesh surface comparisons using discrete approximation of mean curvature given in [3] and utilized by Benedens in [6] for revealing presence of a watermark for his free form deformation based algorithm for a CAD mesh (fandisk-mesh), (c) original mesh polygon simplified from 16256 faces to 6000 faces using qslim package [15] (edge collapses based on Quadric error metric). Second Row: (d) watermarked mesh flat shaded, (e) mean curvature diagram of watermarked mesh and (f) watermarked mesh simplified as in (c). Observations: In addition to [6], we notice a regular pattern, an amplification of regular topology, which reasons we discuss in text. In general we see, that for regular tessellated meshes, watermarking affects mesh quality (tessellation quality) of simplified versions.

Using the scheme of Praun et al., preserving a global reflective symmetry, which we allow to be “slightly broken”, could be achieved by symmetrically placing feature-regions on the mesh and choosing matching deformation directions for corresponding regions. In the algorithm proposed by Yin et al. [10], deformations are applied to a coarse scale end of the mesh pyramid, then details are added back in. The coarse scale representation is generated through polygon simplification where edges are collapsed based on local evaluation of the quadric error metric [16]. For this simplification method we found it to preserve visual appearance with respect to symmetries, however small deviations in geometry or topology cause irregularities in the tessellation of the final result. This complicates the task of achieving deformations preserving symmetry on the coarse level and hence on the finest level. We exploited these issues in figure 1, where we utilized Qslim polygon simplification for amplifying effects caused by a watermark through local irregularities in the simplified mesh. We are not aware of any proposed polygon simplification scheme taking symmetry issues into account.

In figure 2 we exemplify that mesh spectral decomposition and editing spectral coefficients, utilized in [2], does not preserve symmetries in general.

So far, robust 3D watermarking schemes do not handle these symmetry issues explicitly. Possible countermeasures for the mentioned problems are adequately limiting embedding strength, selecting suitable embedding regions, e.g. without rotational and reflective symmetries to be observed. However detection of *local* symmetries might prove to be intractable in practice. At this point further investigations are considered as future work: If it is in fact intractable, watermarking algorithms need to severely limit embedding strengths in general in order to preserve mesh quality. The positive side would be, attacks utilizing symmetries for detecting a watermarks presence or removal would also be intractable. For shape analysis with respect to *global* reflective symmetries, according descriptors have been proposed, e.g. [27].

A variety of meshes used in demonstrations do not exhibit *global* reflective symmetries, e.g. the Stanford bunny, happy buddha or dragon models. In the other extreme, CAD meshes of engines or machinery, e.g. the distcap model in [28], exhibit local rotational symmetries. For CAD like meshes, embedding strength has to be limited in general in order to maintain high frequent components such as ridges, edges and planar areas in the embedding process (as can be seen in the fandisk test case of [1]).

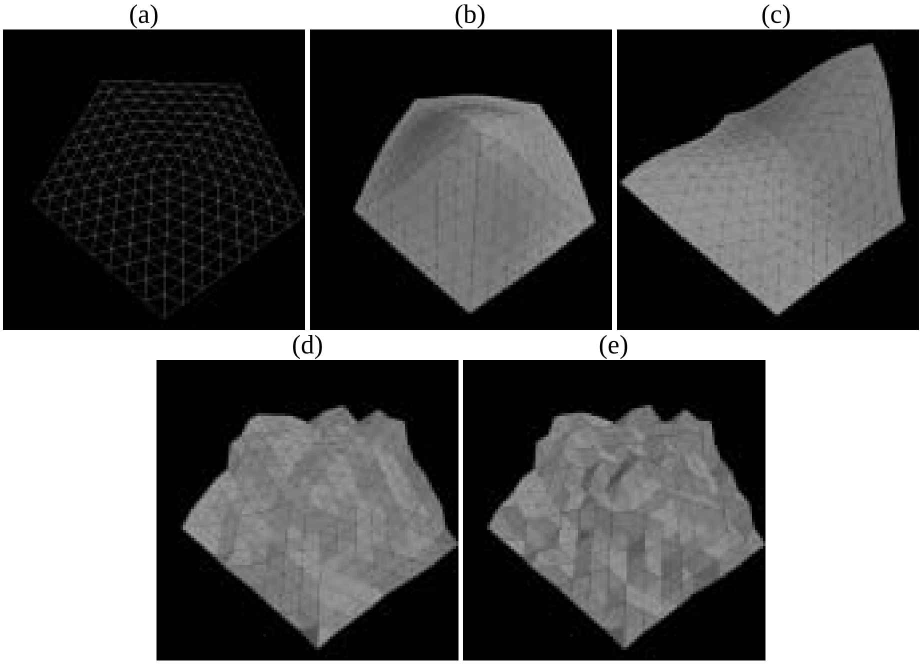

Figure 2: Experiments on symmetry- and continuity preservation of mesh editing based on spectral decomposition. All vertices are represented with respect to a local frame whose origin is at the center vertex and the Y-axis matches the center vertex normal. Only a spectral decomposition and editing for Y-coordinates is applied, therefore all deformations are performed in vertex normal (or reverse) direction of the center vertex. (a) original (planar) mesh, (b) adding $-0.05*\mathit{bbd}$, *bbd* is bounding box diameter of mesh depicted in (a), to 6th-lowest frequency spectral coefficient preserves all 5 reflective symmetries, (c) adding $-0.05*\mathit{bbd}$ to the 4th-lowest frequency coefficient destroys all reflective symmetries. Second, row: Adding uniform noise $\in [-0.002*\mathit{bbd}..0.002*\mathit{bbd}]$ to (d) 100 of 181 spectral coefficients and (e) all coefficients. We observe: Deformations do not preserve symmetries in general, changing certain coefficients causes largest changes on the boundary (problem when trying to achieve smooth blending of displacements across borders of embedding regions) and altering low to medium frequencies generates noticeable noise if embedding energy or range of embedding-coefficients is not carefully limited.

4. RESAMPLING

In this section we perform experiments to investigate how errors introduced in resampling affect the performance of a watermarking system and we develop design criteria and methods as countermeasures to minimize these effects.

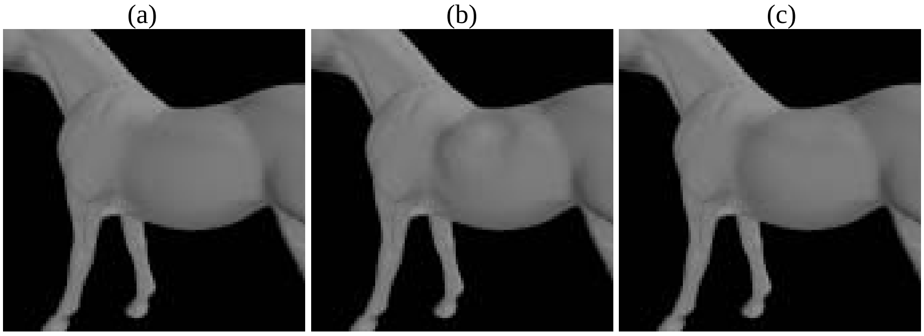

Figure 3: Experiments on continuity properties of mesh editing based on the watermarking scheme proposed by Praun et al [1]. We are interested in the "behavior" of large embedding strength. (a) Original mesh, (b) one region deformed applying the algorithm of Praun et al., using sombrero scaling function and embedding energy of 1% (with respect to bounding box diameter), (c) using derby scaling function. In the sombrero scaling function case, which results in [1] document to be the best performing scaling function with respect to robustness of the resulting algorithm, effects are visible, in the derby case not.

Problems related to resampling were first mentioned by Praun et al [1]. In their watermarking scheme they require projecting the original mesh topology onto the watermarked copy prior to watermark retrieval. They discuss how errors arise from this process and propose an energy minimization approach, later adopted by Yin et al [10], in which original vertices are displaced towards the watermarked copy while minimizing deformation energy.
In addition Praun et al. core algorithm is designed effectively to minimize resampling induced errors and we give an analysis in this section. In [5], Benedens proposed an algorithm for which he tried to minimize resampling related errors simply by using resampling directions matching the embedding directions or reverse.

In [28], Ohbuchi et al. proposed a robust watermarking method based on spectral decomposition [14] and further refinded it with respect to robustness to retesselations and time complexity in [2]. By investigation of their algorithm, we found, that resampling induced errors are compensated mainly by repetitive embedding and we propose a general method for minimizing resampling-induced-errors based on spectral decomposition in *non-canonical directions*. We work out improvements and recommendations for design in the next section.

4.1. SPECTRAL DECOMPOSITON IN NON-CANONICAL DIRECTIONS

Let $V=v_1,..,v_n$ be the vertices of mesh M with $v_i=(v_{ix}, v_{iy}, v_{iz})^T$ $(1 \le i \le n)$. As proposed by Ohbuchi in [28], denote K as the nxn Kirchhoff matrix, calculated from the connectivity information of mesh M. K is a real valued symmetric matrix and has n real valued eigenvalues $\lambda_1,..,\lambda_n$, sorted in ascending order and associated orthogonal normalized eigenvectors $e_1,..e_n$. Let E denote the orthogonal nxn matrix with e_i in column i $(1 \le i \le n)$. In Ohbuchis algorithm spectral coefficient vectors $r_j=(r_{j1},..,r_{jn})^{T,}$, $j \in \{x,y,z\}$ are calculated through projection $r_j = E^T (v_{1j},..,v_{nj})^T$.
In watermark embedding, the spectral coefficient vector r_j is updated to $r_j'=(r_{j1}',..,r_{jn}')^T=r_j+c_j$, with $c_j=(c_{1j},..,c_{nj})^T$ being a key derived pseudo random noise sequence of length n and updated vertices $V'=v_1',..,v_n'$ are calculated by $(v_{1j},..,v_{nj})^T = E \cdot r_j'$.
One important thing to note is the equality

$$E \cdot r_j' = (v_{1j},..,v_{nj})^T + E \cdot c_j \tag{1}$$

$j \in \{x,y,z\}$. Displacements to vertices only depend on c_j and E, which is derived from mesh connectivity information only. They do not depend on the vertex coordinates. From (1) we developed our idea to apply spectral decomposition in directions other than directions of axes in canonical coordinate frame: in each respective vertex normal direction. The reasoning to chose normal/reverse normal directions in particular is to minimize resampling induced errors by minimizing angles between resampling and deformation directions (close to 0 or 180 degrees) as stated by Benedens in [5] for his watermarking algorithm.

In our proposed variant there is only one pseudorandom sequence $c=(c_1,..,c_n)^T$ and spectral coefficient vector which is set to $r=(0,..,0)^T$ and the vector $d=(d_1,..,dj)^T = Ec$ contains displacements of vertices in their respective normal direction. Updated vertices are determined as follows:

$$v'_i = v_i + n_i \cdot E^{(i)} \cdot c \tag{2}$$

for $1 \leq i \leq n$. n_i is vertex normal (determined by averaging normals of faces adjacent to vertex) of original vertex v_i and $E^{(i)}$ denotes the i-th row-vector of E. In the following we perform experiments of our experimental variant in comparison to independent spectral decomposition in canonical (x-,y-,z-) directions.

For our investigations we implemented experimentally an algorithm and used this (early) prototype of a watermarking system based on spectral decomposition which differs to Ohbuchis scheme in the in following respects: Prior to embedding and retrieval a simple geometry consisting of 5 connected triangles is projected onto the object embedding region and further subdivided through 1:4 triangle splits. Each newly introduced vertex position is found by projecting the edge-midpoint into direction of averaged face normals/reverse normals of edge adjacent faces and selecting the nearest intersection point with the object surface. We calculate local coordinates of each introduced vertex based on local frames derived from geomertry of previous (coarser) level and predicted positions using the butterfly subdivision rule [30]. We apply I iterations of this projection step and end up with a mesh with a priori known connectivity which allows for precalculation of eigenvectors of the associated Kirchhoff matrix K. We determine nearest intersections of original vertices in direction of vertex normals (or reverse) with subdivided mesh on level I and compute baricentric coordinates for intersection points with respect to faces of the subdivided mesh.

For the original mesh, we “assign” zero spectral coefficients (remind, that displacements do not depend on original coordinates). In case of independent spectral decomposition in canonical (x-,y-,z-) directions, altering a spectral coefficient causes a changes to coordinates in the respective canonical direction. In non-canonical directions case, altering a coefficient causes displacements of vertices in their respective vertex-normal or vertex-reverse normal direction. For canonical-directions we alter 50 “lowest” coefficients for embedding in each decomposition, giving a total of 150 coefficients. In the non-canonical-directions case we alter 50 lowest coefficients. Assume editing (of coefficients) takes place on subdivision level J. After altering coefficients, we add in details of subdivision levels J+1 to I and update the original mesh vertices using previously calculated baricentric coordinates.

In the following example, the editing take place on subdivision level J=5 and I=6.

For our prototype-watermarking algorithm we realize *binary features*, each of the altered coefficients stores one bit of information. Denote the bounding box diameter of mesh M with *bbd*. To embed “1”, the initial spectral coefficient C, initially zero, set to C=0.008**bbd*, to embed a “0”, C is set to -0.008**bbd*. A “1” is said to be embedded, if the sign of a coefficient is ≥ 0, “0” otherwise. Figure 1 shows regions before and after embedding.

The left image of figure 5 shows the feature values retrieved from the region displayed in figure 4c) in which x,y,z directions were treated independently.

Prior to retrieval we apply a simple resampling by projecting original region of vertices in their respective (reverse) normal directions and selecting the nearest intersection point with the watermarked mesh repectively.

Assume coefficients were set to values $c_{1,j},..,c_{50,j}$ in embedding and $c_{1,j}',..,c_{50,j}'$ ($j \in$ {x,y,z}) denote the retrieved values. The amount of falsification for coefficient $c_{i,j}$, drawn in left image of figure 5, is $f_{i,j} = (c_{i,j} - c_{i,j}')/bbd$ ($1 \leq \mathrm{i} \leq 150, j \in$ {x,y,z}). A feature is falsified if $f_{i,j} \leq -0.008$.

As can be seen, for most coefficients, the alteration reduces the stability of encoded value. Values corresponding to x-direction are the most stable ones. The reason is quite simple: The face normals of the embedding region are close to the x-direction and therefore to resampling directions.

In the right image of figure 5, 50 coefficients and displacements in “normal direction” were utilized. The according region is displayed in figure 4d). Here the degradation of the stability of feature values is significantly smaller.

One may argue, why there is still a loss of information even no attack was applied. The reason is, adding of details of subsequent detail levels and additional scaling of embedding strength towards the border (to achieve smooth transition from embedding region to outer region) cause a loss of information.

When treating x,y,z directions independently, we loose 59 of 150 bits, while in the “normal”-direction case we loose 2 of 50 bits.

Applying independent spectral decomposition in x,y,z directions increases the number of features by factor of three. However, this does not necessarily improve information theoretic properties of the watermark detector: Assume spectral decomposition is applied in “normal direction” and the accomplished falsification probability for each of 50 binary features is, hypothetically, f_{err}=0.1 (source of errors are inaccuracies in registration- and resampling-process or e.g.

attacks). Independent spectral decomposition in canonical directions raises number of features to 150, for which we assume the falsification probability to be f_{err}'. This allows for embedding the original 50 bits three times and the feature values be determined through majority voting. Under these assumptions the falsification probability for each of the 50 bits drops to f_{err}''

$$f_{err}''=1-\sum_{i=0}^{1}\binom{3}{i}(f_{err}')^{i}(1-f_{err}')^{3-i} \tag{3}$$

In our example we achieve $f_{err}''=0.1$ for $f_{err}':=0.1958$. This means, if $f_{err}'>0.1958$, decomposition in normal direction would yield the better detector.

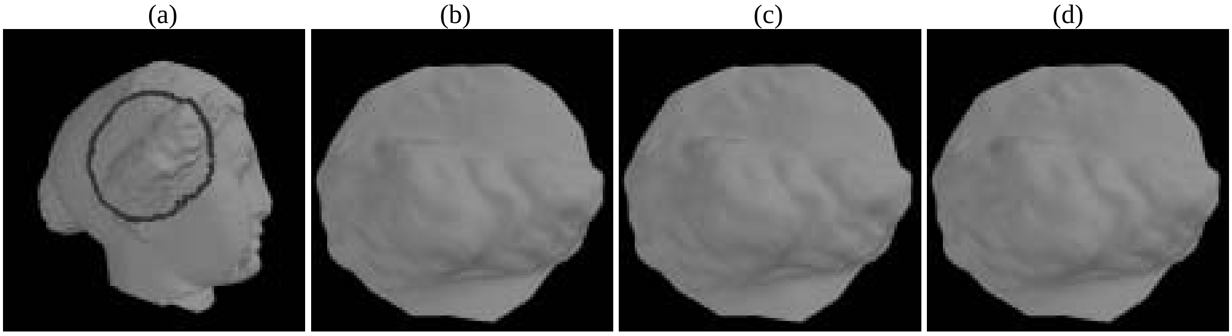

Figure 4: (a) Original venus head with region marked for embedding, (b) original region enlarged, (c) region carrying 150 bit watermark (independent embedding in x,y,z direction) and (d) region carrying 50 bit watermark (embedding in “normal” direction). Embedding strength was 0.8% percent per coordinate and 0.8% for displacements in normal direction (relative to bounding box diameter). For spectral decomposition in canonical directions (c) and our proposed variant (d), no difference in visual quality can be perceived.

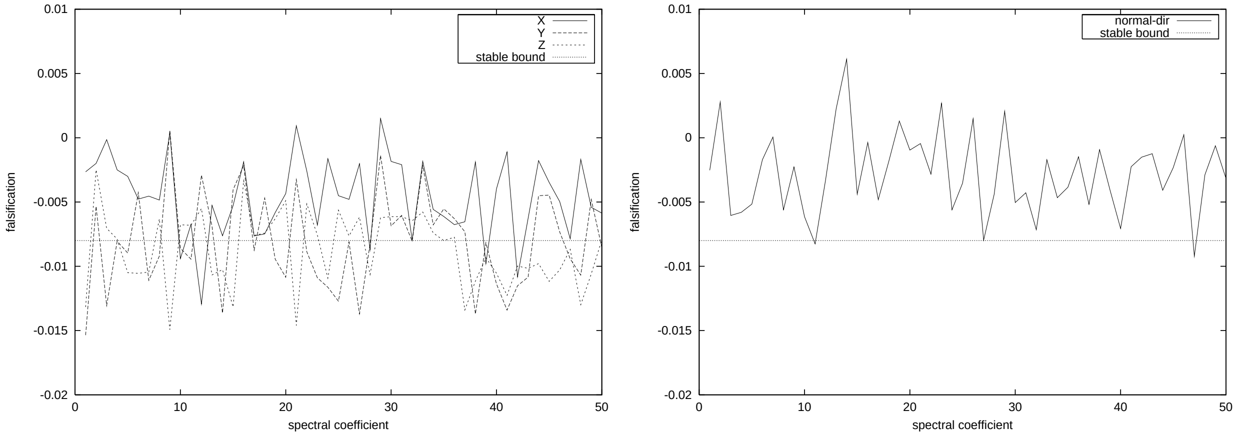

Figure 5: Left image shows stability of features if spectral decomposition is applied in x,y,z-directions independently. Right image shows results if decomposition is applied in “normal-direction”. A negative values means alteration of a binary feature values in direction leading to falsification of the feature. When the horizontal line at y=-0.008 is reached, the feature value is falsified.

4.2. OPTIMALITY OF PRAUN ET AL ALGORITHM WITH RESPECT TO RESAMPLING

Next we briefly discuss, why the algorithm proposed by Praun et al. [1] can be considered optimal with respect to compensation of resampling induced errors:

Assume a gaussian distributed feature value *w* was embedded into a region consisting of original vertices $V=v_1,..,v_n$ and associated surface normals $N=n_1,..,n_n$. In embedding, each vertex was displaced to position $v_i' = v_i + n_i \cdot c_i \cdot w$ where $c_i := s(r_i) \cdot e \cdot h$. r_i denotes the length of shortest path of v_i to its region center. $s()$ is a smooth scaling function for which Praun et al. stated three alternatives in their paper, termed *hat*, *derby* and *sombrero*, used to assign a scaling factor between 1 and 0 (0 for vertices with distances larger than minimum distance of a region boundary vertex). *e* denotes the embedding energy used in general, while a region specific *h* additional scales this energy (to account for the amount of high frequencies in the region).
Now assume positions v_i'' where assigned to v_i after registration and resampling ($1 \le i \le n$). Then the embedded value *w* is estimated by minimizing function *f(x)* (we just give an alternative formulation of their detector):

$$f(x) = \sum_{i=1}^{n} (n_i * v_i'' - d_i - c_i x)^2 \qquad (4)$$
$$d_i := v_i * n_i$$

The estimated *w*, denoted with *w'*, is determined as follows:

$$w' = \frac{k_1 c_1 + .. + k_n c_n}{c_1^2 + .. + c_n^2} \qquad (5)$$
$$k_i := n_i * v_i'' - d_i \quad (1 \le \mathrm{i} \le \mathrm{n})$$

We see in (4), that *w* is chosen so squared differences to distances of resampled vertices to planes centered at original vertices with original surface normal are minimized. Therefore errors in resampling process, in particular applying to components close to tangential plane of surface normal are compensated (by not taking them into account).
We summarize, that errors associated with the resampling process have to be considered in the design of a watermarking scheme. For spectral decomposition, we identified measuring differences of original vertices and their resampled counterparts in normal directions instead of independent treatment of “dimensions” as one possibility to accommodate for errors in registration- and resampling steps. In general there are two possibilities for improving algorithms with respect to resampling errors, both realized by Praun et al [1]: Improve accuracy of resampling,-method e.g. through a energy minimization approach or reflect possible inaccuracies of mesh pre-processing (registration, resampling) in the design of the core watermarking method (detector).

5. POLYGON SIMPLIFICATION

From benchmarking results of Praun et al. [1], we consider polygon simplification as a critical operation. In order to investigate the reasons, we first determine the effects (distortions) caused by polygon simplification in experiments. Then we propose a method for increasing the performance of the method proposed by Praun et al. based on insights about “nature” of polygon simplification and justify it in experiments.

5.1. NATURE OF POLYGON SIMPLIFICATION

We first determine effects (distortions) caused by polygon simplification by “reverse engineering” through following experiment: We randomly selected *N*=50 points, termed region centers, from the vertex set of a bunny mesh.
Then, each region center constitutes a test case as follows: We collect all vertices whose shortest path along edges of the mesh to the region center is less or equal than *range*=0.1**bbd*, forming together a region. Then we apply Qslim polygon simplification [15, 16] to the mesh. Next we uniformly resample the coarse mesh with 1000 vertices using the method proposed by Funkhauser [29] et al. and register the resampled mesh with the original (rigid registration+uniform scaling). The resulting transformation is applied to the non-resampled coarse mesh on which we cast rays from original vertices in original mesh normal- and reverse normal and chose the respective nearest intersection point with simplified mesh. Then we sort distances for intersection points to their respective original vertex of all regions into 20 intervals and determine the mean of signed distances for each interval. A distance is sorted to an interval based on the original region points shortest path length to its region center: The *i*-th of 20 intervals ($1 \le i \le 20$) is associated with distance interval $[d_{i-1}, d_i)$, whereby d_0=0 and $d_i = \sqrt{i/20} \cdot range \cdot bbd$ ($1 \le i \le 20$). This way, the surface “occupied” by each interval is equal if points were lying in a plane. In figure 6 we show cases for two

regions: We see that simplification caused an approx. constant displacement of region vertices in normal or reverse vertex normal direction.

We note that effects observed are due to the shrinking effect when simplifying a (mostly) convex mesh and part of these effects is still present after global registration (including rescaling). From this observation we developed two general idears for compensating for the mentioned displacements: Local registration of regions, as it was is done by Ohbuchi et al. [2], and/or compensation in the detector of a watermarking scheme which we discuss next.

5.2. PRAUN ET AL ALGORITHM AND POLYGON SIMPLIFICATION

From the results of previous two sections we deduce that if we would use a region for embedding a watermark data as proposed by Praun et al. random uniform or gaussian noise can be expected to "cancel itself out". From effects of polygon simplification previously studied, we propose to add a constant displacement *e* to *f(x)* in their detector (compare with (4)):

$$f(x,e) = \sum_{i=1}^{n} (n_i * v_i'' - d_i - c_i x + e)^2 \tag{6}$$

$$w' = \frac{(k_1 c_1 + .. + k_n c_n) \cdot n - (k_1 + .. + k_n) \cdot (c_1 + .. + c_n)}{n \cdot (c_1{}^2 + .. + c_n{}^2) - (c_1 + .. + c_n)^2} \tag{7}$$

w' is determined as solution of d*f(x,e)*/d*w*=0 and d*f(x,e)*/d*e*=0). Definition of d_i and k_i as given in (4) and (5).

In figure 7 we applied experiments for the bunny mesh using the original algorithm, the modified algorithm and a third variant in which the embedded value is determined from (4) using only the original and displaced center vertex of the region.

The setup was as follows: As described in 5.1 we determined *N*=50 region centers. Each region contains the vertices whose shortest path to region center vertex was ≤ 0.16**bbd*. Then we draw 50 values from a gaussian N(0,1) distribution (we reject values with absolute values ≥ 2.5 or ≤ 0.1). For each value we apply the following steps: We embed the value in region using *e*=0.01 and *h*=1 and derby scaling function, simplify the mesh from 69451 down to 1000,900,..,400 faces using QSlim, uniformly resample the simplified mesh with 1000 points and carry out a conventional rigid registration (plus uniform scaling). Then we apply a simple resampling by projecting the original vertices in normal direction on the simplified mesh and retrieve the embedded value using all three types of detectors. We removed features, for which distances where >0.1**bbd* or for which retrieved feature values were ≥ 3.0 (absolute values).

The left image of figure 7 shows the mean squared feature difference for varying rates of polygon simplification. Our modification successfully lowers differences with respect to the original algorithm, however the improvement is rather small. In the right image of figure 7, we list, according to the left image, the associated (analytically determined) false positive probability for each test case which is calculated as proposed by Praun et al (determine correlation coefficient, then turn it into a probability using student-t test).

From the experiments performed, we propose, similar to section 4, to compensate these affects through local registration of embedding regions as it was is done by Ohbuchi et al. [2] or when actually deriving watermark feature values from regions, as it was done in our experiments. Since the improvement was noticeable but not significant, we consider experiments utilizing local mesh registrations as future work.

6. CONCLUSIONS AND OUTLOOK

We investigated a selection of proposed robust watermarking algorithms with respect to surface continuity, symmetry preservation, errors related to the resampling process and effects of polygon simplification. We derived simple 3D design principles for watermarking algorithms: Carefully selection of embedding regions and adjustment of embedding strength. The design of detectors, registration- and resampling-methods should reflect errors associated with resampling process and in particular effects of polygon simplification, e.g. shrinking effects in simplification of (mostly) convex meshes.

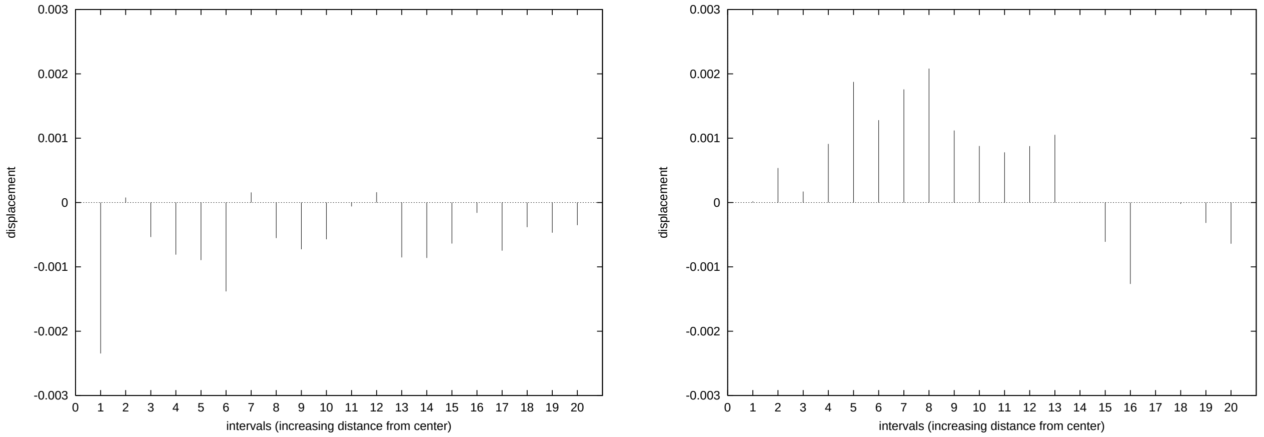

Figure 6: Left image: Signed distances of original vertices (in normal direction) to simplified bunny mesh (800 faces) in relation to original vertices distance to center of embedding region (x-axis). Vertices of one embedding region are sampled to intervals depending on distance to the region-center and the mean value for each interval is plotted. Interval-borders increase from left to right and are chosen so intervals cover same area on a circle. Right image: Same as left, for another embedding region. Please note that we already compensated for the shrinking effect due to simplification in mesh registration (the bunny mesh is mainly convex).

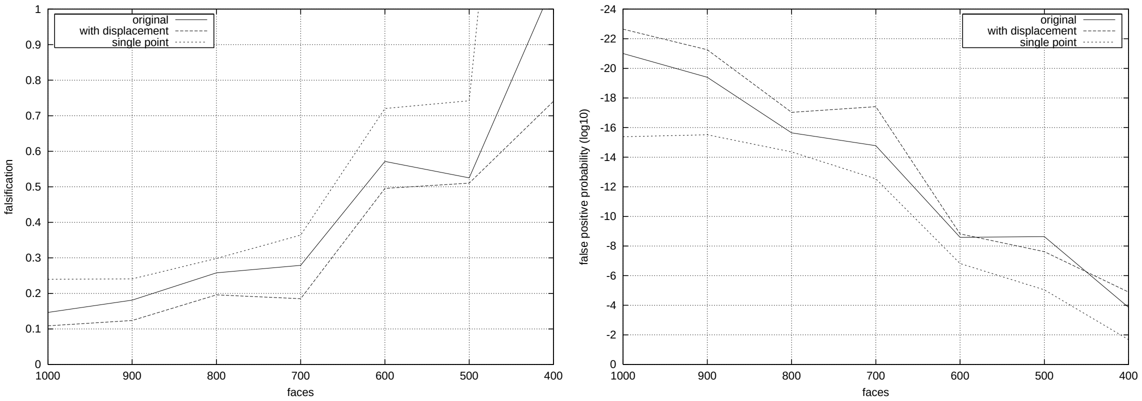

Figure 7: Left image: Mean squared difference of original feature and feature derived from simplified mesh for varying simplification rates. Interestingly, the single point variant follows both other detectors closely for simplification rates ≤800 faces. Right image: Resulting false positive probability determined as proposed by Praun et al. for varying simplification rates. Performance ranking is similar to the left image, however we note that minimum mean distortion of vertices does not necessarily correspond to minimum false positive probability as can be seen for the case of simplification to 500 faces where the original algorithm outperforms our proposed variant.

Insights and principles lead to attacks on 3D watermarking schemes: Presence of watermarks may be revealed through curvature diagrams, as proposed by Benedens in [6], or through amplifying irregularities in tessellations (through polygon simplification). Particular attacks may be designed, exploiting mesh symmetries, either by locating watermarks through “broken” symmetries or removal of watermarks through restoring symmetry, although the task of identifying local symmetries might prove to be intractable. In general, exploiting resampling induced errors through retesselations may disable a detector. Observations of the polygon simplification test case lead to coarse scale deformations of a mesh, which should “cause” similar problems, but maintain surface details. Based on our discussion we motivate to integrate a standard set of associated tools for design evaluation into the Stirrmark benchmark, which currently consists of image and audio tools [18].

There are varieties of not discussed issues in this paper and it is worth to consider further evaluations in the field of removal attacks, geometrical attacks, cryptographic attacks and protocol attacks, see [22] or [23], for example mosaic or coalition attacks (averaging of copies).

Acknowledgements

The bunny mesh originates from the Stanford Scanning Repository, the horse mesh from the Large Geometric Models Archive at Georgia Tech., the venus mesh from the Cyberware website and the teapot was created from Standard Procedural Database.

References

[1] E. Praun, H. Hoppe and A. Finkelstein. Robust Mesh Watermarking. In: SIGGRAPH 99, pp. 69–76, 1999.

[2] R. Ohbuchi, A. Mukaiyama and S. Takahashi. A Frequency-Domain Approach to Watermarking 3D Shapes, In: EUROGRAPHICS 2002 Proceedings, Saarbrücken, September 2-6, 2002.

[3] M. Desbrun, M. Meyer, P. Schröder and A. Barr. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In: SIGGRAPH 99 Proceedings, 1999.

[4] L. Zhou and A. Pang. Metrics and Visualization Tools for Surface Mesh Comparison. In: Proc.of SPIE conference on Visual Data Exploration and Analysis, 2001. http://www.cse.ucsc.edu/research/avis/mesh.html

[5] O. Benedens. Robust Watermarking and Affine Registration of 3D Meshes. In: Proc. of 5th International Workshop on Information Hiding, Noordwijkerhout, Netherlands, October 7-9; Springer (Lecture Notes in Computer Science), 2002.

[6] O. Benedens. 3D Watermarking Algorithms in Context of OpenSG Plus. Technical Report 02i002-figd, 2002. Download through http://publica.fhg.de/ (person=”benedens”).

[7] G. Taubin. A Signal Processing Approach to Fair Surface Design. In: Computer Graphics Proceedings, pp. 351-358, August 1995.

[8] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking Three-Dimensional Polygonal Models. In: Proc. of the ACM International Conference on Multimedia ’97, pp. 261–272, 1997.

[9] X. Mao, M. Shiba and A. Imamiya. Watermarking 3D Geometric Models Through Triangle Subdivision. In: Proc. of SPIE Vol. 4314, Security and Watermarking of Multimedia Contents III, pp. 253-260, San-Jose, January, 2001.

[10] K. Yin, Z. Pan, J. Shi and D. Zhang. Robust mesh watermarking based on multiresolution processing. In: Computers & Graphics, vol. 25, pp. 409-420, 2001.

[11] J. Foley, A. van Dam, S. Feiner, J. Hughes. Computer Graphics. Principle and Practice. Addison Wesley, Reading, MA, 1990.

[12] C. Loop. Smooth Subdivision Surfaces Based on Triangles. Master Thesis, University of Utah, 1987.

[13] I. Guskov, W. Sweldens and P. Schröder. Multiresolution Signal Processing for Meshes, SIGGRAPH 99 Proceedings, 1999.

[14] Z. Karni and C. Gotsman. Spectral Compression of Mesh Geometry, SIGGRAPH 2000 Proceedings, 2000.

[15] QSlim simplification package available from http://graphics.cs.uiuc.edu/~garland/software.html.

[16] M. Garland, P. Heckbert. Surface Simplification Using Quadric Error Metrics. SIGGRAPH 97 Proceedings, 1997.

[17] I. Cox, M. Miller, J. Bloom, Digital Watermarking, 2002 Academic Press, San Diego, USA, ISBN 1-55860-714-5.

[18] F. Petitcolas, R. Anderson: Evaluation of copyright marking systems. In: Proc. of IEEE Multimedia Systems, Multimedia Computing and Systems, June 7–11, 1999, Florence, Italy, Vol. 1, pp. 574–579, 1999.

[19] J. Dittmann, Digitale Wasserzeichen, Springer Verlag, ISBN 3 - 540 - 66661 - 3, 2000.

[20] J. Seok, J. Hong, and J. Kim: A Novel Audio Watermarking Algorithm for Copyright Protection of Digital Audio, ETRI Journal, Volume 24, Number 3, June 2002, pp. 181-189.

[21] Heather Yu, Xiangyang Kong, Wayne Wolf: techniques for Content-based Graph Authentication, IEEE MultiMedia, October-December 2001, Vol. 8, No. 4, pp. 38-45, ISSN 1070-986X, 2001.

[22] M. Kutter, S. Voloshynovskiy, A. Herrigel: Watermark Copy Attack, In: Proceedings of SPIE: Security and Watermarking of Multimedia Contents II, 24–26 January, San Jose, California, USA, Vol. 3971, ISBN 0-8194-3589-9, pp. 371–381, 2000.

[23] S. Pereira, S. Voloshynovskiy, M. Madueño, S. Marchand-Maillet and T. Pun, “Second generation benchmarking and application oriented evaluation”, in Information Hiding Workshop III, Pittsburgh, PA, USA, April 2001. http://www.cl.cam.ac.uk/~fapp2/watermarking/evaluation/

[24] N. Aspert, D. Santa-Cruz and T. Ebrahimi, MESH: Measuring Errors between Surfaces using the Hausdorff Distance, Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 705-708, 2002.

[25] B. Yeo, M. Yeung, Watermarking 3D objects for verification, IEEE Computer Graphics and Applications , Volume: 19 Issue: 1 , Jan.-Feb., pp. 36 –45, 1999.

[26] R. Ohbuchi, H. Masuda and M. Aono. A shape-preserving data embedding algorithm for NURBS curves and surfaces, Computer Graphics International Proceedings , pp. 180 –187, 1999.

[27] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser and S. Rusinkiewicz. A Reflective Symmetry Descriptor for 3D Models, to appear in Algorithmica, 2003. http://www.cs.princeton.edu/~funk/algorithmica02.pdf

[28] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama. Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain, in the proceedings of the Graphics Interface 2001, pp. 9-17, Ontario, Canada, June, 2001.

[29] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D Models with Shape Distributions, Shape Modeling International , Genova, Italy, May, 2001.

[30] D. Zorin , P. Schröder and Wim Sweldens. Interpolating Subdivision for Meshes with Arbitrary Topology, SIGGRAPH 96 Proceedings, pp. 189-192, August, 1996.