XI International SAUM Conference
on Systems, Automatic Control and Measurements
Niš, Serbia, November 14th-16th, 2012

Effects of 3-D Surface Smoothing to Watermark Data Stability

B. Vasic +

+ Department of Electronics, University of Nis, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Nis, Republic of Serbia
Phone: (+38163) 417-696, E-Mail: bata.vasic@elfak.ni.ac.rs, WWW: http://www.batavasic.com

Abstract *- This paper deals with a conditionality of watermark data stability from smoothing process of a three-dimensional (3D) surface. The topological stability of watermark data depends on a resistance of data carriers to destructive processes. Mesh smoothing is a common software tool for a 3D surface fairing, but it also drastically changes geometric structure of mesh surface. The changes are reflected in an insertion of new vertices and a non-uniform translation of existing vertices. We show that insertion and translation of geometric primitives affect positions of original vertices in Euclidian space, thus data carriers are not stabile. However, vertices are retained within initial regions and the most of their topological features is still computable. Our algorithm selects the most important vertices in topological sense, thus it results in well extraction of watermark data.*

Key words: *3-D mesh, 3D geometry, surface smoothing, watermark stability*

I. INTRODUCTION

Data protection of multimedia contents is a research field that is permanently progressing in last two decade. High speed of the internet network and mass use of such content increase a risk of an unauthorized use and misuse. Firstly, protection has been needed for digital pictures, documents and digital audio files. Early developed good compressions for these types of digital data has reduced the amount of data, thus they have increased a transmission speed over the internet network. One of the most popular ways of protection is a watermarking, i.e. protective data hiding [3]. Watermarking is used to protect all kind of data, whereas its basic characteristics have trodden by researchers in the fields of digital image and audio processing.

Other multimedia contents such as video files and three-dimensional (3D) models, in comparison to digital pictures and digital audio, are bigger and their internet transmissions are slower. Therefore, a copyright protection of video materials and movies has been performed firstly by disabling DVD copying [4], and watermark protection has not been necessary. The need for internet distribution of 3D models grows with a development of video and computer games, which use three-dimensional meshes. Computer games in last decade become demanding regarding to a complexity of 3D models, thus the real expansion of their use is conditioned by increasing of the internet network speed. Since recently, 3D objects are widely used in medical simulations and architectural visualizations. This widespread use needs good protection. However, existing watermarking techniques are not completely usable for the watermark protection of 3D meshes. Specificity of these data format requires new adaptable watermark techniques.

Additional conditions for the spread use of 3D models are met by development of software tools and their relatively low cost. On the other hand, a "perfection" of 3D software allows easy work with numerous processes for mesh manipulation, which may cause a destroying of watermark data. Mesh smoothing [7] is a very useful process for a 3D surface fairing, but it can be easy used for a malicious watermark removal. Even in a case of the non-malicious use, smoothing process changes topology of the 3D surface that causes watermark damage.

Considering various problems in processes of watermark embedding and extraction [9] this paper presents an influence of the 3D surface smoothing on watermark data stability. Using our software tool for a geometric analysis of 3D meshes we show that insertion and translation of geometric primitives affect positions of existing vertices in Euclidian space. However, basic discrete geometric features such as local Gaussian curvature and mean curvature are not changed significantly. Our algorithm for a stabile vertex extraction selects most important and stabile primitives in topological sense, thus it results in well extraction of watermark data.

This paper is organized as follows. Section II gives background of related research and some notions of watermark process. In this section we also explain the mesh smoothing as a very useful non-linear processes for 3D mesh manipulation and 3D surface fairing. Section III presented watermarking method using our stabile vertex extraction algorithm. Next section shows experimental and numerical results of a smoothing process influence on watermark data stability. Finally, we discuss results and shortcomings in Conclusions sections, where we give briefly some indications of the future work.

II. BACKGROUND AND RELATED WORKS

Hiding data is very old research discipline and first story with an example of a hiding technique and such data transmission can be traced back to 440 B.C. in ancient Greece and mentioned by Herodotus [1]. Histiaeus tattooed a warning message about Persian invasion plans on the shaved head of slave. The message was transmitted to Greece, hidden by the hair that afterwards grew over it and exposed by shaving the head again. First notions and terminology of data hiding and cryptography originate also from the ancient Greeks.

A. Steganography and digital watermarking

Steganography is the art and science of hiding information by embedding messages within other, seemingly harmless message [2]. Steganography means “covered writing” in Greek and this term was used originally as a synonym for *data hiding*. Difference between steganography and cryptography meanings is reflected in fact that cryptography hides content of message, whereas steganography hides a presence of message within another message called *cover message*. In this context we underline following terms that we use along the paper [3]:

1. *Cover-object*, c: the original object (host object) where the message has to be embedded. Cover-text, cover-image, cover-3D mesh.
2. *Stego-message*, m: the message that has to be embedded in the cover-object. In the watermarking context it called *mark* or *watermark*.
3. *Stego-object*, s: The cover-object that contents the embedded stego-message.
4. *Stego-key*, k: The secret shared between an author and user to embed and retrieve the message.

Embedding and extraction processes are defined respectively by the embedding function E that maps the tripled cover-object c, message m and stego-key k to a stego-object s, and the retrieving function D that is a mapping from s to m using the stego-key k.

$$E(c,m,k)=s,\ \ D(s,k)=m \tag{1}$$

Non-blind stegosystems needs the original cover-object c as input for the function D, but in that case it can be assumed that $k = c||k'$ where k' is the secret key.

In comparison to the data hiding and undetectable data hiding (modern steganography meaning), where only presence of message is important, digital watermarking is a process of embedding the important stego-message into important cover-object. The main watermarking goal is to avoid the remove of watermark message from stego-object without stego-key. Although in some applications the message is not hidden, watermarking of 3D models usually involves blind watermarks to provide a proof of an authorship or ownership in a case of unauthorized use.

B. Watermark robustness

Requirements for watermark data embedding is unique in each application, however, some of them are common for several applications. One of the most important requirement is a robustness i.e. resistance of watermark data retrieving to modifications and manipulation of watermarked object, which is readily modified using widely available software packages. Modifications change the stego-object, but also may damage both cover-object and stego-message.

Swanson, Kobayashi and Tewfik have noted in [4] several common types of malicious and incidental modifications of digital pictures digital audio and video. These modification include additive Gaussian noise; linear filtering such as low-pass and high-pass filtering; compression such as Joint Photographic Experts Group (JPEG), Moving Picture Experts Group (MPEG) and wavelet; quantization; rotation; scaling; removal or insertion of pixels or video frames etc.

The geometry of 3D model is very sensitive even to linear modifications such as translation, rotation and uniform scaling. An idea for an increasing the watermark robustness of 3D meshes to affine transformations has been explained in [5]. The optimization and smoothing of 3D geometric models are non-linear and they can be compared to a JPEG, MPEG and wavelets transformations. The effects of 3D topological features to the mesh optimization process we have discussed in [6].

C. Mesh smoothing

Mesh smoothing is also non-linear process, which is common used for noise filtering and 3D modeling. Several conventional mesh smoothing methods such as the Laplacian and bilaplacian smoothing methods, mean curvature flow, and the Taubin λ|μ scheme have been given by Belyaev and Ohtake in [7]. For one iteration of the iterative mean mesh-filtering scheme, they have noted two basic steps: Averring normals, and Fitting mesh to modified normals.

1. Consider an oriented triangle mesh. For each triangle T with its area $A(S)$ and the set $\mathcal{N}(T)$ of all triangles that have a common edge or vertex with T, we compute the triangle normal $\boldsymbol{n}(T)$ and apply the following weighted averaging procedure to the field of normals:
$$\boldsymbol{m}(T)=\sum_{S\in\mathcal{N}(T)} A(S)\boldsymbol{n}(S). \tag{2}$$
2. Modifying mesh vertex position in order to fit the mesh to the set of modified normals $\boldsymbol{m}(T)$, for each mesh vertex P, we introduce an error function that measures how good a modified mesh fits a field of modified normals:
$$E_{fit}(P)=\sum A(T)\,|\,\boldsymbol{n}(T)-\boldsymbol{m}(T)\,|^2, \tag{3}$$
where the sum is taken over all triangles adjacent to P. Minimizing the E_{fit}, position of new vertices are found:
$$P_{new}=\arg\min_{P} E_{fit}(P). \tag{4}$$

Result of MeshSmooth modifier [8] to geo sphere mesh model is shown in the next figure.

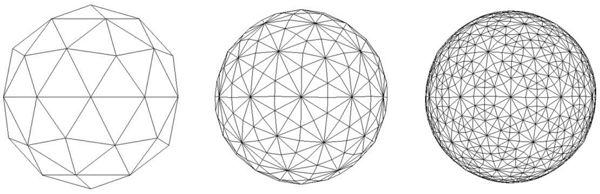

(a) (b) (c)

Fig. 1. Effect of the Mesh smoothing on discrete spherical surface: original object (a) and smoothed objects using iterations: (b) 1 and (c) 2.

III. 3D MESH WATERMARKING

We have already mentioned differences between watermarking 3D models and two-dimensional (2D) objects, but non-unique representation of 3D meshes introduces a new problem with a capacity of data carrier [9], because system should allow embedding of nontrivial amounts of watermark data. Using some coding scheme to improve watermark robustness we have to embed certain number of redundant bits, thus they not leave to much space for watermark embedding. Benedens in [10] has discussed about the capacity, but he first introduces a new threat to robustness, called multiple watermarking. In a topological sense, an embedding a new watermark over the old one changes geometric structure of 3D model, thus old watermark is also damaged or even destroyed.

Taking all in account we suggest new watermarking method that include two parallel directions: first, dealing with important geometric features to stabile vertex selection [11] and second, using error correction codes for watermark message encoding and decoding [12]. Next block diagram represents flowchart of watermarking process that we suggest.

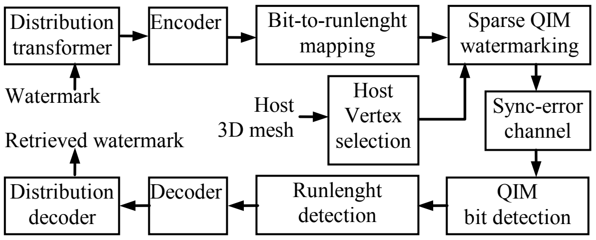


Fig. 2. A block diagram of the proposed coded watermarking system

A. Host vertex selection

Our method uses space based watermarking method, which embeds watermark bits in the geometric structure of 3D mesh by changing coordinates of vertices in the three-dimensional space. Hence, a good selection of host vertices is the first challenge.

We have to keep in mind that many modifications, used for improving geometric or rendering characteristics of 3D model, remove vertices or change their space coordinates. Therefore, in unpublished paper [11] we have analyzed a set of geometric features for assessment of vertices by invariance to the optimization. Thus, we show that well curvature estimation is important for a vertex stability determination. Considering both methods for local curvature evaluation, the fitting quadrics method [13] and the differential geometry method [11], we define several criteria for a geometric importance assessment. However, some of criteria are used in assessment as important features of optimization [14],[15] and simplification [16] process.

Result of our algorithm [17] is two vectors: the vector of vertex stabilities arranged in a decreasing order, and the vector of corresponding indices. Hence, host vertices are selected and ordered with respect of decreasing stability.

B. Runlenght mapping

Difficulty of code construction for our watermarking system is the fact that channels with synchronization errors have infinite memory. In order to transform the channel with infinite memory into a memoryless channel, the bits (symbols) at the input of the channel are represented by runs of bits. To explain the main idea, let us consider a case in which binary zeros are represented by runs of length *two*, and binary ones with runs of length *three*. As an illustration, consider the sequence of information bits b = (0, 1, 1, 0, 1, 1). The encoded sequence c is initialized to an empty string. The encoding proceeds as follows.

The first bit b_1 is 0. To encode b_1, a run consisting of *two* bits, namely c_1 through c_2, is added to c. This can be done by setting $c_1 = c_2 = 1 \bmod 2 = 1$ (c = (11)). Next, since b_2 is 1, we construct a run consisting of *three* bits, namely, c_3 through c_5, is added to c. This can be done by setting $c_4 = c_5 = c_6 = 2 \bmod 2 = 0$ (c = [11 000]). Proceeding thus, we find that b is encoded as encoded bits c=(11 000 111 00 111 000). In other words, input symbols with odd indices are encoded in runs of k 1's, with k = 2 for symbol 0 and k = 3 for symbol 1. Runs of 0's similarly encode symbols with even indices.

C. Sparse Quantized Index Modulation (QIM)

Let $\mathbf{u} \in \{0,1\}^n$ and $\mathbf{x} \in \mathbf{R}^n$ be the watermark sequence and the cover sequence, respectively. The *embedder* combines the n-dimensional vectors $\mathbf{u}$ and $\mathbf{x}$ and produces the watermarked sequence $\mathbf{y} \in \mathbf{R}^n$. The difference $\mathbf{w}=\mathbf{y}-\mathbf{x}$ is referred to as the *watermarking displacement* signal. The distortion is typically defined as the simple Euclidian distance. The QIM operates on independently on the elements u and x of the vectors $\mathbf{u}$ and $\mathbf{x}$. To embed the bit $u \in \{0,1\}$, the QIM requires two uniform quantizers $\mathbf{Q}_0$ and $\mathbf{Q}_1$ defined as the mappings

$$Q_u(x) = \Delta\left[\frac{1}{\Delta}\left(x-(-1)^u\frac{\Delta}{4}\right)\right]+(-1)^u\frac{\Delta}{4} \qquad (5)$$

where [ ] denotes the rounding operation, i.e. for a real x, $[x]$ is the integer closest to x. Thus, the quantization level of the "nominal" quantizer $\Delta\,[x/\Delta]$ is moved up or down by $\Delta/4$ depending on the value of u. The watermark bit u dithers the input x by the amount $\pm\Delta/4$. The watermark bit u determines the selection of a quantizer, so that $y=\mathbf{Q}_u(x)$.

Sparse QIM, spreads out the watermark bit over L elements of cover signal $\mathbf{x}$. The cover sequence $\mathbf{x}_L$ of length is projected to a L-dimensional vector $\mathbf{p}$ of the unit norm, and the norm of the corresponding projection is quantized. The resulting watermarked vector $\mathbf{y}_L$ can be written as

$$\mathbf{y}_L = \mathbf{x}_L + \left(Q_u(\mathbf{x}_L^T\mathbf{p}) - \mathbf{x}_L^T\mathbf{p}\right)\mathbf{p} \qquad (6)$$

The detector projects the received watermarked cover vector $\mathbf{r}_L$ to $\mathbf{p}$ and recovers the embedded bit as

$$\hat{u} = \arg\min_{u\in\{0,1\}} \left\|\mathbf{r}_L^T\mathbf{p}_L - Q_u(\mathbf{r}_L^T\mathbf{p}_L)\right\|_2 . \qquad (7)$$

IV. NUMERICAL RESULTS

All calculations are performed using our OSVETA software [17] and Naissa by Bata 3D model [18]. For faster computation the model has optimized by a Face threshold value 10, thus a total number of faces has decreased from 66922 to 11732. Then, the simplified mesh has subjected to the MeshSmooth modifier using I=1 and I=2 iterations. Finally, from both smoothed meshes we have extracted vectors of 1000 vertices that are used as host vertices for watermark embedding (se Fig. 3.).

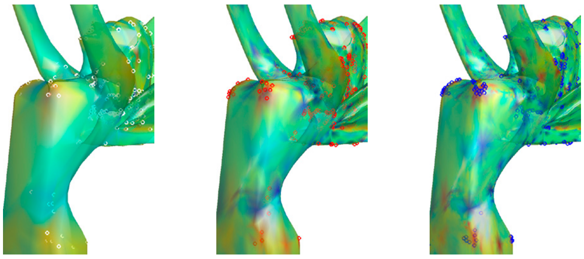
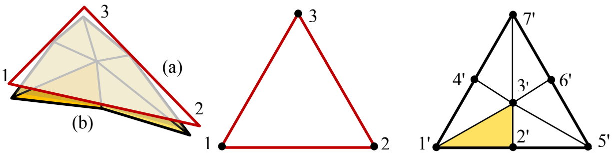

(a) (b) (c)

Fig. 3. Host vertices selection: original object (a), smoothed objects using iterations: (b) 1 and (c) 2.

One can observe even a perceptually that Euclidian positions of existing vertices are changed, but new faces are created without removing existing vertices (See Fig. 4).



(a) (b)

Fig. 4. Triangular face creation: (a) original mesh, (b) smoothed mesh.

Since new vertex indices that correspond to indices of original vertices are known (1→1', 2→5', 3→7', etc.), a watermark extraction from host vertices is possible. The following table presents a number of host vertices out of 5870, which is selected using our extraction algorithm in relation to number of random selected host vertices.

TABLE I

The number of selected host vertices for 0, 1, and 2 smoothing iterations, using random extraction and OSVETA vertex extraction algorithms.

3D model	I=0	I=1	I=2
Total number of vertices	5870	35200	140788
Random	5870	968	1254
OSVETA	5870	1316	1423

The error probability of the both vertex selections is shown in Fig. 5.

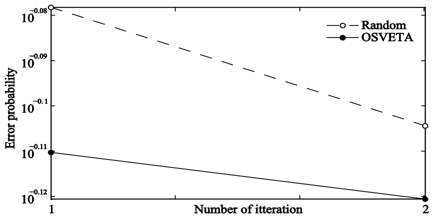


Fig. 5. Error probability of random and OSVETA vertex selection methods for the smoothed mesh using one and two iterations.

V. CONCLUSIONS AND FUTURE WORKS

In this paper we have explained principles of the 3D mesh watermarking, and resistance of embedded data to smoothing process. One can conclude that the low robustness of watermark data in relation to the smoothing process is consequence of drastically changed topological features of 3D geometry.

For better watermark retrieving, we have used our vertex extraction algorithm. Better results have achieved, and thus watermark robustness has increased, but shortcoming of the applied method is slower watermark extraction. However, a large coefficient of face multiplication leads to even slower manipulation of the 3D model. This refuses potential misusers from a mesh densification.

REFERENCES

[1] Herodotus, *The Histories (trans. A. de Selincourt)*. Middlesex, England: Penguin, 1972.

[2] C. A. Henk, *Encyclopedia of cryptography and security, Van Tilborg ed.*, p.159, Springer, 2005.

[3] B. Pfitzmann, "Information Hiding Terminology", in *Proc. Int. Workshop on Information Hiding*, Cambridge, May/June 1996, vol. 1174, pp. 347-350.

[4] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies," in *Proc. the IEEE*, June 1998, vol. 86, no. 6, pp 1064-1087.

[5] B. Vasic, "Increasing of 3D mesh watermark robustness to affine transformations using barycenter as spherical coordinate origin," *In Proc. ETRAN 2012,* Zlatibor, June 2012.

[6] B. Vasic and V. Pavlovic, "Effects of the 3D model geometric region curvature to a topological structure simplification," in *Proc. TELFOR 2011,* Belgrade, November 2011, no. 19, pp. 1135-1138.

[7] A. Belyaev and Y Ohtake, "A comparison of mesh smoothing methods," in *Proc. Conf. on Geometric Modeling and Computer Graphics IKBN* 2003, Tel-Aviv, February 2003, pp. 83–87.

[8] T. Daniele, *Poly-Modeling with 3ds Max - Think outside of the box, TurboSmooth and MeshSmooth*, p. 61 – 62, Elsevier, 2009.

[9] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models, " in *Proc. ACM Intern. Conf. on Multimedia '97*, Seattle, November 1997, pp. 261–272.

[10] O. Benedens, "Geometry-Based Watermarking of 3D Models", in *Proc. IEEE CG&A*, January/February 1999, vol. 19, no. 1, pp. 46-55.

[11] B. Vasic, "Ordered Statistics Vertex Extraction and Tracing Algorithm (OSVETA)," unpublished, submitted to *the Advances in Electrical and Computer Engineering (AECE),* April 2012, [Online]. Available: http://www.batavasic.com/ research/OSVETA.pdf

[12] B. Vasic and B. Vasic, "Simplification Resilient LDPC-Coded Sparse-QIM Watermarking for 3D-Meshes," unpublished, submitted to *the IEEE Transactions on Multimedia,* [Online]. Available: http://arxiv.org/abs/1204.2214

[13] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy and M. Desbrun, "Anisotropic Polygonal Remeshing, " *ACM Transactions on Graphics,* in *Proc. SIGGRAPH '2003*, San Diego, July 2003, vol. 22, no 3, pp. 485 – 493.

[14] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh Optimization," in *Proc. SIGGRAPH 93,* Anaheim, August 1993, ACM SIGGRAPH, pp. 19–26.

[15] H. Hoppe, "Progressive Meshes," in *Proc. SIGGRAPH 96,* New Orleans, August 1996, ACM SIGGRAPH, pp. 99 - 108.

[16] M. Garland and P.S. Heckbert, "Surface Simplification using Quadric Error Metrics," in *Proc. SIGGRAPH 97*, Los Angeles, August 1997, ACM SIGGRAPH, pp. 209–216.

[17] B. Vasic, Ordered Statistic Vertex Extraction and Tracing Algorithm (OSVETA) - MatLab Software, [Online], Available: http://www.batavasic.com/research/OSVETA.zip.

[18] Serbian Film Center, Naissa Trophy [Online], Available: http://www.batavasic.com/research/Naissa_by_Bata.zip.