A Blind Roubst Watermarking Scheme based on Progresive Mesh and Self Organization Maps

Mona M. Soliman, Aboul Ella Hassanien and Hoda M. Onsi

Cairo University, Faculty of Computers and Information, Cairo, Egypt
Scientific Research Group in Egypt (SRGE)
http://www.egyptscience.net

Abstract. Most of progressive mesh (PM) transmission techniques, consist in iteratively decimating the mesh, while storing the information necessary to the process inversion. During the transmission or visualization the 3D content can be duplicated and redistributed by a pirate. Digital watermarking is considered as a good solution to this emerging problem. This paper focus on introducing a novel robust and blind mesh watermarking schema by converting the original triangle mesh into a multi-resolution format, consisting of a coarse base mesh and a sequence of refinement, the watermark bits are inserted through progressive mesh level of details, and extracted at refinement stage without any need for the original model. The watermark insertion is performed only on set of marked vertices come out from Self Organization Maps (SOM) clustering neural network. These vertices are used as candidates for watermark carriers that will hold watermark bits through progressive mesh transmission. The robustness of proposed techniques is evaluated experimentally by simulating attacks such as mesh smoothing, noise addition and mesh cropping.

1 Introduction

The 3D geometric model is one of the most fundamental techniques in the fields of computer-aided design (CAD), computer-assisted manufacturing (CAM), computer-aided engineering (CAE), and computer graphic (CG) animation [1]. The current situation of the distributed engineering environment increases opportunities to globally exchange digital data of geometric models between various organizations through a computer network. This also increases the possibility of theft of the geometric model by illegal duplication. Therefore the copyright property of the geometric model must be strongly protected from theft [2].

Since 3-D mesh watermarking techniques were introduced, there have been several attempts to improve the performance in terms of transparency and robustness. Robustness is achieved when the watermark can be retrieved even after the watermarked model has been processed or attacked intentionally specific algorithms. Depending on whether the original cover-media is needed or not in the detection stage we have non-blind and blind watermarking. The methods from the first category usually have good robustness, but they are not suitable for most applications [5].

There are two kinds of 3D mesh watermarking algorithms that are similar to image watermarking. One is based on the spatial domain [3]- [6], which provides many insights into mesh watermarking. [7]- [10] suggest several spatial domain based algorithms in which the blindness has been achieved. These schemes has a certain level of robustness against common geometry attacks and even cropping, but are in general fragile to connectivity attacks because the used geometric watermarking primitives may disappear or be seriously disturbed after such attacks. The other algorithms are based on the transformation domain [11]- [17]. In these algorithms, spectral decomposition and multi resolution techniques such as wavelet transform and progressive meshes are used to decompose a 3D model into a lower resolution and the watermark is inserted in the bit stream. The model is then reconstructed from lower resolutions.

Progressive meshes is one of the multi resolution techniques. This technique was introduced by Hoppe in 1996 [19]. Progressive mesh introduced as a new format for storing and transmitting arbitrary triangle meshes. For a given mesh, the PM representation defines a continuous sequence of level-of-detail (LoD) approximations, allows smooth visual transitions (geomorphs) between these approximations, supports progressive transmission, and makes an effective compression scheme. In short, progressive meshes offer an efficient, lossless, continuous-resolution representation.

progressive compression [21] allows to achieve high compression ratio (and thus fast transmission) and also to produce different levels of detail (LoD), allowing to adapt the complexity of the data to the remote device by stopping the transmission when a sufficient LoD is reached. Most of progressive compression techniques, consist in iteratively decimating the mesh (vertex/edge suppressions), while storing the information necessary to the process inversion, i.e. the refinement.

The problem facing the progressive mesh compression that the transmitted model can be duplicated and redistributed by a pirate, So there is a need to protect such transmission from attacks and protect it from illegal duplication

This paper focus on introducing a robust and blind mesh watermarking schema by converting the original triangle mesh into a multi-resolution format, consisting of a coarse base mesh and a sequence of refinement, the watermark bits are inserted through progressive mesh level of details, and extracted at refinement stage without any need for the original model. Watermark insertion is performed on specific set of vertices that are selected by utilizing Self Organization Maps (SOM) [20]. SOM is a kind of competitive neural network in which the networks learn to form their own classifications of the training data without external help. Watermark bits sequence are inserted based on the mean and stander deviation of selected vertices' neighbours in the original mesh.

The remainder of this paper is organized as follows. Section (2) reviews related work of watermark 3D model within progressive mesh. Section (3) illustrate some useful and important preliminary ideas relating to this work. Section (4) discusses the proposed watermarking scheme using both SOM and PM . Section (5) shows the experimental results. Conclusions are discussed in Section (6).

2 Related work

The concept of progressive mesh compression was introduced for the first time by Hoppe [19]; by generating lower resolution models of the mesh by performing a sequence of vertex split operations. A set of methods were then introduced, [22]-[23], to improve the compression ratio by applying the collapse/split operations on sets of independent vertices. Other methods are based on vertex removal, they consist in removing sequences of vertices and re-triangulating the holes left by the deletions at no cost [21]. In [24] Peng aims at carefully preserving the relevant features at each intermediate level of detail using a hierarchical clustering. [25] proposed a new progressive approach based on a reconstruction scheme, the algorithm starts from a coarse version of the original model which is refined progressively by inserting a vertex to the longest edge using edge split operation, aiming to generate uniformly sampled intermediate meshes.

Having improvement in the techniques of the progressive compression lead into the question how to protect this content during its transmission. Many attemps try to answer this question by utilizing 3D watermark procedures. Praun and Hoppe [12] reported robust mesh-watermarking algorithm that works using Hoppe's progressive meshes. Such algorithm provide a scheme for constructing a set of scalar basis functions over the mesh vertices. Then they adapt the spread-spectrum principles used in image watermarking to embed information into the basis functions corresponding to perceptually significant features of the model. Chen and Chen [26] proposed 3D mesh watermarking approach on the basis of the progressive mesh and the discrete wavelet transform. By embedding the transformed sequence of the watermark image into the vertex split sequence of the progressive mesh, the watermark and the cover object are transmitted synchronously such that the progressive decode of the cover object and the extraction and decoding of the watermark can be performed on-the-fly with the transmission. Lee [21] presented a joint reversible watermarking and progressive compression of 3D meshes. Each LoD is compressed and watermarked by modifying the geometry of refined vertices with respect to the center of mass of the original 3D mesh. The watermark process is reversible in the sense that the geometrical modifications introduced by the embedding processing can be removed after watermark extraction.

3 Preliminaries

3.1 Self organizing maps

Self-Organizing means no supervision is required. SOMs learn on their own through unsupervised competitive learning. Maps means they attempt to map their weights to conform to the given input data [27]. In competitive learning, the output neurons compete amongst themselves to be activated, with the result that only one is activated at any one time. This activated neuron is called a winner-takes-all neuron or simply the winning neuron. Such competition can be induced/implemented by having lateral inhibition connections (negative feedback paths) between the neurons. The result is that the neurons are forced to organise themselves [28], the n-dimensional input is processed by exactly the same number of computing units as there are clusters to be individually identified as shown in Figure 1.In this work we utilize the particular kind of SOM known as a Kohonen Network. This SOM has a feed-forward structure with a single computational layer arranged in rows and columns. Each neuron is fully connected to all the source nodes in the input layer. Clearly, a one dimensional map will just have a single row (or a single column) in the computational layer.

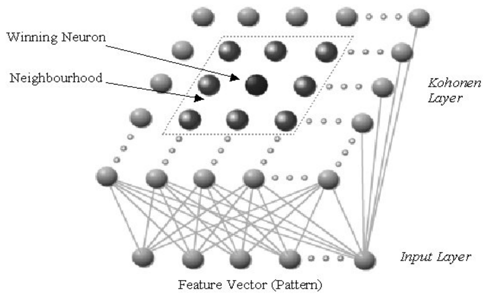

Fig. 1. Structure of SOM Kohonen network

3.2 Progressive mesh

In the progressive mesh (PM) representation as introduced by Hoppe [19], an arbitrary mesh $\hat{M}$ is stored as a much coarser mesh M^0 together with a sequence of n detail records that indicate how to incrementally refine M^0 exactly back into the original mesh. Each of these records stores the information associated with a vertex split, an elementary mesh transformation that adds an additional vertex to the mesh. The PM representation of $\hat{M}$ thus defines a continuous sequence of meshes M^0,M^1,. . . ,M^n of increasing accuracy, from which LOD approximations of any desired complexity can be efficiently retrieved. edge collapse, is an operation that is sufficient for effectively simplifying meshes. As shown in Figure 2, an edge collapse transformation unifies 2 adjacent vertices v_1 and v_2 into a single vertex v_1. The vertex v_2 and the two adjacent faces f_1 and f_2 vanish in the process. A position v_1 is specified for the new unified vertex. A key observation is that an edge collapse transformation is invertible. [19] call that inverse transformation a vertex split. A vertex split transformation adds near vertex v_1 a new vertex v_2 , and two new faces f_1 and f_2

Because edge collapse transformations are invertible, it can represent an arbitrary triangle mesh $\hat{M}$ as a simple mesh M^0 together with a sequence of n v_{split} records. Hoppe call (M^0 ,(v_{split_0},v_{split_1},. . . ,$v_{split_{n-1}}$)) a progressive mesh (PM) representation of $\hat{M}$.

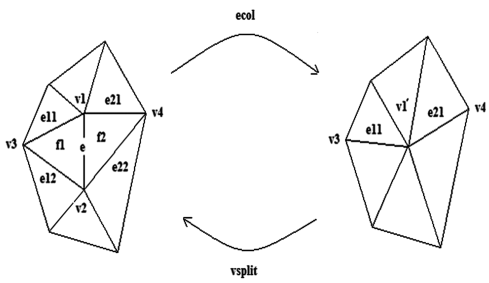

Fig. 2. Edge collapse operation

4 The Proposed 3D watermarking scheme

Most of progressive compression techniques, consist in iteratively decimating the mesh, while storing the information necessary to the process inversion, i.e. the refinement [21]. During the transmission or visualization the 3D content can be duplicated and redistributed by a pirate. Digital watermarking is considered as a good solution to this emerging problem. The main problem is how to combine compression and watermark sequence. A simple solution consists in inserting the watermark bit sequence during the decimation process. since the mesh is transmitted in a progressive way until a certain level of detail (not necessary the finest one), the watermark have to be readable in all intermediate LoDs. The watermark insertion is performed only on set of marked vertices come out from SOM neural network as candidates for watermark carriers. Watermark bit sequence are inserted based on the mean and stander deviation of marked vertices' neighbours in the original mesh. The Extraction is performed during the refinement sequence without any need to the original model. Figure 3 show the whole watermarking scheme for both insertion and extraction. The following subsection illustrate in more details the basic steps necessary of performing watermark insertion and extraction.

4.1 Vertex Selection Using SOM

The first step in our proposed scheme is to select set of vertices and mark it as watermark carrier. We aim to make this selection in an intelligent way that guarantee preserving model from distortion during watermark insertion process. Based on this requirement we utilize the SOM neural network in clustering the vertices of original mesh into suitable and non-suitable candidates for watermark carriers. This can be performed as proposed in our previous work [20] by clustering vertices based on smoothness feature. The smoothness feature measure the angle variation between surface normals and the average normal corresponding to a vertex. The feature vector extraction method results in a set of angles derived by computing the orientation of the surface normals to the average normal of the triangular faces that form a 1-ring neighbourhood for a vertex. Such a

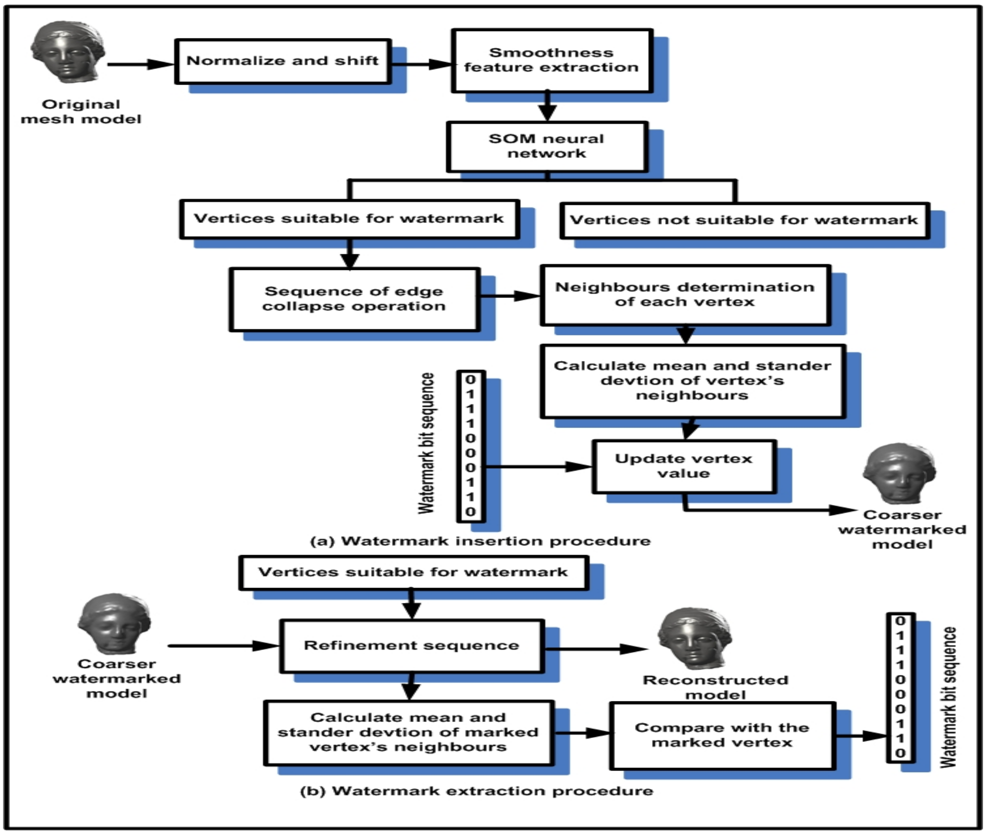

Fig. 3. The proposed watermarking scheme

smoothness measure reflect the local geometry of a surface or region. Flat and peak surface are not suitable for inserting watermark bits sequence.

4.2 Watermark Insertion Procedure

The watermark insertion is performed with each edge collapse iteration. Since each edge collapse result in one vertex removal and two faces removal, the collapse iteration is performed around marked vertices resulting from SOM clustering. The selected edge for collapse contain a vertex come out from the SOM clustering and this vertex is used to carry the watermark bit, The other vertex is eliminated. The insertion of watermark bits sequence w is performed by searching for the neighbours of marked vertices ,estimating their mean and stander deviation and insert watermark using these two values. figure shown an example of one edge collapse iteration around marked vertex. Basic steps of watermark insertion procedure is illustrated in algorithm 1

4.3 Watermark Extraction method and 3D Mesh Reconstruction

Here we utilize a blind watermarking procedure thus there is no need to the original model during the extraction process. To extract the watermark, we need the position of marked vertices as watermark carrier used in the insertion procedure. Therefore, we need to store these data for the correct watermark extraction. Once these locations are detected watermark bits w are extracted by comparing the value of marked vertices with their immediate neighbours' mean and stander deviation.This comparison is performed by a shifted value α to take consideration of high correlation in case attacks is happen. During the Extraction procedure the current intermediate mesh is refined with insertion of a set of vertices to formulate the collapsed edge around marked vertex carried the watermark bit. Algorithm 2 illustrate in more details the basic steps of watermark extraction procedure.

Algorithm 1 The watermark insertion procedure

Input: list of vertex positions, list of faces, marked vertex as watermark carriers, watermark bit sequence w

for Each iteration of edge collpase **do**

Call the marked vertex as watermark carrier.

Create a list of marked vertex's immediate neighbours $V\{NH\}$.

Determine the edges conecting marked vertex and its neighbors.

Pick the first neighbour $V\{NH_1\}$

⇒ Eliminate $V\{NH_1\}$ from the neighbours list.

⇒ Eliminate $V\{NH_1\}$ from the vertex mesh model.

⇒ Eliminate two faces connecting $V\{NH_1\}$ with marked vertex from the face list.

Calculate mean and stander devition of marked vertex neighbours' list.

Update the value of marked vertex according to the following rule.

if watermark bit w_i=1 **then**

$$\begin{aligned} V_x^{'} &= \mu(V\{NH_x\}) + 2 * \sigma(V\{NH_x\}) \\ V_y^{'} &= \mu(V\{NH_y\}) + 2 * \sigma(V\{NH_y\}) \\ V_z^{'} &= \mu(V\{NH_z\}) + 2 * \sigma(V\{NH_z\}) \end{aligned} \tag{1}$$

end if

if watermark bit wi==0 **then**

$$\begin{aligned} V_x^{'} &= \mu(V\{NH_x\}) - 2 * \sigma(V\{NH_x\}) \\ V_y^{'} &= \mu(V\{NH_y\}) - 2 * \sigma(V\{NH_y\}) \\ V_z^{'} &= \mu(V\{NH_z\}) - 2 * \sigma(V\{NH_z\}) \end{aligned} \tag{2}$$

end if

end for

Output: sequence of coarser watermarked mesh model

Algorithm 2 The watermark insertion procedure

Input: sequence of coarser watermarked mesh model,marked vertex as watermark carriers.

for Each iteration of model refinment. **do**

Call the marked vertex as watermark carrier.

Create a list of marked vertex's immediate neighbours $V\{NH\}$

⇒ Predict the vertex $V\{NH_1\}$ that eliminated during edge collapse iteration according to its neighbour.

⇒ Update coarser mesh model with the predicted vertex.

⇒ Update the face list of the coarser model.

Calculate mean and stander devition of marked vertex neighbours' list.

Compare the marked vertex with calculted mean and stander devition.

$$\begin{array}{ll} \mu_1 x = \mu(V\{NH_x\}) + 2 * \sigma(V\{NH_x\}) & \mu_2 x = \mu(V\{NH_x\}) - 2 * \sigma(V\{NH_x\}) \\ \mu_1 y = \mu(V\{NH_y\}) + 2 * \sigma(V\{NH_y\}) & \mu_2 y = \mu(V\{NH_y\}) - 2 * \sigma(V\{NH_y\}) \\ \mu_1 z = \mu(V\{NH_z\}) + 2 * \sigma(V\{NH_z\}) & \mu_2 z = \mu(V\{NH_z\}) - 2 * \sigma(V\{NH_z\}) \end{array} \tag{3}$$

if $(V_x^{'} \geq \mu_1 x\text{-}\alpha)\&(V_x^{'} \leq \mu_x\text{+}\alpha)$ **then**

$$w_i = 1; \tag{4}$$

end if

if $(V_x^{'} \geq \mu_2 x\text{-}\alpha)\&(V_x^{'} \leq \mu_x\text{+}\alpha)$ **then**

$$w_i = 0; \tag{5}$$

end if

end for

Output:Reconstructed model, watermark bit sequence w

5 Experimental results and discussion

The proposed algorithm is developed in $MATLAB$7.6 environment. Several models are used for the evaluation and three of them are shown in Figure 4 : bunny (34,835 vertices;69,666 faces), dragon (50,000 vertices; 100,000 faces),venus (100,579 vertices;201,514 faces), The watermark bits are generated randomly by a length of 50 bits. This work is compared the algorithm of Cho et al. [29] that considered one of the best 3D watermarking algorithm in terms of robustness and blindness. Cho proposed n watermark insertion in the vertex norm distribution.

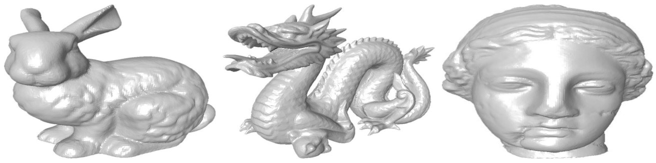

Fig. 4. Original 3D mesh models

5.1 Distortion Evaluation

Once edge collapse is performed the mesh model is stored as coarser base mesh and a sequence of refinements. Reconstruction of the original mesh from the coarser base mesh introduce a smoothed model close to the original one but not identical to it. Since the watermark is inserted during the edge collapse iterations the reconstructed model is degraded due to the presence of watermark bits sequence

Vertex Signal-to-Noise Ratio(VSNR) quantify the visual differences between the original and watermarked models. In this work it also measure the distortion between the original model and the reconstructed model holding watermark bit sequence. Table 1 shown the VSNR values for the tested models using different refinement iterations. Obliviously as number of refinement iterations increased, more smoothed model is generated,and VSNR is decreased but as shown in the table this is performed by very small rate.

Table 1. VPSNR values at different refinement levels M

Model	$(M = 50)$	$(M = 80)$	$(M = 120)$
Bunny	104.40	104.19	103.98
Dragon	113.95	113.17	112.19
Venus	137.16	136.24	135.14

5.2 Robustness Evaluation

To investigate the robustness of watermark scheme, each watermarked model is attacked by simulating set of geometric attacks such as :

• Additive binary noise : random noise is added to the watermarked model with different error rates (e.g. Amp= 0.005, 0.009, 0.01, 0.03, 0.05). Here, the error rate represents the noise amplitude as a fraction of the maximum vertex norm of the object.

• Smoothing attack: Laplacian smoothing is applied to the watermarked model smooths the sharp edges in the model by applying a low pass gradient filter to the vertices. (e.g. setting λ=0.03 with different iterations N_{it}=5, 10, 15,20)

• Cropping attack : cropping is performed by cutting 3D model with different levels indicating the percentage of cropped vertices (e.g. 3%,5%,10%,20%).

We give here the robustness evaluation against only common geometric attacks. As our algorithm is based on a connectivity-based compression technique, the refinement is not possible when a change of connectivity occurs. Hence our algorithm is not robust to connectivity change. This work is compared with cho approach [29] for both random noise attacks and smoothing noise attacks, clipping attack simulation shows that cho approach are very vulnerable to such attacks that cause severe alteration to the center of gravity of the model so it does not shown in the reported results.

The robustness is evaluated in terms of correlation coefficient between the extracted watermark bit sequence $w_n^{'}$ and the originally inserted one w_n as given by the following equation [20]:

$$Corr = \frac{\sum_{n=1}^{N-1}(w_n^{'} - w^{-'})(w_n - w^{-})}{\sqrt{\sum_{n=1}^{N-1}(w_n^{'} - w^{-'})^2 \sum_{n=1}^{N-1}(w_n - w^{-})^2}} \qquad (6)$$

Where $w^{-'}$ and w^{-} indicate respectively the averages of the watermark bit sequence $w_n^{'}$ and w_n. This correlation value measures the similarity between two strings and varies between 1 (orthogonal sequence) and +1 (the same sequence). Tables 2-4 show the correlation results for different types of attacks with different parameters.

Our proposed approach as shown in the following tables is superior that Cho approach in terms of geometric attacks, the proposed method can not cope with connectivity attacks.

6 Conclusions and future work

In this paper, we have presented a novel approach of combining watermarking within progressive compression. Each LoD is compressed and watermarked by modifying a marked vertex selected by SOM as suitable watermark carrier. This modification of vertex value is performed with respect to the mean and stander deviation of its immediate neighbours in the original 3D mesh. The proposed method is robust against different geometric attacks like random noise,smoothing,and even cropping. For future works, we plan to extend the robustness of our method to connectivity attacks. .

.

.

Table 2. Evaluation of Robustness Against Additive Noise Attacks

	Proposed Scheme			Cho Approach		
Amp	Bunny	Dragon	Venus	Bunny	Dragon	Venus
0.005	1	0.95	0.90	0.28	0.46	0.41
0.009	0.90	0.75	0.54	0.17	0.22	0.15
0.01	0.83	0.56	0.66	0.16	0.20	0.06
0.03	0.43	0.31	0.20	0.14	0.21	0.19
0.05	0.28	0.28	0.08	0.12	0.22	0.15

Table 3. Evaluation of Robustness Against Smoothing Attack ($\lambda = 0.03$)

	Proposed Scheme			Cho Approach		
N_{it}	Bunny	Dragon	Venus	Bunny	Dragon	Venus
5	1	0.954	0.954	1	1	0.95
10	0.95	0.80	0.95	0.90	0.95	0.84
15	0.66	0.546	0.80	0.90	0.83	0.83
20	0.61	0.54	0.75	0.85	80	0.87

Table 4. Evaluation of Robustness Against Cropping Attack (Cropping %)

%	Bunny	Dragon	Venus
3	0.90	0.85	0.80
5	0.90	0.75	0.65
10	0.71	0.546	0.45
20	0.71	0.36	0.45

References

1. Shinichi, M., Y.: Watermarking for 3D Polygons Using Wavelet Transform and Modified Traveling Salesman Problem. In:Journal of the Operations Research Society of Japan, vol.52, no.4, pp.402-416, (2009).
2. Kanai, S.,Data, H., Kishinami, T.:Digital Watermarking for 3D Polygons Using Multiresolution Wavelet Decomposition.In:Proceedings of 6th IFIP WG 5.2 GEO-6, pp.296307,(1998).
3. Ohbuchi, R., Aono, M.:Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications. IEEE Journal on Selected Areas in Communications, vol. 16, pp. 551560, (1998).
4. Benedens, O.:Geometry-based watermarking of 3D models. In:IEEE Computers and Applications. vol.19, no.1, pp.46-55, (1999).
5. Yeo, B., Yeung, MM.: Watermarking 3D objects for verification. IEEE Computers and Applications, vol.19, no.1, pp.36-45, (1999).
6. Ohbuchi R, Masuda H, Aono M.: Watermarking Three Dimensional Polygonal Models Through Geometric and Topological Modifications. IEEE Journal on Selected Areas in Communications, vol.16, no.4, pp.551-60, (1998).
7. Ohbuchi R, Masuda H, Aono M.: Data Embedding Algorithms for Geometrical and non-Geometrical Targets in Three-Dimensional Polygonal Models. In: Computer Communications, vol.21, no.15, pp.13441354, (1998).
8. Cayre, F.,Macq, B.: Data Hiding on 3-D Triangle Meshes. In:IEEE Trans. on Signal Processing, vol.51, no.4, pp. 939949, (2003).
9. Agarwal, P.,Prabhakaran, B.: Robust Blind Watermarking Mechanism for Point Sampled Geometry. in: Proc. of the ACM Multimedia and Security Workshop, pp. 175186, (2007).
10. Wang, Y.P., Hu, M.: A New Watermarking Method for 3D Models Based on Integral Invariants. IEEE Trans. on Visualization and Computer Graphics, vol.15, no.2, pp.285-294, (2009).
11. Valette, S., Prost, R.: Wavelet-Based Multiresolution Analysis of Irregular Surface Meshes. In:IEEE Trans. on Visualization and Computer Graphics, vol.10, no.2, (2004).
12. Praun, E., Hoppe, H., Finkelstein, A.: Robust Mesh Watermarking. In: SIGGRAPH Proceedings, pp. 6976, (1999).
13. Kanai, S., Date, H., Kishinami, T.: Digital Watermarking for 3D Polygons Using Multi-Resolution Wavelet Decomposition. In: Proceedings of the sixth IFIP WG 5.2 international workshop on geometric modeling: fundamentals and applications (GEO-6), Japan, pp. 296307, (1998).
14. Yin, KK., Pan, ZG., Shi, JY., Zhang, D. : Robust Mesh Watermarking Based on Multiresolution Processing. In:Computers & Graphics, vol.25, no.3, pp.40920, (2001).
15. Guskov, I., Sweldens, W., Shroder, P.: Multiresolution Signal Processing for Meshes. In: Proceedings of SIGGRAPH 99, pp. 4956,(1999).
16. Ohbuchi, R., Takahashi, S., Miyazawa, T., Mukaiyama, A.: Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain. In: Proceedings of the graphics interface, Canada, pp. 917, (2001).
17. Ohbuchi, R., Takahashi, S.: A frequency Domain Approach to Watermarking 3D Shapes. In EUROGRAPHICS, Saarbrucken, Germany, vol.21, no.3, pp. 26, (2002).

18. Motwani M.C.: Third Generation 3D Watermarking: Applied Computational Intelligence Techniques, A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering , (2011).
19. Hoppe, H.: Progressive Mesh. In: ACM SIGGRAPH, vol. 96, pp. 99-108 , (1996).
20. Soliman, M.M., Aboul Ella Hassanien, Onsi, H.M. :Robust Watermarking Approach for 3D Triangular Mesh using Self Organization Map, submitted in Federated Confrence on Computer Science and Information System, Krakw, Poland, September, 2013.
21. Lee, H., Dikici, C., Lavou,G., Dupont, F. : Joint Reversible Watermarking and Progressive Compression of 3D Meshes. In:Visual computer, vol.27, Issue. 6-8, pp. 781-792, (2011).
22. Pajarola, R., Rossignac, J.: Compressed Progressive Meshes. IEEE Transactions on Visualization and Computer Graphics, vol.6, no.1, pp.79-93, (2000).
23. Taubin, G., Gueziec, A., Horn, W., Lazarus, F.: Progressive Forest Split Compression. In: ACM SIGGRAPH, pp.123-132, (1998).
24. Peng, J., Kuo, Y., Eckstein, I., Gopi, M.: Feature Oriented Progressive Lossless Mesh Coding. In: Computer Graphics Forum. vol.29, no.7, pp.2029-2038, (2010).
25. Valette, S., Chaine, R., Prost, R.: Progressive Lossless Mesh Compression via Incremental Parametric Refinement. In: Proceedings of the Symposium on Geometry Processing, vol. 28, pp. 1301-1310 (2009).
26. Hung-Kuang, C., Yung-Hung, C.: Progressive Watermarking on 3D Meshes. In:Broadband Multimedia Systems and Broadcasting (BMSB), IEEE International Symposium on, pp.1-7, (2010).
27. Guthikonda, S.M.: Kohonen Self-Organizing Maps, 2005.
28. Rojas R.: Neural Networks A Systematic Introduction, A book Foreword by Jerome Feldman, Springer-Verlag, Berlin, New-York(502 p.350 illustrations), (1996).
29. Cho, J.W., Prost, R., Jung, H.Y.: An Oblivious Watermarking for 3-D Polygonal Meshes Using Distribution of Vertex Norms, IEEE transaction on signal processing, vol. 55, issue.1 , pp. 144-152, DOI:10.1007/11551492-24, 2005.