This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2010 proceedings

3D MULTIMEDIA PROTECTION USING ARTIFICIAL NEURAL NETWORK

Mukesh C. Motwani*, Bobby D. Bryant*, Sergiu M. Dascalu*, and Frederick C. Harris, Jr.*,

*Department of Computer Science and Engineering
University of Nevada, Reno
Reno NV USA 89507

ABSTRACT

Watermarking based DRM implementations insert imperceptible information or watermark in digital media to trace owner of the content and deter the illegal distribution of media. In geometry based 3D watermarking algorithms, a watermark is inserted by modifying the coordinates of vertices in the mesh. It is a requirement of watermarking algorithms that this change in vertex coordinates shouldn't cause perceptible distortion. It has always been a challenge to select vertices in the 3D model which would not cause perceptible distortion on addition of watermark. This paper proposes a novel approach to overcome this challenge using Artificial Neural Networks (ANN). Feature vectors representing the geometry of the vertex and its surrounding vertices are extracted and used to train and simulate ANN. ANN is used as a classifier to determine which vertices should be selected for watermarking. Experimental results simulate various attacks to test the robustness of the algorithm.

Index Terms—3D, Artificial Neural Network, Back Propagation, Training, Watermarking

1. INTRODUCTION

Multimedia watermarking provides a solution to piracy of multimedia content by embedding a watermark or hidden piece of information in the original digital content. Integration of watermarking in Digital Rights Management (DRM) systems has also been proposed by [1] where identifiers such as IP address are used as watermarks. These identifiers can then be used within the DRM framework to trace the buyer of the digital content to deter illegal distribution. The authors in [2] propose a watermark-based document distribution protocol to address the problem of tracing unauthorized distribution of sensitive digital documents such as images and audio. The scope of this paper to propose a novel algorithm for watermarking of 3D meshes.

3D watermarking approaches have been discussed in [3]. In [3], various algorithms on watermarking of three dimensional models are explained briefly and classified into spatial and spectral domain watermarking. Alface [4] has done a thorough survey with classification and critical analysis of watermarking algorithms for 3D models. Benedens [5] has proposed selection of feature points on the 3D model and the comparison with the original model to determine whether the feature point has been moved inside or outside the surface along the normal.

It has always been a challenge in watermarking to insert more information without causing any visible distortion. Moreover, the perception of the human eye for 3D meshes cannot be modeled as a linear mathematical model. This paper proposes a novel approach to use artificial neural networks (ANN) to model non-linear perception of the human eye to the addition of hidden information or watermark in a 3D model. Neural networks have been trained to perform complex functions in various fields including pattern recognition, identification, classification, speech processing, computer vision, and control systems. ANN has also been used for image watermarking [6]. There is no published work which uses ANN for watermarking of 3D models.

Neural networks [7] are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. Neural networks are trained to perform a particular function by adjusting the values of the connections (weights) between elements. Neural networks are trained so that a particular input leads to a specific target output. During training the network parameters or weights are adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are needed to train a network. Once trained, the neural network can be used to determine the output when input is fed.

This paper discusses watermarking of 3D models in the spatial domain. The uniqueness of the paper lies in using an artificial neural network as a classifier to select the vertices which are suitable for watermarking.

2. WATERMARKING ALGORITHM

2.1 3D Model

A 3D model is a collection of polygonal faces approximating a real 3D object. The vertices represent a model's location and orientation in space, whereas edges connect the vertices to form faces which approximate the surface. Fig. 1 show the wire frame models or mesh structures of commonly used models. All the vertices that a vertex under consideration is connected to, is called the 1-ring neighborhood of a vertex. Fig. 2 shows the 1-ring neighborhood of a vertex in a model.

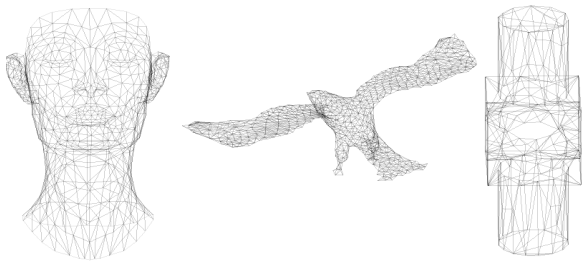

Fig.1. Mesh Structure of 'Nefertiti', 'Eagle'and 'Mechanical' Models

978-1-4244-5176-0/10/$26.00 ©2010 IEEE

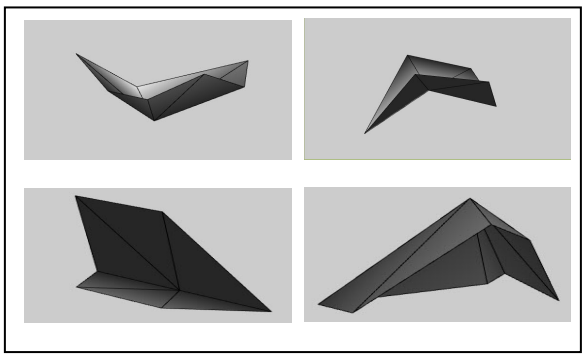

Fig.2. Plots of Vertex and their Adjoining Vertices (1-Ring)

2.2 Normalizing and Shifting of 3D Model

Initially, the 3D model's center of mass is determined and shifted to the origin (if it is not at the origin) of the rectangular co-ordinate system. Also, the coordinates of the vertices in the model are scaled to lie between -1 and +1 units, that is, the model is normalized. Normalizing and shifting the model to the origin ensures robustness of the system to be able to train ANN with models of widely varying shapes and sizes.

2.3 Artificial Neural Network

The objective is to watermark a 3D model at those locations which will produce imperceptible distortions in the final watermarked model. This task can be achieved by selecting vertices where addition of watermark data will not produce visible distortion. The artificial neural network needs to be trained to recognize different topologies of 1-rings of vertices of a model. Fig. 3 shows the process of training the neural network by feeding the geometry of 1-ring vertices as feature vectors.

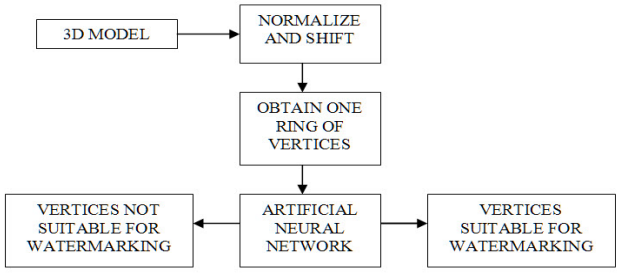

Fig.3. Training of Artificial Neural Network

2.3.1 Calculation of Feature Vector

Feature vector, F includes angles formed between the surface normal corresponding to the triangular face with v as centre vertex and the average normal N as shown in Fig. 4. A combined measure such as an average of these angles won't capture the fine variations in geometry. Thus, the proposed feature vector exploits the geometry of the 1-ring vertex neighborhood to capture the fine variations. Valence of a vertex is the count of how many other vertices the vertex is connected to in the 3D model. Thus, the length of the feature vector would be equal to the valence of the vertex. However, feature vectors used for training of ANN have to be of fixed length. Thus, vertices with only valence 6 are selected for feature extraction. Semi-regular meshes have vertices with valence of either 6 or 4. Valence 6 vertices are also called as regular vertices since most of the vertices in a 3D model have valence 6. Thus, the feature vectors used in the proposed algorithms also have length of 6.

The following steps are implemented to compute the feature vector corresponding to a vertex:

Step 1: Consider a vertex v with valence equal to 6 from the mesh. Thus, the number of its adjacent faces, M is equal to 6. Compute normal's N_i to each face which is formed by v and its neighboring vertices v_i as shown in Fig.4

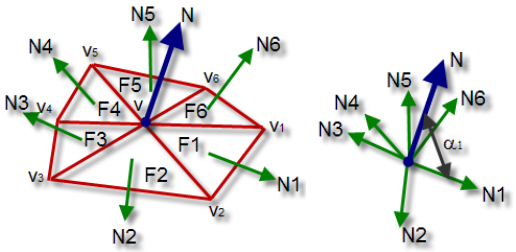

Fig.4. Surface normal's (in green) and average normal (in blue) for a 1-ring vertex neighborhood.

Step 2: Compute average resultant vector N of all the above normals passing through v.

$$\mathbf{N} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{N}_i \qquad \ldots (1)$$

Step 3: Now compute angles α_i between each pair of N_i and N.

$$\alpha_i = \cos^{-1}\left(\frac{\mathbf{N}_i.\mathbf{N}}{|\mathbf{N}_i||\mathbf{N}|}\right) \qquad \ldots (2)$$

Step 4: Assign angles α_i to the feature vector to be fed to ANN.

$$\text{Feature Vector } F = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ \alpha_5 \ \alpha_6] \qquad \ldots (3)$$

The above parameter α_i represents local geometry or shape of a surface or region and is fed as features vectors to ANN for training and simulation. Thus, if the region around the considered vertex is mostly flat, the angles α_i will be small in magnitude since the face normals will be almost parallel to the average normal. However, if the region represents a peak as shown in one of the vertices in Fig. 2, the angle between the face normal and the average normal through the vertex, α_i will have a larger magnitude.

2.3.2 Training of Artificial Neural Network

The artificial neural network is trained to recognize which types of 1-rings are suitable for insertion of watermark and which ones are not. The angle variations of each vertex having valence 6 are computed as given by Eq. 2 and will be fed to the neural network in order to train it to recognize different types of geometry of the vertices.

The neural network is trained to recognize vertices suitable for watermarking according to the topology of the one-ring surrounding the vertices. Training is done by manually adding noise to the vertices and the human operator determines if the addition of noise is perceptible or not. Typically, vertices having a surrounding 1-ring which is flat, or vertices which represent peaks of raised surfaces cause visible distortion even at the slighest addition of watermark. The vertices in which visible distortion is produced after adding small amount of information are labelled as

unsuitable for watermark insertion. On the other hand, the vertices in which addition of information does not produce visible distortion are labelled as suitable for watermark insertion. 325 sets of vertex rings with different geometrical structures and extracted from 5 normalized 3D models were evaluated by 2 human operators to reduce the human error of deciding whether to insert the watermark or not. Since there is no Human Visual System for 3D models, the labelling of output vectors is a manual process. Thus, human intelligence is transferred to the classifier.

Fig. 5 shows the architecture of the back propagation neural network being used to select vertices for watermarking. The ouput of the neural network is trained to be either 1 for a vertex to be watermarked, or -1 otherwise. 20 neurons are selected in the hidden layer.

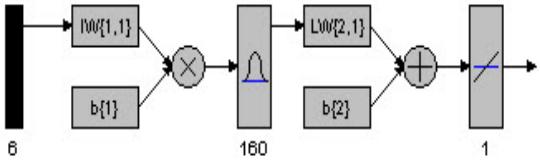

Fig.5. Neural Network Architecture

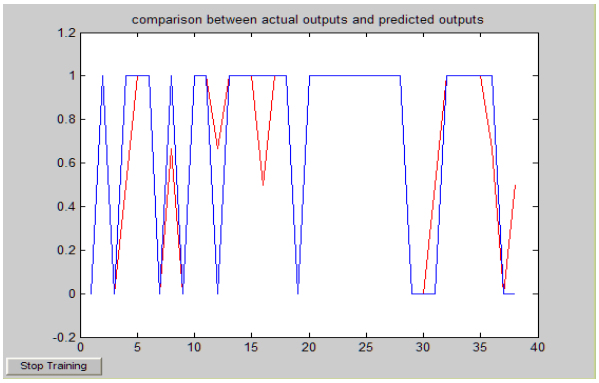

Fig.6. Testing of Neural Network
----- Actual Output
----- Produced Output

Fig. 6 shows the output produced after testing of the trained neural network. The *x*-axis of Fig. 6 is the number of epochs and *y*-axis is the value of the output layer node. The output produced by the neural network follows the actual output very closely. Thus, the neural network can be said to be successfully trained.

2.4 Insertion of watermark

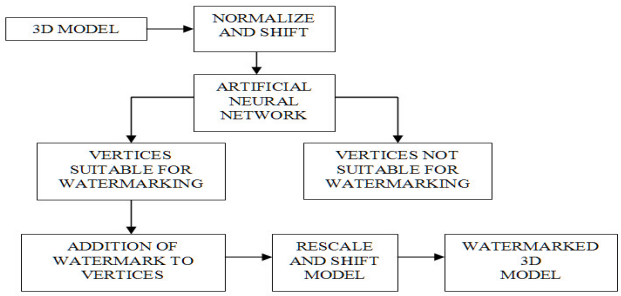

Fig.7. Watermarking Insertion Process

Fig. 7 gives the overview of the watermarking insertion process. Feature vectors are extracted from vertices with valence equal to 6 from the unprotected 3D model and fed as inputs to the trained neural network. The trained ANN then selects the vertices for watermarking. Fig. 8 shows these vertices (in dark red) in the models. It can be easily verified from these figures that the neural network is indeed choosing vertices on the mesh which are suitable for watermarking, and no vertices are selected from smooth regions such as the Nefertiti's cheeks. A random sequence is inserted in the selected vertices. Thus, for each co-ordinate (x, y, z) of a vertex selected to be watermarked, we have:

$$v'(x, y, z) = v(x, y, z) + KW \quad \ldots (1)$$

where,
$v'(x, y, z) =$ Watermarked Vertex,
$K =$ Scaling Factor,
$W =$ Watermark Data

A private key stores the indices of the vertex coordinates and the value of (x, y, z) where watermark was inserted. Finally, the model is re-shifted to its initial location in space and the co-ordinates are also re-scaled. Thus, the watermark is inserted in the geometry of the model and this watermarked model can be distributed for use by others. The watermark inserted can be the logo of a company, the designer's identification, the user's signature or any other intellectual property. This watermarking method modifies only the locations of vertices, without changing the connectivity of vertices. Plots of some of the watermarked models are shown in Figs. 9, 10 and 11. As it can be seen from the plots, there is imperceptible distortion between the original model and the watermarked one. This proves that the neural network adds watermark data to those vertices where visible distortion will not be produced in the model.

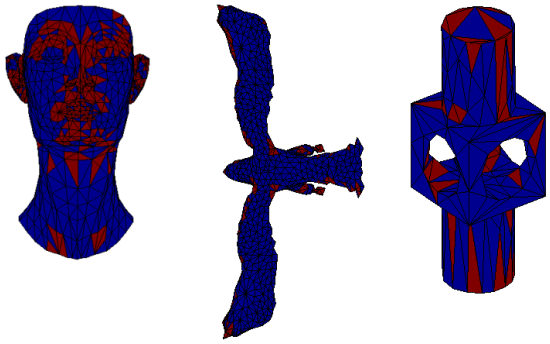

Fig.8. Vertices Selected for Watermarking (in Red)

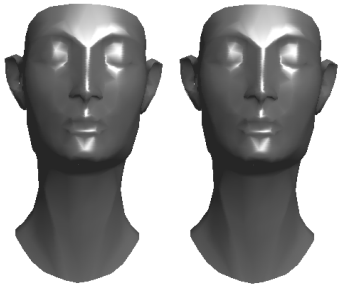

Fig.9 (a). Original 'Nefertiti' Fig.9 (b). Watermarked 'Nefertiti'

Fig.10 (a). Original 'Eagle'

Fig.10 (b). Watermarked 'Eagle'

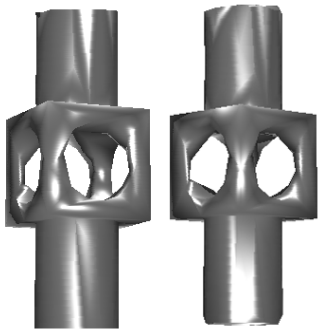

Fig.11 (a). Original 'Mechanical'

Fig.11 (b). Watermarked 'Mechanical'

3. EXTRACTION OF WATERMARK

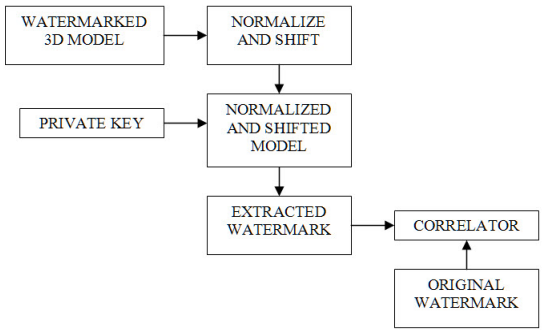

Fig.12. Watermark Extraction Process

The process of semi-blind watermark extraction is as shown in Fig. 12. In the watermark extraction process, the watermarked 3D model is normalized and shifted to the origin. The private key contains the indices of the watermarked vertices and the vertex co-ordinates of the original unprotected model.

The correlation coefficient is a number between -1 and +1 which measures the degree to which two variables are linearly related. Computing the correlation is a common method used to determine the extent of similarity between the original watermark, and the extracted watermark, as seen in [8]. The output of the correlator is a Pearson's correlation coefficient. The extracted watermark and the original watermark are correlated and a high ad-hoc value of correlation coefficient greater than 0.7 will prove the ownership of the 3D model.

Percentage of correlation between the recovered watermark and original watermark is 100% in the absence of any attacks on the watermarked model.

4. EXPERIMENTATION AND RESULTS

An attack on a 3D model is an attempt to remove the watermark, but still retain enough of the model so that it can be used. 3D models are prone to operations like cropping, smoothing, noise addition, translation, rotation and scaling, which may destroy the watermark. This is not desired as the 3D model's ownership or copyright integrity inserted as watermark may be destroyed as well. Thus, it is important that the watermark inserted should be robust enough to handle such attacks. To prove the efficiency of our method, typical attacks were simulated on the watermarked models. Table 1 gives the summary of tests and results for some models.

Model	Nefertiti	Eagle	Mechanical
Total Number of Vertices	645	1000	175
Total Number of Faces	1252	1996	358
Vertices Modified by Watermarking	269	153	62
Correlation after Uniform Scaling	100%	100%	100%
Correlation after Noise insertion	72.69%	69.67%	73.32%
Correlation after HC Smoothing	75%	72%	62%
Correlation after cropping	50.1%	95%	85.01%

Table 1. Correlation Results for the Models After Perfoming Various Attacks

4.1 Noise:

This attack was simulated by adding normally distributed random numbers (with mean 0 and variance 0.3). Such an attack does affect the extracted watermark, but the correlation is still above the predetermined threshold of 0.7. This threshold was found after attacking the watermarked model, and finding how much of the original watermark remains after the noise attack. Fig. 13 shows the result of noise attack on the watermarked models.

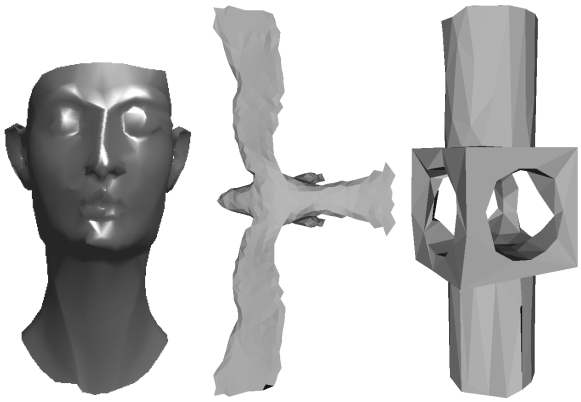

Fig.13.Models with Additive Gaussian Noise

4.2 Smoothing:

The HC smoothing algorithm is described in detail in [9]. Smoothing has a considerable effect on the watermarked model. By smoothing, large transitions in surface levels are minimized by shifting or removal of some vertices. This resulted in degradation of the watermark. Fig. 14 shows the effect of HC smoothing on watermarked models.

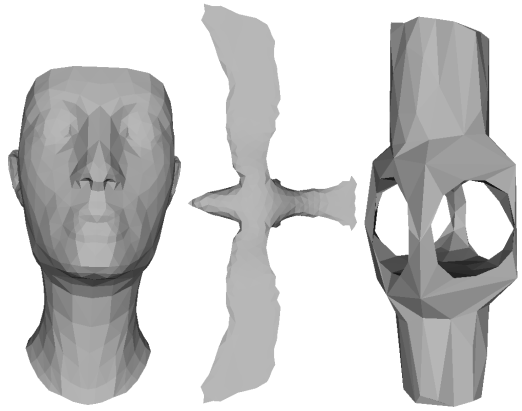

Fig.14. HC Smoothed Models

4.3 Cropping:

Cropping refers to removal/chopping of a part or parts of a model. The amount of watermark destroyed depends upon the extent of cropping. This necessitates adequate presence of the watermark in various regions. Fig. 15 shows the results after cropping of models. The technique is robust against cropping; a high value of correlation is obtained between the original watermark and the extracted one. Since no watermark was inserted in the legs and part of the wings of the 'Eagle' model, no information was lost when the legs were cropped, thus giving a 95% correlation between the original watermark and extracted watermark.

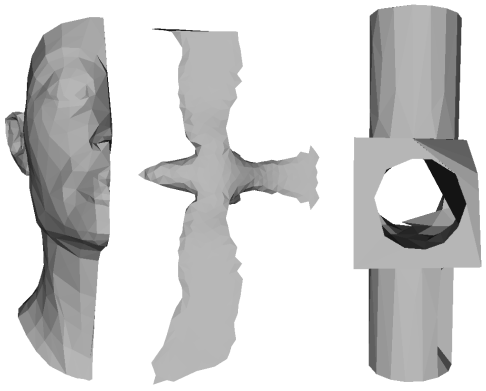

Fig.15. Cropped Models

4.4 Scaling, Translation, Rotation and Affine attacks:

This method is completely resistant to uniform scaling and affine attacks. The change in these parameters does not affect the relative orientation of the normal's at the vertices. Thus, our algorithm is invariant to scaling and affine attacks and gives 100% correlation. Fig. 16 (a) and (b) shows the translation and rotation of the watermarked model.

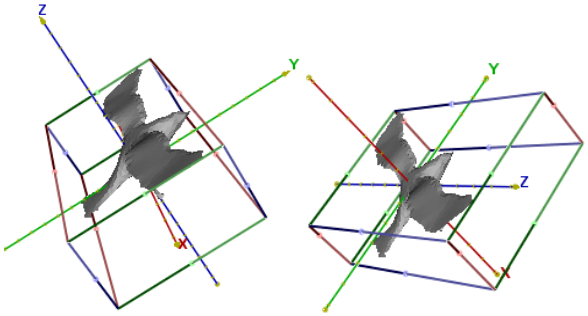

Fig.16. (a). Translated 'Eagle' Model

Fig. 16. (b). Rotated 'Eagle' Model

5. CONCLUSION

This paper proposes a novel watermarking algorithm in which vertices are selected from the 3D model for watermarking by ANN without causing perceptible distortion. The proposed method uses an artificial neural network to select vertices based on the geometry of the ring of vertices surrounding them, and gives good results (visual and analytical) for various types of surfaces (flat, curved, uneven, etc.) present in 3D models. Also, the system is robust against various possible attacks. Future work will focus on using multiple ANN's with different number of inputs so that vertices other than valence 6 can also be watermarked. ANN will also be evaluated to determine the amount of watermark to be added in addition to selecting the vertices to be watermarked.

6. REFERENCES

[1] L. Quan and L. Hong, "An intelligent digital right management system based on multi-agent," in Computer Science and Software Engineering, 2008 International Conference on, vol. 1, Dec. 2008, pp. 505–507.

[2] S.-C. Cheung, D. K. W. Chiu, and C. Ho, "The use of digital watermarking for intelligence multimedia document distribution," J. Theor. Appl. Electron. Commer. Res., vol. 3, no. 3, pp. 103–118, 2008.

[3] Kai Wang, Guillaume Lavou´e, Florence Denis, and Atilla Baskurt : "Three-Dimensional Meshes Watermarking: Review and Attack-Centric Investigation," *Conference on Information Hiding 2007*, March 2007.

[4] P. R. Alface : "Perception and Re-Synchronization Issues for the Watermarking of 3D Shapes," *Universite catholique de Louvain(UCL)*,http://edoc.bib.ucl.ac.be:81/ETD-db/collection/available/BelnUcetd-10192006-155635/ 2006.

[5] O. Benedens: "Robust Watermarking and Affine Registration of 3D Meshes," *Proc. of 5th International Workshop on Information Hiding, NoordWijkerhout, Netherlands*, October 7-9, pages 177-195, 2002.

[6] Shi-chun Mei; Ren-hou Li; Hong-mei Dang; Yun-kuan Wang, "Decision of image watermarking strength based on artificial neural-networks," *Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on* , vol.5, no., pp. 2430-2434 vol.5, 18-22 Nov. 2002

[7] Alexander I. Galushkin, "Neural Networks Theory," *Springer* September 2007 ISBN-13: 978-3540481249

[8] Mauro Barni, Franco Bartolini, Vito Cappellini, Massimiliano Corsini, Andrea Garzelli: "Digital watermarking of 3D meshes," *SPIE proceedings series*, SPIE, Bellingham WA, ETATS-UNIS (2004).

[9] J. Vollmer, R. Mencl, and H. Muller: "Improved Laplacian Smoothing of Noisy Surface Meshes," *Computer Graphics Forum*, 18(3):131-138, 1999.