Tamper Proofing 3D Models

Mukesh Motwani[#1], Balaji Sridharan[*2], Rakhi Motwani[#1], and Frederick C. Harris, Jr.[#1]

[#1]*Univerity of Nevada, Reno USA*

[1]mukesh@cse.unr.edu

[3]fredh@cse.unr.edu

[*2]*Vishwakarma Institute of Technology, Pune India*

[2]balaji.sridharan@ieee.org

Abstract - **This paper describes a novel algorithm designed for tamper proofing of 3D models. Fragile watermarking is a known technique for detecting location of tamper in the artwork. However, to detect even minute changes, the watermark needs to be distributed throughout the 3D model. This poses as a challenge since watermarking all vertices can cause perceptible distortion. The proposed algorithm solves this problem by inserting a watermark in all the vertices of a 3D model. The watermark is randomly added to each and every vertex of the 3D model by modifying the coordinate location of the vertices. Genetic Algorithms have been used to find the near optimal coordinate location of the watermarked vertex. The fitness function chosen is the Signal to Noise Ratio of the 1-ring neighbourhood of the vertex. This ensures that there exists no distortion in the watermarked model. The proposed approach is computationally inexpensive and experimental results indicate that the algorithm can efficiently detect location of any kind of unauthorized data modification.**

***Keywords*— fragile watermarking, genetic algorithms, 3D mesh model, SNR**

I. Introduction

3D Digital content has developed by leaps and bounds in the past few years. The designers exhibit, share and sell this valuable artwork as 3D models or meshes. However, the duplication and unauthorized modification of these meshes have made watermarking an essential part of copyright protection while selling these models. Watermarking can be broadly classified into 2 types. *Robust Watermarking* is used to indicate ownership. Thus, in such a type of watermarking scheme, the watermark should be preserved even after attacks. On the other hand, *Fragile Watermarking* is used to prevent unauthorized modification of the data. It also helps in locating the regions where the data has been modified illegally. Fragile Watermarking is a common used technique for tamper detection. For tamper proofing, it is imperative to detect the region or location of tamper. This can be done efficiently if and only if we modify the position of each and every vertex in the model. However, that poses a challenge since perceptible distortion can be caused by added watermark in all vertices. This paper successfully solves that challenge by watermarking all the vertices in a model without causing perceptible distortion.

A comprehensive survey of 3D model watermarking techniques has been done by [1]. There has been a significant amount of research to embed watermarks in images, video sequences and audio sequences using Genetic Algorithms (GA). In [2], the algorithm uses GA to find the optimum embedding density to modify the DCT coefficients of the host image for image watermarking. The fitness function is Peak Signal to Noise Ratio (PSNR) for imperceptibility of the watermark and Normalized Cross Correlation (NCC) for the robustness of the watermark. GA has also been used in a dual image watermarking Scheme [3]. The above scheme employs insertion of a robust watermark into a secondary image which acts as the fragile watermark for the host image. Embedding of fragile watermark uses a watermark embedding factor which is obtained using GA. Thus, the objective of the GA is to maximize PSNR. Genetic Algorithms have also been used for fragile watermarking of images [4]. In this algorithm, the block edge characteristics of an image are used and the edge information is used as the fitness function parameter. In Audio watermarking [5], a pseudo noise (PN) sequence is embedded in the audio sequence. During extraction, GA is used with the fitness function being the cross correlation between estimated PN sequence and the spread spectrum. Genetic Algorithms (GA) have never been used for fragile watermarking of 3D models. The novelty of this paper lies in the use of Genetic Algorithms to generate a fragile watermark, which is inserted in all the vertices of the 3D model.

Genetic Algorithms [6] are a branch of evolutionary algorithms that use evolution and Darwin's theory of Survival of the Fittest as a source of inspiration to solve computational problems. All the possible solutions to the given optimization problem are called chromosomes. Each generation has a specific number of chromosomes, which constitute the population of that generation. The fundamental block of GA is the fitness function. The fitness function contains the parameter that has to be optimized. Each chromosome is evaluated through this fitness function and returns a value known as the fitness value of that chromosome. According to the best fitness values, some chromosomes are selected to reproduce and they populate the next generation. The most common operators used for reproduction are selection, mutation and crossover operators.

II. Proposed Approach

A. *3D Model Representation*

A 3D mesh consist of a vertex list which gives us the co-ordinates in space of each and every vertex that constitutes the model and a face list which tells us the manner in which the vertices are connected to each other. Each vertex in the model is represented by (x, y, z) coordinates in the Cartesian axis and also represents a chromosome. The 1 ring neighborhood of a

978-0-7695-3960-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ICSAP.2010.85

vertex V is defined as the surfaces formed by that vertex V with its neighbors as shown in Fig. 1.

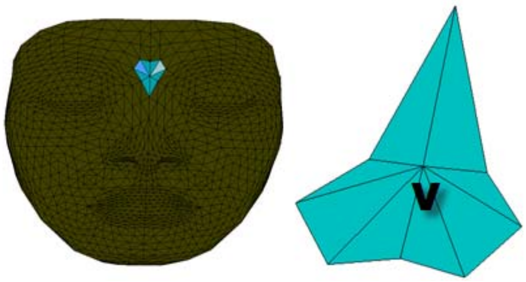

Fig 1. *Nefertiti* triangular mesh model and one ring neighborhood of vertex index 106. The colored patch on the model's forehead shows the position of the one ring in the *Nefertiti* model.

B. *Fitness Function*

Modifying the coordinate position of a vertex can be considered as adding random noise with reference to the original location of the vertex. Thus, the amount of watermark added to a vertex is equivalent to noise being added to the vertex. The objective of the Genetic Algorithm is to maximize Signal to Noise Ratio for the entire model. The choice of fitness function determines what parameter GA is going to optimize. Chromosome with the best fitness value at the end of pre-determined number of generations is considered as the optimized output of the algorithm. The best fitness value from a pool of chromosomes corresponds to the chromosome with the least fitness value.

For efficient fragile watermarking, it is necessary to modify all the vertices in the model. While doing so, great care has to be taken to avoid causing any distortion in the model. One of the parameter that controls the amount of distortion is the Signal to Noise Ratio (SNR) obtained from [7].

$$SNR = \frac{\sum_{i=1}^{N} X_i^2 + Y_i^2 + Z_i^2}{\sum_{i=1}^{N}((X_i - X'_i)^2 + (Y_i - Y'_i)^2 + (Z_i - Z'_i)^2)} \quad (1)$$

$$Fitness\ Function = \frac{1}{SNR} \quad (2)$$

C. *Genetic Algorithm parameters*

TABLE I – Genetic Algorithm Parameters

Population Size	100	
No. of generations	20 generations	
Initial range	Upper bound	Max value of the coordinates of 1-ring
	Lower bound	Min value of the coordinates of 1-ring
Creation Function	Uniform	
Selection Operator	Stochastic uniform	
Crossover Operator	Scattered Crossover	
Crossover Rate	0.8	
Mutation Operator	Uniform Mutation	
Mutation Rate	0.001	
Elite count	2	

D. *Watermark Generation and Insertion*

The flow chart for the algorithm is shown in Fig.2

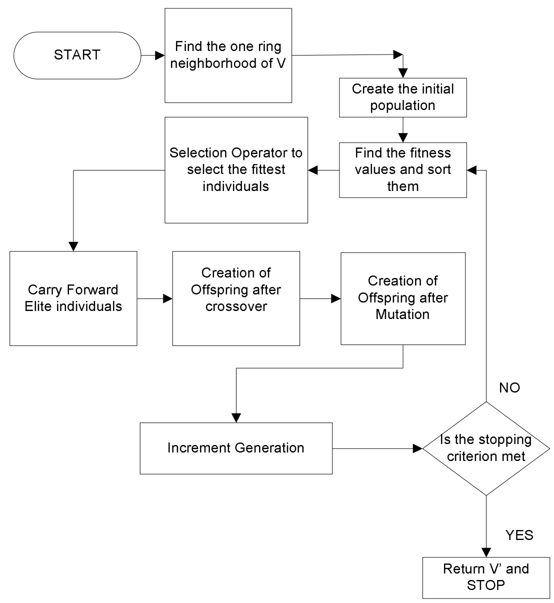

Fig 2. Flow chart for generation of watermark for individual vertex

E. *Watermark Extraction*

The extraction of watermark could be of 2 types, *blind* and *non-blind.* In blind extraction, the original model is not needed to extract the watermark. In non-blind approach, original model is subtracted from the watermarked model and thus the watermark is extracted. The extraction algorithm that is used is non-blind as shown in Fig 3.

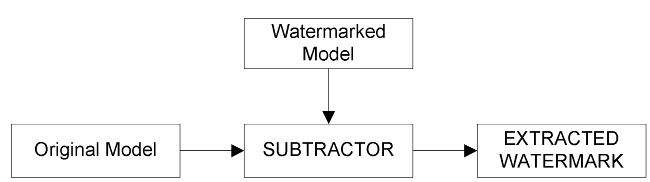

Fig 3. The extraction process.

III. EXPERIMENTAL RESULTS

A. *Imperceptibility*

The algorithm was run with a population size of 100 for 20 generations. The algorithm was tested for 4 different models – Mannequin, Bunny, Horse, and Smiley. Results for the Mannequin, Bunny, Horse and Smiley models are shown in this section.

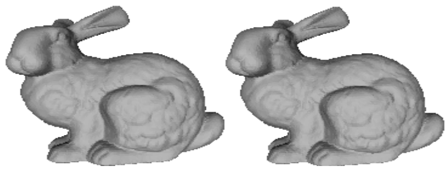

Fig 4a) Bunny

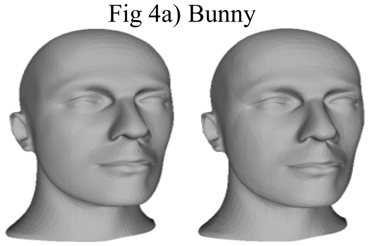

Fig 4b) Mannequin

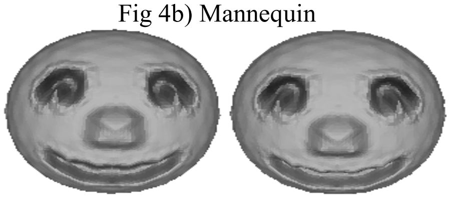

Fig 4c) Smiley

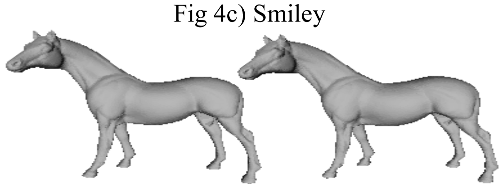

Fig 4d) Horse

Fig 4. The models to the left are original models and the models to the right are watermarked models.

TABLE II

Model	Bunny	Mannequin	Smiley	Horse
No. of vertices in the original model	17446	6743	8194	19851
No. of vertices watermarked	17446	6743	8194	19851
SNR (Signal to noise ratio in decibels)	137.75 dB	126.64 dB	121.5 dB	132.5 dB
Average time in seconds taken per vertex	0.43 seconds	0.48 seconds	0.32 seconds	0.37 seconds

As the results in table II and the figure 4 indicate there is no perceptible distortion between watermarked and original models.

B. *Number of generations*

The algorithm was run on many different types of 1 ring neighbourhood before deciding the parameters for the GA. The advantage of using 1 ring representation of the vertex is that it's easy to visualize the change in location of the vertex.

1. Nefertiti – Vertex number 25: GA applied for 100 generations

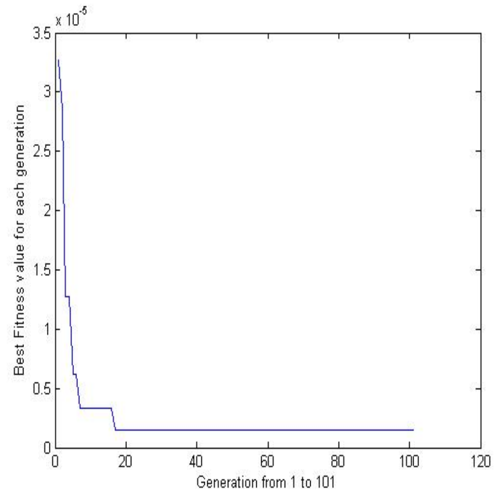

Fig 5. The change of fitness value after 100 generations

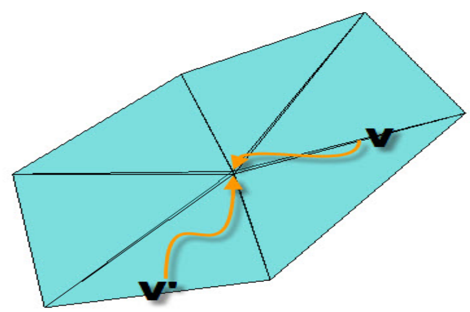

Fig 6. The change in position of vertex V to V' after 100 generations.

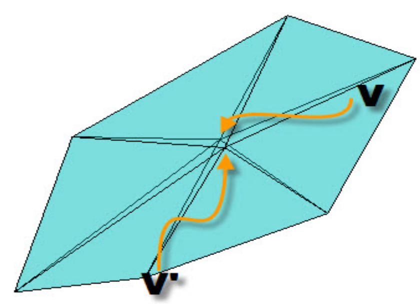

Fig 7. The change in position of vertex V to V' after 20 generations.

2. Smiley – Vertex number 25: GA applied for 100 generations

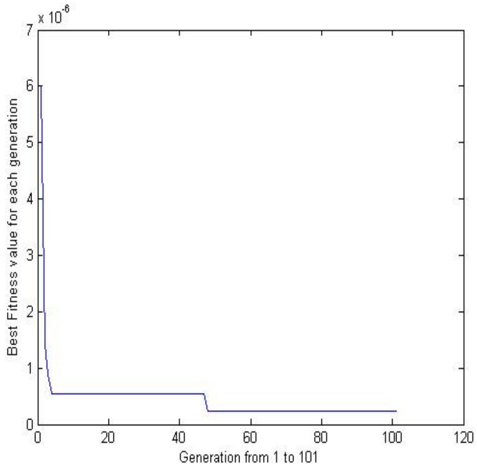

Fig 8. The change in fitness value after 100 generations

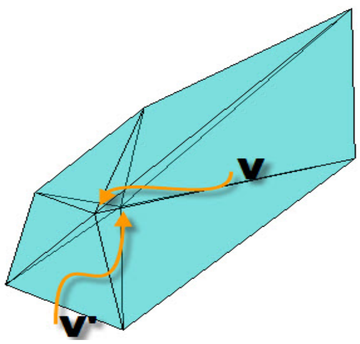

Fig 9. The change in position of vertex V to V' after 20 generations.

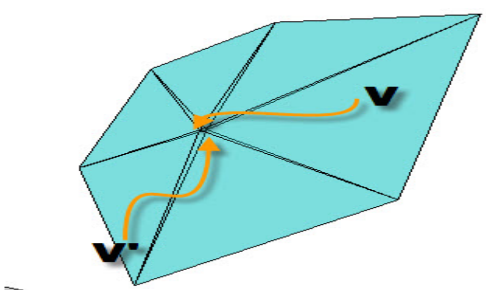

Fig 10. The change in position of vertex V to V' after 100 generations.

As shown in the graphs (Fig 5,Fig 8) there is a very small change in fitness values after 20 generations. The ideal output of the GA is when the SNR is maximized so that V' overlaps with V. This would definitely cause zero distortion. But, for zero distortion it is not at all necessary for the point V to overlap with V' as shown in Fig 6, 7, 9, 10. Thus, the algorithm was run only for 20 generations and still the change in location of the vertex coordinate does not cause any perceptible distortion. Such an approach also decreases the computational costs of the algorithm.

C. *Attacks on the model.*

The attacks for fragile watermarking of 3D model differ from the attacks for robust watermarking. The attacks on the fragile watermarked 3D model deal only with unauthorized modification of any region in the model.

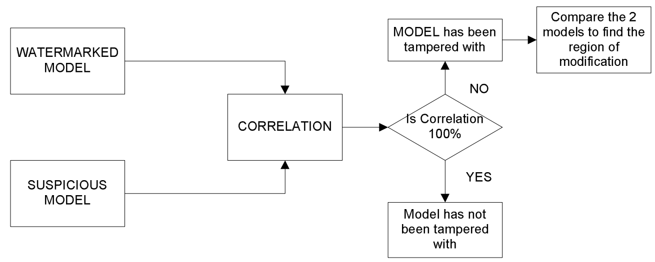

Fig. 11 Detection of unauthorized modification

The watermark is not affected by attacks such as translation, rotation and scaling as shown in. Fig 12 and Fig 14. Correlation of 100% is obtained for affine transformations.
The other type of attacks are deformation or cropping attacks, where the vertices are deformed as shown in Fig 13a. In such a case, the algorithm should be able to find out the location at which the deformation or cropping has taken place as shown in Fig 13c.

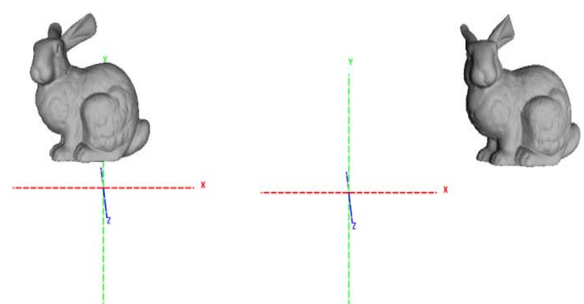

Fig 12a) Watermarked Bunny 12b) Translated model
(Red, green and blue lines represent the X, Y and Z axes resp.)

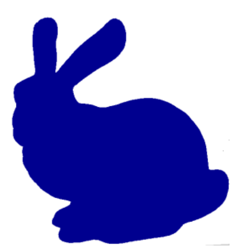

Fig 12c) Tamper Region detection

Since in Fig. 12c, there are no red colored patches, the model hasn't been tampered with.

Fig 13a) The original bunny

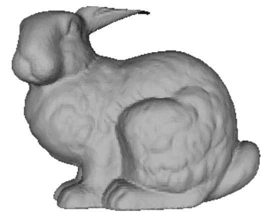

Fig 13b) The deformed bunny

Fig 13c) The modified region shown by red patch

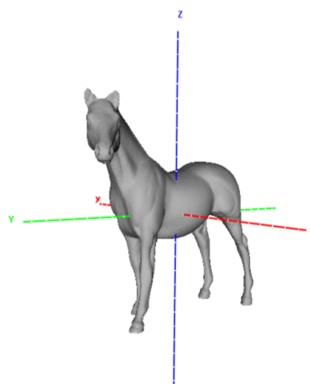

Fig 14a) Original horse

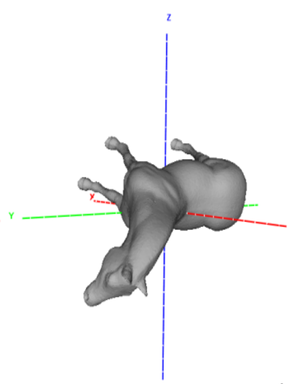

Fig 14b) Rotate by 250°

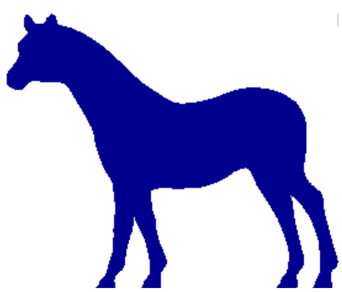

Fig 14c) No red patches indicating the model has not been modified

Apart from these attacks, simultaneous modifications at more than one location can be detected due to the fragile watermarking as shown in Fig 15. The above figures were generated using tools given in [8] and [9].

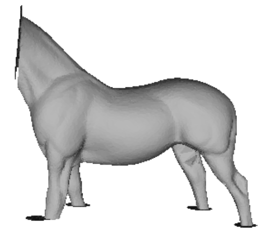

Fig 15a) Horse deformed

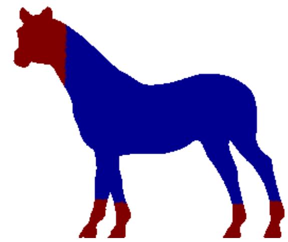

Fig 15b) The deformed regions

IV. CONCLUSION

In this paper, a fragile watermarking of 3D models for tamper proofing using Genetic Algorithms is proposed. The novelty of the algorithm is that 100% of the vertices in 3D model are watermarked. Although Genetic Algorithms are known to be slow, the proposed unique approach of halting the GA before it reaches the global optima relaxes the constraints on computational complexity drastically. Thus, the challenge of computational constraints to provide a fragile watermarking algorithm that is truly efficient and still maintaining randomness in the watermarking process is overcome. The results showed that by doing so the perceptibility of the model was not compromised and the Signal to Noise Ratio achieved is significantly high. The proposed algorithm does exceptionally well detecting location of multiple regions that were tampered in 3D model.

REFERENCES

[1] Jean-Luc Dugelay , Atilla Baskurt, Mohamed Daoudi, "3D Object Processing: Compression, Indexing and Watermarking, " Wiley, 2008

[2] Yongqiang, Chen; Lihua, Peng, "Optimal Image Watermark Using Genetic Algorithm and Synergetic Neural Network," *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on* , vol.3, no., pp.209-212, 10-11 Oct. 2009

[3] ElShafie, D.R.; Kharma, N.; Ward, R., "Parameter optimization of an embedded watermark using a genetic algorithm," *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on* , vol., no., pp.1263-1267, 12-14 March 2008

[4] [Sang-Kwang Lee; Yo-Sung Ho, "Fragile watermarking scheme using a simple genetic algorithm," *Consumer Electronics, 2002. ICCE. 2002 Digest of Technical Papers. International Conference on* , vol., no., pp. 190-191, 2002

[5] Sedghi, S.; Mashhadi, H.R.; Khademi, M., "Detecting Hidden Information from a Spread Spectrum Watermarked Signal by Genetic Algorithm," *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* , vol., no., pp.173-178, 0-0 0

[6] Mitchell Melanie, "An Introduction to Genetic Algorithms," The MIT Press, 1999.

[7] Bennour, J.; Dugelay, J.-L., "Toward a 3D watermarking benchmark," *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on* , vol., no., pp.369-372, 1-3 Oct. 2007

[8] G. Peyré: "A Toolbox to Perform Computations on Graph," URL:http://www.ceremade.dauphine.fr/~peyre/, Toolbox_Graph.

[9] MeshLab: "An open source, portable, and extensible system for the processing and editing of unstructured 3D triangular meshes," URL: http://meshlab.sourceforge.net/ ..