Watermarking of 3D Mesh Models Robust to Basic Geometric Transformations

S. Zafeiriou, A. Kalivas, A.Tefas, N. Nikolaidis, I.Pitas

Computer Vision & Image Processing Group, AIIA Laboratory
Department of Informatics, Aristotle University
54124 Thessaloniki, Greece
E-mail: pitas@zeus.csd.auth.gr
Web: http://poseidon.csd.auth.gr

Abstract. A novel method for 3D model watermarking robust to geometric distortions such as rotation, translation and scaling, is proposed. A ternary watermark is embedded in the 3D model using geometric deformations of the vertices, without altering the vertex topology. Prior to embedding and detection the model is rotated and translated so that its center of mass and its principal axis coincide with the origin and the z-axis of the Cartesian coordinate system. This transformation grants the method robustness to translation and rotation. Robustness to scaling is achieved by restricting the vertex deformations to occur only along the r coordinate of the corresponding (r, θ, ϕ) spherical coordinates system. Simulation results indicate the ability of the proposed method to deal with the aforementioned attacks giving more than satisfactory results.

1 Introduction

In the last decades, many new technologies for representation, storage and distribution of digital media information became available. Furthermore, the amount of digital data distributed through communication networks (e.g. through the Internet), has increased rapidly. The danger of copying, tampering and transmitting copyrighted data through these networks, generated an increased demand for robust methods of copyright protection and ownership security.

A new important element in security research that aims to protect intellectual property rights and data ownership is information hiding [1]. Numerous systems have been proposed for the watermarking of 2D image data and sound, addressing with a varying success the existing difficulties, but few methods have been proposed for watermarking of 3D models. Watermarking of 3D models is gaining a lot of popularity due to the increased processing power of today's computers and the demand for a better representation of virtual worlds and scientific data.

An algorithm for embedding affine invariant watermarks is proposed in [2]. It is named *Tetrahedral Volume Ratio Embedding* and uses the ratio of volume of tetrahedrons generated from three neighboring triangles along the range of triangles in a model.

Praun et al. [3] presented a technique for embedding secret watermarks, using a spread spectrum approach. The algorithm is robust against many attacks but requires

that the user supplies not only the original model but also a manual compensation for all affine transforms.

Benedens et al. presented a technique for embedding watermarks robust against remeshing and more specifically triangle reduction attacks, named *Normal Bin Encoding* [4]. The method needs a model reorientation to provide successful detection. A second method, *Affine Invariant Embedding*, withstands geometrical transforms but needs extra information, for the detection, appended to the secret key.

In this paper, a novel technique for 3D model watermarking robust against geometric transforms is proposed. It is based on a still image watermarking technique [5]. The algorithm uses the center of mass and the principal component of the model in order to transform the data to a space, which is not affected by geometric transformations, such as translation, rotation and scaling.

2 3D Surface Watermarking

2.1 Transform of the 3D surface

A 3D model is comprised of a set of vertices $\mathbf{V}$ and a set of connections between these vertices. Each vertex $\mathbf{v_i}$ has three coordinates in the cartesian space, $\mathbf{v_i} = \{\mathbf{x_i}, \mathbf{y_i}, \mathbf{z_i}\}$. The purpose of the transform is to convert the 3D data to a 1D signal so that the embedding of the watermark can be then applied. The resulting space is invariant to rotation, translation and scaling of the 3D model and thus robustness against these attacks is achieved. A description of each step of the transform follows.

- **Center of Mass Calculation**. To find the center of mass the following equation is used

$$\mathbf{K} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i}} \mathbf{v_i} \tag{1}$$

- **Model Translation**. The model is translated so that the center of mass falls on the center of the axes.

$$\begin{aligned} x_i^{'} &= x_i - k_x \\ y_i^{'} &= y_i - k_y \\ z_i^{'} &= z_i - k_z \end{aligned} \tag{2}$$

  where k_x, k_y and k_z are the coordinates of the center of mass, x_i, y_i and z_i are the original coordinates of vertex $\mathbf{v_i}$ and $x_i^{'}$, $y_i^{'}$ and $z_i^{'}$ are the coordinates of the translated vertex $\mathbf{v_i^{'}}$. Thus the watermarking method becomes robust against model translation.
- **Principal Component Calculation**. The principal component $\mathbf{u}$ of the vertices is the eigenvector that corresponds to the greatest eigenvalue of the covariance matrix C of the vertices coordinates. The covariance matrix C is calculated in the following way:

$$C = \begin{bmatrix} \sum_{i=0}^{N} x_i^2 & \sum_{i=0}^{N} x_i y_i & \sum_{i=0}^{N} x_i z_i \\ \sum_{i=0}^{N} x_i y_i & \sum_{i=0}^{N} y_i^2 & \sum_{i=0}^{N} y_i z_i \\ \sum_{i=0}^{N} x_i z_i & \sum_{i=0}^{N} z_i y_i & \sum_{i=0}^{N} z_i^2 \end{bmatrix} \tag{3}$$

where x_i, y_i and z_i are the coordinates of the vertex $\mathbf{v_i}$.

- **Model Rotation**. The model is rotated so that $\mathbf{u}$ coincides with the z axis. Thus robustness against rotation of the watermarked model is achieved.
- **Conversion to Spherical Coordinates**. The model is converted to Spherical Coordinates. Each vertex is represented as (r, θ, ϕ). This is done in order to achieve robustness against scaling by embedding the watermark in the r component of each vertex.

Robustness to translation, rotation and scaling is achieved by applying the described transformation prior to watermark embedding. The signal $r(\theta)$ is then formed and is used in the embedding procedure. All the extra information in the original signal is discarded.

2.2 Watermark Generation and Embedding

The watermark generation procedure aims at generating a three-valued watermark $w(\theta) \in \{-1, 0, 1\}$, from the transformed signal of the vertices $r(\theta)$, given a digital key K. The algorithm uses the digital key as a seed for a gaussian random number generator. The mean and variance of the θ angles of the model vertices are the parameters of the generator that produces the watermark.

Watermark embedding is performed by altering the transformed signal according to the following formula:

$$r_w(\theta) = \begin{cases} r(\theta) & \text{if } w(\theta) = 0 \\ g_1(r(\theta), N(\theta)) & \text{if } w(\theta) = 1 \\ g_2(r(\theta), N(\theta)) & \text{if } w(\theta) = -1 \end{cases} \tag{4}$$

where g_1, g_2 are suitably designed functions based on θ and $N(\theta)$ denotes a function that depends on the neighborhood of θ. The functions g_1, g_2 are called *embedding functions* and they are selected so as to define an inverse detection function $G(r_w(\theta), N(\theta))$. The detection function, when applied to the watermarked model $r_w(\theta)$, gives the watermark $w(\theta)$:

$$G(r_w(\theta), N(\theta)) = w(\theta) \tag{5}$$

Obviously several embedding functions and the appropriate detection function can be designed giving different watermarking schemes. The function that is used in our method is based on the values of the neighboring surface vertices of the vertex to be modified.

$$g_1(r(\theta), N(\theta)) = N(\theta) + a_1 r(\theta) \tag{6}$$

$$g_2(r(\theta), N(\theta)) = N(\theta) + a_2 r(\theta) \tag{7}$$

where a_1, a_2 are suitably chosen constants and $N(\theta)$ is a local neighborhood operation of the vertices around $r(\theta)$. The sign of a_1, a_2 is used for the detection function and its value determines the watermark power.

2.3 Watermark Detection

In the watermark detection procedure we generate first the watermark $w(\theta)$ according to the watermark key K. Afterwards the model under investigation is transformed according to the transformation presented in Section 2 and the signal under investigation $r_w(\theta)$ is formed. The detection function resulting from (6),(7) is defined by:

$$G(r_w(\theta), N(\theta)) = \begin{cases} 1 & \text{if } r_w(\theta) - N(\theta) > 0 \\ -1 & \text{if } r_w(\theta) - N(\theta) < 0 \end{cases} \tag{8}$$

The detection function is valid if $a_1 > 0$ and $a_2 < 0$. This fact should be accounted for, in the design of the embedding functions. By employing the detection function in the transformed watermarked signal $r_w(\theta)$, a bi-valued detection signal $d(\theta)$ is produced:

$$d(\theta) = G(r_w(\theta), N(\theta)) \tag{9}$$

Based on the watermark $w(\theta)$ and the detection signal $d(\theta)$, we can decide whether the watermark under investigation is embedded in the model or not. The detection is based on the value by value comparison for the nonzero samples in $w(\theta)$. By comparing the watermark $w(\theta)$ and the detection signal $d(\theta)$ we form the false detection signal:

$$e_w(\theta) = \begin{cases} 1 \text{ if } & w(\theta) \neq 0 \text{ and } w(\theta) \neq d(\theta) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The false detection signal has value 1 if a watermarked vertex is falsely detected and 0 otherwise. The detection ratio is given by the ratio of the correctly detected vertices to the sum of the watermarked vertices in the 3D model.

$$D_w = 1 - \frac{\text{card}\{e_w(\theta)\}}{\text{card}\{w(\theta)\}} \tag{11}$$

The embedding functions are designed in such way, so as the probability p of a pixel to be detected as signed with g_1 or g_2, for an unwatermarked model, to be 0.5. Thus, the detection ratio in an unwatermarked model forms a binomial distribution. The cumulative distribution function (*cdf*) of the watermark detection ratio is given by:

$$P_n = p^k \sum_{i=0}^{n} \frac{k!}{i!(k-i)!} \tag{12}$$

where k is the total number of the watermarked vertices and n is the number of correctly detected watermarked vertices.

The decision about the image ownership is taken by comparing the watermark detection ratio of the model to a predefined threshold T. The value of the threshold determines the minimum acceptable level of watermark detection.

3 Watermarking Algorithm Performance

To evaluate the performance of the algorithm we use two measures. For the visual quality of the model we use the SNR, and for its robustness the ROC curves are evaluated. A more detailed description follows.

3.1 SNR Measure

To measure the SNR of a 3D model the following formula is used:

$$SNR = \frac{\sum_{i=0}^{N-1} x_i^2 + y_i^2 + z_i^2}{\sum_{i=0}^{N-1} (x_i^{'} - x_i)^2 + (y_i^{'} - y_i)^2 + (z_i^{'} - z_i)^2} \tag{13}$$

where x_i, y_i and z_i are the coordinates of vertex $\mathbf{v_i}$ before the embedding of the watermark and $x_i^{'}$, $y_i^{'}$ and $z_i^{'}$ are the coordinates of the same vertex after the embedding of the watermark.

3.2 ROC Curves

The decision on whether there is a watermark in the model, is taken by comparing the detection ratio D_w to the threshold T. For a given threshold, the performance of the system can be expressed as a function of the false alarm probability $P_{fa}(T)$ (i.e. the probability of detecting a watermark on a non watermarked model or a watermarked model with another key) and the false rejection probability $P_{fr}(T)$ (i.e. the probability of not detecting a watermark in a watermarked model using the correct key):

$$P_{fa}(T) = Prob(D_w > T|H_1) \tag{14}$$

$$P_{fr}(T) = Prob(D_w < T|H_0) \tag{15}$$

where H_0 is the event that the watermark exists in the model and H_1 is the event that the watermark under investigation doesn't exist in the model.

In an ideal case, there should be a threshold T so that P_{fa} and P_{fr} are both zero. The values of P_{fa} and P_{fr} can be calculated using the following formulas:

$$P_{fa}(T) = \int_T^{\infty} f_{D_w|H_1}(t)dt \tag{16}$$

$$P_{fr}(T) = \int_{-\infty}^{T} f_{D_w|H_0}(t)dt \tag{17}$$

where $f_{D_w|H_1}(t)$ and $f_{D_w|H_0}(t)$ are the probability density functions of D_w as a random variable and given the event H_1 and H_0 respectively.

These two equations can be solved for the independent variable T and as a result P_{fa} can be expressed as a function of P_{fr}. This function forms the receiver operating characteristic (ROC) curve of the watermarking system. The operating point where $P_{fa} = P_{fr}$ is called equal error rate (EER) and is used as a quantitative estimation of the algorithm robustness.

We assume a Gaussian distribution for both $f_{D_w|H_0}$ and $f_{D_w|H_1}$ and we calculate their means $\mu_{D_w|H_0}$, $\mu_{D_w|H_1}$ and variances $\sigma^2_{D_w|H_0}$, $\sigma^2_{D_w|H_1}$

We can then use the following formula for evaluating the ROC curve.

$$P_{fa} = \frac{1}{2}[1 - erf(M)] \tag{18}$$

where M is given by:

$$M = \frac{\sqrt{2}\sigma_{D_w|H_0} erf^{-1}(2P_{fr} - 1) + \mu_{D_w|H_0} - \mu_{D_w|H_1}}{\sqrt{2}\sigma_{D_w|H_1}} \tag{19}$$

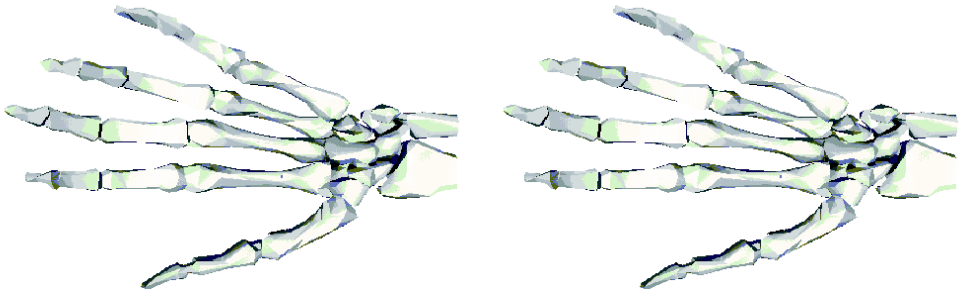
4 Experimental Results

The detection performance and robustness of the proposed approach against geometrical transformations was verified on a number of experiments involving a set of 3D models. The models used for testing were Hand, Stanford Bunny, Klingon, Dino. The experiments were realized on a Pentium II double processor 500 machine, and both the embedding and detection execution time ranged between 3 and 15 seconds, depending on the number of vertices we wish to watermark. Thus the method is sufficiently fast, although the code was not optimized for speed.

The geometrical attacks that were tested are translation, rotation and uniform scaling. Due to the invariance properties of the transform that is applied to the model prior to watermark embedding and detection, the results for these attacks were identical to the ones obtained when no attack is performed and thus they will not be presented separately. The watermark embedding power is related to the constants a_1 and a_2 and for the tests we used $a_1 = -a_2$. Several tests were conducted using various values for the embedding parameters. As shown in [5] the minimum detection threshold T that can be used for the algorithm depends on the length of the watermark $w(\theta)$. The value of T increases as the amount of watermarked samples decreases. Of course the performance is improved if we increase the number of watermarked vertices.

The Equal Error Rate(EER) and SNR values for the 3D models used in the experiments and for various values of the embedding power are summarized in Table 1.

The first model, Hand, had 1055 vertices and 2130 triangles. The original model can be seen in Figure 1(a) whereas the watermarked model with embedding power equal to 0.005 is depicted in Figure 1(b). The ROC curves for this model are depicted in Figure 2.



(a) (b)

Fig. 1. (a)Hand original model. (b)Hand model with 100 watermarked vertices. Embedding power: 0.005.

Model Used	Watermarked Vertices	Embedding Power	EER	SNR
Hand	100	0.001	1.33×10^{-1}	125.56
	100	0.002	1.15×10^{-2}	114.9
	100	0.003	2.93×10^{-4}	109.7
	100	0.004	5.79×10^{-5}	104.74
	100	0.005	4.18×10^{-5}	100.5
Standford Bunny	200	0.001	2.88×10^{-3}	107.22
	200	0.002	3.44×10^{-5}	93.4
	200	0.003	2.88×10^{-6}	90.28
	200	0.004	2.04×10^{-7}	87.57
	200	0.005	9.09×10^{-8}	86.91
	200	0.006	3.27×10^{-8}	86.48
	200	0.007	1.44×10^{-8}	85.8
	200	0.008	7.59×10^{-9}	85.18
	200	0.009	1.82×10^{-9}	84.86
	200	0.01	9.43×10^{-10}	84.28
Klingon	600	0.001	6.11×10^{-5}	128.8
	600	0.002	4.23×10^{-5}	119.02
	600	0.003	2.95×10^{-5}	108.83
	600	0.004	9.81×10^{-7}	103.06
	600	0.005	3.34×10^{-8}	100.2
	600	0.006	3.09×10^{-9}	98.39
Dino	600	0.001	6.02×10^{-9}	117.3
	600	0.002	4.03×10^{-9}	111.28
	600	0.003	1.16×10^{-10}	108.1
	600	0.004	1.61×10^{-10}	106.32
	600	0.005	6.90×10^{-11}	103.54

Table 1. Performance results for the models under examination. The SNR value was measured in dB.

The second model, Stanford Bunny, had 1494 vertices and 2915 triangles. The original model can be seen in Figure 3(a) whereas the watermarked model with embedding power equal to 0.005 is depicted in Figure 3(b). The ROC curves for this model are depicted in Figure 4.

The third model, Klingon, had 5213 vertices and 8887 triangles. The original model can be seen in Figure 5(a) whereas the watermarked model with embedding power equal to 0.004 is depicted in Figure 5(b). The ROC curves for this model are depicted in Figure 6.

The last model, Dino, had 5497 vertices and 10778 triangles. The original model can be seen in Figure 7(a) whereas the watermarked model with embedding power equal to 0.001 is depicted in Figure 7(b). The ROC curves for this model are depicted in Figure 8.

Examination of the detection results and visual inspection of the watermarked mod-

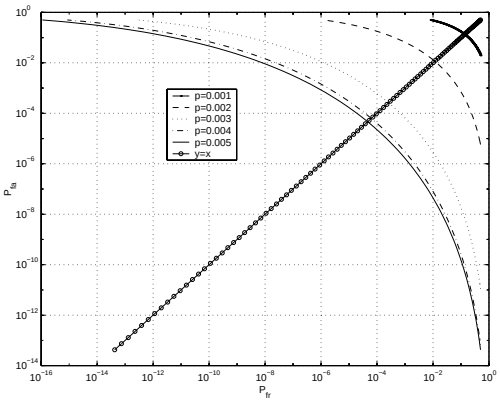
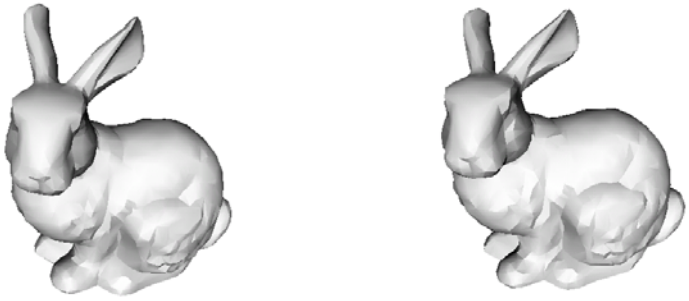


Fig. 2. ROC curves for Hand model with embedding power p from 0.001 to 0.005, for 100 watermarked vertices.



(a) (b)

Fig. 3. (a)Stanford Bunny original model (b)Stanford Bunny model with 200 watermarked vertices. Embedding power: 0.005.

els, prove that the method can indeed achieve very good detection performance and robustness to geometric transformations while maintaining invisibility. Furthermore, the experimental results verify that the detection performance improves when watermarking more vertices or when increasing the embedding power. However, even for a small number of watermarked vertices (e.g. 100) one can achieve low enough EER values (e.g. 10^{-5}) without visible distortions of the model.

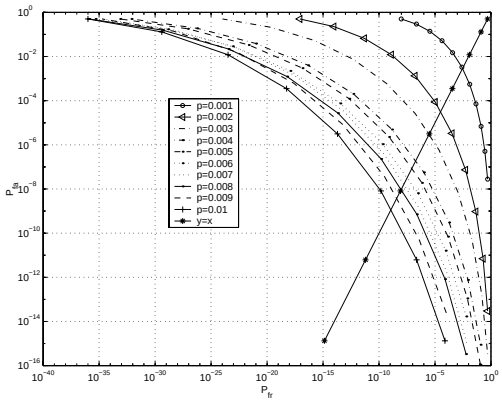


Fig. 4. ROC curves for Stanford Bunny model with embedding power p from 0.001 to 0.01, for 200 watermarked vertices.

(a) (b)

Fig. 5. (a)Klingon original model. (b)Klingon model with 600 watermarked vertices. Embedding power: 0.004.

5 Conclusions

A novel 3D model watermarking method has been presented in this paper. A transform that makes the proposed method completely immune to geometrical attacks like rotation, translation and scaling has been proposed. The algorithm also proved to give good results even for small values of the embedding powers and short watermark lengths. Future work includes the improvement of the algorithm to robustify it against retriangulation, mesh simplification and general affine transformations.

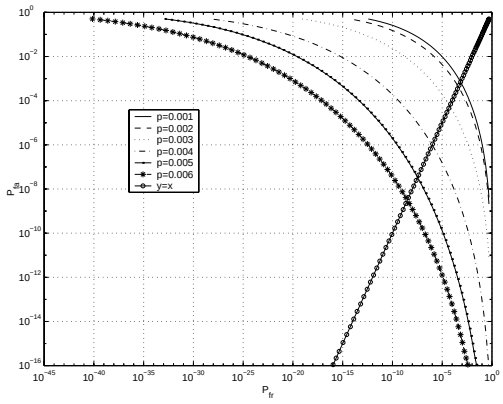
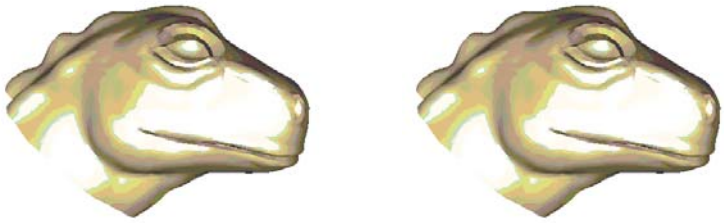


Fig. 6. ROC curves for Klingon model with embedding power p from 0.001 to 0.006, for 600 watermarked vertices.



(a) (b)

Fig. 7. (a)Dino original model. (b)Dino model with 600 watermarked vertices. Embedding power: 0.001.

References

1. F.A.P. Petitcolas, R.J Anderson, and M.G. Kuhn, "Information hiding - a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, July 1999.
2. R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.
3. E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proceedings SIGGRAPH99*, 1999, pp. 69–76.
4. O. Benedens and C. Busch, "Towards blind detection of robust watermarks in polygonal models," in *Proceedings Eurographics2000 Conference*, August 2000, pp. 199–209.

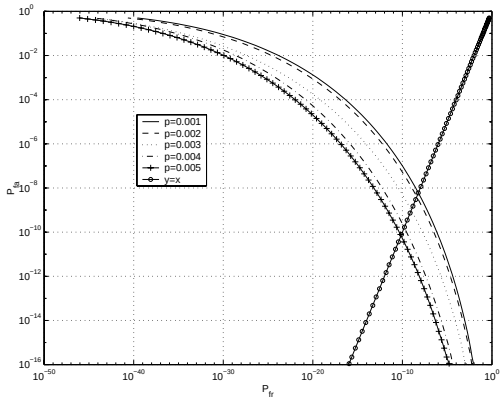


Fig. 8. ROC curves for Dino model with embedding power p from 0.001 to 0.005, for 600 watermarked vertices.

5. A. Tefas and I. Pitas, "Robust spatial image watermarking using progressive detection," in *Proceedings 2001 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, 7-11 May 2001.