A NEW FRAGILE MESH WATERMARKING ALGORITHM FOR AUTHENTICATION

Hao-Tian Wu and Yiu-Ming Cheung
Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

Abstract: In this paper, we propose a new fragile watermarking algorithm based on the global characteristics of the mesh geometry to authenticate 3D mesh models. In our method, a sequence of data bits is adaptively embedded into the mesh model by properly adjusting the vertex positions, and the bit information can be blindly extracted from the watermarked mesh model using a key. The embedding process is adaptive to the mesh model so that the watermarked mesh is perceptually indistinguishable from the original. We show that the embedded watermark is invariant to affine transformation but sensitive to other operations. Besides, the embedding strength is adjustable and can be controlled to a certain extent that even a trivial tampering with the watermarked mesh would lead the watermark signal to change. Therefore, unauthorized modifications of the mesh models can be detected and estimated.

Key words: 3D models; fragile watermarking; mesh authentication; dither modulation

1. INTRODUCTION

While the prevalence of network facilitates people's acquirement and distribution of multimedia works, it also challenges the protection of digital works' copyrights. As a potential technique for copyright protection of digital works, digital watermarking for multimedia data (e.g. digital images, video and audio streams) has been proposed and arduously studied in the literature[1, 2].

Recently, watermarking 3D objects, such as 3D polygonal meshes and various 3D geometric CAD data, has received much attention in the community and considerable progress has been made. In the literature[3-24], a variety of watermarking algorithms have been proposed to embed watermarks into 3D models, mainly 3D polygonal meshes. For instance, the

algorithms[3, 9-16] embed the watermarks by modifying the geometry of the meshes such as vertex coordinates, surface normal distribution, and so forth. They have shown the robustness against some operations to which 3D models are routinely subjected, e.g., affine transformations and mesh simplification. Furthermore, some algorithms[14, 15] modify the topology, i.e. the connectivity, to embed watermarks robust against geometrical operations, but weak to topological modifications. Additionally, some works[16, 24] have used the appearance attributes associated with mesh models to embed the watermarks. In the paper[17], data embedding algorithms for NURBS and other types of parametric curves and faces are also proposed.

To enhance the robustness of 3D watermarking systems, some frequency approaches[18-23] have been recently proposed. In the paper[19, 20], an algorithm that employs multi-resolution wavelet decomposition of polygonal mesh models is presented. Furthermore, the paper[21] proposes an informed watermarking algorithm that constructs a set of scalar basis functions over the mesh vertices, through which the watermark is embedded into the “low frequency" of the polygonal meshes. In the paper[22], Guskov’s multi-resolution signal processing method[27] is adopted and a 3D non-uniform relaxation operator is used to construct a Burt-Adelson pyramid[28] for the mesh to embed watermark information into a suitable coarser mesh. Mesh spectral analysis techniques[26] are also employed to transform the original meshes to the frequency domain and watermark information is embedded into the low frequency of the meshes[18, 23].

Nevertheless, few fragile algorithms[4-8] have been proposed to authenticate the integrity of 3D models. Actually, the first fragile watermarking of 3D objects for verification purpose was addressed by Yeo and Yeung[4], as a 3D version of the approach proposed for 2D image watermarking. Because their algorithm heavily relies on the vertex position, a translation operation, which does not affect the integrity of the mesh model, would easily break the authentication mechanism.

In this paper, we shall present a new fragile watermarking algorithm to authenticate 3D mesh models. Compared to the former methods, our approach makes the embedded watermark invariant to integrity-reserved affine transformation, including translation, rotation and uniformly scaling, but sensitive to other geometrical or topological operations. The rest of this paper is organized as follows. In the following section, a new fragile mesh watermarking algorithm is proposed to authenticate the integrity of 3D mesh models. The experiment results using the proposed method are given in Section 3. Finally, we draw a conclusion in Section 4.

2. A NEW FRAGILE MESH WATERMARKING ALGORITHM

We perform the watermarking process on meshes, which are the "lowest common denominator" of surface representations. It is easy to convert other representations of 3D models to meshes. The mesh geometry can be denoted by a tuple (K, V) [25], where K is a simplical complex specifying the connectivity of the mesh simplices (the adjacency of the vertices, edges, and faces), and $V = \{v_1, \ldots, v_m\}$ is the set of vertex positions defining the shape of the mesh in V^3.

2.1 Extending Dither Modulation to 3D Meshes

We aim to authenticate the integrity of the mesh model, i.e., both the positions and connectivity of vertices need to be verified not having been modified. Our approach extends an implementation of quantization index modulation (QIM) [1] called dither quantization[2] to 3D meshes, and embeds a sequence of data bits by properly adjusting the distances from the faces to the centroid of the mesh geometry.

To extend dither quantization to the mesh model, we choose the quantization step adaptive to the mesh geometry. Suppose $V = \{v_1, \ldots, v_m\}$ is the set of vertex positions in V^3, the position of the mesh centroid is defined by

$$v_c = \frac{1}{m}\sum_{i=1}^{m} v_i \ . \tag{1}$$

The Euclidean distance d_i from a given vertex with the position v_i to the mesh centroid is given by

$$d_i = \sqrt{(v_{ix} - v_{cx})^2 + (v_{iy} - v_{cy})^2 + (v_{iz} - v_{cz})^2} \ , \tag{2}$$

where $\{v_{ix}, v_{iy}, v_{iz}\}$ and $\{v_{cx}, v_{cy}, v_{cz}\}$ are the coordinates of the vertex and the mesh centroid on X-axis, Y-axis and Z-axis, respectively. Using Eq. (2), the furthest vertex with the position v_d to the mesh centroid can be found out and its corresponding distance D is denoted as

$$D = \sqrt{(v_{dx} - v_{cx})^2 + (v_{dy} - v_{cy})^2 + (v_{dz} - v_{cz})^2} \ . \tag{3}$$

We refer to the distance D as the largest dimension of the mesh model and the quantization step S is chosen as

$$S = D / N , \tag{4}$$

where N is a specified value. The distance from a given face to the mesh centroid is defined as the Euclidean distance from the centroid of the face to that of the mesh. Furthermore, the centroid position of a given face f_i with u edges can be obtained by

$$v_{ic} = \frac{1}{u}\sum_{j=1}^{u} v_{ij} , \tag{5}$$

where v_{ij}, $j \in \{1,2,\cdots,u\}$ is the vertex position in the face f_i. The distance d_{fi} from the face f_i to the mesh centroid can be calculated by

$$d_{fi} = \sqrt{(v_{icx} - v_{cx})^2 + (v_{icy} - v_{cy})^2 + (v_{icz} - v_{cz})^2} . \tag{6}$$

Subsequently, we obtain the integer quotient Q_i and the remainder R_i by

$$Q_i = d_{fi} / S , \tag{7}$$

and

$$R_i = d_{fi} \% S . \tag{8}$$

To embed one bit value *w(i)*, we modify the position v_{ic} of the face centroid so that Q_i is an even value for the bit value 0, and an odd value for 1. In order to make $Q_i\%2=w(i)$ always hold meanwhile reducing the false-alarm probability, we modulate the distance d_{fi} according to the bit value in the following way:

$$d_{fi}' = \begin{cases} Q_i \times S + S/2 & if \quad Q_i \% S = w(i) \\ Q_i \times S - S/2 & if \quad Q_i \% S = \overline{w(i)} \,\&\, R_i < S/2 \\ Q_i \times S + 3S/2 & if \quad Q_i \% S = \overline{w(i)} \,\&\, R_i \geq S/2 \end{cases} , \tag{9}$$

where d_{fi}' is the modulated distance from the face f_i to the mesh centroid. Suppose the face f_i consists of u vertices with the centroid position v_{ic}, the position v_{is} of one selected vertex in f_i will be adjusted using d_{fi}' by

$$v_{is}' = (v_c + (v_{ic} - v_c) \times \frac{d_{fi}'}{d_{fi}}) \times u - \sum_{j=1, j \neq s}^{u} v_{ij} \ , \qquad (10)$$

where v_{ij} refers to the vertex position in f_i and v_{is}' is the adjusted position of the selected vertex.

The watermark information embedded in our method is inherently invariant to affine transformations that include any transformation preserving collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation). So the ratio between the distance from each surface face to the mesh centroid and the quantization step, which is proportional to *D*, remains the same after the model is translated, rotated or uniformly scaled. Otherwise, if the mesh model is processed by other operations that change the ratios, the formula $Q_i\%2=w(i)$ will not always hold and the embedded watermark will be changed. Since we need to detect a trivial modification on the mesh model, the integer value Q_i should be sensitive to the distance from the mesh centroid to the surface face. In practice, we assign *N* a large value to obtain a small quantization step *S*. Please note that the precision of the arithmetic operations must be regarded; otherwise, it may increase the false-alarm probability.

The face index of the mesh model is used to represent the connectivity of vertices. If there is any change in the mesh topology, such as mesh decimation or mesh resampling, the face index will be modified and the information hidden in the distances to the mesh centroid would be undermined, therefore the unauthorized modification on mesh topology can be detected.

2.2 The Encoding Process

In this subsection, we will elaborate on how to adjust the mesh surface faces and eventually move them to the desired positions. Please note that faces share edges and vertices with their neighbors, adjusting one face's position may also modify its neighbors' positions. To successfully retrieve the embedded information and preserve the mesh geometry, the centroid position and the largest dimension of the mesh model must remain the same during the encoding process.

Since the position of a given face depends on the coordinates of its vertices, we can lock the face position by locking the coordinates of its vertices. To move one face centroid to the desired position so that one bit of the watermark information is embedded, only one vertex position in the face

need to be adjusted. To avoid the embedded information is changed by the following encoding operations, the vertices in the moved faces need to be marked. Once one watermark bit value is hidden in the distance from the face to the mesh centroid, all the vertices in the face will be marked and their positions can not be modified any more.

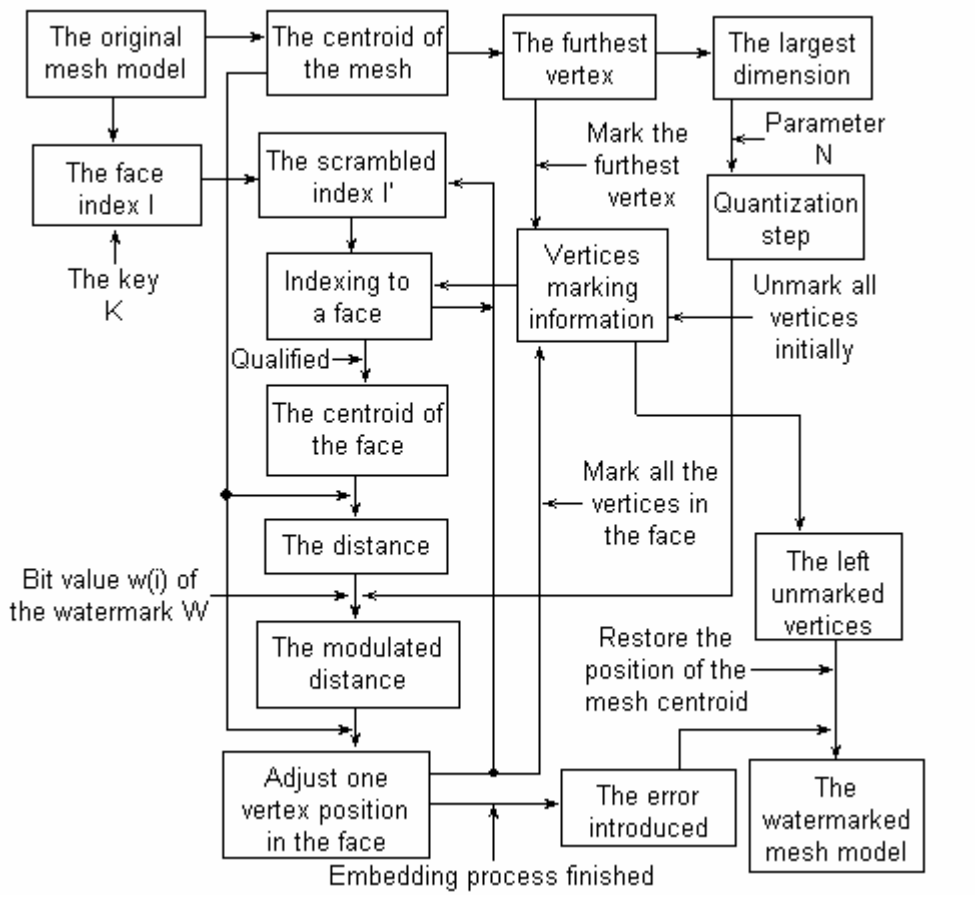

Figure 1. The flow chart of the encoding process

The overall encoding process is as shown in Fig. 1. At first, all vertices in the original mesh are unmarked, the position of mesh centroid is obtained by Eq. (1). Then the furthest vertex to the mesh centroid is found out using Eq. (2) and its corresponding distance to the mesh centroid is calculated using Eq. (3). Since the value of D should not be changed, the furthest vertex is marked and its coordinate will not be modified. The quantization step S can be chosen by specifying the value of N. Using a key, we scramble the face index I to obtain the scrambled index I', which we will follow in the

following encoding process. Before one bit information is embedded in the distance from a face to the mesh centroid, all vertices in the face need to be checked. If there is at least one unmarked vertices in the face f_i, it is qualified to carry one bit value. The distance from the face to the mesh centroid is calculated by Eq. (6) and modulated by Eq. (9) according to the bit value. Noting that the value of D must be maintained in the encoding process, if the modulated distance exceeds it, twice of the quantization step should be subtracted from it so that the embedded bit value is held. Then the coordinate of one unmarked vertex is adjusted using Eq. (10), whereby the face centroid is moved to the desired position. At the end of the embedding operation, all vertices in f_i will be marked. If there is no unmarked vertex in a face, which means the face is not qualified, the checking mechanism will skip to the next face until all watermark information is embedded.

The above embedding process inevitably introduces the distortion of the mesh geometry as some of the vertex coordinates are changed. However, the distortion can be limited to a predefined range, since the elongate or the reduction of the distance from a face to the mesh centroid is no more than twice of the quantization step in the proposed embedding algorithm. The distortion of the mesh geometry also changes the position of the mesh centroid, although adjusting the vertex coordinate may counteract each other. So in the encoding process, not all faces can be used to embed the watermark information. Otherwise, the centroid position of the mesh model will be lost. A small portion of the vertices are needed to restore it after the embedding process. We refer to this process as the centroid restoration process, which modifies the coordinates of the unmarked vertices in the last faces indexed by I' to compensate the error introduced by the embedding process.

The centroid restoration process begins with the calculation of the introduced error E using

$$E = \sum_{j=1}^{m} v_j' - \sum_{j=1}^{m} v_j , \tag{11}$$

where v_j is the original vertex position while v_j' is the adjusted vertex position after the embedding process. Since the value of D should be maintained in the encoding process, the distance from the mesh centroid to the adjusted vertex should not exceed it. So we adjust the unmarked vertices in the centroid restoration process by the following way:

Firstly, we calculate the admissible adjusting radius r_j of an unmarked vertex with the position v_j by

$$r_j = D - \sqrt{(v_{cx} - v_{jx})^2 + (v_{cy} - v_{jy})^2 + (v_{cz} - v_{jz})^2} . \tag{12}$$

Then we use the value of r_j to weight the adjusting vector of each unmarked vertex to ensure that the vertex will not be moved outside its admissible range. Suppose the sum of the unmarked vertices used in the centroid restoration process is L, the individual adjusting weight e_j can be obtained by

$$e_j = E \cdot \frac{r_j}{\sum_{k=1}^{L} r_k}. \tag{13}$$

Subsequently, we subtract the individual adjusting weight from vertex position v_j to restore the position of the mesh centroid by

$$r_j' = r_j - e_j, \tag{14}$$

where v_j' represents the adjusted vertex position after the centroid restoration process and v_j the original one. The encoding process ends as the centroid position of the mesh model is restored.

2.3 The Authentication Process

In the authentication process, only the parameter N, the key K and the original watermark W are needed to authenticate the watermarked mesh geometry. The detailed procedure is shown in Fig. 2.

At first, similar to the encoding process, all the vertices of the original mesh are unmarked, the centroid position v_c' of the suspect mesh geometry is obtained by Eq. (1), which should equal to the centroid position v_c of the original mesh. Then the furthest vertex to the mesh centroid is found using Eq. (2) and its corresponding distance D' is calculated by Eq. (3), which should equal to the largest dimension D of the original mesh model. The quantization step is calculated by $S'=D'/N$ with the provided parameter N. Since the furthest vertex is marked before the embedding process, it should also be marked before the authentication process. Then the face index I of the mesh is scrambled using the key K to produce the scrambled index I'. Before retrieving one bit value from the distance from a given face to the mesh centroid, the vertex marks in the face need to be checked. If there is at least one unmarked vertex in a face f_i', the face will be qualified to extract the embedded bit information and its centroid position v_{ic}' will be calculated using Eq. (6). Then the distance D_{fi} from the face f_i' to the mesh centroid can be calculated by

$$D_{fi} = \sqrt{(v_{icx}' - v_{cx})^2 + (v_{icy}' - v_{cy})^2 + (v_{icz}' - v_{cz})^2}, \tag{15}$$

and the integer quotient Q_i' can be obtained by

$$Q_i' = D_{fi} / S' . \tag{16}$$

The embedded bit information *w'(i)* can be extracted by

$$w'(i) = Q_i' \% 2 . \tag{17}$$

At the end of the extracting operation, all the vertices in f_i' will be marked. If there is no unmarked vertex in a face, no information is extracted and the authentication mechanism will automatically skip to the next face indexed by *I'*. Since the original watermark *W* is known, the extraction process will cease once the extracted bit number matches the embedded bit number.

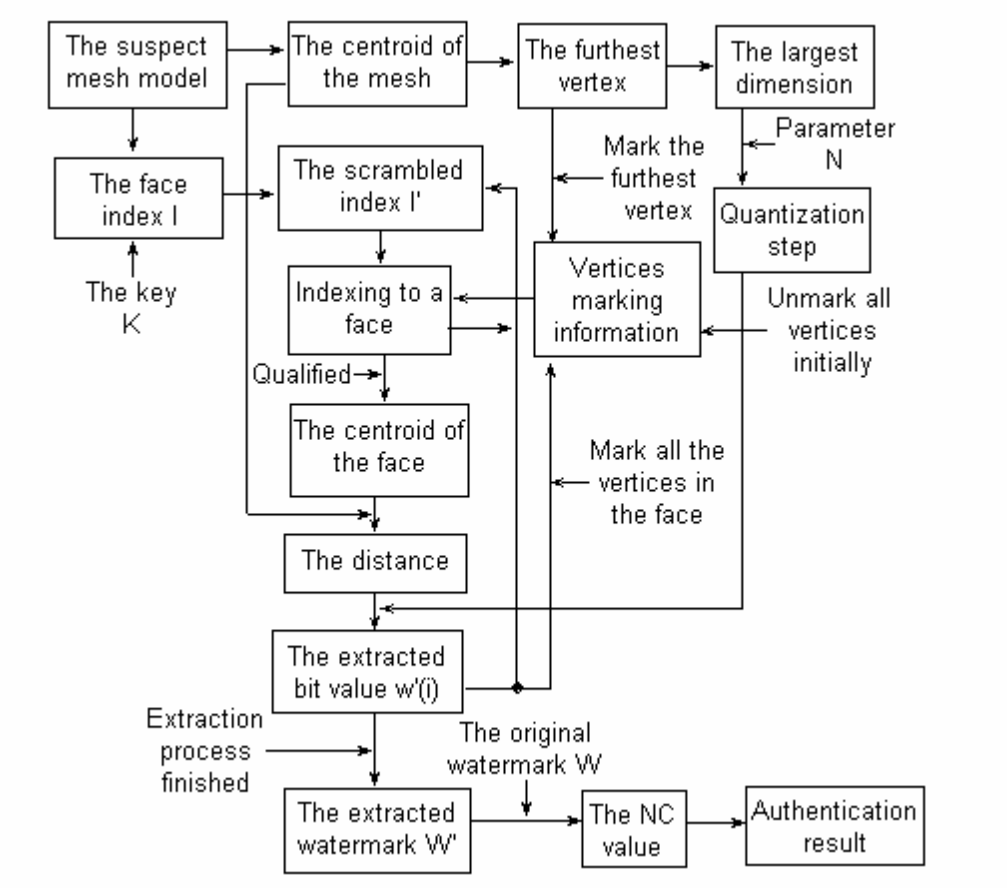

Figure 2. The flow chart of authentication process

After the extraction process, the extracted watermark *W'* is compared with the original watermark *W* using the following cross-correlation function, given their lengths are both identical to *K*:

$$NC = \frac{1}{K}\sum_{i=1}^{K} I(w'(i), w(i)), \quad I(w'(i), w(i)) = \begin{cases} 1 & w'(i) = w(i) \\ 0 & w'(i) \neq w(i) \end{cases}, \quad (18)$$

where *NC* refers to the normalized cross-correlation value between the original and the extracted watermarks. If the watermarked mesh model has not been tampered, the *NC* should be 1; otherwise, it will be less than 1. We claim the mesh geometry as being tampered if the resulting *NC* from Eq. (18) is less than 1.

3. EXPERIMENTAL RESULTS

We have tested the proposed algorithm on several mesh models listed in Table 1 and used a 2D binary image as the watermark. The original mesh model "dog" and its watermarked version are shown in Fig. 4*a* and Fig. 4*b*, respectively. The capacities of the mesh models using the proposed method are also shown in Table 1.

Table 1. The mesh models used in the experiments [*]

Models	vertices	faces	capacity(bits)
dog	7158	13176	4219
wolf	7232	13992	4450
raptor	8171	14568	5695
horse	9988	18363	5731
cat	10361	19098	6149
lion	16652	32096	10992

[*] About 1% vertices of each mesh model are used in the restoration process.

To evaluate the imperceptibility of the embedded watermark using the proposed algorithm, we used the Hausdorff distance between the original and the watermarked mesh models to measure the distortion introduced by the encoding process, upon the fact that the mesh topology is not changed during the watermarking process. Fig. 3 describes the amount of the distortion subject to the parameter of *N*, given that the percent of vertices used for the restoration operation is about 1%. The Hausdorff distance is normalized by the largest dimension *D* of the mesh geometry. From the experimental results, it can be seen that the introduced distortion on mesh model decreases as the parameter *N* is increased.

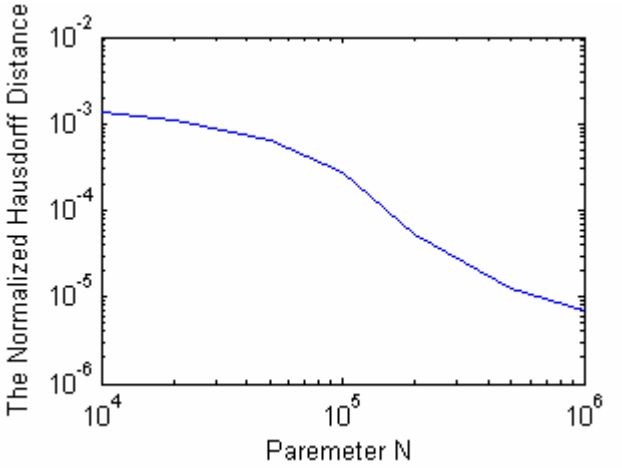

Figure 3. The normalized Hausdorff Distance subject to the parameter N

In our proposed approach, the global characteristics such as the centroid position and the largest dimension of the mesh model are used. If these global characteristics are slightly altered, the watermark information will be dramatically changed and the modifications on the watermarked mesh model can not be located. However, it is easy to locate the tampering if it has little impact on the global characteristics. Since we used a 2D binary image as the watermark, the impact of trivial modifications can be visualized in the extracted watermark image while severe modifications make it meaningless. With the extracted watermark, we can detect the unauthorized modifications and estimate the strength of tampering, if any.

Table 2. The NC values between the original and extracted watermarks *

Models	Affine transformation	changing one vertex position	reducing one face	adding 0.0001% noise	cropping 0.1% faces	extracting without the key
dog	1.0000	0.5375	0.0215	0.6721	0.0133	0.0200
wolf	1.0000	0.9276	0.0685	0.5934	0.0083	0.0029
raptor	1.0000	0.9996	0.0063	0.6972	0.0024	0.0225
horse	1.0000	0.8249	0.0737	0.4661	0.0039	0.0102
cat	1.0000	0.9993	0.0308	0.7072	0.0195	0.0103
lion	1.0000	0.9996	0.0905	0.6363	0.0059	0.0088

* About 1% vertices of each mesh model are used in the restoration process and N=1,000,000.

In the experiments, the watermarked mesh models went through affine transformations (including translation, rotation and uniformly scaling), modifying one vertex coordinate with the vector {*D/500*, *D/500*, *D/500*}, reducing one face from the mesh, adding noise signal that is uniformly distributed within [-*S*,*S*] to all vertex coordinates, and cropping 0.1% faces

of the mesh model. The processed mesh models after these operations are shown in Fig. 4 (from Fig. 4*c* to Fig. 4*g*), respectively. The watermarks are extracted from the processed mesh models with and without the key and the *NC* values between the original and the extracted watermarks are calculated using Eq. (11). The results are listed in the Table 2.

Figure 4a. The original mesh model "dog"

Figure 4b. The watermarked mesh model

Figure 4c. The watermarked mesh model after modifying one vertex position

Figure 4d. The watermarked mesh model after reducing one face

Figure 4e. The watermarked mesh model after adding noise

Figure 4f. The watermarked mesh model after cropping 0.1% faces

Figure 4g. The watermarked mesh model after affine transformations

4. CONCLUSION

In this paper, we have proposed a new fragile mesh watermarking method to authenticate the integrity of 3D mesh model. The watermarking process is conducted in spatial domain and applies to all the mesh models without any restriction. The experimental results have demonstrated that the proposed method is able to imperceptibly and adaptively embed a considerable amount of information into the mesh model, and the embedded watermark can be blindly extracted from the watermarked mesh model to authenticate the watermarked mesh model. In our method, the distortion introduced by the encoding process is quite small and can be controlled within a predefined range. Compared to the previous works, the embedded watermark using our method is invariant to integrity-reserved affine transformation, but sensitive to other processing that alters the mesh model.

Therefore, unauthorized modifications of the mesh models can be successfully detected and estimated.

ACKNOWLEDGEMENT

For the use of the 3D models, we would like to thank the web sources of Department of Computer Science, the University of North Carolina at Chapel Hill, USA. Also, many thanks go to Dr. Zheming Lu for helpful discussions at Harbin Institute of Technology, China.

REFERENCE

1. B. Chen and G. W. Wornell, Quantization index modulation: A class of provably good methods for digital watermarking and information embedding, *IEEE Trans. Inform. Theory*, **47**, 1423-1443 (2001).
2. B. Chen and G. W. Wornell, Digital watermarking and information embedding using dither modulation, *IEEE Second Workshop on Multimedia Signal Processing*, 273-278 (1998).
3. Z. Q. Yu, H. H. S. Ip and L. F. Kwork, A robust watermarking scheme for 3D triangle mesh models, *Pattern Recognition*, **36**(12), 2603-2614 (2003).
4. M. M. Yeung and B. L. Yeo, Fragile watermarking of three dimensional objects, *Proc. 1998 Int'l Conf. Image Processing, ICIP98*, **2**, 442-446 (IEEE Computer Society, 1998).
5. B. L. Yeo and M. M. Yeung, Watermarking 3D objects for verification, *IEEE Comput. Graph. Applicat*, 36-45 (1999).
6. F. Cayre and B. Macq, Data hiding on 3D triangle meshes, *IEEE Trans. Signal. Processing*, **51**(4), 939-949 (2003).
7. HsuehYi Lin, Hongyuan Mark Liao, ChunShien Lu and JaChen Lin, Fragile Watermarking for Authenticating 3D Polygonal Meshes, *Proc. 16th IPPR Conf on CVGIP*, 298-304 (2003).
8. C. Fornaro and A. Sanna, Public Key Watermarking for Authentication of CSG Models, *Computer-Aided Design*, **32**, 727-735 (2000).
9. O. Benedens, Watermarking of 3D polygon based models with robustness against mesh simplification, *Proc. SPIE: Security Watermarking Multimedia Contents*, 329-340 (1999).
10. O. Benedens, Geometry based watermarking of 3D models, *IEEE Comput. Graph., Special Issue on Image Security*, 46-55, Jan./Feb. 1999.
11. O. Benedens, Two high capacity methods for embedding public watermarks into 3D polygonal models, *Proc. Multimedia Security Workshop ACM Multimedia*, 95-99 (1999).
12. O. Benedens and C. Busch, Toward blind detection of robust watermarks in polygonal models, *Proc. EUROGRAPHICS Comput. Graph. Forum*, **19**(C), 199-208 (2000).
13. M. G. Wagner, Robust watermarking of polygonal meshes, *Proc. Geometric Modeling Processing*, Hong Kong, 201-208 (2001).
14. R. Ohbuchi, H. Masuda and M. Aono, Watermarking Three Dimensional Polygonal Models, *Proc. ACM Multimedia*, Seattle, 261-272 (1997).
15. R. Ohbuchi, H. Masuda and M. Aono, Watermarking Three Dimensional Polygonal Models Through Geometric and Topological Modifications, *IEEE J. Select. Areas Commun.*, **16**, 551-560 (1998).

16. R. Ohbuchi, H. Masuda and M. Aono, Geometrical and Non-geometrical Targets for Data Embedding in Three Dimensional Polygonal Models, *Computer Communications*, Elsevier, **21**, 1344-1354 (1998).
17. R. Ohbuchi, H. Masuda and M. Aono, A shape preserving data embedding algorithm for NURBS curves and surfaces, *Proc. Comput. Graph. Int.*, June, 1999.
18. R. Ohbuchi, S. Takahashi, T. Miyasawa and A. Mukaiyama, Watermarking 3D polygonal meshes in the mesh spectral domain, *Proc.Graphics Interface*, Ottawa, 9-17 (2001).
19. H. Date, S. Kanai and T. Kishinami, Digital watermarking for 3D polygonal model based on wavelet transform, *Proc. ASME Des. Eng. Techn. Conf.*, Sept 1999.
20. S. Kanai, H. Date and T. Kishinami, Digital watermarking for 3D polygons using multi-resolution wavelet decomposition, *Proc. Sixth Int.Workshop Geometric Modeling: Fundamentals Applicat.*, Sept 1998.
21. E. Praun, H. Hoppe and A. Finkelstein, Robust mesh watermarking, *Proc. SIGGRAPH*, 69-76 (1999).
22. Kangkang Yin, Zhigeng Pan, Jiaoying Shi and David Zhang, Robust mesh watermarking based on multi-resolution processing, *Computers & Graphics*, **25**, 409-420 (2001).
23. F. Cayre, P. RondaoAlface, F. Schmitt, B. Macq and H. Maitre, Application of Spectral Decomposition to Compression and Watermarking of 3D Triangle Mesh Geometry, *Signal Processing: Image Communications*, **18**(4), 309-319 (2003).
24. Liangjun Zhang, Ruofeng Tong, Feiqi Su and Jinxiang Dong, A Mesh Watermarking Approach for Appearance Attributes, *Pacific Conference on Computer Graphics and Applications*, Beijing, 450-451 (2002).
25. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, Mesh optimization, *Computer Graphics (SIGGRAPH '93 Proceedings)*, 19-26 (1993).
26. Z. Karni and C. Gotsman, Spectral compression of mesh geometry, *Proc. SIGGRAPH*, 279-286 (2000).
27. I. Guskov, W. Sweldens and P. Schroeder, Multi-resolution signal processing for meshes, *Proc. SIGGRAPH*, 325-334 (1999).
28. P. J. Burt and E. H. Adelson, Laplacian pyramid as a compact image code, *IEEE Transactions on Communications*, 532-540(1983).