A Public Mesh Watermarking Algorithm Based on Addition Property of Fourier Transform

Li Li[a], Zhigeng Pan[b], and David Zhang[c]
[a]*Hangzhou Dianzi University,Hangzhou 310018, P.R.China*
[b]*State Key Lab of CAD&CG,Zhejiang University, Hangzhou 310027, P.R.China*
[c]*Center for Multimedia Signal Processing and Department of Computing,Hong Kong Polytechnic University, Kowloon, Hong Kong*
(E-mail: lili@cad.zju.edu.cn)

Abstract

This paper presents a public mesh watermarking algorithm. First, the resultant watermarked image minus the original image is the watermark information. According to the addition property of the Fourier transform, a change of spatial domain will cause a change in the frequency domain. Then, the watermark information is scaled down and embedded in one part of the *x*-coordinate of the original mesh. Finally, the *x*-coordinate of the test mesh is amplified before extraction. Experimental results prove that our algorithm is resistant to a variety of attacks without the need for any preprocessing.

Keywords: Mesh watermarking; Fourier transform; Log Polar Map.

1. Introduction

As more and more CAD-based 3D data are entering the World Wide Web, it is inevitable that companies or copyright owners who present or sell their products in the virtual space will face copyright-related problems. One promising area of research in the search for a solution to this issue is digital watermarking. Indeed, a number of studies in the area of useful image watermarking systems have already been developed [1-6].

In comparison to image data, it is difficult to watermark 3D mesh models [7]. While traditional media has functions or regularly sampled signals that are defined on a flat geometry, there is no unique representation of model data. Arbitrary meshes also lack a natural parameterization for frequency-based signal processing, and thus they cannot use the corresponding image watermarking algorithms that are based on the frequency domain, such as the wavelet transforms and the Fourier transforms.

As a result, many 3D mesh watermarking algorithms are private and need some processing before their watermarks can be extracted. Benedens [7] proposes embedding private watermarks. The system primarily achieves robustness against randomization of points, mesh altering (re-meshing) operations or attacks and polygon simplification. Yin, *et al.* [8] reported an informed-detection, robust mesh-watermarking algorithm that worked in a transformed domain based on the work proposed by Guskov, *et al* [9]. However, registration and re-sampling are needed to bring the attacked mesh model back into its original location, orientation, scale, topology and resolution level when the watermarked mesh is cropped or similarity transformation is performed. Recently, Ohbuchi, *et al.* [10] presented an algorithm based on their previous works [11] in an attempt to watermark much larger meshes within a reasonable time. Although the watermarking is more robust against connectivity alteration, cropping, mesh simplification and smoothing, there was room for improvement. For example, watermark extraction requires a lot of information such as the original mesh, the patch key and the information that uniquely determines the partitioning of the original mesh. Mesh alignment and re-meshing are also needed during watermark extraction.

There are few public mesh watermarking algorithms. Kanai, *et al.* [12] applied a transformed-domain watermarking approach to 3D meshes - a robust, blind-detection watermarking algorithm that works in the mesh's wavelet-transformed domain.

These watermarks are resistant to affine transformation, partial resection and random noise added to the vertex coordinates, among other attacks. However, the method requires the mesh to have 1-to-4-subdivision connectivity.

Praun and Hoppe [13] reported an informed-detection and robust mesh watermarking algorithm that works in a transformed domain but is applicable to polygonal meshes having arbitrary vertex connectivity.

From papers mentioned above, we can express their watermark insertion process of 3D mesh as a single equation $v' = v + B(v, w)$, v' are the coordinates of watermarked mesh vertices, v are the coordinates of original mesh vertices. w is the watermark. B is a function matrix decided by those algorithms. Therefore, we outline our mesh watermark insertion process as following equations: $v' = v + \alpha * (I_w - I)$, I_w is the watermarked image and I is the original image. α is a scale parameter. In our algorithm, we do not embed watermark into the coordinate of 3D mesh directly, but into a gray image by a watermarking algorithm based on the work of O'Ruanaidh and Pun [6]. Take advantage of the image watermarking algorithm and the addition property of Fourier transform, we propose a blind mesh watermarking algorithm. Watermark can be extracted without original mesh or any preprocessing. Furthermore, computational complexity is low.

2. Description of the Algorithm

2.1. Watermark Embedding

Our watermarking algorithm embeds information into a mesh in the four main steps outlined below (Fig. 1). The yellow arrows in Fig. 1 indicate steps by which the watermark information is obtained from the image watermarking algorithm. The black arrows indicate steps in which the watermark information is embedded into the original mesh.

(1) Encoding the watermark

Watermarks are meaningful characters and every character is encoded into a sequence in which the total number of digital "*0*" is 5 and the total number of digital "*1*" is 1.

(2) Sorting the x-coordinate of the mesh into an ascending order

Reorder the *x*-coordinate of the original mesh according to the size to get a new *x*-coordinate with good smoothness. Thus, watermark can be regarded as noise after the new *x*-coordinate is watermarked. We can scale the noise of the *x*-coordinate of the watermarked mesh during the algorithm.

(3) Using the image watermarking algorithm to obtain the watermark information

- Embed a gray image $I(X,Y)$ of 64*64 using the steps, X=1,2… 64; Y=1,2…64. The watermark W is "1234". The image watermarking algorithm is based on a Fast Fourier transform (FFT) and a Log Polar Map (LPM) that can produce the invariants of rotation, scale and translation (RST).
- Encode W into a binary form using the method mentioned above. $W = (w_1, w_2, ..., w_{24})$, where $w_i \in \{0,1\}$.
- Take Fourier Transform of $I(X,Y)$ and get Fourier magnitude $B(X,Y)$. X={1,2… 64}; Y={1,2…64}.
- Select middle spectrum of $B(X,Y)$ to form a traffic circle $C(x, y)$ and resample $C(x, y)$ into log-polar coordinates by bilinear interpolation. $x \in S_1$, $y \in S_2$. $S_1 \subset X$, $S_2 \subset Y$. (ρ, θ) denotes log-polar coordinate system; (x, y) denotes Cartesian coordinate system. Eg. (1) shows that Cartesian coordinate system is transformed into log-polar coordinate system.

$$\begin{cases} x = e^{\rho * \Delta\rho} \cos(\theta * \Delta\theta) \\ y = e^{\rho * \Delta\rho} \sin(\theta * \Delta\theta) \end{cases} \quad (1)$$

where $\Delta\rho = 15/64; \Delta\theta = 2\pi/64;$ $\rho = 1,2,...,64; \theta = 1,2,...,64$.

- Take Fourier Transform of $C(\rho, \theta)$ and obtain Fourier magnitude $D(i, j)$. i=1,2,…,64; j=1,2,…,64.
- Embed w_i in $D(i, j)$ and get $D^*(i, j)$. When i =5, 7, -5, -7; j=1, 2…6, w_i =1,d=10^8; w_i =0,d=0.

$$D^*((i+32),(12+j)) = D((i+32),(12+j)) + d \quad (2)$$

- Invert FFT transform, LPM transform and FFT transform to D^* respectively and obtain the watermarked image I^*.
- Get the watermarking information W^* in spatial domain by the Eq. (3). W^* is too weak to be visible in the spatial domain.

$$W^*(i,j) = (I^*(i,j) - I(i,j)) \times (\frac{1}{500}) \tag{3}$$

where i=1,…,64; j=1,…,64.

(4) Adding the watermark information to the new sorted x-coordinate of the original mesh

- Select one part of the new *x-coordinate* of the original mesh *M* to form a matrix *A* of size 64*64. The matrix A is regarded as a function defined on grid [64, 64]. Add watermarking information W^* to *A* and obtain A^* thereby producing the watermarked mesh M^*.

  F is the Discrete Fourier Transform. The addition property of the Fourier transform is defined as follows:

$$F[f(x_1,x_2)+g(x_1,x_2)] = F[f(x_1,x_2)] + F[g(x_1,x_2)] \tag{4}$$

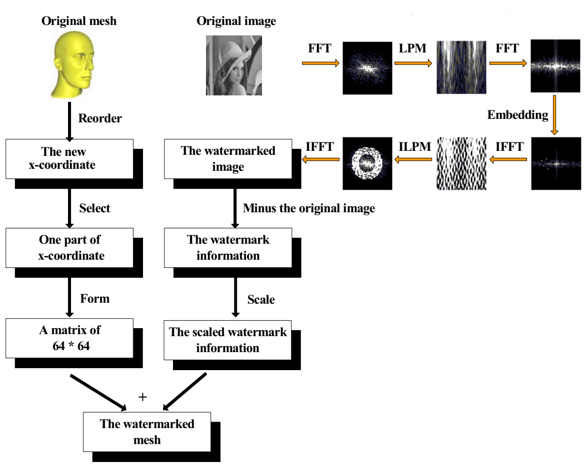

Fig. 1 The framework for embedding watermarks. The binary images are the corresponding magnitude images that are mapped into [0,255] to be visible.

2.2. Watermark Extraction

As a public watermark, extraction requires neither the original nor the *reference mesh,* nor mesh alignment and re-meshing. Fig. 3 shows the framework for extracting watermarks.

(1) Amplifying the watermark information of the test mesh

Since the energy of the watermark is weaker than that of the original mesh, the watermark information is too difficult to detect. Obviously, in watermark extraction it is useful to amplify such weak information.

(2) Recovering the watermark information

- Select the corresponding part of the *x-coordinate* of the test mesh M^* to form a matrix *A* of size 64*64.
- Take a Fourier Transform of *A* and get the Fourier magnitude *B*.
- Select the middle spectrum of *B* and get the traffic circle *C* and resample *C* into log-polar coordinates using bilinear interpolation.
- Take a Fourier Transform of *C* and get the Fourier magnitude *D*.
- Select the corresponding data in *D* and get a sequence that consists of 24 data. Divide them into four groups and decode every group to one character according to the encoding method mentioned above.
- Four characters are the watermark extracted.

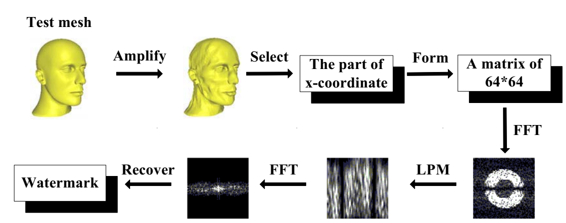

Fig. 3 The framework for extracting watermarks. The binary images are the corresponding magnitude images that are mapped into [0,255] to be visible.

3. Experimental Results

The algorithm described above is implemented in C++. The Watermark is "2541", which is encoded into a 24-bit binary sequence. For example, a Dinosaur model (10,002 vertices, 20,000 faces) and a Venus head model (15,002 vertices, 30,000 faces).

These attacks include noise addition, smoothing, enhancement, rotation, translation, scaling, re-sampling, cropping and some combinations of them. Figs. 4-5 show some experimental results.

Our filtering, Re-sampling attack and enhancement of the watermarked mesh are done under the framework of mesh spherical parameterization and spherical harmonic transformation [14]. When the sampling rate is 512*512 or 256*256, the watermark still can be extracted exactly.

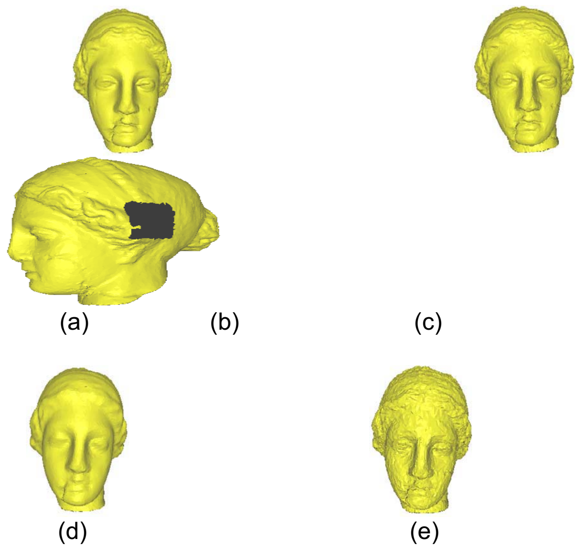

Fig. 4 Some experimental results of attacks to the Venus head model. (a) Venus head model, (b) Watermarked model, (c) Attack B, (d) Attack K, (e) Attack A.

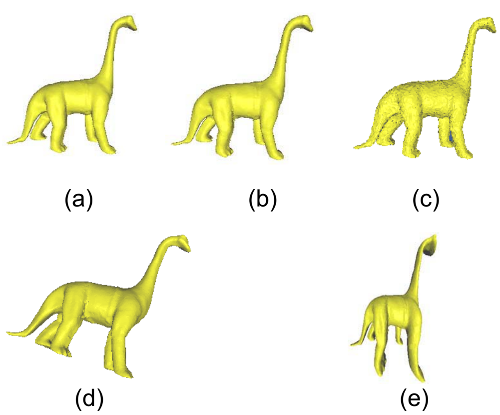

Fig.5 Some experimental results of the Dinosaur model. (a) Dinosaur model, (b) Watermarked model, (c) Attack O, (d) Attack E, (e) Attack I.

4. Conclusions

Meshes can be watermarked within a short time of less than 60 seconds; No original mesh is needed to extracting the watermarks; Our algorithm has proven to be resistant to a variety of attacks .

Acknowledgement

This research is co-supported by Natinoa Natural Science Foundation and the Excellebt Young Teacher Program of MOE, PRC.

References

[1] I.J.Cox and M. L. Miller, "A review of watermarking and the importance of perceptual modeling", *Proc. SPIE. Conf. Human Vision and Electronic Imaging II*, 1997,pp. 92-99.
[2] S. Pereira, J. J. K. O Ruanaidh, F. Deguillaume, G. Csurka , and T. Pun, "Template based recovery of Fourier-based watermarks using Log-polar and Log-log maps", *IEEE Int. Conf on Multimedia Computing and Systems*, Florence, Italy, June,1999, pp. 07-11.
[3] Shusen Sun, Zhigeng Pan, Li Li, and Jiaoying Shi, "Robust 3D model watermarking against geometric transformation", *CAD/CG'2003*, Macao, Oct.29-31, pp. 87-92.
[4] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack modelling: Towards a second generation watermarking benchmark", *Signal Processing*, June, 2001, pp. 1177 – 1214.
[5] Licks and R. Jordan, "Geometric transformation robust watermark", *IEEE International Conference on Image Processing*, Vancounver, Canada, 2000.
[6] J.J.K.O Ruanaidh, and T.Pun. Rotation, "Translation and Scale Invariant Sprectrum Digital Image Watermarking, Signal Processing", *Special Issue On Copyright Protection and Control*, 66(3), 1998, pp. 303-317.
[7] O. Benedens, "Geometry-based watermarking of 3D models", *IEEE Computers and Applications*, 19(1), 1999, pp. 46-55.
[8] K.K. Yin, Z.G. Pan, J. Y. Shi , and D. Zhang, "Robust mesh watermarking based on multiresolution processing", *Computers & Graphics*, 25, 2001, pp. 409-420.
[9] I. Guskov, W. Sweldens, and P. Shröder, "Multiresolution signal processing for meshes", *Proc. SIGGRAPH* , 1999, pp. 49-56.
[10] R. Ohbuchi, A.Mukaiyama , and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes", *EUROGRAPHICS 2002,* Saarbrucken, Germany, September , 2002, pp. 2-6.
[11] R. Ohbuchi, S. Takahashi, T. Miyazawa , and A. Mukaiyama, "Watermarking 3D polygonal meshes in the mesh spectral domain", *Proc. of the Graphics Interface*, Canada, 2001, pp. 9-17.
[12] S. Kanai, H. Date , and T. Kishinami, "Digital watermarking for 3D polygons using multi-resolution wavelet decomposition", *Proc. of the Sixth IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications* (GEO-6), Japan, 1998, pp. 296-307.
[13] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking", *In Proc. of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 1999, pp. 49--56.

[14] K. Zhou, H. J. Bao , and J. Y. Shi, "Uniform digital geometry processing", *Journal of Computer,* 25(9), 2002, pp.904-909.