

A Robust Digital Watermarking Algorithm for Three Dimensional Meshes

Geri Bo

City Institute Dalian University of Technology, Dalian, 116600, China

ececity@dlut.edu.cn

Keywords: Watermarking; DFT; Affine transformation; Robust

Abstract. First, the algorithm converts 3d models into an affine invariant space, in order to obtain rotation in-variance and scale in-variance of the model; then we transform the distance of the grid point to the center to a signal in frequency domain using one-dimensional DFT. We embed a watermark into the signal in frequency domain, and then transform it back to a digital signal in spatial domain. Finally, according to the value of the new spatial signal, we modify the coordinates of points on the grid to get the digital watermark grid. Experimental results show that the algorithm can resist similar mesh simplification, plus noise and shearing attacks, and has good robustness.

Introduction

With CAD, CAM, CAE and other advanced manufacturing technology and the development of virtual reality technology, and networks are widely used in these areas. Three-dimensional model as a type of digital products, there are also issues of copyright protection. Adding watermark information in the 3D model to effectively protect 3D model's copyright information has become a research direction.

This paper proposes a digital watermarking algorithm in spatial domain with high robustness and relatively simple calculations. First, we should transform the 3D model into geometric invariant space. Secondly, by using DFT transform technique to modify the distance from the center to the vertex of 3D model is to embed the watermark information. Finally, the embedded watermark geometric invariant space model changes back to the original geometric space. In this paper, a large number of experiments carried out quantitative validation, introduction of several common attacks robust experimental results show that the digital watermarking algorithm in this paper is invisible on the common attacks robust. In this paper, through a large number of experimental quantitative verification, introduce several common attack experiments in robustness, results show that the proposed watermarking embedding algorithm is invisible and robust against common attacks.

The Basic Principle of the Algorithm

According to different 3D model modeling method, the three-dimensional model of digital watermark can be basically divided into grid watermark, parametric surface watermark and body data watermark, etc. In all types of three-dimensional model watermarking technology, the grid watermark technology is the direction of most studied. Triangular mesh representation is the most common representation of three-dimensional model representation. Other three-dimensional model representation can be converted into the triangle mesh representation, therefore, to study the watermarking technology through the triangular mesh representation. Triangle mesh representation has some of the same or substantially the same global geometric features, such as the size of the 3D model, surface curvature and the distance of each point on the surface to the center etc. To make the watermark embedded in the 3D solid model, you must choose a global geometric feature as an embedded object. In this paper, the algorithm selects the global geometric features of the distance of the surface point to the center as an embedded object.

Preprocessing Affine Invariance of 3D Model

Three-dimensional model consists of a vertex set $p = \{p_i\}$ and a collection of C which has linking relationships. Each vertex p_i has three-dimensional coordinates, i.e., $p_i = (x_i, y_i, z_i)$. The purpose of 3D model transformation is to transfer three dimensional data to one dimension signal for embedding watermarking. The 3D model obtained by affine transformation with rotation, scale transformation and translation invariant features. Specific process is as follows:

(1) Calculating all vertexes' the center of mass. The center of mass is defined as follows:

$$\mu = \frac{1}{k} \sum_{i=0}^{k-1} p_i = (\mu_x, \mu_y, \mu_z) \quad (1)$$

(2) Model translation. $p_i = (x_i, y_i, z_i)$ obtained by subtracting the center of mass:

$$p'_i = (x'_i, y'_i, z'_i) = (x_i - \mu_x, y_i - \mu_y, z_i - \mu_z) \quad (2)$$

As a result, the new vertex coordinates are translation invariant.

(3) Principal component calculation [1]. The main ingredient is a vertex feature vector T , and the feature vector corresponds to the largest eigenvalue of the co-variance matrix of vertex coordinates. Among them, the co-variance matrix represented by the following formula:

$$H = \begin{bmatrix} \sum_{i=0}^{k-1} x_i'^2 & \sum_{i=0}^{k-1} y_i'x_i' & \sum_{i=0}^{k-1} z_i'x_i' \\ \sum_{i=0}^{k-1} x_i'y_i' & \sum_{i=0}^{k-1} y_i'^2 & \sum_{i=0}^{k-1} z_i'y_i' \\ \sum_{i=0}^{k-1} x_i'z_i' & \sum_{i=0}^{k-1} y_i'z_i' & \sum_{i=0}^{k-1} z_i'^2 \end{bmatrix} \quad (3)$$

(4) Model rotation. Rotate the model so that a feature vector T coincides with the z-axis, in order to achieve rotation in-variance.

The Watermark Embedded

Assuming that the original mesh model is $M_0 = \{V_0, C_0\}$, M_0 composed of the vertex set V_0 and a collection of C_0 which has a link relationship, and the set contains N vertexes, and every coordinate value of vertex is determined by (x_{0i}, y_{0i}, z_{0i}) , that is, $V_0 = \{v_{0i}\}, i = 0, 1, \dots, N-1$. And the collection C_0 represents $C_0 = \{\{i_k, j_k\}\}, k = 0, 1, \dots, m-1, 0 \leq i_k \leq N-1, 0 \leq j_k \leq N-1$. Specifically watermark embedding process is as follows:

(1) Pre-treatment of the model.

(2) Input the watermark information to be embedded or the key K into the pseudo random sequence generator, and generate the corresponding binary watermark sequence $W = \{w_l\}$, N_w is the length of the watermark sequence.

(3) Rowing the grid vertex into a one-dimensional sequence of vertexes, and transform into spherical x_i, y_i, z_i coordinates $\theta_i, \phi_i, \gamma_i$, the relationship between them is shown in Figure 1.

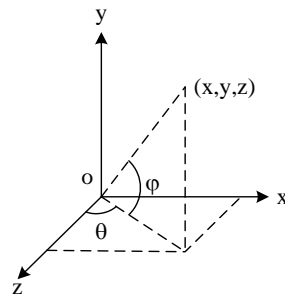


Fig.1. Relationship between x, y, z and r, θ, ϕ

$$\begin{aligned}
r_i &= \sqrt{x_i^2 + y_i^2 + z_i^2} \\
\theta_i &= \begin{cases} \arccos(z_i/\sqrt{x_i^2 + z_i^2}), & x_i \geq 0 \\ 2\pi - \arccos(z_i/\sqrt{x_i^2 + z_i^2}), & x_i < 0 \end{cases} \\
\phi_i &= \begin{cases} \arccos(\sqrt{x_i^2 + z_i^2}/r_i), & y_i \geq 0 \\ -\arccos(\sqrt{x_i^2 + z_i^2}/r_i), & y_i < 0 \end{cases}
\end{aligned} \tag{4}$$

(4) r_i Obtained in the previous step is arranged according to certain order, and order takes $S \times N_w$ elements, and is divided into N_w groups, then each group contains S elements, that is $R = \{r_{lj}\}, l = 0, 1, \dots, N_w, j = 0, 1, \dots, S-1$. Each group r_{lj} is as an embedding unit to embed a watermark information. Embedding method as follows:

DFT transform is performed on a set of r_{lj} to get the corresponding DFT coefficient R_{lj} . Using the formula (5), the l bit of the watermark information w_l is embedded into the DFT coefficients R_{lj} .

$$|R_{lj}^w| = |R_{lj}| + \alpha \varepsilon_j w_l \tag{5}$$

Among them, α is embedding strength coefficient, and it controls the total energy of the embedded watermark signal; And $\varepsilon = (\varepsilon_0, \varepsilon_1, \dots, \varepsilon_j)$ is $(0, 1)$ uniformly distributed random numbers used for modulation. This is important to note that in the embedded watermark can't select dc component R_{l0} and coefficient $R_{s/2}^l$. Then IDFT of R_{lj}^w to get the r_{lj}^w that contains the watermark information.

(5) Inverse sort on r_{lj}^w to get r_i^w , Using equation (6) can be obtained watermarked middle transition vertex coordinates:

$$\begin{aligned}
x'_i &= r_i^w \times \cos \phi_i \times \sin \theta_i \\
y'_i &= r_i^w \times \sin \phi_i \\
z'_i &= r_i^w \times \cos \phi_i \times \cos \theta_i
\end{aligned} \tag{6}$$

(6) Vertex coordinate (x'_i, y'_i, z'_i) obtained above are pre-treated as 2.1 inverse transformation, then obtains the final watermarked vertexes (x_i^w, y_i^w, z_i^w) .

The Watermark Extraction

(1) Using model pre-processing, the original model M_o and the detected model M_d is of the same transformation respectively, after transformation vertex coordinates as: $v_{oi} = \{(x_{oi}, y_{oi}, z_{oi})\}$,

$$v_{di} = \{(x_{di}, y_{di}, z_{di})\}.$$

(2) Because some of the attacks will change the original mesh representation, such as simplifying the attack, cropping attack and re-gridding etc. the watermark detection algorithm can't extract the watermark information directly from the attack model. In response to this situation, the use of re-sampling to restore the original mesh representation of model. Re-sampling process is as follows[2]: from the center of the original model elicits a ray to the vertex v_{oi} and intersects with a three-dimensional model to be tested. If there is more than one intersection points v_{di} and which are the nearest intersection of v_{oi} , v_{di} and v_{oi} is considered a matching point, otherwise take $v_{di} = v_{oi}$.

(3) Using the previous coordinate conversion formula (4), (x_{oi}, y_{oi}, z_{oi}) , (x_{di}, y_{di}, z_{di}) are respectively converted to $(r_{oi}, \varphi_{oi}, \theta_{oi})$, $(r_{di}, \varphi_{di}, \theta_{di})$. Similarly, the obtained r_{oi} , r_{di} are sorted and

grouped in the same order of watermark embedding, namely: $R_{ol} = \{r_{olj}\}, R_{dl} = \{r_{dlj}\}, l = 0, 1, \dots, N_w;$
 $j = 0, 1, \dots, S - 1.$

(4) Calculating a difference between each set of the DFT coefficients R_{dlj} and R_{olj} :

$$\Delta_{lj} = R_{dlj} - R_{olj} \quad (7)$$

(5) Using formula (8) to extract the watermark sequence:

$$W_l^d = \text{sign}\left(\sum_{j=0}^{s-1} \Delta_{lj}\right) \quad l=0, 1, \dots, N_w-1 \quad (8)$$

(6) Correlation calculation . The correlation coefficient C_{or} is defined as:

$$Cor = \frac{\sum_{i=0}^{N_w-1} (w_i^d - \bar{W}^d)(w_i - \bar{W})}{\sqrt{\sum_{i=0}^{N_w-1} (w_i^d - \bar{W}^d)^2 (w_i - \bar{W})^2}} \quad (9)$$

W^d of equation (9) is extracted watermark sequence , W is the original watermark sequence , \bar{W}^d is the average of the various in the sequence W^d , \bar{W} is the average of the sequence W of each one , N_w is the length of the watermark sequence.

The Experimental Results

In order to verify the algorithm in this paper, this experiment uses a rabbit 3D model to test, the model consists of 1751 vertexes and 2999 triangular faces.



Fig.2. Original model

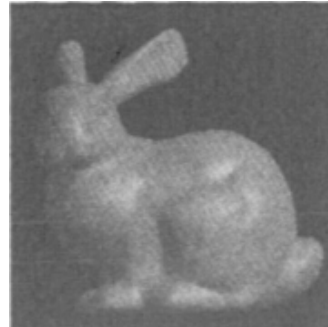


Fig.3. Add watermarks model

In the experiment, the original watermark sequence length N_w is 32, the number of vertexes S of each set is 48, the embedding strength A is 0.04. Figure 2 and Figure 3 are the original 3D model and the embedded watermark model. Comparing Figure 2 and Figure 3 show that the watermarking algorithm satisfies the conditions of watermark invisibility. Testing the robustness of the algorithm, respectively tested it with simplified attack, noise attack and shear attacks.

When testing the ability of this algorithm to resist the simplify attack, taking the StanMelax algorithm to simplify the 3D model of embedded watermark, the experimental results are shown in Table 1 and Figure 4. As can be seen from Table 1, even if the attack model has only a few vertexes, the algorithm still showed strong robustness.

Tab.1. The results of simplifies attack

The remaining number of vertexes	1178	785	392
Simplified rate	25%	50%	75%
Cor	1.0	0.9393	0.6888

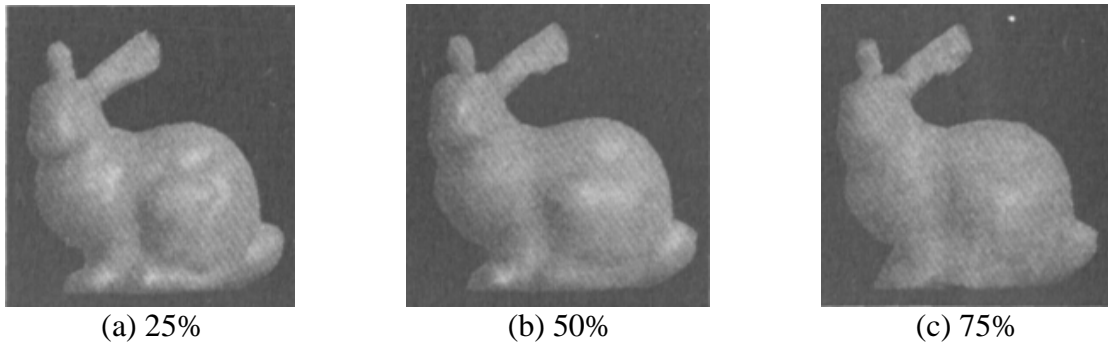


Fig.4. Simplifies attack

Noise attack experiments, added uniform noise to the vertex coordinates that have been added the watermark model. Joins a noise vector to each vertex of model that have been added the watermark. Noise vector direction along the grid vertex to the origin of attachment, the magnitude of the noise is defined as the ratio of the length of the noise vector and the distance of the grid vertex to the grid center , which reflects the strength of the noise. Noise attack experiment results are shown in Table 2, mesh model added noise is shown in Figure 5.

Tab.2. The results of noise attack

Noise amplitude	0.005	0.01	0.015
Cor	0.9393	0.875	0.7559

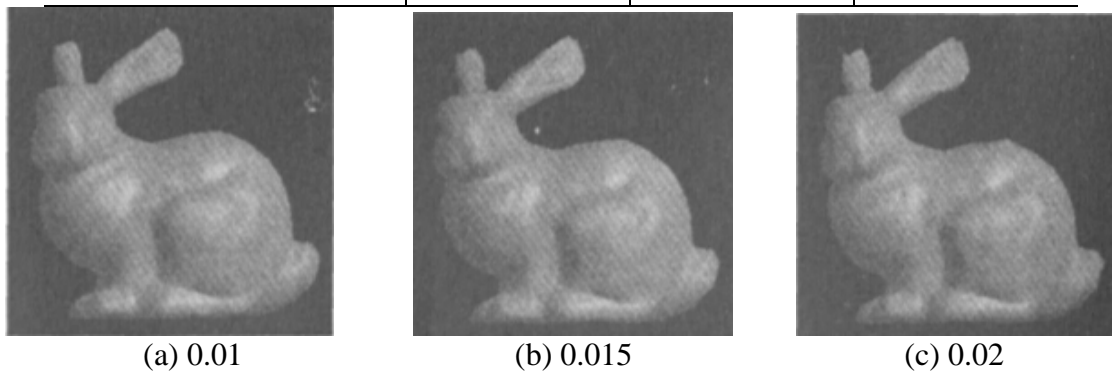


Fig.5. Noise attack

This article from the five kinds of different situations shear test. After shearing, the rest of the vertexes respectively change to 90% , 75%, 50%, 25% and 15% of the original model, and the results are shown in Figure 6 and Table 3. Table 3 shows that the algorithm has strong robustness to shear attack.

Tab.3. The results of share attack

The remaining number of vertexes	1414	1178	785
Shear rate	10%	25%	50%

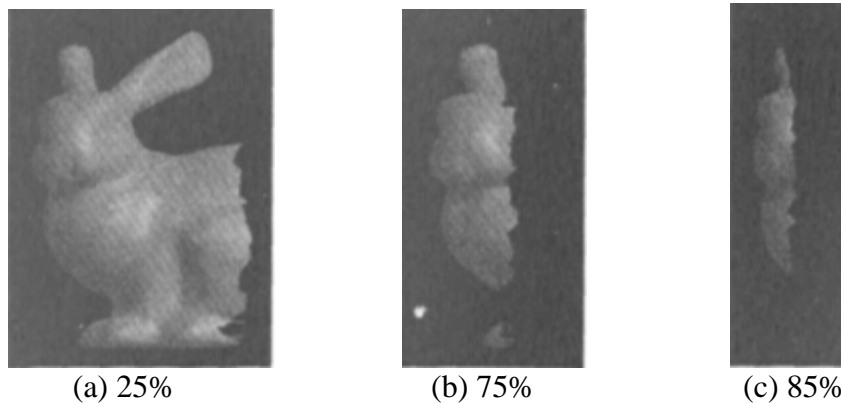


Fig.6. Share attack

Conclusion

This paper presents a robust 3d mesh model digital watermarking algorithm, and the algorithm achieves the DFT transform for vector length of model center to vertex, then realize of watermark embedding by modifying the modulus value of DFF coefficients. Simulation results show that embedded watermark is invisible, and can resist the general 3d mesh models attacks, such as mesh simplification attack, shear attack, noise attack, etc.

References

- [1] A. Tefas, A. Kallivas and I. Potas. Watermarking of 3D Models Using Principal Component Analysis[J].IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP'03),April 6-10,2003,5:676-679.
- [2] H H S Ip, Z Q Yu, L F Kowk. A Robust Watermarking Scheme for 3D Triangular Mesh Models[J].Pattern Recognition,2003,36(11):2603-2614.
- [3] Stan Melax. A imple, fast, and effective polygon reduction algorithm[J].Game Developer,1998,11:44-49.
- [4] Zhang Jing, Zheng Guoqin. Three-dimensional network of digital watermarking algorithm based on geometric features[J].Computer Aided Design and Computer Graphics,2005,17(14):740-747.