A Blind Watermarking Algorithm Based on Wavelet Transform for 3D Model

Guangmin Sun*, Zixi Zhu and Zihao Zhang
Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China

*Corresponding author email: gmsun@bjut.edu.cn

Abstract. In order to protect the visual effect of 3D model, watermarking algorithm should have good invisibility, thus a novel frequency-domain watermarking method is proposed in this paper. Robust regions of 3D model are selected as the watermarking regions in this algorithm, and watermarking regions are transformed into two-dimensional images depending on vertices distribution. Then the two-dimensional images are transformed into wavelet transform images by using wavelet transform algorithm and the watermark is embedded in low frequency coefficients of wavelet transform images. The experimental results show that the watermarking algorithm in this paper has high invisibility, and can resist common attacks such as noise, rotation, cropping and scaling.

Keywords: Blind watermarking; Wavelet transform; Invisibility.

1. Introduction
With the development of network and computer, 3D model is widely used in various fields. A quality 3D model has highly valued, but the illegal act such as unauthorized download, transmission and usage become more and more serious in the network.

Digital watermarking technology is one of the technologies that protect 3D model copyright. The special information which can protect copyright of 3D model can be embedded and extracted by watermarking algorithm. At present, 3D model watermarking algorithms can be divided into spatial watermarking and frequency-domain watermarking algorithm. The spatial watermarking algorithm embeds the watermark by altering the vertices structure of the 3D model. Spatial watermarking is easy to complete and can embed large amount of watermark data, but its robustness is not good. Jing Liu [1] proposed a method of embedding watermark by subdividing 3D model vertices. The vertices are divided into coarse level and fine level in this method. The coarse level vertices were used to build an invariant coordinate; some of the fine level vertices were used to embed watermark. This method has good robustness and invisibility. The frequency-domain watermarking has good invisibility and high robustness, but it is highly complex and has small capacity. Q.S. Ai [2] proposed an algorithm which project 3D model to two-dimensional image and embedded watermarking in DCT coefficients of two-dimensional image. This method has good robustness to various watermarking attacks.

In this paper, a 3D mesh model watermarking algorithm which has great invisibility is proposed. The robust regions of 3D mesh model are selected as the watermarking regions and these regions are transformed into two-dimensional images depending on vertices distribution of 3D mesh model. Then watermark is embedded into wavelet component of two-dimensional images, in order to protect the copyright of 3D model, the low frequency region of wavelet transform image is chosen to embed the watermark. The algorithm proposed in this paper has robustness and high invisibility.
2. Transformation Algorithm between Watermarking Region and Two-dimensional Image

In this paper, the salient value [3] of model vertices is calculated at first. The vertices with larger salient value are selected as the center vertices of the watermarking regions. Then the center of gravity for 3D model is calculated and Cartesian coordinate system is converted to spherical coordinate system. Watermarking regions are transformed into two-dimensional images depending on vertices position and structure in the spherical coordinate system.

2.1. Method of Watermarking Regions Selecting

To convert 3D model vertices into two-dimensional image, the first step is to calculate the salient value of the model vertices. The vertices salient value can be calculated by its normal vector and the vertices in its 1-ring neighborhood, which is denoted as equation (1).

\[ p(v_i) = \sum_{v_i \in N(v_i)} \frac{v_i \cdot v_j \cdot v_k}{|v_i \cdot v_j | \cdot |v_j |} \]  

(1)

Where \( v_i \) is the normal vector of vertex \( v_i \), \( v_j \) are the vertices in 1-ring neighborhood of \( v_i \), \( v_k \) is vertex from \( v_j \) to \( v_i \), \( p(v_i) \) is the salient value of \( v_i \).

The angle between vertices is used to judge the distance between vertices. The vertices with high salient value and far from other high salient value vertices are select as center vertices \( v_m \) of watermarking regions. In the follow-up experiments, three watermarking regions are selected for each model to embed the watermark.

2.2. Converting Watermarking Region to Two-dimensional Image

Firstly, the 3D model is converted into spherical coordinates system. The gravity center \[4] \( v_c \) of model is calculated by using area-moment \[5\]. The 3D model is converted into spherical coordinate system which use \( v_c \) as coordinate center, and vertices have three new coordinate components \((r, \theta, \phi)\), \( r \) is distance between vertex and \( v_c \).

Secondly, the watermarking region should be defined and polar coordinate system should be established. In order to improve the robustness of algorithm, the 1-ring neighborhood vertices of \( v_m \) aren’t embedded watermark and 1-ring neighborhood vertices are used to establish polar coordinate system. Center point of polar coordinate is \( v_m \), The vector \( v_m v_a \) from vertex \( v_a \) which is closest to the center vertex, to \( v_m \) is select as polar axis.

Then the vertex \( v_i \) is moved 10% distance between \( v_i \) and \( v_m \) to increase the robustness of polar coordinate.

In order to embed n*n bits watermark, the watermarking region should have at least \((n+2)-rings\) neighborhood vertices.

Thirdly, \((2n+3)*(2n+3)\) empty pixel matrix is needed to be built to convert the watermarking region into two-dimensional image. If the number of vertices \( N_e \) of watermarking region is smaller than \((2n+3)*(2n+3)\), the watermarking region should be expanded \(j\)-rings until \( N_e \geq (2n+3)*(2n+3) \).

The steps of method for converting the watermarking region into two-dimensional image are denoted as follows:

Step 1. \( r \) value of \( v_m \) is assigned to the center of the matrix.

Step 2. The vertices in the \( i+j \)-ring neighborhood are divided into \(8*(i-1)\) blocks by k-means algorithm base on their angle in polar coordinate.

Step 3. The blocks of step 2 are matched to \( i-th \) circle of pixels which are around the center point in the pixel matrix. The average value \( r \) of each block vertices is calculated as the pixel value. The pixel values are assigned to the \( i-th \) circle pixels of pixel matrix in counterclockwise order is started with the x-axis of the pixel matrix.

Step 4. It is repeated above step 2 and 3 until the watermarking region is completely converted. The pixel matrix is obtained in the above steps is taken as two-dimensional image which is used to embed or extract watermark in this paper’s algorithm.

3. Watermark Embedding, Extraction and Detection Algorithm

The embedding and extraction algorithm in this paper is based on two-dimensional image which is converted from watermarking region. The flow charts of the proposed method for a 3D model
watermark embedding and extraction are shown in figure 1.

![Flow chart of watermark embedding and extraction](image)

**Figure 1.** Flow chart of watermark embedding and extraction

### 3.1. Watermark Embedding

The data format of watermark is binary. In the follow-up experiment, randomly binary sequence is used as the watermark to do the testing for our algorithm.

Firstly, the pixels of two-dimensional image are normalized [6], and pixel range is expanded to 0-255. Then the first-order wavelet transform algorithm is used to transform this two-dimensional image to wavelet transform image. The steps of embedding watermark into wavelet transform image are denoted as follows:

1. **Step 1.** The (x, y) pixel value of low frequency region of wavelet transform image is extracted. The pixels of outermost circle in wavelet transform image are easily affected by noise. Thus the pixels of outermost circle aren’t embedded any message.
2. **Step 2.** (x, y) pixel value is converted to binary format.
3. **Step 3.** If watermark value at position (x, y) is 1, the last four bits of wavelet transform image pixel value at (x, y) is changed to 1000, otherwise it is changed to 0000;

Secondly, wavelet transform image is transformed to embedded two-dimensional image by inverse wavelet transform algorithm. The embedded two-dimensional image is subtracted from the original two-dimensional image to get the D-value two-dimensional image. Watermark is embedded by changing \( r \) value of each vertex in watermarking regions according to D-value two-dimensional image. After all the watermarking regions are embedded the watermark, the embedding of 3D model watermarking is accomplished.

### 3.2. Watermark Extraction

The method of watermark extraction is the reverse process of watermark embedding. Firstly, the watermarking regions are selected from 3D model, and they are transformed to two-dimensional images. Secondly, first-order wavelet transform algorithm is used to transform this two-dimensional image to wavelet transform image. Thirdly, the (x, y) pixel value in low frequency region of wavelet transform image is changed to binary format. If the last four bits of (x, y) pixel value is 1XXX (X value is 1 or 0), watermark value at position (x, y) is assigned 1. If the last four bits of (x, y) pixel value is 0XXX, watermark value at position (x, y) is assigned 0.

After all the watermarks are extracted from watermarking regions, the final value of each position in the watermark is determined by majority voting method.

### 3.3. Watermark Checking

In this paper, the correlation coefficient [7] \( \rho \) is used as the basis to judge whether the watermark is extracted successfully or not. The correlation coefficient \( \rho \) is calculated using equation (2).

\[
\rho = \frac{\sum_{i=1}^{L} (w_i - \bar{w})(w_i' - \bar{w}')}{\sqrt{\sum_{i=1}^{L} (w_i - \bar{w})^2 \sum_{i=1}^{L} (w_i' - \bar{w}')^2}}
\]

Where \( L \) is watermark data bits, \( w_i \) is the value of the i-th original watermark, \( \bar{w} \) is average value of original watermark, \( w_i' \) is the value of the i-th extracted watermark, \( \bar{w}' \) is average value of the extracted watermark. If \( \rho \) is greater than 0.5, the watermark information, which is extracted from 3D model, is extracted successfully.
4. Experiment Results and Discussion
In this section, the invisibility and robustness of the proposed method in this paper is analyzed base on the follow-up experiments. The experiments is used to compare the proposed method with the 3D model watermarking algorithms in [8]. The Bunny, Dragon and Venus models are selected as experiments models. The watermark which is embedded into 3D models is 9*9 bits random binary sequence. Three watermarking regions are selected to embed watermark for each model. These original models are shown in figure 2, the Bunny model consists of 35947 vertices and 69451 meshes, the Dragon model consists of 50107 vertices and 100071 meshes and the Venus model consists of 100759 vertices and 201514 meshes. Among them, the Dragon is a simplified model, the original model was created by Stanford Lab.

Figure 2. Original models: Bunny, Dragon, Venus

4.1. Watermark Invisibility
The watermarked models are shown in figure 3. As is shown, the watermark embedded into each model is invisible.

Figure 3. Watermarked models: Bunny, Dragon, Venus

In this paper, SNR is used to measure the invisibility of the algorithm and compare with the algorithms of [8] with same length of watermark. As is show in table 1, SNR of our proposed method is higher than method in [8].

Table 1. Comparison of watermarking invisibility in terms of SNR

| Mesh Model | Proposed Method | Method in [8] |
|------------|-----------------|--------------|
| Bunny      | 68.632          | 59.5         |
| Dragon     | 73.218          | 61.65        |
| Venus      | 77.941          | 69.53        |

4.2. Watermarking Robustness

4.2.1. Noise attack. Every vertex of each model is added with Gaussian noise [9]. Amplitude of noise is 0.1%, 0.3% and 0.5% of the average value r of model’s vertices and the value r of the vertices is affected by Gaussian noise attack. The robustness is evaluated base on correlation coefficient $\rho$ between the original watermark and extracted watermark. Result of different noise attacks is shown in table 2.
Table 2. Evaluation of robustness against noise attack

| Mesh Model | Noise Intensity | Correlation Coefficient |
|------------|-----------------|-------------------------|
| Bunny      | 0.1%            | 1                       |
|            | 0.3%            | 0.82716                 |
|            | 0.5%            | -0.03704                |
| Dragon     | 0.1%            | 0.95062                 |
|            | 0.3%            | 0.90123                 |
|            | 0.5%            | 0.72840                 |
| Venus      | 0.1%            | 1                       |
|            | 0.3%            | 0.48418                 |
|            | 0.5%            | 0.18519                 |

The algorithm in this paper has good robustness when the noise is less than 0.5%. The watermarked models which are added noise with 0.5% value $r$ amplitude are shown in figure 4. Because the surfaces of Bunny and Venus are seriously damaged, the watermark cannot be extracted correctly.

![Models attacked by noise](image)

Figure 4. Models are attacked by 0.5% noise

4.2.2. Rotation and Scaling attack. This paper’s algorithm is not based on the original coordinate system of model. Thus, it has good robustness against rotation and scaling attack. The result of watermark extraction after rotation attack is shown in table 3. In the three models, watermark can be extracted correctly after scaling attack, and all the correlation coefficient $\rho$ are 1.

Table 3. Evaluation of robustness against rotate attack

| Mesh Model | Rotation Angle | Correlation Coefficient |
|------------|----------------|-------------------------|
| Bunny      | 45             | 0.95062                 |
|            | 90             | 0.75309                 |
|            | 135            | 0.80247                 |
|            | 180            | 1                       |
| Dragon     | 45             | 1                       |
|            | 90             | 1                       |
|            | 135            | 1                       |
|            | 180            | 1                       |
| Venus      | 45             | 1                       |
|            | 90             | 1                       |
|            | 135            | 1                       |
|            | 180            | 1                       |

4.2.3. Cropping attack. The algorithm in this paper is unable to extract watermark blindly after cropping attack, because cropping attack skews the gravity center of 3D model. The gravity center of 3D model is need to be calculated by using the original 3D model when model is not cropped, and the original gravity center is used to extract watermark. 10%, 20% and 30% of vertices are cropped of each model from three different directions a, b and c. The results of watermark extraction after cropping attack are shown in table 4. The watermark extraction results are different from same degree cropping of different directions. In this paper the watermark is embedded in three watermarking regions of each model. Therefore, the results are depended on whether the watermarking regions in the
model are cropped or not. If more than two watermarking regions are cropped, it is difficult to extract watermark correctly.

Table 4. Evaluation of robustness against cropping attack

| Mesh Model | Shear Proportion | Correlation Coefficient |
|------------|------------------|-------------------------|
|            | a                | b           | c              |
| Bunny      | 10%              | 1           | 1              | 0.58025        |
|            | 20%              | 1           | 1              | 0.45679        |
|            | 30%              | 0.85185     | 0.48148       | 0.4321         |
| Dragon     | 10%              | 0.97531     | 0.85185       | 0.67901        |
|            | 20%              | 0.97531     | 0.82716       | 0.50617        |
|            | 30%              | 0.97531     | 0.38272       | -0.03704       |
| Venus      | 10%              | 1           | 0.97531       | 0.75309        |
|            | 20%              | 1           | 0.92593       | 0.4321         |
|            | 30%              | 0.97531     | 0.92593       | 0.4321         |

5. Conclusion

In this paper a novel watermarking algorithm is proposed, which convert the watermarking regions of 3D model into two-dimensional images depending on vertices distribution. Then the algorithm from 2D image watermarking technology is used to embed and extract watermark for 3D model. The position of model’s vertices is changed little during the program of embedding watermark, thus the algorithms of this paper has high invisibility. In this paper embedding watermark doesn’t destroy the value of 3D mode and can resist various kinds attacks such as noise, rotation, cropping and scaling.

Acknowledgments

This work was supported by the National Key R&D Program of China (2018YFF01012300) and National Natural Science Foundation of China (11527801, 41706201)

References

[1] Jing Liu, Yinghui Wang, Ye Li, Ruijiao Liu, Jinlei Chen. A robust and blind 3D watermarking algorithm using multiresolution adaptive parameterization of surface[J]. Neurocomputing, 2016.
[2] Q.S. Ai, Q. Liu, Z.D. Zhou, L. Yang, S.Q. Xie. A new digital watermarking scheme for 3D triangular mesh models[J]. Signal Processing, 2009, 89(11).
[3] XinChen An. Research on Robust Watermarking Algorithm for 3D model[D]. Beijing JiaoTong University, 2018.
[4] Mikata Y., Yamanaka M., Kameoka K., Okunosono A., Kinoshita T.. Measuring the center of gravity with truck scale[C]. Sice Conference. IEEE, 2011.
[5] Tudor Barbu, Mihaela Costin, Adrian Ciobanu. Content-based image recognition technique using area moments[C]. Soft Computing Applications (SOFA), 2010 4th International Workshop on. IEEE, 2010.
[6] Sung Jun Ban, Hyeonwoo Cho, Jae Jin Jeong, Sang Woo Kim. A normalized LMF (XE-NLMF) algorithm with variable data-reusing[C]. International Conference on Information. IEEE, 2010.
[7] Zhen Li, Wei-Min Zheng, Zhe-Ming Lu. A Robust Geometry-Based Watermarking Scheme for 3D Meshes[C]. First International Conference on Innovative Computing, Information and Control - Volume. IEEE Computer Society, 2006.
[8] Soliman M M, Hassanien A E , Onsi H M . Robust watermarking approach for 3D triangular mesh using self organization map[C]. International Conference on Computer Engineering & Systems. IEEE, 2013.
[9] Gini, F., Farina, A., Greco, M.V.. Detection of multidimensional Gaussian random signals in compound-Gaussian clutter plus thermal noise[C]. Fourth International Conference on Signal Processing. IEEE, 1998.