

A 3D Mesh Watermarking Based on Improved Vertex Grouping and Piecewise Mapping Function

Song Li, Rongrong Ni() and Yao Zhao

Institute of Information Science
and Beijing Key Laboratory of Advanced Information Science and Network Technology
Beijing Jiaotong University, Beijing 100044, China
14120326@bjtu.edu.cn; rрни@bjtu.edu.cn; yzhao@bjtu.edu.cn

Received July, 2016; revised August, 2016

ABSTRACT. *For the three-dimensional (3D) mesh watermarking, the trade-off between the robustness and transparency has been pursued by researchers. In this paper, a blind watermarking algorithm is proposed to improve the robustness by embedding the stronger watermark while ensuring a good performance in the transparency. In the embedding process, we choose the distance from the vertex to the model center as the eigenvalue. Eigenvalues are divided into the corresponding bins according to the improved vertex grouping method, which enhances the robustness of the watermark in the marginal bins. In the process of adjusting vertices, the proposed algorithm effectively controls the transparency of 3D models by the piecewise mapping function. The experimental results demonstrate that the proposed approach is remarkably robust against common attacks including rotation, translation, uniform scaling, vertex reordering, noise, smoothing, quantization and subdivision.*

Keywords: 3D mesh watermarking, Blind, Robust, improved vertex grouping, Piecewise mapping function

1. Introduction. With the rapid development of 3D modeling and multimedia technology, more and more artificial 3D works and realistic 3D objects are permanently preserved in the form of digital works. However, these digital works are easy to be illegally copied and tampered for further usages in the transmission process, which makes security and copyright of digital works face severe challenges. 3D mesh digital watermarking technology [1] is the process of embedding the secret information into 3D models to achieve the purpose of copyright protection.

Compared with images [2], the amount of research literatures about 3D mesh watermarking technology is still relatively less, which mainly results from two major reasons. Firstly, vertices of 3D mesh models [3] are disordered, which virtually increases the difficulty of the watermark embedding. Secondly, the watermark detection of 3D models becomes extremely hard due to a series of complicated attacks.

Although the research for 3D mesh digital watermarking technology is facing a variety of difficulties, lots of excellent algorithms have emerged since the first paper [4] about digital watermarking of 3D models was published in 1997. In the initial period of development, Ohbuchi et al. [5] proposed several feasible spatial algorithms including Triangle Similarity Quadruple(TSQ) and Tetrahedral Volume Ratio(TVR) methods. Although the robustness of these algorithms is weak, it is worth mentioning that these algorithms

provide a feasible way for the subsequent 3D mesh digital watermarking technology. According to the difference of watermark embedding positions, watermarking algorithms are classified into transform domain algorithms [6]- [8] and spatial domain algorithms [9]- [16]. The former inserts the watermark in the transform domain coefficients, while the latter directly modifies vertex positions to embed the watermark. In transform domain methods, the multiresolution wavelet decomposition [6] and the mesh Laplacian spectral decomposition [7, 8] are applied to the 3D watermarking. However, most of the watermarking algorithms are spatial domain algorithms at present. In spatial domain approaches, vertex norm [12] is used to embed the watermark in the Euclidean distance between the vertex and a reference structure, which is a common strategy. In [12], two classical methods were proposed to embed watermark bits by different histogram mapping functions. The first method changes the mean value of the vertex norm and the second shifts its variance. Later, in order to improve the performance of the transparency, several methods [13, 14] were proposed to obtain the minimal surface distortion by different optimization methods. For the sake of robustness against different attacks, some model surface characteristics such as integral invariant [15] and vertex curvature [16] represent some good properties for the watermark embedding.

Obviously, the robustness and transparency of the watermarking algorithm have been the focus of the research in order to satisfy the demand on the copyright protection of 3D models in the actual applications. At the same time, considering the process of the watermark detection without the original model, the blind watermarking has greater advantages in practical applications. Based on the facts above, we propose a robust and blind watermarking algorithm based on improved vertex grouping and piecewise mapping function of 3D models in this paper. In the embedding process, the distance from the vertex to the model center is calculated for every vertex of the model, which is called the eigenvalue. Then an ordered set of eigenvalues is obtained. According to the histogram of ordered eigenvalues, 10% of the total number of vertices are discarded in the histogram on both ends. Next, vertices are separated into corresponding bins on the principle that each bin has the same amount of effective vertices. In each bin, the eigenvalues are normalized to a range of [0,1] and then a bit watermark is embedded in a bin by the piecewise mapping function to adjust the mean of eigenvalues.

The rest of this paper is organized as follows. Section 2 explains the proposed method. Experiment results are given in Section 3. Conclusions are included in Section 4.

2. The Proposed Method. In general, the information for 3D mesh models includes vertices, facets and other attributes [17]. A 3D mesh model can be represented as $M = (V, F)$, where $V = \{v_i \in R^3 | 0 \leq i \leq L_v - 1\}$ represents the set of all vertices of the model, $v_i = (x_i, y_i, z_i)$ denotes cartesian coordinate, L_v is the number of vertices, $F = \{f_j \in N_+ | 0 \leq j \leq L_f - 1\}$ denotes the set of all triangular facets representing mesh topology, where f_j represents a facet and L_f is the number of facets in the 3D mesh model.

In this paper, the proposed method focuses on improving the robustness of the watermark and guaranteeing the transparency of the model. In this section, the watermark embedding and extracting processes are described in details. The block diagram of watermark embedding and extracting is shown in Fig. 1.

2.1. Watermark Generation. Usually, the embedded watermark information is a stream of binary bits, which consists of "0" and "1". For the sake of simplicity, the binary pseudo random sequence is adopted as the watermark information. The original watermark can be denoted by $W^o = (w_0^o, w_1^o, \dots, w_{N-1}^o)$, where $w_i^o \in \{0, 1\}$, $0 \leq i \leq N - 1$. N is the length of the original watermark.

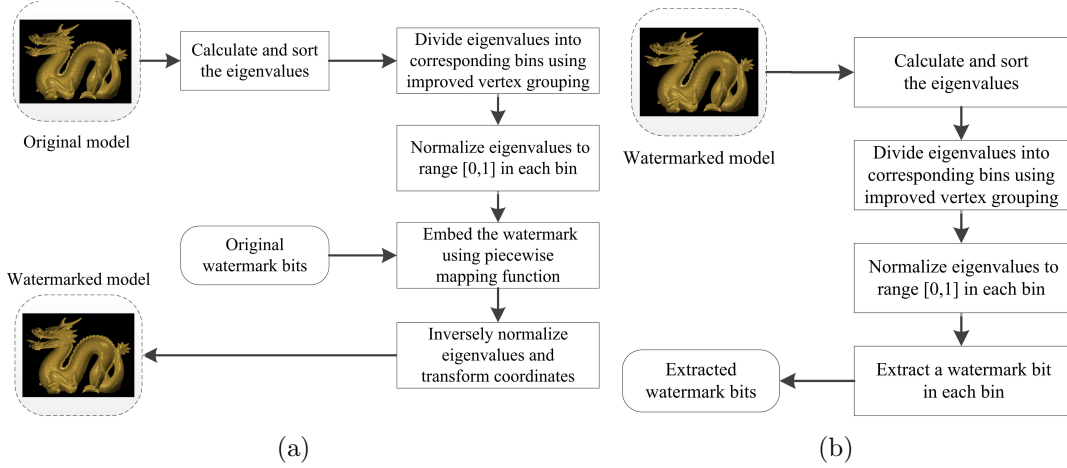


FIGURE 1. The block diagram of watermark embedding (a) and watermark extracting (b).

2.2. Watermark Embedding Process.

2.2.1. *Calculate and sort eigenvalues.* The eigenvalue is the distance from the vertex to the model center. The process of calculating and sorting eigenvalues can be executed in the following steps:

(a) Calculate the center of the model $\bar{v} = (\bar{x}, \bar{y}, \bar{z})$ by following formulas:

$$\bar{x} = \sum_{i=0}^{L_v-1} x_i, \bar{y} = \sum_{i=0}^{L_v-1} y_i, \bar{z} = \sum_{i=0}^{L_v-1} z_i \quad (1)$$

(b) Calculate the eigenvalue ρ_i for every vertex of the model. In order to obtain the i^{th} eigenvalue, cartesian coordinate (x_i, y_i, z_i) of the vertex is transformed to spherical coordinate $(\rho_i, \theta_i, \psi_i)$ by formulas as follows:

$$\begin{cases} \rho_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2} \\ \theta_i = \tan^{-1} \frac{(y_i - \bar{y})}{(x_i - \bar{x})} \\ \psi_i = \cos^{-1} \frac{(z_i - \bar{z})}{\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2}} \end{cases} \quad (2)$$

where $0 \leq i \leq L_v - 1$, ρ_i is the i^{th} eigenvalue. Eigenvalues are used to embed the watermark bits. θ_i and ψ_i are invariable in the watermark embedding process. These two parameters are used to convert coordinates after the completion of the watermark embedding.

(c) All the eigenvalues are sorted in an ascending order.

In order to intuitively show the distribution of eigenvalues, eigenvalues are divided into 100 bins. Fig. 2 shows the original model Dragon and the corresponding histogram of normalized eigenvalues. For convenience, eigenvalues are normalized to the range of $[0, 1]$ in the horizontal axis. The vertical axis represents the number of eigenvalues in each bin. The histogram of Dragon roughly obeys the normal distribution.

2.2.2. *Divide eigenvalues into corresponding bins by the improved vertex grouping method.* In order to embed the watermark, eigenvalues are assigned to the corresponding bins according to the watermark length N . In [12], eigenvalues are divided into 64 bins and

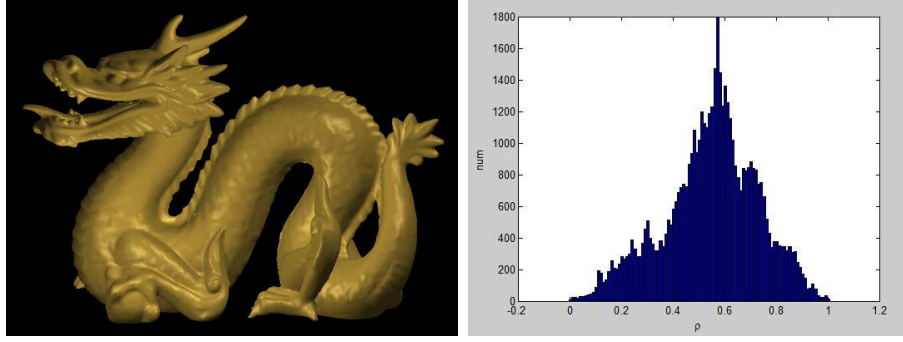


FIGURE 2. The original model Dragon and the corresponding histogram of normalized eigenvalues.

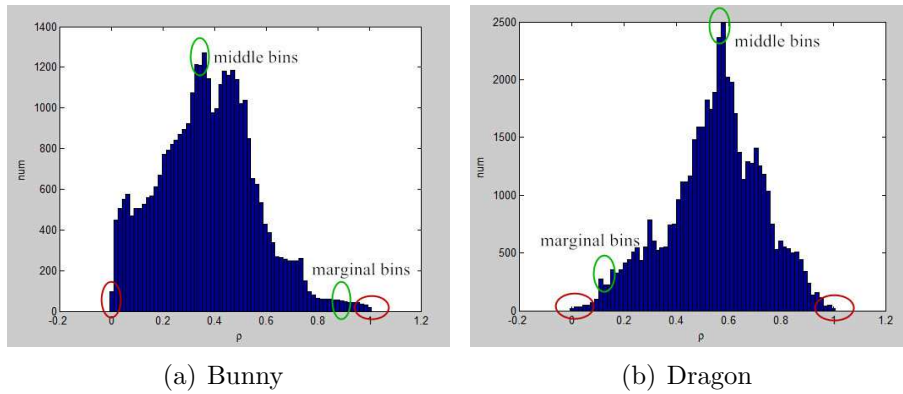


FIGURE 3. The eigenvalues are divided into 64 bins using the grouping method in [12].

histograms of eigenvalues are showed in Fig. 3. For the convenience of display, eigenvalues are normalized to the range of $[0, 1]$ in the horizontal axis. In the bins with green circle highlighted, the number of vertices in the marginal bins is far less than that of the middle bins, which leads to the weak robustness in the marginal bins compared with the middle bins when a bit watermark is embedded a bin in the histogram. In addition, when the capacity is too large, the bins marked in red on both ends may fail to embed the watermark because the number of vertices in these bins is minimal or empty. In order to solve the above mentioned problems, we propose an improved vertex grouping method which promises the same amount of effective vertices in each bin. Eigenvalues are divided into N bins according to the steps below:

(a) Get rid of 10% of the total number of vertices in the histogram on both ends. The rest number of vertices is L'_v , where $L'_v = \lfloor 0.9L_v \rfloor$.

(b) The number of vertices in each bin is denoted as $M = \lfloor L'_v/N \rfloor$. The remainder $L'_v \% N$ vertices are discarded.

(c) The m^{th} bin B_m is represented as follows:

$$B_m = \{\rho_{m,j} | \rho_{m,j} = \rho_i, i = \lfloor 0.05L_v \rfloor + M \cdot m + j\}$$

where,

$$\begin{aligned} \lfloor 0.05L_v \rfloor + M \cdot m &\leq i < \lfloor 0.05L_v \rfloor + M \cdot (m + 1) \\ 0 \leq m \leq N - 1, 0 \leq i &\leq M \cdot N - 1, 0 \leq j \leq M - 1 \end{aligned} \quad (3)$$

$\rho_{m,j}$ is the j^{th} eigenvalue of the m^{th} bin. $M \cdot N$ is the number of actual used vertices in the watermark embedding process. In the proposed method above, eigenvalues are divided into N bins to possess the same amount of effective vertices in each bin, which improves the robustness of the watermark in the marginal bins. After the vertex grouping process, all the groups for watermark embedding are obtained.

2.2.3. *Normalize eigenvalues in each bin.* Eigenvalues in the m^{th} bin are normalized to a range of $[0, 1]$ by the following formula:

$$\rho'_{m,j} = \frac{\rho_{m,j} - \rho_m^{\min}}{\rho_m^{\max} - \rho_m^{\min}} \quad (4)$$

where $\rho'_{m,j}$ is the j^{th} normalized eigenvalue in the m^{th} bin, ρ_m^{\max} and ρ_m^{\min} are the maximum eigenvalue and the minimum eigenvalue in the m^{th} bin, respectively.

2.2.4. *Embed the watermark using the piecewise mapping function for each bin.* To ensure the robustness of the algorithm, one bit watermark is embedded in a bin by adjusting the mean of eigenvalues to satisfy the setting requirements. The m^{th} watermark bit meets the following embedding conditions:

$$\begin{cases} \bar{\rho}_m < 0.5 - \alpha, & \text{if } w_m^o = 0 \\ \bar{\rho}_m > 0.5 + \alpha, & \text{if } w_m^o = 1 \end{cases} \quad (5)$$

$$\bar{\rho}_m = \frac{1}{M} \cdot \sum_{j=0}^{M-1} \rho'_{m,j} \quad (6)$$

where $\bar{\rho}_m$ is the mean of eigenvalues in the m^{th} bin, α is the watermark strength. M is the number of vertex in the m^{th} bin. The mapping function Eq. 7 is introduced to adjust the eigenvalues $\rho'_{m,j}$ to satisfy the condition of watermark embedding.

$$Y = X^k \quad (7)$$

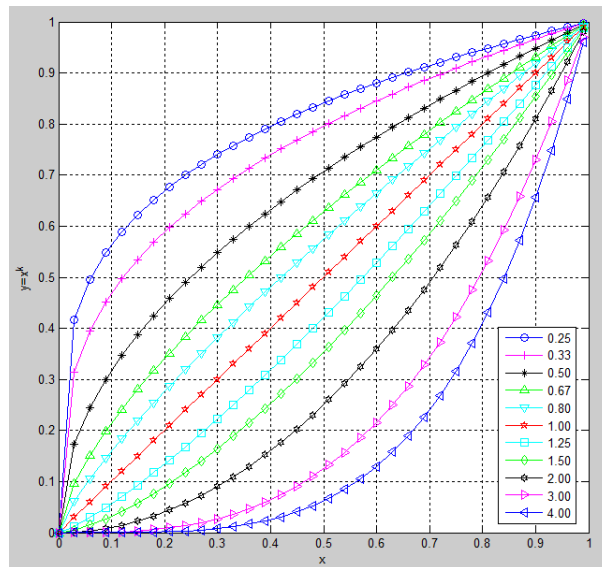


FIGURE 4. The mapping function $Y = X^k$.

TABLE 1. The mapping results in the condition of the value $k = 2$

X	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$Y = X^2$	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1
$\Delta = Y - X $	0	0.09	0.16	0.21	0.24	0.25	0.24	0.21	0.16	0.09	0

TABLE 2. The mapping results in the condition of the value $k = 0.5$

X	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$Y = X^{0.5}$	0	0.32	0.45	0.55	0.63	0.71	0.77	0.84	0.89	0.95	1
$\Delta = Y - X $	0	0.22	0.25	0.25	0.23	0.21	0.17	0.14	0.09	0.05	0

Fig. 4 shows the mapping function curves with different value k . The different value k has the different influence in the mapping process. When the value k is greater than 1, the mapping value Y is smaller than the original value X . On the contrary, the mapping value Y is larger than the original value X . For example, Table 1 and Table 2 show the mapping results in the condition of the value $k = 2$ and $k = 0.5$, respectively. The variable quantity Δ reflects the different degree of modification to the original value X in the mapping process. The larger variable quantity Δ will lead to the larger local distortion of 3D model. However, good transparency requires the smaller amplitude changes of eigenvalues. Based on this observation, the piecewise mapping function is proposed as follows:

$$\rho''_{m,j} = (\rho'_{m,j})^{k_{m,j}^{(n)}} \quad (8)$$

when $w_i = 0$:

$$k_{m,j}^{(n)} = 1 + n \cdot \Delta k \quad (9)$$

$$\Delta k = \begin{cases} 0.002, & \text{if } \rho'_{m,j} \in [0, 0.3) \\ 0.001, & \text{if } \rho'_{m,j} \in [0.3, 0.9) \\ 0.0015, & \text{if } \rho'_{m,j} \in [0.9, 1] \end{cases} \quad (10)$$

when $w_i = 1$:

$$k_{m,j}^{(n)} = 1 - n \cdot \Delta k \quad (11)$$

$$\Delta k = \begin{cases} 0.001, & \text{if } \rho'_{m,j} \in [0, 0.6) \\ 0.0015, & \text{if } \rho'_{m,j} \in [0.6, 0.8) \\ 0.002, & \text{if } \rho'_{m,j} \in [0.8, 1] \end{cases} \quad (12)$$

$k_{m,j}^{(n)}$ denotes the mapping parameter of the j^{th} eigenvalue in the m^{th} bin during the n^{th} iteration and Δk is the step size. In view of the magnitude of eigenvalues, the algorithm adopts different value Δk to obtain the piecewise mapping function, which will ensure that the eigenvalue is mapped to a controllable range. The initial n is 0 and $k_{m,j}^{(0)}$ is 1. When the watermark bit is "0", $\rho''_{m,j}$ can be obtained by Eq. 8 and the mean of eigenvalues $\bar{\rho}_m$ is calculated by Eq. 6. If the embedding conditions in the Eq. 5 are met, a watermark bit is embedded in the bin. Otherwise, n adds 1 and the updated $\rho''_{m,j}$ is obtained. With the increase of n , a watermark bit is embedded successfully in the bin once meeting the embedding conditions. After the watermark bit is embedded, the modified eigenvalue is $\rho''_{m,j}$.

TABLE 3. The watermark strength in the experiment

α	Bunny	Dragon	Rabbit	Venus
Cho approach	0.03	0.03	0.03	0.03
Proposed approach	0.05	0.05	0.05	0.045

2.2.5. *Inversely normalize the eigenvalue and transform coordinates.* Upon the completion of the watermark embedding, the inverse normalization process is executed corresponding to the normalization process. The eigenvalues of m^{th} bin are converted to the original range by the following formula:

$$\tilde{\rho}_{m,j} = (\rho_{m,j}'' \cdot (\rho_m^{max} - \rho_m^{min}) + \rho_m^{min}) \quad (13)$$

The last step is that spherical coordinate $(\tilde{\rho}_i, \theta_i, \psi_i)$ of the vertex is converted to cartesian coordinate (x'_i, y'_i, z'_i) . Conversion formula is as follows:

$$\begin{cases} x'_i = \tilde{\rho}_i \cdot \cos \theta_i \cdot \sin \psi_i + \bar{x} \\ y'_i = \tilde{\rho}_i \cdot \sin \theta_i \cdot \sin \psi_i + \bar{y} \\ z'_i = \tilde{\rho}_i \cdot \cos \psi_i + \bar{z} \end{cases} \quad (14)$$

the coordinate (x'_i, y'_i, z'_i) is the cartesian coordinate of the new vertex in the watermarked model.

2.3. Watermark extracting process. The proposed algorithm is a blind watermarking algorithm and the watermark extraction process is relatively simple. Firstly, the eigenvalue is calculated for every vertex of the 3D model and all the eigenvalues are sorted in an ascending order. Secondly, after 10% of the total number of vertices are discarded, vertices are assigned to the corresponding bins by the improved vertex grouping method in each bin. The number of bins is determined by the foregone length of the embedded watermark. Then eigenvalues in each bin are normalized to the range of $[0, 1]$. In the end, the mean eigenvalue $\bar{\rho}_i^d$ of the i^{th} bin is calculated and the watermark bit in the i^{th} bin is obtained by the following formula:

$$w_i^d = \begin{cases} 0, & \text{if } \bar{\rho}_i^d \leq 0.5 \\ 1, & \text{if } \bar{\rho}_i^d > 0.5 \end{cases} \quad (15)$$

where $0 \leq i \leq N - 1$. So the extracted watermark bits are represented by the following sequence:

$$W^d = (w_0^d, w_1^d, \dots, w_{N-1}^d) \quad (16)$$

3. Experimental Results. In this paper, the proposed algorithm is implemented on a DELL computer with an Intel Core i5 3.30 GHz processor and 8 GB of memory using OpenGL library and Visual Studio 2013. In the experiments, the length N of the watermark is 64. Table 3 shows the embedding watermark strength α of the experimental parameter. The proposed algorithm is compared with the algorithm of Cho et al. [12]. The contrasted algorithm has been implemented with the same models using the given parameters.

3.1. The experimental models. We choose the Bunny, Dragon, Rabbit and Venus as experimental models in this section. Experimental models have different number of vertices and facets. Their data information are shown in Table 4 and the experimental models are shown in Fig. 5.

TABLE 4. The data information of the experimental models

Model	Bunny	Dragon	Rabbit	Venus
Vertices	34835	50000	70658	100759
Facets	69666	100000	141312	201514

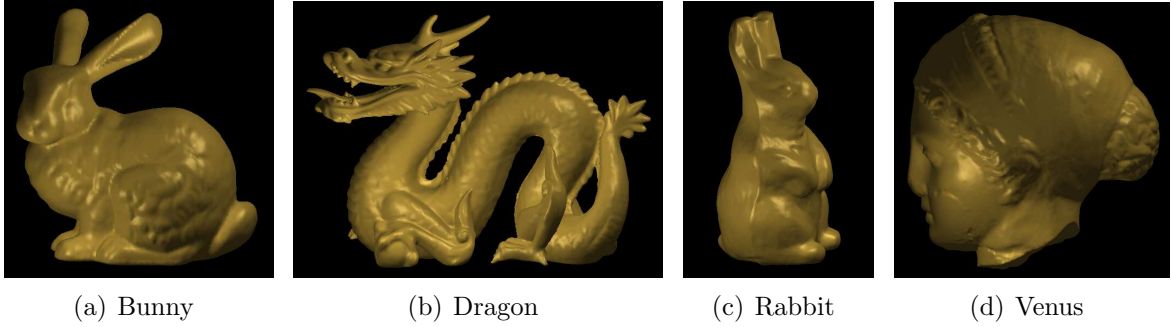


FIGURE 5. The original models in the experiment.

TABLE 5. The results of the transparency in the experiment ($\times 10^{(-3)}$)

MRMS	Bunny	Dragon	Rabbit	Venus
Cho approach	0.326	0.306	0.354	0.223
Proposed approach	0.244	0.306	0.261	0.216

3.2. The evaluation of the transparency. In the experiment, we use maximum root mean square (MRMS) as the evaluation of transparency. The MRMS is calculated using the following formula by Metro [18].

$$E(V, V') = \max \left\{ e_f(V, V'), e_b(V', V) \right\} \quad (17)$$

$$e_f(V, V') = \frac{1}{|V|} \int_V \left\{ \min_{v' \in V'} \left\{ \|v - v'\| \right\} \right\} dv \quad (18)$$

$$e_b(V', V) = \frac{1}{|V'|} \int_{V'} \left\{ \min_{v \in V} \left\{ \|v' - v\| \right\} \right\} dv' \quad (19)$$

Note that $e_f(V, V')$ and $e_b(V', V)$ denote the forward and backward root mean square (rms) errors, respectively. V and V' represent the original and watermarked mesh model, respectively. The smaller MRMS value represents the lower distortion of the model. Table 5 shows the experimental results of the transparency under the embedding watermark strength in Table 3.

Table 3 and Table 5 show that the MRMS values of our method are smaller than the Cho's method when our method embeds the larger watermark intensity. Therefore, our method have the stronger robustness under the condition of the better transparency.

3.3. The evaluation of the robustness. The robustness of the watermark is measured by the correlation value between the extracted watermark and the original watermark.

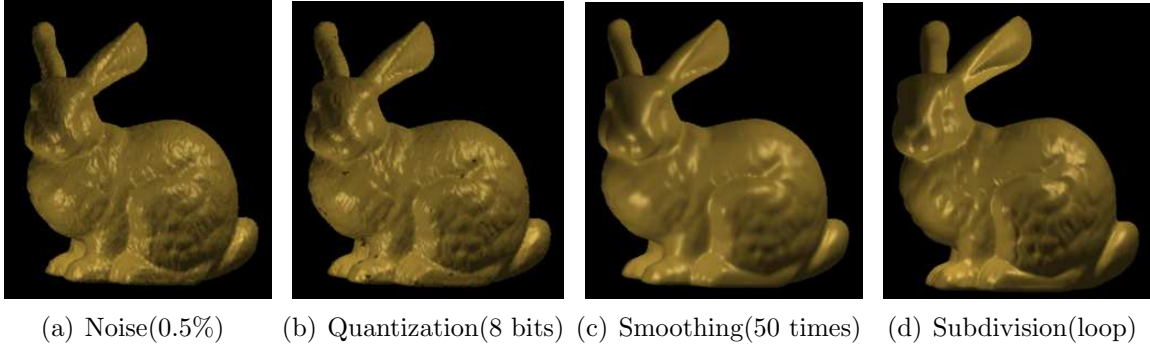


FIGURE 6. The different attacked Bunny models.

TABLE 6. The robustness results of noise attacks

Model	Amplitude (%)	MRMS		Corr	
		Cho	Proposed	Cho	Proposed
Bunny	0.1	0.397	0.327	0.97	1.00
	0.3	0.740	0.712	0.84	0.90
	0.5	1.165	1.141	0.61	0.68
Dragon	0.1	0.389	0.391	0.90	1.00
	0.3	0.795	0.795	0.87	1.00
	0.5	1.254	1.255	0.65	0.75
Rabbit	0.1	0.410	0.334	1.00	1.00
	0.3	0.718	0.680	0.97	0.97
	0.5	1.111	1.084	0.81	0.87
Venus	0.1	0.395	0.391	0.97	1.00
	0.3	1.005	1.006	0.71	0.72
	0.5	1.653	1.654	0.26	0.25

The correlation value is defined as follows:

$$Corr = \frac{\sum_{i=0}^{N-1} (w_i^d - \overline{w^d}) \cdot (w_i^o - \overline{w^o})}{\sqrt{\sum_{i=0}^{N-1} (w_i^d - \overline{w^d})^2 \cdot \sum_{i=0}^{N-1} (w_i^o - \overline{w^o})^2}} \quad (20)$$

where $\overline{w^d}$ is the mean of the extracted watermark bits, w_i^d is the i^{th} watermark bit of the extracted watermark. In the same way, $\overline{w^o}$ is the mean of the original watermark bits, w_i^o is the i^{th} watermark bit of the original watermark. The correlation value is on the range of $[-1,1]$. The greater correlation value shows that the extracted watermark is more similar to the original watermark.

In order to evaluate the robustness of the algorithm, the attacked models are obtained with different strengths and attacks using the software which is provided by LIRIS lab [19]. For noise attacks, noise intensities are 0.1%, 0.3% and 0.5%. The parameter λ for smoothing attacks is 0.03 and iterations choose 10, 30 and 50, respectively. The vertex coordinates of the model are quantified by intensities including 9 bits, 8 bits and 7 bits. Three types of subdivision attacks are loop subdivision, midpoint subdivision and sqrt(3) subdivision. Fig. 6 shows the part models of the attacked Bunny.

TABLE 7. The robustness results of smoothing attacks

Model	Number of iterations	MRMS		Corr	
		Cho	Proposed	Cho	Proposed
Bunny	10	0.411	0.342	0.87	1.00
	30	0.746	0.705	0.50	0.77
	50	1.079	1.046	0.35	0.38
Dragon	10	0.412	0.412	0.90	1.00
	30	0.854	0.856	0.56	0.80
	50	1.298	1.301	0.17	0.55
Rabbit	10	0.342	0.250	1.00	1.00
	30	0.366	0.292	0.84	1.00
	50	0.428	0.377	0.63	0.88
Venus	10	0.238	0.238	0.90	1.00
	30	0.372	0.379	0.72	0.94
	50	0.525	0.535	0.52	0.72

TABLE 8. The robustness results of quantization attacks

Model	Intensity (bits)	MRMS		Corr	
		Cho	Proposed	Cho	Proposed
Bunny	9	0.616	0.578	0.94	0.97
	8	1.106	1.076	0.57	0.58
	7	2.093	2.085	0.22	0.17
Dragon	9	0.645	0.643	0.74	0.97
	8	1.173	1.181	0.43	0.52
	7	2.327	2.319	0.22	0.25
Rabbit	9	0.539	0.482	0.88	0.97
	8	0.901	0.865	0.69	0.88
	7	1.705	1.691	0.07	0.37
Venus	9	0.701	0.697	0.74	0.88
	8	1.338	1.340	0.48	0.58
	7	2.712	2.704	-0.01	-0.12

The proposed algorithm can completely resist rotation, translation, uniform scaling and vertex reordering attacks. Because the algorithm uses the distance from the vertex to the model center as the eigenvalue. The eigenvalue is the relative value and all the eigenvalues are executed the sorting operation. So the extracted watermark bits and the original watermark bits are exactly same, which means that their correlation value is 1. Table 6, 7, 8, 9 show the robustness and the corresponding transparency after four attacks. It is noticed that our approach has better correlation value than those of Cho in a series of attacks. In addition, because of the increased watermark strength, the robustness for mild noise and smoothing attacks is quite strong so that the correlation value between the extracting watermark and the original watermark is 1. Along with the increase of the attack strength, the accuracy of the extracted watermark has the different degree of reduction under the condition of common attacks. The transparency is evaluated by MRMS value between the original model and the attacked watermarked model. The greater the attack intensity is, the larger the distortion of 3D models is. In terms of the

TABLE 9. The robustness results of subdivision attacks

Model	Types of subdivision	MRMS		Corr	
		Cho	Proposed	Cho	Proposed
Bunny	Loop	0.414	0.350	0.38	0.72
	Midpoint	0.323	0.239	0.81	0.93
	Sqrt(3)	0.387	0.317	0.70	0.70
Dragon	Loop	0.409	0.410	0.65	0.80
	Midpoint	0.301	0.300	0.82	0.94
	Sqrt(3)	0.371	0.349	0.74	0.80
Rabbit	Loop	0.343	0.250	0.91	1.00
	Midpoint	0.352	0.259	1.00	1.00
	Sqrt(3)	0.344	0.251	0.94	1.00
Venus	Loop	0.239	0.237	0.74	0.94
	Midpoint	0.221	0.213	0.87	0.97
	Sqrt(3)	0.232	0.230	0.78	0.94

transparency, the smaller MRMS value represents the lower distortion of 3D models. Our algorithm has an advantage over Cho's method in the aspect of the transparency.

4. Conclusions. In this paper, we propose a blind and robust 3D watermarking algorithm based on improved vertex grouping and piecewise mapping function to improve the robustness while ensuring a good performance in the transparency. On the one hand, the algorithm reasonably divides eigenvalues into corresponding bins to enhance the robustness of the watermark in the marginal bins by the improved vertex grouping method. On the other hand, the algorithm adjusts the mean of eigenvalues to embed the watermark by the piecewise mapping function, which ensures the low distortion of the model. The experiment results demonstrate that the proposed method is more robust against common attacks compared with the Cho's method.

Acknowledgment. This work was supported in part by National NSF of China (61332012, 61272355, 61672090), Beijing Nova Programme Interdisciplinary Cooperation Project, Fundamental Research Funds for the Central Universities (2015JBZ002), the PAPD, the CICAET, CCF-Tencent Open Research Fund.

REFERENCES

- [1] N. Medimegh, S. Belaid, and N. Werghi, 3D triangular mesh watermarking: a survey, *International Conference on Communications, Signal Processing, and Their Applications*, Sharjah, United Arab Emirates, 2015.
- [2] X. Ding, Z. M. Lu and F. X. Yu, A robust blind image watermarking scheme based on classified vector quantization, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 74-80, 2015.
- [3] M. Botsch, M. Pauly, L. Kobbelt, P. Alliez, B. Levy, S. Bischoff, and C. Rossl, Geometric modeling based on polygonal meshes, *Proc. ACM SIGGRAPH Course Notes*, 2007.
- [4] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking three-dimensional polygonal models, *Proceedings of the 5th ACM International Conference on Multimedia*, New York, USA, pp. 261-272, 1997.
- [5] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking three-dimensional polygonal models through geometric and topological modifications, *IEEE Journal Selected Areas in Communications*, vol. 16, no. 4, pp. 551-560, 1998.

- [6] S. Kanai, H. Date, and T. Kishinami, Digital watermarking for 3D polygons using multiresolution wavelet decomposition, *Proceedings of the 6th IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications(GEO)-6*, Tokyo, Japan, pp. 296-307, 1998.
- [7] R. Ohbuchi, A. Mukaiyama, and S. Takeaways, A frequency domain approach to watermarking 3D shapes, *Proceedings of Annual Conference of the European Association for Computer Graphics*, Saarbrucken, Germany, vol. 21, pp. 373-382, 2002.
- [8] E. E. Abdallah, A. B. Hamza, and P. Bhattacharya, Watermarking 3D models using spectral mesh compression, *Signal Image & Video Processing*, vol. 3, no. 4, pp. 375-389, 2009.
- [9] Z. Q. Yu, H. H. S. Ip, and L. F. Kwok, A robust watermarking scheme for 3D triangular mesh models, *Pattern Recognition*, vol. 36, no. 11, pp. 2603-2614, 2003.
- [10] K. Kim, M. Barni, and H. Z. Tan, Roughness-adaptive 3-D watermarking based on masking effect of surface roughness, *IEEE Trans. on Information Forensics and Security*, vol. 5, no. 4, pp. 721-733, 2010.
- [11] A. M. Molaei, H. Ebrahimnezhad, and M. H. Sedaaghi, Robust and blind 3D mesh watermarking in spatial domain based on faces categorization and sorting, *3D Research*, vol. 7, no. 2, pp. 1-18, 2016.
- [12] J. W. Cho, R. Prost, and H. Y. Jung, An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms, *IEEE Trans. on Signal Processing*, vol. 55, no. 1, pp. 142-155, 2007.
- [13] A. G. Bors, and M. Luo, Optimized 3D watermarking for minimal surface distortion, *IEEE Trans. on Image Processing*, vol. 22, no. 5, pp. 1822-1835, 2013.
- [14] M. M. Soliman, A. E. Hassanien, and H. M. Onsi, A robust 3D mesh watermarking approach using genetic algorithms, *Intelligent Systems*, pp. 731-741, 2015.
- [15] Y. P. Wang, and S. M. Hu, A new watermarking method for 3D models based on integral invariants, *IEEE Trans. on Visualization and Computer Graphics*, vol. 15, no. 2, pp. 285-294, 2009.
- [16] Y. Z. Zhan, Y. T. Li, X. Y. Wang, and Y. Qian, A blind watermarking algorithm for 3D mesh models based on vertex curvature, *Journal of Zhejiang University Science C (Computers & Electronics)*, vol. 15, no. 5, pp. 351-362, 2014.
- [17] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, A comprehensive survey on three-dimensional mesh watermarking, *IEEE Trans. on Multimedia*, vol. 10, no. 8, pp. 1513-1527, 2008.
- [18] P. Cignoni, C. Rocchini, and R. Scopigno, Metro: Measuring error on simplified surfaces, *Computer Graphics Forum*, vol. 17, no. 2, pp. 167-174, 1998.
- [19] K. Wang, G. Lavoue, F. Denis, A. Baskurt, and X. He, A benchmark for 3D mesh watermarking, *Shape Modeling International Conference IEEE Computer Society*, vol. 5, pp. 231-235, 2010.