IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 18, 2018, accepted September 15, 2018, date of publication October 1, 2018, date of current version October 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2872783

A Watermarking Method for 3D Models Based on Feature Vertex Localization

JING LIU[1], YAJIE YANG[1], DOULI MA[1], YINGHUI WANG[1], AND ZHIGENG PAN[2], (Member, IEEE)

[1]Institute of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[2]Digital Media and Interaction Research, Hangzhou Normal University, Hangzhou 310000, China

Corresponding author: Jing Liu (liujing@xaut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61472319, in part by the Natural Foundation of Shaanxi Province Key Project under Grant 2015JZ015, and in part by the Xi'an Science and Technology Project under Grant 2017080CG.

ABSTRACT This paper presents a new blind watermarking scheme for 3-D point cloud models by using feature vertices to locate the embedded watermark. The vertices with larger mean curvature are considered to be feature vertices carrying watermarking information; non-feature vertices are used to build a new coordinate system in which the 3-D point cloud model is classified into bins containing several longitudes. Each bit of the watermarking information is repeatedly embedded into a bin by altering azimuth angles of feature vertices lying in the bin. The non-feature vertices are not exploited to embed the watermarking information, which enables the built coordinate system to avoid the impacts of the embedding of watermark. The choice of the embedding position and redundant embedding provide an optimal balance between imperceptibility and robustness of watermark. Simulation results testify that our watermarking method for 3-D point cloud models has desired robustness against common attack and geometric attack while demonstrating good transparency.

INDEX TERMS 3D point cloud, watermarking, curvature, azimuth angle, redundant embedding.

I. INTRODUCTION

With the increasing use of 3D models in many applications such as medical imaging, computer aided design, virtual reality, and 3D movies, the necessity to protect the copyright and authentication for 3D content is becoming more pressing and crucial [1]. Watermarking technology is considered as an efficient solution to protect the copyright and authentication for3D models [2].

Both 3D point cloud and mesh are basic representations for 3D models. In recent years, a lot of watermarking algorithms for 3D mesh models have been proposed. According to the embedding domain of watermark, the watermarking methods can be divided into two categories: watermarking algorithms involving frequency domain [3]–[5] carry watermarking information by modifying the coefficients of frequency domain; while spatial domain approaches [6]–[12] embed the watermark by altering the model geometry properties or structure connectivity. In [3], Wang *et al.* presented a watermarking framework for 3D sem-iregular mesh, and the embedding of watermark was performed in wavelet coefficients of different resolution levels. Hamidi *et al.* [5] proposed a method that the watermark was inserted by altering the normal vectors of the wavelet coefficients. Zhan *et al.* [11] calculated the root mean square curvature of every vertex in the local window and divided the vertices into bins; the curvature fluctuation values of the vertices of each bin were modulated to insert watermarking information. Jiang *et al.* [12] mapped decimals of the vertex coordinates into integers firstly and then embedded data into the chosen vertices by operating their least-significant bits. These watermarking methods for 3D mesh models have desired good performances and have successfully solved some difficult problems.

However, compared with the research for the mesh model, the attention to the 3D point cloud model has been very few. Ke *et al.* [13] presented a self-similarity based robust spatial watermarking approach for 3D point cloud models. By modifying local vector length of a certain points in each patch, the watermarking information bits were repeatedly inserted. Feng [14] proposed a watermarking scheme for 3D point cloud model by modulating angle quantization. In this algorithm, the model was divided into many patches

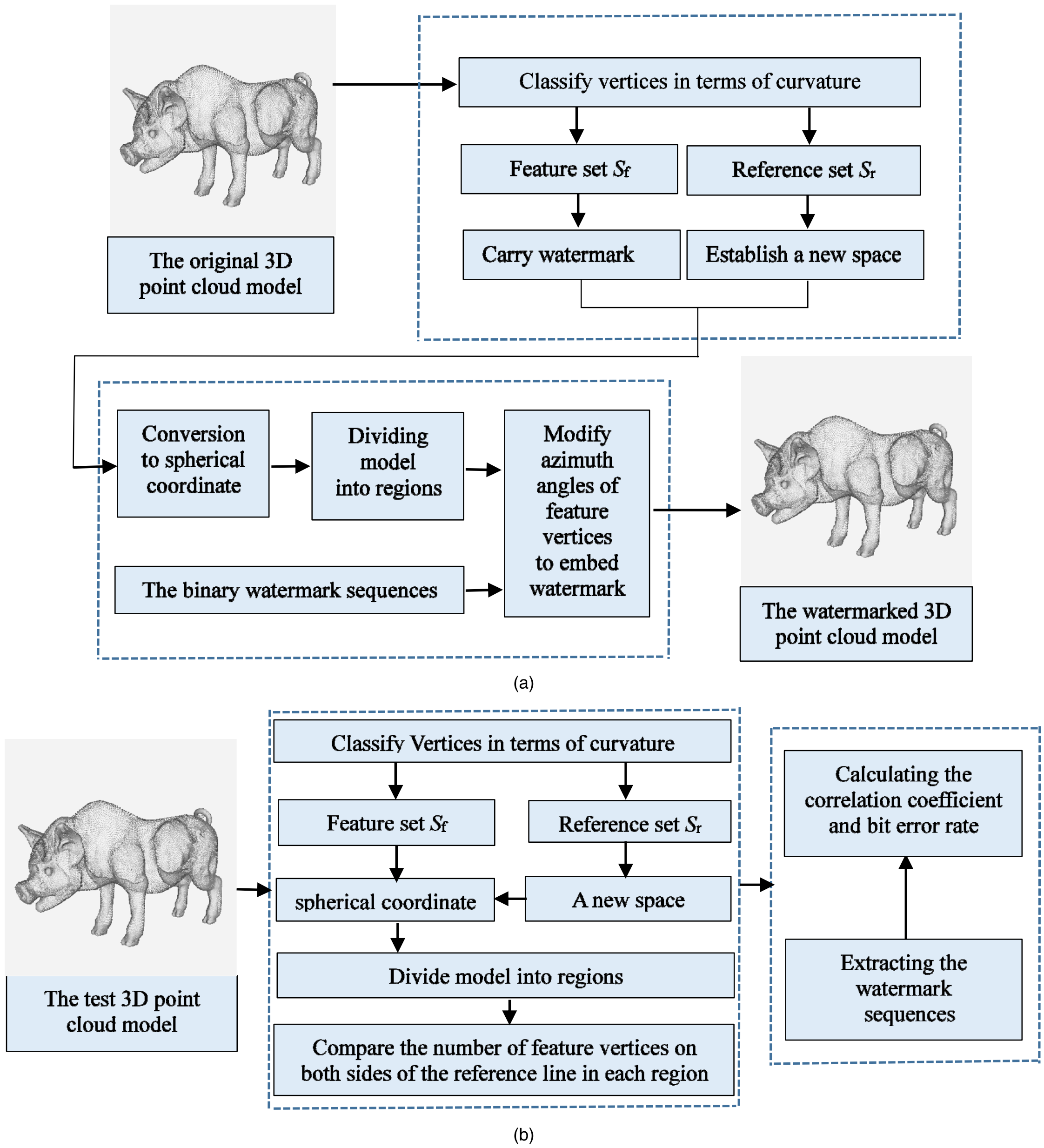


FIGURE 1. Framework of the proposed method: (a) module for watermark embedding; (b) module for watermark extraction.

and watermarking information was embedded into these patches. Cotting *et al.* [15] presented the watermarking algorithm based on point-sampled geometry and spectral analysis. In this method, a hierarchical clustering approach was applied to solve the large data sets efficiently. Ohbuchi *et al.* [16] proposed a method of 3D model defined as a set of unoriented points. A shortcoming of the technique was that it needed to generate a non-manifold mesh for each cluster of the points. In [17], Feng embedded watermarking bits into the 3D model by modifying the average value of the normalized distance of each bin. Agarwal and Prabhakaran [18] derived an order among 3D points at global and local levels according to the cluster tree built from cluster of the 3D points, thus embedding the watermarking information into these ordered 3D points.

Overall, although watermarking algorithms for 3D point cloud model have good imperceptibility and have great potential of practical application, designing a robust point cloud watermarking algorithm is a difficult task. This is because 3D point cloud models have arbitrary topology and no implicit order of data [17], which makes it quite difficult for extracting the watermark from the 3D point cloud models.

In this paper, we propose a novel watermarking method for 3D point cloud models to achieve an appropriate trade-off between transparency and robustness. All the vertices of the 3D cloud point model are divided into two categories

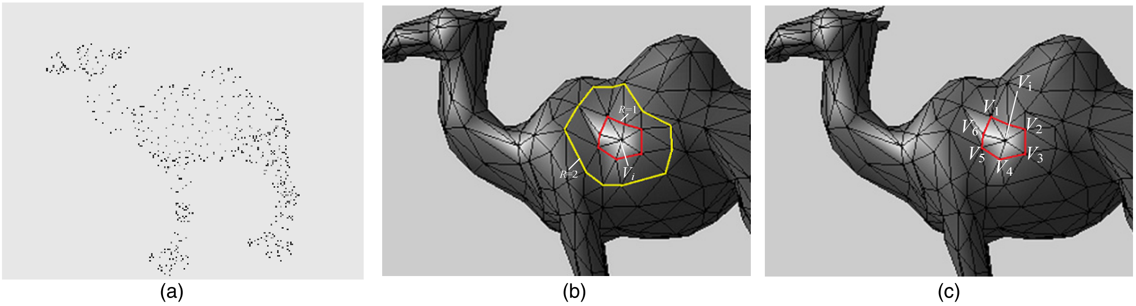


FIGURE 2. (a) 3D point cloud model for Camel; (b) 1-ring neighborhood and 2-ring neighborhood of vertex V_i are enclosed by red and yellow lines, respectively; and (c) All the vertices used to calculate normal direction of vertex V_i.

by using the mean curvature of 1-ring neighborhood: the vertices with larger mean curvature are considered as feature vertices carrying the watermarking information; the vertices with smaller mean curvature are exploited to build a new coordinate system. The model is divided into bins containing several longitudes, and then each bit of the watermarking information is inserted into each bin several times. The main advantages of our method are: (1) The selected feature vertices can carry the large capacity of watermarking information because these vertices come from the bumpy changing regions. The mechanism related to the selecting watermark position ensures the proposed method achieves a optimal tradeoff between imperceptibility and robustness of watermark; (2) The non-feature vertices used to build the new coordinate system don't carry any watermarking information, which eliminates the impacts of the watermarking information on the built coordinate system and keeps the relative positions of the watermarked vertices unchanged in the built coordinate system, thus improving the robustness of the proposed method against geometric attacks; and (3) Every bit of watermarking information is inserted by altering azimuth angles of feature vertices of each bin. The bin embedding and redundant embedding enable the method to overcome the shortcoming of no implicit order of data for 3D point cloud model. Figure 1(a) and Figure 1 (b) show the framework of our watermarking method for 3D point cloud models.

The rest of paper is organized as follows. Section II presents the proposed algorithm in detail. Section III provides the simulation results and analyses about the performances of the proposed method. Finally, we put forward the conclusions and further direction to study in the future in Section IV.

II. THE PROPOSED WATERMARKING ALGORTIHM

A. FEATURE VERTICES SELECTION

The suitable vertices selected to carry watermarking information plays an important role in balancing transparency and robustness of watermark. In theory, the watermarking information embedded into the 3D models should not produce any visual perception for the 3D point cloud models. Disturbing the vertices of the even regions would result in marked distortion of the point cloud model, whereas disturbing the vertices in the abrupt regions of the 3D point cloud model would not lead to perceptible visual distortion. If some vertices lying in the abrupt regions can be caught to carry the watermarking information, it enables the method to have a better imperceptibility. Hence, the embedding vertices, i.e., feature vertices as we called, should be situated in the bumpy regions. We use the mean curvature value of 1-ring neighborhood to select these feature vertices from the model. The set of feature vertices is obtained by the following two steps: (1) determining the neighborhoods of the vertices and (2) computing the mean curvatures of the vertices.

1) DETERMINING THE NEIGHBORHOOD OF A VERTEX

3D point cloud model is a set of data points belonging to the external surface of an object and has no any connectivity information (Figure 2(a)). Obviously, such data should be first converted a 3D surface before calculating their mean curvatures. We use Delaunay triangulation approach to build R-ring neighborhood for the existing vertices of the point cloud model.

For each vertex of the point cloud model, if a Delaunay edge connects vertices V_i and V_j, vertex V_j is a neighbor of vertex V_i. The 1-ring neighborhood of vertex V_i is a collection containing all its neighbors. Each vertex of 1-ring neighborhood of vertex V_i as well as its neighbor of each vertex constitutes its 2-ring neighborhood. The formula is as following:

$$N\left(V_i\right) = \left\{V_j | 0 \leq \left|\overline{V_i V_j}\right| \leq 2, j = 0, 1, \cdots, N\right\} \tag{1}$$

where $\left|\overline{V_i V_j}\right|$ denotes the number of vertices between vertex V_i and vertex V_j, and N represents the number of all the vertices of the model. Figure 2(b) exhibits the1-ring and 2-ring neighborhoods of vertex V_i of the point cloud model.

2) CALCULATING THE CURVATURE

The vertices that are suitable for carrying the watermarking information, i.e., feature vertices, should situate in the abrupt regions. In order to capture such vertices, we need to calculate the normal vector of each vertex. The mean of vectors of all the vertices within 1-ring neighborhood is regarded as the

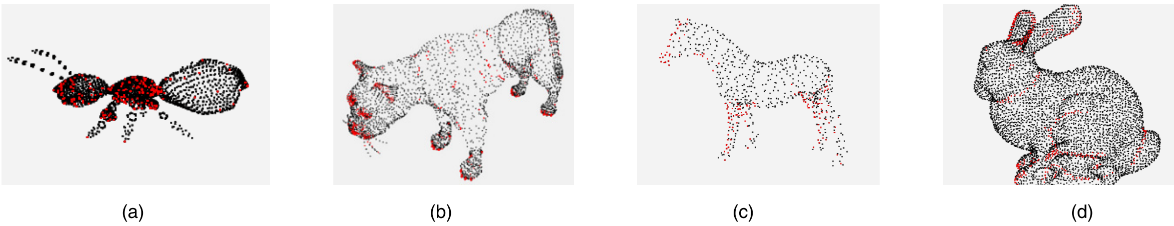


FIGURE 3. The feature vertices marked with red carry the watermarking information and the non-feature vertices marked with black are used to build the coordinate system.

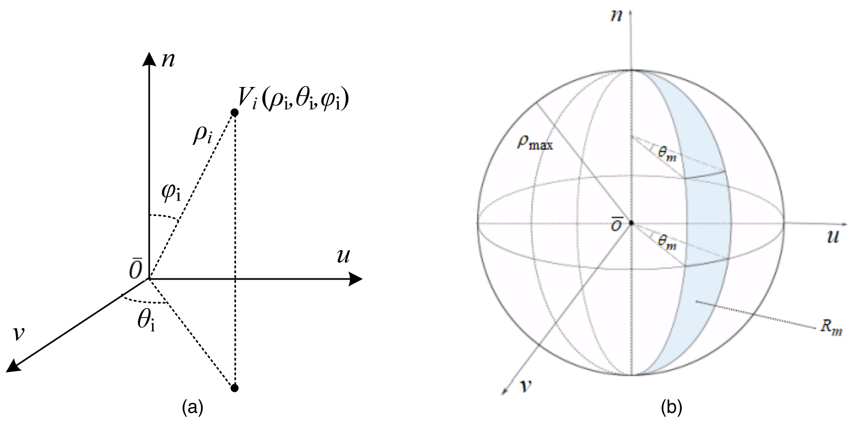


FIGURE 4. (a) The spherical coordinates of vertex V_i; and (b) Dividing the model.

normal vector of vertex V_i. The formula describes as follows.

$$\vec{p}_i = \frac{\sum_{j=1}^{N_i}\left(\vec{V}_i - \vec{V}_j\right)}{N_i} \tag{2}$$

where N_i is the number of all the vertices within 1-ring neighborhood of vertex V_i, $\vec{V}_i$ denotes direction vector of vertex V_i and points to vertex V_i from mass center of the model; and vector $\vec{p}_i$ represents normal vector of vertex V_i. Vertex V_j $(j = 1, 2, \ldots, N_i)$ represents vertices of 1-ring neighborhood of vertex V_i.

After we obtained the normal vector of each vertex of 3D cloud point model, we can determine the abrupt region of the 3D model by using the following Formula:

$$S(V_i) = \sum_{j=1}^{N_i} (\vec{p}_i \cdot \vec{p}_j) = \sum_{j=1}^{N_i} (x_i x_j + y_i y_j + z_i z_j) \tag{3}$$

where (x_i, y_i, z_i) and (x_j, y_j, z_j) correspond to the three coordinate components of two normal vectors $\vec{p}_i$ and $\vec{p}_j$, respectively. The meaning of Formula (3) is that normal vectors of each neighbor V_j of vertex V_i is projected on normal vector of vertex V_i and the obtained value $S(V_i)$ is used to measure the bumpy performance of the 1-ring neighborhood. The neighborhood of vertex V_i with a smaller value $\mathrm{S}(V_i)$ denotes that vertex V_i has a larger mean curvature value, and thus we can classify vertex V_i into a bumpy region. That is, vertex V_i is a feature vertex that is used to carry watermarking information embedding. In Figure 2(c), those vertices marked with white represent all the vertices of 1-ring neighborhood of vertex V_i. The normal vector of vertex V_j (j = 1, 2, 3,4,5 and 6) is first calculated by using Formula (2). Then, we can obtain the value$S(V_i)$ of vertex V_i according to Formula (3).

All the vertices of 3D point cloud model are sorted in terms of their value $S(V_i)$. If the threshold is T and value $S(V_i)$ of vertex V_i is not more than T, vertex V_i is considered as a feature vertex. Such vertices constitute feature set S_f and the rest form reference set S_r (see Figure 3). The vertices belonging to the feature set S_f are used to carry watermarking information while the vertices of the reference set S_r are exploited to establish the coordinate system. Threshold value T is related to the watermarking capacity. If the watermark capacity is M, the model will be divided into M bins each of which contains several longitudes (see Figure 4(b)). Then, threshold value T is considered as the value $\mathrm{S}(V_\mathrm{m})$ of the vertex depending on that more than one feature vertex is found in each divided bin.

B. WATERMARK EMBEDDING

Unlike 2D image watermark carrier, the data order of 3D point cloud models is not implicit. In order to make data order same between the process of watermark embedding and extraction so as to improve robustness against geometric attacks, we establish a new coordinate system and perform watermark embedding and extraction in the established coordinate system. In addition, the 3D model is assumed to be a sphere which is divided into M bins (M represents watermarking capacity) in the established coordinate space. The selected feature vertices within one bin carry the same bit of watermarking information by modifying azimuth angles of these feature vertices. The watermark process consists of establishment of a new coordinate system, dividing the model, and embedding watermark.

1) ESTABLISHING A NEW COORDINATE SYSTEM

The steps in building the new coordinate system are described as following:

- The origin of the new coordinate system:

The origin of the new coordinate system is the mass center of the reference set S_r and can be obtained from

$$V_c = \frac{1}{N_l}\sum_{i}^{N_l} V_i \tag{4}$$

where vertices V_i ($i = 1, 2, \cdots, N_l$) is in the reference set S_r.

- The axes of the new coordinate system:

If x_i, y_i and z_i represent three coordinate components of vertex V_i that is the reference set S_r, and N_l denotes the number of vertices of the reference set S_r, the covariance matrix *Cov* is calculated by using Formula (5).

$$Cov = \begin{bmatrix} \sum_{i=0}^{N_l} x_i^2 & \sum_{i=0}^{N_l} x_i y_i & \sum_{i=0}^{N_l} x_i z_i \\ \sum_{i=0}^{N_l} y_i x_i & \sum_{i=0}^{N_l} y_i^2 & \sum_{i=0}^{N_l} y_i z_i \\ \sum_{i=0}^{N_l} z_i x_i & \sum_{i=0}^{N_l} z_i y_i & \sum_{i=0}^{N_l} z_i^2 \end{bmatrix} \tag{5}$$

Afterwards we proceed eigenvalue decomposition of covariance matrix *Cov*,

$$Cov = U\boldsymbol{H}U^T = U(diag(h_1, h_2, \cdots, h_{N_l}))U^T \tag{6}$$

where $\boldsymbol{H}$ denotes a diagonal matrix that consists of eigenvalues sorted by magnitude, $h_1 > h_2 > \cdots, h_{N_l} h_1 > h_2 > \cdots, h_{N_l}$ and $\boldsymbol{U}$ represents an orthonormal matrix. The $\boldsymbol{i}^{th}$ column of matrix $\boldsymbol{U}$, i.e., the $\boldsymbol{i}^{th}$ eigenvector, corresponds to eigenvalue $\boldsymbol{h}_i$ of matrix $\boldsymbol{H}$. While the first three columns in matrix $\boldsymbol{U}$ are regarded as the directions of u, v and n axes of the new coordinate system. If the original coordinate system is represented as O_{xyz} and the established coordinate system is $\bar{O}_{uvn}$; the orthogonal transformation $M_{O_{xyz}\to\bar{O}_{uvn}}$ can be expressed in Formula (7).

$$M_{O_{xyz}\to\bar{O}_{uvn}} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -V_{cx} \\ 0 & 1 & 0 & -V_{cy} \\ 0 & 0 & 1 & -V_{cz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where the first matrix denotes a rotation matrix, each parameter of which depends on the angles between vector u(v or n) and plane xoz(yoz or xoy) in coordinate system O_{xyz}; the second matrix is a translation matrix, and its three parameters V_{cx}, V_{cy} and V_{cz} are the origin coordinates of the new coordinate system. The coordinates of each vertex in coordinate system $\bar{O}_{uvn}$ is equal to that the coordinates of the corresponding vertex in coordinate system O_{xyz} are multiplied by the matrix $M_{O_{xyz}\to\bar{O}_{uvn}}$. The embedding of watermarking information is executed in the established coordinate system $\bar{O}_{uvn}$.

- Conversion to spherical coordinates

In the proposed method, the watermarking information is embedded by altering azimuth angles of feature vertices. Therefore, the vertices of the 3D point cloud model need to be represented as spherical coordinates. The spherical coordinates (ρ_i, θ_i, ϕ_i) converted from the Cartesian components (V_{iu}, V_{iv}, V_{in}) of vertex V_i are shown in Figure 4(a).

In order to improve performance of the proposed method against scaling attacks, component ρ_i of vertex V_i need also to be normalized and be amplified 10^{num+1} times described as Formula (8).

$$\rho_i^* = \frac{\rho_i - \rho_{min}}{\rho_{max} - \rho_{min}} \times 10^{num+1} \tag{8}$$

where ρ_{max} and ρ_{min} correspond to the maximum and minimum values of component ρ_i of all the vertices in the 3D model, respectively; *num* represents the number of bits of integer of component $\rho_{max}\rho$.

2) MODEL DIVIDING

In order to be still able to locate the watermark position in the process of watermark extraction, 3D point cloud model is divided into M bins with equal range. The number M is the length of watermark information. Let us suppose that 3D model is surrounded by a sphere whose radius is equal to the maximum value ρ_{max} of the spherical coordinate component ρ of the 3D model. The sphere is classified into M bins in the established coordinate system. Each watermark bit is embedded into some feature vertices within one bin by modifying azimuth angles of these feature vertices. The number of embedded times of each bit depends on the number of feature vertices in this bin. These bins are obtained by using following Formula (9).

$$R_m = \{\theta_m | (m-1)\times(360/M) \le \theta \le m \times (360/M)\}\ m \le M, \tag{9}$$

where watermark capacity M determines the total number of bins, R_m represents all vertices located in the m^{th} bin, θ_m represents the azimuth angles of the feature vertices belonging to the m^{th} bin (see Figure 4(b)).

3) MODIFYING AZIMUTH ANGLE TO EMBED WATERMARK

In the process of embedding watermark, the azimuth angle θ_i of vertex V_i in the feature set S_f is modified while keeping ρ_i and φ_i unchanged. Each bit of the watermarking information

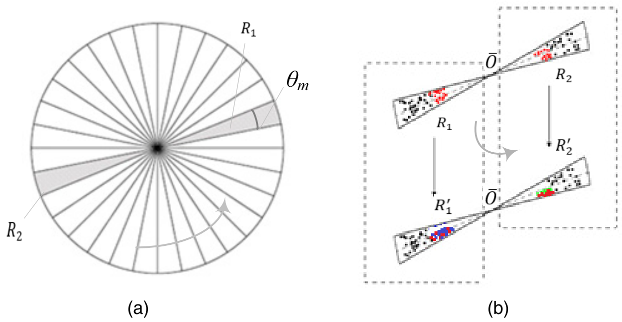


FIGURE 5. **(a) The two sub-regions on the largest cross section, and (b) The vertices are modified to carry the watermarking information.**

is inserted into one bin by modulating azimuth angles of feature vertices using the parameter Δ. If the angle interval is $\boldsymbol{\alpha = 360/M}$ between the arbitrary two bins, the parameter value $\boldsymbol{\Delta = (1/2) \times \alpha}$. If feature vertex $V_i\,(\rho_i, \theta_i, \phi_i)$ is selected to carry a watermark bit and the corresponding watermarked vertex is $\boldsymbol{V_i'}\left(\boldsymbol{\rho_i, \theta_i', \varphi_i}\right)$ component $\boldsymbol{\theta_i'}$ of watermarked vertex $\boldsymbol{V_i'}$ is computed by using

$$\boldsymbol{\theta_{mi}'} = \begin{cases} \boldsymbol{\theta_{mi} + \Delta\,,\ if\ \ w_m = 1} \\ \boldsymbol{\theta_{mi} - \Delta\,,\ if\ \ w_m = -1} \end{cases} \tag{10}$$

Where $\boldsymbol{\theta_{mi}'}$ and $\boldsymbol{\theta_{mi}'}$ represent the azimuth angle of the i^{th} watermarked vertex in the m^{th} bin, corresponding to the watermarked model and the original model. The feature vertex V_i selected from each bin to carry watermarking information should meet the requirement of Formula (11) to ensure that the modified vertex is still in the bin.

$$\begin{aligned} &\boldsymbol{if\ w_m = 1,\ (m-1) \times \alpha \le \theta_i < \left(m - {}^1\!/_2\right) \times \alpha} \\ &\boldsymbol{if\ w_m = -1,\ \left(m - {}^1\!/_2\right) \times \alpha \le \theta_i \le m \times \alpha} \end{aligned} \tag{11}$$

The meaning of Formula (11) is explained as shown in Figure 5. Suppose that region R_1 and region R_2 correspond to the projections of two bins on the largest cross section (see Figure 5(a)). In Figure 5(b), the points of the two regions R_1 and R_2 correspond to vertices within the two bins; the black points represent the non-feature vertices while the red points represent the feature vertices. Regions R_1' and region R_2' are the two watermarked regions into which watermark bit '1' and '-1' were embedded, respectively. If the watermark bit embedded is '1', only the red points (feature vertices) on the upper half of region R_1 are modified to carry the watermarking information (blue points) while the red points (feature vertices) on the lower half of region R_1 remained unchanged; If the watermark bit embedded is '-1', only the red points (feature vertices) on the lower half of region R_2 are modified to carry the watermarking information (green points) while the red points (feature vertices) on the upper half of region R_2 remained unchanged. Notice that the upper part or the lower part is identified by using the right-handed coordinate system. That is, if thumb points n-axis, anticlockwise represents a movement from u-axis to v-axis. The steps to embed the watermark are summarized as following.

Algorithm 1 Overview of Watermark Embedding Procedure

Input: 3D point cloud model, watermark sequence with M bits.
Output: watermarked 3D point cloud model.
Stage 1: *determine feature set S_f and reference set S_r*
for*each vertex V_i* **do**
-Compute the value $S(V_i)$ of its 1-ring neighbourhood by using Formula (3).
-if $S(V_i)$ is not more than T, vertex V_i is in feature set S_f **else** vertex V_i is in reference set S_r.
end

Stage 2: *coordinate transformation*
-Establish a new coordinate system $\bar{O}_{uvn}$ by using vertices of reference set S_r outlined in section2.2.1.
-Compute the spherical coordinates of each vertex in the new coordinate space $\bar{O}_{uvn}$.
Out: new coordinates of each vertex.

Stage 3: d*ivide the 3D model in coordinate system $\bar{O}_{uvn}$*
-create a sphere bounding box for the model.
-divide the model into M bins.

Stage 4: *embed watermark sequence*
for each bit of watermark sequence **do**
-Modify some feature vertices of each bin outlined in section2.2.3.
end

-Coordinate inverse-transformation.
Result: Obtain watermarked 3D point cloud model.

C. WATERMARK EXTRACTION

The watermark extraction is blind, i.e., it does not require the original 3D point cloud model at the extraction stage. The same as the process of embedding the watermark, the curvature values $S(V_i)$ of vertices of the test 3D model are calculated and then all vertices are classified into feature set S_f and reference set S_r. The vertices of reference set S_r are exploited to build the coordinate system while the vertices of feature set S_f are utilized to detect watermark sequence. After converting from Cartesian to spherical coordinates and creating a sphere bounding box for the model, the model is divided into M bins. Each bit of watermark sequence is determined by comparing the number of feature vertices belonging to the upper part and the lower part of one bin. The watermark bit hidden in the m^{th} bin is extracted by following Formula (12).

$$w_m' = \begin{cases} 1, & if\ sum1 \ge sum2 \\ -1, & if\ sum1 < sum2 \end{cases} \tag{12}$$

Where $sum1$ and $sum2$ denote the number of feature vertices lying in the lower part and the upper part in the m^{th} bin, respectively. If the feature vertex situates in the lower part, add to $sum1$ with a value that is equal to1; if the feature vertex is located in the upper part, add 1 to $sum2$. The process of watermark extraction was summarized as following.

Algorithm 2 Overview of Watermark Extraction Procedure

Input: the tested 3D point cloud model, the value M.
Output: the extracted binary sequence w$'$.
1 Stage 1: *determine feature set S_f and reference set S_r*
2 for *each vertex V_i* **do**
3 - Compute the value $S(V_i)$ of its 1-ring neighbourhood.
4 - **if** $S(V_i)$ is not more than T, vertex V_i is in feature set S_f
else vertex V_i is in reference set S_r.
5 end
6
7 Stage 2: *coordinate transformation*
8 -Establish a new coordinate system $\bar{O}_{uvn}$ by using vertices of reference set S_r outlined in section 2.2.1.
9 -Calculate the spherical coordinates of each vertex in new coordinate space $\bar{O}_{uvn}$.
10 Out: new coordinates of each vertex.
11
12 Stage 3: d*ivide the 3D model in coordinate system* $\bar{O}_{uvn}$
13 -Create a sphere bounding box for the model.
14 -Divide the model into M bins.
15
16 Stage 4: *extract binary sequence*
17 for a feature vertex of each bin **do**
18 - Compute watermark bit w_m' using Formula (12).
19 end
20 Result: Obtain binary sequence w'.

Now, we use the model *Horse* (11105 vertices, 22258 facets) to further explain the key steps of the proposed method. The embedded the watermarking information is the size of 16 × 16 bits and the model is divided into 16 × 16 bins, each of which is the same longitude angle ($\alpha = {}^{360}/_{M} = 1°$ $24.4'$, $\Delta = \left({}^{1}/_{2}\right) \times \alpha = 42.2'$). The twentieth watermarking bit is '1' and should be embedded into between longitude angles 26° 43.5′ and 28° 7.9′. There are 13 feature vertices that are selected by using Formula (3). Amongst of them, five feature vertices are located in the lower half of region R_1, the six feature vertices lying in the upper half of region R_1 and the two feature vertices exactly lying in the middle. When we embed the watermark bit '1' into the 20^{th} bin according to Formula (10), component θ of six feature vertices lying in the upper half of region R_1 are all added a Δ value, and thus being moved to the lower half of region R_1.

In the process of the watermark extracting, suppose that we have located to the 20th bin of the model according to the steps outlined in subsection 2.C, the extracted watermark bit is '1' by comparing the number of feature vertices belonging to the upper part and the lower part of the bin ($sum1 = 11 > sum2 = 0$).

Based on the main steps, the number of operations per vertex can be determined from approximately four factors: calculating the value $S(V_i)$ of each vertex; establishing the new coordinate system; embedding and extracting the watermark. Amongst the four parts, the computational complexity of the second part is N^2(N represents the number of vertices of the 3D point-cloud model) while the computational complexity of the other parts is all N. In fact, the computational complexity of the second part is far less than N^2 because only non-feature vertices are used to construct the covariance matrix Cov. Therefore, the computational complexity of the proposed method is O(N^2) and is able to finish the embedding and extracting of the watermark in polynomial time.

III. EXPERIMENTS AND RESULTS

The performance of the proposed method, including invisibility and robustness, will be verified in this section by comparison with the well-known watermarking methods for cloud point model in [13] and [14]. The experiments were executed on many 3D point cloud models from the Princeton Shape Benchmark model library as well as the Stanford model library. Figure 6 shows four of these models: *Ant*, *Leopard*, *Horse* and *Bunny*. We embed 16 × 16 bits of the watermarking information into each model and verify the robustness of the proposed method by performing noise, simplification, cropping or rotation attack on these watermarked models. The root mean square error (RMSE) [18] was used for the invisibility evaluation. The RMSE is given by

$$\boldsymbol{RMSE} = \sqrt{\sum_{i=1}^{N}\left(x_i' - x_i\right)^2 + \left(y_i' - y_i\right)^2 + \left(z_i' - z_i\right)^2} \tag{13}$$

In addition, the invisibility of the watermark is also evaluated by objective metric signal-to-noise ratio (SNR). The SNR is defined by

$$\boldsymbol{SNR} = 10log_{10}\left(\frac{\sum_{i=1}^{N}\left(x_i^2 + y_i^2 + z_i^2\right)}{\sum_{i=1}^{N}((x_i' - x_i)^2 + (y_i' - y_i)^2 + (z_i' - z_i)^2)}\right) \tag{14}$$

where N represents the number of all the vertices of the 3D point cloud model, $\left(x_i,y_i,z_i\right)$ ***and*** $\left(x_i',y_i',z_i'\right)$ denote the coordinates of vertices V_i and V_i', corresponding to the original model and the tested model, respectively. Lower RMSE and higher SNR values indicate smaller degradation of the 3D model caused by the embedding of watermark. We use the correlation coefficient ($Corr$) to measure the robustness of the proposed method. The correlation coefficient ($Corr$), i.e., the similarity between the extracted watermark sequence w' and the embedded sequence w, was given as

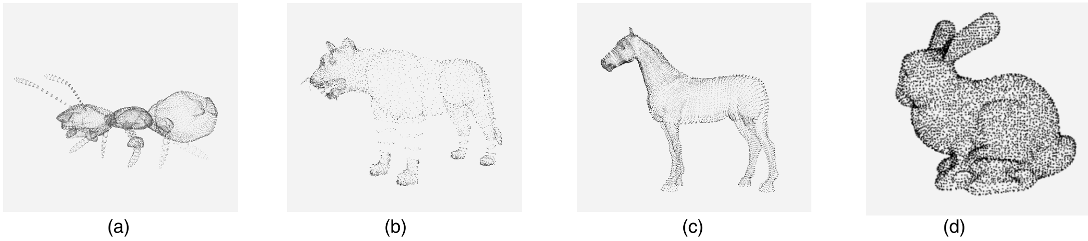


FIGURE 6. **Original illumination models (a) Ant, (b) Leopard, (c) Horse, and (d) Bunny.**

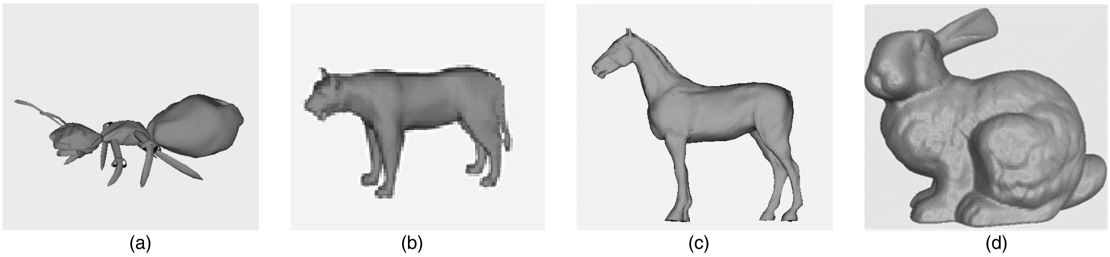


FIGURE 7. **Watermarked models: (a) Ant, (b) Leopard, (c) Horse, and (d) Bunny.**

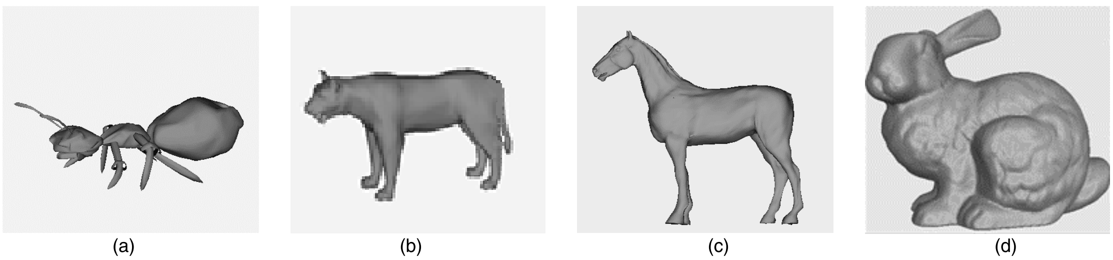


FIGURE 8. **Original models: (a) Ant, (b) Leopard, (c) Horse, and (d) Bunny.**

following.

$$\boldsymbol{Corr}\left(\boldsymbol{w},\boldsymbol{w}'\right)=\frac{\sum_{i=1}^{M}w_i\times w_i'}{\sqrt{\sum_{i=1}^{M}\left(w_i'\right)^2}\times\sqrt{\sum_{i=1}^{M}\left(w_i\right)^2}}\tag{15}$$

The value of correlation coefficient is in a range of $0 \sim 1$. In addition, we also used the bit error rate (*BER*) obtained under some attacks to compare robustness of the proposed method. BER means the error rate of the extracted watermark bits compared with the original watermark bits.

A. IMPERCEPTIBILITY COMPARISONS OF THE METHODS

The four models carrying watermarking information with our method are shown in Figure 8. Some vertices whose curvatures of 1-ring neighbourhoods are larger were selected from each bin to carry the watermarking information. The watermarking information is inserted by modulating components θ of feature vertices while remaining other two components ρ and φ unchanged. These strategies are favorable to improving transparency of the watermark. As seen in Figure 7 and Figure 8, there is no perceptible visual differences between the watermarked models and the original 3D models. Comparing the two values of RMSE and SNR shown in Table 1, the values obtained using the proposed method are lower and higher, respectively, than the corresponding results obtained from the two methods in [13] and [14].

In addition, we also notice that the transparency of the watermark is related with the percentage of altered vertices in the 3D models. For watermarks of the same capacity, if the model contains plenty of vertices, it means that the occupying percentage of watermarked vertices is lower, and thus the transparency of the watermark is better. For four 3D cloud models shown in Figure 6, model*Bunny* has the greatest number of vertices and model *Ant* has the least number of vertices. Table 1 shows that, for each method, the index values of RMSE and SNR obtained from model *Bunny* are all the best whereas the values obtained from model *Ant* are the worst. However, whether for model *Bunny* with abundant vertices or model *Ant* with sparse vertices, the RMSE values obtained using the proposed method are less than 69×1.0^{-4} and the SNR values are greater than 40dB. This illustrates that our method can guarantee the visual quality of the watermarked 3D point cloud model under the condition of meeting the requirement of the watermarking capacity.

TABLE 1. Imperceptibility comparisons of the methods in terms of RMSE × 10^{-4} and SNR.

Methods / Models	Method in [13]		Method in [14]		Our proposed method	
	RMSE	SNR(dB)	RMSE	SNR(dB)	RMSE	SNR(dB)
Ant	72	36.9	71	37.5	68	40.9
Leopard	66	38.5	68	39.1	64	43.2
Horse	59	42.9	62	44.6	56	45.6
Bunny	47	45.4	48	46.4	42	47.7

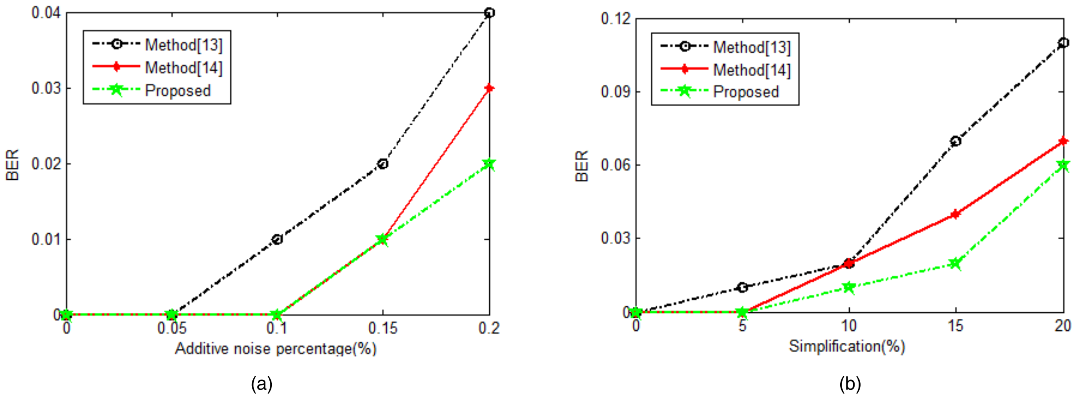


FIGURE 9. Comparison of the BER: robustness against noise and simplification.

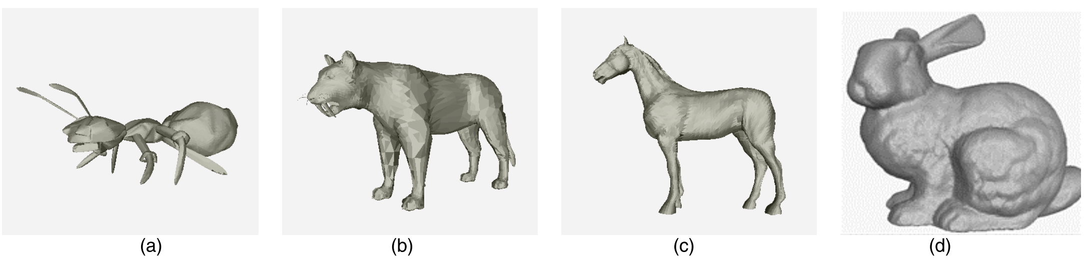


FIGURE 10. The models attacked by simplification (20% reduction): (a) Ant, (b) Leopard, (c) Horse, and (d) Bunny.

B. ROBUSTNESS COMPARISONS OF THE METHODS

Robustness is an important evaluation criterion of 3D point cloud watermarking algorithm. The robustness comparisons of the three methods were executed when embedding the watermark of the same capacity (16 × 16 bits). The test results shown in Table 2, Figure 9 and Figure 11 indicate the performances of these methods to resist attacks such as noise, simplification, cropping, and rotating. Table 2 exhibits the values Corr obtained from each method when the four 3D point cloud models were subjected to different attacks while Figure 9 and Figure 11 show the BER average values.

- Additive noise attack

The second row in Table 2 shows *Corr* values obtained from the watermarked 3D point cloud models attacked by Gaussian noise. Figure 9(a) displays the BER values obtained from the three methods when the noise strength varies from 0.05% to 0.2%. As can be seen from the experimental results, the three methods all have a very good robustness when attacked by weak noise attacks; however, with the increase of attack strength, methods in [13] and [14] achieve results inferior to the results obtained from our proposed method. In the proposed method, the feature vertices are chosen from the 3D point cloud models in light of the mean curvature values of the 1-ring neighbourhoods. The embedding of watermark is performed by modifying azimuth angles θ of these feature vertices. The noise attacks have few impacts on the mean curvature of 1-ring neighbourhood of each vertex. Thus, the method is able to still locate the embedding positions of the watermark so as to improve the robustness against the noise attack.

- Simplification attack

Simplification is a process of removing some vertices while maintaining the shape of the model. The attack is not arbitrary, and its intensity should be limited within a certain

TABLE 2. The values *Corr* of four 3D models when subjected to different attacks.

Attacks		Method	Models				Average
			Ant	Leopard	Horse	Bunny	
Additive noise	0.05%	Method in [13]	0.9312	0.9388	0.9467	0.9579	0.9437
		Method in [14]	0.9349	0.9485	0.9496	0.9584	0.9479
		Proposed method	0.9497	0.9513	0.9567	0.9648	**0.9556**
	0.1%	Method in [13]	0.7534	0.7547	0.7585	0.7783	0.7612
		Method in [14]	0.7546	0.7691	0.7614	0.7792	0.7661
		Proposed method	0.7692	0.7741	0.7754	0.7802	**0.7747**
	0.2%	Method in [13]	0.5489	0.6010	0.6324	0.6562	0.6096
		Method in [14]	0.5592	0.5987	0.6493	0.6716	0.6197
		Proposed method	0.5675	0.6143	0.6539	0.6792	**0.6287**
Simplification	5%	Method in [13]	0.9257	0.9417	0.9284	0.9397	0.9339
		Method in [14]	0.9321	0.9378	0.9288	0.9543	0.9383
		Proposed method	0.9346	0.9459	0.9367	0.9573	**0.9436**
	10%	Method in [13]	0.8416	0.8499	0.8496	0.8728	0.8535
		Method in [14]	0.8387	0.8493	0.8598	0.8754	0.8558
		Proposed method	0.8474	0.8546	0.8613	0.8831	**0.8616**
	15%	Method in [13]	0.6428	0.6589	0.6527	0.6874	0.6605
		Method in [14]	0.6472	0.6591	0.6586	0.6685	0.6584
		Proposed method	0.6531	0.6601	0.6613	0.6728	**0.6618**
	20%	Method in [13]	0.4623	0.4796	0.4849	0.5007	0.4819
		Method in [14]	0.4658	0.4726	0.4867	0.4976	0.4807
		Proposed method	0.4754	0.4832	0.4936	0.5016	**0.4885**
Rotation	10º	Method in [13]	0.8938	0.9406	0.9426	0.9499	0.9317
		Method in [14]	0.8994	0.9312	0.9417	0.9515	0.9310
		Proposed method	0.9195	0.9326	0.9503	0.9527	**0.9388**
	20º	Method in [13]	0.7765	0.78341	0.7834	0.8152	0.7896
		Method in [14]	0.7692	0.7734	0.7809	0.8122	0.7839
		Proposed method	0.7805	0.7921	0.8109	0.8452	**0.8072**
	30º	Method in [13]	0.4936	0.4895	0.4989	0.5203	0.5006
		Method in [14]	0.4884	0.4902	0.5053	0.5128	0.4992
		Proposed method	0.5184	0.5243	0.5268	0.5479	**0.5294**
Cropping	5%	Method in [13]	0.8594	0.8690	0.8874	0.9127	0.8821
		Method in [14]	0.8518	0.8677	0.8798	0.9228	0.8805
		Proposed method	0.8673	0.8795	0.8973	0.9246	**0.8922**
	10%	Method in [13]	0.7098	0.7295	0.7669	0.8344	0.7602
		Method in [14]	0.7106	0.7312	0.7799	0.8397	0.7654
		Proposed method	0.7154	0.7307	0.7825	0.8351	**0.7659**
	20%	Method in [13]	0.4876	0.4965	0.5315	0.5609	0.5191
		Method in [14]	0.4588	0.4801	0.5296	0.5549	0.5059
		Proposed method	0.4564	0.4798	0.5247	0.5318	**0.4982**

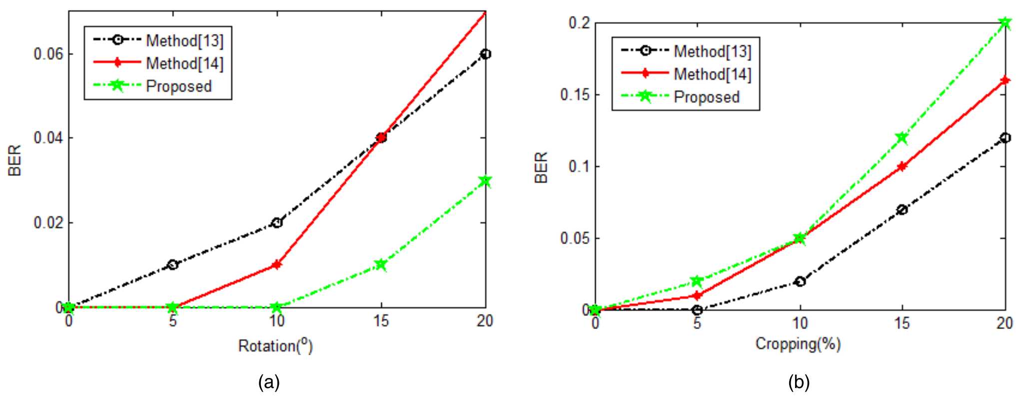


FIGURE 11. Comparison of the BER: robustness against rotation and cropping.

range. If the attack strength makes the 3D point cloud model severely distorted, it is not necessary to protect and authenticate the copyright of the models because they haven't had any practical application values. From the third row in Table 2 and Figure 9 (b) it can be seen that, when vertex reduction does not cause any perceptible deformation of the models, our method can successfully extract the watermarking information; for example, when the simplified vertices vary from 5% to 15%, our method achieves the objective values (*Corr* and *BER*) superior to the results in [13] and [14]; however, when the number of vertices simplified is more than 20% which leads to causing marked visual degradations of the models (Figure 10), the robustness of the three methods is not very well(*Corr*<0.5). Our method embeds the watermark bit into more than one feature vertex in each bin, i.e. redundant embedding. When some vertices in one bin are simplified, the watermark bit is still extracted from the rest of this bin. Therefore, the proposed method has the relatively better resistance to simplification.

- Rotation attack

We investigate the robustness against geometric attack such as rotation transformations. In the proposed method, we exploit the vertices of reference set S_r to build an invariant coordinate system and choose the vertices of feature set S_f to carry the watermarking information. The embedding of the watermarking information has no any impacts on the vertices of reference set S_r. Therefore, compared with the original coordinate system, the established space is invariant and robust. From another point of view, if the attack amplitudes of the rotation transformations cause the changes of relative position of the vertices of reference set S_r, the models would be severely distorted and lose their practical application value. It is also no necessity to protect and authenticate the copyright of such models. If the rotation transformations do not affect the relative positions of the vertices in the established coordinate system, watermark extraction and embedding are conducted in the same coordinate system, and therefore, geometric attacks such as a rotation, scaling, or translation transformation does not obviously weaken the performance of the proposed method. The fourth row in Table 2 and Figure 11(a) demonstrate the good robustness of the proposed method against rotation attacks.

- Cropping attack

Cropping is the removal of some parts from a 3D point cloud model, which is considered as an attack extremely difficult to resist. The fifth row in Table 2 and the graphic comparison in Figure 11(b) show the performance of the proposed method against cropping attacks. When attacked by a slight cropping attack, the objective values (*Corr* and *BER*) obtained from our proposed method are similar to the results of methods in [13] and [14]; as the amplitude of the cropping attack increases, the performance of our method doesn't outperform the results of the other two methods. In the proposed method, each bit of the watermarking information is embedded into a bin of the point cloud model several times, i.e. redundant embedding. This measure helps to successfully extract the watermark information during a cropping attack. However, more strong attacks are bound to make lots of watermarked vertices lose, thus potentially resulting in considerable disturbance in watermark extraction. Therefore, the robustness of our method is limited in the case of strong cropping attacks.

IV. CONCLUSIONS

In this study, we propose a new blind 3D point cloud watermarking algorithm. First, the vertices are classified into two categories according to the mean curvature of 1-ring neighborhood. The vertices with smaller mean curvature are exploited to build the invariant coordinate system; the vertices with larger mean curvature are considered as feature vertices which usually possess the abrupt geometric property of the 3D point cloud models. The model is separated into bins, each of which contains several longitudes. The feature vertices of a bin all carry the same one bit of watermarking information. Non-feature vertices are not affected by the embedding of watermark. Therefore, the coordinate system established by these non-feature vertices would not change due to the embedding of the watermark. As a result of which, it ensures that watermark extraction and watermark embedding are performed in the same coordinate system. The major findings of our study are as follows: (1) Those vertices carrying the watermarking information are selected from the bumpy changing regions of the 3D point cloud models, thus being helpful for the improvement of the transparency of the proposed method; (2) The building of the coordinate system helps to locate the watermarked position because that the relative positions of the watermarked vertices are unchanged in the established coordinate system, thus improving the robustness against geometric attacks; and (3) Region embedding and redundant embedding make the watermark extraction independent of connectivity information of 3D point cloud models, thus significantly enhancing the performances of the watermarking method for 3D point cloud models. Based on the experiments above, the proposed method has good watermark transparency and improves the robustness performance of 3D point cloud watermarking algorithm for resisting geometric attacks such as cropping simplification, additive noise, and rotation transformation.

In the future, the proposed method will be extended to the following two orientations: (1) designing a more effective feature vertices extraction pattern for carrying watermarking information, such as geometric feature framework of 3D model [19]; and (2) combining a powerful learning algorithm [20] to improve robustness of geometric attacks, especially improving robustness against strong combined attacks.

REFERENCES

[1] X. Rolland-Nevière, G. Doërr, and P. Alliez, "Triangle surface mesh watermarking based on a constrained optimization framework," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 9, pp. 1491–1501, Sep. 2014.

[2] G. N. Pham, S.-H. Lee, O.-H. Kwon, and K.-R. Kwon, "A 3D printing model watermarking algorithm based on 3D slicing and feature points," *Electronics*, vol. 7, no. 2, p. 23, Feb. 2018.

[3] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, "Hierarchical watermarking of semiregular meshes based on wavelet transform," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 620–634, Dec. 2008.

[4] J.-U. Hou, D.-G. Kim, and H.-K. Lee, "Blind 3D mesh watermarking for 3D printed model by analyzing layering artifact," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2712–2725, Nov. 2017.

[5] M. Hamidi, M. El Haziti, H. Cherifi, and D. Aboutajdine, "A robust blind 3-D mesh watermarking based on wavelet transform for copyright protection," in *Proc. ATSIP*, Fez, Morocco, May 2017, pp. 1–6.

[6] J. Liu, Y. Wang, Y. Li, R. Liu, and J. Chen, "A robust and blind 3D watermarking algorithm using multiresolution adaptive parameterization of surface," *Neurocomputing*, vol. 237, pp. 304–315, May 2017.

[7] Q. S. Ai, Q. Liu, Z. D. Zhou, L. Yang, and S. Q. Xie, "A new digital watermarking scheme for 3D triangular mesh models," *Signal Process.*, vol. 89, no. 11, pp. 2159–2170, Nov. 2009.

[8] Y. Wang, J. Liu, Y. Yang, D. Ma, and R. Liu, "3D model watermarking algorithm robust to geometric attacks," *IET Image Process.*, vol. 11, no. 10, pp. 822–832, Nov. 2017.

[9] X. Feng, W. Zhang, and Y. Liu, "Double watermarks of 3D mesh model based on feature segmentation and redundancy information," *Multimedia Tools Appl.*, vol. 68, no. 3, pp. 497–515, Feb. 2014.

[10] S.-M. Mun, H.-U. Jang, D.-G. Kim, S. Choi, and H.-K. Lee, "A robust 3D mesh watermarking scheme against cropping," in *Proc. IC3D*, Liege, Belgium, Dec. 2015, pp. 1–6.

[11] Y.-Z. Zhan, Y.-T. Li, X.-Y. Wang, and Y. Qian, "A blind watermarking algorithm for 3D mesh models based on vertex curvature," *J. Zhejiang Univ. Sci. C*, vol. 15, no. 5, pp. 351–362, May 2014.

[12] R. Jiang, H. Zhou, W. Zhang, and N. Yu, "Reversible data hiding in encrypted three-dimensional mesh models," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 55–67, Jan. 2018.

[13] Q. Ke and D.-Q. Xie, "A self-similarity based robust watermarking scheme for 3D point cloud models," *Int. Inf. Inst.*, vol. 16, no. 6, pp. 287–291, Jun. 2010.

[14] X. Feng, "A new watermarking algorithm for point model using angle quantization index modulation," in *Proc. NCEECE*, Xi'an, China, 2016, pp. 962–968.

[15] D. Cotting, T. Weyrich, M. Pauly, and M. Gross, "Robust watermarking of point-sampled geometry," in *Proc. Shape Modeling Appl.*, Genova, Italy, Jun. 2004, pp. 233–242.

[16] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "Watermarking a 3D shape model defined as a point set," in *Proc. Int. Conf. Cyberworlds*, Tokyo, Japan, Nov. 2004, pp. 392–399.

[17] X. Feng, "A watermarking for 3D point cloud model using distance normalization modulation," in *Proc. ICCSNT*, Harbin, China, Dec. 2015, pp. 1449–1452.

[18] P. Agarwal and B. Prabhakaran, "Robust blind watermarking of point-sampled geometry," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 36–48, Mar. 2009.

[19] Y. Wang *et al.*, "Ridge-valley-guided sketch-drawing from point clouds," *IEEE Access*, vol. 6, pp. 13697–13705, Mar. 2018.

[20] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Comput. Graph. Forum*, vol. 17, no. 2, pp. 167–174, 1998.

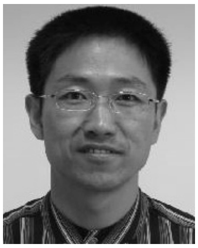
[21] X. Gao, C. Zhang, Y. Huang, and Z. Deng, "A robust high-capacity affine-transformation-invariant scheme for watermarking 3D geometric models," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 8, no. 2S, p. 34, Sep. 2012.

[22] Z. Su, L. Zhou, W. Li, Y. Dai, and W. Tang, "Topology authentication for CAPD models based on Laplacian coordinates," *Comput. Graph.*, vol. 37, no. 4, pp. 269–279, Jun. 2013.

JING LIU received the Ph.D. degree from the Xi'an University of Technology, Xi'an, China, in 2017. She is currently an Associate Professor with the Xi'an University of Technology. Her current research interests include digital watermarking, image processing, and 3-D model representation.

YAJIE YANG received the B.S. degree in information management and information system from the Gansu Political Science and Law Institute, Gansu, China, in 2016. She is currently a Graduate Student in computer software and theory with the Xi'an University of Technology. Her research interests include 3-D watermarking and 3-D model representation.

DOULI MA received the B.S. degree in computer science and technology from Shangluo University, Shanxi, China, in 2016. She is currently a Graduate Student in computer application technology with the Xi'an University of Technology. Her research interests include digital image watermarking and pattern recognition.



YINGHUI WANG received the Ph.D. degree from Northwest University, Xi'an, China, in 2002. From 2003 to 2005, he was a Post-Doctoral Fellow with Peking University, Beijing, China. He is currently a Professor with the Institute of Computer Science and Engineering, Xi'an University of Technology, China. His research interests include image analysis and pattern recognition.

ZHIGENG PAN (M'11) received the bachelor's and master's degrees from the Computer Science Department, Nanjing University, in 1987 and 1990, respectively, and the Ph.D. degree from Zhejiang University in 1993. Since 1996, he has been with the State Key Lab of CAD&CG, Zhejiang University, as a Full Professor. He is currently the Director of the DMI Center, Hangzhou Normal University. His research interests include virtual reality/virtual environment, multi-resolution modeling, real-time rendering, distributed VR, visualization, and multimedia and image processing. He is also a member of the SIGGRAPH, Eurographics, and a Senior Member of the China Image and Graphics Association.