

Holographic Blind Watermarking Algorithm of a Three-Dimensional Mesh Model Based on the QR Decomposition

Wenju Wang^{1,*}, Liuji Sun¹, Qin Yang¹ and Zhang Xuan²

¹University of Shanghai for Science and Technology, Shanghai 200093, China; ²Shanghai Conservatory of Music, Shanghai, China

Abstract: We propose a holographic blind watermarking algorithm of the three-dimensional mesh model based on the QR decomposition. First, the three-dimensional model is pre-processed by moving the model's center to the coordinate origin and performing the PCA analysis. Second, the model's geometric feature matrix is constructed by using the distance from the three-dimensional mesh model's vortex to the model's center in the column coordinate system. Then the watermarking information generated by the holographic encrypted technology is embedded into the normalized and blocked model's geometric feature matrix based on the QR decomposition. Finally, the watermark is embedded into the three-dimensional mesh model via the inverse QR decomposition and inverse normalization operation. The experimental results indicate that this algorithm can resist translation, rotation, scaling, vertex reordering, noise, and cropping attacks, which shows it has a good robustness. This method can effectively protect the three-dimensional mesh model's intellectual property.

Keywords: Digital watermark, Holographic encryption, QR decomposition, Three-dimensional mesh mode.

1. INTRODUCTION

With the increasing popularity of network exchanges and high-speed development of electronic commerce, effectively protecting three-dimensional models in digital publications has become an area of significant interest. The digital watermark of a three-dimensional model is one important means of effectively protecting a three-dimensional model, since it can prevent against infringement and piracy in information exchanges. In addition, digital watermarks can help standardize the digital market and promote the healthy and sustainable development of information [1, 2]. The three-dimensional mesh digital watermarking algorithm is divided into the space domain and transformation domain according to the working domain of the watermarking algorithm.

1.1. Transformation Domain Algorithm

In 1999, Praun [3] extended the spread spectrum technology of a 2D digital watermark scheme into a digital watermark algorithm for the three-dimensional model. Although this method is robust, it has some weaknesses. For example, this method requires many calculations, is fully independent of the common mesh handling and editing algorithm, and performs multi-resolution decomposition for the model. In 2001, Ohbuchi obtained the Laplace operator based on the mesh topology and realized a three-dimensional mesh model watermarking algorithm by analyzing the mesh's pseudo-spectrum [4], but the amount of embedded data was limited. In 1998, Wang performed wavelet decomposition for the three-dimensional mesh model and embedded the robust watermark, vulnerable watermark, and high-capacity watermark

into the proper wavelet resolution. This is a blind watermarking algorithm [5], but it is not very robust. In 2008, Liu transformed the original mesh into the frequency domain by using the stream harmonic wave transformation and then embedded the watermark, but the shape information can be lost during cropping attacks [6]. In 2009, Konstantinides embedded the watermarking information into the spherical harmonic coefficient. However, this method fully depends on the global match of the mesh, so it cannot withstand a cropping attack [7].

The frequency domain watermarking algorithm is very complicated and the amount of embedded data is small. In addition, the three-dimensional model lacks a natural parameterization method, so it is difficult to realize the direct frequency domain decomposition for the three-dimensional model. Compared to the frequency domain algorithm, the space domain algorithm has the advantage of a simple embedding method and can embed a large amount of data, so it is valuable in actual practice.

1.2. Space Domain Algorithm

In 1997, Ohbuchi published the first paper on a three-dimensional mesh digital watermark [8] and then proposed several watermarking algorithms for the triangle mesh based on concepts such as the mesh replacement, topology replacement, and visible mode [9-11]. Typical algorithms include the Triangle Similarity Quadruple (TSQ) algorithm and Tetrahedral Volume Ratio (TVR) algorithm. These algorithms are very sensitive to the noise and topology change. In 2005, Zafeiriou proposed to embed the watermarking information by changing the vortex coordinate in a spherical coordinate system, but this algorithm cannot resist the affine transformation [12]. In 2008, Salman embedded the watermarking information by using the normal vector of the three-dimensional model, but this algorithm requires the detailed internal organization information for the three-dimensional model during the extraction of the

*Address correspondence to this author at University of Shanghai for Science and Technology, 516 Jun Gong Road, Shanghai, China, 200093; Tel: 1-363-667-5266; E-mail: wangwenju666@163.com

watermarking information, so this algorithm is only applicable to the embedment of a private watermark [13]. In 2009, Wang and Hu from Tsinghua University proposed a space domain half-vulnerable blind watermarking algorithm based on the integral invariant that can resist vortex disorder, RST transformations, and small noise attacks, but this algorithm can easily lead to the whole distortion of the original three-dimensional model [14]. In 2009, Qing from Wuhan University of Technology divided the three-dimensional model into multiple Voronoi patches by the three-dimensional model feature points and embedded the watermarking information. This algorithm has a better anti-cropping performance, but it spends much time to divide the Voronoi patches by the selected feature points in advance, so it is difficult to operate and implement [15]. In 2011, Ho embedded the watermark by using the vertex information of the three-dimensional model's surface and the progressive mesh compression to protect the three-dimensional detailed model's watermark, but the histogram bin shifting technology changes the geometric information of the vortexes [16]. Liu proposed a non-blind watermarking algorithm to construct the spherical coordinate mapping matrix and apply the SVD decomposition for constructing a more stable watermarking element to embed the watermark. This algorithm can resist noise and cropping, which improves the robustness of this space domain watermarking algorithm, but its calculation is slow due to the time complexity $O(N^3)$ of the SVD decomposition [17].

The above space domain watermarking algorithms are intuitive and can embed massive amounts of information, but it is difficult to realize the blind watermark with them and their robustness is poor. These algorithms are restricted in actual applications. In addition, most current three-dimensional mesh digital watermark algorithms use the pseudorandom sequence [3, 4, 12, 15, 17] or Arnold transformation to realize the scrambling encryption of the watermarking images [14] as the watermarking information. The pseudorandom sequence has a low linear complexity and the Arnold transformation encryption is susceptible to an exhaustive attack. The above algorithms can be easily decrypted and imitated, so they cannot accurately perform the anti-counterfeiting authentication.

Here, we propose a holographic digital watermark algorithm based on the QR decomposition for a three-dimensional mesh model that makes decryption and imitation difficult. This algorithm can resist translation, rotation, scaling, vertex reordering, noise, and cropping attacks. It has a high robustness and practical application value.

Section 2 of this paper gives some preliminary knowledge. Section 3 introduces the holographic digital watermark algorithm based on the QR decomposition for a three-dimensional mesh model. Section 4 gives the experimental results and analysis. Finally, our conclusions appear in section 5.

2. PRELIMINARY KNOWLEDGE

2.1. Cylindrical Coordinates

Assume that P is any point in the space and is projected as Q on the oxy plane. (ρ, θ) ($\rho \geq 0, 0 \leq \theta < 2\pi$) indicates

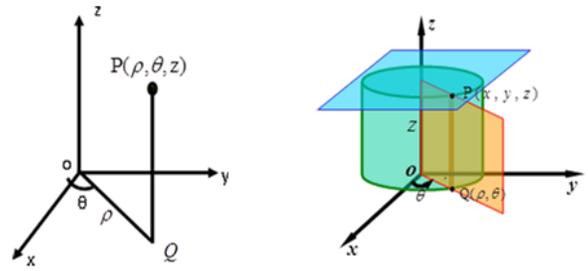


Fig. 1. Cylindrical coordinate system.

the polar coordinate of the point Q on the plane oxy . The position of the point P can be represented with the orderly array (ρ, θ, z) , where $\rho \geq 0, 0 \leq \theta < 2\pi, -\infty \leq z \leq +\infty$. The coordinate system with the above mapping is called the cylindrical coordinate system, and is shown in Fig. 1. The rectangular coordinate (x, y, z) and polar coordinate (ρ, θ, z) of the space point P can be transformed with Equations (1) and (2).

$$\begin{cases} \rho^2 = x^2 + y^2 \\ \theta = \arctg \frac{y}{x} (x \neq 0), 0 \leq \theta < 2\pi \\ z = z \end{cases} \quad (1)$$

$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \\ z = z \end{cases} \quad (2)$$

If ρ is constant in the cylindrical coordinates, the equation expresses a cylindrical surface. If θ is constant, the equation expresses a half plane. If z is constant, the equation expresses a plane.

2.2. QR Decomposition

As the intermediate computing step in the SVD, the time complexity of the matrix QR decomposition is $O(N^2)$ [18]. QR decomposition requires little computational time.

The matrix QR decomposition method includes the *Gram-Schmidt*, *Householder* and *Givens* method. The accumulative rounding error of the *Gram-Schmidt* algorithm is larger than that of the *Givens* method. When the order of the matrix is larger, the *Givens* method will generate a heavy computing workload, so the *Householder* method is frequently used in the QR decomposition.

Assume the matrix $A \in R^{m \times n}$ and $m \geq n$, so the QR decomposition of the matrix A can be expressed as shown in Equation (3).

$$A = Q \cdot R \quad (3)$$

$Q = H_1 H_2 \dots H_{n-1}$, H_i is obtained via the Householder transformation of i th column vector in A . Q is a $m \times m$ matrix with the standard orthogonal vector. $R = H_{n-1} \dots H_2 H_1 A$, where R is the $m \times n$ upper triangle matrix.

Assume that A and Q are expressed as $A = [a_1, a_2, \dots, a_n]$ and $Q = [q_1, q_2, \dots, q_m]$, and $a_i (i=1, 2, \dots, n)$ and $q_j (j=1, 2, \dots, m)$ are the column vector of the matrices A and Q and include the m element. R is expressed as shown in Equation (4):

$$\begin{aligned} R &= Q^T \cdot Q \cdot R = Q^T \cdot A \\ &= [q_1, q_2, \dots, q_m]^T \cdot [a_1, a_2, \dots, a_n] \\ &= \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & r_{mn} \end{bmatrix} \end{aligned} \quad (4)$$

After the QR decomposition, matrix R has an important property. If the columns of matrix A are associated, the absolute values of the elements in the first row of matrix R may be bigger than the corresponding elements in other rows. The bigger values permit bigger change ranges, so the elements of the first row of matrix R are suitable for embedding the watermarking information.

2.3. Digital Holographic Information

The holographic watermarking theory has been only applied to 2D images and has not been explored for three-dimensional images, but it encrypts information by using the double-random phase and has segmentation characteristics (namely, fragments that can also reflect the whole). With the above features, the watermarking technology based on the optical holographic theory provides a new means to solve problems such as easy decryption, accurate anti-counterfeiting authentication, susceptibility to cropping attacks, and easy loss of the original three-dimensional mesh information.

The digital holographic technology used in this paper refers to the method used by Takai and belongs to the Fourier transformation holograph [19].

The watermarking image is defined as $g_{mark}(x, y)$. After the watermarking images are modulated by a random phase template, the modulated images can be expressed as

$$g_0(x, y) = g_{mark}(x, y) \exp[i\phi(x, y)] \quad (5)$$

The 2D phase $\phi(x, y)$ is determined by the Gauss random number. The modulated watermarking image is processed with the Fourier transformation and interferes with the reference light. The strength distribution field generat-

ed by coherence is the Fourier transformation holographic image expected in this paper.

The Fourier transformation of the watermarking images is expressed as follows:

$$G_{mark}(\xi, \eta) = \iint g_0(x, y) \exp[-2\pi i(\xi x + \eta y)] dx dy \quad (6)$$

The reference light is defined as

$$R(\xi, \eta) = R_0 \exp[2\pi i(a\xi + b\eta)] \quad (7)$$

The watermarking images and light field distribution interfere with the reference light, and are expressed as follows:

$$\begin{aligned} H_1(\xi, \eta) &= |G_{mark}(\xi, \eta) + R(\xi, \eta)|^2 \\ &= |G_{mark}(\xi, \eta)|^2 + |R(\xi, \eta)|^2 \\ &\quad + G_{mark}^*(\xi, \eta)R(\xi, \eta) + G_{mark}(\xi, \eta)R^*(\xi, \eta) \end{aligned} \quad (8)$$

In Equation (8), elements 1 and 2 indicate the halo light and center bright point of the Fourier transformation holographic image, which affect the reoccurrence of the watermark and must be removed to obtain Equation (9).

$$\begin{aligned} H(\xi, \eta) &= G_{mark}^*(\xi, \eta)R(\xi, \eta) \\ &\quad + G_{mark}(\xi, \eta)R^*(\xi, \eta) \end{aligned} \quad (9)$$

Equation (9) expresses the digital holograph to use, which records the amplitude and phase of the physical light wave; these waves become the watermarking signals embedded in the home images.

The light field strength distribution can be obtained by the mathematic expression to describe the reconstructed light multiplied with the holographic images after the inverse Fourier transformation. The reconstructed light can be defined as follows: $S(\xi, \eta) = |S(\xi, \eta)| \exp[i\phi_s(\xi, \eta)]$. For the simplest case, $|S(\xi, \eta)| = 1$ and $\phi_s(\xi, \eta) = 0$. The reconstructed images obtained via the inverse Fourier transformation are expressed as follows:

$$g_R(x, y) = \iint H(\xi, \eta) \exp[2\pi i(\xi x + \eta y)] d\xi d\eta \quad (10)$$

Equations (9), (6), and (7) are substituted into Equation (10) to obtain the reconstructed light field:

$$g_R = g_0^*[(x-a), (y-b)] + g_0[-(x+a), -(y+b)] \quad (11)$$

Equation (11) indicates that the original images and conjugated images redisplay on the same plane. (a, b) and $(-a, -b)$ are their center. The position of the redisplayed images can be controlled by a and b .

3. HOLOGRAPHIC DIGITAL BLIND WATERMARK ALGORITHM OF THE MESH MODEL

3.1. Embedding the Watermark

The watermark is embedded as follows:

(1) The generation of the optical holographic encrypted watermarking information occurs first.

To realize the difficult decryption of the watermarking information embedded into the three-dimensional model, we use a binary image $g_{mark}(x,y)$ ($s \times s$ indicates the size) as the watermarking image, which is processed by the algorithm described in section 2.3 and generates the optical holographic encrypted watermarking information $H(\xi,\eta)$. $H(\xi,\eta)$ includes massive amounts of double-precision data, so it is difficult to embed this watermark into the three-dimensional mesh model. Generally, most watermarking information is in the form of binary sequences.

$$w = \{w_i | w_i \in \{0,1\}, 0 \leq i \leq length - 1\} \quad (12)$$

The *length* is the bits of the watermarking information, so $H(\xi,\eta)$ will be transformed into a gray image $H(x,y)$. Each pixel includes 8 bits of data in this gray image. The binary sequence information embedded for the three-dimensional model watermarking is the optical holographic encrypted binary sequence information W_{H_i} ($i=1,2,\dots, length, length = s \times s \times 8$) generated by $H(x,y)$. The optical holographic encrypted watermarking information is generated as shown in Fig. 2.

(2) Pre-processing the three-dimensional mesh model

Any three-dimensional model is composed of the vertex set V and connections among the vertices [6]. For any three-dimensional mesh model O , the number of the vertex is n and the vertex set is $V\{v_i, i=1,2,\dots,n\}$. In the Cartesian coordinate system, the coordinate of the vertex v_i is $v_i(x_i, y_i, z_i)$.

① Calculation for the mesh's center point coordinates

$$v_c(x_c, y_c, z_c) = \frac{1}{n} \sum_{i=1}^n v_i(x_i, y_i, z_i) \quad (13)$$

v_i is the i th vertex of the three-dimensional model in Equation (13).

② The mesh center is moved to the origin of the coordinate. $v_i(x_i, y_i, z_i)$ becomes $v'_i(x'_i, y'_i, z'_i)$ by Equation (14).

$$\begin{cases} x'_i = x_i - x_c \\ y'_i = y_i - y_c \\ z'_i = z_i - z_c \end{cases} \quad (14)$$

③ Principal component analysis

To make the three-dimensional mesh model resist to the rotation attack, we automatically adjusted the model to a unique posture for calibration and pre-processing by using

the PCA [20]. The covariance matrix C_v of the three-dimensional model vertex is shown in Equation (15).

$$C_v = \begin{bmatrix} \sum_{i=1}^n (x'_i)^2 & \sum_{i=1}^n x'_i y'_i & \sum_{i=1}^n x'_i z'_i \\ \sum_{i=1}^n x'_i y'_i & \sum_{i=1}^n (y'_i)^2 & \sum_{i=1}^n y'_i z'_i \\ \sum_{i=1}^n x'_i z'_i & \sum_{i=1}^n y'_i z'_i & \sum_{i=1}^n (z'_i)^2 \end{bmatrix} \quad (15)$$

Three matrix eigenvalues of C_v are calculated and sorted by descending order $\lambda_{max}, \lambda_{mid}, \lambda_{min}$. The corresponding eigenvector is $\eta_{max}, \eta_{mid}, \eta_{min}$. The maximum eigenvalue λ_{max} and corresponding eigenvector η_{max} act as the principal component p_c of the three-dimensional model. The angle α between η_{min} and the y axis is then computed. η_{min} is rotated by α to align with the y axis, which identifies the rotary matrix T_1 . The angle β between η_{mid} and x axis is calculated. η_{mid} is rotated by β to align with the x axis, which identifies the rotary matrix T_2 .

After the three-dimensional model is analyzed with the PCA method, shown as Equation (16), the vertex coordinate of the three-dimensional model can be expressed as $v''_i(x''_i, y''_i, z''_i)$. At this time, the principal component direction p_c of the three-dimensional model overlaps with the z axis, so the three-dimensional model can be adjusted to a unique posture and orientation.

$$v''_i = v'_i \times T_1 \times T_2 \quad (16)$$

④ The vertex $v''_i(x''_i, y''_i, z''_i)$ in the Cartesian coordinate system is transformed into the coordinate $v''_i(\rho''_i, \theta''_i, z''_i)$ $0 \leq \theta''_i < 2\pi$ in the cylindrical coordinate system by Equation (1).

(3) Construction of the mesh's geometric feature matrix

The vertex $V\{v''_i, i=1,2,\dots,n\}$ is sorted by Z in descending order. The z values of the partial vertices are the same, so they are classified into one class as one subset V_{sub} of the vertex set V_i . The number of sub-nets divided from the vertex set V_i by z value is expressed as n_z , $n_z \in [1, n]$. n is the total number of vertices for the 3D mesh model.

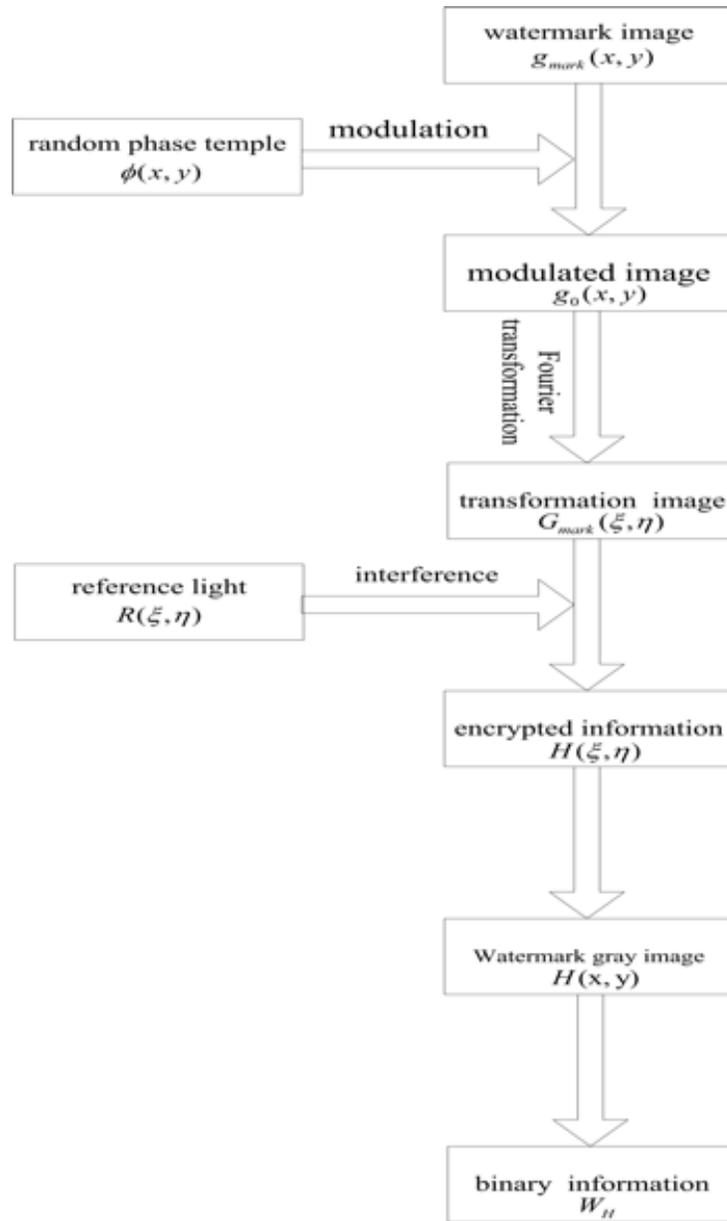


Fig. 2. Generation of the watermarking information.

The vertex set V_i is sorted by θ in ascending order. If the partial vertices have the same θ values, they will be classified into one class as one subset V_{θ_j} of the vertex set V_i . The number of sub-nets V_{θ_j} divided from the vortex set V_i by the θ value is expressed as n_{θ} , $n_{\theta} \in [1, n]$ $j \in [1, n_{\theta}]$. The vertices are sorted by the z value in descending order in each subset V_{θ_j} . If the partial vertices have the same z value, they will be classified into one class as one subset $V_{\theta_{jk}}$ of V_{θ_j} . The distance from the vertex to the model's center in this subset is $d_{v_k} = \sqrt{\rho_{v_k}^2 + z_{v_k}^2}$, $v_k \in V_{\theta_{jk}}$. All vertices in $V_{\theta_{jk}}$ are sorted in

ascending order with d_{v_k} . The number of vertices num_{jk} in each subset $V_{\theta_{jk}}$ is computed and compared to get the vertex number of the maximum subset, namely $\max(num_{jk})$ in V_{θ_j} .

The three-dimensional mesh model in the cylindrical coordinate system can be expressed with the matrix $D[n_z, n_{\theta}]$, $n_{\theta} = \sum_{j=1}^{n_{\theta}} \max(num_{jk})$. $D[i, j]$ indicates the distance $d_{i,j} = \sqrt{\rho_{ij}^2 + z_{ij}^2}$ from the vertex v_{ij} to the model's center, which is identified by the z value in the i th subset divided by the θ value. ρ_{ij} indicates the polar coordinate

radius of this vertex on the plane Oxy . z_{ij} indicates the z value of this vertex in the cylindrical coordinate. If no vertex is available in $D[i,j]$, $d_{i,j} = 0$. The i th row has the same z value and the j th column has same θ values in the matrix $D[n_z, n_\theta]$.

The matrix D can store the d value of $n_z \times n_\theta$ vertices, which is far bigger than the vertex number of the three-dimensional mesh model, so matrix D includes massive amounts of 0 values and is a sparse matrix. The three-dimensional model surfaces are different, so distribution of the 0 values is so different in matrix D that it is difficult to find the rule for further processing. Non-zero d_{ij} values in matrix D are obtained by the row sequence to construct a non-negative real number sequence $\vec{d} = (d_1, d_2, \dots, d_n)$ with the length n for n vertices.

Assume $L = \lfloor \sqrt{n} \rfloor$, and the following vector is defined as

$$D'_K = (d_{(K-1)L+1}, d_{(K-1)L+2}, \dots, d_{KL})^T, \quad (17)$$

where $1 \leq K \leq L$.

The model's geometric feature matrix is obtained:

$$D' = (D'_1, D'_2, \dots, D'_L) = \begin{bmatrix} d'_1 & d'_{L+1} & \dots & d'_{(L-1)L+1} \\ d'_2 & d'_{L+2} & \dots & d'_{(L-1)L+2} \\ \vdots & \vdots & \ddots & \vdots \\ d'_L & d'_{2L} & \dots & d'_{L \times L} \end{bmatrix}. \quad (18)$$

The element in row i and column j of D' is denoted d'_{ij} .

(4) QR decomposition and watermarking

① The normalization of the elements in the geometric feature matrix D' indicates the need to normalize the distances from all vertices to the model's center, namely

$d''_{ij} \in [0, 1]$, using Equation (19).

$$\begin{cases} d'_{\max} = \max \{d'_{ij} \mid d'_{ij} \in D'\} \\ d''_{ij} = \frac{d'_{ij}}{d'_{\max}}, \quad d'_{ij} \in D' \end{cases} \quad (19)$$

② Block division of the matrix

The geometric feature matrix D' is divided into $N_m = \lfloor \frac{s \times s \times 8}{m} \rfloor$ non-overlapping matrix blocks with the

size $s \times s$. $s \times s \times 8$ is the length of the optical holographic encrypted binary sequence information W_{H_i} . m is the row number and column number of the QR decomposition matrix.

③ QR decomposition

QR decomposition is performed for each selected matrix with the size $m \times m$ according to Equation (3).

④ Embedding the watermark

The watermarking information W_{H_i} is embedded by changing the element $r_{11}, r_{12}, \dots, r_{1m}$ in the first row of the matrix R via the following process:

$\min = \min(r_{1j})_{j=1,2,\dots,m}$, $\max = \max(r_{1j})_{j=1,2,\dots,m}$ and Δ refers to the equation (20).

$$\Delta = \frac{\max - \min}{m} \quad (20)$$

We use \min as the starting point, as the quantification interval Δ is used to quantify r_{1j} ($j=1,2,\dots,m$); thus, we obtain Equations (21) and (22).

$$M_j = \left[\frac{(r_{1j} - \min)}{\Delta} \right] \quad (21)$$

$$r_{1j}^* = \min + \begin{cases} M_j \times \Delta, & \text{if } (M_j \text{ is even and } w_{H_i} = 0) \\ (M_j - 1) \times \Delta, & \text{if } (M_j \text{ is odd and } w_{H_i} = 0) \\ (M_j + 1) \times \Delta, & \text{if } (M_j \text{ is even and } w_{H_i} = 1) \\ M_j \times \Delta, & \text{if } (M_j \text{ is odd and } w_{H_i} = 1) \end{cases} \quad (22)$$

In Equation (21), $[\]$ indicates to round a number. In Equation (22), W_{H_i} is the i th optical holographic encrypted binary sequence information.

⑤ Inverse QR decomposition

r_{1j} is replaced with the quantified r_{1j}^* and we perform the inverse QR decomposition by using Equation (23) to get the matrix with the watermark.

$$D^* = Q \times R^* \quad (23)$$

⑥ Repeat

Repeatedly execute steps ③-⑤ until all W_{H_i} watermarking information is embedded into the geometric feature matrix of the three-dimensional model.

⑦ The inverse normalization of the elements in the geometric feature matrix D^* is executed, resulting in Equation (24).

$$d^w_{ij} = d^*_{ij} \times d'_{\max}, \quad d^*_{ij} \in D^* \quad (24)$$

⑧ The cylindrical coordinates of the vertices are transformed into the Cartesian coordinates by Equation (25).

$$\begin{cases} \rho_{ij}^w = \sqrt{(d_{ij}^w)^2 - (z_{ij}^w)^2} \\ x_{ij}^w = \rho_{ij}^w \cos \theta + x_c \\ y_{ij}^w = \rho_{ij}^w \sin \theta + y_c \\ z_{ij}^w = z_{ij}^w + z_c \end{cases} \quad (25)$$

In Equation (25), z_{ij}^w is the value of z when d_{ij}^w is calculated. The coordinates $(x_{ij}^w, y_{ij}^w, z_{ij}^w)$ of the vertex set are denoted v_{ij}^w . The vertex matrix v_i^w is obtained by adjusting v_{ij}^w according to the storage order of the vertices in the 3D model.

⑨ The three-dimensional model is adjusted to the original posture and orientation.

T_2' indicates the rotation matrix with the x axis rotating $(-\beta)$. T_1' indicates the rotation matrix with the y axis rotating $(-\alpha)$ (Equation (26)).

$$v_i^{w'} = v_i^w \times T_2' \times T_1' \quad (26)$$

After performing the above steps, the watermarking data can be embedded into the three-dimensional model with the watermark.

3.2. Extraction of the Watermark Information

The original watermarking image and original three-dimensional model are not required in the watermark extraction algorithm, so this algorithm is blind. The specific watermark extraction process is as follows:

(1) Pre-processing of the mesh model with the watermark

The three-dimensional model with the watermark is pre-processed by using the method mentioned above. The geometric feature matrix is constructed in the cylindrical coordinate system and is normalized to get the matrix D' .

(2) Block processing of the matrix

The geometric feature matrix D' is divided into $m \times m$ non-overlapping matrix blocks.

(3) QR decomposition

QR decomposition is performed for each geometric feature matrix with the watermark according to Equation

(3) to get the matrix R^* .

(4) Extracting the watermark

Δ , min, max, are computed again using r_{1j}^* ($j=1,2,\dots,m$).

Extract the watermark $w_{H_i}^*$ from the element r_{1j}^* ($j=1,2,\dots,m$) in the first row of the matrix R^* by using Equation (27). $[\]$ indicates to round a number.

$$w_{H_i}^* = \begin{cases} 0 & \text{if } \lceil (r_{1j}^* - \min) / \Delta \rceil \text{ is even} \\ 1 & \text{if } \lceil (r_{1j}^* - \min) / \Delta \rceil \text{ is odd} \end{cases} \quad (27)$$

(5) Repeat

Steps (2)-(4) are repeatedly executed until all matrix blocks with embedded watermarks are treated. The extracted watermark sequence $w_{H_i}^*$ is grouped with 8 bits as one group. The binary data of each group are transformed to their decimal data values. The final data is stored as the gray image $H^*(x, y)$.

(6) Optical holographic decryption

A Fourier transformation for the gray images is performed according to Equation (10) and filtered with a 2nd order Butterworth high-pass filter to get the resulting watermarking image $g_{mark}^*(x, y)$.

4. EXPERIMENTAL RESULTS AND ANALYSIS

To check the performance of our watermarking algorithm, we realized a three-dimensional model watermarking prototype system by using Matlab2013 as the experimental platform. The three-dimensional models Stanford Bunny, Horse, and Dragon, available at http://www.cc.gatech.edu/projects/large_models/, were used as the home images. The Stanford Bunny model includes 35,947 vertices and 69,451 triangle faces. The Horse model includes 48,485 vertices and 96,966 triangle faces. The Dragon model includes 50,000 vertices and 100,000 triangle faces. A binary image of size 32×32 was selected as the watermarking image. The size of the QR decomposition matrix was 4×4 , namely $s = 32$, $m = 4$. Fig. 3 shows the model, a binary watermarking image called "USST" and the encrypted image (8 bits per pixel).

The watermarking mesh model experienced some processing, transformations, and attacks, so the extracted watermark was not exactly the same as the embedded watermark. To assess the robustness of this watermarking algorithm, we computed the related coefficients between the extraction of the watermark sequence and embedment of the watermark sequence [3], which resulted in Equation (28).

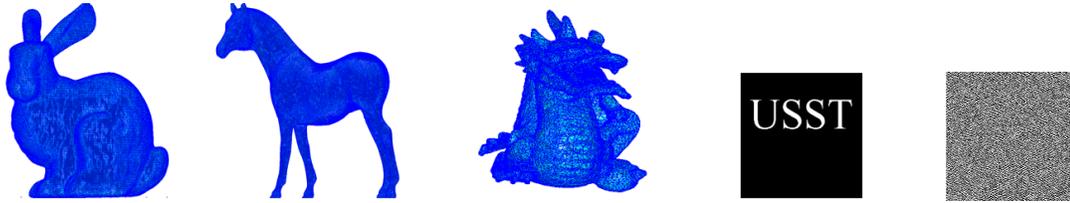


Fig. 3. Original models, watermarking image, and holographic encrypted image.

Table 1. Robustness for the affine attack and reordering attack.

Model	Corr of Rotation Attack	Corr of Translation Attack	Corr of Scaling Attack	Corr of Reordering Attack
Bunny	1	1	1	1
Horse	1	1	1	1
Dragon	1	1	1	1

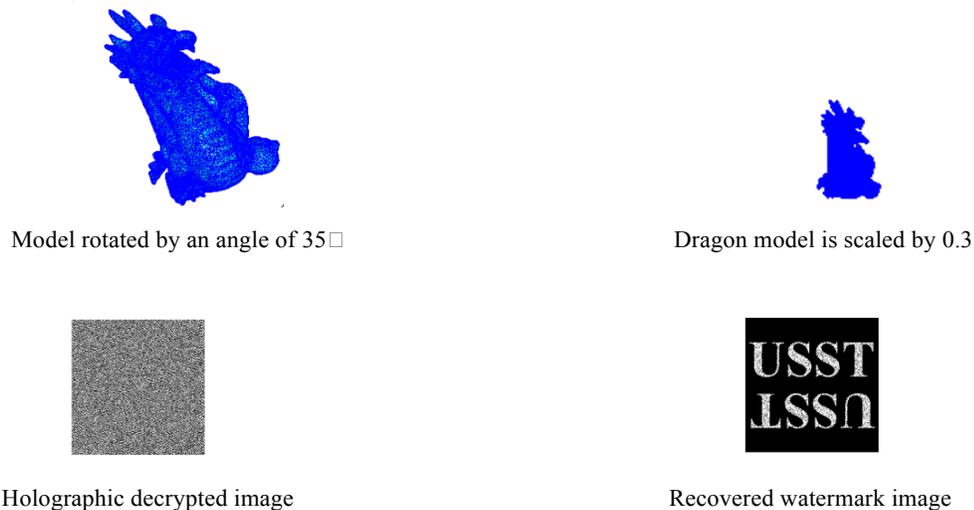


Fig. 4. Affine transformation attack.

$$corr = \frac{\sum_{i=1}^{length} (w_i - \bar{w})(w'_i - \bar{w}')}{\sqrt{\sum_{i=1}^{length} (w_i - \bar{w})^2 \sum_{i=1}^{length} (w'_i - \bar{w}')^2}} \quad (28)$$

w_i is the embedded watermark w_{H_i} , w'_i is the extracted watermark $w_{H_i}^*$ that needs to be determined. \bar{w} is the mean of w_i and \bar{w}' is the mean of w'_i .

$corr \in [0,1]$. A bigger $corr$ indicates a higher similarity of the watermark and stronger robustness of the watermark in this algorithm.

To detect the attack effects on the algorithm, we tested multiple attacks. The results and analysis are as follows.

(1) Vertex reordering attack

Two randomly selected vertices exchanged their sequences and were repeated $10 \times n$ times. n indicates the number of the vertices in the model. The experimental results (shown in Table 1) indicate that the watermarking information can be completely extracted after the reordering attack because the distances from the vertices to the model's center are sorted by the z and θ values in the construction of the model's geometric feature matrix. The same processing is required in the watermark extraction to ensure that the geometric feature matrix of the model is kept constant for the vertex reordering attack. The vertex reordering attack did not affect the watermark extraction.

(2) Affine transformation attack

The affine transformation attacks include rotation, translation, and scaling. The Dragon model with a watermark was rotated by an angle of 35 degrees around the z axis and scaled by 0.3; the effect is shown in Fig. 4. The experimental results of the translation, rotation, and scaling attack are shown in Table 1. The mesh's watermark extrac-

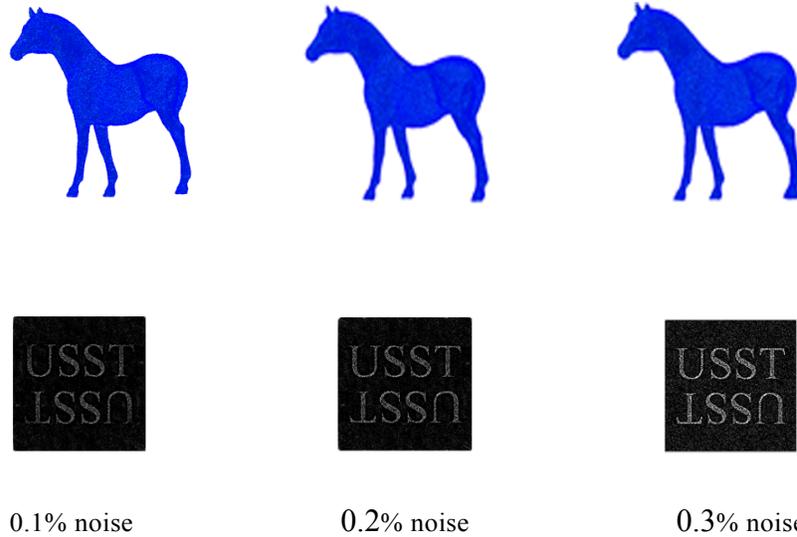


Fig. (5). Noise attack effect.

Table 2. Model robustness for noise attack.

Model	Noise Intensity	Corr
Bunny	0.1%	0.8634
	0.2%	0.8478
	0.3%	0.7356
Horse	0.1%	0.8718
	0.2%	0.8536
	0.3%	0.7489
Dragon	0.1%	0.8836
	0.2%	0.8615
	0.3%	0.7658

tion was performed after the origin of the coordinates moved the model’s center, so the translation attack did not affect the watermark extraction. The PCA method was used for the automated alignment to make the watermarking three-dimensional model maintain the same orientation and posture before and after the attack, so the rotation op

eration did not affect the watermark extraction. After the elements of the geometric feature matrix were normalized, the watermark was embedded and extracted. The scaling attack did not affect watermark extraction.

(3) Noise attacks

A random noise vector was added for each vertex in the three-dimensional model and the lengths of this vector were 0.1%, 0.2%, and 0.3% of the mean distance from the vertices to the model’s center. The experimental effect is shown in Fig. 5. The experimental results are shown in

Table 2. When the noise intensity reached 0.3%, the watermark was still able to be extracted because the watermarking information was embedded into the elements of the first row in the R matrix obtained after the QR decomposition. The elements in this row were the largest in the entire R matrix. Their small changes will not lead to an information extraction failure. This indicates that the QR decomposition was very stable.

(4) Cropping attacks

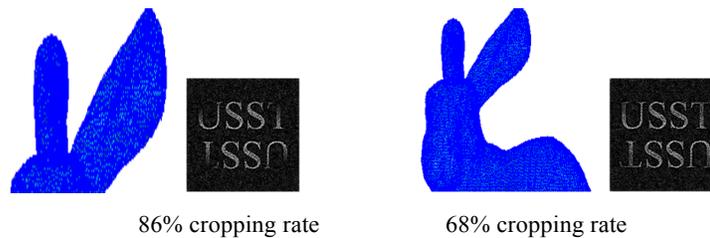
For the cropping attack, the vertex number and topology of the three-dimensional model were changed. The re-sampling method [4] was used to recover the vertex (with 0.1% noise, 0.2% noise, and 0.3% noise).

① Cropping attacks

The effect that occurred after the Stanford Bunny model with the watermarking information was cropped is

Table 3. Model robustness for cropping attack.

Model	Noise Intensity	Corr
Bunny	68%	0.8495
	86%	0.6115
Horse	68%	0.8595
	86%	0.6245
Dragon	68%	0.8652
	86%	0.6354

**Fig. (6).** Cropping attack effect.

shown in Fig. 6. The experimental effect of the cropping attack is shown in Table 3.

The cropping attack results in Fig. 6 show that, even though the cropping rate of the Stanford Bunny model reached 86%, the extracted watermarking image was still clearly seen because the embedded watermarking information was processed with the optical holographic encrypted technology that can resist the cropping well.

CONCLUSION

Spatial domain mesh blind watermarking algorithms have some problems, such as a difficult implementation, poor robustness, and easy decryption, so we proposed a three-dimensional mesh model optical holographic blind digital watermark algorithm based on the QR decomposition. This algorithm pre-processes the three-dimensional model and embeds the watermarking information generated by the optical holographic encrypted technology into the normalized and blocked model's geometric feature matrix with QR decomposition. By performing the inverse QR decomposition and inverse normalization, the watermark is embedded into the three-dimensional mesh model. The experimental results indicate that this algorithm can resist translation, scaling, vertex reordering, noise, and cropping attacks, has a high robustness, and an important application value. This algorithm does require significant computing time, so we will improve this algorithm by using the GPU-based parallel computing method for the QR decomposition of the model's geometric feature matrix to further speed up this algorithm.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported in part by a grant from the Starting Project of Doctor of University of Shanghai for Science and Technology (No. 1D-13-309-005), Funding Scheme for Training Young teachers in Shanghai Colleges in 2013 (No. slg14039), Innovation Program of Shanghai Municipal Education Commission (No. 13ZZ111), and the Bidding Project of the Shanghai Research Institute of Publishing and Media (No. SAYB1409) funded by the National Higher Vocational and Technical Colleges construction project of the Shanghai Publishing and Printing College.

REFERENCES

- [1] B.T. Andrade, C.M. Mendes, J.O. Santos Jr., O.R.P. Bellon, and L. Silva, "Three-Dimensional Preserving xviii Century Baroque Masterpiece: Challenges and Results on the Digital Preservation of Aleijadinho's Sculpture of the Prophet Joel", *J. Cultural Heritage*, vol. 13, pp. 210-214, Jun. 2012.
- [2] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "A Comprehensive Survey on Three-Dimensional Mesh Watermarking", *IEEE Trans. Multimedia*, vol. 10, pp. 1513-1527, Dec. 2008.
- [3] E. Praun, H. Hoppe, and A. Finkelstein, "Robust Mesh Watermarking", In: *Computer Graphics Proceedings, Annual Conference Series*, ACM SIGGRAPH, 1999, pp. 49-56.
- [4] R. Ohbuchi, S. Takahashi, T. Miyazawa and A. Mukaiyama, "Watermarking Three-Dimensional Polygonal Meshes in the Mesh Spectral Domain", In: *Proceedings of Graphics Inter-face*, pp. 9-17, 2001.
- [5] K. Wang, G. Lavoué, F. Denis, and A. Baskurt "Hierarchical Watermarking of Semiregular Meshes Based on Wavelet Trans-

- form”, *IEEE Trans. Inf. Forensics Security*, vol. 3, pp. 620-634, Dec. 2008.
- [6] Y.P. Liu, “A robust spectral approach for blind watermarking of manifold surfaces”, In: *Proceedings of the 10th ACM Workshop on Multimedia and Security*, pp. 43-52, 2008.
- [7] J.M. Konstantinides, A. Mademlis, P. Daras, P.A. Mitkas, and M.G. Strintzis, “Blind ROBUST 3-D mesh watermarking based on oblate spheroidal harmonics”, *IEEE Trans. Multimedia*, vol. 11, pp. 23-38, Jan. 2009.
- [8] R. Ohbuchi, H. Masuda, and N. Aono, “Watermarking Three-Dimensional Polygonal Models”, In: *Proceedings of ACM Multimedia 97*, 1997, pp. 261-272.
- [9] R. Ohbuchi, H. Masuda, and N. Aono, “Embedding Data in Three-Dimensional Models”, In: *Proceedings of the 1997 4th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services*, pp. 1-10, 1997.
- [10] R. Ohbuchi, H. Masuda, and N. Aono, “Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modification”, *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 551-560, May 1998.
- [11] R. Ohbuchi, H. Masuda, and N. Aono, “Watermarking Multiple Object Types in Three-Dimensional Models”, In: *Proceedings of the Workshop on Multimedia & Security at ACM Multimedia ’98*, pp. 83-91, 1998.
- [12] S. Zafeiriou, A. Tefas, and I. Pitas, “Blind Robust Watermarking Schemes for Copyright Protection of Three-Dimensional Mesh Objects”, *IEEE Trans. Vis. Comput. Graph.*, vol. 11, pp. 596-607, Sep. 2005.
- [13] M. Salman, Z. Ahmad, S. Worrall, and A.M. Kondoz, “Robust Watermarking of 3-D Polygonal Models”, In: *3rd International Symposium on Communications, Control and Signal Processing*, pp. 340-343, 2008.
- [14] Y. Wang, and S. Hu, “A New Watermarking Method for Three-Dimensional Models Based on Integral Invariants”, *IEEE Trans. Vis. Comput. Graph.*, vol. 15, pp. 285-294, Mar. 2009.
- [15] Q.S. Ai, Q. Liu, Z.D. Zhou, L. Yang, and S. Q. Xie, “A new digital watermark scheme for three-dimensional triangular mesh models”, *Signal Proces.*, vol. 89, pp. 2159-2170, Nov. 2009.
- [16] H. Lee, C. Dikici, G. Lavoue, and F. Dupont, “Joint reversible watermarking and progressive compression of three-dimensional meshes”, *Visual Comput.*, vol. 27, pp. 6-8, June 2011.
- [17] Q. Liu and X. Zhang, “SVD Based Digital Watermark Algorithm for Three-Dimensional Models”, In: *International Conference on Signal Processing Proceedings*, pp. 1-6, 2007.
- [18] Q. Su, Y. Niu, G. Wang, S. Jia, and J. Yue, “Color image blind watermarking scheme based on qr decomposition”, *Signal Proces.*, vol. 94, pp. 219-235, January 2014.
- [19] N. Takai, and Y. Mifune, “Digital watermark by a holographic technique”, *Appl. Opt.*, vol. 41, pp. 865-873, Feb 2002.
- [20] A. Kalivas, A. Tefas, and I. Pitas, *Watermarking of Three-Dimensional Models Using Principal Component Analysis*, pp. 676-679, 2003.

Received: September 22, 2014

Revised: November 03, 2014

Accepted: November 06, 2014

© Wang *et al.*; Licensee Bentham Open.

This is an open access articles licensed under the terms of the Creative Commons Attribution-Non-Commercial 4.0 International Public License (CC BY-NC 4.0) (<https://creativecommons.org/licenses/by-nc/4.0/legalcode>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided that the work is properly cited.