A Data Hiding Scheme based on Local Coordinate System for 3D Triangle Mesh Models

ShanchaoYang*
Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, China
Email: snadms@163.com

Zhiqiang Yao
Faculty of Software, Fujian Normal University, Fuzhou, China
Email: yzq@fjnu.edu.cn

***Abstract*—Recently there is increased interest in hiding data in 3-D objects due to the ever-increasing diffusion of such objects in many areas such as architecture design, entertainment and cultural heritage. A data hiding scheme for 3D triangle mesh models is proposed in this paper. Constructed by the vertex and its 1-ring neighbors, rather than by only its 1-ring neighbors, the local coordinate system can represent the local geometry better and visual quality of 3D model can be controlled better when message is embedded. Message is embedded by modifying coordinate of the central vertex without changing the coordinate systems. Being built on special local features of the 3D model, the local coordinate systems are invariable under RST (rotation, uniform scaling, translation) transformation. Therefore, the algorithm is resilient to RST attack. Combining the local coordinate systems with a special quantization technique, the high capacity and imperceptibility are achieved. These properties are also validated by experimental results in the paper.**

***Index Terms*—Data Hiding, 3D Model, Local Coordinate System**

I. INTRODUCTION

As a promising media data, 3D models become increasingly important in daily life and industrial applications. There is much redundancy in the media. Message can be embedded by modifying the 3D model imperceptibly, which leads to the emergence of data hiding in 3D models. Data hiding is the science to hide messages in multimedia so that the message is undetected to all but the owner [1]. It can be applied in many fields, such as secret communication and copyright protection. This paper is designed mainly for secret communication.

Many 3D mesh watermarking algorithms have been proposed, Ohbuchi[2] first published the paper of 3D watermarking, the TSQ and TVR algorithms [3] are not robust against common attacks.

The robust watermarking algorithms are usually designed in frequency domain or based on statistic features. The obvious disadvantage is that capacity is low. Besides, the frequency domain strategy often needs original model to extract watermark. Based on a multiresolution set of scalar basis functions over the mesh, Praun[4] proposed a robust algorithm. Kanai[5] presented a 3D watermarking based on wavelet transform for semi-regular meshes, but the algorithm is not fit for a universal topological meshes. Yin presented a robust mesh watermarking algorithm [6] based on the multiresolution decomposition of 3D mesh proposed by Guskov et al [7]. These strategies need original model to extract watermark. Based on face normal modification, Benedens[8,9] proposed a watermarking scheme which is robust against mesh simplification, Lee[10] improved the scheme by use of EGI and CEGI, The scheme is more robust against mesh cropping attack. Based on statistic features, the two schemes show excellent robustness, but they are very complicated and too many parameters are needed.

Comparatively, spatial domain algorithms usually can embed more bits, but the robustness may be less strong, so many of them are designed as data hiding algorithm. Some schemes are designed based on local geometry to resist RST attack. Wagner [11] introduced the conception of discrete normal and used it in designing watermarking algorithms. Aspert [12] used the method for embedding meaningful information. Maret [13] improved the algorithm by constructing similarity space, but a large linear equational group is needed to solve. Sun proposed a watermarking algorithm [14] against affine transform based on Nielson norm, Sun also proposed an algorithm against RST attack based on local coordinate system [15]. Bors and Harte [16-17] presented a scheme by local geometry modification according to bounding ellipsoids. In these schemes, imperceptibility performances are not fully considered. For example, in [13] quality of the embedded model cannot be estimated, the data hiding algorithm cannot provide a good imperceptibility. Although Nielson norm [14] based algorithm can resist general affine transform, but modifications of 3D model under affine system may not provide a good quality of the 3D model under original orthogonal coordinate system, so is the paper [16,17]. Besides, the above local coordinate system is based only on the neighbors of a vertex, not on the vertex and its neighbors, so the

*Corresponding author. Tel.: +86 13615010978.

doi:10.4304/jsw.5.4.437-446

appearance of the local geometry is not fully considered. Therefore, in the paper, a 3D data hiding algorithm based on a new local coordinate system is designed so that appearance of the model can be controlled better by parameters and so that the algorithm is capacity high and resilient to RST attack. We have proposed primary implementations in [18]. As an extended version, this paper is to describe the scheme in detail, including necessary deductions, detailed results and deep analysis.

The article is organized as follows: Section II gives basic definition and the scheme; Section III and IV describe message embedding process and message extracting process respectively; Section V shows experimental results and analysis; and Section VI concludes the paper.

II. BASIC DEFINITIONS AND THE SCHEME

A. Basic Definations

A triangle mesh model can be defined as a vertex set V, a face set F and an edge set E as shown in (1), (2) and (3).

$$V = \{V_i = \{x_i, y_i, z_i\} \mid 0 \le i \le n_v\}. \quad (1)$$

$$F = \{F_i = \{i_i, j_i, k_i\} \mid 0 \le i \le n_f\}. \quad (2)$$

$$E = \{E_i = \{i_i, j_i\} \mid 0 \le i \le n_e\}. \quad (3)$$

Vertex i and its 1-ring-neighbors form a set, which is considered as an EP (Embedding Primitive). Each vertex corresponds with an EP, but not all EPs are selected to embed information. 1-ring neighbors of vertex i is defined as a 1-ring vertex set V_{nb}^i, a 1-ring face set F_{nb}^i and a 1-ring edge set E_{nb}^i as shown in (4), (5) and (6).

$$V_{nb}^i = \{V_j \mid E_k = \{i,j\} \text{ is an edge of } E, j = 1,2,..., n_v^i\}. \quad (4)$$

$$F_{nb}^i = \{F_j \mid i = i_j \text{ or } i = j_j \text{ or } i = k_j, j = 1,2,..., n_f^i\}. \quad (5)$$

$$E_{nb}^i = \{E_j \mid \{i, i_j, j_j\} \text{is a } 1-\text{ring face of } V_i, j = 1,2,...,n_e^i\}. \quad (6)$$

B. Scheme of the Algorithm

The algorithm consists of two processes: message embedding process and message extracting process, the corresponding flow charts are shown in Fig. 1 (a) and (b). Let i be the index of the center vertex, let j be the counter of used EPs, then message embedding process (is described in section III.) consists of 4 steps:

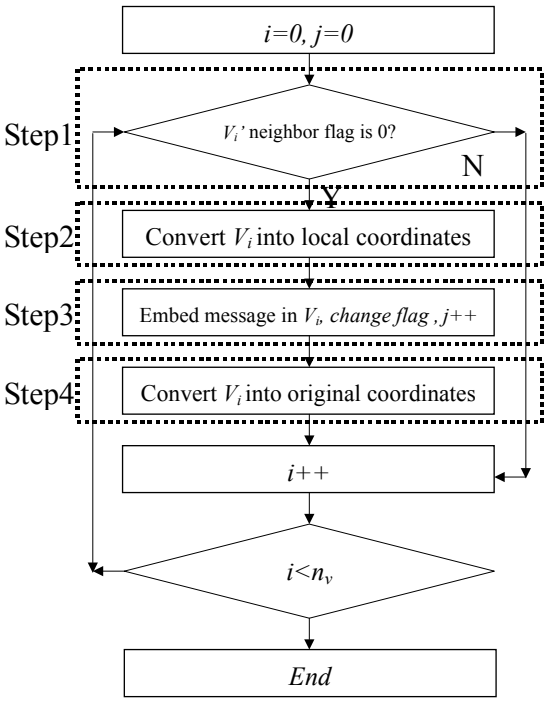
Step1 selecting one embedding primitive EP i, which is described in Section III.A.

Step2 converting vertex i of EP i into its local coordinate system, which is described in Section III.B.

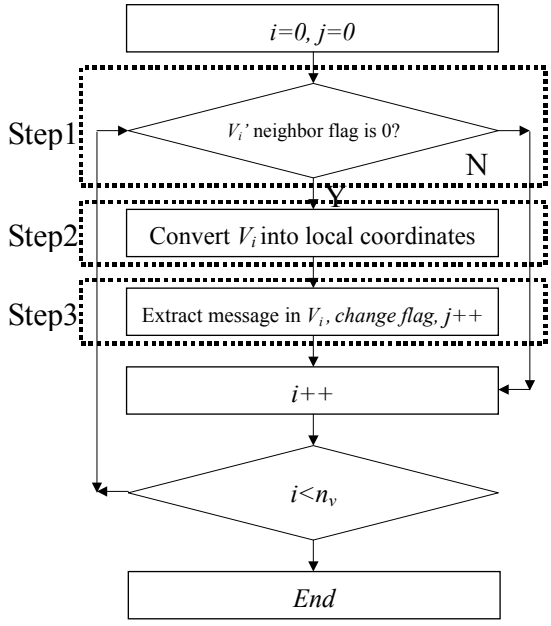
Step3 embedding message into EP i by modifying vertex i's local coordinates, changing flags of its neighbors and changing j 's value. Here EP i is j th used EP. This step is described in Section III.C.

Step4 converting vertex i into original coordinate system, which is described in Section III.D.

Compared with the embedding process, the extracting process needs not converting vertices into their original coordinate system. It has 3 Steps, most of which is very similar with the embedding process except for Step3. The Step3 of the extracting process is the inverse process of the Step3 of message embedding process. Message extracting process is described in Section Ⅳ.



(a) Message embedding process



(b) Message extracting process

Figure 1. Flow chart of the scheme

III. EMBEDDING PROCESS

We will give detailed description on each step of message embedding process as shown in Fig. 1 (a).

A. Selection of EP

Embedding order of EPs is important for synchronization of message extracting process. Our strategy is similar with [16,17], which has been described detailedly in Fig. 1. Each vertex has a flag indicating whether the vertex is a neighbor. If the flag of a vertex is TRUE, it means the vertex has been used as a neighbor. If a vertex has been used as a neighbor, it cannot be used as a central vertex again. If the flag of a vertex is FALSE, that means the vertex has not been checked or it has been used as a central vertex.

Concretely, each flag of vertex in the vertex list is initialized to be FALSE, indicating that each vertex has not been used as a neighbor vertex and that each vertex has the chance to be a central vertex. Each vertex is checked in sequence to determine whether the vertex can be used as a central vertex or neighbor, if the flag of the vertex is TRUE, the vertex cannot be used as a central vertex; else the vertex can be used as a central vertex and one EP is found. The EP is used to construct a local coordinate system, and then message is embedded by modifying local coordinate of the central vertex. When message embedding process of one EP is finished, flag of the neighbor vertices must change to be true.

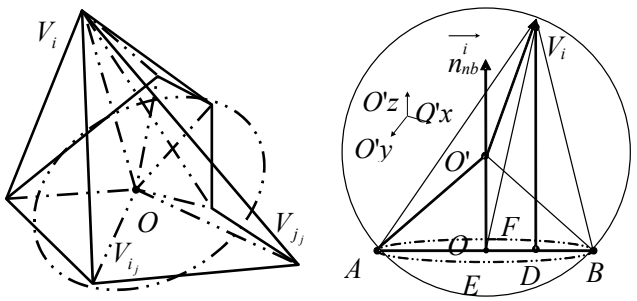
B. Mapping EP into Local Coordinate System

To embed message, the center vertex of EP i is firstly transformed into its local coordinate system, so in this sub-section, local coordinate system is first defined, then the mutual conversion of the local coordinate system and original coordinate system is described, finally the features of the local coordinate system is discussed.

1) Definition of the local coordinate system

EP i is shown in Fig. 2 (a), for a given vertex V_i, Centroid O of vertices of its 1-ring vertices is first calculated according to (7).

$$O = \sum_{1}^{n_v^i} V_i \,. \tag{7}$$



(a) Circle of 1-ring neighbors (b) Circular cone and circumsphere

Figure 2. The local coordinate system

Each member $\{i_j, j_j\}$ of 1-ring edges and O can form a triangle $\Delta OV_{i_j}V_{j_j}$ corresponding with V_i' s neighbor triangle $\Delta V_iV_{i_j}V_{j_j}$. A plane Π_{ni} is defined by giving its normal as in (8):

$$\overrightarrow{n_{nb}^i} = \sum_{1}^{n_e^i} A_j^i \overrightarrow{n_j^i} \,. \tag{8}$$

Where A_j^i, $\overrightarrow{n_j^i}$ is respectively area, normal of triangle $\Delta OV_{i_j}V_{j_j}$, n_e^i is the corresponding number of triangles. Then a circle Co in the plane with center O, area A, $A = \left\| \overrightarrow{n_{nb}^i} \right\|$ is formed. The radius r can be calculated by $r = \sqrt{A/\pi}$. Since O is derived from V_i' s neighbor vertices and r is related only with O and the neighbor vertices, it easy to see the circle Co is independent with vertex V_i, but it depends on V_i' neighbor vertices.

As shown in Fig. 2 (b), the circle and vertex V_i can form a circular cone, and a circumsphere $S_{o'}$ of the circular cone with the center O' can be defined. Let us calculate the center and radius of the circumsphere, First V_i is projected on to the plane Π_{ni} at point D, a line through D and O intersects with circle Co at A and B, then circumcircle $C_{o'}$ of triangle ΔV_iAB must be a big circle of the circumsphere $S_{o'}$, the radius and center of $S_{o'}$ can be calculated. The calculation of circumsphere's center and radius is described as follows.

$$\overrightarrow{OD} = \overrightarrow{OV_i} - (Normalize(\overrightarrow{n_{nb}^i}) \cdot \overrightarrow{OV_i}) \cdot Normalize(\overrightarrow{n_{nb}^i}) \,, \tag{9}$$

$$\overrightarrow{OB} = r \cdot Normalzie(\overrightarrow{OD}) \,, \tag{10}$$

$$B = O + \overrightarrow{OB} \,, \tag{11}$$

$$A = O + \overrightarrow{OA} = O - \overrightarrow{OB} \,, \tag{12}$$

$$\overrightarrow{V_iA} = A - V_i \,, \tag{13}$$

$$\overrightarrow{V_iB} = B - V_i \,, \tag{14}$$

$$\angle AV_iB = \arccos(normalize(\overrightarrow{V_iA}) \cdot normalize(\overrightarrow{V_iB})) \,, \tag{15}$$

$$R = \overrightarrow{O'B} = \overrightarrow{OB} / \sin\angle OO'B = r / \sin\angle AV_iB \,. \tag{16}$$

There are generally six possible cases to describe the relation of center vertex V_i and the circle Co, as shown in Fig. 3.

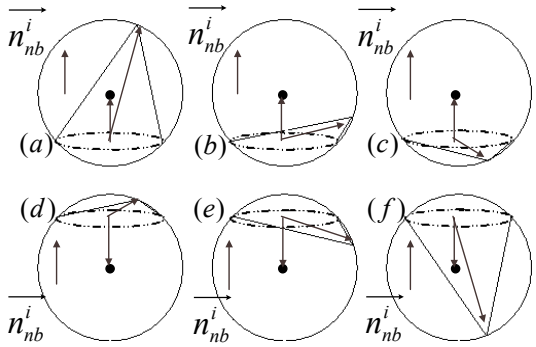


Figure 3. Six cases of the relation of center vertex and the little circle.

The six cases are:

(a) $\overrightarrow{OO'} = R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i})$; (17)

(b) $\overrightarrow{OO'} = R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i})$; (18)

(c) $\overrightarrow{OO'} = R \cdot (-\cos\angle AV_iB) \cdot Normalize(\overrightarrow{n_{nb}^i})$

$$= -R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i}) ; \quad (19)$$

(d) $\overrightarrow{OO'} = R \cdot (-\cos\angle AV_iB) \cdot (-Normalize(\overrightarrow{n_{nb}^i}))$

$$= R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i}) ; \quad (20)$$

(e) $\overrightarrow{OO'} = R \cdot \cos\angle AV_iB \cdot (-Normalize(\overrightarrow{n_{nb}^i}))$

$$= -R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i}) ; \quad (21)$$

(f) $\overrightarrow{OO'} = R \cdot \cos\angle AV_iB \cdot (-Normalize(\overrightarrow{n_{nb}^i}))$

$$= -R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i}) . \quad (22)$$

So the six cases can be merged into two cases to find $\overrightarrow{OO'}$:

if($\overrightarrow{OV_i} \cdot \overrightarrow{n_{nb}^i} \geq 0$)

$$\overrightarrow{OO'} = R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i}) ; \quad (23)$$

if($\overrightarrow{OV_i} \cdot \overrightarrow{n_{nb}^i} < 0$)

$$\overrightarrow{OO'} = -R \cdot \cos\angle AV_iB \cdot Normalize(\overrightarrow{n_{nb}^i}) . \quad (24)$$

When $\overrightarrow{OO'}$ is obtained, combining with O, O' can be easily obtained.

$$O' = O + \overrightarrow{OO'} \quad (25)$$

Based on the above data the local coordinate system can be constructed. Coordinate system axes should be defined first. The local orthogonal coordinate system $O'xyz$ is defined by (26), (27) and (28), where O' is origin , $\overrightarrow{O'x}$ $\overrightarrow{O'y}$ and $\overrightarrow{O'z}$ are respectively x, y, z coordinate axes.

$$\overrightarrow{O'z} = Normalize(\overrightarrow{n_{nb}^i}) \quad (26)$$

$$\overrightarrow{O'x} = \{Normalize(\overrightarrow{OV_j} - (\overrightarrow{OV_j} \cdot Normalize(\overrightarrow{n_{nb}^i})) \cdot Normalize(\overrightarrow{n_{nb}^i}))$$

$$| V_j \in V_{nb}^i, and\ j = arg\ max\{\overrightarrow{OV_j}\}\} \quad (27)$$

$$\overrightarrow{O'y} = \overrightarrow{O'x} \times \overrightarrow{O'z} \quad (28)$$

2) mutual conversions of original coordinate system and the local coordinate system

When the local coordinate system of EP i is set up, coordinates of central vertex V_i is determined. Before message is embedded in coordinates of V_i, it should be first transformed into local orthogonal coordinate system then into local spherical coordinate system. The conversions are shown in Fig. 4.

Converting V_i from the original coordinate system to the local spherical coordinate system is needed for message embedding process. When $\overrightarrow{O'x}$, $\overrightarrow{O'y}$, $\overrightarrow{O'z}$ and O' are obtained, $\overrightarrow{O'V_i}$ can be gotten, the local orthogonal coordinates of V_i can be computed according to (29), (30) and (31).

$$x = \overrightarrow{O'x} \cdot \overrightarrow{O'V_i} \quad (29)$$

$$y = \overrightarrow{O'y} \cdot \overrightarrow{O'V_i} \quad (30)$$

$$z = \overrightarrow{O'z} \cdot \overrightarrow{O'V_i} \quad (31)$$

When the local orthogonal coordinate of V_i is gotten, spherical coordinates can be easily calculated by (32) and (33). Here the local orthogonal coordinates can be easy mapped into a unit sphere with a longitude φ and a latitude θ. The radius is omitted because we must keep the local coordinate system unchanged.

$$\theta = \arccos(z / \sqrt{x^2 + y^2}) \quad (32)$$

$$\varphi = \begin{cases} \arccos(x/\sqrt{x^2 + y^2}) & y \geq 0 \\ 2\pi - \arccos(x/\sqrt{x^2 + y^2}) & y < 0 \end{cases} \quad (33)$$

To refresh the coordinates of the central vertex V_i, the local spherical coordinates are needed to convert into the original coordinate system. The mapping process is the inverse process of mapping from the local orthogonal coordinate system to the local spherical coordinate system.

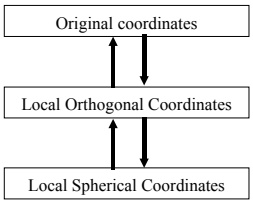


Figure 4. Conversions of coordinate systems

$$z' = R\cos\theta' \tag{34}$$

$$x' = R\sin\theta'\cos\varphi' \tag{35}$$

$$y' = R\sin\theta'\sin\varphi' \tag{36}$$

The vector decomposition technique is used to convert coordinates from the local orthogonal coordinate system to the original coordinate system. The conversion is realized by (37).

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} x' & y' & z' \end{bmatrix} \cdot \begin{bmatrix} \overrightarrow{O'x} \\ \overrightarrow{O'y} \\ \overrightarrow{O'z} \end{bmatrix} \tag{37}$$

3) Features of the local coordinate system

Let us analyze the curvature feature at center vertex of one EP. In Fig. 5, the plane V_iAB that passes through V_i and O is vertical with circle O, AB is diameter. $\angle AV_iB$ can be a metric of the maximum principal curvature (we call it PC1) of tip of the cone. The vertical plane of ΔAV_iB intersects with the circumsphere at circle I, which intersect with circle O at E, F and intersect with circle $C_{o'AV_iB}$ at H, V_iH must be angle bisector of $\angle AV_iB$, then H must be middle point of the arc AHB and H, O and O' are on the same line. Then $\angle EV_iF$ can be a metric of another minimum principal curvature (we call it PC2) of the tip of the cone. The value of $\angle AV_iB$ and $\angle EV_iF$ can be calculated.

$\angle AV_iB$ can be calculated according to (15). As we can see, in the circumcircle O', if the length of line segment AB is fixed, the length of arc AHB is fixed, then $\angle AV_iB$ is invariant. So the value of PC1 can be measured by $\angle AV_iB$, which is invariant with θ and φ coordinate of V_i.

To compute $\angle EV_iF$, we can find length of V_iG and V_iH, V_iH can be computed directly.

$$H = O' - R \cdot O'z \tag{38}$$

$$V_iH = H - V_i \tag{39}$$

Then V_iH is the diameter of circle I. Let the radius of circle I be r_2, then $r_2 = V_iH/2$.

V_iG is the angle bisector of $\angle AV_iB$, according to law of sines, we get

$$\frac{V_iG}{V_iB} = \frac{\sin\angle V_iBA}{\sin(\pi - \angle V_iBA - \angle AV_iB/2)}, \tag{40}$$

then

$$V_iG = \frac{V_iB \cdot \sin\angle V_iBA}{\sin(\pi - \angle V_iBA - \angle AV_iB/2)}. \tag{41}$$

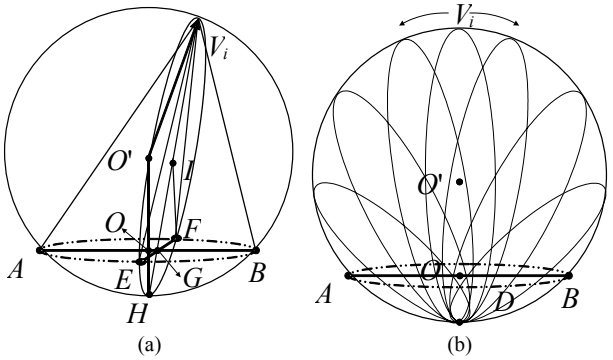


Figure 5. Curvature feature analysis

In ΔIGF,

$$IF = r_2 = V_iH/2, \tag{42}$$

$$IG = V_iG - IV_i = V_iG - r_2 = V_iG - V_iH/2. \tag{43}$$

So

$$\angle EV_iF = \angle FIG = \arccos(IG/IF) \tag{44}$$

or

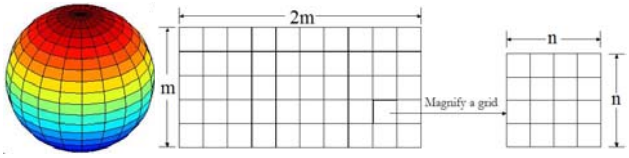
$$\angle EV_iF = \arccos\left(\frac{V_iH/2}{(V_iG - V_iH/2)}\right). \tag{45}$$

It is easy to see when the latitude of V_i varies from north polar to the latitude of circle O, the value of $\angle EV_iF$ becomes larger, so PC2 becomes smaller as shown in Fig. 5 (b). Concretely, in (44), $\angle EV_iF = \arccos(IG/IF) = \arccos(IG/IV_i)$, with the latitude of V_i varying from north polar to the latitude of circle O, $IG\downarrow$, $IV_i\downarrow$, but the varying speed of IV_i is lower than IG, so $(IG/IV_i)\downarrow$, then $\angle EV_iF\uparrow$. So the value of PC2 can be measured by $\angle EV_iF$, which is related with the value of θ coordinate of V_i.

But in another aspect, the direction of PC1 and PC2 is related with θ and φ coordinate of V_i. So the two principal curvature, mean curvature, and Gaussian curvature can be controlled by θ and φ. In the proposed data hiding algorithm, message is embedded by modifying the θ and φ coordinates of center vertex V_i of EP i, the modification is related with the important geometrical features of the vertex, so the modification can be controllable and the modification is good for 3D model visual quality.

C. Data Embedding in EPs under Local Coordinate System

The range of θ and φ is respectively [0,2π] and [0, π]. Here a special quantization technique is used. As Fig. 6 shows, longitude and latitude is divided respectively into m and $2m$ section and each section have a length $\omega = \pi/m$. Then totally $2m^2$ grids are formed. The



(a) (b)

Figure 6. Division of the spherical coordinate system

parameter m can be used to decide tradeoff of imperceptibility and embedding capacity for m decide the varying range of the vertex's coordinates, this point will be testified by experimental results. Each grid is re-divided into about n^2 sub-grids which represents embedding states, by dividing n sections in longitude direction and n Sections in latitude direction: $n = \lfloor \omega/\delta \rfloor$, δ is a parameter to decide tradeoff of robustness and capacity, which also will be testified by experimental results. Let $n' = \lfloor \log_2 n \rfloor$, totally $2n'$ bits are embedded in an EP, each direction of latitude and longitude embedding n' bits. Then embedding capacity of 3D model can be computed. Let BPE be embedding bits per EP, then

$$BPE = 2n' = 2 \cdot \lfloor \log_2 n \rfloor = 2 \cdot \lfloor \log_2 \lfloor \omega/\delta \rfloor \rfloor = 2 \cdot \lfloor \log_2 \lfloor \pi/m/\delta \rfloor \rfloor . \quad (46)$$

Let number of the used EPs be EPN, then the embedding capacity of the 3D model should be

$$Capacity = EPN \cdot BPE = 2 \cdot EPN \cdot \lfloor \log_2 \lfloor \pi/m/\delta \rfloor \rfloor . \quad (47)$$

Because $BPE = 2n'$, that is $2n'$ bits of information can be embedded in one EP. Let j th EP that is used to embed information in EP i. The $2n'$-bit information can be represented as $W^j_{2n'} = W^j_0 W^j_1 W^j_2 \cdots W^j_{2n'-1}$. The high $2n'$ bits of information $high = W^j_0 W^j_1 W^j_2 \cdots W^j_{n'-1}$, and the low $2n'$ bits of information $low = W^j_n W^j_{n+1} W^j_{n+2} \cdots W^j_{2n'-1}$, where $high$ and low are respectively values of high and low n' bits of j th $2n'$ bits of information to be embedded in EP i. Information is embedded by using (48) and (49).

$$\theta = \lfloor \theta/\omega \rfloor \cdot \omega + (high + 0.5) \cdot \omega/n \quad (48)$$

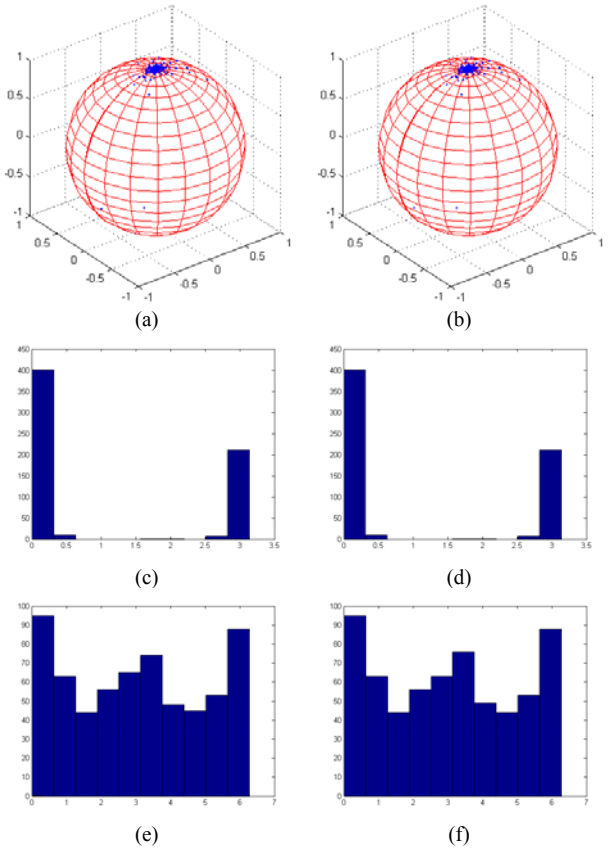
$$\varphi = \lfloor \varphi/\omega \rfloor \cdot \omega + (low + 0.5) \cdot \omega/n \quad (49)$$

For the purpose of synchronization, when information is embedded in EP i, flags of V_i's neighbor must be changed.

Considering six cases shown in Fig. 3, under certain circumstances message cannot be embedded into the n^2 sub-grids of the local spherical coordinate system. One is the that when the little circle Co is located at a too low or too high latitude, the other is that when R is too larger than r. If message is not embedded in the two cases, capacity performance is limited, we can see the capacity losses by comparing this paper with [18]. Experimental results also show vertex distribution in the local coordinate system. Fig. 7 (a) shows the vertex distribution in the local coordinate system before message embedding process. The figure indicates vertices locate mostly at very low latitude (near 0 radian) or very high latitude (near π radian). Fig. 7 (c) and (e) respectively shows histogram of the vertex latitudes and histogram of the vertex longitudes in the local coordinate system, the figures also reflect the above feature.

In the two cases, message cannot be directly embedded, because the varying range of latitude of central vertex is very small. So to eliminate the two limits, we are intended to amplify latitude range that the EP covers meanwhile let the little circle Co unchanged but reduce radius of sphere $S_{o'}$. The method to change the latitude range is: (1) move the position of the little circle Co, that is change the latitude of the little circle Co on the cirumsphere; (2) keep the latitude and longitude coordinate of the central vertex unchanged; (3) Combining the little circle Co and the central vertex V_i, radius R and center O' of cirumsphere $S_{o'}$ are re-calculated. When the two circumstances are eliminated, the regular data embedding process is carried out.

The vertex distribution in the local coordinate system after message embedding process is shown in Fig. 7 (b). The histograms of latitude coordinates and longitude coordinates are shown in Fig. 7 (d) and (f). The figures before message embedding process are similar with those after message embedding process and all selected EP are embedded. This indicates that the adjusting process in special condition is effective.



(a) (b) (c) (d) (e) (f)

Figure 7. Vertex distribution

D. Mapping EP back from Local Coordinate System

The EP must be transformed back into the original coordinate system. Firstly, θ, φ coordinate of local spherical coordinate system is transformed into x, y, z coordinate of the local orthogonal coordinate system; Secondly x, y, z coordinate of the local orthogonal coordinate system is transformed into x, y, z coordinate of the original orthogonal coordinate system, which can be a reverse process of Section III.B, the detailed conversions have been described in subsection III.B.2), so it is omitted in this subsection for the length of the paper.

IV. EXTRACTING PROCESS

Message extracting process is very similar with the message embedding process, the major difference is step 3 in Fig. 1 (b). After the EP is transformed into the local coordinate system, the coordinate θ and φ are calculated. High n' bits and low n' bits are obtained by (50) and (51).

$$high' = \lfloor n \cdot (\theta - \lfloor \theta / \omega \rfloor \cdot \omega) / \omega \rfloor \tag{50}$$

$$low' = \lfloor n \cdot (\varphi - \lfloor \varphi / \omega \rfloor \cdot \omega) / \omega \rfloor \tag{51}$$

$high'$ and low' can be represented by (52) and (53).

$$high' = W_0^i 'W_1^i 'W_2^i ' \cdots W_{n'-1}^i ' \tag{52}$$

$$low' = W_n^i 'W_{n+1}^i 'W_{n+2}^i ' \cdots W_{2n'-1}^i ' \tag{53}$$

Then n' bits of binary information $W_{2n'}^i$' is extracted by (54).

$$W_{2n'}^i ' = W_0^i 'W_1^i 'W_2^i ' \cdots W_{2n'-1}^i ' \tag{54}$$

V. EXPERIMENTAL RESULTS AND ANALYSIS

The experiment is carried out in Visual C++ 6.0, on the platform of Intel (R) Pentium (R) Dual CPU T2310 @ 1.46GHz, 512M RAM, The statistical result of the experiment are plotted in Matlab 7.0. Plain text is used for secret message. In the embedding process, character streams of the text are transformed into bit streams, and the bit streams are embedded in 3D model. In the extracting process, bit streams are extracted from the 3D model, and then it is transformed back into character streams. The tested models are listed in Table I. Several tests are carried out to analyze the imperceptibility, capacity and robustness performance and their relations with the controlling parameter m and δ.

A. Imperceptibility Performance

To test imperceptibility performance of the proposed algorithm, original models and embedded models are rendered. Their results are shown in Fig. 8 and Fig. 9 respectively. It can be found that the embedded models are the same as the original models in appearance; this indicates that the proposed algorithm has a good imperceptibility (The embedded models is obtained when $m = 1000$ and $\delta = 0.0001$).

TABLE I. THE TESTED MODELS IN THE EXPERIMENT.

File Name	Vertex number	Face number
Bunny	10002	20000
cat	4539	8976
Dino	10002	20000
Feline	19998	40000
Fish	3164	6280
Rabbit	10002	20000
Rorder	2904	5804
Torus	6480	12878
Venushead	15002	30000

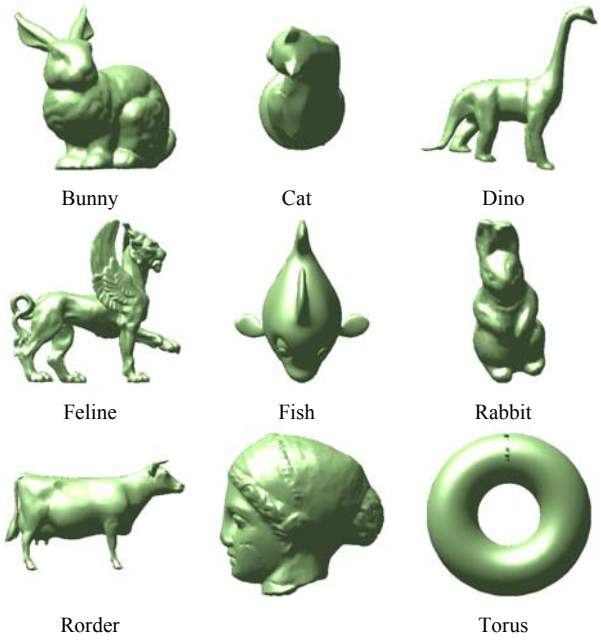


Figure 8. Original models

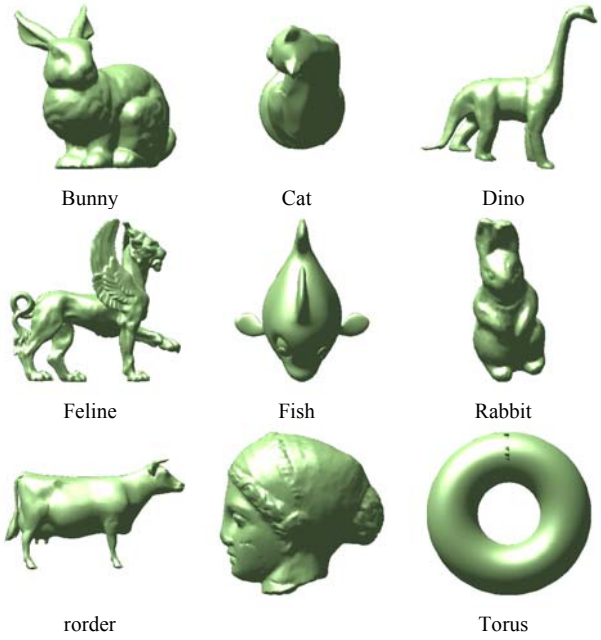


Figure 9. Embedded models

We use the SNR to evaluate imperceptibility of the data hiding algorithm. The SNR is defined as $10log_{10}(var||v\text{-}M||/var||v\text{-}v'||)$, $var(x)$ is the variance of random variable x, M is the centroid coordinates of the original model. v and v' are respectively vertex coordinates of the original model and the embedded model. Fig. 10 shows relationship between m and SNRs when $\delta = 0.00003$. In the figure, the SNR is mostly higher than 50dB. So, the scheme has a good imperceptibility. Besides, the curves generally ascend with the increment of parameter m , which is consistent with the description of subsection III.C, because m is used to control the varying ranges of vertex coordinates, smaller m leads to large varying ranges of vertex coordinates and larger changes of vertex coordinates lead to a lower SNR.

B. Capacity Performance

As has been shown in (47), the capacity is related with EPN , m and δ . According to III.A, the number of EPs used for message embedding process is fixed. Table II shows the used EP number of each 3D model. If δ is fixed ($\delta = 0.000035$), the relation of BPE and m can be shown by Fig. 11. The relation of capacity and m can be shown by Fig. 12. If m is fixed ($m = 700$), the relation of BPE and δ can be shown by Fig. 13, the relation of capacity and δ can be shown by Fig. 14.

But m is related with δ , message cannot be embedded for any m and δ . To ensure message can be embedded, also according to (47), there should exists

$$Capacity = EPN \cdot BPE$$
$$= 2 \cdot EPN \cdot \lfloor \log_2 \lfloor \pi / m / \delta \rfloor \rfloor > 0$$

$EPN > 0$, Then $BPE > 0$,

$\lfloor \log_2 \lfloor \pi / m / \delta \rfloor \rfloor > 0$,

$\pi / m / \delta > 2$,

Finally,

$$m \cdot \delta < \pi / 2 . \qquad (55)$$

Equation (55) is the condition that message can be embedded.

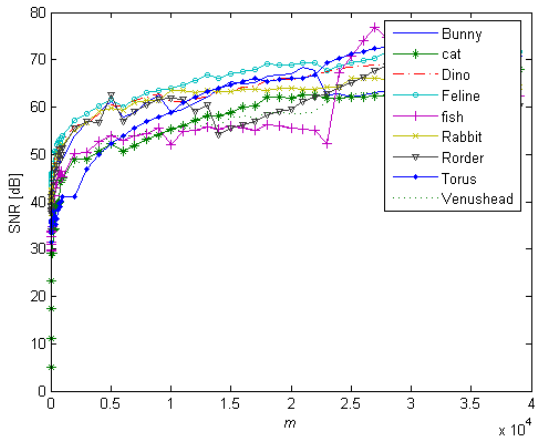


Figure 10. The relation of SNR and m

TABLE II. *EPN* OF EACH TESTED 3D MODELS.

File Name	*EPN*
Bunny	1742
cat	765
Dino	1707
Feline	3553
Fish	563
Rabbit	1894
Rorder	462
Torus	1069
Venushead	2661

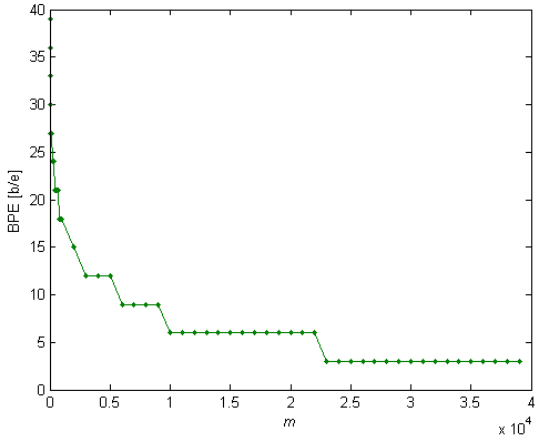


Figure 11. The relation of BPE and m

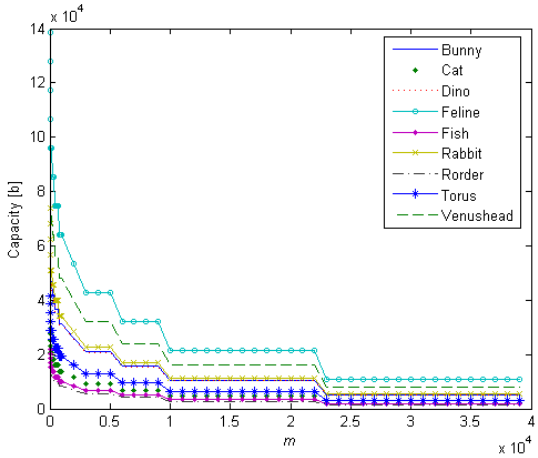


Figure 12. The relation of capacity and m

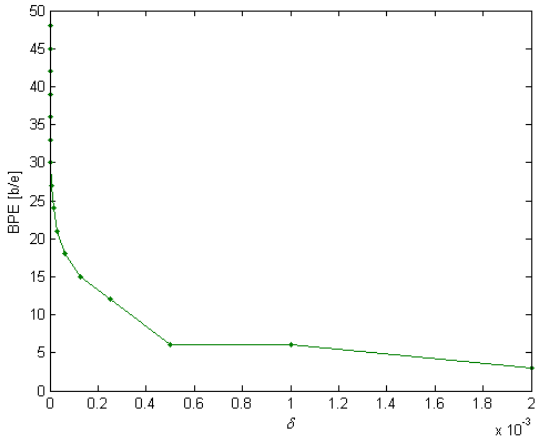


Figure 13. The relation of BPE and δ

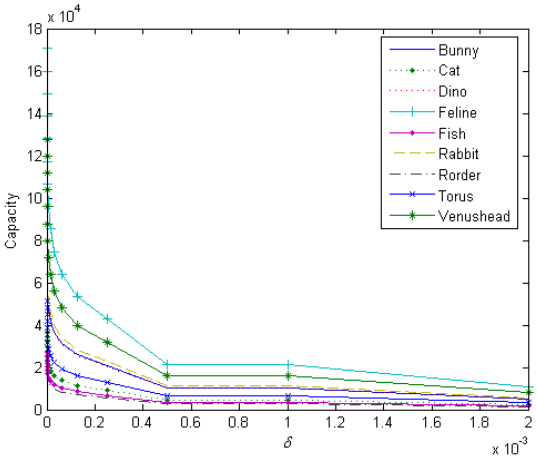


Figure 14. The relation of capacity and δ

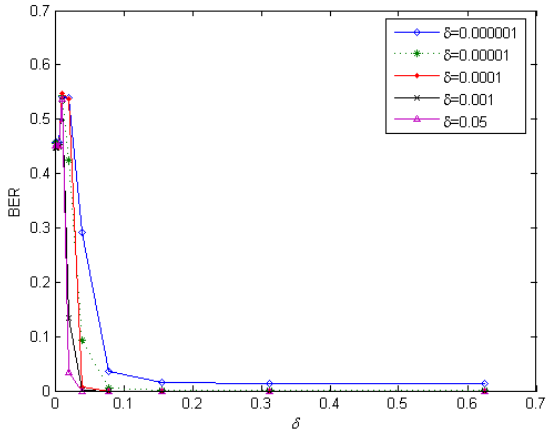


Figure 15. The BER of scaling-down attack

C. Robustness Performance

The proposed local coordinate system is invariant while the model has suffered from RST attack, the local coordinates of the 3D model is also invariant against RST attack, so the proposed data hiding algorithm can resist RST attack theoretically. But in fact, although translation and rotation scaling-up attack makes not bit error in the experiment, scaling-down attack certainly brings bit error. Because the data has limited precision, and scaling-down attack can lead to precision loss of vertex coordinate. The robustness against scaling-down attack is related with m and δ. We have carried out a great deal of tests. For $m \in [10,50000]$, we selected 8 values, for $\delta \in [0.00000001,0.005]$, we selected 8 values, thus we got about 64 embedded models. For each model, for scaling factor $\in [0.00000119,10.0]$, we selected 24 scaling factors, and use these scaling factor to generate about 1536 attacked models. Running the message extracting algorithm, message can be extracted. Combining with original message, BER (bit error ratio) can be calculated. But it is difficult to draw the results, for m, δ and scaling factor form 3D data. Plotting 2D figure also needs a large length of the paper. So the test data is fully provided. Here we just show Fig. 15 and Table III.

Curves when $m = 100$ and δ equal 5 different values are shown in the Fig. 15. From the figure we can find, generally the smaller δ is, the larger the BER is. Besides, when the embedded 3D model is scaled down to some extent, bit error occurs. There exist a scaling factor that BER is near 0.5, which means message is hardly retrievable, we call the scaling factor HRSF (Hardly Retrievable Scaling Factor). In Table III, we list HRSFs for different m and δ. From the table we can find, generally the smaller δ is, the larger the HRSF is. From the figure and the table we also can find BER and HRSF is correlative with δ and is not correlative with m. So we can draw a conclusion: the robustness of the algorithm mainly related with the parameter δ.

TABLE III HRSFS FOR DIFFERENT m AND δ, THE FIRST ROW IS VALUES OF m, AND THE FIRST COLUMN IS VALUES OF δ.

	10	100	1000	10000
0.000001	0.039	0.019	0.039	0.039
0.00001	0.039	0.019	0.019	0.019
0.0001	0.019	0.019	0.019	0.019
0.001	0.009	0.009	0.009	NULL

VI. CONCLUSION AND FUTURE WORK

In the paper, a new data hiding algorithm of 3D model based local coordinate system is designed directly for good imperceptibility and high capacity. The proposed data hiding algorithm has been tested to be robust against rotation, translation and uniform scaling. The proposed algorithm can be used in secret communication. m and δ can be used as secret key.

Although our algorithm shows several excellent properties, there is also some space to improve, if the message is encrypted before message embedding process, better security performances will be achieved. In other aspects, the local coordinate system can be used to design strong robust watermarking algorithms. Synchronization can be realized by self-synchronization techniques. Cropping attack can be resisted by repeated embedding technique. Independent with topology of 3D meshes, the EP structure can be used on some robust elements, the algorithm then can resist topology changing, noise adding and mesh simplification attack. That will be our future work.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under Grant No.60673014, Natural Science Foundation of Fujian Province under Grant No. 2008J0013 and Department of Education Foundation B of Fujian Province under Grant No.JB08049.

REFERENCES

[1] Alexander Bogomjakov, Craig Gotsman, and Martin Isenburg, "Distortion-free Steganography for Polygon

Meshes," Computer Graphics Forum, Proceedings of Eurographics'08, vol. 27 (2), pp. 637-642, April 2008.
[2] Ohbuchi R, Masuda H, Aono M., "Watermarking Three-Dimensional Polygonal Models," Proceedings of the ACM International Conference on Multimedia '97. Seattle, USA, November 10-13, pp. 261–272, 1997.
[3] Ohbuchi R,Masuda H,Aono M, "Watermarking three-dimensional polygonal models through geometric and topological modifications," IEEE Journal on Selected Areas in Communication, vol. 16(4), pp. 551-560, 1998.
[4] Praun E,Hoppe H,Finkelstein A, "Robust mesh watermarking," Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, pp. 325-334, 1999.
[5] S. Kanai, H. Date, and T. Kishinami, "Digital watermarking for 3d polygons using multireso-lution wavelet decomposition," Proc. Of International Workshop on Geometric Modeling: Fundamentals and Applications, Tokyo, Japan, pp. 296–307, 1998.
[6] Kangkang Yin, Zhigeng Pan, Jiaoying Shi, David Zhang, "Robust mesh watermarking based on multiresolution processing," Computers & Graphics vol. 25(3), pp. 409-420 , 2001.
[7] I. Guskov, W. Sweldensy, and P. Schroder, "Multiresolution signal processing for meshes," Proceedings of SIGGRAPH'99, New York, ACM Press, pp. 325-334, 1999.
[8] Benedens O., "Geometry-based watermarking of 3D models," IEEE Computer Graphics and Applications, vol. 19(1), pp. 46～55, 1999.
[9] Benedens O., "Watermarking of 3D polygon based models with robustness against mesh simplification," Proceedings of SPIE: Security and Watermarking of Multimedia Contents, San Jose, California, pp. 329～340, 1999.
[10] S.-H. Lee, and K.-R. Kwon, "A watermarking for 3D mesh using the patch CEGIs," Digital Signal Processing vol. 17, pp. 396–413, 2007.
[11] Wagner M., "Robust watermarking of polygonal meshes," Proceedings of Geometric Modeling and Processing, Hong Kong, pp. 201-208, 2000.
[12] Aspert N,Drelie E,Maret Y,et al., "Steganography for three-dimensional polygonal meshes," Proceedings of SPIE, Seattle, 4790, pp. 705-708, 2002.
[13] Maret Y, Ebrahimi T., "Data hiding on 3D polygonal meshes," Proceedings of ACM Multimedia and Security Workshop, Magdeburg, pp. 68-74, 2004.
[14] Sun S, Pan Z, Li L, et al., "Robust 3D model watermarking against geometric transformation," Proceedings of the 8th International Conference on Computer-Aided Design and Computer Graphics, Macao, pp. 87-92, 2003.
[15] Shusen Sun, Zhigeng Pan, "A Blind 3D Mesh Watermarking Scheme Based on Local Coordinate System," IEEE 7th Workshop on Multimedia Signal Processing, pp. 1-4, Oct, 2005.
[16] Thomas Harte, Adrian G. Bors, "Watermarking 3D models," ICIP (3), pp. 661-664, 2002.
[17] Adrian G. Bors, "Watermarking mesh-based representations of 3-D objects using local moments," IEEE Transactions on Image Processing vol. 15(3), pp. 687-701, 2006.
[18] ShanchaoYang, Xiaoqing Feng, Zhiqiang Yao, Zhigeng Pan, Bin Weng, "A 3D Model Data Hiding Algorithm based on Local Coordinate System," 2008 11th IEEE International Conference on Communication Technology Proceedings, pp. 781-784, 2008

Shaochao Yang received the B.S. degree in computer science & technology from School of Computer Science, Qufu Normal University, Qufu, China, in 2003, the M.S. degree in compute application technology from School of Computer Science & Technology, Soochow University, Suzhou, China, in 2006.

He works as a LECTURER in School of Mathematics and Computer Science, Fujian Normal Unversity, he also does research at Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, China. His current research interests include 3D watermarking and steganography, processing and analysis of image and graphics.

Zhiqiang Yao received the B.Sc. degree from Fujian Normal University, China in 1989, the M.Sc. from East China Normal University in 1992.

He is currently a Professor at Faculty of Software of the Fujian Normal University where he leads the Key Lab of Network Security and Cryptology, the Intelligent Software Engineering Center. His current research interests lie in Pattern Recognition, Graphics and Image Processing, and multimedia security.