A Robust Three-Dimensional Model Watermarking Algorithm Based on Connected Vertices Clustering and Star Topology

Chen-Chung Liu[1], Pei-Chung Chung[2]
National Chin-Yi University of Technology, Department of Electronic Engineering
No.35, Lane215, Sec.1, Chung-Shan Rd., Taiping City, Taichung County 411 Taiwan, R.O.C.
[1]ccl@ncut.edu.tw
[2]iam.abaw@gmail.com

***Abstract*— Nowadays, with the fast development of internet and the rapid development of digital media information processing and content distribution, 3D models are easy to transmit and duplicate unauthorized reproduction to become a serious problem. The intellectual properties protection of 3D model is necessary and not negligible. The problems of the 3D model watermarking scheme are only small amount embeddable space and not robust enough.**

This paper proposes a cluster algorithm based on vertices connectivity and the 3D model reconstruction with star- topology to construct a robust watermarking scheme for 3D model. In the proposed scheme, the center of each star- topology is selected to be a candidate vertex and several no overlapped edges are selected as the candidate edges for watermark embedding. There are three advantages compare with other methods: (A) The secret image has very high security owing to using chaotic mechanism to scatter the watermark. (B) The result host 3D model after watermark embedding possesses excellent imperceptibility without noticeable degradation. (C) The watermarks embedded in the 3D models are robust against the similarity transformation attacks and the cropping attack.

***Keywords*— 3D model, watermark, cluster, star- topology.**

1. INTRODUCTION

Nowadays, with the fast development of internet and the rapid development of digital media information processing and content distribution, digital information is easy to transmit and duplicate unauthorized reproduction becomes a serious problem. There is an urgent demand for techniques to protect the copyright of the original digital data and to prevent unauthorized duplication or tampering. Generally there are two good technologies for the intellectual property and copyright protection; cryptography and watermarking.

The cryptography technology cut off the access of the unauthorized person after the multimedia information is encrypted [1- 5]. However, it cannot prevent the unlawful action of an authorized person and cannot solve the problem that some copyright owners assert their ownerships for one content. To solve the problems of the cryptography, there have been much researched in watermarking technology, which is the end-step in information security and protects the copyright of owner by embedding the watermark into the multimedia information. A lot of research has been carried out to protect the copyright protection of image, video, and audio. Digital watermarking is a technique designed to hide information in a certain type of digital data.

Embedded watermarks can be used to enforce copyright, data authentication or to add information to the data. Ideally, the watermark should not interfere with the intended purposes of the data. In last decades, most of the research on watermarking has concentrated on audio signals, images, or video sequences [6- 9]. The watermarking algorithm for 3D models are few because the watermarking technique for 3D model has many difficulties for the following reasons: (A) compared with images, only a small amount of data (ie. vertices) is available for watermark embedding; (B) no unique representation nor implicit ordering of 3D model data exists; and (C) no robust transformation field could be used to embed watermark [10]. In recent years, 3D graphic models, such as VRML, MPEG-4, and 3D geometrical CAD, have become very popular leading the development of 3D watermarking algorithms to protect the copyright of 3D graphic models [11]. Ohbuchi [12] proposed several watermarking algorithms for 3D models: triangle similarity quadruple (TSQ) embedding algorithm, tetrahedral volume ratio (TVR) embedding algorithm, and mesh density pattern embedding algorithm. However, these algorithms are not sufficiently robust against attacks. Beneden [13] also described a watermarking system that is based on affine registration of meshes in order to compensate for affine transformations and used it in the watermarking detection procedure. Although this algorithm is robust against the randomization of points, mesh altering, and polygon simplification, it is not robust against cropping attacks. Kang kang Yin et al. [14] proposed a new mesh watermarking scheme for triangular meshes. In their scheme watermark information is embedded into a suitable coarser mesh which consists of the low-frequency components. The scheme is not robust against crop operation.

The term “cluster” is an unusual aggregation of events that are grouped together in time or space [15]. Cluster analysis is one of the basic tools for exploring the underlying structure of a given data set. The primary objective of cluster analysis is to partition a given data set of multidimensional vectors (patterns) into so-called homogeneous clusters such that patterns within a cluster are more similar to each other than patterns belonging to different clusters. Cluster seeking is very experiment-oriented in the sense that cluster algorithms that can deal with all situations are not yet available; each approach has its own merits and disadvantages [16]. Further information on clustering and clustering algorithms can be found in the literature [17-20]. Clustering has been applied in a wide variety of engineering and scientific disciplines such as medicine, psychology, biology, sociology, pattern recognition, and image processing. We believe that clustering for 3D model vertices will have much effect in the 3D model watermark. In this proposed, we develop a cluster algorithm based on vertices connectivity and the 3D model reconstruction with star- topology to construct a robust watermarking scheme for 3D model.

To explore the utility and demonstrate the efficiency of the proposed scheme, simulations under various conditions are conducted. The experiment results show that our proposed scheme is a robust watermarking scheme for 3D model. The remainder of this paper is organized as follows: In Section 2, reference point and candidate vertices finding. In Section 3, the watermark embedding algorithm is presented. The watermark extraction process from the host 3D model is illustrated in Section 4. Empirical results are presented in Section 5. Finally, Section 6 concludes this paper.

2. WATERMARK EMBEDDING

A robust watermarking scheme for 3D models must be extremely secure without reducing the visual quality of the host 3D model, and the embedded watermark must be robust to against the attacks. In order to construct a superior watermarking algorithm for 3D model, several schemes are used in this paper to achieve the goal. The overall watermark embedding process is shown in Fig.1.

For enhancing the security of embedded watermark, the two dimensional white- black watermark W is first arranged into a binary sequence, S, using arranging watermark scheme (AWS). The binary watermark sequence is hashed into four hashed watermark sequences, $\{\mathrm{m}\}_{\mathrm{i}}, \mathrm{i} = 1,2,3,4$, by chaotic mechanism (CM) using 4 pseudo random sequences $\{\mathrm{PNs}\}$ generated by 4 private keys $(\mathrm{Key}_{\mathrm{i}}, \mathrm{i} = 1,2,3,4)$, respectively [21]. The proposed algorithm combines $\{\mathrm{m}\}_{\mathrm{i}}, \mathrm{i} = 1, 2,3$ to form a sequence of three bits radix 2 number,$\{(\mathrm{m}_1\ \mathrm{m}_2\ \mathrm{m}_3)\}_2$, and then converts the sequence of three bits radix 2 numbers to the sequence of decimal (radix 10) numbers,$\{\mathrm{D}\}_{10}, 0 \leq \mathrm{D}_{10} = (\mathrm{m}_1\ \mathrm{m}_2\ \mathrm{m}_3)_2 \leq 7$.

On the other hand, the proposed algorithm finds the feature vertex, the feature triangular mesh, and the feature edge for the host 3D model X. Then, the proposed algorithm translates and rotates the host 3D model X into the standard pose; the feature vertex is coincided with the original point of the Cartesian coordinate system, the feature edge is lying on the positive x-axis of the Cartesian coordinate system, and the feature triangular mesh is lying on the first octant of the Cartesian coordinate system. The weight of each vertex of the calibrated 3D model is then calculated. At the same time, the host 3D model with is discomposed into several sub-models (clusters) according the connectivity structure of vertices of the host 3D model. The vertex that has the maximum weight in a cluster is assigned as the center of the first star- topology element of that cluster. Then, the algorithm reconstructs each cluster, respectively, with star- topology elements expanded from the first star- topology element of each cluster.

Some edges of the host 3D model are collected as the candidate edges for watermark embedding. These candidate edges are selected by the pseudo random sequence $\{\mathrm{PN_e}\}$, the current element of $\{\mathrm{PN_e}\}$ and the corresponding element of $\{\mathrm{D}\}_{10}$ are embedded into the selected edge one by one until all the elements of $\{\mathrm{PN_e}\}$ are used. Then, the center vertex of each star topology element is collected and arranged into a sequence $\{\nu_c\}$ according to their weights. The vertices for watermarking are selected from $\{\nu_c\}$ by taking the elements of $\{\mathrm{PN_v}\}$ as the subscripts of the elements of $\{\nu_c\}$. The corresponding element of $\{\mathrm{m}\}_4$ is embedded into the selected vertices one by one until all the elements of $\{\mathrm{PN_v}\}$ are used. The detail descriptions of each block of the proposed watermark embedding algorithm are described in the follows.

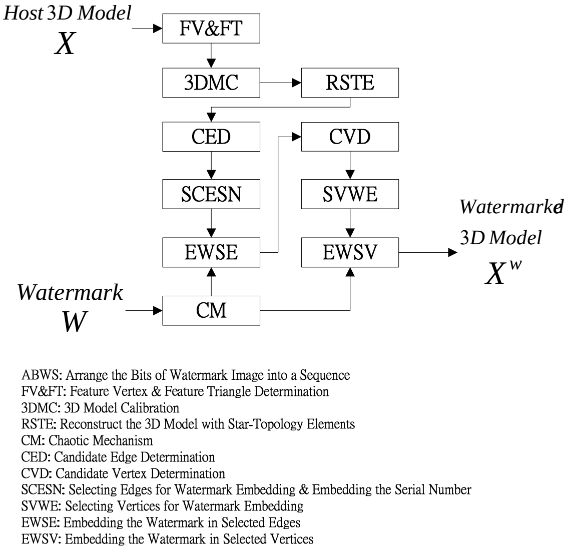

Host 3D Model
X
FV&FT
3DMC
RSTE
CED
CVD
SCESN
SVWE
EWSE
EWSV
Watermarked 3D Model
X^w
Watermark
W
CM
ABWS: Arrange the Bits of Watermark Image into a Sequence
FV&FT: Feature Vertex & Feature Triangle Determination
3DMC: 3D Model Calibration
RSTE: Reconstruct the 3D Model with Star-Topology Elements
CM: Chaotic Mechanism
CED: Candidate Edge Determination
CVD: Candidate Vertex Determination
SCESN: Selecting Edges for Watermark Embedding & Embedding the Serial Number
SVWE: Selecting Vertices for Watermark Embedding
EWSE: Embedding the Watermark in Selected Edges
EWSV: Embedding the Watermark in Selected Vertices

Fig. 1. The flow chart of watermark embedding.

2.1 PN Sequence

PN sequence is a sequence with no repeating elements, for example a type two sequence of length=4096 is a sequence that is a random and no repeating permutation of numbers 1, 2, 3,…, 4096 in a line. In this paper, we used the language instruction "random()" and a random seed (a number of $10^6 \sim 10^{11}$) to generate a string of random number(0.0000001~0.9999999). The random numbers times the number 4096, and then the products are taken round into random integer sequence. The following PN sequence generator flowchart shown in Fig. 2 is used to generate the PN sequence with the string of random integer number.

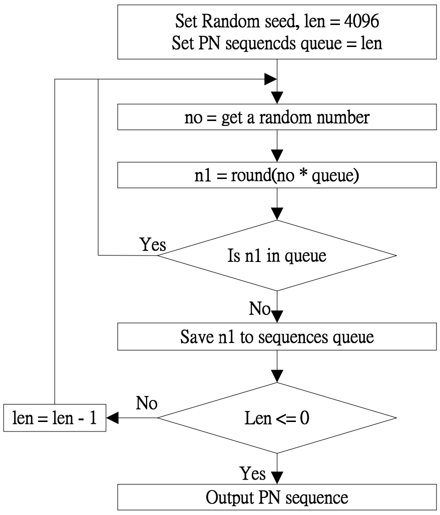


Fig.2. The flowchart of PN sequence generator.

2.2. Chaotic Mechanism

For security, the sequence of watermark bits is pre-permuted into chaotic state by the chaotic mechanism with a user's key. In this paper, the proposed algorithm uses a pseudorandom number traversing as the chaotic mechanism to permute the original sequence of watermark bits. The relation between the chaotic bit sequence and the original sequence of watermark bits is expressed in the following equations:

$$\begin{aligned} \mathrm{m_c(j)} &= \mathrm{m(j')}, 1 \le \mathrm{j}, \mathrm{j'} \le L \\ \mathrm{j} &= \mathrm{permutation}\ (\mathrm{j'}), \end{aligned} \qquad \textbf{(1)}$$

where L is the length of the original sequence of watermark bits. The permutation operation is carried out with using equation (1) and PN sequences generated by the PN sequence generator. Fig. 3 shows the watermark image hashed and recovered by chaotic mechanism.

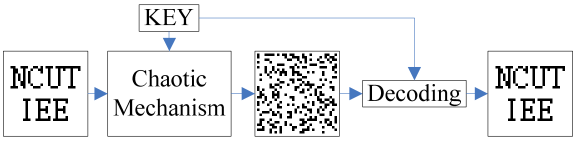


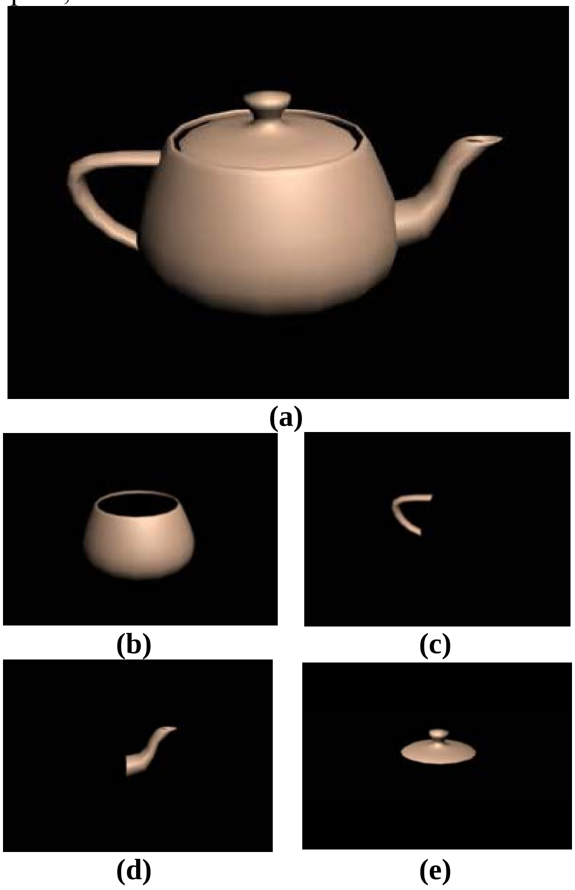
Fig. 3. Chaotic Mechanism

2.3. Clustering of 3D model

Recently, three dimensional (3D) models are used in various applications, such as computer graphics, virtual reality, 3D animation and synthetic imaging systems. Many representations have been proposed for 3D models [22].In particular, triangular meshes are frequently being used to represent 3D model surfaces. A triangulated meshes 3D model is represented by its topological, geometry and attributes list. The topological list describes the connectivity relations among vertices and the incidence relation between triangles and vertices. The geometry list specifies the locations of the vertices. The attribute list generally consists of colors, normal vectors and texture information, which are needed to paint and shade the model. Geometry and attribute list are specified by floating-point numbers, whereas topological list are represented by integer indices. A common scheme for representing and storing polygon meshes is to use a list of vertex geometry coordinates to store the geometry and a list of vertex indices for each face to store mesh connectivity. Edges are implied and not explicitly stored (edge is a line segment that connected two adjacent vertices). [23]

The coordinates of 3D model vertices should be changed when people embed watermarks into 3D model vertices. On the other hand, vertices are connected to adjacent vertices with edges to form triangle meshes, connected triangle meshes are collected to form a simple 3D model, and several simple 3D models are grouped to construct a 3D model. For the convenience of finding candidate vertices for watermark embedding, one has better to decompose the 3D model into none-overlapped simple 3D models according to the 3D model's original structure.

Clustering of data in multi- dimensions has been applied in a variety fields like as image segmentation, pattern recognition, and so on. Clustering is the process of partition the data into groups of items such that items within a group are similar to one another and different from those in other groups, the similarity between items is determined based on their features. We cluster a 3D model into clusters (simple 3D models) based on the connectivity relation of vertices from the topology list of the 3D model. Fig. 4 shows the result clusters of a 3D model teapot with 4 objects: body, handle, spout, and lid.


(a)

(b) (c)

(d) (e)

Fig. 4. The result clusters of a 3D model teapot constructed with 4 simple objects: (a) integrated teapot, (b) cluster body, (c) cluster handle, (d) cluster spout, and (e) cluster lid.

2.4. Star- topology

A three dimensional model is often represented with its surface that composed by triangular meshes. In a triangular mesh representing 3D model, each edge is either shared by two triangles, called as an interior edge, or belongs to a single triangle, called a boundary edge. A closed loop formed by linking up such boundary edges forms a boundary of the mesh, and a 3D model may have multiple boundaries. Two triangles are called as adjacent triangles if they share an edge. A star- topology is a polygon that admits a triangulation in which all triangles have a common vertex, called as center vertex. The other vertices of the polygon are called as front-vertices. The front-vertices and the edges between them construct the front of a star- topology described in Fig. 5. The number of front-vertices *d* is the degree of the center of the star- topology. [24]

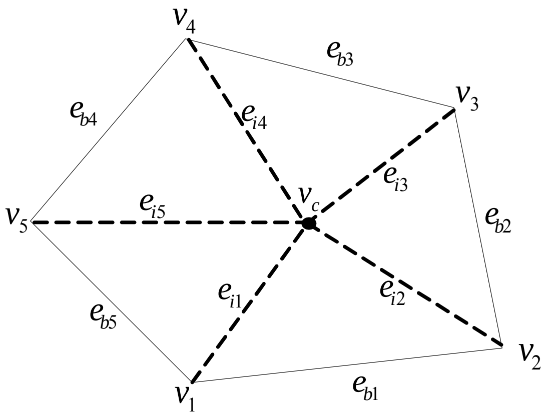
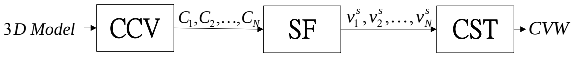


Fig. 5. A star- topology with center vertex v_c, front vertices $v_1, v_2, \dots, v_5$, interior edges $e_{1^1}, e_{1^2}, \dots, e_{1^5}$, and front edges $e_{b^1}, e_{b^2}, \dots, e_{b^5}$.

When there is a revision against to a vertex, the relative positions among the revised vertex and those vertices that connected to the revised vertices are actually changed. If this kind of chain-reaction does not be overcome, then the 3D model may be distorted and the information which already embedded at vertices before shall be destroyed. In order to overcome this question, one need to avoid embedding watermark bits on adjacent vertices. In order to get the goal, the proposed algorithm uses star- topology elements to reconstruct the 3D model such that any two adjacent star- topologies have several sharing front- vertices, and embeds watermark at the center vertex of each star- topology element.

The proposed algorithm uses a kind of adjacent vertices searching method to reconstruct the 3D model with the star- topology elements. The searching method is shown in Fig. 6. Fig. 7 shows examples of candidate vertices selected with our proposed method.



CCV: Cluster the 3D Model with the Connectivity of Vertices
SF: Starting Vertex Finding
CST: Construct the 3D Model with Star-Topology
CVW: Candidate Vertices For Watermarking

Fig. 6. The flow chart of searching method of candidate vertices for watermarking.

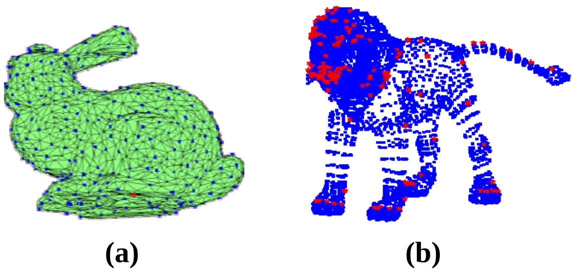


Fig. 7. Candidate vertices (points in blue) and starting vertices (points in red): (a) simple 3D model Bunny; (b) compound 3D model lion with twelve 165 objects

2.5. Watermark embedding by edge vector modulation

The front edges of each star topology element are selected and collected to be the set of candidate edges according the rule that the intersection of any two candidate edges is empty set. Fig. 8 shows an example of the candidate edges (red edges) in the star topology structure 3D model.

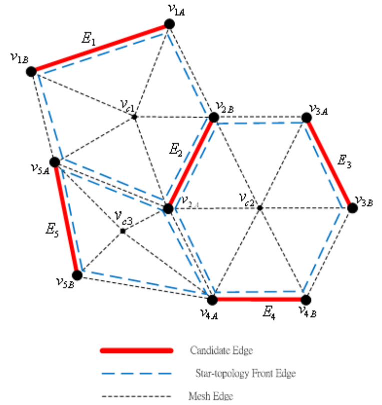


Fig.8. The candidate edges in the star topology structure 3D model

After the set of candidate edges is constructed, the length of the watermark bit sequence and a key are input to the PN sequence generator to produce a PN sequence PN_e. The edges for embedding are selected one by one from the set of candidate edges according to the PN sequence PN_e. For the *j*-th embedding edge $\overline{A_jB_j}$ the *j*-th embedding edge vector vertex $\overrightarrow{A_jB_j}$, A_j is defined as the unmovable vertex if it has larger weight, the less weight vertex B_j is defined as the movable vertex. The decimal watermark number is embedded into the z-component of the corresponding embedding edge vector. At the same time, the corresponding three watermark bits are embedded into the decimal fraction part of the x- component, and the check bits are embedded into the y-component of the edge vector, respectively. After the watermark embedding is finished, the proposed algorithm uses the x- component, y-component, and z- component of the edge vector of the watermark embedded to determine the new coordinate of the movable vertex B of the watermark embedded edge, and replaces the original coordinate of the movable vertex B of the watermark embedded edge with the new coordinate to finish a bit's watermark embedding. These steps are repeated until all bits of watermark are embedded.

2.6. Watermark embedding with angle shift keying modulation (ASKM)

We input the length of the watermark bit sequence and a seed into the PN sequence generator to produce a PN sequence. The embedding vertex is selected (EVS) one by one from the candidate vertices sequence $\{ \nu_c \}$ according to the PN sequence. We choose a triangle from the star- topology of the embedding vertex as the embedding triangle; the embedding triangle has the minimum area among the triangles that shearing the embedding vertex. In the embedding triangle, we define that the shorter side $\overline{\nu_c s}$ between the two sides sharing the embedding triangle as the feature side and its corresponding angle φ as the feature angle. The geometry meaning of embedding triangle, feature side, and feature angle are shown in Fig. 9

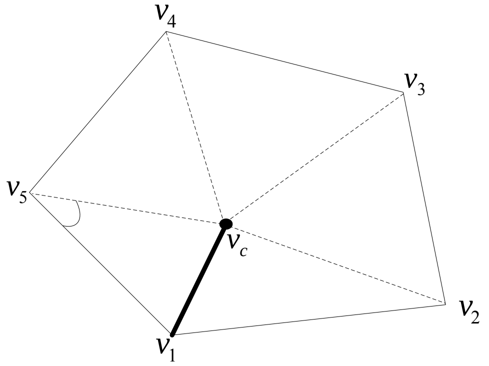


Fig. 9. The geometry meaning of embedding triangle $\Delta\mathbf{\nu_c\nu_5\nu_1}$, feature side $\overline{\mathbf{\nu_c\nu_1}}$, and feature angle $\angle\mathbf{\nu_c\nu_5\nu_1}$

In the angle shift keying modulation (ASKM), the angle interval $[0^o, 90^o]$ are equally divided into 4×10^n, n = 1, 2, …, 5 subintervals shown as in Fig. 10.

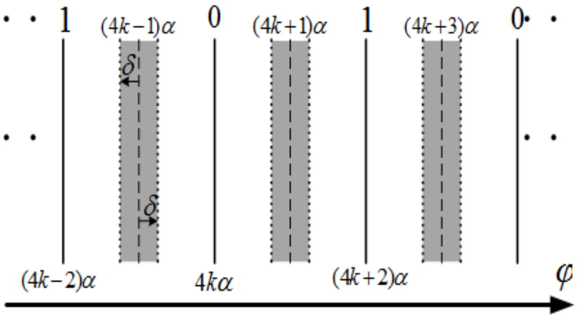


Fig. 10. The signal constellation of the ASKM

After EVS and ASK, the secret information is then embedded into host 3D model X at the selected embedding vertex by watermark bits embedding (WBE). In the watermark bits embedding, the embedding vertex $\nu_c(x_c, y_c, z_c)$ is translated to $\nu'_c(x'_c, y'_c, z'_c)$ along the feature side $\overline{\nu_c s}$ such that the modulation feature angle equals to φ′. The coordinates of ν'_c are determined by the following formulas:

$$x'_c = (1+t) * x_c - t * x_s \qquad \textbf{(3)}$$
$$y'_c = (1+t) * y_c - t * y_s \qquad \textbf{(4)}$$
$$z'_c = (1+t) * z_c - t * z_s \qquad \textbf{(5)}$$

where parameter *t* is obtained from the sine theorem in triangular geometry by the formula

$$t = \sin(\varphi - \varphi') * \sin(A) / (\sin(A+\varphi) * \sin(\varphi)) \qquad \textbf{(6)}$$

The translation of embedding vertex is shown in Fig. 11. The above embedding process is repeated for each secret bit in {m} and its corresponding vertex, then the watermark embedding is finished.

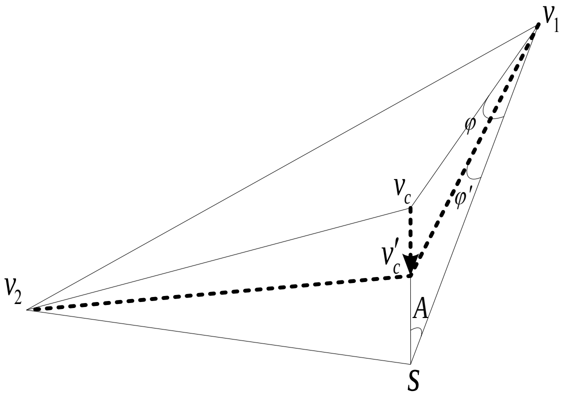


Fig. 11. The translation of embedding vertex from original position ν_c to new position ν'_c.

3.Watermark Extraction Algorithm

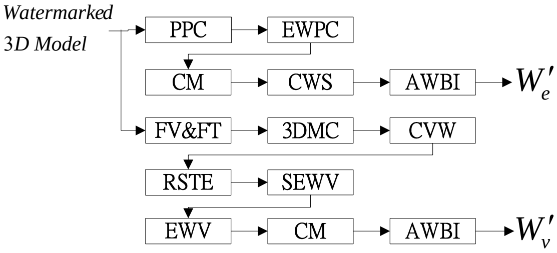
The watermark extraction is the inverse process of a watermark embedding process; it is to recover the original watermark image from a watermarked 3D model. Most of The retrieval algorithm is identical to the embedding process. The flow chart of the proposed watermark recovering process is shown in Fig. 12. The watermarked 3D model X’ is divided into several sub-models (clusters) $C'_1, C'_2, \ldots, C'_n$ according the connection of vertices X’. The feature vertex, feature edge, and the feature triangle of the watermarked 3D model are determined, respectively. Then, each cluster is reconstructed with star- topology elements, the candidate watermarked vertices and the candidate watermarked edges are collected, respectively. The same PN sequences used in the embedding process is used to select the watermarked vertex and the watermarked edges for watermarks extraction.

In the ASK demodulation, the secret bit m’ is extracted from the modulated feature angle φ' of the selected feature triangle for each selected star- topology. In the process of the ASK demodulation the modulated feature angle φ' is processed to recover the embedded secret bit m’ according to the following formula:

$$m' = \begin{cases} 0\,,\ if\ \ (4k-1)\alpha < \varphi^{w} \le (4k+1)\alpha \\ 1\,,\ if\ \ (4k+1)\alpha < \varphi^{w} \le (4k+3)\alpha \end{cases} \qquad \textbf{(7)}$$
.

After all the secret bits are extracted from the ASK demodulation, they are rearranged into the two- dimensional image H'. By passing H_v’ through the inverse chaotic Mechanism (ICM), the recovered watermark W'_v embedded in vertices is obtained.

On the other hand, the proposed algorithm extracts the watermark bits from the x-component, extracts the check bits from the y-component, and extracts the decimal watermark number from the z- component of the selected edge vector of the watermarked 3D model, respectively. After all the secret bits are extracted from the selected edges, they are rearranged into the two- dimensional image H'_e. By passing H' through the inverse chaotic Mechanism (ICM), the recovered watermark W'_e embedded in vertices is obtained.



PPC: Pick out the Possible Candidate Edge by Identifying Serial Number
EWPC: Extract Watermark Information from Possible Candidate Edge
FV&FT: Feature Vertex & Feature Triangle Determination
3DMC: 3D Model Calibration
CVW: Calculating Vertices Weight
CMCV: Clustering the Model with the Connectivity of Vertices
RSTE: Reconstruct the 3D Model with Star-Topology Elements
SEWV: Selecting Candidate Vertices
EWV: Extracting Watermark from Candidate Vertices
ITA: Inverse Total Automorphism
CWS: Combine all watermark information to Calculate real watermark sequence
CM: Chaotic Mechanism
AWBI: Arrang the Extracted Watermark Bit Sequence into a 2D Image

Fig. 12. The flow chart of the watermark recovering process.

4. EXPEREMENT RESULTS

Several 3D models are conducted to simulate under various conditions to explore the utility and demonstrate the efficiency of the proposed scheme. The 3D models Bunny (simple model, 1494 vertices, and 2915 triangular meshes) and lion (165 objects, 17352 vertices, and 32098 triangular meshes) are used in simulation for demonstrating the performance of the proposed scheme. The logotype image of NCUT with sizes 8*8 and 16*16 are used as watermarks for the host 3D model Bunny, the chess board with size (8*k*)*(8*k*), k= 1, 2, . . ., 9 are used as watermarks for the host 3D model lion. In order to demonstrate host 3D model possesses excellent imperceptibility without noticeable degradation, we compare it with the test 3D models. Fig. 13(a) shows the original 3D model of Bunny. Fig. 13(b) is the embedded 3D model of Bunny with watermark of size16*16. Fig. 14(a) shows the original 3D model of lion. Fig. 14(b) is the embedded 3D model of lion with watermark of size 64*64.

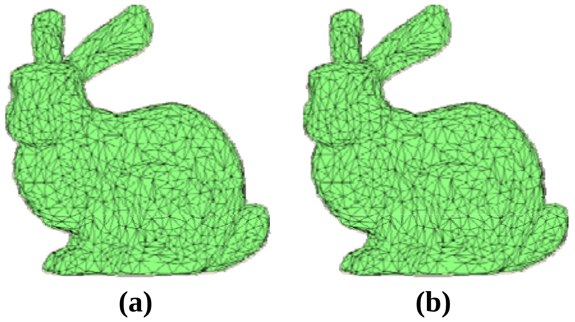


Fig. 13. The quality of the host 3D model: (a) the Original 3D model Bunny, (b) the Watermarked 3D model Bunny with SNR 144.4442.

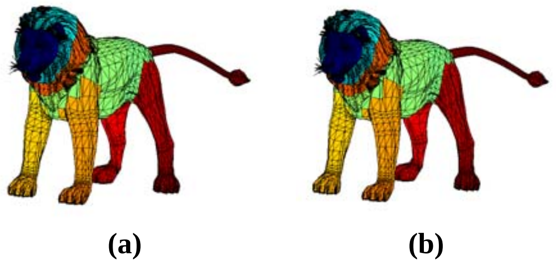


Fig. 14. The quality of the host 3D model: (a) the Original 3D model lion, (b) the Watermarked 3D model lion with SNR 140.4638.

Imperceptibility is an important factor in watermarking. We employ the SNR [25] to measure the degree of transparency in this paper. To measure the SNR of a watermarked 3D mesh object, the following formula is used:

$$SNR = 10\log_{10}\frac{\sum_{i=0}^{N-1}\left(x_i^2 + y_i^2 + z_i^2\right)}{\sum_{i=0}^{N-1}\left((\tilde{x}_i + x_i)^2 + (\tilde{y}_i + y_i)^2 + (\tilde{z}_i + z_i)^2\right)} \quad \textbf{(8)}$$

Where (x_i, y_i, z_i) and $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ are the coordinates of vertex $\mathrm{v}_\mathrm{i}^\mathrm{c}$ before and after the watermark embedding, respectively. We calculate the SNR for embedded Bunny and embedded lion. They are arranged in table 1 to show the watermark's perceptibility and sketched into SNR_ Watermark Size to describe the change of SNR caused by the watermark size. The SNR of the watermarked 3D model is almost reciprocal to the size of watermark.

Table.1
The performance of our scheme for the test 3D model lion

No. of Watermark Bits	64	1024	4096
SNR of Compound Scheme	156.1	144.3	140.4
SNR of Center Vertex Scheme	174.1	162.6	156.9
SNR of Edge Scheme	156.2	144.4	140.5

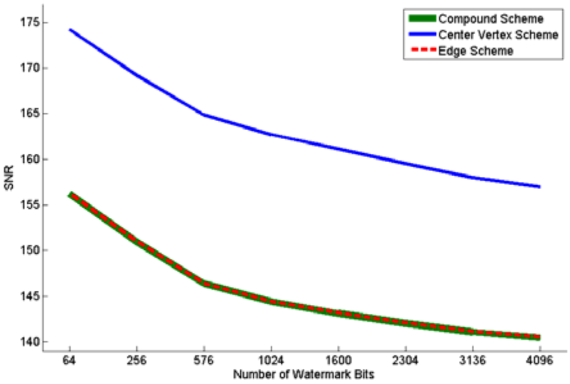


Fig. 15. The performance of our scheme for the test 3D model lion

As a practical watermarking system, other than imperceptibility and the quality of secret watermark, robustness of the system is another important issue. Here, we select the vertices for watermarks embedding from the ordered sequence of star- topology centers. The order of embedded vertices is unchanged when the watermarked 3D model is attacked with rotations, scaling, translation, and their combination. So, we can extract the watermark bit by bit from the watermarked 3D model correctly. In order to demonstrate the robustness, we used rotation, translation, scaling, and cropping attacks to test our scheme. The similarity measurement of a watermark depends on the knowledge of the experts, the experimental conditions, etc. Therefore a quantitative measurement is necessary to provide a fair judgment of the extracted fidelity. In this paper, we use the normalized correlation (NC) and the standard correlation coefficient (SCC) between the original watermark W and the extracted watermark W' as the similarity measurement. We also use the bit error rate (BER) to indicate the extraction fidelity. The NC, SCC, and BER value of extracted watermark are defined respectively as the following formulas:

$$NC = \frac{\sum_i\sum_j W(i,j)W'(i,j)}{\sum_i\sum_j [W(i,j)]^2} \quad \textbf{(9)}$$

$$SCC = \frac{\sum_i\sum_j (W(i,j)-W)(W'(i,j)-W')}{\sqrt{\sum_i\sum_j (W(i,j)-w)^2}\sqrt{\sum_i\sum_j (W'(i,j)-W')^2}} \quad \textbf{(10)}$$

$$BER = \frac{Ne}{W_x \times W_y} \quad \textbf{(11)}$$

We calculate the NC, SCC, and BER after attacks for embedded Bunny and embedded lion and arranged them in a table to show the watermark's fidelity and robustness. The simulation results of embedded 3D models after similarity transformation attacks are shown in Fig. 16, Fig. 17, and Table2 respectively. Fig. 16 presents an example of Bunny under attacks. **;** (a) shows watermarked Bunny under the attack with rotation -213° about X-axis, 125° about Y-axis , and -36° about Z-axis, (b) shows watermarked Bunny under the attack with scaling to 0.3times , (c) shows watermarked Bunny under the attack with rotation 57° about the vector A=[0, 0.6, 0.8], (d) is the extracted watermark form (a), (e) is the extracted watermark form (b), and (f) is the extracted watermark form (c). Fig. 17 presents an example of the 3D model lion attacks. **;** (a) shows watermarked Dragon under the attack with rotation 333°about X-axis, -58° about Y-axis , and -197° about Z-axis, (b) shows watermarked Dragon under the attack with scaling to 0.5times , (c) shows watermarked Dragon under the attack with rotation 299° about the vector A=[0, 0.6, 0.8], (d) is the extracted watermark form (a), (e) is the extracted watermark form (b), and (f) is the extracted watermark form (c). Fig. 16 and Fig. 17 show that our watermark scheme is robust enough against the affine transformation attacks.

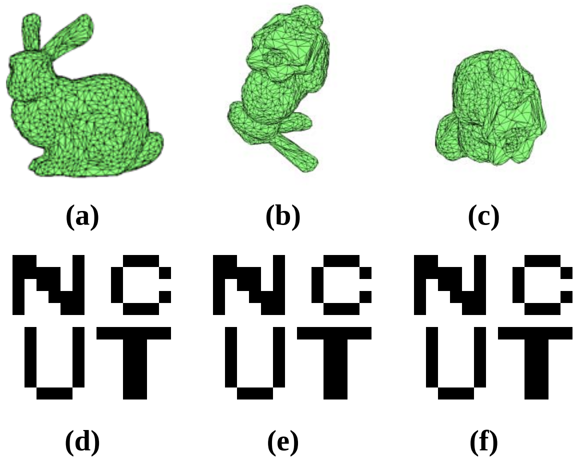


Fig. 16. The extraction watermarks from the attacked 3D model Bunny watermarked with a 16×16 "NCUT" watermark; (a) Attack with Scaling to 0.5 times, (b) Attacks with Rotation 15° about X-axis, -20° about Y-axis , and 25° about Z-axis, (c) Attacks with the combination of Scaling to 0.7 ,Rotation 32° about X-axis, -87° about Y-axis , -57° about Z-axis, and translation along Y-axis 10 units(d) Extraction Watermark for (a), (e) Extraction Watermark for (b), (f) Extraction Watermark for (c)

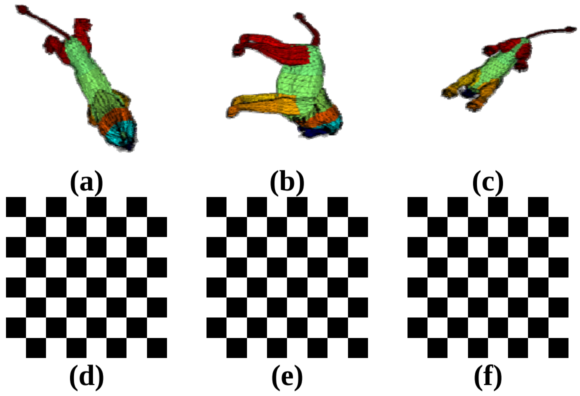


Fig. 17. The extraction watermarks from the attacked 3D model lion watermarked with a 16×16 chess-board watermark; (a) Attack with Scaling to 0.7 times, (b) Attack with scaling to Rotation 25° about X-axis, 45° about Y-axis , and 65° about Z-axis, (c) Attack with Scaling to 1.4 and Rotation -23° about X-axis, -67° about Y-axis , and 55° about Z-axis, (d) Extraction Watermark for (a), (e) Extraction Watermark for (b), (f) Extraction Watermark for (c)

Table.2
The performance of our scheme for the recovered watermark under attacks

3D Model	Attacks	BER	NC	SCC
Bunny	S(0.5)	0	1	1
	R(15°,-20°,25°)	0	1	1
Lion	S(0.7)	0	1	1
	R(63°,51°,-37°)	0	1	1
	T(15, 10, 0)	0	1	1

	R(-18°, 38°, 60°)
	S(1.5)
		0	1	1

	T(30, 17, 23)
	R(34°, -67°, 59°)
	S(0.9)
		0	1	1

T(*x*, *y*, *z*): Translation with displacement < *x*, *y*, *z* >
S: Scaling
R $\left(\theta_x^0, \theta_y^0, \theta_z^0\right)$:Rotation θ_x^0 about X-axis, θ_y^0 about Y − axis, and θ_z^0 about Z − axis.

According to Table 2, the BER are zero, the NC and SCC are one for each extracted watermark from each attacked 3D model. This means that we can accurately and completely extract the watermark from the watermarked 3D model under the attacks of the combination of translation, rotation, and scaling. These results prove that the proposed approach is a robust watermarking scheme that is strength enough against the similarity transformation attack for 3D model.

On the other hand, the watermarked 3D model lion is conducted to test its robust against the cropping attack. The cropping attack is taken by cutting several clusters from the watermarked 3D model first, then the survived watermark is extracted from the residual 3D model, and the error of the extracted watermark is also evaluated. The cropping is 5% of the total clusters each time upon 95%. The percentage of cropped vertices versus the percentage of cropped clusters, the percentage of cropped triangular meshes versus the percentage of cropped clusters, the correct rate of extracted watermark versus the percentage of cropped clusters, and error rate of extracted watermark versus the percentage of cropped clusters are illustrated in table 3 and fig. 18. The simulation results of embedded 3D models after cropping attacks are shown in Fig. 19**;** (a) shows the residual watermarked lion and the corresponding extracted watermark under the cropping attack with cutting off 10 objects, (b) shows the residual watermarked lion and the corresponding extracted watermark under the cropping attack with cutting off 70 objects, (c) shows the residual watermarked lion and the corresponding extracted watermark under the cropping attack with cutting off 130 objects. The errors of the extracted watermarks are all less than 22.3%.

Table.3
The percentage of cropping cluster, cropping vertices, cropping meshes, and the error of extracted watermark.

Percent of Cropping Clusters (%)	30	50	70	90
Percent of Cropping Vertices (%)	25.8	43.4	70.1	91.4
Percent of Cropping Mashes (%)	25.6	43.2	70.5	91.8
Error Ratio (%)	2.9	14.4	31.8	75.8

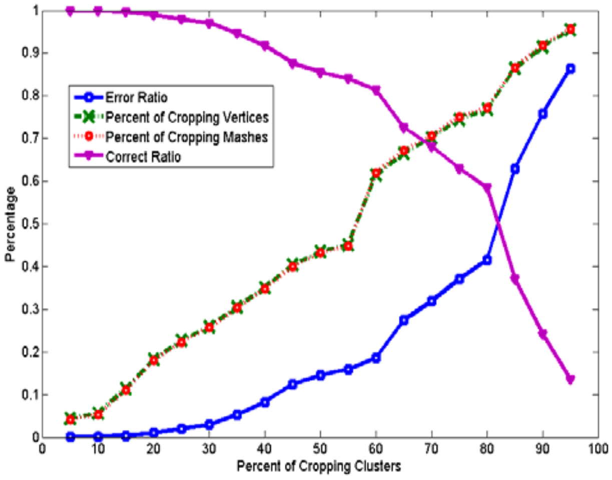
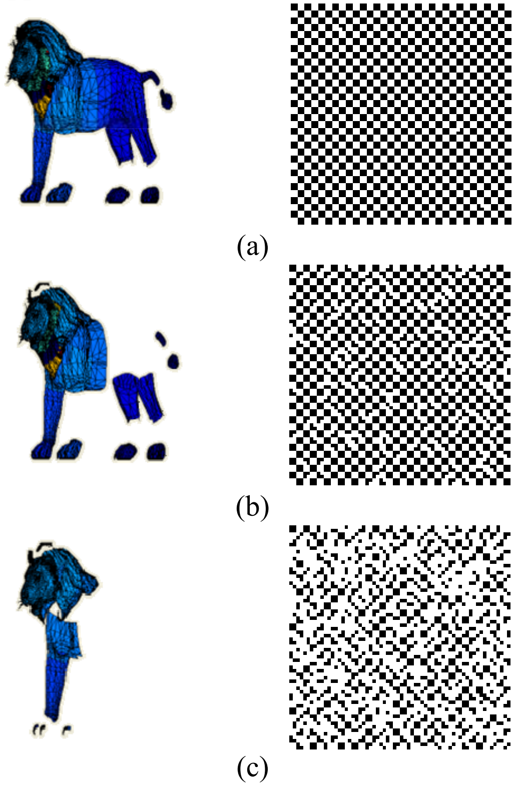


Fig. 18. The percentage of cropped vertices versus the percentage of cropped clusters, the percentage of cropped triangular meshes versus the percentage of cropped clusters, the correct rate of extracted watermark versus the percentage of cropped clusters, and error rate of extracted watermark versus the percentage of cropped clusters.


(a)

(b)

(c)

Fig. 19. The extraction watermarks from the cropping 3D model lion watermarked with a 16×16 chess-board watermark. (a) Cropping 10 Objects; Extracted Watermark with 0.0732 % Error.(b) Cropping 70 Objects; Extracted Watermark with 4.1748 % Error. (c) Cropping 130 Objects; Extracted Watermark with 20.2637 % Error.

5. CONCLUSION

Digital watermarking is a promising method to discourage unauthorized copying or to attest the origin of digital data, including audio, video, images, and 3D models. This paper proposes a cluster algorithm based on vertices connectivity and the 3D model reconstruction with star-topology to construct a robust watermarking scheme for 3D model. In the proposed scheme, the center of each star- topology is selected to be a candidate vertex and several no overlapped edges are selected as the candidate edges for watermark embedding. There are three advantages compare with other methods: (A) The secret image has very high security owing to using chaotic mechanism to scatter the Watermark. (B) The result host 3D model after watermark embedding possesses excellent imperceptibility without noticeable degradation. (C) The watermarks embedded in the 3D models is robust against the similarity transformation attacks and the cropping attack.

REFERENCES

[1] L. M. Marvel, C. G. Boncelet, J.r., and C. T. Retter, "Spread Spectrum Image Steganography," IEEE Trans. on Image Processing, Vol. 8, No. 8, pp. 1075-1083, 1999.

[2] W. Y. Chen and C. C. Liu, "Multiple-Watermarking Scheme of the European Article Number Barcode Using Adaptive Phase Shift Keying Technique", optical Engineering Vol.46, No.6, pp.1- 12, 2007.

[3] J. Fridrich, M. Goljan, and R. Du, "Deteting LSB Steganography in Color and Gray-Scale Images," IEEE trans.on Multimedia, Vol. 8, No.. 4, pp. 22-28, 2001.

[4] R. Chandramouli and N. Memon, "Anlysis of LSB Based Image Steganography Techniques," Proceeding of Image processing, Vol. 3, pp. 1019-1022, 2001.

[5] G. Mastronardi, M. Castellano, and F. Marino, "Steganography Effects in Various Formats o Images," International Workshop on Intelligent Data Acquistion and Advanced Computing System: Technology and Applications, pp.116-119, 1-4 Jul 2001.

[6] C. C. Liu and W. Y. Chen, "Multiple-watermarking scheme for still images using the discrete cosine transform and modified code division multiple- access techniques", Optical Engineering Vol.45, No. 7, 2006, P1- 12.

[7] W. Y. Chen and C. C. Liu, "Robust watermarking scheme for binary images using a slice-based large-cluster algorithm with a Hamming Code", Optical Engineering , Vol.45, No.1, 2006, P1-P10.

[8] W. Y. Chen and C. H. Chen, " A robust watermarking scheme using phase shift keying with the combination of amplitude boost and low amplitude block selection", Pattern Recognition, Vol. 38, 2005, P. 587-598.

[9] H. M. Chao, and Y. S. Tsai, "A Bi-Polar Multiple-Base Data Hiding Technique on Information Security and Authentication," IPPR Conference on Computer Vison, Graphics and Image Processing, pp. 307-313, 2000.

[10] Zhou Zude, Ai Qingsong, Liu Quan, "A SVD-based Digital Watermarking Algorithm for 3D Mesh Models", The 8th International Conference on Signal Processing 2006 Volume: 4 pp. 16-20

[11] Ki-Ryong Kwon, Jae-Sik Sohn, Young Huh, Suk-Hwan Lee , "The Watermarking for 3D CAD Drawing using Line, ARC, 3DFACE Components" , 2006 IEEE International Conference on Multimedia and Expo, 9-12 July 2006 pp:1361-1364

[12] Yang, Shu-Guo Li, Chun-Xia Sun, Sheng-He Xu, Yao-Qun, "A Robust 3D Model Watermarking Scheme Based on Feature Recognition" , Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Volume: 3 pp 989 - 993.

[13] Benedens O, "Geometry-based watermarking of 3D models", IEEE Computer Graphics and Applications, 19 (1) (1999) pp 46-55.

[14] Y.KangKang, P.Zhigeng, S.Jiaoying, Z.David,"Robust mesh watermarking based on multiresolution processing" ， Comput Graphics, Vol. 25, pp. 409 – 420, 2001.

[15] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu, “An efficient k-means clustering algorithm: analysis and implementation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 7, pp. 881-892, 2002.

[16] M. C. Su and C.H. Chou, “A modified version of the K-means algorithm with a distance based on cluster symmetry”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 6, pp. 674-680, 2001.

[17] C. Dematte, N. Molinari, and J. P. Daurès, “Arbitrarily shaped multiple spatial cluster detection for case event data”, Computational Statistics and Data Analysis, Vol. 51, No. 8, pp. 3931-3945, 2007.

[18] A. Mattias, G. Joachim, and L. Christos, “Approximate distance oracles for graphs with dense clusters”, Computational Geometry: Theory and Applications, Vol. 37, No. 3, pp. 142-154, 2007

[19] Y. Zhang , W. Wang, X. Zhang, and Y. Li, “A cluster validity index for fuzzy clustering”, Information Sciences, Vol. 178, No. 4, pp. 1205-1218, 2008

[20] W. Q. Xu, and S. W. Golomb, “Optimal interleaving schemes for correcting two-dimensional cluster errors”, Discrete Applied Mathematics, Vol. 155, No. 10, pp. 1200-1212, 2007

[21] G. Voyatzis and I. Pitas, “Application of Toral automorphisms in Image Watermarking” Proceedings., International Conference on Image Processing, 1996, pp. 237-240.

[22] R. Pajarola, and J. Rossignac, “ Compressed Progressive Meshes”, IEEE Transactions on Visulization and Computer Graphics, Vol. 6, No. 1, pp. 79-93, 2000

[23] J. H. Ahn, C.S. Kim, and Y. S. Ho, “Predictive Compression of Geometry, Color and Normal Data of 3-D Mesh Models”, IEEE Transactions on Circuits and System for Video Technology, Vol. 16, No. 2, pp. 291-299, 2006

[24] S.P. Mudura, S. V. Babjib, and D. Shikhare, “Advancing fan-front: 3D triangle mesh compression using fan based traversal ”, Image and Vision Computing, Vol. 22, pp. 1165–1173, 2004

[25]S. Zafeiriou, A. Tefas, and I. Pitas, “Blind robust watermarking schemes for copyright protection of 3D mesh objects” , IEEE Transactions on Visualization and Computer Graphics, Vol. 11, No 5, pp.596 – 607, 2005.