A Cluster- Based Multiple Watermarking Algorithm for Compound 3D Models

Chen-Chung Liu
National Chin-Yi university of Technology, Department of Electronic Engineering
Associate Professor
ccl@ncut.edu.tw

Chih-Chao Yang
National Chin-Yi university of Technology, Department of Electronic Engineering
Postgraduate
guitar3931@yahoo.com.tw

Jiun-You Chen
National Chin-Yi university of Technology, Department of Electronic Engineering
Undergraduate
gn00139173@yahoo.com.tw

***Abstract*— Multimedia watermarking techniques are used to embed extra information into multimedia contents (e.g. images, audio signals, and 3D models). Frequently, embedding of multiple watermarks in a single multimedia content is required. When a multimedia content is a common work created together by several authors, it might be necessary to embed each co-author's watermark into the common multimedia content. For this purpose, a watermarking technique capable of embedding multiple watermarks for compound 3D model based on clustering is proposed in this paper. In the proposed technique, the original compound 3D model is clustered into several disconnected simple 3D models by means of a vertices connectivity filter. These watermarks are then simultaneously embedded into individual clusters. Imperceptibility of the watermarks was confirmed through the computation of SNR of the watermarks embedded 3D model. Robustness against several kinds of geometric attacks was evaluated by NC, SCC, and BER value of extracted watermarks. The experiment results show that the proposed algorithm can efficiently embed several watermarks simultaneously into compound 3D models, and the embedded watermark is robust against to rotation, translation, and uniformly scaling attacks.**

***Keywords*— multiple, watermarking, compound, 3D model, clustering.**

1. Introduction

Recently, 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.

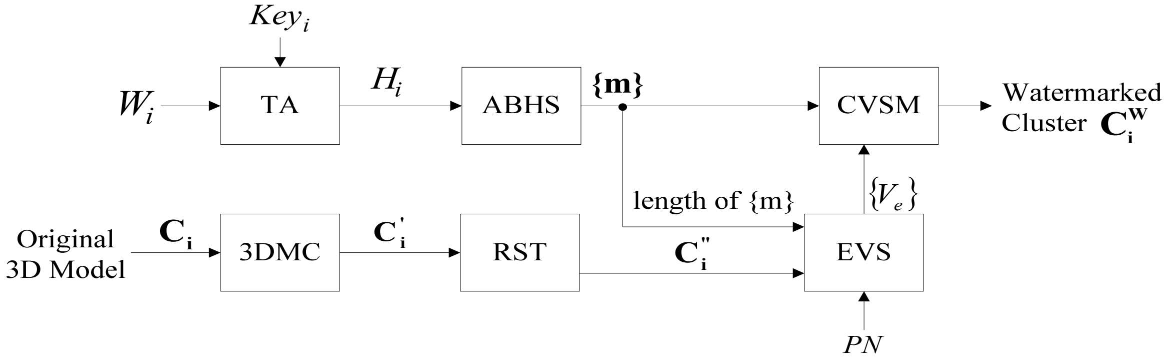
The cryptography technology cut off the access of the unauthorized person after the multimedia information is encrypted [1- 5]. However, it cannot prevent the unlawful action of an authorized person and cannot solve the problem that some copyright owners assert their ownerships for one content. To solve 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 separate 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.

2. Watermarking Embedding Algorithm

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



Wi: Watermark embedded into the ith cluster Ci
TA: Toral Automorphism
ABHS: Arrange the Bits of Hashed Watermark image into a Sequence
CVSM:Embed the Watermark with Center vertex shifting modulation
3DMC: 3D Model Calibration
RST: Reconstruct the 3D Model with Star-Topology Elements
EVS: Embedding Vertices Selection

Fig. 1. The flow chart of watermark embedding for the ith cluster C_i.

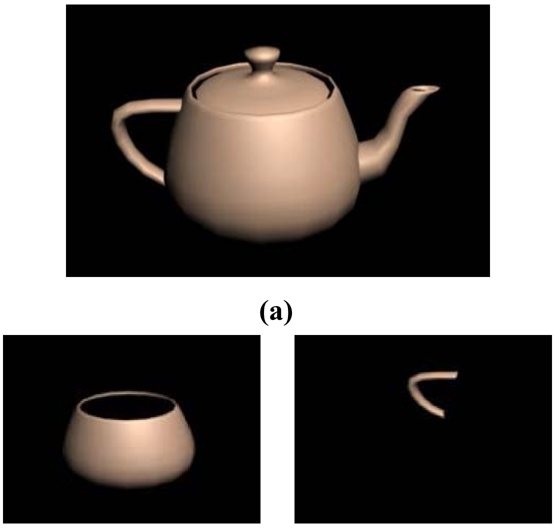
The i- th watermark $\mathrm{W_i}$ is transformed to $\mathrm{H_i}$ by toral automorphism (TA) using a pseudo random sequence (PN) generated by a private key to enhance the security [21]. The hashed watermark $\mathrm{H_i}$ is then arranged to a binary random sequence $\{\mathrm{m_i}\}$. On the other hand, the host 3D model X is divided into several sub-models (clusters) $\mathrm{C_1, C_2, \dots, C_N}$ according the structure of the host 3D model. The starting vertex of each cluster is found and the reference vertex of the 3D model is selected from these starting vertices. Each cluster is reconstructed with star- topology elements and candidate vertices for watermark embedding are collected. In the same time, the 3D model is translated such that the reference vertex is coincide with the original point of the globe coordinate system, and then is rotated such that the reference triangle is coincide with the xy-plane. These candidate vertices are then arranged into a sequence $\{\mathrm{v_c}\}$ according to their weighted spherical coordinates. EVS uses PN sequence and the number of watermark bits to arrange the sequence of the order pairs of watermark bit and watermarking vertex. In the CVSM, the ratio of the distance between the star topology center and the geometry centre to the length of the feature front edge of the 3D mode is adjusted by the embedding watermark bit. The above embedding process is repeated for each secret bit in $\{\mathrm{m_i}\}$ and its corresponding vertex. The details of each block of our watermark embedding algorithm are described in the following subsections.

2.1. Clustering of 3D model's vertices

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 [21].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) [22].

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. 2 shows the result clusters of a 3D model teapot with 4 objects: body, handle, spout, and lid.



(a)

(b) **(c)**

Fig. 2. The result clusters of a 3D model teapot with 4 objects: (a) integrated teapot, (b) cluster body, (c) cluster handle, (d) cluster spout, and (e) cluster lid.

2.2. 3D model Calibration

The watermark embedding of the proposed algorithm is to embed secret bits into collected candidate vertices one by one. There must have an unchanged order among the collected candidate vertices, otherwise the embedded watermark should not be extracted correctly and efficiently. The algorithm uses the translation and rotation of 3D model to calibrate 3D model's pose to guarantee the correction and efficiency of the extracted watermark.

The matrix $R_z(\theta)$ that performs a rotation through the angle θ about the z-axis is given by

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

Similarly, we can derive the following 3×3 matrices $R_x(\theta)$ and $R_y(\theta)$ that perform rotation through an angle θ about the x- and y-axes, respectively.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \qquad (2)$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad (3)$$

The input host 3D model is calibrated in the following steps:

1. Find the reference vertex A that has maximum total neighbor area (summation of areas of triangles that sharing the vertex) among maximum degree vertices in the 3D model, reference edge $\overline{AB}$, and reference triangle ΔABC.

2. Translate the input 3D model M with a translation vector $\vec{T} = \overrightarrow{OA} = \langle x_A, y_A, z_A \rangle$, such that the principal reference vertex A is coincide with the origin of the Cartesian coordinate system.

3. Rotate the translated 3D model about z-axis with angle $2\pi - \alpha_z$ such that the xy- plane projection vector of the principal vector $\overrightarrow{AB} = \langle x_B - x_A,\ y_B - y_A, z_B - z_A \rangle$ coincides with the positive x-axis. Where α_z is the angle between the xy-plane projection vector of the principal vector $\overrightarrow{AB}$ and the positive x-axis, is compute by the following formula.

$$\alpha_z = \tan^{-1}((y_B - y_A)/(x_B - x_A)) \qquad (4)$$

The result principal vector $\overrightarrow{AB}$ after the rotation is

$$(\overrightarrow{AB}')^T = R_z(\alpha_z) *(\overrightarrow{AB})^T = \langle x'_p,\ 0,\ z'_p \rangle^T \quad (5)$$

4. Rotate the 3D model about y- axis with angle α_y such that the principal vector $\overrightarrow{AB}'$ coincides with the positive x-axis, where $\alpha_y = \tan^{-1}(z'_p/x'_p)$ is the angle between $\overrightarrow{AB}'$ and the positive x-axis. The result principal vector $\overrightarrow{AB}''$ after the rotation is $(\overrightarrow{AB}'')^T = R_y(\alpha_y) * (\overrightarrow{AB}')^T = \langle x''_p,\ 0,\ 0 \rangle^T$ and the coordinate of the reference triangle's third vertex C should be ($x''_c,\ y''_c,\ z''_c$), where x''_c, y''_c, and z''_c are calculated from $\langle x''_c,\ y''_c,\ z''_c \rangle^T = R_y(\alpha_y) * R_z(\alpha_z) *(\overrightarrow{AB})^T$.

5. Rotate the 3D model about x- axis with angle α_x such that the reference triangle ΔABC coincides with the positive xy-plane, where $\alpha_x = \tan^{-1}(z''_c/y''_c)$ is the angle between the reference triangle ΔABC and the xy-plane. The result coordinate of the reference triangle's third vertex C is (x'''_c, y'''_c, 0), where x'''_c and y'''_c are not less than zero, and are calculated from the following formula

$$\langle x'''_c,\ y'''_c,\ z'''_c \rangle^T = R_x(\alpha_x) * R_y(\alpha_y) * R_z(\alpha_z) *(\overrightarrow{AB})^T \qquad (6)$$

2.3. Star- topology

A three dimensional model is often represented with its surface that composed by triangles. 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. Note that a 3D model may have multiple boundaries. We call two triangles 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. 3. The number of front-vertices d is the degree of the center of the star- topology [23].

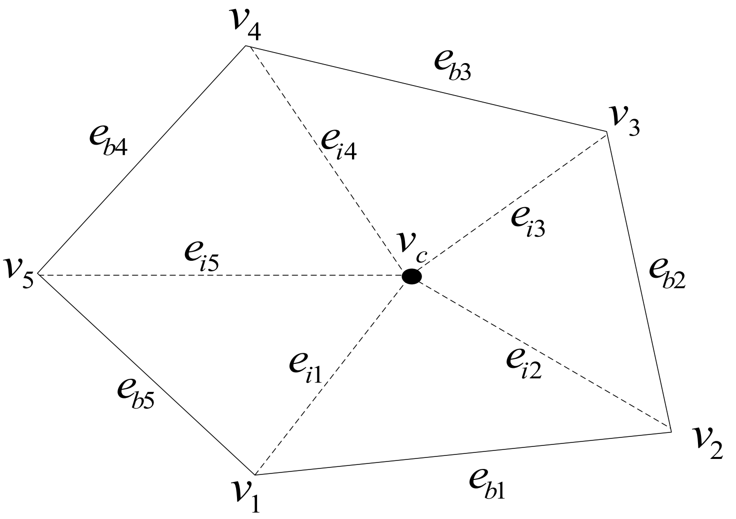


Fig. 3. A star- topology with center vertex v_c, front vertices v_1, $v_2, \dots, v_5$, interior edges e_{i1}, $e_{i2}, \dots, v_{i5}$, and front edges e_{b1}, $e_{b2}, \dots, v_{b5}$.

When we make 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 other vertices before shall be destroyed. In order to overcome this question, we must avoid embedding watermark bits on adjacent vertices. In order to get the goal, we reconstruct the simple 3D model using star- topology as the elemental element to divide a simple 3D model into non- overlapped star- topologies such that any two adjacent star- topologies have several sharing front- vertices. Then, we can embed watermark at the center vertex of each star- topology.

We use a kind of adjacent vertices searching method to reconstruct the triangular mesh 3D model with the star- topology 3D model. The searching method is shown in Fig. 4 and is described as follows:

1. Cluster the vertices of a 3D model into several clusters according to whether the vertices are connected or not. The number of clusters N is one for a simple 3D model, and the number of clusters N is more than one for a compound 3D model that is constructed with several simple 3D models.

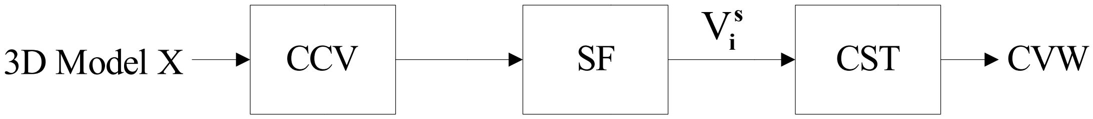
2. Find a feature vertex that has maximum total neighbor area (summation of areas of triangles that sharing the vertex) among maximum degree vertices in the i-th cluster does as the starting vertex v_i^s of the i-th cluster, i = 1, 2, . . ., N. The feature vertex of a cluster is also the center vertex of the first star- topology of this cluster.

3. Connect these front- vertices of that feature vertex to form the first star- topology for the cluster.

4. Construct new star- topologies which are adjacent to the existent star- topology.

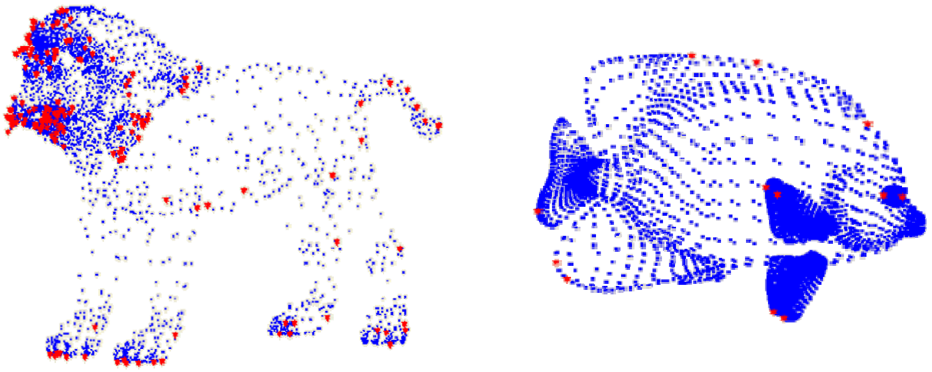
5. Repeat step 4 till each 3D object of the 3D model is completely composed with star- topologies.

6. The length of the shortest front edge l_m of the 3Dmodel is found. And, these centers of star- topology except the reference vertex of a 3D model are collected to be the candidate vertices for watermark embedding. Fig. 5 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. 4. The flow chart of searching method of candidate vertices for watermarking.



(a) **(b)**

Fig. 5. Candidate vertices (points in blue) and starting vertices (points in red): (a) compound 3D model lion with 165 objects; (b) compound 3D model angel_ fish with twelve 12 objects.

2.4. Coordinate transformation

In order to construct a blind robust watermark for a 3D model, the candidate vertices must be sorted into an ordered sequence. The ordered sequence has to be invariant when the watermarked 3D model is attacked by geometry attacks. Otherwise, the watermark embedded in a 3D model shall not be extracted validly. For sorting the candidate vertices into an ordered sequence, the 3D model is taken a translation transform such that the reference vertex $v_r(x_r, y_r, z_r)$ is coincided with the original point O (0, 0, 0) of the coordinate system. The new Cartesian coordinates of vertex $v_j(x'_j, y'_j, z'_j)$ after the translation transform is determined by the following formulas:

$$X'_j = x_j - x_r \quad (7)$$

$$y'_j = y_j - y_r \quad (8)$$

$$Z'_j = z_j - z_r \quad (9)$$

Each candidate vertex's rectangular coordinate is transformed into its corresponding spherical coordinate (r, θ, φ) according to the following formulas:

$$r = \sqrt{(x')^2 + (y')^2 + (z')^2} \quad (10)$$

$$\theta = \begin{cases} \cos^{-1}(z'/r), & z' \geq 0 \\ \cos^{-1}(z'/r) + \pi, & z' < 0 \end{cases} \quad (11)$$

$$\emptyset = \tan^{-1}(y'/x') + \pi/2 \quad (12)$$

The spherical coordinates (r_j, θ_j, φ_j) of the j-th candidate vertex is taken to calculate the weight of the j-th candidate vertex by the following formula:

$$w_j = \text{floor}(r \times 10^6) \times 10^{13} + \text{floor}(\theta \times 10^6) \times 10^7 + \text{floor}(\varphi \times 10^6) \qquad (13)$$

These candidate vertices are sorted into an ordered sequence $\{ \nu_c \}$ according to each candidate vertex's weight.

2.5. Center vertex shifting modulation (CVSM) embedding

The length of the watermark bit sequence and a seed are conducted into the PN sequence generator to generate a PN sequence. The PN sequence selects embedding vertex one by one from the candidate vertices sequence $\{\nu_c\}$. For each selected embedding vertex, the ratio of the distance between the front vertices geometry center and the star topology center to the model's shortest edge is calculated by the following formula.

$$\cdots R_2R_1R_0.R_{-1}R_{-2}R_{-3}\cdots = R = d_{GC}/l_m \qquad (14)$$

Where d_{GC} is the distance between the front vertices geometry center G and the star topology center $\boldsymbol{\nu}_c$, and l_m is the length of the shortest front edge l_m of the 3D model. The numbers R_{-4}, R_{-5} and R_{-6} are respectively converted to radix 2 and their LSB are replaced with the corresponding secret bit, and are finally convert to decimal R'_{-4}, R'_{-5} and R'_{-6}. The ratio is adjusted to $R' = \cdots R_2R_1R_0.R_{-1}\cdots R'_{-4}R'_{-5}R'_{-6}\cdots$. 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 vector $\overrightarrow{G\nu_c}$. The coordinates of ν'_c are determined by the following formulas:

$$x'_c = (1-t) * x_G + t * x_c \qquad (15)$$

$$y'_c = (1-t) * y_G + t * y_c \qquad (16)$$

$$z'_c = (1-t) * z_G + t * z_c \qquad (17)$$

where parameter t is obtained from the formula

$$t = R'/R \qquad (18)$$

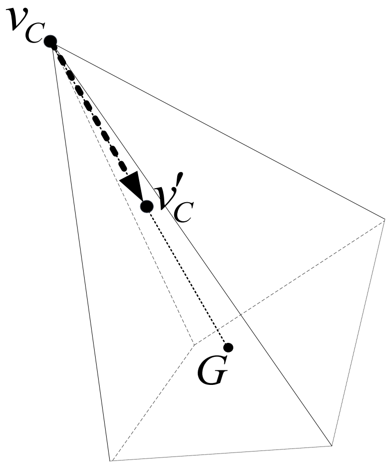


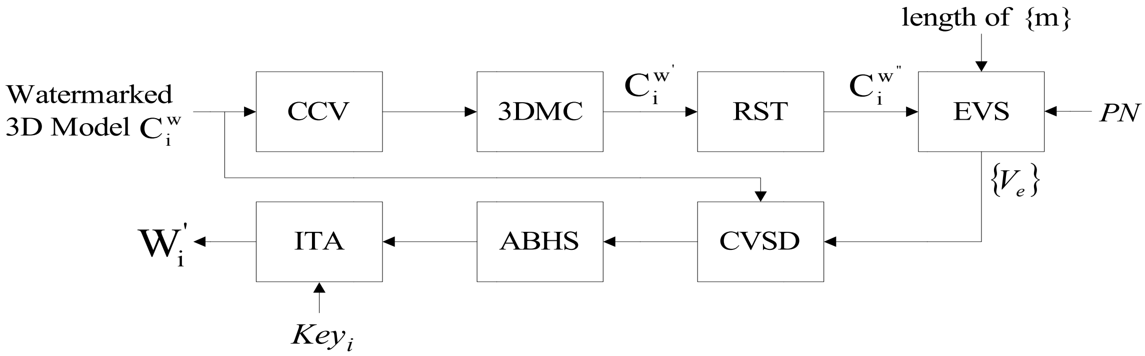
Fig. 6. The shifting of embedding vertex from original position ν_c to new position ν'_c.

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 until it reaches the center vertex shifting modulation process described in the previous section. The flow chart of the watermark recovering process is shown in Fig. 7. The watermarked 3D model X' is divided into several sub-models (clusters) $C'_1, C'_2, \dots, C'_N$ according the structure of X'. The starting vertex of each cluster, the reference vertex, and the reference triangle of X' is found. Then, each cluster is reconstructed with star- topology elements and candidate watermarked vertices are collected. In the same time, X' is calibrated such that the reference vertex is coincide with the original point of the globe coordinate system, and then the coordinates of the candidate embedded vertices are transformed into spherical coordinate. These candidate embedded vertices are then arranged into a sequence $\{\nu'_c\}$ according to their weights calculated by formula (13). The same PN sequence used in the embedding process is used to selected the watermarked vertex ν'_m from $\{\nu'_c\}$ for the CVS demodulation. In the CVS demodulation, the secret bit m' is extracted from the modulated ratio R'' of the selected vertex for each selected star- topology. In the process of the CVS demodulation the modulated ratio R'' is processed to recover the embedded secret bit m' according to the following formula:

$$m' = \begin{cases} 0, & \text{if } N_0 > N_1 \\ 1, & \text{if } N_0 < N_1 \end{cases} \qquad (19)$$

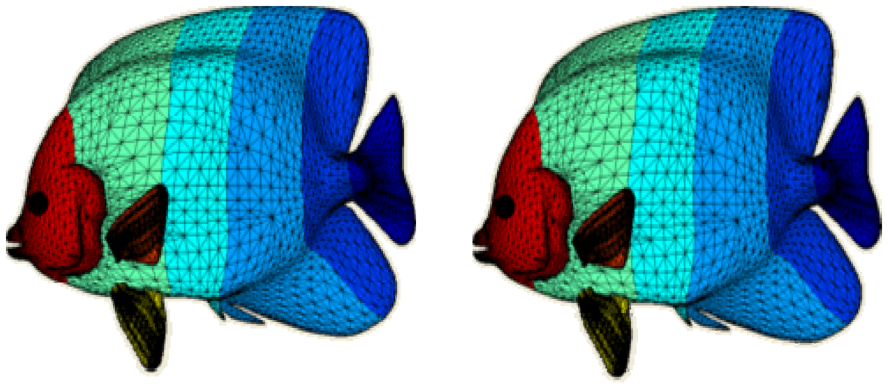
Where $N_0 and\, N_1$ are respectively the number of "0" and all the number of "1" of the LSBs of R''_{-4}, R''_{-5} and R''_{-6} *in* radix 2. After all secret bits are extracted from the CVS demodulation, they are rearranged into the two-dimensional image H'. By passing H' through the inverse toral automorphism (ITA), the recovered watermark W' is obtained.



CCV: Clustering the Model with the Connectivity of Vertices
3DMC: 3D Model Calibration
RST: Reconstruct the 3D Model with Star-Topology Elements
EVS: Embedding Vertices Selection
CVSD: Extract the Watermark with Center vertex shifting demodulation
ABHS: Arrange the Bits of Hashed Watermark image into a Sequence
ITA: Inverse Toral Automorphism
Wi': Watermark Extracted from the ith cluster of the Watermarked 3D Model

Fig. 7. The flow chart of the watermark extraction process.

4. Experiment Results



(a) (b)

Fig. 8. The quality of the host 3D model: (a) the Original 3D model angel_ fish, (b) the Watermarked 3D model angel_ fish (SNR 88.5105s) with 3 watermarks of 16*16 chess board image , logotype images of NCUT and YANG.

Imperceptibility is an important factor in watermarking. We employ the SNR [24] to measure the degree of transparency in this paper. To measure the SNR of a watermarked 3D mesh object, the following formula is used:

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{i=0}^{N-1}(x_i{}^2+y_i{}^2+z_i{}^2)}{\sum_{i=0}^{N-1}((\tilde{x}_i-x_i)^2+(\tilde{y}_i-y_i)^2+(\tilde{z}_i-z_i)^2)}\right) \qquad (20)$$

Where (x_i, y_i, z_i) and $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ are the coordinates of vertex v_i^c before and after the watermark embedding, respectively. We calculate the SNR for embedded angel_ fish. They are arranged in table 1, table 2, table 3, and table 4 to show the watermark's perceptibility. The SNR of the watermarked simple 3D models are almost reciprocal to the sizes of watermarks, are direct proportion to the size of each simple 3D model, and are independent of type of watermark.

Table.1: The performance of our scheme for the test 3D sub models with different size watermarks of logotype images of NCUT.

3D Model	Objects Num.	Mesh	Vertices	Candidate Vertices	Size of Watermark	SNR	Num. of ASKM	Ratio of ASKM
Angel_Fish	12	4168	2086	1041	8×8	92.9590	30	46.88%
					16×16	81.8645	140	54.69%
					24×24	78.7774	332	57.64%
					32×32	77.1924	615	60.06%
	11	4168	2086	1041	8×8	93.1111	39	60.94%
					16×16	81.8321	162	63.28%
					24×24	78.1148	357	61.98%
					32×32	76.7171	636	62.11%
	10	4168	2086	1041	8×8	84.1561	34	53.13%
					16×16	80.1853	141	55.08%
					24×24	77.4560	346	60.07%
					32×32	75.3515	639	62.40%

Table.2: The performance of our scheme for the test 3D sub models with different size watermarks of logotype images of YANG.

3D Model	Objects Num.	Mesh	Vertices	Candidate Vertices	Size of Watermark	SNR	Num. of ASKM	Ratio of ASKM
Angel_Fish	12	4168	2086	1041	8×8	90.3920	39	60.94%
					16×16	81.5188	144	56.25%
					24×24	78.7857	360	62.50%
					32×32	77.1730	596	58.20%
	11	4168	2086	1041	8×8	88.5834	47	73.44%
					16×16	81.9189	167	65.23%
					24×24	78.5262	338	58.68%
					32×32	77.0736	603	58.89%
	10	4168	2086	1041	8×8	83.2315	45	70.31%
					16×16	79.5454	148	57.81%
					24×24	77.0781	346	60.07%
					32×32	75.9006	634	61.91%

Table.3: The performance of our scheme for the test 3D sub models with different size watermarks of chess boards.

3D Model	Objects Num.	Mesh	Vertices	Candidate Vertices	Size of Watermark	SNR	Num. of ASKM	Ratio of ASKM
Angel_Fish	12	4168	2086	1041	8×8	90.7845	40	62.50%
					16×16	82.6679	161	62.89%
					24×24	78.2794	335	58.16%
					32×32	76.9569	606	59.18%
	11	4168	2086	1041	8×8	90.1138	44	68.75%
					16×16	81.7008	167	65.23%
					24×24	77.8006	346	60.07%
					32×32	76.7215	633	61.82%
	10	4168	2086	1041	8×8	85.7872	32	50.00%
					16×16	79.5022	155	60.54%
					24×24	77.4693	336	58.33%
					32×32	75.9006	611	59.67%

Table.4: The performance of our scheme for the test compound 3D model with different watermarks.

3D Model	Mesh	Vertices	Candidate Vertices	Watermarks	Size of Watermarks	SNR
Angel_Fish	38620	19724	7940	NCUT YANG chess board	8×8	96.0546
					16×16	88.5105
					24×24	85.6461
					32×32	84.1136

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, and scaling attacks to test our scheme.

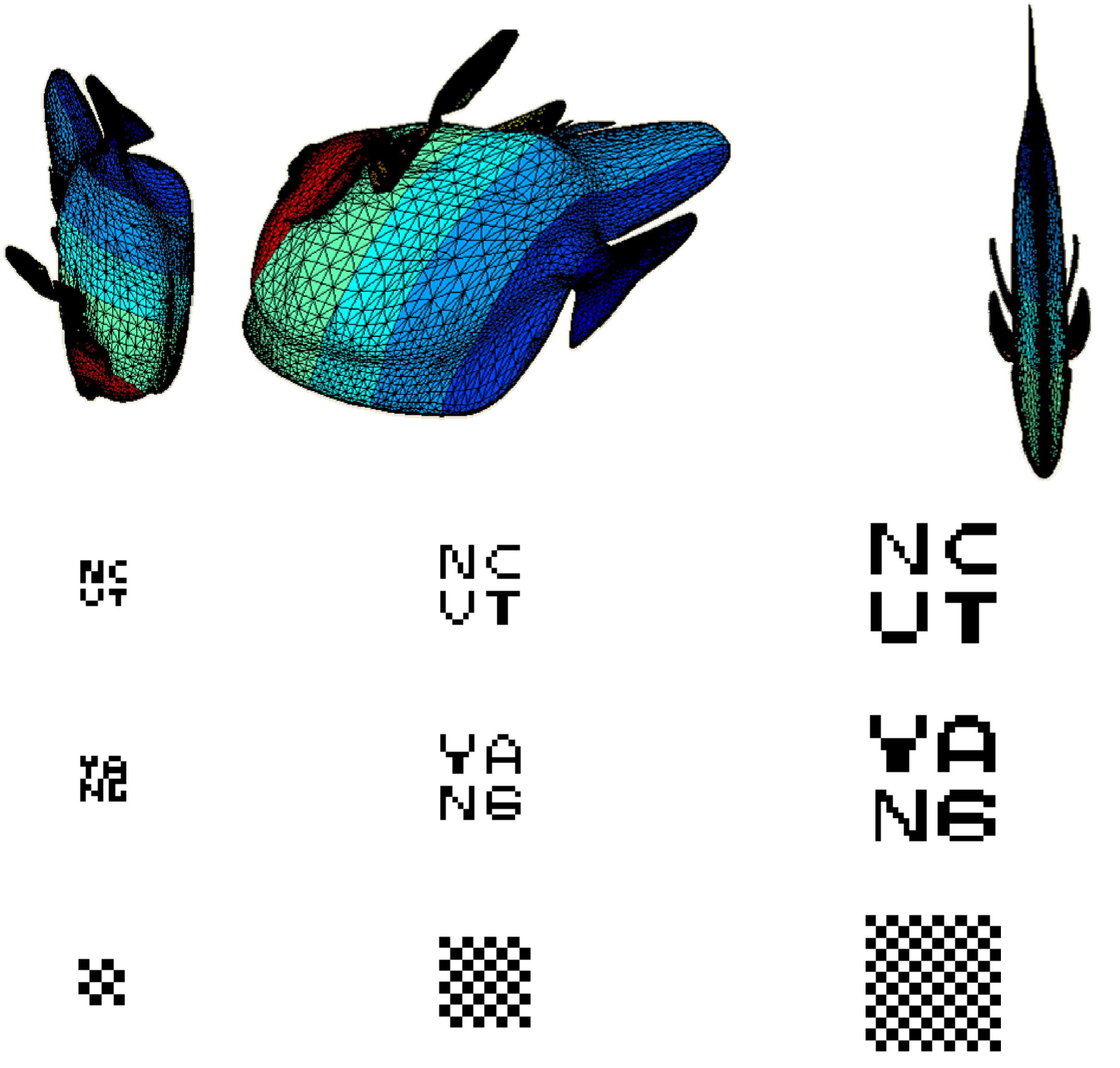
The similarity measurement of a watermark depends on the knowledge of the experts, the experimental conditions, etc. Therefore a quantitative measurement is necessary to provide a fair judgment of the extracted fidelity. In this paper, we use the normalized correlation (NC) and the standard correlation 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:

$$\mathrm{NC} = \frac{\sum_{i}\sum_{j} W(i, j)\, W'(i, j)}{\sum_{i}\sum_{j} \left[W(i, j)\right]^2} \tag{21}$$

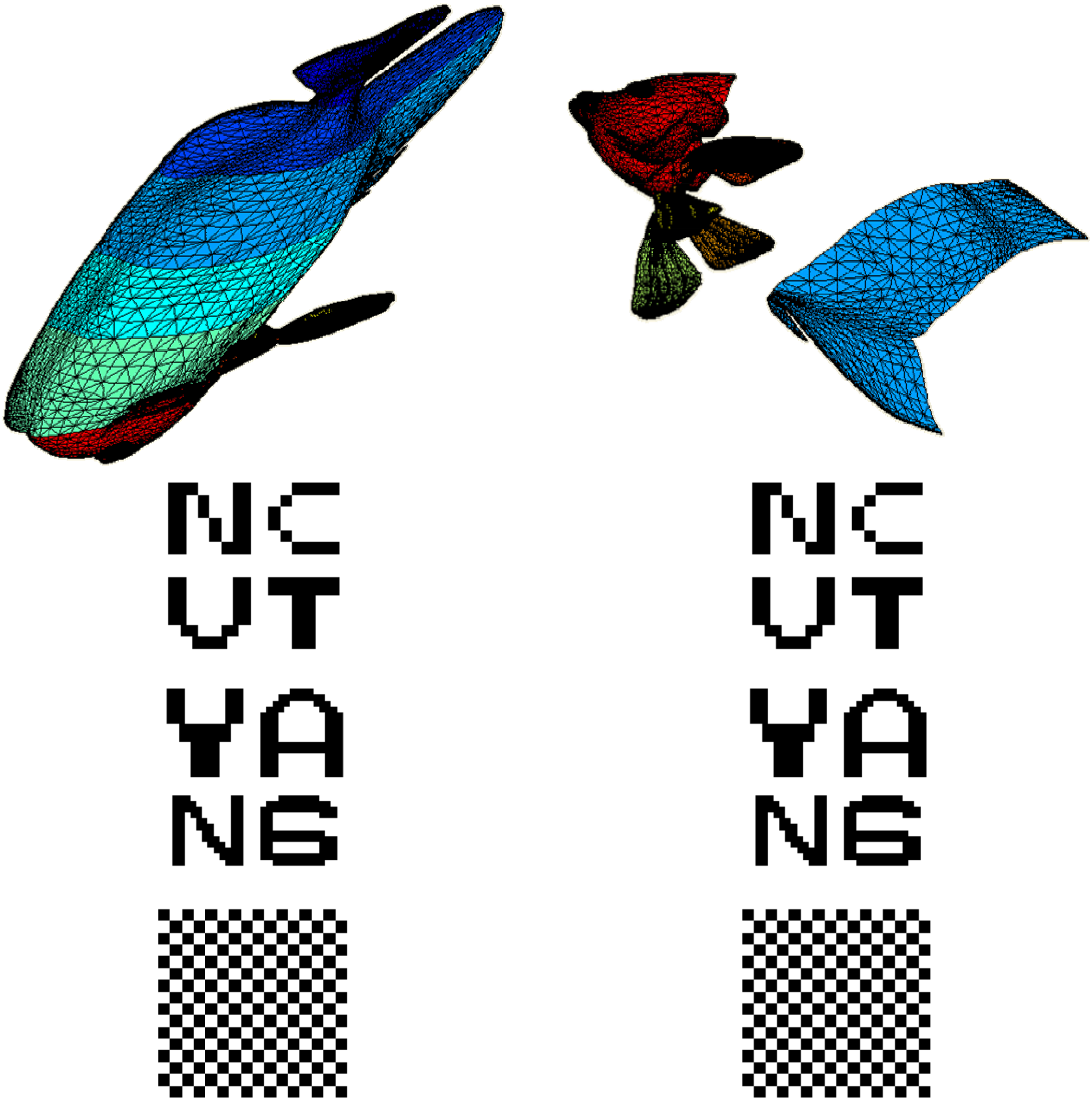
$$\mathrm{SCC} = \frac{\sum_{i}\sum_{j}(W(i,j)-W)(W'(i,j)-W')}{\sqrt{\sum_{i}\sum_{j}(W(i,j)-w)^2}\sqrt{\sum_{i}\sum_{j}(W'(i,j)-W')^2}} \tag{22}$$

$$BER = \frac{Ne}{W_x \times W_y} \tag{23}$$

We calculate the NC, SCC, and BER after attacks for embedded angel_ fish and arranged them in table 5 and figure 9 to show the watermark's fidelity and robustness.



(a) (b) (c)



(d) (e)

Fig. 9. The robust of the extracted watermarks under attacks: (a) S(0.6), (b) R(53°,61°,-27°), (c) T(25, 15, 0), (d) R(-13°, 29°, -85°) and S(1.6), (e) T(35, 12, 23), R(43°, -77°, 39°), S(1.4), and chops.

Table.5 The performance of our scheme for the recovered watermark under attacks.

3D Model	Num. of Objects	Mesh	Vertices	Candidate Vertices	Size of Watermar	Attacks	BER	NC	SCC
Angel_Fish	12	38620	19724	7940	8×8	S(0.6)	0	1	1
					16×16	R(53°,61°,-27°)	0	1	1
					24×24	T(25, 15, 0)	0	1	1

					32×32	R(-13°, 29°,-85°)
						S(1.6)
							0	1	1

					32×32	T(35, 12, 23)
						R(43°, -77°, 39°)
						S(1.4), & chops
							0	1	1

T(x, y, z): Translation with displacement $< x, y, z >$, S: Scaling,
R(θ_x^o, θ_y^o, θ_z^o): Rotation θ_x^o about X − axis, θ_y^o about Y − axis, and θ_z^o about Z − axis.

According to Table 5, the BER are zero, the NC and SCC 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 our approach is a robust enough watermarking scheme for 3D model.

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 multiple watermarks embedding algorithm for compound 3D model based on clustering; the algorithm clusters the 3D model into several clusters bases on the connectivity of 3D model vertices, reconstructs each cluster with star topology elements, and embeds watermark into the center vertex of each selected star topology element with adjusting the coordinates of the embedded center vertex of watermarking star topology element. There are three advantages compare with other methods: (A) The secret image has very high security owing to using toral automorphism to scatter the Watermarks, (B) The result host compound 3D model after watermark embedding possesses excellent imperceptibility without noticeable degradation , and (C) All the extracted watermarks from the watermarked compound 3D models have the same quality (100 % validity even under the combinational attacks of rotations, translations, and scaling).

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

[22] 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.

[23] 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.

[24] 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.