IJCI V9-02(2022) 14–24
International Journal of Computers and Information (IJCI)
Available online at https://ijci.journals.ekb.eg/

A blind fragile watermarking for 3D triangular model based on curvature features and chaos sequence

Noura A.Semary[a],Kariman M.Mabrouk[a], Hatem Abdul-Kader[b]
[a] Information Technology dept., Faculty of Computers and Information, Menoufia University, Menoufia 32511, Egypt
[b] Information system dept., Faculty of Computers and Information, Menoufia University, Menoufia 32511, Egypt
kariman.mamdouh@ci.menofia.edu.eg

Abstract

3D models unauthorized modification is an attractive research problem nowadays due to the widespread availability of technologies and designs on the internet. In this paper, we propose a blind fragile watermarking scheme in the spatial domain for 3D model authentication based on the curvature features of the 3D model and chaos sequence. First, we compute the curvature feature for every vertex of the 3D model, after that, the vertices are classified into 3 classes: flat, peak, and In-between vertices using K-means clustering algorithm. We suggested two methods for embedding the watermark based on the least significant bit (LSB) substitution technique; Cluster Type-based Embedding (CTE) and Cluster Size-based Embedding (CSE). The proposed methods employ a Chaos sequence generator to generate a Chaos sequence that is used to generate the embedded watermark, where the tampering region can be verified and located by the Chaos sequence-based watermark check. Many assessment methods are employed to evaluate the proposed method with various unauthorized attacks like rotation, translation, scaling, cropping, and noise addition. The experiment results show an improvement of embedding imperceptibility as well as tempered regions detection compared to existing literature works.

Keywords: fragile watermarking; content authentication; Chaos sequence; curvature features.

1. Introduction

With the widespread of digital content and with the easy modification by a variety of software editing, watermark techniques had to be developed to protect and authenticate the digital content. The goal of the watermarking is to protect the cover signal by hiding data (watermark) in it. Therefore, watermarking is considered one of the best solutions in data protection that protects digital contents as it maintains the external shape rather than the encryption techniques, which convert the digital data into a non-recognizable form (changes the content itself). According to the application, digital watermarking can be classified into robust or fragile watermarking techniques. The goal of the robust watermarking is to protect the ownership of the digital media, so the embedded watermark should remain detectable after being attacked. while the fragile watermarking aims to authenticate the digital data itself, verify and locate any tampering region [1], so the embedded watermark should be sensitive to any attacks and identifies tamper localization and possibly what the model was before modification. According to the watermark extraction strategy, the fragile watermarking algorithm was classified into public, and private (blind). In the public watermarking there is a need for the original model in the extraction stage, while in in the private watermarking, the original model is not needed in the extraction stage [2]. The general watermark taxonomy is shown in Fig.1 where the selected methods is bounded by rectangle.

Recently, 3D models are widely used by different applications like medical imaging, computer-aided design, video games, and virtual reality applications [3]. The main requirements to provide an effective watermark are Imperceptibility, Robustness, and Capacity. For the Imperceptibility, the watermarked models should look similar to the original model. While the Robustness refers to the keep the watermark resistant to different attacks and manipulation operations. Finally, the Capacity corresponds to the amount of the information that can be embedded in the model. To design a watermarking algorithm, there are a clear trade-off between these requirements. So, the most challenges face the researcher are the model distortion after embedding and the accuracy of tempering detection.

In this paper, we suggest two methods for watermark embedding that provide minimal distortion and high accuracy of tampering detection to a variety of unauthorized attacks. This paper is organized as follows: Section 2 presents an overview of the most fragile watermarking techniques. In section 3, we discuss the proposed method for watermark embedding. Section 4 shows the experimental results and discussions of how our method of embedding

provides minimal distortion and better tampering detection. Finally, the conclusion and future work are provided in Section 5.

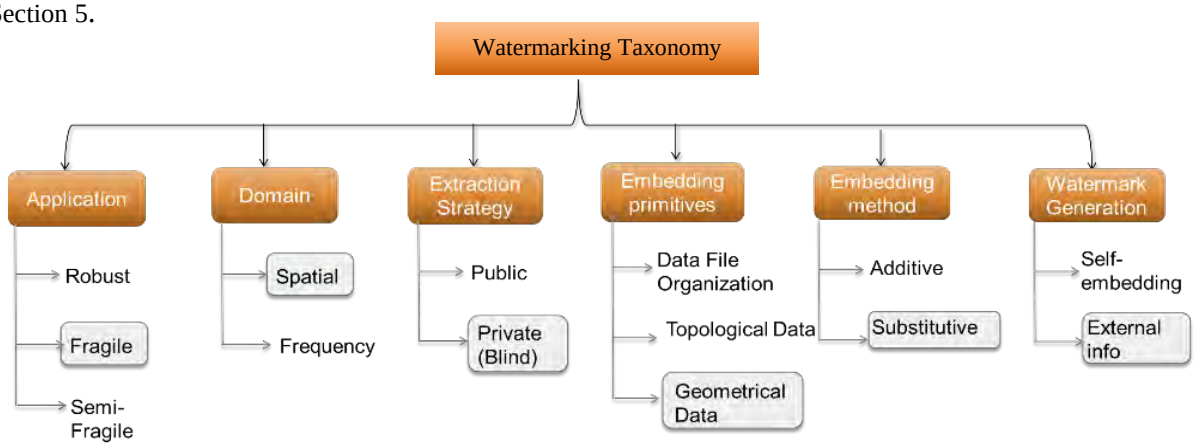

Fig. 1. *watermarking taxonomy*

2. Fragile watermarking techniques overview

Compared to image watermarking techniques, 3D watermarking techniques relatively is less noticed. According to the embedding style, the watermark could be inserted in different embedding primitives like data file organization, topological data, and geometrical data. Where the first type uses the redundancy of polygon models to embed the watermark information, the second type utilizes the topology (connectivity) of the 3D polygonal model to insert the watermark information which affects the triangulation of the 3D model, and the last type modifies the geometry of the 3D polygonal to embed the watermark information as we clarify in our overview paper [1].

The earliest research focused on embedding watermark depending on the curvature features of the model to reach the best method in terms of minimal distortion, high accuracy of tampering detection, and high robustness to different types of attacks. The curvature features are local features that measure the uneven degree of the 3D model. There are three types of curvature for a vertex of a 3D model like root mean square curvature, mean curvature, and Gaussian curvature.

Yeo and Yeung [4] are the pioneers who proposed a public fragile watermarking technique for 3D model authentication. But their method had two problems: the causality problem and the convergence problem. Their method was modified by H. Y. Lin et al. [5], where they achieved localization of any deformation in visual inspection. Ohbuchi et al. [6] [7] [8] have proposed a great variety of 3D polygonal watermarking techniques.

Y. Zhan et al. [9] presented a public 3D watermarking technique based on the curvature of the vertex, as they selected the root mean square to calculate the fluctuation values, then embedded the watermark by modulating the mean normalized fluctuation values. J. Liu et al. [10] proposed a blind watermarking method for 3D point-cloud models. They calculated the root mean square curvature (RMSC) of each vertex. They used the vertices that have large RMSC to embed the watermarking information while the vertices with smaller RMSC were used to synchronize between the watermark embedding and extraction. Without loss of generality, most of the watermarking techniques that utilize the curvature feature as a local feature usually are robust watermarking. On the other side, in this research, we look forward to using the curvature features to present a fragile watermark technique with minimal distortion and best tampering detection accuracy.

With regard to geometrical embedding style, Wang et al. [11] proposed a fragile watermarking algorithm based on a chaotic sequence generator to authenticate 3d models. They embedded the 3 LSB of the chaotic sequence point in each vertex coordinate of the 3D model. Also, they proposed another method [12], where they depended on generating the watermark from the model by using a hamming code. Also, they used the 3LSB of all vertices for embedding. The authors claimed that their methods had minimal distortion to the model and could localize any tampering region. A proposed modification of Wang et al [12] was presented in [13] with a detailed comparative analysis of Wang's methods.

Recently F. Peng [14] reduce the embedding distortion and improve tampering location precision of reversible watermarking for 3D models authentication. They proposed a semi-fragile reversible watermarking based on virtual polygon projection and double modulation strategy. They first obtain a corresponding virtual polygon by constructing the virtual adjacent vertices for each vertex, and then they generate the watermark according to the projection value of the current vertex on the corresponding polygon. Then use the double modulation to move each vertex to realize watermark embedding. The experiment results show that the proposed method provide an improvement of embedding imperceptibility as well as and tampering location precision. from point of view of attack applied to the model, G. Liu [15] proposes a zero-watermarking scheme for 3D models, using the Beamlet transform method to relocate the embedded positions of the watermark when 3D models after rotation attacks, as the vertex data of 3D models has no implicit order. Mourad R. Mouhamed [16] propose a 3D watermarking algorithm based on Coyote Optimization Algorithm (COA) to optimizing statistical watermark embedding for 3D models. they exhibit an intelligent layer on the watermarking process. As they bank on selecting the best vertices to embed the watermarking bits by using the k-means clustering method. In the watermark embedding step they use the COA to determine the best local statistical measure modification value.

Accordingly, we observed that most literary works used all the vertices of the model for embedding equally. We believe that the features of each vertex may affect the quality of embedding. From this point of view, this paper suggests two embedding methods based on the curvature features of the model to provide minimal distortion and accurate tampering detection.

3. Fragile the proposed curvature-based embedding methods

3.1. Overview of the Proposed Method

The proposed embedding method contains four steps, as shown in Fig.2. The first step is the smoothness feature extraction. After that, K-means clustering algorithm is utilized to cluster the model vertices into peak *p*, flat *f* and in-between *in* vertices for selecting the best watermark carrier. The third step is the Chaos sequence (watermark) generation. The final step is the insertion process of the watermark where a Chaos sequence is inserted into different locations in the model. The details of each step will be explored in the following subsections.

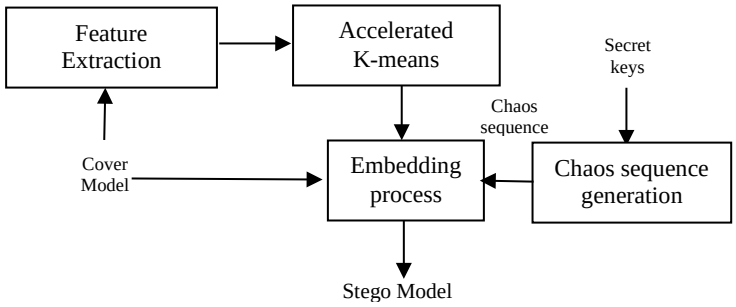

Fig. 2. *General block diagram of embedding procedure*

3.2. Feature Vector Extraction

One of the used features is the shape smoothness as presented in [17]. Where the feature vector contains the angles derived by computing the orientation of the surfaces normal to the average normal of the triangular faces that form a 1-ring neighborhood for a vertex, as shown in Fig.3. In [17], authors have considered only the vertices of valence 6 only in their computations. We modified the algorithm of the vertex smoothness method suggested in [17] by computing the smoothness feature for every vertex in the 3D model instead of considering only the vertex with valence 6. So, the length of the proposed feature vector is equal to the valence of the vertex, which is the number of adjacent faces to the vertex.

The following formulas (1) and (2) are used to determine the feature vectors, where ***M*** refers to the number of adjacent faces to each vertex, ***Ni*** refers to the normal of each face adjacent to each vertex, ***n*** is the number of neighbors to the vertex.

$$N_{avg} = \frac{1}{M} \sum_{1}^{M} N_i \quad (1)$$

$$\alpha_i = cos^{-1}\left(\frac{N_i \cdot N_{avg}}{|N_i||N_{avg}|}\right) \tag{2}$$
$$FeatureVectors = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n]$$

After computing the feature vectors, the accelerated K-means clustering algorithm [18] was performed to determine the topical geometry of the area. The K-means clusters all the mesh vertices into three clusters (flat, peak, and In-between) according to the mean of each vertex feature vector. The vertex is considered a peak, if it is in the highest value cluster center, while the lowest cluster center refers to flat faces.

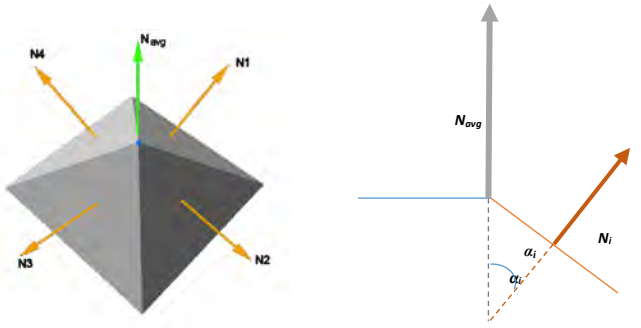

Fig. 3. *Surface normal Ni , average normal Navg for a 1-ring vertex, and the angle αi.*

3.3. *Chaos sequence generation*

Chaotic systems are deterministic systems that are governed by nonlinear dynamics, which recently have been used for digital watermarking to increase security. The most attractive feature of chaos in information hiding is its extreme sensitivity to initial conditions.

In the sense that two chaotic systems generated from different initial conditions, the parameters are uncorrelated statistically and seem random. These special characteristics make chaotic systems excellent candidates for watermarking and encryption [19]. We choose the chaotic sequence generated from the Chen-Lee system [20] according to the formula (3) to represent the embedded watermark. Assume the point in the Chaos sequence is c_i^r, where r refers to the axis used and $0 < i < n$, where the 3 LSBs of chaos sequence point are regarded as the watermark w_i^r.

$$x` = -yz + \alpha x\ ;\ y` = xz + \beta y\ ;\ z` = \frac{1}{3}\, xy + \gamma z \tag{3}$$

3.4. *The embedding process*

The main objective of our proposed system is to choose the best vertices to embed the watermark in with a minimal model distortion. Two methods for embedding the watermark are proposed;

I. Cluster Type-based Embedding (CTE):

In this method, after classifying the vertices of the model into 3 clusters. The standard deviation of the centers of the 3 classes is utilized with K-means algorithm to classify the model into sharp or smooth. According to this classification only one cluster is used for embedding the watermark. The 3 LSB of chaos sequence point are embedded into the 3 LSBs of the vertices of only one cluster as shown in fig.4, to identify any of the clusters that give better results.

The experiments have been performed to evaluate how the clusters type my affect the results. In our experiments, we have tested embedding the message in one cluster at a time. And it was found that for sharp models, the peak vertices give the best visual results. While for smooth models, the flat vertices are the best.

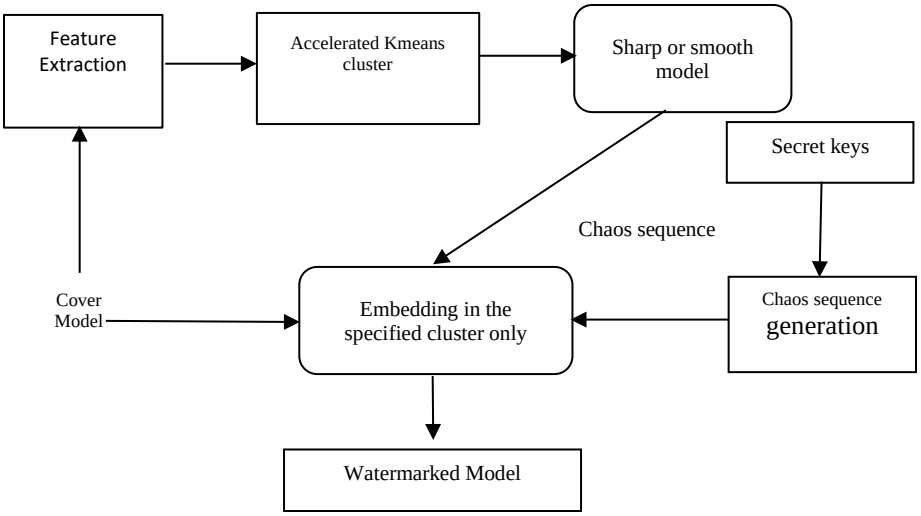

Fig. 4. *Cluster Type-based Embedding (CTE) life cycle.*

II. Cluster Size-based Embedding (CSE):

In the second proposed method Cluster Size-based Embedding (CSE), we embed the watermark in all the clusters with different distribution according to the cluster size. Where the method embeds 1 bit in the cluster with the maximum number of vertices using the LSB substitution, 2 bits in the cluster with the medium number of vertices, and 3 bits in the cluster with the least number of vertices as shown in fig.5. The traversal order of embedding depends on the result of clustering, whereas embedding 1 bit from the chaos sequence depends on the order of vertices in the cluster has a maximum number of vertices and so on for other clusters.

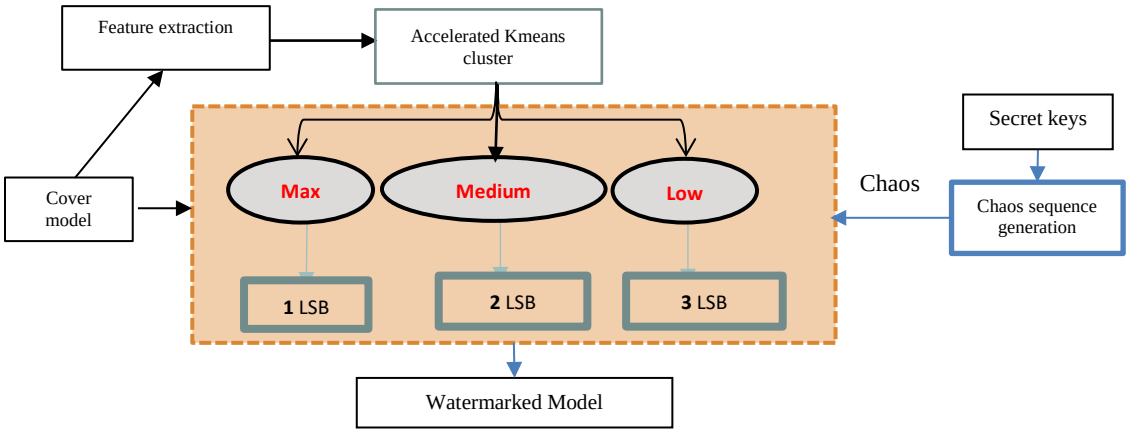

Fig. 5. *Cluster Size-based Embedding (CSE) life cycle.*

3.5. Watermark Extraction

In the watermark extraction stage (Fig.6), the embedded watermark was extracted to verify the integrity of the stego model. The extraction process takes only the stego model with a secret key, which means that the proposed method is blind (doesn't need the original model). It extracts the watermark according to the following five steps:

- Step 1: extract the feature vector according to the same steps in the watermark insertion to be fed into the accelerated K-means clustering method (Step 2) producing 3 clusters (flat, peak, and In-between).
- Step 3: The embedded watermark is extracted from the LSBs of each vertex $\boldsymbol{w_i^{r'}}$ in the stego model according to each cluster.
- Step 4: The Chaos sequence is generated by using the same keys used in the embedding stage. The number of LSBs of point $\boldsymbol{c_i^r}$ is regarded as a watermark $\boldsymbol{w_i^r}$ extracted according to each cluster.

- Step 5: The verification is achieved when the watermark $\boldsymbol{w_i^r}$ (Chaos sequence) is equal to the extracted watermark $\boldsymbol{w_i^{r'}}$.

4. Experimental Result

This section presents the experimental results obtained for both imperceptibility and attacks detection assessment. Five 3D models called "Hemi_Bumpy", "Dodecahedron", "Cow", "Sphere" and "Bunny" [21] [22] are used in our experiments as shown in Fig.7 (a-f). MATLAB R2018a is used for implementing the proposed methods as well as the other comparative systems. We set the initial conditions of the chaotic sequence $[x(0), y(0), z(0)] = [0.2, 0.2, 0.2]$, and the parameters (α, β, γ) are set to $(5, -10, -3.8)$.

Table 1 shows the models information. The models M1-M5 refers to "Hemi_Bumpy", "Dodecahedron", "Cow", "Sphere" and "Bunny" respectively. The table includes information about the number of vertices in the model ***N***, the number of peak $\boldsymbol{N_p}$, flat $\boldsymbol{N_f}$, and in-between vertices $\boldsymbol{N_{in}}$, and the standard deviation (**Std**) of the clusters' centers of each model as well as the type of the model.

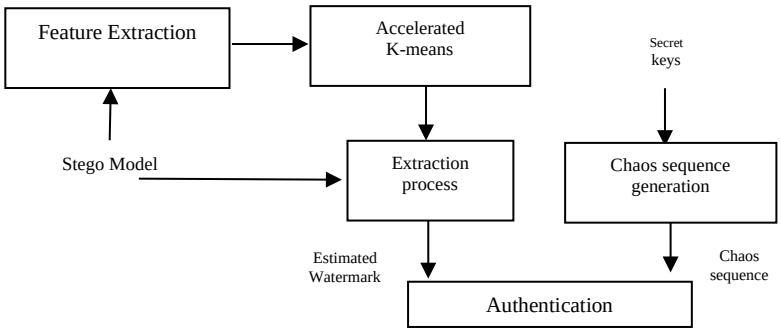

Fig. 6. *General block diagram of extraction procedures.*

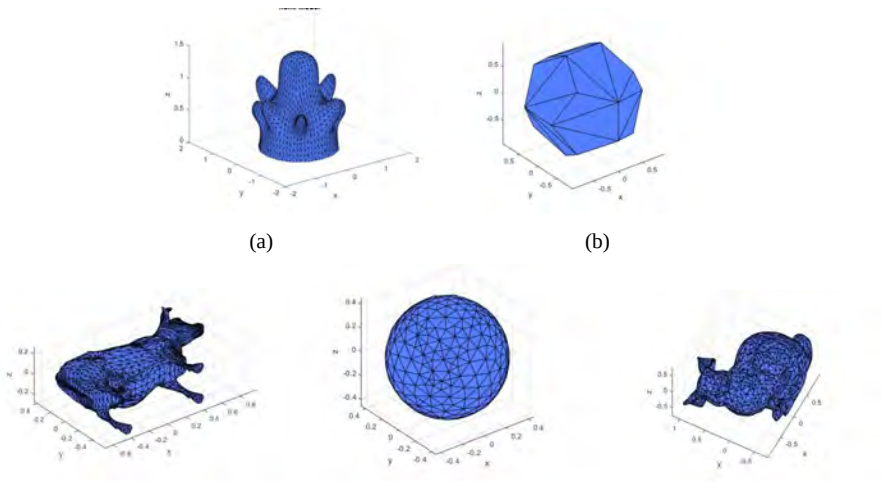

Fig. 7. *Sample of models used for examination*

Table 1: *The experimentation models information and clustering results*

MODEL	N	NP	NIN	NF	STD	CLASS
HEMI_BUMPY (M1)	1441	384	728	329	4.7726759	SMOOTH
DODECAHEDRON (M2)	20	9	9	2	2.5780402	SMOOTH
COW (M3)	2904	274	1069	1561	13.314162	SHARP
SPHERE (M4)	360	131	91	138	1.2944694	SMOOTH
BUNNY (M5)	1355	169	475	711	9.3378793	SHARP

4.1. Imperceptibility Assessment

First, we evaluate the imperceptibility of the proposed methods by using the following metrics: Hausdroff distance (HD), Modified Hausdroff distance (MHD) [1], Root Mean Square Error (RMSE), and the Vertex Signal-to-Noise Ratio (VSNR) [16]. The RMSE measures the differences between the watermarked model and the original model, where the small RMSE values indicate insignificant positional changes during the watermark embedding. Therefore, lower values of RMSE indicates better transparency of the watermark and minimal distortion to the model. The HD measures the similarity of two sets in the metric sense. If the HD is small between two sets, these means that they look almost the same. The MHD computes the forward and reverse distances, and outputs the minimum of both. VSNR determines the perceptual variations between the original and watermarked models.

The values of the assessment metrics after inserting the watermark according to the proposed methods are shown in Tables 2 – 5. The results show the greatest value of VSNR and the lowest value of RMSE, HD, and MHD compared to the methods proposed by Wang et. al [11] and the modified Hamming based [13], where the method of CTE presents better results than the methods of [11, 13]. The tables show also that CSE method gives better results than CTE method. This means that the distortion of the model is affected by the curvature features of the model faces. As well as the proposed CSE method provides better results in terms of imperceptibility. Table 6 show additional VSNR result with recently methods on the mentioned models in the table, where the results show greatest value of VSNR of our method than others, This is due to our proposed method try to find what is the best vertices suitable for watermark embedding to authenticate 3D models and the same time achieve minimal detorsions to the models, Depending on the type of class to which each model belongs, and determining the appropriate method of embedding according to this class.

Table 2: *RMSE evaluation for the proposed methods*

MODEL	CSE	CTE	CHAOS-BASED [11]	MODIFIED HAMMING-BASED [13]
M1	1.43E-16	1.43E-16	3.52E-16	6.54E-16
M2	1.22E-16	1.22E-16	3.36E-16	3.76E-16
M3	6.81E-17	6.81E-17	1.94E-16	3.44E-16
M4	8.68E-17	8.68E-17	1.37E-16	2.40E-16
M5	1.02E-16	1.019E-16	2.62E-16	4.73E-16

Table 3*: VSNR evaluation for the proposed methods*

MODEL	CSE	CTE	CHAOS-BASED [11]	MODIFIED HAMMING-BASED [13]
M1	159.8	163.18	156.71	155.24
M2	158.5	161.76	153.41	156.12
M3	160.6	166.37	156.72	155.08
M4	158	159.21	155.06	154.55
M5	161.3	165.9	157.37	155.83

Table 4 *: Hausdroff dist. evaluation for the proposed methods*

MODEL	CSE	CTE	CHAOS-BASED [11]	MODIFIED HAMMING-BASED [13]
M1	1.5908E-15	1.59E-15	1.6514E-15	3.1563E-15
M2	6.6844E-16	6.68E-16	9.992E-16	1.1538E-15
M3	8.9034E-16	8.90E-16	8.9034E-16	1.5685E-15
M4	5.3820E-16	5.49E-16	5.4958E-16	1.0404E-15
M5	1.5555E-15	1.56E-15	1.5555E-15	1.9375E-15

Table 5 : *Modified Hausdroff dist. evaluation for the proposed methods*

MODEL	CSE	CTE	CHAOS-BASED [11]	MODIFIED HAMMING-BASED [13]
M1	2.40E-16	1.02E-16	5.32E-16	9.85E-16
M2	1.96E-16	6.68E-17	5.22E-16	5.92E-16
M3	1.17E-16	3.17E-17	2.85E-16	5.02E-16
M4	1.04E-16	8.43E-17	2.12E-16	3.67E-16
M5	1.63E-16	1.41E-16	3.94E-16	7.26E-16

Table 6 : *VSNR evaluation of the proposed method with recent methods*

MODEL [23]	NUMBER OF VERTICES	CSE	CTE	MODIFIED HAMMING-BASED [13]	CHAOS-BASED [11]	MOURAD R.[16]
COW	2904	163	159.58	155	156	137
DRAGON	50,000	163	168.27	154.9	157	128
BUNNY	34,835	164	170.91	155.6	157	150
HAND	36,619	164	179.45	155	157	125

4.2. Attack Detection Assessment

For fragile systems evaluation, three assessment measures are usually applied: 1) The precision rate in equation (4) which represents the positive prediction value (PPV) which is calculated as the ratio of the correctly detected true positive parts TP to the predicted positive conditions; True positive TP and false positive FP. 2) The sensitivity rate in equation (5) which represents the true positive rate (TPR) which is calculated as the ratio of the TP to the summation of TP and false negative FN. 3) The accuracy (ACC) in equation (6) of tampering detection that represents the ratio of the total true detected parts (TP and TN) to the total number of samples. We refer to the tampered vertex as a positive instance, therefore the TP parts represent the tampered vertices that are classified as tampered vertices. The FP parts represent the untampered vertices that are classified as tampered vertices. And so, the FN parts represent the tampered vertices that are classified as untampered vertices. The TN parts represent the untampered vertices that are classified as untampered vertices. Generally, Precision and accuracy are used interchangeably, but they have different meanings. Where the measurements close to the known value is considered as accurate, whereas the close measurements to each other are precise.

$$\text{Precision rate (PPV)} = \frac{TP}{TP+FP} \tag{4}$$

$$\textit{Sensitivity rate (TPR)} = \frac{TP}{TP+FN} \tag{5}$$

$$\textit{Accuracy (ACC)} = \frac{TP+TN}{TP+FN+FP+FN} \tag{6}$$

To investigate the sensitivity of the proposed watermarking methods, we performed different types of attacks on the test models. There selected attacks are: 1) Affine transformation; rotation, translation, and scaling. 2) Topology attacks; cropping and noise addition. Since the goal of any fragile watermarking algorithm is to be sensitive to any attacks, we considered very small values modification that are not detected by human vision.

Translation Attack: The translation was performed by two different strategies; equal translation parameters: X=Y=Z=0.5 and by different values: X=0.5, Y=1.1, and Z=0.9. As shown in Fig. 8. The proposed CTE method detects that the vertices have been translated by sensitivity (TRP) from 90% to 99.63%. While the proposed CSE achieves TPR from 85% to 96.11%. Those highly sensitive rates means that there are a few false-negative results, and our proposed methods detects most of the attacked vertices. The PPV of this attack on all the models is 100% for both CTE and CSE, and the accuracy is equal to TPR.

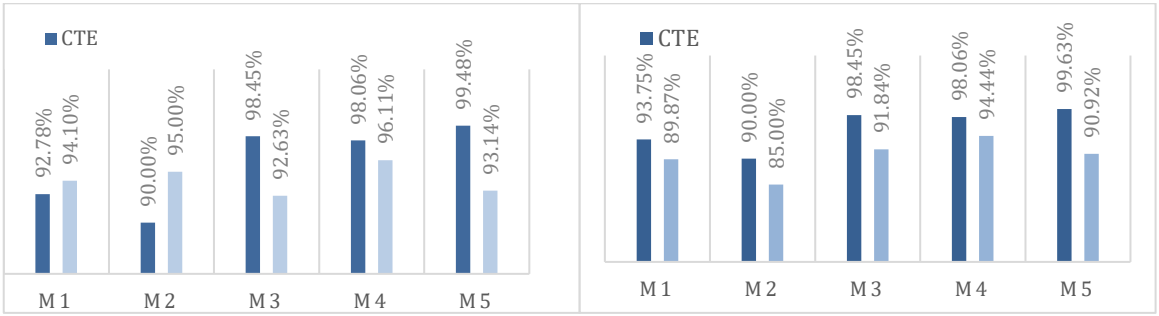

Fig. 8. *Translation detection Sensitivity/Accuracy with values (0.5,0.5,0.5)(left) and with values (0.5,1.1,0.9)(right)*

Scaling Attack: In this experiment, the watermarked model was scaled by two different strategies; equal values: X=Y=Z=1.1 and different parameters: X=1.1, Y=0.9, and Z=1.1. As shown in Fig. 9, the proposed CTE method detects scaling tampered vertices with sensitivity percentage of 90% to 99.63%, while the CSE achieves TPR and accuracy from 60% to 100%. Those highly sensitive rates meaning that there are a few false-negative results, and our proposed method detects most of the attacked vertices. Also, for scaling attack, the PPV is 100% for all cases due to the zero value FP and TN cases which leads also to the equality of TPR and ACC.

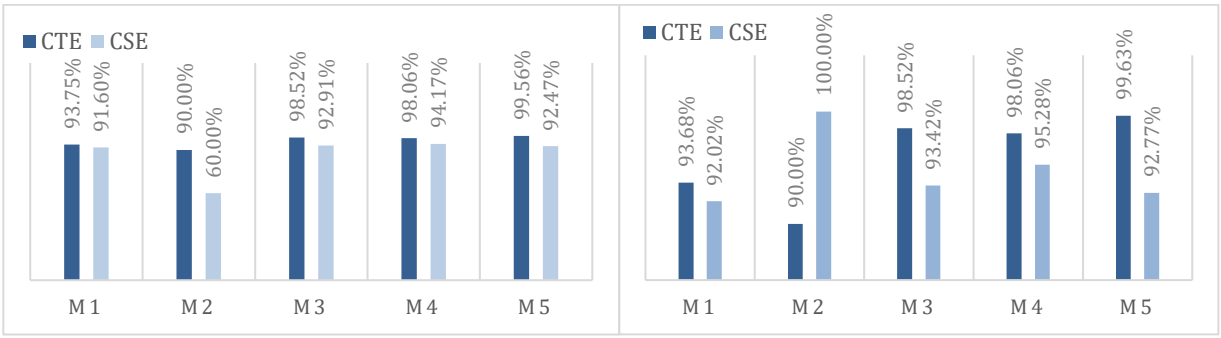

Fig. 9. *Scaling detection Sensitivity/Accuracy results with values of (X=Y=Z=1.1)(left) and with values (1.1,0.9,1.1)(right)*

Rotation Attack: is another affine transform which has been performed by rotating all the vertices of the model. The tempering was also done by two different strategies; equal rotation values (roll, pitch and yaw), X=Y=Z=2° and different values; X=2°, Y=3°, and Z=4°. The results are presented in Fig. 10. As shown in the figures, the proposed CTE method detects that tempered vertices with sensitivity and accuracy of 90 % - 99.56 %, while CSE sensitivity and accuracy from 90% to 95%. Also, as in translation and scaling attacks, the PPV reported 100% for all case due to zero FP and TN.

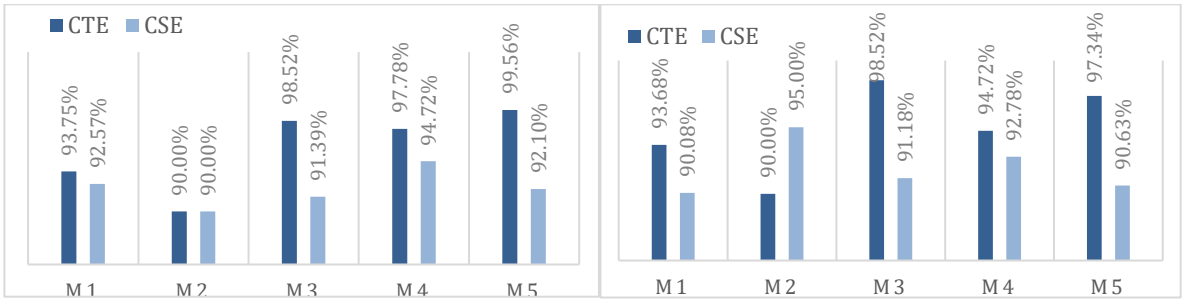

Fig. 10. *Rotation detection Sensitivity/Accuracy with (X,Y,Z = 2°) (left) and with (2, 3, 4)(right)*

Cropping Attack: The cropping attack is the most well-known topological attack that changes the topological features of the mesh shape. Fig.11 shows the visual effect of the detection result for a sample 3D model after deleting one vertex (the vertex of maximum number of adjacent vertices). It was not expected to detect the deleted vertex! But as an effect of our embedding methodology (the watermark is embedded sequentially), a lot of vertices are affected by this cropping, what results in a lot of False positives (FP). Therefore, the deleted vertex results in wrong comparison of all the following vertices in the file where it is compared to a different counterpart, and that is why many vertices have been identified as tempered vertices. Fig. 12 left image shows the TPR of both proposed methods. The TPR ranges from 60% up to 100% with CTE method, while it ranges from 50% to 100% for CSE. According to the existence of FP, the accuracy of detection has different values as presented in Fig. 12 right image. CTE achieves accuracy from 7.5% to 52.63% while CSE achieves from 9.19% to 57.89%. The low accuracy values are expected due to the strategy of sequential embedding.

Noise Addition Attack: Normally distributed random numbers are used to represent the added noise. The vertices to be attacked have been selected randomly. As shown in Fig. 13 left image, the results show a TPR ranges from 59% - 94% for CTE and 92.73% - 94.44% for CSE. Fig. 13 right image presents the accuracy values of noise addition detection. CTE recorded 55.94% - 76.34% while CSE recorded 84.44% - 95.42%. Fig. 14 shows the Noise attack visual results for a sample model.

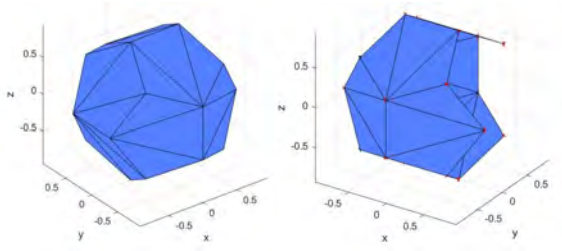

Fig. 11. *A sample model before and after cropping attack*

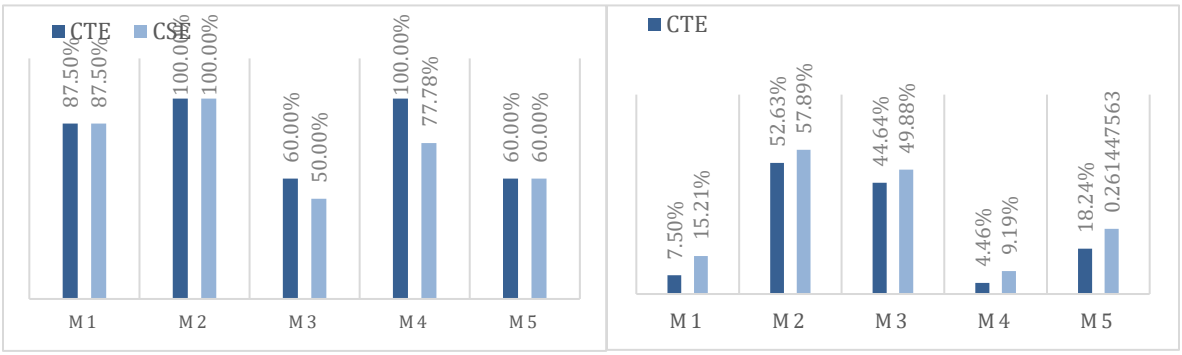

Fig. 12. *One vertix cropping TPR (left), and One vertix cropping Accuracy (right)*

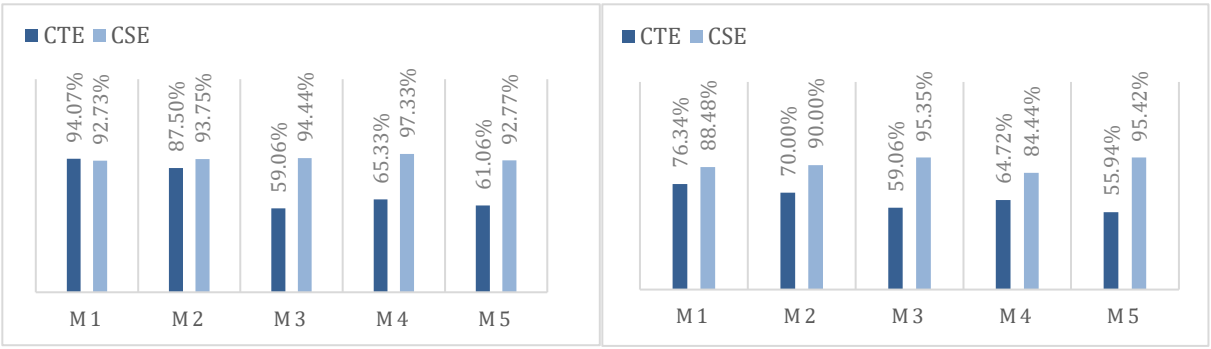

Fig. 13. *Random Noise attack TPR (left), and Random Noise attack Accuracy(right)*

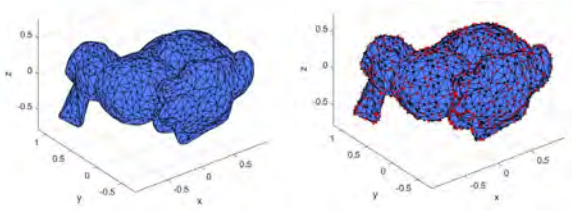

Fig. 14. *The model before and after adding random noise*

5. Conclusion

In this paper, we proposed a scheme for 3D model blind fragile watermarking based on the curvature features of the model. The proposed methods are based on computing the curvature feature to all vertices of the 3D model, then clustering the feature vectors using an accelerated K-means cluster into peak, flat, and in-between vertices. The proposed method employs a Chaos sequence generator to generate a Chaos sequence, which represents the embedded watermark. We proposed two methods for embedding; Cluster Type-based Embedding (CTE) that embedded 3 LSB in only one cluster based on the shape of the model. The curvature features are utilized to classify the objects to sharp or smooth. For smooth shapes, the peak vertices are selected while in smooth models the flat vertices are selected.

The other method, Cluster Size-based Embedding (CSE) embedded 1 bit in the cluster of the maximum number of vertices, 2 bits in the cluster of the medium number of vertices, and 3 bits in the cluster of the minimum number of vertices. The experimental results showed that the imperceptibility evaluation of both proposed methods of embedding is better than the methods of [11], the modified algorithm of [12] as well as the method of M.R. Mouhamed [16]. Attack detection sensitivity assessment were performed. The results showed that the proposed method was sensitive to rotation, translation, scaling, cropping, and noise attack, with a high percentage of tampering detection except for the cropping attack. For future work, we recommend improving the results of cropping detection.

References

[1] K. M. Mabrouk, N. A. Semary and H. Abdul-Kader, "Fragile watermarking techniques for 3d model authentication," in *International Conference on Advanced Machine Learning Technologies and Applications.*, 2019.

[2] C.-M. Chou and D.-C. .. Tseng, "A public fragile watermarking scheme for 3D model authentication," *Computer-Aided Design*, vol. 38, no. 11, pp. 1154-1165, 2006.

[3] K. Tanaka, Y. Nakamura and K. Matsui, "Embedding secret information into a dithered multi-level image," *EEE Conference on Military Communications*, vol. 1, pp. 216-220, 1990.

[4] B.-L. Yeo and M. M. Yeung, "Watermarking 3D objects for verification," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 36-45, 1999.

[5] H. Lin, H. Liao, C. Lu and J. Lin, "Fragile watermarking for authenticating 3-D polygonal meshes," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 997-1006, 2005.

[6] R. Ohbuchi, H. Masuda and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE Journal on selected areas in communications*, vol. 16, no. 4, pp. 551-560, 1998.

[7] R. Ohbuchi, S. Takahashi, T. Miyazawa and A. Mukaiyama, "Watermarking 3D polygonal meshes in the mesh spectral domain," *Graphics interface*, vol. 2001, no. June, pp. 9-17, 2001.

[8] R. Ohbuchi, A. Mukaiyama and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes," *Computer Graphics Forum*, vol. 21, no. 3, pp. 373-382, 2002.

[9] Y. Zhan, Y. Li, X. Wang and Y. Qian, "A blind watermarking algorithm for 3D mesh models based on vertex curvature," *Journal of Zhejiang University SCIENCE C* , vol. 15, no. 5, pp. 351-362, 2014.

[10] J. Liu, Y. Yang, D. Ma, W. He and Y. Wang, "A novel watermarking algorithm for three-dimensional point-cloud models based on vertex curvature," *International Journal of Distributed Sensor Networks*, vol. 15, no. 1, 2019.

[11] J. T. Wang, W. H. Yang, P. C. Wang and Y. T. Chang, "A novel chaos sequence based 3d fragile watermarking scheme," in *In 2014 International Symposium on Computer, Consumer and Control*, Taichung, Taiwan, 2014.

[12] J. T. Wang, Y. C. Chang, S. S. Yu and C. Y. Yu, "Hamming code based watermarking scheme for 3D model verification," in *2014 International Symposium on Computer, Consumer and Control*, Taiwan, 2014.

[13] K. M. Mabrouk, N. A. Semary and H. Abdul-Kader, "Analysis of Substitutive Fragile Watermarking Techniques for 3D Model Authentication," in *International Conference on Advanced Intelligent Systems and Informatics.*, Cairo, 2019.

[14] F. Peng, B. Long and M. Long, "A Semi-fragile Reversible Watermarking for Authenticating 3D Models Based on Virtual Polygon Projection and Double Modulation Strategy," *IEEE Transactions on Multimedia*, 2021.

[15] G. Liu, Q. Wang, L. Wu, R. Pan, B. Wan and Y. Tian, " Zero-watermarking method for resisting rotation attacks in 3D models.," *Neurocomputing*, vol. 421, pp. 39-50, 2021.

[16] M. R. Mouhamed, M. M. Soliman, A. Darwish and A. E. Hassanien, "Watermarking 3D Printing Data Based on Coyote Optimization Algorithm," *In Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*, pp. 603-624, 2012.

[17] M. C. Motwani, "Third generation 3D watermarking:applied computational intelligence techniques," University of Nevada, Reno, 2011.

[18] C. Elkan, "Using the triangle inequality to accelerate k-means," in *Proceedings of the 20th international conference on Machine Learning (ICML-03)*, Washington D.C. , 2003.

[19] C. Zhu and K. Sun., "Chaos Applications in Digital Watermarking," *Applications of Chaos and Nonlinear Dynamics in Science and Engineering*, vol. 2, pp. 187-232, 2012.

[20] H.-K. Chen and C.-I. Lee, "Anti-control of chaos in rigid body motion," *Chaos, Solitons & Fractals* , vol. 21, no. 4, pp. 957-965, 2004.

[21] F. S. U. The Department of Scientific Computing, "A 3D Object Format," 15 8 2016. [Online]. Available: https://people.sc.fsu.edu/~jburkardt/data/obj/obj.html. [Accessed 29 8 2022].

[22] K. Wang, G. Lavoué, F. Denis, A. Baskurt and X. He, "A benchmark for 3D mesh watermarking," in *Shape Modeling International Conference*, 2010.

[23] l. LIRIS, "Mesh Watermarking Benchmark," [Online]. Available: https://projet.liris.cnrs.fr/meshben/. [Accessed 29 8 2022].