

## Research Article

# A Blind Watermarking Model of the 3D Object and the Polygonal Mesh Objects for Securing Copyright

Hanan S. Al-Saadi <sup>1</sup>, A. Ghareeb <sup>2,3</sup> and Ahmed Elhadad <sup>4,5</sup>

<sup>1</sup>Department of Mathematics, Faculty of Applied Sciences, Umm Al-Qura University, Makkah, Saudi Arabia

<sup>2</sup>Department of Mathematics, Faculty of Science, Al-Baha University, Al Bahah, Saudi Arabia

<sup>3</sup>Department of Mathematics, Faculty of Science, South Valley University, Qena, Egypt

<sup>4</sup>Department of Computer Science, Faculty of Computers and Information, South Valley University, Qena, Egypt

<sup>5</sup>Computer Science and Information Department, College of Science and Arts, Jouf University, Sakakah, Saudi Arabia

Correspondence should be addressed to Hanan S. Al-Saadi; [hsssaadi@uqu.edu.sa](mailto:hsssaadi@uqu.edu.sa)

Received 8 October 2021; Accepted 19 October 2021; Published 30 November 2021

Academic Editor: Ahmed Mostafa Khalil

Copyright © 2021 Hanan S. Al-Saadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose a novel model for 3D object watermarking. The proposed method is based on the properties of the discrete cosine transform (DCT) of the 3D object vertices to embed a secret grayscale image three times. The watermarking process takes place by using the vertices coefficients and the encrypted image pixels. Moreover, the extraction process is totally blind based on the reverse steps of the embedding process to recover the secret grayscale image. Various performance aspects of the method are measured and compared between the original 3D object and the watermarked one using Euclidean distance, Manhattan distance, cosine distance, and correlation distance. The obtained results show that the proposed model provides better performances in terms of execution time and invisibility.

## 1. Introduction

Recently, browsing online is an imperative portion of our life so digital data and objects are easy to be duplicated by users to make unauthorized and fake copies of the original work. All kinds of digital data and objects have intellectual-property protection from the owner which is called copyrights. Therefore, copyright is the exclusive legal right of the owner that gives him the right to make copies of creative work, literary, and artistic works. In digital media, the creative work, literary, and artistic works refer to digital objects such as e-books, images, videos, music, databases, and 3D objects. Intensively, the 3D objects are utilized in various purposes such as games, computer graphics, medical imaging, manufacturing, and human models. Thus, the copyright of the 3D object needs more requesting to push research towards developing protection techniques. Watermarking is one of the most important proposed solutions for intellectual-property protection of the 3D object.

For securing copyright, the watermark must be robust against the unauthorized use.

In modern times, watermarking techniques for a wide range of digital media were utilized as a host cover to hide or embed a piece of information message in such a way that it is imperceptible to a human observer. Usually, the digital media covers can take any form such as images [1–7], videos [8–12], audio [13–15], and DNA sequences [16, 17]. Even so the 3D objects are widely available and important, there are a few existing watermarking techniques. The various watermarking techniques for 3D objects can be classified according to the embedding domains such as the spatial domain [18, 19], the spectral-domain [20, 21], and the transform domain [22, 23]. The transform domain techniques such as Fourier, Laplace, cosine, and wavelet transform provide a good trade-off between invisibility and robustness.

In this paper, we propose a novel model for 3D object watermarking. The proposed method is based on the

properties of the discrete cosine transform (DCT) of the 3D object vertices to embed a secret grayscale image three times. Different performance aspects of the method are measured and compared between the original 3D object and the watermarked one. The rest of the paper is organized as follows. Section 2 gives a related work on data hiding and watermarking in 3D objects. Section 3 describes the watermarking model of the proposed technique. Experimental results are presented and analyzed in Section 4, where a performance comparison was held between the original 3D object and the watermarked object. Finally, Section 5 summarizes the findings and conclusions.

## 2. Related Work

In [24], Ai et al. proposed a 3D triangular mesh models watermarking scheme. The proposed method was based on selecting the significant geometry information of triangular meshes. Then, the watermark bits are repeatedly embedded into the corresponding Voronoi patch using the high frequency coefficients of the discrete cosine transform. The advantages of the proposed method are resistant to common attacks and provide good imperceptibility. Based on the graph Fourier transform in [22], Ferreira et al. proposed a nonblind 3D point cloud watermarking method. The proposed method assumes the points of connectivity information to represent the cloud as a graph. Then, the watermark bits are embedded in the spectral graph of the Fourier domain. Similarly, in [25], the authors presented a robust 3D point cloud models watermarking in spatial domain. In [26], Zhang et al. presented a 3D geological model watermarking algorithm based the principal component analysis of the point cloud model of geological body.

In [27], El Zein et al. presented two 3D watermarking schemes based on fuzzy *c*-means clustering technique in the spatial domain. The proposed methods start by selecting the appropriate vertices using the fuzzy *c*-means clustering using the feature vector. Then, the watermark bits are embedded into the selected vertices using two insertion methods based on statistical measurements. The first method achieved higher results in robustness while the second method achieved higher results in imperceptibility. In the same context in [23], Liu et al. proposed the blind and robust 3D object watermarking method based on the multi-resolution adaptive parameterization of surface approach for vertices classification. The results of the proposed method showed that it has good imperceptibility and it resists to common attacks. Moreover in [28, 29], Medimegh et al. presented a statistical 3D object watermarking method based on extracting the salient feature points using autodiffusion function. The proposed method segments the 3D object into regions according to the salient points, and then it inserts the watermark bits using the embedding method in [30].

In [31], Molaei et al. proposed a fragile and blind watermarking method based on the geometrical properties of the 3D object. The proposed method selects a specific triangular and embeds the watermark bit using the medians of the triangular faces in spherical coordinates and reconstructs a new triangle. Based on the geometric property of

vertices in [32], Hansda et al. proposed a nonblind method to watermark the 3D object using the mean curvature of vertices. The method divided the vertices into two groups: the first group includes the convex curvature vertices and the second group is the rest of vertices. Then, the watermark bits are embedded by modifying the vertices using the first group. Instead of using the geometric distortions, in [33], Son et al. presented a blind 3D object watermarking method based on mesh saliency. The proposed method uses the distribution of the vertex norm histogram and combines between the spatial domain embedding and the frequency-based weight map.

Based on the above, the main contributions of this paper are as follows: (1) we introduce a 3D object watermarking scheme that takes advantage properties of the discrete cosine transform (DCT) to hide a grayscale image into the 3D object vertices; (2) we propose a blind extraction based on the reverse steps of the embedding process to recover the secret grayscale image; (3) we brought evidence that the proposed watermarking scheme performed across the different 3D objects ensures a minimum shape distortion; and (4) we present comprehensive experimentation examining the performance of our method and comparing it with other methods.

## 3. The Proposed Model

The 3D object watermarking means adding integrated hidden information into polygonal mesh object without leaving visual marks or causing any structural changes. It is typically used to identify ownership rights of such that original 3D object and protect it from theft. In this inevitable scenario, in this paper, there is a proposed method for watermarking the 3D object, which takes advantage properties of the discrete cosine transform (DCT) to hide a grayscale image into the 3D object vertices.

Figure 1 depicts the overall process of the proposed watermarking model. Firstly, the proposed method normalizes both the original 3D object vertices and the secret grayscale image. This step allows us to apply the appropriate fuzzy algorithms for modifying the membership values. In general, there are different approaches to fuzzification and many applications such as [34, 35]. Then, a preprocessing is applied on the normalized 3D object vertices and applies a discrete cosine transform (DCT) for every three vertices. In the same time, reshape and encryption processes are carried out on the normalized watermark image. After that, the watermarking process will take place by using the vertices coefficients and the encrypted image pixels. Finally, the inverse DCT and denormalization process of the modified vertices will be applied to produce the watermarked 3D object.

*3.1. The Embedding Procedure.* As mentioned above, the process of embedding the watermark starts by normalizing both the original 3D object vertices and the secret grayscale image. Generally, normalization is a typical process that expands the range of data values dynamically. Therefore,

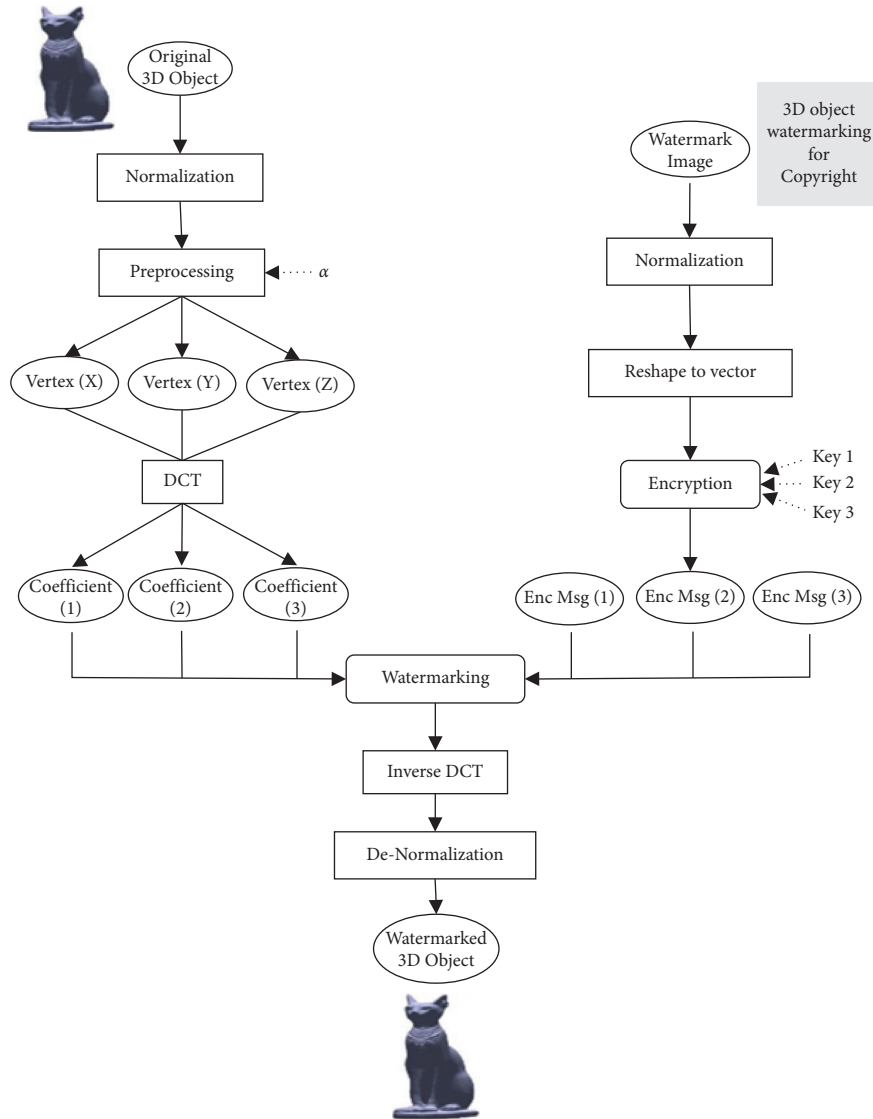


FIGURE 1: The overall model for the proposed watermarking method.

normalization changes the data object  $\text{Obj}: \{X \subseteq \mathbb{R}^d\} \rightarrow \{\text{Min}, \dots, \text{Max}\}$  from the range values  $[\text{Min}, \text{Max}]$  into a new data object  $\text{Obj}_{\text{New}}: \{X \subseteq \mathbb{R}^d\} \rightarrow \{\text{Min}_{\text{New}}, \dots, \text{Max}_{\text{New}}\}$  with the range values  $[\text{Min}_{\text{New}}, \text{Max}_{\text{New}}]$ . The linear normalization of a data object is performed according to the following formula:

$$\text{Obj}_{\text{New}} = (\text{Obj} - \text{Min}) \frac{\text{Max}_{\text{New}} - \text{Min}_{\text{New}}}{\text{Max} - \text{Min}} + \text{Min}_{\text{New}}. \quad (1)$$

In the proposed method, the normalization process refers to change both the original 3D object vertices and the secret grayscale image data range with intensity values in the range of  $[0, 1]$ . So, the normalization is achieved according to the following formula:

$$\text{Obj}_{\text{New}} = \frac{(\text{Obj} - \text{Min})}{\text{Max} - \text{Min}}. \quad (2)$$

For security issue, the secret grayscale image is encrypted using the seed numbers of a pseudorandom generator, and this scrambles the position of every pixel in the original grayscale image. The encryption process is applied to the reshaping to vector of the grayscale image to increase the complexity of the scrambling. In addition, it changes the position of the pixels using three secret keys to provide better robustness in this system.

In the preprocessing phase, the normalized vertices of the original 3D object are adjusted using  $\alpha$ : ( $\alpha \in \mathbb{R}, 0 < \alpha \ll 1$ ). This phase assures that saturated vertex value would not eventually result in an overflow in the embedded vertex coefficient. So, the preprocessing phase takes place according to the following formula:

$$3D \text{ Obj}(\text{Vertex}) = \begin{cases} \alpha, & \text{Vertex} = 0, \\ 1 - \alpha, & \text{Vertex} = 1. \end{cases} \quad (3)$$

The discrete cosine transform (DCT) is derived from the Fourier-related transformation [36], and Ahmed et al. firstly proposed the discrete cosine transform (DCT) in [37]. DCT transforms a sequence of real data points into its real spectrum to avoid the problem of redundancy. Thus, the DCT is the process of decomposing a finite sequence of digital signal data points in terms of a sum of cosine functions oscillating at different frequencies to be the equivalent of the original digital signal. Formally, the DCT is an invertible function  $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ , so for a signal  $x$  of length  $N$  and with  $\delta$  the Kronecker delta, the DCT has four standard variants according to one of the following formulas.

The type DCT-1:

$$y(k) = \sqrt{\frac{2}{N-1}} \sum_{n=1}^N x(n) \frac{1}{\sqrt{1+\delta_{n1}+\delta_{nN}}} \frac{1}{\sqrt{1+\delta_{k1}+\delta_{kN}}} \cdot \cos\left(\frac{\pi}{2N}(n-1)(k-1)\right). \quad (4)$$

The type DCT-2:

$$y(k) = \sqrt{\frac{2}{N}} \sum_{n=1}^N x(n) \frac{1}{\sqrt{1+\delta_{k1}}} \cos\left(\frac{\pi}{2N}(2n-1)(k-1)\right). \quad (5)$$

The type DCT-3:

$$y(k) = \sqrt{\frac{2}{N}} \sum_{n=1}^N x(n) \frac{1}{\sqrt{1+\delta_{n1}}} \cos\left(\frac{\pi}{2N}(n-1)(2k-1)\right). \quad (6)$$

The type DCT-4:

$$y(k) = \sqrt{\frac{2}{N}} \sum_{n=1}^N x(n) \cos\left(\frac{\pi}{4N}(2n-1)(2k-1)\right), \quad (7)$$

where

$$k = 0, 1, \dots, N, \quad \delta_{ij} = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases} \quad (8)$$

The proposed method supposes that the 3D object has  $L$  set of vertices, and each vertex is defined as Vertex( $X, Y, Z$ ). Thus, the next phase is applying the DCT transform for each vertex as a vector using the following formula:

$$f(\text{Vertex}(X_l, Y_l, Z_l)) \rightarrow (\text{Coefficient}_{1l}, \text{Coefficient}_{2l}, \text{Coefficient}_{3l}), \quad (9) \quad \forall (\text{Vertex}(X_l, Y_l, Z_l)), \quad l = 1, 2, \dots, L.$$

Once the DCT is applied on the normalized 3D object vertices, the next step carries out the watermarking process on the normalized coefficients of the vertices. Therefore, the first coefficient value will be within the interval  $[0, 2]$ , and the values of the second and the third coefficient will be within the interval  $[-1, 1]$ . Using these facts, we construct the following equation system for watermarking the secret grayscale image in the transformed regions of the 3D object coefficients:

$$\begin{aligned} 3D \text{ obj}(\widehat{C}) &= \frac{2}{\beta} (\text{EncMsg} + i), \\ \frac{2i}{\beta} &\leq 3D \text{ obj}(C) < \frac{2(i+1)}{\beta}, \\ i &= \begin{cases} 0, 1, 2, 3, \dots, (\beta-1), & C = \{\text{Coefficient}_1\}, \\ -2, -1, 0, 1, \dots, (\beta-3), & C = \{\text{Coefficient}_2, \text{Coefficient}_3\}, \end{cases} \end{aligned} \quad (10)$$

where in these equation system,  $3D \text{ obj}(\widehat{C})$  refers to the current coefficient in the watermarked 3D object vertices,  $3D \text{ obj}(C)$  is the corresponding coefficient in the 3D object vertices,  $\text{EncMsg}$  is the embedded pixel in the secret grayscale image, and  $\beta$  is the number of intervals which satisfy that the coefficient on the interval of  $[0, 2]$  or  $[-1, 1]$  corresponding to the coefficients  $C$ . Finally, the inverse of the DCT transform and denormalization process are used to reconstruct the watermarked 3D object with the secret embedded grayscale image. Full details of the embedding

process are shown with the mathematical relationships contained as pseudocode in Algorithm 1.

**3.2. The Extraction Procedure.** In the extraction process, the steps carried out in the embedding process are generally reversed to recover the secret grayscale image. The steps of the extraction module are illustrated in Figure 2. Thus, the process starts by normalizing the watermarked 3D object and then computing the DCT transform decomposition of

```

(1) Function [W_3D_Obj] = Embedding (Obj, Msg,  $\alpha$ ,  $\beta$ , Key1, Key2, Key3)
(2) //Obj : Original 3D object vertices, Msg : Secret grayscale image, Key1, Key2 and Key3: Encryption keys
(3) //Control parameters:  $\alpha \in [0, 0.5]$ ,  $\beta \in [1000, 9000]$ 
(4) Normalized Obj = Normalize(Obj)
(5) [Vertex x, Vertex y, Vertex z] = Preprocessing(Normalized Obj,  $\alpha$ )
(6) Normalized Msg = Normalize(Msg)
(7) Msg vector = Reshape to vector(Normalized Msg)
(8) [Enc Msg1] = Encrypt(Msg vector, Key1)
(9) [Enc Msg2] = Encrypt(Msg vector, Key2)
(10) [Enc Msg3] = Encrypt(Msg vector, Key3)
(11) for j = 1 to length(Msg vector) do
(12) [Coefficient1, Coefficient2, Coefficient3] = DC T (Vertex x (j), Vertex y (j), Vertex z (j))
(12) Embed the Enc Msg value within the Coefficients values as the following equation system:
(13)  $\hat{C} = (2/\beta)(\text{EncMsg}(j) + i)$ ,  $(2i/\beta) \leq C < (2(i+1)/\beta)$ ,  $i = \begin{cases} 0, 1, 2, 3, \dots, (\beta-1) & C = \{\text{Coefficient}_1\} \\ -2, -1, 0, 1, \dots, (\beta-3) & C = \{\text{Coefficient}_2, \text{Coefficient}_3\} \end{cases}$ 
(14) [Vertex x' (j), Vertex y' (j), Vertex z' (j)] = Inverse DC T (Coefficient'1, Coefficient'2, Coefficient'3)
(15) end
(16) W_3D_object = DeNormalize(Vertex x', Vertex y', Vertex z')
(17) return Watermarked 3D object W_3D_Obj
(18) end

```

ALGORITHM 1: Embedding function in the proposed method

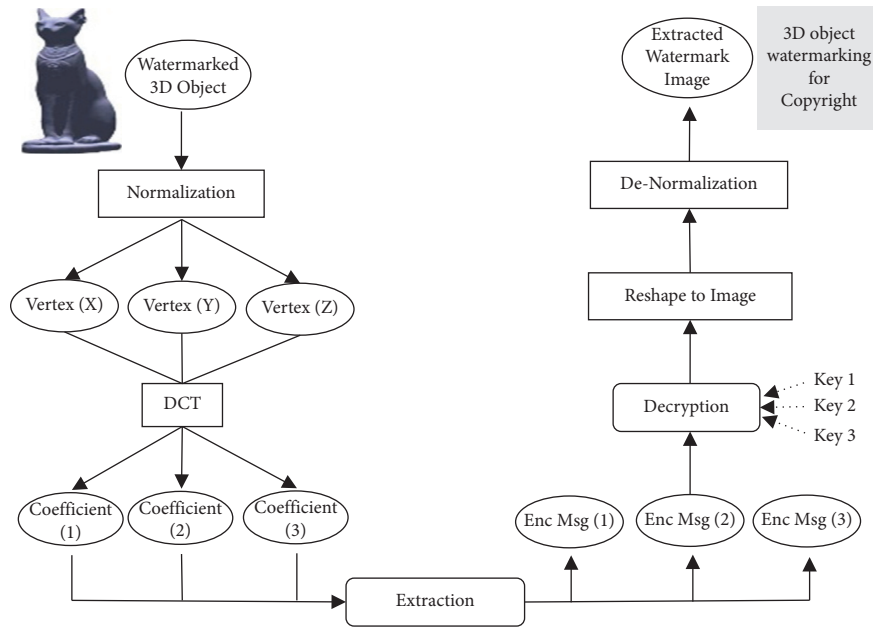


FIGURE 2: The extraction flowchart of the proposed model.

the vertices. So, the encrypted grayscale image pixel can be extracted using the parameters  $\beta$  and the 3D obj( $\hat{C}$ ) coefficients according to the following equation:

$$\begin{aligned} \text{EncMsg} &= \frac{\beta}{2} \left( 3D \text{ obj}(\hat{C}) - \frac{2i}{\beta} \right), \\ \frac{2i}{\beta} &\leq 3D \text{ obj}(\hat{C}) \leq \frac{2(i+1)}{\beta}, \\ i &= \begin{cases} 0, 1, 2, 3, \dots, (\beta-1), & \hat{C} = \{\text{Coefficient}_1\}, \\ -2, -1, 0, 1, \dots, (\beta-3), & \hat{C} = \{\text{Coefficient}_2, \text{Coefficient}_3\}. \end{cases} \end{aligned} \quad (11)$$

In blind manner and using  $\beta$ , the secret EncMsg will be extracted from the watermarked 3D object. Next, the three keys are required to identify the position at which the secret original pixels were located. Since the pixel values were normalized, they need to be denormalized to convert the pixel values back to their original integer domain. Notice that the secret grayscale image is extracted correctly three times. The full details of the extraction part are displayed as pseudocode in Algorithm 2.

## 4. Experimental Results and Discussion

**4.1. Implementation.** Throughout the following sets of experiments, Figures 3(a) to 3(f) show six standard 3D objects which were used for testing the proposed method performance. In addition, Figure 3(g) shows the secret grayscale image which was used as watermark image in size of  $114 \times 57$ ,  $315 \times 128$ ,  $597 \times 349$ ,  $615 \times 473$ ,  $1119 \times 453$ , and  $1728 \times 823$ . The proposed model was implemented using Intel(R) Core (TM) i7-4700MQ CPU, 2.40 GHz processor with 8 GB of RAM. Moreover, the MATLAB version 9.0.0.341360 (R2016b) was used in coding the implementation. In the encryption step, three seeds for the random number generator were selected to be 1987, 1989, and 1993.

In the field of data hiding, techniques are compared according to several parameters such as capacity and payload. The definition of capacity is that the maximum bits hidden within the 3D object vertices. The actual payload is the percentage between the current embedded bits and the capacity of the 3D object in bits. Table 1 presents the resultant capacity and the actual payload for each 3D object and the corresponding embedded secret grayscale image. That is, given  $L$  vertices in the original 3D, so the capacity in bits per vertex (bpv) and the actual payload in percent (%) can be computed as the following:

$$\begin{aligned} \text{capacity} &= \frac{\text{Max}(\text{number of embedded pixels}) \times 8}{\text{number of vertices}} \\ &= \frac{L \times 8}{L} = 8 \text{ bpv}, \\ \text{actual payload} &= \frac{\text{secret image size in bits} \times 3 \times 100}{3D \text{ object capacity in bits}}. \end{aligned} \quad (12)$$

Figure 4 shows the experiments and time taken to watermark and extract the maximum capacity for each 3D object and values of  $\beta$  between 1000 and 9000 to evaluate the time performance of the proposed method. Obviously, the extraction execution time is less than the watermarking execution time for the same 3D object. Thus, the average time performance is 3.11, 18.72, 94.56, 133.19, 227.15, and 636.90 seconds for watermarking process the Glock, egg, bunny, horse, cat, and angel models, respectively. In the same context, the average time performance for the extraction process is 1.99, 12.19, 60.98, 86.83, 146.38, and 406.65 seconds for the Glock, egg, bunny, horse, cat, and angel models, respectively. Obviously, the parameter  $\beta$  has a very little effect on the time execution for the same 3D object.

The imperceptibly and the transparency performances of the proposed method were evaluated using Euclidean distance, Manhattan distance, cosine distance, and the correlation distance. Let the original 3D object be  $u$  and the watermarked object be  $v$ , the invisibility performance term details were explained in the following equations.

The Euclidean distance is as follows:

$$\text{Euclidean dist}(u, v) = \sqrt{|u_x - v_x|^2 + |u_y - v_y|^2 + |u_z - v_z|^2}. \quad (13)$$

Manhattan distance is as follows:

```

(1) Function [EW_img] = Extraction ( $W\_3D\_Obj$ ,  $\beta$ ,  $Key_1$ ,  $Key_2$ ,  $Key_3$ )
(2) // $W\_3D\_Obj$ : Watermarked 3D object vertices,  $Key_1$ ,  $Key_2$  and  $Key_3$ : Encryption keys
(3) Normalized  $W\_3D\_Obj$  = Normalize( $W\_3D\_Obj$ )
(4) [Vertex  $x$ , Vertex  $y$ , Vertex  $z$ ] = Preprocessing(Normalized  $W\_3D\_Obj$ )
(5) for  $j = 1$  to length(Msg vector) do
(6) [Coefficient1, Coefficient2, Coefficient3] = DC T(Vertex  $x(j)$ , Vertex  $y(j)$ , Vertex  $z(j)$ )
(7) Extract the  $RecEncMsg$  value within the  $Coefficients$  values as the following equation system:
(8)  $RecEncMsg_{1,2,3} = (\beta/2)(\hat{C} - (2i/\beta)), (2i/\beta) \leq \hat{C} \leq (2(i+1)/\beta)$ ,

$$i = \begin{cases} 0, 1, 2, 3, \dots, (\beta - 1) & \hat{C} = \{Coefficient_1\} \\ -2, -1, 0, 1, \dots, (\beta - 3) & \hat{C} = \{Coefficient_2, Coefficient_3\} \end{cases}$$

(9) end
(10) [Rec Msg1] = Decrypt( $RecEncMsg_1$  vector,  $Key_1$ )
(11) [Rec Msg2] = Decrypt ( $RecEncMsg_2$  vector,  $Key_2$ )
(12) [Rec Msg3] = Decrypt ( $RecEncMsg_3$  vector,  $Key_3$ )
(13) Rec Msg vector = Average (Rec Msg1, Rec Msg2, Rec Msg3)
(14) Rec Msg = Reshape to image(Rec Msg vector)
(15) EW_img = DeNormalize(Rec Msg)
(16) return Extracted Watermark grayscale image EW_img
(17) end

```

ALGORITHM 2: Extraction function in the proposed method

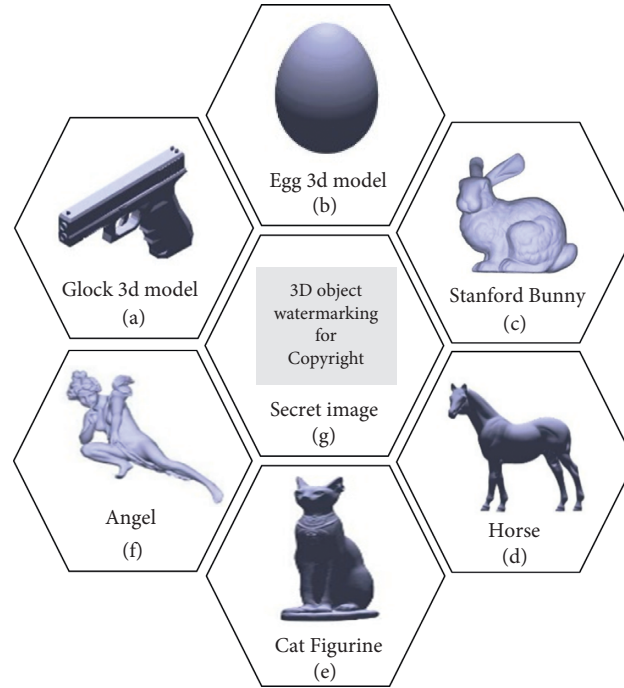


FIGURE 3: The 3D objects used in the implementation and the watermark image.

$$\text{Manhattan dist}(u, v) = |u_x - v_x| + |u_y - v_y| + |u_z - v_z|. \quad (14)$$

$$\text{cosine dist}(u, v) = 1 - \frac{u_x v_x + u_y v_y + u_z v_z}{\sqrt{|u_x|^2 + |u_y|^2 + |u_z|^2} \sqrt{|v_x|^2 + |v_y|^2 + |v_z|^2}} \quad (15)$$

Cosine distance is as follows:

TABLE 1: Embedding capacity results.

Model	Vertices	Faces	Secret image size (pixel)	Max capacity (bits)	Actual payload (%)
Glock 3D model	6564	2188	114 × 57	157536	98.995
Egg 3D model	40320	13440	315 × 128	967680	100
Stanford bunny	208353	35947	597 × 349	5000472	100
Horse	290898	48485	615 × 473	6981552	99.999
Cat figurine	506910	168970	1119 × 453	12165840	99.999
Angel	1422144	237018	1728 × 823	34131456	100

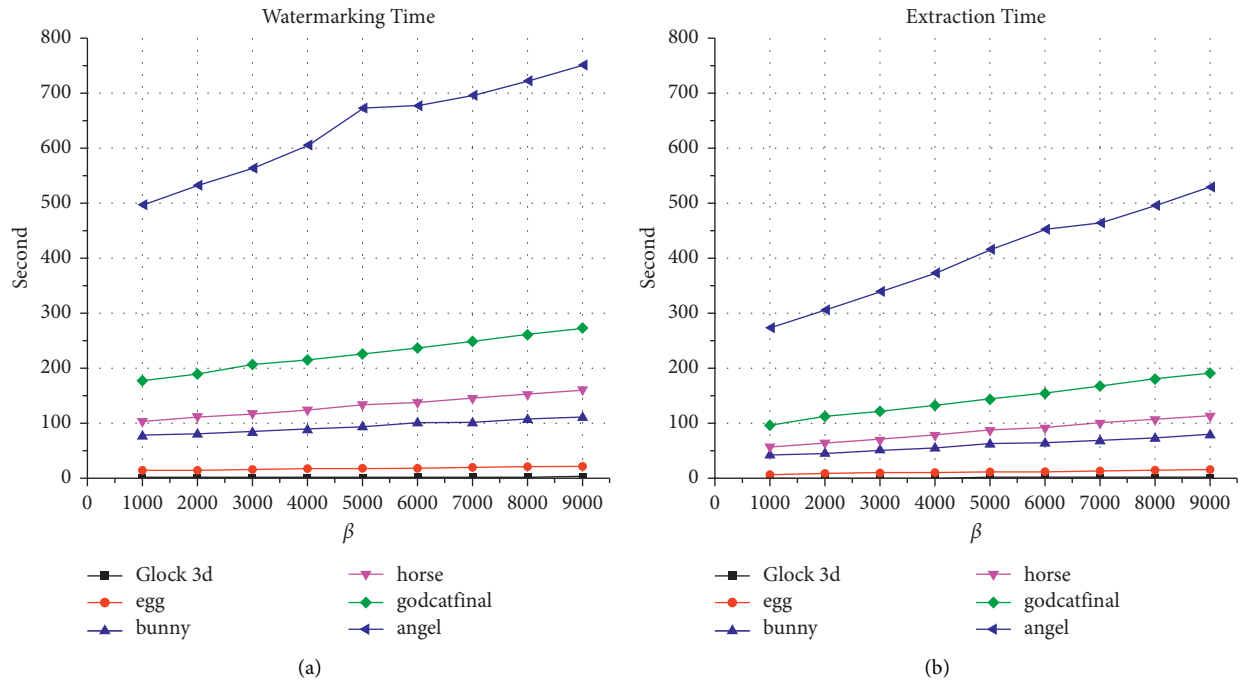


FIGURE 4: The watermarking and extraction time performance of the proposed method: (a) watermarking time; (b) extraction time.

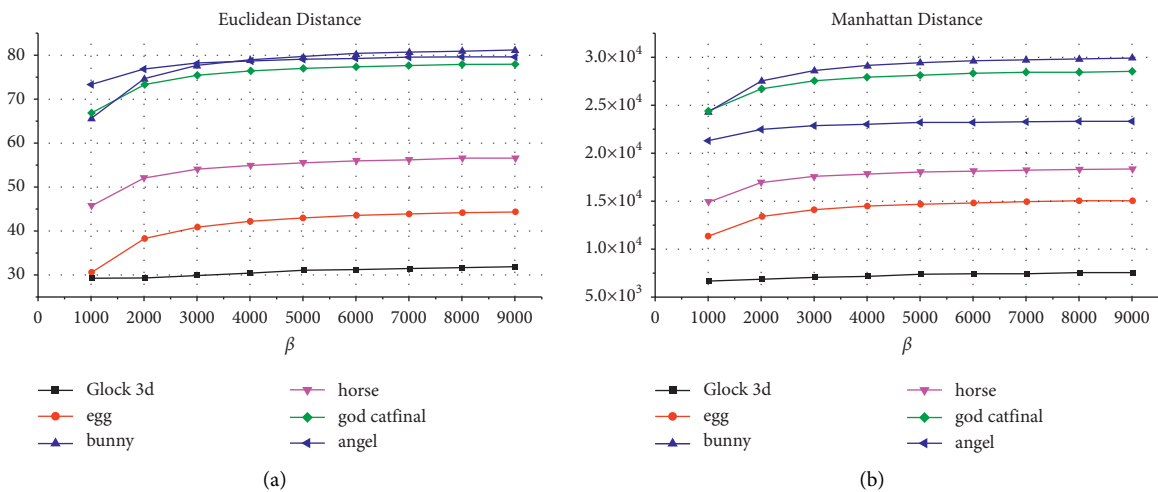


FIGURE 5: Continued.



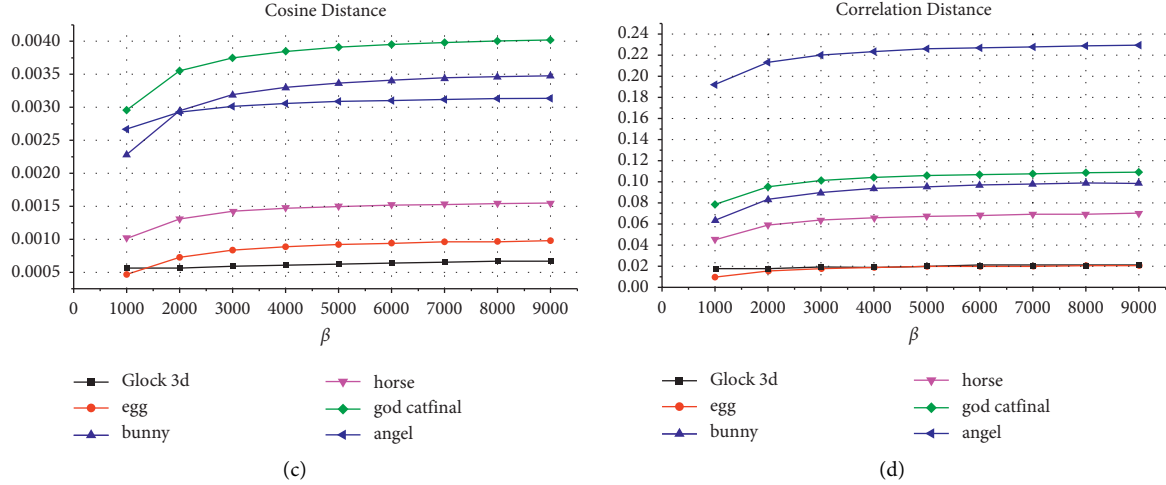


FIGURE 5: The invisibility performance of the proposed method: (a) Euclidean distance; (b) cosine distance; (c) Manhattan distance; (d) correlation distance.

TABLE 2: Comparison of recent schemes.

Scheme	Cover media	Watermark sequence	Embedding color space	Domain	Capacity	Is blind?
Chow et al. [38]	Grayscale image	QR code	Grayscale	DWT-DCT	0.0002	Yes
Amini et al. [40]	Grayscale image	Binary bits	Grayscale	DWT	0.0156	Yes
Rosales-Roldan et al. [39]	Color image	QR code	YCbCr	SVD-DWT-DCT	0.0833	Yes
Patvardhan et al. [41]	Color image	QR code	YCbCr	SVD-DWT	0.1667	No
Ayubi et al. [12]	Video	2D binary image	RGB	IWT-DWT-CT	0.0034	Yes
Hou et al. [42]	3D printed object	Binary pattern	High masking regions	Visual masking	1 bit	Yes
Delmotte et al. [43]	3D printed object	Binary bits	G-code	Layer thickness	>64 bits	Yes
Jiang et al. [44]	3D object	Binary bits	Vertices	Encrypted domain	0.3692	Yes
Cayre et al. [45]	3D object	Binary bits	Vertices	Spatial	0.8772	Yes
Wu et al. [46]	3D object	2D binary image	Vertices	Spatial	0.9969	No
Khalil et al. [18]	3D object	Grayscale image	Vertices	Spatial	2.6667	Yes
The proposed	3D object	Grayscale image	Vertices	DCT	8	Yes

The correlation distance is as follows:

correlation dist( $u, v$ ) = 1

$$\frac{((1/3)(-u_x - u_y - u_z) + u_x)((1/3)(-v_x - v_y - v_z) + v_x) + ((1/3)(-u_x - u_y - u_z) + u_y)((1/3)(-v_x - v_y - v_z) + v_y) + ((1/3)(-u_x - u_y - u_z) + u_z)((1/3)(-v_x - v_y - v_z) + v_z)}{\sqrt{|u_x + (1/3)(-u_x - u_y - u_z)|^2 + |u_y + (1/3)(-u_x - u_y - u_z)|^2 + |(1/3)(-u_x - u_y - u_z) + u_z|^2} \sqrt{|v_x + (1/3)(-v_x - v_y - v_z)|^2 + |v_y + (1/3)(-v_x - v_y - v_z)|^2 + |(1/3)(-v_x - v_y - v_z) + v_z|^2}} \quad (16)$$

Figure 5 shows the obtained Euclidean distance, Manhattan distance, cosine distance, and the correlation distance result to compare between the original 3D object and the resultant watermarked using values of  $\beta$  between 1000 and 9000. Moreover, the experiments investigate the effect of the embedding parameters  $\beta$  on the fidelity of the embedding when applying the maximum capacity. The average resultant Euclidean distance for Glock, egg, bunny, horse, cat, and angel objects is 30.73, 41.14, 77.62, 54.22, 75.48, and 78.16, respectively. The average resultant Manhattan distance for Glock, egg, bunny, horse, cat, and angel objects is  $7.29E + 03$ ,  $1.42E + 04$ ,  $2.87E + 04$ ,  $1.77E + 04$ ,  $2.76E + 04$ , and  $2.29E + 04$ ,

respectively. The average resultant cosine distance for Glock, egg, bunny, horse, cat, and angel objects is 0.0006, 0.0008, 0.0032, 0.0014, 0.0038, and 0.0030, respectively. The average resultant correlation distance for Glock, egg, bunny, horse, cat, and angel objects is 0.02, 0.02, 0.09, 0.06, 0.10, and 0.22, respectively. The obtained results show that  $\beta$  provides better invisibility performances of the watermarked object for large values in various terms.

The main characteristics comparison of the proposed method with other existing methods are introduced to confirm its validity and efficiency. The comparative study is conducted in order to verify the used cover media, the

watermark sequence, the embedding space, the domain, the capacity, and the blindness extraction process between the proposed method and other methods. Table 2 shows a comparison of the recorded details of the related methods. In [38–41], the presented methods were based on embedding the QR code and binary bits into images based on various domains. Moreover, Ayubi et al. in [12] presented a video watermarking method using 2D binary image. On the other hand, in [42, 43], the proposed methods watermarked the 3D printed object, and in [18, 44–46], the presented methods were based on watermarking the 3D object using a different watermark sequence. The capacities are recorded by the number of bits per pixel (bpp) and the number of bits per vertex (bpv). The proposed 3D mesh watermarking technique achieves the advantage characteristic of using high-capacity number of bits per vertices. On the contrary, watermarking schemes for copyright protection target the shape and their capacity are usually a fewer number of bits.

## 5. Conclusions

In this paper, a novel model for 3D object watermarking was proposed. The proposed method is based on the properties of the discrete cosine transform (DCT) of the 3D object vertices to embed a secret grayscale image three times. The extraction process is totally blind based on the reverse steps of the embedding process to recover the secret grayscale image. The proposed model was implemented using MATLAB, and the time performance of the proposed method was recorded. The resultant maximum capacity of the proposed method for each 3D object and the corresponding embedded secret grayscale image is 8 bits per vertex (bpv). The imperceptibility and the transparency performances of the proposed method were evaluated using Euclidean distance, Manhattan distance, cosine distance, and the correlation distance.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

## Acknowledgments

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant code: 19-SCI-1-03-0018.

## References

- [1] R. S. Rashid and J. R. Mohammed, "Securing speech signals by watermarking binary images in the wavelet domain," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 1096–1103, 2020.
- [2] G. Kaur, R. Agarwal, and V. Patidar, "Crypto-watermarking of images for secure transmission over cloud," *Journal of Information and Optimization Sciences*, vol. 41, no. 1, pp. 205–216, 2020.
- [3] W. Sun, J. Zhou, Y. Li, M. Cheung, and J. She, "Robust high-capacity watermarking over online social network shared images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, 2020.
- [4] L.-Y. Hsu and H.-T. Hu, "Blind watermarking for color images using EMMQ based on QDFT," *Expert Systems with Applications*, vol. 149, Article ID 113225, 2020.
- [5] S.-H. Nam, S.-M. Mun, W. Ahn et al., "NSCT-based robust and perceptual watermarking for DIBR 3D images," *IEEE Access*, vol. 8, pp. 93760–93781, 2020.
- [6] S. Hamad, A. Khalifa, and A. Elhadad, "A blind high-capacity wavelet-based steganography technique for hiding images into other images," *Advances in Electrical and Computer Engineering*, vol. 14, no. 2, pp. 35–42, 2014.
- [7] M. Yousefi Valandar, M. Jafari Barani, and P. Ayubi, "A blind and robust color images watermarking method based on block transform and secured by modified 3-dimensional Hénon map," *Soft Computing*, vol. 24, no. 2, pp. 771–794, 2020.
- [8] A. Elhadad, S. Hamad, A. Khalifa, and A. Ghareeb, "High capacity information hiding for privacy protection in digital video files," *Neural Computing & Applications*, vol. 28, no. 1, pp. 91–95, 2017.
- [9] R. Singh, S. Nigam, A. K. Singh, and M. Elhoseny, "On wavelet domain video watermarking techniques," in *Intelligent Wavelet Based Techniques for Advanced Multimedia Applications*, pp. 65–76, Springer, Berlin, Germany, 2020.
- [10] K. A. Zhang, L. Xu, A. Cuesta-Infante, and K. Veeramachaneni, "Robust invisible video watermarking with attention," 2019, <https://arxiv.org/abs/1909.01285>.
- [11] S. Das, A. K. Sunaniya, R. Maity, and N. P. Maity, "Efficient FPGA implementation of corrected reversible contrast mapping algorithm for video watermarking," *Microprocessors and Microsystems*, vol. 76, Article ID 103092, 2020.
- [12] P. Ayubi, M. Jafari Barani, M. Yousefi Valandar, B. Yosefzad Irani, and R. Sedagheh Maskan Sadigh, "A new chaotic complex map for robust video watermarking," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 1237–1280, 2021.
- [13] H. Karajeh, T. Khatib, L. Rajab, and M. Maqableh, "A robust digital audio watermarking scheme based on DWT and Schur decomposition," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18395–18418, 2019.
- [14] M.-J. Lin, M. Zhao, S.-T. Chen, and C.-J. Chen, "Digital audio watermarking by amplitude embedding system," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, pp. 143–150, Springer, Berlin, Germany, 2020.
- [15] W. Lu, L. Li, Y. He, J. Wei, and N. N. Xiong, "RFPS: a robust feature points detection of audio watermarking for against desynchronization attacks in cyber security," *IEEE Access*, vol. 8, pp. 63643–63653, 2020.
- [16] S. Hamad, A. Elhadad, and A. Khalifa, "DNA watermarking using Codon Postfix technique," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1605–1610, 2017.
- [17] S.-H. Lee, E.-J. Lee, and K.-R. Kwon, "A reversible watermarking for DNA sequence using an adaptive least square prediction error expansion," *Nonlinear Theory and Its Applications, IEICE*, vol. 11, no. 1, pp. 2–15, 2020.
- [18] O. H. Khalil, A. Elhadad, and A. Ghareeb, "A blind proposed 3D mesh watermarking technique for copyright protection," *The Imaging Science Journal*, vol. 68, no. 2, pp. 90–99, 2020.

- [19] S. Borah and B. Borah, "Watermarking techniques for three dimensional (3D) mesh authentication in spatial domain," *3D Research*, vol. 9, no. 3, p. 43, 2018.
- [20] H. Al-Khafaji and C. Abhayaratne, "Graph spectral domain blind watermarking," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2492–2496, IEEE, Brighton, UK, 2019.
- [21] J. Wang, J. Feng, and Y. Miao, "A robust confirmable watermarking algorithm for 3D mesh based on manifold harmonics analysis," *The Visual Computer*, vol. 28, no. 11, pp. 1049–1062, 2012.
- [22] F. A. Ferreira and J. B. Lima, "A robust 3D point cloud watermarking method based on the graph Fourier transform," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 1921–1950, 2020.
- [23] J. Liu, Y. Wang, Y. Li, R. Liu, and J. Chen, "A robust and blind 3D watermarking algorithm using multiresolution adaptive parameterization of surface," *Neurocomputing*, vol. 237, pp. 304–315, 2017.
- [24] Q. S. Ai, Q. Liu, Z. D. Zhou, L. Yang, and S. Q. Xie, "A new digital watermarking scheme for 3D triangular mesh models," *Signal Processing*, vol. 89, no. 11, pp. 2159–2170, 2009.
- [25] J. Liu, Y. Yang, D. Ma, Y. Wang, and Z. Pan, "A watermarking algorithm for 3D point cloud models using ring distribution," in *Transactions on Edutainment XIV*, pp. 56–68, Springer, Berlin, Germany, 2018.
- [26] X. Zhang and Z. Shen, "Copyright protection method for 3D model of geological body based on digital watermarking technology," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 334–346, 2019.
- [27] O. M. El Zein, L. M. El Bakrawy, and N. I. Ghali, "A robust 3D mesh watermarking algorithm utilizing fuzzy C-Means clustering," *Future Computing and Informatics Journal*, vol. 2, no. 2, pp. 148–156, 2017.
- [28] N. Medimegh, S. Belaid, M. Atri, and N. Werghe, "A novel robust statistical watermarking of 3D meshes," in *International Workshop on Representations, Analysis and Recognition of Shape and Motion From Imaging Data*, pp. 27–38, Springer, Berlin, Germany, 2016.
- [29] N. Medimegh, S. Belaid, M. Atri, and N. Werghe, "3D mesh watermarking using salient points," *Multimedia Tools and Applications*, vol. 77, no. 24, pp. 32287–32309, 2018.
- [30] J.-W. Cho, R. Prost, and H.-Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 142–155, 2006.
- [31] A. M. Molaei, H. Ebrahimnezhad, and M. H. Sedaaghi, "A blind fragile watermarking method for 3D models based on geometric properties of triangles," *3D Research*, vol. 4, no. 4, p. 4, 2013.
- [32] B. Hansda, V. Maheshkar, and S. Maheshkar, "A robust watermarking scheme for 3D polygon mesh models using mean curvature," in *Recent Findings in Intelligent Computing Techniques*, pp. 511–517, Springer, Berlin, Germany, 2018.
- [33] J. Son, D. Kim, H.-Y. Choi, H.-U. Jang, and S. Choi, "Perceptual 3D watermarking using mesh saliency," in *Information Science and Applications*, pp. 315–322, Springer, Berlin, Germany, 2017.
- [34] H. Liu, S.-G. Li, H.-X. Wang, and G.-J. Li, "Adaptive fuzzy synchronization for a class of fractional-order neural networks," *Chinese Physics B*, vol. 26, no. 3, p. 030504, 2017.
- [35] H. Liu, S. Li, H. Wang, Y. Huo, and J. J. E. Luo, "Adaptive synchronization for a class of uncertain fractional-order neural networks," *Entropy*, vol. 17, no. 10, pp. 7185–7200, 2015.
- [36] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2174–2193, 1992.
- [37] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 23, no. 1, pp. 90–93, 1974.
- [38] Y.-W. Chow, W. Susilo, J. Tonien, and W. Zong, "A QR code watermarking approach based on the DWT-DCT technique," *Information Security and Privacy*, vol. 32, pp. 314–331, 2017.
- [39] L. Rosales-Roldan, J. Chao, M. Nakano-Miyatake, and H. Perez-Meana, "Color image ownership protection based on spectral domain watermarking using QR codes and QIM," *Multimedia Tools and Applications*, vol. 77, no. 13, pp. 16031–16052, 2018.
- [40] M. Amini, M. O. Ahmad, M. J. I. T. O. C. Swamy, and S. F. V. Technology, "A robust multibit multiplicative watermark decoder using a vector-based hidden Markov model in wavelet domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 402–413, 2016.
- [41] C. Patvardhan, P. Kumar, and C. Vasantha Lakshmi, "Effective color image watermarking scheme using YCbCr color space and QR code," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12655–12677, 2018.
- [42] J.-U. Hou, D.-G. Kim, and H.-K. Lee, "Blind 3D mesh watermarking for 3D printed model by analyzing layering artifact," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2712–2725, 2017.
- [43] A. Delmotte, K. Tanaka, H. Kubo, T. Funatomi, and Y. M. Mukaigawa, "Blind watermarking for 3-D printed objects by locally modifying layer thickness," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2780–2791, 2019.
- [44] R. Jiang, H. Zhou, W. Zhang, and N. Yu, "Reversible data hiding in encrypted three-dimensional mesh models," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 55–67, 2017.
- [45] F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 939–949, 2003.
- [46] H.-T. Wu and Y.-M. Cheung, "A high-capacity data hiding method for polygonal meshes," in *Proceedings of the International Workshop on Information Hiding*, pp. 188–200, Springer, Alexandria, VA, USA, 2006.