A DCT-Domain Watermarking Algorithm for Three-Dimensional Triangle Meshes

Jeonghee Jeon, Sang-Kwang Lee, and Yo-Sung Ho

Kwangju Institute of Science and Technology (K-JIST)
1 Oryong-dong Puk-gu, Kwangju, 500-712, Korea
{jhjeon, sklee, hoyo}@kjist.ac.kr

Abstract. Most watermarking techniques insert watermark signals in the frequency domain because it is easier to design robust or imperceptible watermarking algorithms against malicious attacks. However, three-dimensional (3-D) geometric models lack natural parameterization for frequency-based decomposition. In this paper, we propose a new scheme for 3-D model watermarking in the DCT domain. In order to insert and extract watermark signals in the frequency domain, we generate triangle strips by traversing the 3-D mesh model and transform its vertex coordinates in the spatial domain into DC and AC coefficients in the DCT domain. Watermarks are inserted into the mid-frequency band of AC coefficients for robustness and imperceptibility. We show that the inserted watermarks survive the attacks, such as additive random noise, geometry compression by the MPEG-4 SNHC standard, affine transformation, and multiple watermarking.

1 Introduction

Digital media data that have become very common on the Web in recent years, provide a lot of advantages, such as complete duplication and ease distribution through the Web. However, owners of digital data are now being faced with unauthorized threaten of illegal users. In order to protect the ownership or copyright of digital media data, such as image, video, and audio, we try to employ data encryption and watermarking techniques. While data encryption techniques are mainly used to protect digital data during the transmission from the sender to the receiver, digital watermarking techniques are used for copyright protection, fingerprinting, broadcast monitoring, data authentication, indexing, medical safety, and data hiding [1][2].

Digital watermarking for three-dimensional (3-D) geometric models, such as surface model, solid model, or polygonal model, has received less attention from researchers because digital watermarking techniques for image, video and audio data cannot directly be applied to the work for 3-D data. The major reasons are that arbitrary surfaces of 3-D models lack natural parameterization for frequency-based decomposition and that simplification or other attacks may modify connectivity of the 3-D mesh model [3]. Most watermarking schemes for 1-D and 2-D data insert watermarks in the frequency domain because it is easier to design robust and imperceptible watermarking schemes in the frequency bands against possible malicious attacks.

Kanai et al. proposed a watermarking scheme for 3-D polygonal models based on the wavelet transform [4]. Their paper is the first one that applied a transformed-domain watermarking approach to 3-D meshes. Their watermarks are robust against affine transformation, partial resection, and random noise added to vertex coordinates. However, their scheme requires the mesh to have 1-to-4 subdivision connectivity. Praun et al. presented a robust watermarking scheme that is applicable to 3-D models of arbitrary vertex connectivity in the transform domain [3]. Their scheme modifies the shape of the mesh by a spatial kernel to insert watermarks in the low frequency band of the shape information. Yin et al. [5] reported a watermarking algorithm based on multiresolution decomposition of polygonal meshes by Guskov's signal processing method [6], which separates a mesh into detail and coarse feature sequences. They demonstrated that the algorithm is robust against vertex reordering, noise addition, simplification, filtering and enhancement, cropping, etc. Ohbuchi, et al. proposed a frequency-domain approach to watermarking 3-D shapes [7]. The algorithm employs mesh-spectral analysis to modify mesh shapes in their transform domain and it is robust against mesh simplification, and remeshing combined with resection, similarity transformation, and other attacks.

In this paper, we propose a new watermarking scheme for 3-D triangle meshes in the discrete cosine transform (DCT) domain. After we explain the concept of the proposed watermarking scheme, we describe watermark insertion and extraction operations in Section 2. Section 3 presents experimental results of the proposed scheme, and we conclude in Section 4.

2 The Proposed Watermarking Scheme for 3-D Mesh Models

As shown in Fig. 1, our proposed watermarking scheme consists of three main function blocks: creating triangle strips, forward DCT and inverse DCT transform for vertex coordinates of each strip, and watermark generation.

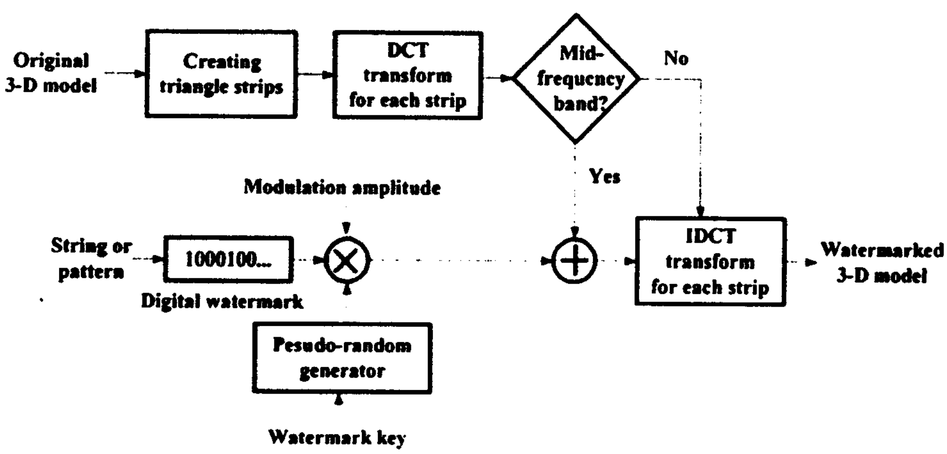


Fig. 1. Watermark insertion scheme

After the original 3-D model is represented by vertex coordinates and connectivity of VRML (virtual reality modeling language) file, we create triangle strips by traversing the model. We then transform the set of the vertex coordinates in the spatial domain into the DCT domain along the x-, y-, and z-coordinate, independently. Watermark signals are inserted into the mid-frequency band of AC coefficients, and then the watermarked 3-D model is represented after the inverse DCT transform.

2.1 Creating Triangle Strip

Efficient rendering of triangle-based meshes often requires that we pack them into triangle strips. Strips provide interesting advantages: the mesh is stored in a more compact way because of wasting less memory space, and we can save bandwidth when we send it to our favorite rendering application program interface (API).

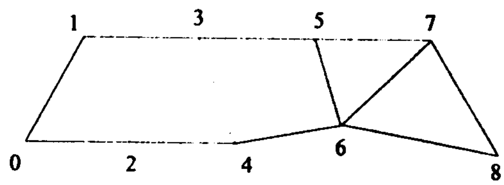


Fig. 2. A triangle strip

As shown in Fig. 2, the 3-D mesh model contains a list of connected triangles. Each triangle is made of three vertex references and two edges of them may be shared with one triangle or another. From this sharing, the list of indices forms a triangle strip. Thus, we can describe the triangulation using the strip {0,1,2,3,4,5,6,7,8} and assume the convention that the i-th triangle is described by i-th, $(i+1)$-th, and $(i+2)$-th vertices of the sequential strip. The sequential strip can reduce the cost to transmit n triangles from $3n$ to $(n+2)$ vertices. In order to create triangle strips from a triangle mesh, we use the following algorithm [8].

Step 1. Choose a starting face for each strip.
Step 2. Choose a direction (i.e. an edge) in which we walk along the strip.
Step 3. Extend the strip in the chosen direction until there are no forward connections.
Step 4. Reverse the strip and extend it in the opposite direction.
Step 5. Go to *Step 1* until all faces have been visited.

Here, we note that the number of the triangle strips created by the algorithm can be different depending on mesh connectivity information. The number of vertices within each strip can also be different. Therefore, we can exploit these characteristics to design a new watermarking algorithm. First, the user who does not know the starting face for creating triangle strips cannot distinguish a watermark pattern because the starting face is determined by connectivity of the 3-D mesh. It is a very useful property for information hiding. Second, the triangle strips also have an attribution of mesh partition, such as a subset of the original mesh. Thus, if this attribution is used for 3-D model watermarking, we can improve robustness of watermarks from malicious attacks that try to remove watermarks. Finally, the multiple triangle strips can play an important role for strengthening robustness because watermarks can be repeatedly inserted into them. As a result, we can make use of these properties of triangle strips in designing a new watermarking scheme for 3-D triangle models.

2.2 DCT and IDCT Transform of Triangle Strips

The discrete cosine transform (DCT) for 3-D geometric models is found in geometry compression of 3-D animation models [9]. They select one root triangle of the given 3-D model randomly and traverse neighboring vertices from the root triangle in the clockwise direction. The traversed edge is called the cutting edge since it cuts the given triangle along the edge. Once a triangle strip is obtained, we segment it by grouping vertices uniformly. Therefore, this segmentation algorithm produces several independent parts according to the topology of the 3-D model and the decomposition can group the similar vertices in the spatial domain. After the segmented blocks are transformed independently by 1-D DCT along the x-, y-, and z-coordinate, we encode the DCT coefficients. The definition of the 1-D forward DCT and inverse DCT are given by

$$\text{Forward DCT: } X(k) = \sqrt{\frac{2}{N}}\, C_k \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \quad k = 0,1,\ldots,N-1 \quad (1)$$

$$\text{Inverse DCT: } x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C_k X(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \quad n = 0,1,\ldots,N-1 \quad (2)$$

where $C_k = \begin{cases} 1/\sqrt{2} & k = 0 \\ 1 & k \neq 0 \end{cases}$.

In order to transform vertex coordinates of each strip, we employ the 1-D DCT transform. While the number of strips used in the algorithm is one, the number of triangle strips generated by the algorithm is not just one, but multiple. Note that the number of strips can be different depending on the connectivity information of the 3-D mesh. The size of the segmented block can be adjusted by padding the last AC coefficient, as shown in Fig. 3.

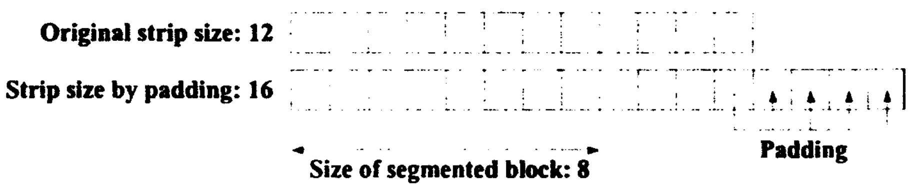


Fig. 3. Adjustment of segmented block size

2.3 Watermark Insertion

In the frequency domain, the low-frequency band represents the global shape and the high-frequency band describes local or detail contents. In various watermarking schemes, the frequency band is frequently used to insert watermark signals because it is not only imperceptible but also robust. In our scheme, we embed watermark signals into the mid-frequency band of AC coefficients in the DCT domain, as shown in Fig.

1, where the watermark insertion operation is similar to the spread-spectrum approach [10][11]. The watermark signal can be a company logo, some meaningful string, or random values. In this paper, a string of five characters is used as a watermark. The string provided by the user is converted into the ASCII code a as

$$a_j = (a_1, \cdots, a_m),\ a_j \in \{0,1\} \tag{3}$$

In order to spread the digit signal into a wide bandwidth, each bit a_j is duplicated by the chip rate c.

$$b_i = a_j,\quad j \cdot c \le i < (j+1) \cdot c \tag{4}$$

The chip rate c plays an important role of increment of robustness against an additive random noise. If b_i is zero, it is changed to the negative sign by

$$b_i' = 2b_i - 1 \tag{5}$$

Let us now consider the modulation operation of DCT coefficients in the x-coordinate only. As shown in Fig. 1, we insert the watermark signal into the mid-frequency band of AC coefficients. New DCT coefficients $\overline{W}_{x,j}$ are derived from the current DCT coefficients $W_{x,i}$ by

$$\overline{W}_{x,j} = W_{x,i} + b_i' \cdot p_i \cdot \alpha \tag{6}$$

where b_i' is the watermark signals, α is the modulation amplitude, and p_i is a pseudo-random number sequence generated by the watermark key. A bandwidth $MidFrequencyBand$ is determined by

$$MidFrequencyBand = (Min, Max] \tag{7}$$

$$\begin{cases} Min = (\frac{BlockSize}{2})/2 \\ Max = Min + (\frac{BlockSize}{2}) \end{cases}$$

where *BlockSize* is the size of the segmented block defined by the user. As a result, Eq. (6) produces modified coefficients. In order to insert watermark signals into the coefficients of y- and z-coordinates, we also apply the same operations. The watermarked 3-D model is easily represented by Eq. (2).

2.4 Watermark Extraction

Fig.4 shows the watermark extraction operation in the proposed private watermarking algorithm, which needs the original and the watermarked 3-D models. This operation can be expressed by

$$b_i' = (\overline{W}_{x,i} - W_{x,i}) \cdot p_i \tag{8}$$

The extracted b' is changed to b by Eq. (5) and b is converted to a by Eq. (4). After a is converted to a string by the ASCII code, we can assert the ownership by the extracted string.

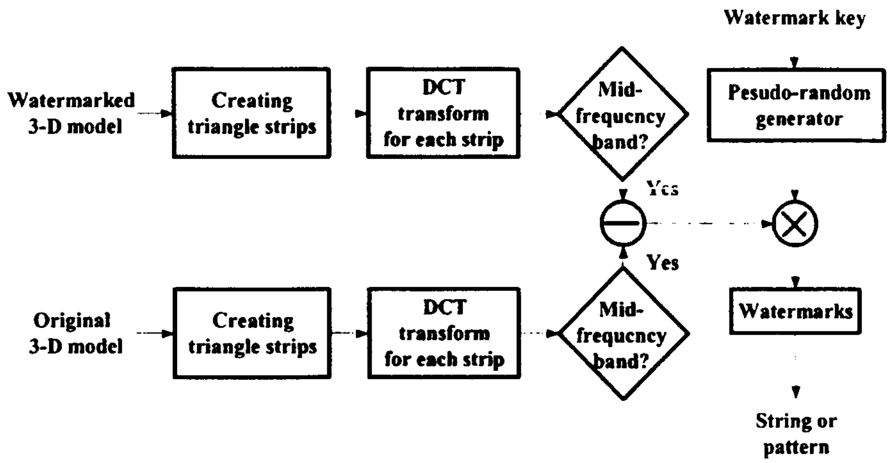


Fig. 4. Watermark extraction scheme

3 Experimental Results and Analysis

An experimental system for 3-D watermark insertion and watermark extraction has been developed in MS VC++ 6.0. In order to evaluate the perceptual invisibility between the original and the watermarked models and resiliency against various attacks, such as additive random noise, mesh compression, affine transformation, and multiple watermarking, we perform computer simulations on 3-D mesh models: Beethoven model with 2521 vertices and 5030 faces, Horse model with 2620 vertices and 5200 faces, and Bunny model with 3116 vertices and 6100 faces, as shown in Fig. 5(a), Fig. 5(b), and Fig. 5(c), respectively.

3.1 Watermarking Parameters

Modulation amplitude. Basically, the modulation amplitude α is selected by the user in such a way that it is small enough to preserve appearance of the model while it is large enough to withstand from malicious attacks.

Perceptual invisibility. We employ the MESH (measuring error between surfaces using the Hausdorff distance) tool [13] to measure the distortion, which indicates the degree deformed by watermarks or attacks. The MESH tool evaluates on a variety of 3-D mesh models with the root-mean-square (RMS) distance as a function of the sampling step, which plays an important role in the precision of the measured distance. Among various methods to calculate RMS distances, we use one with the symmetric distance.

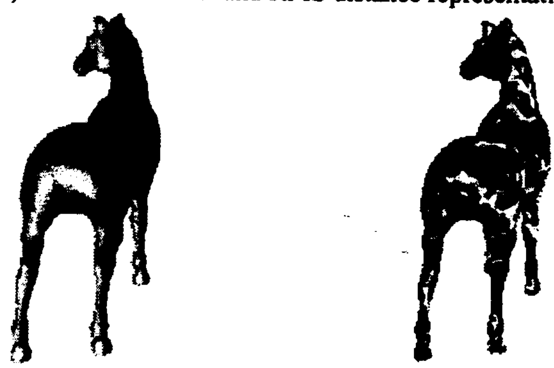
Watermark lengths, chip rate, and block size. In this paper, the watermark string, the chip rate, and the size of segmented block for DCT are "KJIST" with 35 bits, one, and eight, respectively. As we mentioned, even if the chip rate plays an important role for strengthening robustness of watermarks, we use one as the parameter value because watermarks are repeatedly inserted into the mid-frequency band of AC coefficients along the x-, y-, and z-coordinates of multiple triangle strips.

3.2 Perceptual Invisibility

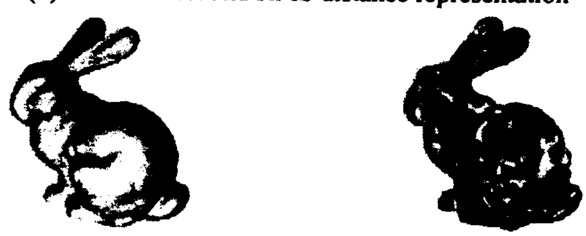
In order to calculate distortions in 1-D signals or 2-D images, we employ various measures, such as the mean square error (MSE) and peak signal-to-noise ratio (PSNR). However, a measure for 3-D mesh models has rarely covered. Recently, Aspert et al. reported a method to estimate the distance between discrete 3-D surfaces [13]. We employ their method to measure the distortion of watermarked 3-D models.

Basically, the original data should be perceptually unchanged by embedding watermarks, but be imposed small modifications on it. In order to minimize a variety of the original 3-D model and withstand from some attacks, we inserted watermarks into the mid-frequency band. Simulation results for the RMS distance between the original and the watermarked models are shown in Fig. 5 and Table 1.

(a) Beethoven model and RMS distance representation



(b) Horse model and RMS distance representation



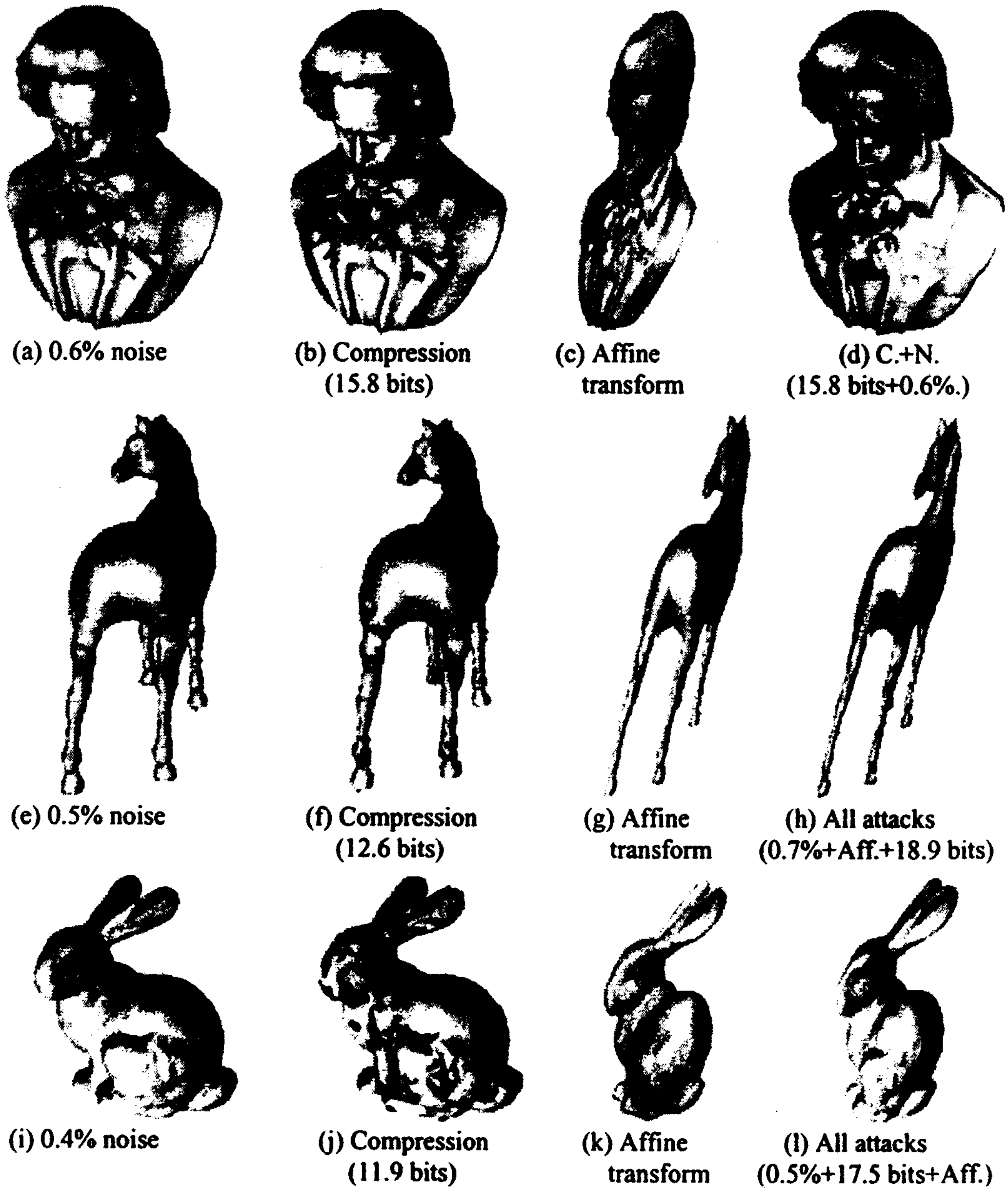
(c) Bunny model and RMS distance representation

Fig. 5. The original model and visual representation of RMS distance between the original and watermarked models

Fig. 5 shows that watermarks are located in the areas with the mid-frequency band, which is not fully smoothing and not roughly bumping. When the RMS distance of the watermarked model is increased, the surface color is dissolved. There is a trade-off between imperceptibility and robustness.

3.2 Resiliency against Various Attacks

Watermark detections for attacked Beethoven, Horse, and Bunny models are listed in Table 1, and some models are shown in Fig. 6. The models embedded with the modulation amplitude α in Table 1 are attacked by additive random noise, geometry compression by the MPEG-4 SNHC coding standard, affine transform, and multiple watermarking.



(a) 0.6% noise (b) Compression (15.8 bits) (c) Affine transform (d) C.+N. (15.8 bits+0.6%.)

(e) 0.5% noise (f) Compression (12.6 bits) (g) Affine transform (h) All attacks (0.7%+Aff.+18.9 bits)

(i) 0.4% noise (j) Compression (11.9 bits) (k) Affine transform (l) All attacks (0.5%+17.5 bits+Aff.)

Fig. 6. Attacked models

Table 1. Resiliency against attacks

Models (Number of inserted string)	α (Alpha)	RMS	Attacks								
			Additive Random Noise (%)			Geometry Compression (96 bits)			Aff.	Multiple Watermarking	
Beethoven (21)	3.05	0.075	0.6*	0.4	0.2	15.8*	18.8	21.8	7*	C+N*	All
			16	20	20	11	20	21		2	0
Horse (24)	0.0005	0.079	0.5*	0.4	0.2	12.3*	15.3	18.3	9*	N+A	All*
			18	20	23	1	15	23		7	6
Bunny (23)	0.0006	0.097	0.5	0.4*	0.2	11.9*	14.6	17.5	7*	N+C	All*
			19	19	21	0	2	19		12	7

(C: Geometry Compression, N: Additive Random Noise)
(Entries with asterisks are shown in Fig. 6)

As the additive random noise attack, we add noises to vertex coordinates of the watermarked model with a uniform random noise. In Fig. 6 and Table 1, the percentage of the additive random noise represents the ratio between the largest displacement and the largest side of the watermarked model [3]. As show in Table 1, for example, our algorithm extracts the full string "KJIST" of sixteen, twenty, and twenty numbers, respectively, when 0.6, 0.4, and 0.2 % random noises are uniformly added to the watermarked Beethoven model.

For the case of the compression attack, we also apply the geometry compression by the MPEG-4 SNHC standard [12]. Generally, the x-, y-, z-vertex coordinates of the 3-D polygonal model are stored by the floating-point variable of 32 bits per each coordinate. Thus, each of the first row elements in the compression attack is the bit number compressed by the MPEG-4 standard compression algorithm, and each of the second row elements represents the number of strings recovered completely.

For the affine transformation, we translate, scale, rotate, and shear the watermarked model. The multiple watermarking attacks have two different types. The first type is a consecutive attack to the first column of multiple watermarking in Table 1. For example, C+N attack of the Beethoven model adds random noise to the watermarked model after geometry compression. The second type is all attack, which applies the affine transformation to the Beethoven model after the first type attack.

In order to evaluate performance of the proposed algorithm, we do not use the bit error rate (BER), which is the ratio of the numbers of the inserted and extracted watermarks because the owner can clearly assert the ownership of the 3-D model through perfect reconstruction of the inserted watermark string. From Table 1 and Fig. 6, we notice that the inserted watermark survives various attacks, such as additive random noise, geometry compression by the MPEG-4 SNHC standard, affine transformation, and multiple watermarking. Especially, our watermarking scheme is very robust against signal processing operations, such as additive random noise; however, it is a little weak against geometry attacks.

4 Conclusions

In this paper, we have developed a new watermarking algorithm for 3-D triangle meshes in the DCT domain. The proposed embedding scheme inserted watermarks into the mid-frequency band of AC coefficients because the mid-frequency band is less sensitive to the human eye than high- or low-frequency bands. Using the RMS distance, we have shown that our watermark insertion is effective for imperceptibility of the watermarked 3-D model and also demonstrated that the watermark embedded by our algorithm has strong resiliency against various attacks, such as additive random noise, geometry compression by the MPEG-4 standard, affine transformation, and multiple watermarking. Furthermore, the proposed scheme is robust against various signal processing attacks.

Acknowledgements

This work was supported in part by Kwangju Institute of Science and Technology (K-JIST), in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting IT Research Center at K-JIST, and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

References

1. Katzenbeisser, L. and Petitcolas, F. A. P.: Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, (2000)
2. Langelaar, G. C., Setyawan, I., and Lagendijk, R. L.: Watermarking Digital Image and Video Data, IEEE Signal Processing Magazine, (2000) 20-46
3. Praun, E., Hoppe, H., and Finkelstein, A.: Robust Mesh Watermarking, SIGGRAPH Proceedings, (1999) 49-56
4. Kanai, S., Date, H., and Kishinami, T.: Digital Watermarking for 3D Polygons using Multiresolution Wavelets Decomposition, Proceedings Sixth IFIP WG 5.2 GEO-6, (1998) 296-307
5. Yin, K., Pan, Z., Shi, J., and Zhang, D.: Robust Mesh Watermarking Based on Multiresolution Processing, Computer and Graphics, Vol. 25, (2001) 409-420
6. Guskov, I., Sweldens, W., and Shröder, P.: Multiresolution Signal Processing for Meshes, SIGGRAPH Proceedings, (1999) 325-334
7. Obuchi, R., Mukaiyama, A., and Takahashi, S.: A Frequency-Domain Approach to Watermarking 3D Shapes, Computer Graphics Forum, Vol. 21, No. 3, (2002) 373-382
8. Terdiman, P's web site: http:// codercorner.com/Strips.htm
9. Ahn, J. H., Kim, C. S., Kuo, C. C., and Ho, Y. S.: Motion Compensated Compression of 3D Animation Models, SPIE Visual Communications and Image Processing, (2002) 593-602
10. Cox. I.J., Kilian, J., Leighton, T., and Shamoon, T.: Secure Spread Spectrum Watermarking for Multimedia, IEEE Transactions on Image Processing, Vol. 6, No. 12, (1997) 1673-1687
11. Hartung, F., Eisert, P., and Griod, B.: Digital Watermarking of MPEF-4 Facial Animation Parameters, Computer and Graphics, Vol. 22, No. 4, (1998) 425-435
12. ISO/IEC 14496-2, Coding of Audio-Visual Objects: Visual (MPEG-4 video), Committee Draft, (1997)
13. Aspert, N., Santa-Cruz, D., and Ebrahimi, T.: MESH: Measuring Errors between Surfaces using the Hausdroff Distance, Proceedings of the IEEE International Conference in Multimedia and Expo (ICME 2002), Vol. 1, (2002) 705-708