A New Masking Scheme for DCT-Domain Watermarking of Three-Dimensional Triangle Meshes

Mohsen Ahourian, Jeonghee Jeon, and Yo-Sung Ho

1Department of Electrical Engineering; Islamic Azad University of Iran
    Majlesi Branch; Isfahan, P.O.Box 86315-11, Iran
2Geosoft Co., 6F Sinsa B/D 627-19 Sinsa-Dong, Kangnam-Gu Seoul, 135-985, Korea
3Kwangju Institute of Science and Technology (K-JIST)
    1 Oryong-dong Puk-gu, Kwangju, 500-712, Korea

Abstract

In this paper, we propose a new digital watermarking scheme for the three-dimensional (3-D) mesh model in the DCT domain. In order to insert and extract watermark signals, we generate triangle strips by traversing the 3-D mesh model. We embed the watermark signal into vertex positions of the 3-D model in the DCT domain. We define a new masking operation for the watermark signal according to the variation of DCT coefficients along the traversed strips so that we can change the embedding strength adaptively and reduce the visibility distortion of the data embedding. We test the proposed watermarking scheme by embedding a random binary sequence and applying different attacks, such as additive random noise, compression by MPEG-4 SNHC, and mesh simplification, which shows the robustness of the watermarking scheme.

Keywords: Watermarking, Data Embedding, Copyright Protection, Mesh Signal Processing

1. Introduction

The World Wide Web (WWW), various digital networks and digital multimedia processing technologies afford virtually unprecedented opportunities to pirate copyrighted materials. Consequently, the idea of digital watermarking has stimulated significant interest among artists and publishers [1]. Watermarking of three-dimensional (3-D) geometric models has received less attention from researchers because the watermarking scheme for image, video and audio cannot directly applied to 3-D data due to the non-regular structure of vertex positions in the 3-D space [2,3,4].

In this paper, we improve our previous data embedding scheme in the DCT domain [5] using a masking operation that adaptively changes the strength of data embedding and makes the embedded data less visible, while making more robust to compression and additive noise attacks.

Section 2 provides an overview of the watermark embedding, the proposed masking, and the watermark extraction operations. After we present experimental results of the proposed scheme in Section 3, we draw conclusions of this paper in Section 4.
2. The Proposed Watermark Scheme

As shown in Fig. 1, the watermarking scheme proposed in this paper consists of three main function blocks: creation of triangle strips, conversion of the traversed vertex coordinate to the DCT domain, and watermark insertion.

![Diagram of watermark insertion](image)

**Fig. 1. Watermark insertion**

2.1 Rendering 3-D Triangular Meshes

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 the bandwidth when we send it to our favorite rendering API (application programming interface). In order to create triangle strips from a triangle mesh, we use the algorithm described in Ref. [5]. The algorithm chooses a starting face and a direction (i.e. an edge) for each strip to walk along the strip. The algorithm extends the strip in the chosen direction until it reaches a triangle with no forward connections and then reverses the strip and extends it in the opposite direction.

2.2 Watermark Insertion

Each vector of vertex positions is transformed by 1-D DCT separately. The size of the segmented block can be adjusted by padding the last AC coefficient. 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 [6,7]. 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 signal. The string provided by the user is converted into the ASCII code, $a_j$, as

\[ a_j = \{a_{1,j}, a_{2,j}, a_{3,j}, a_{4,j}, a_{5,j}\} \]

In order to spread the digital signal into a wide bandwidth, we duplicate each bit $a_j$ by the chip rate $c$.

\[ b_i = a_{j,i}, \quad j \cdot c \leq i < (j+1) \cdot c \]

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

\[ b'_i = 2b_i - 1 \]

As shown in Fig. 1, we insert the watermark signal into the mid-frequency band of AC coefficients. New DCT coefficients $\hat{W}_{x,j}$ are derived from the current DCT coefficients $W_{x,j}$ by

\[ \hat{W}_{x,j} = W_{x,j} + \alpha_{x,j} \cdot b'_i \cdot p_i \]

\[ \hat{W}_{y,j} = W_{y,j} + \alpha_{y,j} \cdot b'_i \cdot p_i \]

\[ \hat{W}_{z,j} = W_{z,j} + \alpha_{z,j} \cdot b'_i \cdot p_i \]

where $b'_i$ is the watermark signals, $p_i$ is a pseudorandom number sequence generated by the watermark key. $(\alpha_{1,j}, \alpha_{2,j}, \alpha_{3,j})$ are masking factors which are derived from a fixed modulation amplitude $\alpha$. At first for each strip we calculate $D_i$, the summation of the $N_g$ middle DCT coefficients of strip $i$

\[ D_i = \sum_{j=1}^{i+N_g-1} |W_{x,j}| + \sum_{j=1}^{i+N_g-1} |W_{y,j}| + \sum_{j=1}^{i+N_g-1} |W_{z,j}| \]

Then we derive the three embedding amplitude $(\alpha_{1,j}, \alpha_{2,j}, \alpha_{3,j})$ from the main modulation amplitude $\alpha$ using the Eq. (8) and Eq.(9) and Eq.(10) for the three coordinates.

\[ \alpha_{1,j} = \alpha \cdot \frac{\sum_{j=1}^{i+N_g-1} |W_{x,j}|}{D_i} \]

\[ \alpha_{2,j} = \alpha \cdot \frac{\sum_{j=1}^{i+N_g-1} |W_{y,j}|}{D_i} \]

\[ \alpha_{3,j} = \alpha \cdot \frac{\sum_{j=1}^{i+N_g-1} |W_{z,j}|}{D_i} \]
2.3 Watermark Extraction

Fig. 2 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

\[ b' = (\hat{W}_{x_i} - W_{x_i}) \cdot p_i \]  

(11)

The extracted \( b' \) is changed to \( b \) by Eq. (3) and \( b \) is converted to \( a \) by Eq. (2). After \( a \) is converted to a string by the ASCII code, we can assert the ownership by the extracted string.

![Watermark extraction diagram](image)

Fig. 2. Watermark extraction

3. Experimental Results

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. 3(a), Fig. 3(b), and Fig. 3(c).

As Fig. 3 shows that watermarks are located in the areas with the mid-frequency band, which are 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. Watermarking Parameters are selected as follow:

- **Modulation amplitude**: Basically, the modulation amplitude \( \alpha \) 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

![3D models](image)

(a) Beethoven model and RMS distance representation

(b) Horse model and RMS distance representation

(c) Bunny model and RMS distance representation
Fig. 3. The original model and visual representation of RMS distance between the original and watermarked 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.

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. 4. The models embedded with the modulation amplitude $\alpha$ in Tables are attacked by additive random noise, geometry compression by the MPEG-4 SNHC standard, affine transform, and multiple watermarking.

As the additive random noise attack, we add noises to vertex coordinates of the watermarked model with a uniform random noise. In Fig. 4 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 seventeen, 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 [9]. 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 in Table 2 represents the number of strings recovered completely.

For the affine transformation, we translate, scale, rotate, and shear the watermarked model. The results are shown in Table 3. The multiple watermarking attacks have two different types. The first type is a consecutive attack to the first column of multiple watermarking in Table 4. 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 watermarked 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, it is clear that the inserted watermark string 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.

**Table 1. Resiliency against additive random noise attack**

<table>
<thead>
<tr>
<th>Models (Number of inserted string)</th>
<th>$\alpha$ (Alpha)</th>
<th>RMS</th>
<th>Additive Random Noise% Number of Recovered Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beethoven (21)</td>
<td>3.05</td>
<td>0.065</td>
<td>0.6</td>
</tr>
<tr>
<td>Horse (24)</td>
<td>0.0005</td>
<td>0.070</td>
<td>17</td>
</tr>
<tr>
<td>Bunny (23)</td>
<td>0.0006</td>
<td>0.086</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 2. Resiliency against geometry compression attack**

<table>
<thead>
<tr>
<th>Models (Number of inserted string)</th>
<th>$\alpha$ (Alpha)</th>
<th>RMS</th>
<th>Geometry Compression (96 bits) Number of Recovered Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beethoven (21)</td>
<td>3.05</td>
<td>0.065</td>
<td>15.8</td>
</tr>
<tr>
<td>Horse (24)</td>
<td>0.0005</td>
<td>0.070</td>
<td>12.3</td>
</tr>
<tr>
<td>Bunny (23)</td>
<td>0.0006</td>
<td>0.086</td>
<td>11.9</td>
</tr>
</tbody>
</table>

**Table 3. Resiliency against affine transform attacks**

<table>
<thead>
<tr>
<th>Models (Number of inserted string)</th>
<th>$\alpha$ (Alpha)</th>
<th>RMS</th>
<th>Number of Recovered Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beethoven (21)</td>
<td>3.05</td>
<td>0.065</td>
<td>7</td>
</tr>
<tr>
<td>Horse (24)</td>
<td>0.0005</td>
<td>0.070</td>
<td>9</td>
</tr>
<tr>
<td>Bunny (23)</td>
<td>0.0006</td>
<td>0.086</td>
<td>7</td>
</tr>
</tbody>
</table>

**Table 4. Resiliency against multiple attacks**

<table>
<thead>
<tr>
<th>Models (Number of inserted string)</th>
<th>$\alpha$ (Alpha)</th>
<th>RMS</th>
<th>Number of Recovered Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beethoven (21)</td>
<td>3.05</td>
<td>0.065</td>
<td>C+N: All 2</td>
</tr>
<tr>
<td>Horse (24)</td>
<td>0.0005</td>
<td>0.070</td>
<td>N-A: All 1</td>
</tr>
</tbody>
</table>

Comments: The table does not show the results for the multiple attacks.
4 Conclusions

In this paper, we have developed a new masking scheme for watermarking of 3-D triangle meshes in the DCT domain. The masking method adapts the strength of data embedding in the three coordinates according to their DCT coefficient amplitude. Using the RMS distance, we have shown that the masking for watermark insertion is effective for imperceptibility of the watermarked 3-D model and we have lower value of error compared to the similar watermarking method without masking [5]. Also the results show that addition of masking to the watermark embedding in the DCT domain slightly improves the system resilience against various types of 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, in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project, and in part by the Azad University research grant.

References
