ABSTRACT

We proposed 3D polygonal meshes watermarking using normal vector distribution to robust against various attacks. The norm vector distribution of mesh is a consistent factor that is robust against various attacks. After calculating these distributions of 6 patches respectively, the same watermark bits are embedded into the same ordered bins in the distributions of 6 patches. The experiment shows that the watermark is imperceptible and robust against various attacks.

1. INTRODUCTION

Digital media, such as images, audio, and video, can be readily manipulated, reproduced, and distributed over information networks. Therefore, a lot of research has been carried out to protect the copyright of digital media, and digital watermarking is one such copyright protection technique. Recently, 3D geometric models, such as 3D geometric CAD data, MPEG-4, and VRML, have been receiving a lot of attention, as such, various 3D watermarking algorithms have also been proposed to protect the copyright of 3D geometric data. Ohbuchi et al. proposed an algorithm that adds a watermark to a 3D polygonal mesh in the mesh spectral domain [1]. However, this algorithm is not robust against attacks that alter the connectivity of meshes, such as mesh simplification and remeshing. Beneden et al. also proposed an algorithm that adds a watermark by modifying the normal distribution of the model geometry [2]. Although this algorithm is robust against the randomization of points, mesh altering, and polygon simplification, it is not robust against cropping attacks, as the normal distribution is calculated for the entire model. Praun et al. proposed an algorithm that provides a scheme for constructing a set of scalar basis functions over the mesh vertices, then adapts the spread-spectrum principle [3]. However, this algorithm requires a complex resampling process of a suspect mesh in order to obtain a mesh with the same geometry, yet with a given connectivity.

Accordingly, the current study presents a new robust watermarking algorithm that is resistant to various attacks. The proposed algorithm is robust to mesh connectivity altering, plus, it does not require any resampling process for extracting the watermark. After calculating the respective EGI distributions for 6 different patches, bins for embedding the watermark, are selected according to the EGI distributions. Thereafter, all the vertices are moved to an optimum position so that the watermark can be embedded into bins where the vertices connecting the meshes are all mapped. Experimental results verify that the proposed algorithm is imperceptible and robust against geometrical and topological attacks.

2. THE PROPOSED 3D WATERMARKING ALGORITHM

A block diagram of the proposed 3D watermark embedding algorithm is shown in Fig. 1. The center points \( I \) of the patches and order information of the patch EGIs are then needed to extract the watermark.
2.1 Watermark embedding

2.1.1 Patch classification

The 3D mesh model is divided into 6 patches using a distance measure. By embedding the same watermark into each patch, the proposed algorithm is robust against partial geometric deformation, such as cropping. Above all, the initial center points \( I \) are specified as the points with the maximum distance in the direction of 6 unit vectors, \( \hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}} \), from the origin and considered as the respective center point for each of the 6 patches. Then, all the vertices are clustered into one patch \( iP \) as follows:

\[
P_j = \{v : d(v, I_j) < d(v, I_i), \text{ all } j \neq i, 1 \leq j \leq 6\}
\]

where \( d(v, I) = \| v - I \|^2 \) which has the minimum distance among the initial center points \( I \). After calculating the center of the clustered vertices for each patch, \( I \) are updated to the new center point. Then, the clustering and updating process is iterated until the distortion measure \( D_m \) satisfies the condition, \( (D_{m-1} - D_m) \leq th \), where \( m \) is the number of iterations, \( th \) is the threshold value, and \( D_m \) is expressed as

\[
D_m = \sum_{i=1}^{6} \sum_{v \in iP} \| I_i - v \|^2 .
\]

2.1.2 EGI per patch

EGI (Extended Gaussian Image) of a 3D model is an orientation histogram that records the variation of surface area with surface orientation [2],[5]. In the current paper, the unit sphere is divided into 240 surfaces that have the same area. The respective patch EGI\( jPE \)s, \( PE = \{PE_1, PE_2, \cdots PE_6\} \) are obtained by mapping the mesh normal vectors in \( P = \{P_1, P_2, \cdots P_6\} \) into the surface that has the closest direction among the 240 surfaces. To calculate a normal vector with a consistent direction, the vector of the vertex at the patch center point

\[
\mathbf{v}_i = \frac{\mathbf{v}_1 - \mathbf{I}_i \cdot \mathbf{v}_2 \times (\mathbf{I}_i \cdot \mathbf{v}_2) / \|\mathbf{I}_i \cdot \mathbf{v}_2 \|^2}{2} .
\]

The normal vector for a mesh surface includes \( \mathbf{n}_i \) and \( -\mathbf{n}_i \) according to two directions, inward and outward. Between these two vectors, the normal vector is determined as the one with the higher value of the dot products of \( \mathbf{m} \), where \( \mathbf{m} \) is the mean vector of \( \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \). The normal vector \( \mathbf{n} \) for each patch is mapped into bins \( iB \) within a certain range as follows:

\[
B_i = \{ \mathbf{v} : \| \overline{BC}_i \cdot \mathbf{n} \| < \| \overline{BC}_i \cdot \mathbf{n} \| \text{ all } j \neq i, 1 \leq j \leq 240 \} .
\]

Half the angle between the \( \overline{BC} \)'s of the neighborhood bins, \( A_m \), is the maximum value of \( \overline{BC} \cdot \mathbf{n} \), if \( \mathbf{n} \) is mapped into \( B_i \). \( \mathbf{n} \) around \( A_m \) is excluded from the EGI, because it can be mapped into a different bin after being attacked. Thus, only \( \mathbf{n} \) within a certain range is selected as follows:

\[
0 \leq \cos^{-1}(\overline{BC} \cdot \mathbf{n}) \leq A_{max}, A_{max} < A_m
\]

The length of \( B_i \) is the sum of area for all the meshes that are mapped into \( B_i \). There are 240 bins \( B_j = \{B_{ij} \mid 0 \leq i < 240\} \) in the \( j \)-th patch EGI \( PE_j \).
2.1.2 Selection of position for watermark embedding

In a patch EGI, the $B_\text{s}$ are arranged in a descending order according to their length. Then the $B_\text{s}$ with a large length are selected as the locations for the watermark embedding. The watermark, a 1-bit random sequence, is embedded in the ordered $B_\text{s}$ of PE. The $j$-th watermark $w_j$ ($1 \leq j \leq N$) is embedded into the $j$-th ordered $B_y$ s ($1 \leq i \leq 6$) in each PE, as shown in Fig. 3. The extraction of the watermark requires the ordered information of PE.

2.1.3 Embedding of PE watermark

All the normal vectors in the selected bin are changed according to the watermark bit. To change a normal vector, the position of the vertex needs to be changed. Yet, if the position of a vertex is changed, the normal vectors of all the meshes connected to this vertex will also be changed. Thus, it is important to identify the position of a vertex, as the normal vectors of all the meshes connected to this vertex will be changed according to the watermark bit of the bin that they are mapped into. The optimum position is identified for all the vertices that satisfy the cost measure for the condition of watermark embedding within the search region. To be invisible, the region for changing the position of a vertex must be below the value of each coordinate in all the vertices connected to the current vertex. Thus, the search regions for each coordinate of the current vertex $v = (x, y, z)$ are $x \pm \Delta x$, $y \pm \Delta y$ and $z \pm \Delta z$; $\Delta x = 0.5 \times \min | x - v_x |$, $\Delta y = 0.5 \times \min | y - v_y |$, $\Delta z = 0.5 \times \min | z - v_z |$. $v_t$ is the $t$-th coordinate value of vertex which connected in $v$.

In the current paper, a step algorithm is used to identify the optimum vertex within the search region. In the 1-th step, 27 positions for the vertex

$$v = \{(x, y, z)| x \in [x - \Delta x, x + \Delta x], y \in [y - \Delta y, y + \Delta y], z \in [z - \Delta z, z + \Delta z]\}$$

are selected for the initial search. The current vertex $v$ is then updated to position $v'$

$$v' = \arg \min_v \{ \cos t(v)|v = [x, y, z]\}$$

thereby minimizing the cost function $\cos t(v)$

$$\cos t(v) = \sum_{i \in S(v)} a_i | A_{w_i} - A_i |$$

where $S(v)$ indicates the mesh surface connected to $v$ and $A_i$ is the angle between the norm vector $\vec{n}_i$ of the $i$-th mesh surface and the $\overline{BC}_i$ of bin $B_i$ into which this vector is mapped among its patch EGI. If the $\overline{BC}_i$ is included in the bins for watermark embedding, $a_i$ is 1, otherwise, $a_i$ is 0. Plus, if the watermark bit of $B_i$ is 1, $A_{w_i}$ is 0, otherwise, $A_{w_i}$ is $A_{max}$. Then, $\Delta x$, $\Delta y$, and $\Delta z$ are decreased by half. In the next step, the vertex $v'$ is updated through the above process, which is performed in 3 iterations.

2.2 Extraction

The watermark can still be extracted from a watermarked model that has been attacked by connectivity altering, such as remeshing and simplification. First, the normal vector and its EGI distribution are calculated after dividing the original model into 6 patches based on the known center point $I$ of each patch. Using the location information obtained from the EGI distribution of each patch with watermark embedding, the watermark can be extracted based on the average difference of the angle between all the normal vectors mapped into each bin and the bin center point. The watermark $w_{ij}$ of the $B_{ij}$ bin with a $j$-th length for the $i$-th patch is extracted as 1, if $0 \leq A_{ij} \leq A_{th}$, otherwise, it is extracted as 0. $A_{ij}$ is represented as $A_{ij} = \frac{1}{N} \sum_{i=0}^{N} \cos^{-1}[\overline{BC}_i, \vec{n}_j]$ where $N$ equals the number of all the mesh normal vectors mapped into the $B_{ij}$ bin. A watermark decision of 1 bit is performed based on the $w_{ij}$ obtained from the 6 patches. Namely, $w_j$ is expressed as $w_j = \text{INT}(\frac{1}{6} \sum_{i=1}^{6} w_{ij} + 0.5)$.

3. EXPERIMENTAL RESULTS

The 3D VRML data of the Stanford bunny model (35,947 vertices and 69,451 faces) was used to evaluate the robustness of the proposed algorithm against geometrical and topological attacks. The experiment used a 1-bit watermark with a 50 length $W_{N=50}$ generated by a Gaussian random sequence. Therefore, each watermark was embedded into 50 bins per patch (total 300 bins in a model). The original model and watermark-embedded model are shown in Figs. 4 (a) and (b), clearly demonstrating the invisibility of the watermark.

To evaluate the robustness of the proposed algorithm, the watermark-embedded model was attacked by mesh
simplification, cropping, and additive random noise. The results are shown in Table 1, where the number of bin bit errors indicates the number of bins with a bit error among the 300 watermarked bins of all patches. Although bit errors due to attacks did occur in some watermarked bins, the watermark still remained due to the embedding of the watermark in each patch. The watermark-embedded model was attacked with remeshing and simplification using MeshToSS [4]. However, the watermark remained until the model was simplified to 30% of the original model vertex. Randomization of a vertex was performed, where vertex $v$ sampled randomly was added to $\nu \alpha \times \text{uniform}()$. The modulation factor $\alpha$ was 0.008 and $\text{uniform}()$ was a uniformly random function of $[-0.5, 0.5]$. In table, 50% and 100% of random noise indicate the percentage of the number of vertices that were randomly sampled to add random noise.

In the case of cropping, although there were some patches that had no vertices, all watermarks could be extracted from the other patches. Plus, in a cropped model with 52.2% of the vertices, 99% of the watermark still remained. The models attacked by simplification, random noise, and cropping are shown in Figs. 4 (c)–(f).

### 4. CONCLUSIONS

We proposed the 3D watermarking algorithm that embed watermark into normal vector distributions of each patches. This algorithm is robust against various attacks and don't need the original model to extract watermark as well as the resampling process.

### ACKNOWLEDGEMENTS

This work was supported by Institute of Information Technology Assessment of Ministry of Information and Communication in Republic of Korea.

### REFERENCES


<table>
<thead>
<tr>
<th>Attacks</th>
<th>Total bit error</th>
<th>Error/bin</th>
<th>watermark bit error</th>
<th>Error/watermark bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplify</td>
<td>28237vertex (78.5%)</td>
<td>11</td>
<td>3.66%</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>18769vertex (52.2%)</td>
<td>23</td>
<td>7.66%</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>12157vertex (33.8%)</td>
<td>44</td>
<td>14.66%</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>6218vertex (17.3%)</td>
<td>87</td>
<td>29%</td>
<td>8</td>
</tr>
<tr>
<td>Random noise 50%</td>
<td>20</td>
<td>6.6%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Random noise 100%</td>
<td>56</td>
<td>18.67%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cropping</td>
<td>13</td>
<td>5.2%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Simplify (52.2%) + Cropping</td>
<td>38</td>
<td>15.2%</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>