Research Article

Hamming Code Based Watermarking Scheme for 3D Model Verification

Jen-Tse Wang,1 Yi-Ching Chang,2 Chun-Yuan Yu,2,3 and Shyr-Shen Yu2

1 Department of Information Management, Hsiuping University of Science and Technology, 11 Gongye Road, Dali District, Taichung City 412, Taiwan
2 Department of Computer Science and Engineering, National Chung-Hsing University, 250 Kuo-Kuang Road, Taichung 402, Taiwan
3 Department of Digital Living Innovation, Nan Kai University of Technology, 568 Chung-Cheng Road, TsaoTun 542, Nantou County, Taiwan

Correspondence should be addressed to Shyr-Shen Yu; pyu@nchu.edu.tw

Received 25 February 2014; Accepted 13 April 2014; Published 30 April 2014

Copyright © 2014 Jen-Tse Wang 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.

Due to the explosive growth of the Internet and maturing of 3D hardware techniques, protecting 3D objects becomes a more and more important issue. In this paper, a public hamming code based fragile watermarking technique is proposed for 3D objects verification. An adaptive watermark is generated from each cover model by using the hamming code technique. A simple least significant bit (LSB) substitution technique is employed for watermark embedding. In the extraction stage, the hamming code based watermark can be verified by using the hamming code checking without embedding any verification information. Experimental results shows that 100% vertices of the cover model can be watermarked, extracted, and verified. It also shows that the proposed method can improve security and achieve low distortion of stego object.

1. Introduction

The digital era already came to our lives thanks to the interestingly sophisticated hardware and software technologies. Especially the internet, it made our life more convenient and even changed our lifestyles. Traditional publishers, for instance, gradually translated the products to online-digitalized stores. This allowed any writer's work to be searched and discovered once they upload them to these online stores by publishers. Besides this, various multimedia, both audio and visual, also take part in this digital era.

But everything has its pros and cons. While the internet makes our lives easier, it also presents a risk towards the copyright or other personal information. All of these lead to piracies, tampering, and stealing of data. In order to fix these leaks and improve the data's security, many experts and scholars endeavor in fields known as information hiding research [1–6], and digital signature [7–10], dedicated to solve these problems.

According to the difference of application purpose, information hiding techniques can be divided into data hiding and watermarking. The former has another common terminology called steganography. The latter, watermarking [11–13], is modeled to embed meaningful text or images with copyright information into cover model to achieve copyright protection or image authentication [14, 15].

Watermarking technique [11–13] can be divided into visible watermarking and invisible watermarking by visualization, or can be separated as blind, semiblind, and nonblind by extraction strategies, or can be distinguished from special domain [16–19] and frequency domain [19–24] by embedding space, or can be differentiated between robust watermarking and fragile watermarking according to application purpose. The main goal of robust watermarking is to make the embedded watermarks remain detectable after being attacked. In contrast, the requirements of fragile watermarking are to verify the slightest unauthorized alteration and be able to locate the changed regions.

2. Background and Related Works

Among the different media types, watermarking of 3D objects is comparatively difficult inherently. Initially, Ohbuchi et al. [25–28] proposed a large variety of techniques for embedding
data into 3D polygonal models. In [26], Ohbuchi et al. proposed a nonblind algorithm that works in the transformation domain. It embeds a watermark into a 3D polygonal model by deforming the low-frequency components of the shape by using the mesh spectral analysis. Cayre and Macq [29] described a blind data hiding scheme in the spatial domain. The key idea is to consider a triangle as a two-state geometrical object. Each triangle that can be embedded is called an admissible triangle. In [30], Alfacs et al. presented a way to extend the robustness of blind and robust 3D watermarking schemes to the difficult cropping attack. C.-M. Wang and P.-C. Wang [31, 32] propose a novel scheme for digital steganography in the spatial domain. It employs a principal component analysis (PCA), resulting in a blind approach. Kanai et al. [33] were the first to apply a transform domain watermarking approach on 3D polygonal model. It is a nonblind algorithm, which first decomposes an original polygonal model by applying wavelet transform several times. This algorithm embeds the watermarks into the wavelet coefficient vectors and it is robust against affine transformation and random noise attacks.

The existing fragile watermarking algorithms that are designed for 3D models are relatively few. In fragile watermarking, the embedded watermark will be modified when the stego object is altered. Therefore, the fragile watermark can be used to verify a stego object. Yeo and Yeung [19] firstly proposed a fragile watermarking algorithm for authenticating 3D polygonal meshes. Two indices were calculated for every vertex: the location index and the value index. According to the coordinates of it and its neighboring vertices, the location index was calculated by a hash function and the value index is calculated by another hash function for each vertex. Then they slightly perturbed every vertex to make these two hash functions equal. Their scheme is both blind and fragile, but there arise two problems: the causality problem and the convergence problem. The causality problem is that the location index of a former processed vertex will be changed if any coordinate of the model is modified. In [36], the coordinates of each vertex of the cover model are transformed to spherical coordinates. The quantization index modulation technique is employed to embed the watermark into the \( r \) coordinates by using the center of gravity of the cover model. The drawback of [36] is that the reference point would be changed if any coordinate of the model is modified. In [37], Wang et al. proposed a multiple reference point embedding process to overcome the drawback of [36].

In this paper, a hamming code based fragile watermarking scheme is performed on 3D polygonal meshes in spatial domain. The proposed method can provide the capability of accurately verifying, locating the tampered region to protect the integrity of 3D objects and there has no embedding holes problem.

Chou and Tseng [34] propose a blind fragile watermarking technique which can embed watermark into at most half the number of total points. In [35], a public fragile marking scheme based on the sensitivity of vertex geometry for 3D model authentication is proposed. In this paper, Chou and Tseng propose a multifunction vertex embedding method and an adjusting-vertex method to overcome the causality problem and the convergence problem. The average distortion of the marked models is under user control with proper key value setting, and this scheme can detect and locate all unauthorized modifications. But this scheme still has the embedding holes problem.

In [36], the coordinates of each vertex of the cover model are transformed to spherical coordinates. The quantization index modulation technique is employed to embed the watermark into the \( r \) coordinates by using the center of gravity of the cover model. The drawback of [36] is that the reference point would be changed if any coordinate of the model is modified. In [37], Wang et al. proposed a multiple reference point embedding process to overcome the drawback of [36].

In this paper, a hamming code based fragile watermarking scheme is described in Section 3. Experimental results and discussion are presented in Section 4. Finally, conclusions are provided in Section 5.

### 3. Proposed Hamming Code Based Fragile Watermark Algorithm

In this paper, a hamming code based fragile watermark scheme for 3D objects is proposed in spatial domain. It is public and fragile. The logistic map [38] is utilized to produce the intermingled embedding order of vertices. There does not need embedding any verification information for tampering detection in the proposed method. An adaptive watermark is generated from each cover model by using the hamming code technique [39–42]. Instead of the comparison of the hash function, the hamming code based watermark can be verified by using the hamming code checking in the extraction stage.
Hamming code is named after Richard Hamming and is a linear forward error correction code which works by adding parity check bits at the output data stream. Because of the simplicity of hamming codes, they are widely used in computer memory RAM [42].

The most widely used hamming code is (7, 4) hamming code, which encodes four data bits ($D_1, D_2, D_3, D_4$) into seven bits by adding three parity check bits ($P_1, P_2, P_3$). Each parity check bit is created by its associated data bits. Table 1 illustrates the relationship between parity check bits and associated data bits.

As shown in Table 1, the parity check bits can be created by using the associated data bits, and the results are shown in Table 2, where “*” denotes an exclusive or (XOR) operation. The hamming code detects errors by ensuring each parity check bit and its corresponding data bits achieve the even parity. This detection procedure is called parity checking.

In mathematical point of view, hamming code is a class of binary linear block code. Regardless the form of $H$ and $G$, $H$ and $G$ for linear block codes must satisfy $HG^T = 0$, where 0 is an all-zeros matrix due to $[7, 4, 3] = [n, k, q] = [2^m - 1, 2^m - m - 1, m]$. There is a code with $m$ parity bits and $2^m - m - 1$ data bits for each integer, where $m$ must satisfy $m \geq 2$. The parity check matrix $H$ of a hamming code is constructed by listing all columns of length $m$ that are pairwise independent. Given a generator matrix $G$ of the code, where $G$ is that so-called systematic form. It means that the leftmost or rightmost $k$ columns of $G$ form a $k \times k$ identity matrix $I_k$. For the detailed formulas, one is referred to [43].

In this paper, the parity check matrix $H$ and the code generator matrix $G$ are:

\[
H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}_{3 \times 7},
\]

\[
G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 7}.
\]

According to the description above, (1) are rewritten as follows:

\[
H = [I_{n-k}, P^T],
\]

\[
G = [P, I_k].
\]

After calculate (3), the matrices $H$ and $G$ used in this paper satisfy the theory of hamming code.

\[
HG^T = [I_{n-k}, P^T][P^T \oplus P^T] = P^T \oplus P^T = 0.
\]

Adding the parity bits allows the receiver checking and correcting if there are errors. The (7, 4) hamming codes can detect up to two-bit errors, and correct single-bit error, but cannot identify the error type. Therefore, hamming codes are an example of perfect codes which exactly match the theoretical upper bound on the number of distinct code words for a given number of bits and ability to correct errors.

Figure 1 illustrates the outline of this proposed scheme. In this paper, the hamming code technology is adopted as the premise of our 3D model fragile watermarking scheme. A simple LSB substitution technique is employed for watermark embedding. Moreover, the hamming code based watermark can be verified by using the hamming code checking without embedding any verification information in the extraction stage. In summary, the proposed scheme has several advantages: (1) it is public and blind; (2) the verification rate achieves 100% (watermarked vertices/total vertices); (3) the vertex based localization of unauthorized modification can be achieved; (4) the key size used in the proposed scheme is small; (5) Finally, the proposed method is immune to the causality, convergence, and embedding holes problems.

3.1. The Embedding Process. Without loss of generality, let each 3D object of interest be described by a sequence of coordinates $P_i, P_{i+1}, P_{i+2}, \ldots, P_{N-1}, 0 \leq i \leq N-1$. Moreover, each vertex of the sequence is represented by its 3D coordinates $(x_i, y_i, z_i)$. The embedding algorithm takes two inputs: the cover model and the secret key to produce the stego model. As shown in Figure 2, it embeds a watermark according to the following four steps:

**Step 1** (normalization). The coordinates of each point are normalized into the range 0 to 1. Consider

\[
\begin{align*}
q_i^x &= \frac{q_{i, \text{original}}^x - \text{Min}(X, \text{original})}{\text{Max}(X, \text{original}) - \text{Min}(X, \text{original})} \\
q_i^y &= \frac{q_{i, \text{original}}^y - \text{Min}(Y, \text{original})}{\text{Max}(Y, \text{original}) - \text{Min}(Y, \text{original})} \\
q_i^z &= \frac{q_{i, \text{original}}^z - \text{Min}(Z, \text{original})}{\text{Max}(Z, \text{original}) - \text{Min}(Z, \text{original})},
\end{align*}
\]

where $x, y, z$ denote the $X, Y, Z$ axis; $i$ is the vertex $0 \leq i \leq N - 1$; $\text{Max}(X, \text{original})$ denotes the maximum value of the cover model at $X$ axis, and so as $\text{Max}(Y, \text{original})$ and $\text{Max}(Z, \text{original})$, $\text{Min}(X, \text{original})$ denotes the minimum value
Table 2: Create parity check bits.

<table>
<thead>
<tr>
<th>Parity check bits</th>
<th>Associated data bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1 )</td>
<td>( D_1 \oplus D_2 \oplus D_4 )</td>
</tr>
<tr>
<td>( P_2 )</td>
<td>( D_1 \oplus D_3 \oplus D_4 )</td>
</tr>
<tr>
<td>( P_3 )</td>
<td>( D_2 \oplus D_3 \oplus D_4 )</td>
</tr>
</tbody>
</table>

of the cover model at \( X \) axis, and so as \( \text{Min} (Y, \text{original}) \) and \( \text{Min} (Z, \text{original}) \).

Step 2 (extraction of data bits). Without loss of generality, suppose \( q^r_i \) is the point to be embedded, where \( r \) denotes the axis used and \( 0 \leq i \leq N - 1 \). The 4 bits, which are the 4th least significant bit to 7th least significant bit of point \( q^r_i \), are considered as an input data bits \( D_1, D_2, D_3, \) and \( D_4 \) of hamming procedure.

Step 3 (generation of parity check bits). Then, the hamming code procedure is performed to generate the parity check bits. As shown in Table 1, the three parity check bits \( P_1, P_2, \) and \( P_3 \) can be generated from the hamming procedure. Moreover, these three parity check bits \( P_1, P_2, \) and \( P_3 \) are regarded as watermark \( w^r_i \) in this paper.

Step 4 (watermark embedding). For security reasons, a secret key \( K \) is employed to generate a random sequence of integers. They represent the index orders for embedding watermark to corresponding point. The parity bits are then embedded into the 3 least significant bits (LSB) of point \( q^r_i \) by user least significant bit substitution.

Note that every numeric of random sequence will never duplicate and only cooperate to one vertex of the cover model. Every vertex has its own unique watermark. If any part of the stego model is modified, then the tampered region will be verified in extraction stage.

3.2. The Extraction Process. The watermark extraction process is performed to extract the embedded watermark and verify the integrity of the stego model. The extraction process takes two inputs: the attacked stego model and the secret key \( K \) to extract the embedded watermark. As shown in Figure 3, it extracts the watermark according to the following five steps:

Step 1 (normalization). The coordinates of each point are normalized within the range 0 to 1.

Step 2 (extraction of data bits). A secret key \( K \) is employed to generate a random sequence of integers. The random number generator is the same as that used in embedding process. The 4 bits which are the 4th least significant bit to 7th least significant bit of point \( q^r_i \) in stego model are extracted and considered as data bits for hamming code.

Step 3 (calculation of parity check bits). After hamming code encoding, the three output parity bits are regarded as watermark \( w^r_i \).

Step 4 (extraction of watermark). The embedded watermark \( h^r_i \) can be extracted from the 3 least significant bits of point \( q^r_i \) in stego model.

Step 5 (verification). The verification is achieved when the watermark \( w^r_i \) is equal to the embedded watermark \( h^r_i \).

\[
R \left( w^r_i \right) = \begin{cases} 
\text{no tempered,} & w^r_i = h^r_i \\
\text{modified,} & w^r_i \neq h^r_i.
\end{cases}
\] (5)

Note that this process is a public blind fragile watermark scheme which does not need original model and original watermark for verify forgery.

4. Experimental Results and Discussions

The proposed public hamming code based fragile watermarking method was implemented using Microsoft Visual C++ programming language.

A series of experiments were conducted to test the performance of the proposed fragile watermarking method. We shall present a set of visualization results that can demonstrate the performance of the proposed method and validate the feasibility of our algorithms.

Table 3 shows the cover models of experiments in this paper and the visual effect of the stego models. No visual distortion can be perceived between the cover models and stego models. Moreover, the precision level 10-10 is used for these experiments. No visual distortion can be perceived between the cover models and stego models.

The drawback of spatial domain is that it has poor resistance to distortion compression. There are some approaches for measure distortion, such as signal-to-noise (SNR), the Haussdorff distance, Laplacian [44], and root mean square (RMS) ratio [31, 32]. SNR used only on the still images. The Hausdorff distance and Laplacian were used to evaluate the distortion of 3D polygonal models. But for a zoomed model, to measure RMS of distortion is a better choice.

Here, we used RMS by formula (7) and RMS ratio by formula (6) to present the comparison of distortion control. The RMS ratio consists of the RMS values over the diagonal
Table 3: The cover model used in this paper.

<table>
<thead>
<tr>
<th>Model (size)</th>
<th>Cover model</th>
<th>Stego model</th>
<th>Model (size)</th>
<th>Cover model</th>
<th>Stego model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angelfish (19724)</td>
<td><img src="image1" alt="Angelfish" /></td>
<td><img src="image2" alt="Angelfish" /></td>
<td>Teapot (28922)</td>
<td><img src="image3" alt="Teapot" /></td>
<td><img src="image4" alt="Teapot" /></td>
</tr>
<tr>
<td>Bunny (34837)</td>
<td><img src="image5" alt="Bunny" /></td>
<td><img src="image6" alt="Bunny" /></td>
<td>Venus (33591)</td>
<td><img src="image7" alt="Venus" /></td>
<td><img src="image8" alt="Venus" /></td>
</tr>
<tr>
<td>Lion (17352)</td>
<td><img src="image9" alt="Lion" /></td>
<td><img src="image10" alt="Lion" /></td>
<td>Wolf (30084)</td>
<td><img src="image11" alt="Wolf" /></td>
<td><img src="image12" alt="Wolf" /></td>
</tr>
</tbody>
</table>

Table 4: The comparison of RMS and RMS ratio of distortion control.

<table>
<thead>
<tr>
<th>Models</th>
<th>Methods</th>
<th>RMS</th>
<th>RMS ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anglefish</td>
<td>$3.74 \times 10^{-4}$</td>
<td>$3.75 \times 10^{-5}$</td>
<td>$5.17 \times 10^{-6}$</td>
</tr>
<tr>
<td>Bunny</td>
<td>$3.75 \times 10^{-4}$</td>
<td>$4.50 \times 10^{-5}$</td>
<td>$5.06 \times 10^{-6}$</td>
</tr>
<tr>
<td>Lion</td>
<td>$3.74 \times 10^{-4}$</td>
<td>$4.63 \times 10^{-5}$</td>
<td>$5.00 \times 10^{-6}$</td>
</tr>
<tr>
<td>Teapot</td>
<td>$3.73 \times 10^{-4}$</td>
<td>$4.53 \times 10^{-5}$</td>
<td>$5.00 \times 10^{-6}$</td>
</tr>
<tr>
<td>Venus</td>
<td>$3.74 \times 10^{-4}$</td>
<td>$4.18 \times 10^{-5}$</td>
<td>$5.07 \times 10^{-6}$</td>
</tr>
<tr>
<td>Wolf</td>
<td>$3.75 \times 10^{-4}$</td>
<td>$4.38 \times 10^{-5}$</td>
<td>$5.00 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

The length of the bounding volume for a 3D stego model. The small RMS ratios indicate insignificant positional changes during the watermark embedding. Consider

$$\text{RMS}_\text{Ratio} = \frac{\text{RMS}}{\text{Diagonal Length}},$$

where

$$\text{RMS} = \sum_{i=0}^{N-1} \left( (q^x_i - q^x_{i, \text{original}})^2 + (q^y_i - q^y_{i, \text{original}})^2 ight)^{1/2} (N)^{-1},$$

where $q^x_i$ denotes the $x$ coordinate of the point in stego model, so as $q^y_i$ and $q^z_i$, $0 \leq i \leq N - 1$,

$$\text{Diagonal Length} = \sqrt{X_{\text{size}}^2 + Y_{\text{size}}^2 + Z_{\text{size}}^2}.$$
Table 5: Comparison of previous methods and the proposed method.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No verification information</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>√</td>
</tr>
<tr>
<td>No causality problem</td>
<td>—</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>No convergence problem</td>
<td>—</td>
<td>—</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>No embedding holes</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>√</td>
</tr>
<tr>
<td>Verification rate</td>
<td>—</td>
<td>—</td>
<td>~42%</td>
<td>12%~16%</td>
<td>100%</td>
</tr>
<tr>
<td>Data hiding rate (bits/vertex)</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>&lt;1.5</td>
<td>&lt;1.5</td>
<td>3</td>
</tr>
<tr>
<td>Tampering detection</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Localization of tampered region</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Distortion control</td>
<td>—</td>
<td>—</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

The main purpose in this paper is to authenticate the integrity of 3D polygonal meshes in the spatial domain. The distortion rate can be controlled below $10^{-10}$. The verification rate achieves 100%. Consequently, every region of the 3D model can be embedded in the embedding stage and can be verified in the extraction stage.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Science Council, Taiwan, under the Grants NSC 101-2221-E-164-022 and NSC 102-2221-E-005-082.

References
