Article

Collusion Attack Resilient 3D Mesh Watermarking Based on Anti-Collusion Fingerprint Code

Jong-Uk Hou , In-Jae Yu and Heung-Kyu Lee *

School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea;
juheo@kaist.ac.kr (J.-U.H.); ijyu@mmc.kaist.ac.kr (I.-J.Y.)

* Correspondence: heunglee@kaist.ac.kr

Received: 1 June 2018; Accepted: 23 June 2018; Published: 26 June 2018

Abstract: Collusion attack is one of the techniques used for unauthorized removal of embedded marks. However, there are no algorithms that utilize the averaging-resilient fingerprint code into 3D mesh watermarking. In this paper, we propose a collusion resilient 3D mesh watermark based on mesh spectral analysis and the anti-collusion fingerprint code. We present two fingerprint schemes by exploiting the theoretically proven anti-collusion codes: the group-divisible partially balanced incomplete block design, and Tardos's fingerprint code. The proposed extraction scheme not only met the requirement for an anti-collusion code but also provided sufficient length of the payload for the fingerprint code. To minimize the detection error, we also modeled the response of the detector and herein present optimized thresholds for our method. Based on the experiments with public benchmarks, the proposed method outperformed conventional robust mesh watermarking against collusion attack, and provided robustness to the combination of a small amount of added noise and collusion attack.

Keywords: 3D mesh watermarking; collusion attack; multimedia security; fingerprinting

1. Introduction

Digital watermarking is the process of hiding digital information in a noise-tolerant signal such as multimedia data. The watermark can be used to determine authorship when a copyright dispute occurs, and can be used as a fingerprint to track a distribution path when a prototype in the hands of only a few people, is leaked. Digital watermarking technology have been applied to various kind of multimedia contents such as 3D mesh watermarking [1], and 2D and 3D video watermarking [2,3]. Collusion attack is one of the main techniques used for unauthorized removal of embedded marks [4]. Attackers can exploit different versions of the fingerprinted content to estimate the original content. Moreover, a simple averaging process of some different protected contents can be used to disable detection of the embedded pattern. To cope with collusion attacks, two categories of countermeasure techniques were previously researched. The first technique works by pre-warping at the signal-processing level, so that each copy of the multimedia content is varied in a slightly different way [5,6]. However, in the case that the method fails to protect against collusion attack, it cannot provide information useful for tracking the attackers. The second countermeasures are based on the averaging-resilient fingerprint, which theoretically guarantees robustness against several attackers [4,7–10]. In contrast to pre-warping-based methods, the anti-collusion-based-techniques work passively by providing explicit clues for tracking the attackers.

Recently, the demand for 3D mesh watermarking has increased due to the emergence of low-cost 3D printers. On the other hand, copyright issues will inevitably occur with the expansion of 3D model sharing platforms such as Thingiverse, Pinshape, and Sketchfab. In spite of various watermarking methods [1,11–13], mesh watermarking techniques resilient to collusion attack were

rarely considered [6,14]. Uccheddu et al. [6] proposed anti-collusion watermarking for 3D meshes by pre-warping, which changes the vertices ordering and the associated faces. However, there is no algorithm that utilizes the averaging-resilient fingerprint code into 3D mesh watermarking.

This paper is an extension of our previous work [15]. Our previous work proposed a mesh watermarking algorithm based on a mesh Laplacian matrix [16] with group-divisible partially balanced incomplete block design (GD-PBIBD). In this extension, we used two anti-collusion fingerprint code system: one is based on the GD-PBIBD [9], and another one is based on Tardos's fingerprint code [17], and the codes were changed as a form of unsigned random pattern to prevent the attenuation of embedded signal by collusion attack. By modeling the detector response of the proposed method, we theoretically determined a threshold to minimize the detection error. As a result of our tests, we experimentally demonstrate that the proposed method does not lose recipient information during collusion attack.

The rest of this paper is organized as follows. Section 2 is a brief introduction of the necessary background knowledge, and Section 3 explains the details of the proposed method. To demonstrate the performance of the proposed scheme, we tested our method with mesh benchmark sets, and report the results in Section 4. In Section 5, the conclusions are presented.

2. Backgrounds

2.1. Model for Collusion Attack

In this section, we will illustrate the frequent types of collusion attack. Collusion attack is a type of unauthorized removal of fingerprint when an adversary obtains several copies of a content marked with different fingerprints [4]. The adversary combines the fingerprinted copies of given contents to attenuate the power of fingerprints of the each copy. The most frequent type of collusion attack is linear collusion. The simplest example of this type of attack is averaging the copies. Figure 1 illustrate the process of linear collusion. Alice's original content x is distributed with fingerprints w_1 to w_n . Suppose Bob obtained the marked copies y_1 to y_n . Bob mix those copies by averaging and add noise z and acquired attacked content y . Attacked content y has reduced power of each fingerprints, the attacked contents y become similar to the original work x . Therefore, Bob may issue that the content y is his original content. As the number of copies increases, the power of each fingerprint becomes weaker, and harder to declare the original ownership.

If the marking algorithm spreads the fingerprint throughout the entire contents, cut-and-paste has a similar effect to averaging attack. Algorithm such as spread spectrum-based embedding in transform domain, mixing the part of copies by cut-and-paste also reduces the power of each fingerprints. For example, suppose Bob takes $1/n$ parts equally from each copies w_k , and paste them into new copy y . If the fingerprints are embedded in transformed domain such as frequency domains, the power of each fingerprints will be attenuated. Another type of collusion attack is non-linear collusion. The concept of this type of attack is taking maximum, minimum or median, rather than taking the average, of among the observed copies.

2.2. Related Work

There are two categories of countermeasure techniques were previously researched to cope with collusion attacks. The first technique works by pre-warping at the signal-processing level, so that each copy of the multimedia content is varied in a slightly different way [5,6]. The pre-warping process is effective to cope with collusion attack, because a simple averaging attack causes degradation of the quality of the protected contents. Colluders must then resort to more complex attacks that bring about the registration of the original contents prior to the attack itself [5]. However, some doubts exist about the real effectiveness of these techniques given that registering the pre-warping contents does not appear to be an impossible task [6]. Moreover, in the case that the method fails to protect against collusion attack, it cannot provide information useful for tracking the attackers.

Figure 1. Model for linear and non-linear collusion.

The second countermeasures are based on the averaging-resilient fingerprint, which theoretically guarantees robustness against several attackers [4,7–10]. In contrast to pre-warpage-based methods, the anti-collusion based techniques work passively by providing explicit clues for tracking the attackers. Boneh et al. [4] first defined the collusion attack for digital data, and they analyzed the characteristics of a collusion-attack resilient code. Based on the combinatorial theory, Stinson et al. [7] proposed a code system for an anti-collusion fingerprint code. Kang et al. [9] proposed a constructive algorithm for an anti-collusion fingerprint code, which is called group-divisible partially balanced incomplete block design (GD-PBIBD), based on group-divisible designs. Tardos has proposed a randomized fingerprinting code that is provably secure against collusion attack [17] which performs state-of-the-art performance. The papers listed in this paragraph only proposed a code system, so that they did not suggest any practical system or experimental results for multimedia data. For fingerprinting system based on the anti-collusion code, Trappe et al. [8] proposed an image fingerprinting system using balanced incomplete block design (BIBD). For video fingerprinting, Kang et al. [18] introduced a practical fingerprint scheme that adopted the anti-collusion fingerprint, which was called group-divisible partially balanced incomplete block design (GD-PBIBD). Maity et al. [19] proposed a collusion resilient optimized image watermarking scheme using genetic algorithms and multiband wavelets. Kuribayashi [20] suggested a simplified maximum a posteriori detector for binary fingerprinting code.

Recently, the demand for 3D mesh watermarking has increased due to the emergence of low-cost 3D printers and 3D model sharing platforms such as Thingiverse, Pinshape, and Sketchfab. In spite of various watermarking methods [1,11–13,21,22], mesh watermarking techniques resilient to collusion attack were rarely considered [6,14]. Uccheddu et al. [6] proposed anti-collusion watermarking for 3D meshes by pre-warpage, which changes the vertices ordering and the associated faces. However, there is no algorithm that utilizes the averaging-resilient fingerprint code into 3D mesh watermarking. As we found in the previous literature, most of the watermarking schemes based on anti-collusion codes were intended to protect image and video data. To the best of our knowledge, anti-collusion code-based 3D mesh watermarking has not been reported yet.

3. Proposed Method

The main assumption for designing the anti-collusion fingerprint codes is that the averaging attack of the anti-collusion code has the same effect as a bitwise AND operation of every bit position. Therefore, the embedded bit information has to be zero for the averaging attack when each bit from each colluded work is different. However, conventional robust mesh watermarking methods such as [11–13,23,24] did not consider this aspect. Moreover, anti-collusion fingerprint code requires relatively several payloads in many cases, so that the existing mesh watermarks are not suitable to carry anti-collusion fingerprint codes. (see Table 1). In contrast to the existing robust mesh watermarking, our proposed method has sufficient capacity to carry the anti-collusion fingerprint code. We will evaluate them in the experimental results section later.

Table 1. A summary of related robust mesh watermarking algorithms.

Algorithms	[11]	[12]	[23]	[13]	Ours
Robustness against collusion	yes	no	yes	no	yes
Bit capacity for fingerprint code	undefined	≤64 bit	≈64 bit	≤24 bit	≤64 bit

Unlike the image and video watermarking described in [8,18], each vertex of a 3D mesh model is non-uniformly sampled, so that the spatial structure of the mesh is not suitable for utilizing fingerprint codes. Alternatively, we can divide the mesh into disjoint bands in the frequency domain and embed a watermark in them. To perform the frequency analysis, we use a technique called *mesh spectral analysis* [25] that can be considered a principal component analysis of the shape. We now describe each step of watermark embedding and detection in detail.

3.1. Watermark Embedding Algorithm

The spectrum of a 3D mesh is computed from connectivity and coordinates of vertices of the mesh. Eigenvalue decomposition of a mesh Laplacian matrix is required to produce spectral coefficients. For the Laplacian matrix, we employed a mesh Laplacian matrix proposed by Bollobás [16], referred to as the Kirchhoff matrix. The Kirchhoff matrix $\mathbf{K}$ is defined as follows:

$$\mathbf{K} = \mathbf{D} - \mathbf{A}, \quad (1)$$

where $\mathbf{D}$ is a diagonal matrix of which an element is defined by a degree of each vertex, and $\mathbf{A}$ is an adjacent matrix of the mesh of which elements $a_{i,j}$ are equal to '1' if vertices i and j are adjacent, and otherwise '0'. A mesh $\mathbb{M}$ with m vertices produces a Kirchhoff matrix $\mathbf{K}$ of size $m \times m$, and its eigenvalue decomposition may be described as follows:

$$\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad (2)$$

where $\mathbf{\Lambda}$ is a diagonal matrix composed of the m eigenvalues λ_i in ascending order, and $\mathbf{U}$ is a unitary matrix composed of the m -dimensional normalized eigenvectors.

The Laplacian eigenvector basis is used to perform an orthogonal decomposition of a function. The spectra of the mesh are obtained by projection of the vector according to the Laplacian basis, as indicated in the following equation.

$$\mathbf{c} = \mathbf{U}^T \mathbf{x}, \quad (3)$$

where $\mathbf{c} = (c_1, c_2, \dots, c_m)^T$ is a spectral coefficient of input mesh, and $\mathbf{x}$ is a vector of the vertices of the mesh.

We generate a watermark pattern using a 'group-divisible partially balanced incomplete block design' (GD-PBIBD) [26]. Using a generator proposed by Kang et al. [9], the generated code set is defined as an $N \times R$ binary matrix $\mathbf{N}$ in which a column represents a unique fingerprint code for each user, and where N is the length of each fingerprint code. The generated fingerprint code set $\mathbf{N}$ for $R = s^{2(p-1)}$ recipients can protect cover work from n attackers ($s - 1$). Detailed information can be found in [26].

For the u -th recipient, we obtain u -th column vector $\mathbf{y}^u$ from the generated fingerprint code set $\mathbf{N}$. Each binary value '0, 1' from $\mathbf{y}^u$ is encoded into two symbol vectors $\mathbf{s}_0, \mathbf{s}_1$ with same length L , then we obtain a watermark sequence $\mathbf{w}$ with length $N \cdot L$. We use zero value vectors for symbol vector $\mathbf{s}_0$. For symbol vector $\mathbf{s}_1$, we use a pseudo-random sequences with positive values generated by a watermark key. We embed the generated pattern $\mathbf{w}$ by changing the amplitude of the spectral

coefficient vector $\mathbf{c}$. First, we determine a frequency band for watermark embedding by selecting a start index i_0 of the coefficient vector. The watermarking process is computed using the following equation:

$$c'_{(i+i_0)} = c_{(i+i_0)} + \alpha \cdot w_i \cdot |c_{(i+i_0)}|, \quad (4)$$

where $1 \leq i \leq (N \cdot L)$, and α is the strength of the watermark ($\alpha > 0$). Here, c_i and w_i denote the i -th index of $\mathbf{c}$ and $\mathbf{w}$, respectively. The watermarked mesh is obtained by combining the Laplacian basis $\mathbf{U}$ with coefficient $\mathbf{c}'$ as in the following equation.

$$\mathbf{x}' = \mathbf{U}\mathbf{c}', \quad (5)$$

where $\mathbf{x}'$ is a vector of the reconstructed vertices for the watermarked mesh model.

3.2. Watermark Extraction

3.2.1. Detection of the Embedded Watermark

We designed the proposed collusion-resilient watermarking based on informed-detection watermarking that requires a reference mesh. For the watermark extraction, we first obtained a spectral coefficient vector $\mathbf{c}^* = (c_1^*, c_2^*, \dots, c_n^*)^T$ and $\mathbf{c}$ from the suspicious mesh $\mathbb{M}^*$, and reference mesh $\mathbb{M}$ using the method described in the embedding section. Then, we calculated the residuals vector $\mathbf{r}^*$ by subtracting $\mathbf{c}^*$ from $\mathbf{c}$.

We obtain two-dimensional vector $\mathbf{s}^* = (\mathbf{s}_1^*, \mathbf{s}_2^*, \dots, \mathbf{s}_N^*)$ from $\mathbf{r}^*$, where

$$\mathbf{s}_j^* = (r_{i_0+Nj+1}^*, \dots, r_{i_0+N(j+1)}^*). \quad (6)$$

Then, we determined whether the suspicious model was watermarked or not as follows:

$$\begin{cases} \text{Watermarked :} & \text{if } \sum_j \text{corr}(\mathbf{s}_j^*, \mathbf{s}_1) / N > \frac{\tau_w}{n}, \\ \text{No watermark :} & \text{otherwise.} \end{cases} \quad (7)$$

where $1 \leq j \leq N$, and $\mathbf{s}_1$ is the symbol vector used in the embedding step, and n is the number of attackers that can be covered by the code set $\mathbf{N}$. Here, τ_w is a threshold to minimize a false-positive rate ($< 10^{-5}$) that was determined by experiment without any collusion attack. Because the embedded signal could be attenuated by $\frac{1}{n}$ (in the worst case), τ_w must be adjusted by the maximum number of attackers to avoid false-negative error. The function term $\text{corr}(\mathbf{r}, \mathbf{s})$ denotes the normalized correlation calculated as in the following equation:

$$\text{corr}(\mathbf{r}, \mathbf{s}) = \frac{(\mathbf{r} - \bar{\mathbf{r}}) \cdot (\mathbf{s} - \bar{\mathbf{s}})}{\|\mathbf{r} - \bar{\mathbf{r}}\| \cdot \|\mathbf{s} - \bar{\mathbf{s}}\|}, \quad (8)$$

where the bar above a symbol denotes the mean value. When the sum of correlation values is smaller than $\frac{\tau_w}{n}$, we reject the suspicious model as a watermarked model whether the extracted code is attacked or not.

3.2.2. Colluder Accusation

To detect bit information '1', detector response ψ was derived using the embedding signal Equation (4),

$$\psi_j = \frac{\alpha(c_j^* \cdot \mathbf{s}_1)}{|\mathbf{s}_j^*|}, \quad (9)$$

where $1 \leq i \leq n_c$, and c_j^* is the corresponded vector of $\mathbf{s}_j^*$. Based on ψ , we reconstruct the fingerprint $\mathbf{y}^*$ based on the embedding equation as follows:

$$y_j^* = \begin{cases} 1 & \text{if } \psi_j > \tau_b, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

where y_j^* denote the j -th index of $\mathbf{y}$, and τ_b is a threshold to minimize the error rate (discussed in Section 3.3).

When we suspect that the extracted fingerprint has been attacked, we examine the extracted fingerprint $\mathbf{y}^*$ using the code set $\mathbf{N}$. The averaging attack on the binary sequences can be considered the logical AND of the fingerprint codes [8]. Using the GD-PBIBD code, we can find the pirates by comparing the bit position of a result sequence whose value is ‘1’ with all columns in the code set $\mathbf{N}$. Therefore, we designed an examination process:

$$\mathbf{v}^* = \mathbf{N}^T \mathbf{y}^*, \quad (11)$$

where $\mathbf{v}^*$ is a vector of which the index indicates each recipient. By normalizing and applying a floor function to each value of $\mathbf{v}^*$, we obtain vector $\mathbf{v}'$ as follows:

$$\mathbf{v}' = \begin{cases} 1 & \text{if } \frac{\mathbf{v}^*}{|\mathbf{y}^*|} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

Using the vector $\mathbf{v}'$, we decided whether the extracted code $\mathbf{w}^*$ was attacked or not. If $|\mathbf{v}'| = 1$ then $\mathbf{w}^*$ was not attacked by pirates. In contrast, if $|\mathbf{v}'| > 1$ then $|\mathbf{y}^*|$ indicates the number of attackers, and values from $\mathbf{v}'$ demonstrate explicit clues for tracking the attackers.

3.3. Detector Response Modeling and Error Minimization

To reflect the statistical characteristics of the actual noise and interference, the threshold τ_b has to be controlled to minimize the bit error ratio. In this section, we model a distribution of the responses from the extracted bit, and determine τ_b by analyzing the proposed model. The averaging attack of the anti-collusion code has the same effect as a bitwise AND operation of every bit position. Therefore, bit information ‘1’ can survive during the averaging when ‘0’ was not exist in the same bit position of the attacker’s code. The probability of ‘1’ from the code set $\mathbf{N}$ is $\frac{n-1}{n}$ [26], where n denotes the number of attackers covered by $\mathbf{N}$. After the collusion with n attackers, the expected value of bit information ‘1’ can be modeled as a binomial distribution as

$$E(n) = \sum_{k=0}^n \binom{n}{k} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-k}, \quad (13)$$

where k is the number of the ‘1’ bit in the same position of the attacker’s code.

To reflect the statistical characteristic of the actual noise and interference, the detector response ψ was modeled as a mixture of the Gaussian distribution. Similar to Equation (13), the embedded signal from Equation (4) is attenuated by the number of attackers (n). Therefore, the distribution of the attenuated signal of bit ‘1’ can be modeled as

$$\Psi(x|n, \sigma) = \sum_{k=0}^n \left(\binom{n}{k} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-k} \cdot f\left(x \middle| \frac{k}{n}, \sigma^2\right) \right), \quad (14)$$

where $f(x|\mu, \sigma^2)$ denotes the probability density function of the Gaussian distribution, and σ is determined by the watermark-to-noise-ratio (WNR) $\sigma = \|\alpha\|^2 / \sigma_d^2$, where σ_d^2 is the strength of the additive noise. Figure 2 shows simulated plots of the $E(n)$, and $\Psi(x|n, \sigma)$, where $n = 8$ and $\sigma = 0.1$. To verify the proposed model in Equation (14), we measured the detector response from 1000 sample models with the same parameters ($n = 8, \sigma = 0.1$). As a result of the tests, Figure 3a shows a histogram of the measured responses that has structure very similar to that of the proposed model (Figure 2b).

Figure 2. Plot of the proposed model: (a) $E(n)$; and (b) $\Psi'(x|n, \sigma)$ where $n = 8$, and $\sigma = 0.1$.

Figure 3. (a) Histogram of the detector responses obtained from the 1000 sample models ($n = 8$, $\sigma = 0.1$); and (b) obtained τ_b using Equation (15) with various σ , n values.

Using the model Ψ , we determine the threshold τ_b by minimizing the code extraction error as follows:

$$\tau_b = \underset{t}{\text{minimize}} (e_1 + e_2), \quad (15)$$

where e_1 and e_2 are false positive error and false negative error, respectively. Each error probability was derived from Ψ as follows,

$$e_1 = \int_0^t \left(\frac{1}{n}\right)^n \cdot f(x|1, \sigma^2) dx, \quad (16)$$

$$e_2 = \int_t^\infty \sum_{k=0}^{n-1} \left(\binom{n}{k} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-k} \cdot f(x|\frac{k}{n}, \sigma^2) \right) dx. \quad (17)$$

From now on, we focus on mathematical derivation, in order to provide the threshold value τ_b illustrated in Figure 3b. From Equation (15), τ_b is value that makes the derivative of $e_1 + e_2$ by t to be zero.

$$\frac{de_1}{dt} = \left(\frac{1}{n}\right)^n \cdot f(t|1, \sigma^2) \quad (18)$$

$$\frac{de_2}{dt} = - \sum_{k=0}^{n-1} \left(\binom{n}{k} \frac{1}{n}^k \left(1 - \frac{1}{n}\right)^{n-k} \cdot f\left(t \mid \frac{k}{n}, \sigma^2\right) \right) \quad (19)$$

Substitute $\binom{n}{k} \frac{1}{n}^k \left(1 - \frac{1}{n}\right)^{n-k}$ to α_k for $k = 1$ to n .

$$\frac{de_1 + de_2}{dt} = \alpha_n \cdot f(t \mid 1, \sigma^2) - \left(\sum_{k=0}^{n-1} \alpha_k \cdot f\left(t \mid \frac{k}{n}, \sigma^2\right) \right) \quad (20)$$

$$= \frac{1}{\sqrt{2\pi}\sigma} \cdot \left(\alpha_n \cdot e^{-\frac{(t-1)^2}{2\sigma^2}} - \left(\sum_{k=0}^{n-1} \alpha_k \cdot e^{-\frac{(t-\frac{k}{n})^2}{2\sigma^2}} \right) \right) \quad (21)$$

$$= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}} \cdot \left(\alpha_n e^{\frac{2t-1}{2\sigma^2}} - \left(\sum_{k=0}^{n-1} \alpha_k e^{\frac{2k t - \frac{k^2}{n^2}}{2\sigma^2}} \right) \right) \quad (22)$$

Substitute $e^{\frac{t}{n\sigma^2}}$ to x and $\alpha_k e^{-\frac{k^2}{2n^2\sigma^2}}$ to β_k for $k = 1$ to n .

$$= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}} \cdot \left(\beta_n x^n - \sum_{k=0}^{n-1} \beta_k x^k \right) \quad (23)$$

Therefore, the value τ_b that minimizes $e_1 + e_2$ can be derived from solving following polynomial of degree n .

$$\beta_n x^n - \sum_{k=0}^{n-1} \beta_k x^k = 0 \quad (24)$$

Finally, we can get τ_b by using solution of Equation (24)

$$\tau_b = n \cdot \sigma^2 \cdot \ln(x) \quad (25)$$

Figure 3b shows τ_b values for various σ and n by solving Equation (15). For instance, $(\tau_b = 0.9378)$ for the test environment $n = 8, \sigma = 0.1$ is presented as a red line in Figure 3a.

3.4. Extended Mesh Fingerprinting Scheme Using Tardos's Code

3.4.1. Tardos's Fingerprint Code

Among the various fingerprint codes against collusion attack, Tardos code [17] performs state-of-the-art performance. Tardos has proposed a randomized fingerprinting code based on the Boneh and Shaw's scheme [4] that is provably secure against collusion attack. Compared with conventional methods, this code reduced the length of the code to the square of its length. Following Tardos [17], we have provides the formal definition of the fingerprint code below.

A fingerprint code of length m for n users over the alphabet Σ is a distribution over the pairs (X, σ) , where X is an $n \times m$ matrix over Σ and σ is an algorithm that takes a string $y \in \Sigma^m$ (the pirated copy) as input, and produces a subset $\sigma(y) \subseteq [n] := 1, 2, \dots, n$ (the set of accused users). For $\phi \neq C \subseteq [n]$ a C-strategy is an algorithm ρ that takes the submatrix of X formed by the rows with indices in C as input, and produces a string $y = \rho \in \Sigma^m$ as output 1 and satisfies the marking condition that, for all positions $1 \leq i \leq m$, if all the values X_{ij} for $j \in C$ agree with some letter $s \in \Sigma$ then $y_i = s$. We say that a fingerprint codes is ϵ -secure against coalitions of of size c , if for any $C \subseteq [n]$ of size $|C| \leq c$ and for any C-strategy ρ , the error probability,

$$P[\sigma(\rho(X)) = \phi \text{ or } \sigma(\rho(X)) \not\subseteq c] \quad (26)$$

is at most ϵ . Tardos's fingerprint code is an $n \times m$ binary matrix, where n denotes number of users and m denotes the length of the fingerprint codes. The number of user n , the size of collusion c and the error bound ϵ are the key parameters that decide the size of the fingerprint code matrix.

3.4.2. Extended Mesh Fingerprinting Scheme Using Tardos's Code

Tardos's fingerprint code directly applied to the embedding equation Equation (4) as the same way in the case of GD-PBIBD code. In addition, the detection of the embedded watermark described in Section 3.2.1 also can be used with the fingerprint code. Then, embedded bit information y^* is extracted using Equations (9) and (10).

In contrast with the embedding and code extraction process, we should modified the accusation process described in Section 3.2.2. First, we examine the extracted fingerprint y^* as follows. Assume that the number of users is n and the length of the fingerprint code is m , Tardos defined the $n \times m$ matrix U and its entries are defined as follows:

$$U_{ij} = \begin{cases} \sqrt{\frac{1-p_i}{p_i}}, & \text{if } X_{ji} = 1, \\ \sqrt{-\frac{p_i}{1-p_i}}, & \text{if } X_{ji} = 0. \end{cases} \quad (27)$$

where p_i values from the probability array used in the fingerprint code generation step, and X_{ji} is an element in the fingerprint code matrix. Let the algorithm accuse user j on the extracted fingerprint $y = \{0, 1\}^m$ as input if:

$$\sum_{i=1}^m y_i U_{ji} > Z, \quad (28)$$

where $Z = 20c \log(1.0/\epsilon)$ as a threshold parameter. As described in [17], both false positive and false negative errors are controlled by the error bound ϵ . When the parameter ϵ for the lower error bound decreases for the robust fingerprint system, the length of the fingerprint code dramatically increases. Therefore, because of the limited number of the spectral coefficients (<3000) from the cover mesh model, the usage of the Tardos's code is relatively limited (e.g., the generated code with $\epsilon = 0.2, c = 6$ has length 4024).

4. Experimental Results

To assess the proposed scheme, we tested its robustness against collusion attacks. Experiments were carried out on six triangular mesh models from 3D mesh watermarking benchmark project [27]: Bunny, Rabbit, Venus, Horse, Cow, and Ramesses. We used the fingerprint code set constructed by [9] with parameters (72, 81, 9, 8, 0, 1), which could protect the model against $n = 8$ attackers. The watermarking algorithm was implemented in MATLAB 2012b using the mesh toolbox [28]. The geometric distortion was measured using maximum root mean square error (MRMS) [29]. Figure 4b shows the watermark with MRMS = 0.02% ($\alpha = 0.05$), and no visual difference can be perceived between Figure 4a,b. In addition, we embedded the watermark with high strength factor ($\alpha = 0.5$) to show the visual shape of the watermark (MRMS = 2%). The embedded watermark appears as noise strongly represented on certain areas of the model's surface. In addition, Figure 5 illustrates the distortion maps which represents the HSV maps of the geometric objective distortions between the original and the watermarked meshes based on the hausdorff distance [29].

Figure 4. Comparison between the original and the watermarked model: (a) original Bunny model; (b) watermarked model; (c) strongly watermarked model.

Figure 5. Sample distortion maps of the watermarked meshes: (a) Bunny; (b) Cow; (c) Horse; (d) Rabbit; (e) Ramesses; and (f) Venus. All watermarks were embedded with $\alpha = 0.05$ (MRMS = 0.02%).

4.1. Robustness Test for Collusion Attack

The experiments were divided into two test environments: (1) collusion attack test with a variety of compared methods; and (2) robustness test with collusion and noise addition attacks. To compare our method against the performance of a conventional scheme, we used a variety of robust watermarking methods as follows: Ohbuchi et al. [11], and Cho et al. [12]. Similar to our method, the method of Ohbuchi et al. [11] is based on the mesh spectral decomposition. The method of Cho et al. [12] embedded a watermark pattern by modifying the statistical features of the vertex norm. Anti-collusion

codes were adopted not only our method but also the compared methods [11,12]. In the experiments, we generated 81 watermarked models by embedding a fingerprint code for each recipient of the models. Then collusion attacks were performed 100 times using a set of randomly selected watermarked models for each iteration. For the collusion attack for our tests, we averaged a set of selected models and obtained one attacked model. We also conducted experiments by varying the number of selected models k ($1 \leq k \leq 8$). Robustness for collusion attack was evaluated by the number of times that tracing all attackers was successful.

Figure 6 shows the experimental results of collusion attack of the proposed method, Ohbuchi et al. [11], and Cho et al. [12]. In this figure, Cho-mean and Cho-var indicate the first and the second method of Cho et al., respectively. The proposed method perfectly found attackers for every attack when the number of attackers was 1–7, and it also performed well when the number of attackers was ‘8’. Because the GD-BIBD codes in the test were generated with $n = 8$, which could protect the model against ($n \leq 8$) attackers, the results decrease drastically when $n > 8$. In contrast, Cho-mean and Ohbuchi et al. did not trace attackers for all attacks when the number of attackers was two or more. Cho-var also did not work when the number of attackers was three or more. The results showed that proposed scheme was much more robust against collusion attack compared to the conventional schemes.

Figure 6. Comparison test results with collusion attack.

4.2. Robustness Test for Combination of Noise Addition and Collusion Attack

Figure 7 shows robustness to the combination of noise addition and collusion attack. We added Gaussian noise $\mathcal{N}(0, \sigma)$ to the averaged model, where $\sigma = 0.1, 0.01, 0.001$, or 0.0001 , and traced the attackers using the proposed detector. The performance of the proposed method for the noise strength $\sigma < 0.001$ was as good as the result for $\sigma = 0$, and the results in cases with $\sigma = 0.001$ was also acceptable. On the other hand, in the experiments with strong noise ($\sigma \geq 0.01$), the accuracy of our method decreased. Therefore, the result shows that the proposed method was robust to the combination of a small amount of added noise and collusion attack.

Figure 7. Accuracy of the proposed method with collusion attack and noise addition.

4.3. Experimental Results of Tardos's Code-Based Scheme

For the experiments with Tardos's fingerprint code, we used two parameter sets to generate the fingerprint code matrix: parameter set 1 ($\epsilon = 0.1, c = 3$, length = 2072), parameter set 2 ($\epsilon = 0.3, c = 4$, length = 1926). In addition, we generated 100 watermarked models by embedding a fingerprint code for each recipient of the models. Then collusion attacks were performed 100 times using a set of randomly selected watermarked models for each iteration. For the collusion attack for our tests, we averaged a set of selected models and obtained one attacked model. We also conducted experiments by varying the number of selected models k ($1 \leq k \leq c$). Robustness for collusion attack was evaluated by the number of times that tracing all attackers was successful.

Figure 8 shows robustness to the combination of noise addition and collusion attack. We added Gaussian noise $\mathcal{N}(0, \sigma)$ to the averaged model, where $\sigma = 0.01, 0.001$, or 0.0001 , and traced the attackers using the proposed detector. The performance of the proposed method for the noise strength $\sigma < 0.001$ was acceptable compare to the results without noise attack. On the other hand, in the experiments with strong noise ($\sigma \geq 0.01$), the accuracy of our method decreased. Therefore, similar to the the result shows that the proposed method was robust to the combination of a small amount of added noise and collusion attack. Compared to the GD-PBIBD-based scheme, Tardos's code-based scheme requires more bit capacity to carry the fingerprint code. This aspect results reduction in the ratio between imperceptibility and robustness for the fingerprinting scheme. Although Tardos's fingerprint code outperforms the other anti-collusion codes in many applications, it does not work well in some cases of the mesh fingerprinting such as the high number of accused users, and the environment in which noise can be added.

Figure 8. Cont.

Figure 8. Accuracy of the proposed method with collusion attack and noise addition. (a) Results with Tardos's code ($\epsilon = 0.1, c = 3$, length = 2072); (b) Results with Tardos's code ($\epsilon = 0.3, c = 4$, length = 1926).

5. Discussion and Conclusions

The main contribution of our work is a collusion resilient watermark based on the anti-collusion fingerprint code. In contrast to the existing robust mesh watermarking, our proposed method has sufficient capacity to carry the anti-collusion fingerprint code. To minimize the detection error, we also modeled the response of the detector and presented optimized thresholds for our method. Based on experiments with a public benchmark, the proposed method outperformed the conventional robust mesh watermarking in all cases.

Nowadays, the demand for 3D mesh watermarking has increased due to the emergence of low-cost 3D printers. On the other hand, copyright issues will inevitably occur with the expansion of 3D model sharing platforms such as Thingiverse, Pinshape, and Sketchfab. In spite of various watermarking methods, mesh watermarking techniques resilient to collusion attack were rarely considered. Therefore, we believe that the proposed method can give advantages to our industry as well as the academic society.

Author Contributions: Conceptualization, J.-U.H.; Investigation, J.-U.H. and I.-J.Y.; Methodology, J.-U.H.; Resources, I.-J.Y.; Software, J.-U.H.; Supervision, H.-K.L.; Validation, I.-J.Y.; Writing of the original draft, J.-U.H.

Funding: This work was funded by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2016R1A2B2009595) and by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIP) (No. R0126-15-1024, Managerial Technology Development and Digital Contents Security of 3D Printing based on Micro Licensing Technology).

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2016R1A2B2009595) and by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIP) (No. R0126-15-1024, Managerial Technology Development and Digital Contents Security of 3D Printing based on Micro Licensing Technology).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Wang, K.; Lavoue, G.; Denis, F.; Baskurt, A. A Comprehensive Survey on Three-Dimensional Mesh Watermarking. *IEEE Trans. Multimed.* **2008**, *10*, 1513–1527. [[CrossRef](#)]
- Hou, J.U.; Park, J.S.; Kim, D.G.; Nam, S.H.; Lee, H.K. Robust video watermarking for MPEG compression and DA-AD conversion. In Proceedings of the 1st International Workshop on Information Hiding and Its Criteria for Evaluation, Kyoto, Japan, 3 June 2014; pp. 2–8.
- Asikuzzaman, M.; Pickering, M.R. An overview of digital video watermarking. *IEEE Trans. Circuits Syst. Video Technol.* **2017**. [[CrossRef](#)]
- Boneh, D.; Shaw, J. Collusion-secure fingerprinting for digital data. *IEEE Trans. Inf. Theory* **1998**, *44*, 1897–1905. [[CrossRef](#)]

5. Celik, M.U.; Sharma, G.; Tekalp, A.M. Collusion-resilient fingerprinting using random prewarping. In Proceedings of the 2003 International Conference on Image Processing, Barcelona, Spain, 14–17 September 2003; Volume 1, p. 509.
6. Uccheddu, F.; Kuo, C.C.J.; Barni, M. Anticollusion watermarking of 3D meshes by prewarping. In Proceedings of the SPIE International Society for Optics and Photonics, San Jose, CA, USA, 18 March 2008; p. 68190S.
7. Stinson, D.R.; Wei, R. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM J. Discrete Math.* **1998**, *11*, 41–53. [[CrossRef](#)]
8. Trappe, W.; Wu, M.; Wang, Z.; Liu, K. Anti-collusion fingerprinting for multimedia. *IEEE Trans. Sign. Process.* **2003**, *51*, 1069–1087. [[CrossRef](#)]
9. InKoo, K.; Sinha, K.; Lee, H.K. New digital fingerprint code construction scheme using group-divisible design. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2006**, *89*, 3732–3735.
10. Zhu, X.; Chen, C.W. A collusion resilient key management scheme for multi-dimensional scalable media access control. In Proceedings of the 18th IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, 11–14 September 2011; pp. 2769–2772. [[CrossRef](#)]
11. Ohbuchi, R.; Takahashi, S.; Miyazawa, T.; Mukaiyama, A. Watermarking 3d polygonal meshes in the mesh spectral domain. In Proceedings of the Graphics Interface, Ottawa, ON, Canada, 7–9 June 2001; Volume 2001, pp. 9–17.
12. Cho, J.W.; Prost, R.; Jung, H.Y. An Oblivious Watermarking for 3-D Polygonal Meshes Using Distribution of Vertex Norms. *IEEE Trans. Sign. Proc.* **2007**, *55*, 142–155. [[CrossRef](#)]
13. Hou, J.U.; Kim, D.G.; Choi, S.; Lee, H.K. 3D Print-Scan Resilient Watermarking Using a Histogram-Based Circular Shift Coding Structure. In Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security, Portland, OR, USA, 17–19 June 2015; ACM: New York, NY, USA, 2015; pp. 115–121.
14. Itier, V.; Puech, W.; Bors, A. Cryptanalysis aspects in 3-D watermarking. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 4772–4776.
15. Jong-Uk, H.; In-Jae, Y.; Hyun-Ji, S.; Lee, H.K. Robust 3D Mesh Watermarking Scheme for an Anti-Collusion Fingerprint Code. In Proceedings of the 18th World Conference on Information Security Applications, Daejeon, Korea, 24–26 August 2017; Volume 1, pp. 74–83.
16. Bollobás, B. *Modern Graph Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1998; Volume 184.
17. Tardos, G. Optimal probabilistic fingerprint codes. *J. ACM* **2008**, *55*, 10. [[CrossRef](#)]
18. Kang, I.K.; Lee, C.H.; Lee, H.Y.; Kim, J.T.; Lee, H.K. Averaging attack resilient video fingerprinting. In Proceedings of the 2005 IEEE International Symposium on Circuits and Systems (ISCAS), Kobe, Japan, 23–26 May 2005; pp. 5529–5532.
19. Maity, S.P.; Maity, S.; Sil, J.; Delpha, C. Collusion resilient spread spectrum watermarking in M-band wavelets using GA-fuzzy hybridization. *J. Syst. Softw.* **2013**, *86*, 47–59. [[CrossRef](#)]
20. Kuribayashi, M. Simplified MAP detector for binary fingerprinting code embedded by spread spectrum watermarking scheme. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 610–623. [[CrossRef](#)]
21. Hou, J.U.; Kim, D.G.; Lee, H.K. Blind 3D Mesh Watermarking for 3D Printed Model by Analyzing Layering Artifact. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2712–2725. [[CrossRef](#)]
22. Jang, H.U.; Choi, H.Y.; Son, J.; Kim, D.; Hou, J.U.; Choi, S.; Lee, H.K. Cropping-resilient 3D mesh watermarking based on consistent segmentation and mesh steganalysis. *Multimed. Tools Appl.* **2018**, *77*, 5685–5712. [[CrossRef](#)]
23. Wang, K.; Lavoué, G.; Denis, F.; Baskurt, A. Technical Section: Robust and Blind Mesh Watermarking Based on Volume Moments. *Comput. Graph.* **2011**, *35*, 1–19. [[CrossRef](#)]
24. Alface, P.R.; Macq, B.; Cayre, F. Blind and robust watermarking of 3D models: How to withstand the cropping attack? In Proceedings of the 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 16 September–19 October 2007; Volume 5, p. 465.
25. Karni, Z.; Gotsman, C. Spectral compression of mesh geometry. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques; ACM Press/Addison-Wesley Publishing Co.: New York, NY, USA, 2000; pp. 279–286.
26. Clatworthy, W.H.; Cameron, J.M.; Speckman, J.A. *Tables of Two-Associate-Class Partially Balanced Designs*; US Government Printing Office: Washington, DC, USA, 1973; Volume 63.

27. Wang, K.; Lavoué, G.; Denis, F.; Baskurt, A.; He, X. A benchmark for 3D mesh watermarking. In Proceedings of the IEEE International Conference on Shape Modeling and Applications, Aix-en-Provence, France, 21–23 June 2010; pp. 231–235.
28. Peyré, G. The Numerical Tours of Signal Processing—Advanced Computational Signal and Image Processing. *IEEE Comput. Sci. Eng.* **2011**, *13*, 94–97. [[CrossRef](#)]
29. Cignoni, P.; Rocchini, C.; Scopigno, R. *Metro: Measuring Error on Simplified Surfaces*; Technical Report; Centre National de la Recherche Scientifique: Paris, France, 1996.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).