Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 241093, 7 pages
http://dx.doi.org/10.1155/2014/241093

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 Yu[2]

[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

Academic Editor: Her-Terng Yau

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 does make 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], Alface 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 by the perturbing of later processed neighboring vertices. The convergence problem is that the user cannot control the distortion induced by the iteratively perturbing process.

Fornaro and Sanna [17] proposed a public key scheme to authenticating constructive solid geometry (CSG) models which has a serious drawback that it cannot locate the tampered region(s). Yeo and Yeung [19] proposed the use of content-based signature to authenticate 3D volume data.

Lin et al. [18] proposed a modified fragile watermarking scheme similar to Yeo and Yeung's method [19]. The causality problem is conquered in this method by applying two different hash functions on the vertex coordinates, without considering the neighboring vertices of a vertex. In the watermark embedding stage, they slightly perturbed every vertex making these two hash function values equal. However, the convergence problem still occurs in this scheme. To avoid heavy distortion due to vertex perturbing, they set a threshold and simply skip the vertices that could not meet the requirement under the threshold. This causes some embedding holes which cause false alarms in the watermark extraction stage.

Chou and Tseng [16] proposed a fragile watermarking scheme for authenticating 3D mesh models. A watermark embedded by this method is robust to translation, rotation, and uniform scaling but is sensitive to other operations. The main idea of the method is to keep the ratio between the distance from the mesh center to each surface face and a quantization step remaining the same after the model is translated, rotated, or uniformly scaled. There are two major drawbacks in this scheme. Firstly, it is a semipublic watermarking scheme since the original watermark is needed in the decoding stage to authenticate the watermarked model. But a public watermarking scheme is preferred in fragile watermarking. Secondly, it fails in locating the changed regions since the center position of the mesh will be changed when any vertex has been changed.

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 watermarking 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 watermark 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.

The remaining sections of this paper are organized as follows. The proposed 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.

Table 1: Relationship between parity check bits and data bits.

	D_1	D_2	D_3	D_4
P_1	×	×		×
P_2	×		×	×
P_3		×	×	×

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, and D_4) into seven bits by adding three parity check bits (P_1, P_2, and P_3). Each parity check bit is created by its associated data bits. Table 1 illustrates the relationship between parity check bits and data bits, where "×" indicates the relationship exists between parity check bits and associated data bits.

As shown in Table 1, the parity check bits can be created by using the associated date bits, and the results are shown in Table 2, where "⊕" donates 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 $HG^T = \mathbf{0}$, where $\mathbf{0}$ 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:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}_{3,7},$$
$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{4,7}. \quad (1)$$

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

$$H = \left[I_{n-k}, P^T\right], \quad G = \left[P, I_k\right]. \quad (2)$$

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

$$HG^T = \left[I_{n-k}, P^T\right] \begin{bmatrix} P^T \\ I_k \end{bmatrix} = P^T \oplus P^T = 0. \quad (3)$$

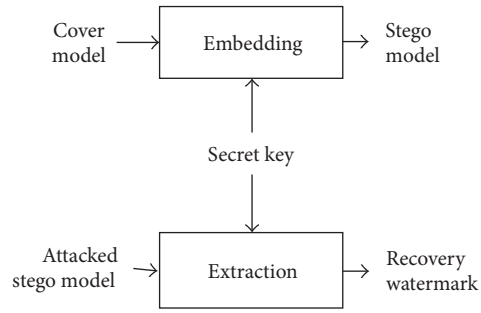

Figure 1: Proposed fragile watermarking scheme.

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 $P_0, P_1, P_2, \ldots, P_{N-1}$, $0 \leq i \leq N-1$. 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

$$q_i^x = \frac{q_i^{x\_\text{original}} - \text{Min}\left(X\_\text{original}\right)}{\text{Max}\left(X\_\text{original}\right) - \text{Min}\left(X\_\text{original}\right)}$$
$$q_i^y = \frac{q_i^{y\_\text{original}} - \text{Min}\left(Y\_\text{original}\right)}{\text{Max}\left(Y\_\text{original}\right) - \text{Min}\left(Y\_\text{original}\right)} \quad (4)$$
$$q_i^z = \frac{q_i^{z\_\text{original}} - \text{Min}\left(Z\_\text{original}\right)}{\text{Max}\left(Z\_\text{original}\right) - \text{Min}\left(Z\_\text{original}\right)},$$

where x, y, z denote the X, Y, Z axis; i is the vertex $0 \leq i \leq N-1$; Max (X_original) denotes the maximum value of the cover model at X axis, and so as Max (Y_original) and Max (Z_original), Min (X_original) denotes the minimum value

Table 2: Create parity check bits.

Parity check bits	Associated date bits
P_1	$D_1 \oplus D_2 \oplus D_4$
P_2	$D_1 \oplus D_3 \oplus D_4$
P_3	$D_2 \oplus D_3 \oplus D_4$

of the cover model at X axis, and so as Min (Y_original) and Min (Z_original).

Step 2 (extraction of data bits). Without loss of generality, suppose q_i^r is the point to be embedded, where r denotes the axis usedand $0 \le i \le N-1$. The 4 bits, which are the 4th least significant bit to 7th least significant bit of point q_i^r, 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_i^r 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_i^r 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_i^r in stego model are extracted and considered as input 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 $w_i^{r\prime}$.

Step 4 (extraction of watermark). The embedded watermark $h_i^{r\prime}$ can be extracted from the 3 least significant bits of point q_i^r in stego model.

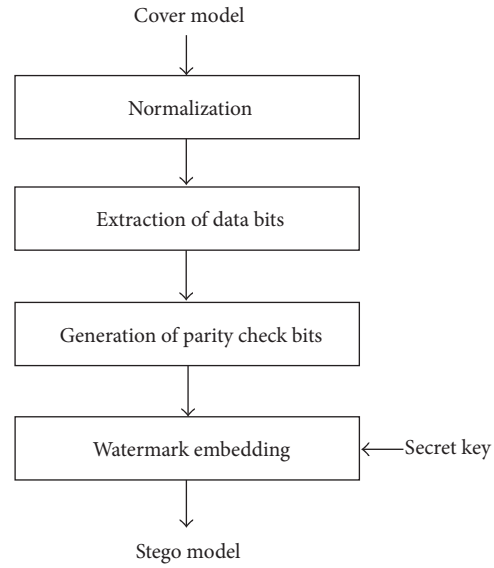

Figure 2: The flow chart of embedding process.

Step 5 (verification). The verification is achieved when the watermark $w_i^{r\prime}$ is equal to the embedded watermark $h_i^{r\prime}$.

$$R\left(w_i^{r\prime}\right) = \begin{cases} \text{no tempered}, & w_i^{r\prime} = h_i^{r\prime} \\ \text{modified}, & w_i^{r\prime} \neq h_i^{r\prime}. \end{cases} \quad (5)$$

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 Haussdorff 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.

Model (size)	Cover model	Stego model	Model (size)	Cover model	Stego model
Angelfish (19724)			Teapot (28922)		
Bunny (34837)			Venus (33591)		
Lion (17352)			Wolf (30084)		

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

Models	Methods					
	RMS			RMS ratio		
	Yang's method	Wang's method	Proposed	Yang's method	Wang's method	Proposed
Anglefish	$3.74 * 10^{-4}$	$3.75 * 10^{-5}$	$5.17 * 10^{-6}$	$2.03 * 10^{-6}$	$1.90 * 10^{-9}$	$2.61852 * 10^{-10}$
Bunny	$3.75 * 10^{-4}$	$4.50 * 10^{-5}$	$5.06 * 10^{-6}$	$1.50 * 10^{-6}$	$1.28 * 10^{-9}$	$1.45118 * 10^{-10}$
Lion	$3.74 * 10^{-4}$	$4.63 * 10^{-5}$	$4.96 * 10^{-6}$	$3.49 * 10^{-7}$	$2.64 * 10^{-9}$	$2.82942 * 10^{-10}$
Teapot	$3.73 * 10^{-4}$	$4.53 * 10^{-5}$	$5.00 * 10^{-6}$	$2.30 * 10^{-7}$	$1.53 * 10^{-9}$	$2.82942 * 10^{-10}$
Venus	$3.74 * 10^{-4}$	$4.18 * 10^{-5}$	$5.07 * 10^{-6}$	$1.58 * 10^{-6}$	$1.62 * 10^{-9}$	$1.50865 * 10^{-10}$
Wolf	$3.75 * 10^{-4}$	$4.38 * 10^{-5}$	$5.00 * 10^{-6}$	$4.62 * 10^{-7}$	$1.45 * 10^{-9}$	$1.65854 * 10^{-10}$

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

$$\text{RMS\_Ratio} = \frac{\text{RMS}}{\text{Diagonal\_Length}}, \tag{6}$$

where

$$\text{RMS} = \sum_{i=0}^{N-1} \left(\left(q_i^x - q_i^{x\_\text{original}}\right)^2 + \left(q_i^y - q_i^{y\_\text{original}}\right)^2 + \left(q_i^z - q_i^{z\_\text{original}}\right)^2 \right)^{1/2} (N)^{-1}, \tag{7}$$

where q_i^x denotes the x coordinate of the point in stego model, so as q_i^y and q_i^z, $0 \le i \le N-1$,

$$\text{Diagonal\_Length} = \sqrt{X\_\text{size}^2 + Y\_\text{size}^2 + Z\_\text{size}^2}. \tag{8}$$

The RMS and the RMS ratio comparison results of [36, 37] and the proposed method are as shown in Table 4.

In this paper, a public hamming code based fragile watermarking technique is proposed for 3D objects verification. An adaptive watermark of each cover model can be generated from itself by using the hamming code technique. In the embedding stage, a simple least significant bit 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. The verification information should be embedded for achieving the verification tasks in all the previously mentioned 3D fragile watermarking schemes. A comparison of the proposed scheme with Lin's method [18], Yeo and Yeung's method [19], and Chou and Tseng's methods [34, 35] is given in Table 5, where "√" indicates that the relationship exists between methods and comparison items and "—" indicates there is no relationship between methods and comparison items.

5. Conclusions

The proposed public hamming code based fragile watermarking is a blind approach in spatial domain. It is simple to implement and it does not need the original model or the watermark for verification and localization of tampering detection of 3D objects. Furthermore, the causality problem, convergence problem, and embedding holes problem can be overcome by the proposed scheme.

Table 5: Comparison of previous methods and the proposed method.

Comparison items	Methods				
	Yeo and Yeung's method	Lin et al.'s method	Chou and Tseng's method (2006)	Chou and Tseng's method (2009)	Proposed method
No verification information	—	—	—	—	√
No causality problem	—	√	√	√	√
No convergence problem	—	—	√	√	√
No embedding holes	—	—	—	—	√
Verification rate	—	—	~42%	12%~16%	100%
Data hiding rate (bits/vertex)	<1	<1	<1.5	<1.5	3
Tampering detection	√	√	√	√	√
Localization of tampered region	√	√	√	√	√
Distortion control	—	√	√	√	√

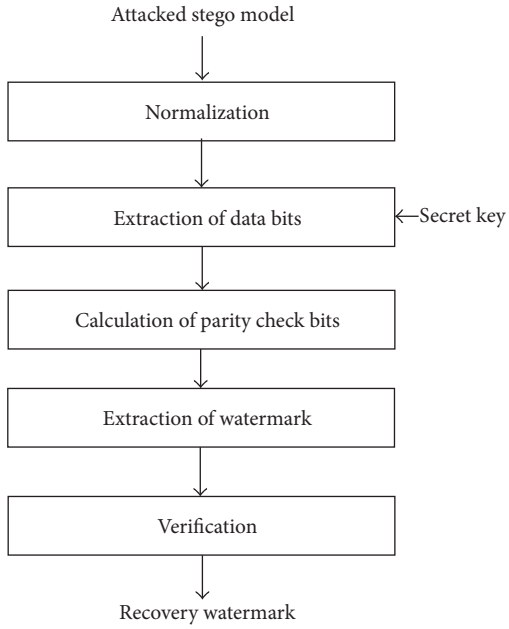

Figure 3: The flow chart of extraction process.

The main purpose in this paper is to authenticate the integrity of 3D polygonal messes 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

[1] B. Pfitzmann, "Information hiding terminology—results of an informal plenary meeting and additional proposals," in *Proceedings of the 1st International Workshop on Information Hiding*, pp. 347–350, 1996.

[2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[3] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking—Attacks and Countermeasures*, Kluwer Academic Publishers, London, UK, 2001.

[4] P. Wayner, *Disappearing Cryptography—Information Hiding: Steganography & Watermarking*, Morgan Kaufmann, San Francisco, Calif, USA, 2nd edition, 2002.

[5] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, London, UK, 2000.

[6] F. Daraee and S. Mozaffari, "Watermarking in binary document images using fractal codes," *Pattern Recognition Letters*, vol. 35, pp. 120–129, 2014.

[7] J. L. Hernandez-Ardieta, A. I. Gonzalez-Tablas, J. M. de Fuentes, and B. Ramos, "A taxonomy and survey of attacks on digital signatures," *Computers & Security*, vol. 34, pp. 67–112, 2013.

[8] C.-F. Chou, W. C. Cheng, and L. Golubchik, "Performance study of online batch-based digital signature schemes," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 98–114, 2010.

[9] J. Zhang, "A kind of message-recoverable fairness blind digital signature scheme," *Procedia Engineering*, vol. 15, pp. 2103–2107, 2011.

[10] D.-R. Lin, C.-I. Wang, Z.-K. Zhang, and D. J. Guan, "A digital signature with multiple subliminal channels and its applications," *Computers and Mathematics with Applications*, vol. 60, no. 2, pp. 276–284, 2010.

[11] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*, Morgan Kauffman Publishers, San Francisco, Calif, USA, 2002.

[12] J. Seitz, *Digital Watermarking for Digital Media*, Information Science Publishers, Hershey, Pa, USA, 2005.

[13] M. Arnold, S. D. Wolthusen, and M. Schmucker, *Techniques and Applications of Digital Watermarking and Content Protection*, Artech House Publishers, Morwood, Mass, USA, 2003.

[14] C. W. Wu, "Multimedia data hiding and authentication via halftoning and coordinate projection," *EURASIP Journal on Advances in Signal Processing*, vol. 2002, Article ID 634070, pp. 143–151, 2002.

[15] J. R. Goldschneider, E. A. Riskin, and P. W. Wong, "Embedded multilevel error diffusion," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 956–964, 1997.

[16] C.-M. Chou and D.-C. Tseng, "A public fragile watermarking scheme for 3D model authentication," *Computer-Aided Design*, vol. 38, no. 11, pp. 1154–1165, 2006.

[17] C. Fornaro and A. Sanna, "Public key watermarking for authentication of CSG models," *Computer-Aided Design*, vol. 32, no. 12, pp. 727–735, 2000.

[18] H.-Y. S. Lin, H.-Y. M. Liao, C.-S. Lu, and J.-C. Lin, "Fragile watermarking for authenticating 3-D polygonal meshes," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 997–1006, 2005.

[19] B.-L. Yeo and M. M. Yeung, "Watermarking 3D objects for verification," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 36–45, 1999.

[20] J. Lang and Z. Zhang, "Blind digital watermarking method in the fractional Fourier transform domain," *Optics and Lasers in Engineering*, vol. 53, pp. 112–121, 2014.

[21] S. Pereira, S. Voloshynoskiy, and T. Pun, "Optimal transform domain watermark embedding via linear programming," *Signal Processing*, vol. 81, no. 6, pp. 1251–1260, 2001.

[22] X. Wang, C. Wang, H. Yang, and P. Niu, "A robust blind color image watermarking in quaternion Fourier transform domain," *Journal of Systems and Software*, vol. 86, no. 2, pp. 255–277, 2013.

[23] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. de Natale, and A. Neri, "A commutative digital image watermarking and encryption method in the tree structured Haar transform domain," *Signal Processing: Image Communication*, vol. 26, no. 1, pp. 1–12, 2011.

[24] H. Yang, X. Wang, and C. Wang, "A robust digital watermarking algorithm in undecimated discrete wavelet transform domain," *Computers & Electrical Engineering*, vol. 39, no. 3, pp. 893–906, 2013.

[25] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models," in *Proceedings of the 5th ACM International Multimedia Conference*, pp. 261–272, November 1997.

[26] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, "Watermarking 3D polygonal meshes in the mesh spectral domain," in *Proceedings of the Graphics Interface*, pp. 9–17, June 2001.

[27] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes," *Computer Graphics Forum*, vol. 21, no. 3, pp. 373–382, 2002.

[28] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 551–559, 1998.

[29] F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 939–949, 2003.

[30] P. R. Alface, B. Macq, and F. Cayre, "Blind and robust watermarking of 3D models: how to withstand the cropping attack?" in *Proceedings of the 14th IEEE International Conference on Image Processing (ICIP '07)*, vol. 5, pp. 465–468, September 2007.

[31] C.-M. Wang and P.-C. Wang, "Data hiding approach for point-sampled geometry," *IEICE Transactions on Communications*, vol. 88-B, no. 1, pp. 190–194, 2005.

[32] C.-M. Wang and P.-C. Wang, "Steganography on point-sampled geometry," *Computers and Graphics*, vol. 30, no. 2, pp. 244–254, 2006.

[33] S. Kanai, H. Date, and T. Kishinami, "Digital watermarking for 3D polygons using multiresolution wavelet decomposition," in *Proceedings of the 6th IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO '98)*, pp. 296–307, 1998.

[34] C.-M. Chou and D.-C. Tseng, "A public fragile watermarking scheme for 3D model authentication," *Computer-Aided Design*, vol. 38, no. 11, pp. 1154–1165, 2006.

[35] C.-M. Chou and D.-C. Tseng, "Affine-transformation-invariant public fragile watermarking for 3D model authentication," *IEEE Computer Graphics and Applications*, vol. 29, no. 2, pp. 72–79, 2009.

[36] Y. W. Yang, *A study on the spherical coordinate based fragile watermarking scheme for 3D models [M.S. thesis]*, Department of Computer Science and Engineering, National Chung-Hsing University, 2011.

[37] J. T. Wang, Y. T. Chang, Y. C. Chang, C. C. Huang, and C. M. Fan, "A reference-point-based fragile watermarking scheme for 3D model in spatial domain," in *Proceedings of the 2nd Conference on Applications of Innovation and Invention*, 2012.

[38] C. M. Kung, "A robust oblivious watermark system base on hybrid error correct code," *Journal of Multimedia*, vol. 5, no. 3, pp. 232–239, 2010.

[39] C.-C. Chang, K.-N. Chen, C.-F. Lee, and L.-J. Liu, "A secure fragile watermarking scheme based on chaos-and-hamming code," *Journal of Systems and Software*, vol. 84, no. 9, pp. 1462–1470, 2011.

[40] C. Desset, B. Macq, and L. Vandendorpe, "Block error-correcting codes for systems with a very high BER: theoretical analysis and application to the protection of watermarks," *Signal Processing: Image Communication*, vol. 17, no. 5, pp. 409–421, 2002.

[41] J.-M. Guo, S.-C. Pei, and H. Lee, "Watermarking in halftone images with parity-matched error diffusion," *Signal Processing*, vol. 91, no. 1, pp. 126–135, 2011.

[42] R. W. Hamming, "Error detecting and error correcting codes," *The Bell System Technical Journal*, vol. 29, pp. 147–160, 1950.

[43] J. S. chitode, *Digital Communications*, Technical Publications, Pune, India, 1st edition, 2007-2008.

[44] J. Bennour and J.-L. Dugelay, "Toward a 3D watermarking benchmark," in *Proceedings of the 9th IEEE International Workshop on Multimedia Signal Processing (MMSP '07)*, pp. 369–372, October 2007.