International Journal of Innovative Computing, Information and Control
Volume **5**, Number **12**, December **2009**
ICIC International ©2009 ISSN 1349-4198
pp. **xx–xx**

ADAPTIVE AUTHENTICATION SCHEMES FOR 3D MESH MODELS

Te-Yu Chen
Department of Computer Science and Engineering
National Chung Hsing University
250 Kuo Kuang Road, Taichung 402, Taiwan, R.O.C

Min-Shiang Hwang[1]
Department of Management Information Systems
National Chung Hsing University
250 Kuo Kuang Road, Taichung 402, Taiwan, R.O.C
mshwang@dragon.nchu.edu.tw

Jinn-Ke Jan
Department of Computer Science and Engineering
National Chung Hsing University
250 Kuo Kuang Road, Taichung 402, Taiwan, R.O.C

Abstract. *With the rapid development of computer technology and the increasing attractiveness of the 3D model, the 3D model has been applied widely in many different fields, such as engineering, medicine, entertainment, etc. Thus, security threats to 3D models must be seriously considered. Many researchers have proposed solutions to security issues with cryptography and still images. However, they are unsuitable for 3D models, because of the 3D models' special characteristics. In this paper, we propose two schemes to solve the problem of authentication in 3D mesh models. First, our schemes can detect and locate any unauthorized modification precisely. Second, both schemes have the features of being public and having the short key. Third, the distortion is controllable and is proven to be $k_1/3$ on average, where k_1 is a key specified by users. In addition, after being slightly modified, both schemes can deal with point-sampled geometry and be robust to affine transformations.*
Keywords: 3D models, Fragile watermarking, 3D (mesh) model authentication, Tamper detection, Distortion control.

1. **Introduction.** Owing to the rapid improvements of computer technology and the attractive characteristics of three-dimensional (3D) models, the application of 3D models have greatly increased in the fields of engineering, medicine, entertainment and so on. Thus, works represented by 3D models have become an epidemic in recent years. The same as all the digital works, 3D models are convenient to store, modify, duplicate and transmit. However, this convenience has brought about many security threats. Hence, the copyright protection and authentication of digital work deserve serious consideration.

Many sophisticated solutions have been proposed for the security issues in Cryptography [1-4]. However, most of the solutions are based on abstruse mathematics, which make

[1]Corresponding Author: Prof. Min-Shiang Hwang

them time consuming and difficult to understand. In addition, cryptographic encryption does not thoroughly protect the work because it can still be harmed after decryption.

Watermarking is the art of imperceptibly embedding information into the original work. Because there is a limit to the sensitivity of human beings' sensory organs, such as the eyes and the ears, we have difficulty noticing the subtle changes made to multimedia works. Hence, watermarking takes advantage of this characteristic to overcome some security threats. As a new kind of multimedia, 3D models are also suitably protected by watermarking. Until recent years, the majority of the proposed schemes for the field of watermarking have focused on still images [5-12]. Arising from the widespread application of 3D models, the technologies for 3D watermarking have gradually attracted more attention.

Most of the concepts and terminologies of 3D model watermarking are similar to those of still image watermarking. The original model, which will be protected and has not been embedded with information, is known as the cover model. The information embedded to the cover model is the watermark. The model with the watermark embedded in it is called the watermarked model.

Different categorizations exist for a variety of watermarking schemes [13-15]. From the view of the watermarking domain, schemes based on the spatial domain and the transformed domain exist. The former embeds a watermark by modifying the model's properties of geometry or topology, while the latter usually embeds a watermark after performing some transformation. The former gains the advantage over the latter in computational efficiency. However, the latter has the advantage over the former in robustness.

From the perspective of extraction method, watermarking schemes can be categorized as public and private. A public scheme is one which extracts the watermark without the help of the original model and watermark, while the original model is necessary for a private scheme to extract the watermark. In addition, a semi-public scheme does not require the original model for watermark extraction, but the original watermark is necessary for comparison with the extracted watermark.

According to the application purposes, watermarking schemes can be categorized as fragile and robust watermarking schemes. The watermark embedded by a fragile watermarking scheme is sensitive to the modification. The embedded watermark is destroyed if any unauthorized modifications are made to the watermarked model. Hence, the goal of fragile watermarking is to verify the integrity of each work. In a robust watermarking scheme, the watermark should survive any attack to the watermarked model. Hence, a robust watermarking scheme is mainly used to protect the model's copyright or ownership. From the capability of locating malicious modifications, fragile watermarking schemes can be further classified into vertex-level and region-level tamper localizations. The former is designed to locate any altered vertex accurately. The latter, having a much coarser tamper localization capacity, is designed to locate suspicious region in which some vertices are altered but some are not.

In general, a fragile watermarking scheme consists of two phases: the encoding (or watermark embedding) phase and the decoding (or watermark extraction) phase. Almost all modern fragile watermarking schemes are based on the special domain, in which vertices are perturbed to arrive at some predefined relationship in the encoding phase. In the decoding phase, if the watermarked model never suffered any malicious modification, the watermark can be extracted and the model can pass through the verification procedure. Otherwise, the malicious modification is detected.

The remainder of this paper is organized into six sections. In Section 2, related works are surveyed. The proposed schemes are introduced in Section 3. Section 4 is the analysis and discussion, experimental results are shown in Section 5, and finally, Section 6 concludes this paper.

2. **Related Works.** To the best of our knowledge, the first 3D fragile watermarking scheme was proposed by Yeo and Yeung in 1999 [16]. In their encoding phase, for each vertex v, they first computed the location index, $L(v)$, and the value index, $p(v)$. The computation of the location index concerns the vertex itself and some of its adjacent neighbors. The value index is computed by mapping the vertex's coordinates to a three-tuple of integers. The location index and the value index are analogous to pixel location and color information in the 2D image. With the key, the value index is further mapped to a bit value, then the vertex is moved so that the bit value is equal to the watermark bit value at location $L(v)$. The verification is achieved by checking whether these two values are equal in the decoding phase.

Flaws in Yeo and Yeung's scheme are evident. First, their scheme belongs to region-level tamper localization. The computation of a vertex's location index involves the vertex and its neighboring vertices. If a vertex is altered, all of its neighboring vertices' location indices are changed accordingly. Hence, the altered vertex and all of its innocent neighboring vertices are regarded as having been altered. Second, the key used in the scheme is too long and would be a burden. The key should be long enough for security, and be a composition of 768 binary entries recommended in [16]. Finally, the distortion is not controllable by the user. Yeo and Yeung's method used to perturb a vertex to satisfy the predefined relationship is based on trial and error. Because this method is not concrete and analytical, it may lead to long distance moving, and result in unacceptable distortion.

In 2004, Cayre et al. extended some concepts of [17] and presented a fragile watermarking scheme for 3D triangle meshes [18]. Each triangle along the traversal of the mesh is processed one after another. For a triangle ABC, the watermark vertex C is moved parallel and perpendicular to the reference edge AB to embed a watermark bit value and a watermark bit number respectively. Cayre et al.'s scheme is robust against affine transformations and cropping of the mesh.

Lin et al. proposed a fragile watermarking scheme to authenticate 3D polygonal meshes in 2005 [19]. The scheme is similar to, and is an improvement upon Yeo and Yeung's scheme. To relax heavy distortion caused by vertex perturbation, they defined an allowable region. The perturbation for a vertex is skipped when it goes beyond the allowable region. The skipping of some vertices leads to some watermark holes and results in imprecise tamper detection. Another drawback, inherited from Yeo and Yeung's scheme, is a long verification key.

Wu and Cheung proposed two watermarking schemes in 2005. The first is a fragile watermarking scheme for 3D meshes [20], while the second can achieve greater reversibility [21]. Both schemes are built on similar concepts. Based on the quantization step, the watermark is embedded in the distance between the centroid and the faces on the model. In addition to the watermark, the information about the perturbation made to the vertex is condensed and embedded into the distance for reversibility in their second scheme. The main drawback of the first schemes is that the function of locating malicious modifications is not provided. The watermark embedding procedure is referred to the global centroid of the mesh. Once any modification happens, the centroid will be changed immediately and results that the entire watermark information will be changed. Furthermore, in order to keep the centroid unchanged after perturbing some vertices, not all faces can be used to embed watermark information. Some vertices are involved in the centroid restoration process. While the centroid restoration process is absent in their second scheme. Thus the centroid and quantization step should be provided in the decoding phase.

In 2006, Chou and Tseng proposed a public fragile watermarking scheme for 3D model authentication [13]. In their scheme, a set of mark vertices is first selected, so that the whole model is covered by the union of these vertices and their neighboring vertices. The three coordinates of a vertex are presented with different intentions. The x coordinate is used to indicate whether the vertex is a mark vertex. A mark vertex's y and z coordinates are used to embed the watermark and the hash value of the watermark, respectively. Because the embedding approach on the y and z coordinates is based on the barycenters, an adjusting vertex method was proposed to keep the barycenters of the mark vertex unchanged. When verifying, if a mark vertex does not satisfy the predefined relationship, it is implied that some neighboring vertex (vertices) or the vertex itself has been tampered with. The scheme can locate the suspicious region, but it cannot precisely locate the tampered vertex. Moreover, causing by the adjusting vertex method, the scheme is not suitable for a model with a smooth surface. In order to keep the barycenters of a mark vertex unchanged, the adjusting vertex should be moved toward the direction exactly opposite against the direction the mark vertex is moved. Although the distortion on an individual vertex is tolerable, the drastic movements on the two adjacent vertices, the mark vertex and its adjusting vertex, will lead to a coarse result.

3. **The Proposed Schemes.** In this section, two new fragile watermarking schemes for the authentication of a 3D polygonal model are proposed. One is the adaptive authentication scheme, termed "AAS" for short. The other is the vertex digest scheme, termed "VDS" for short.

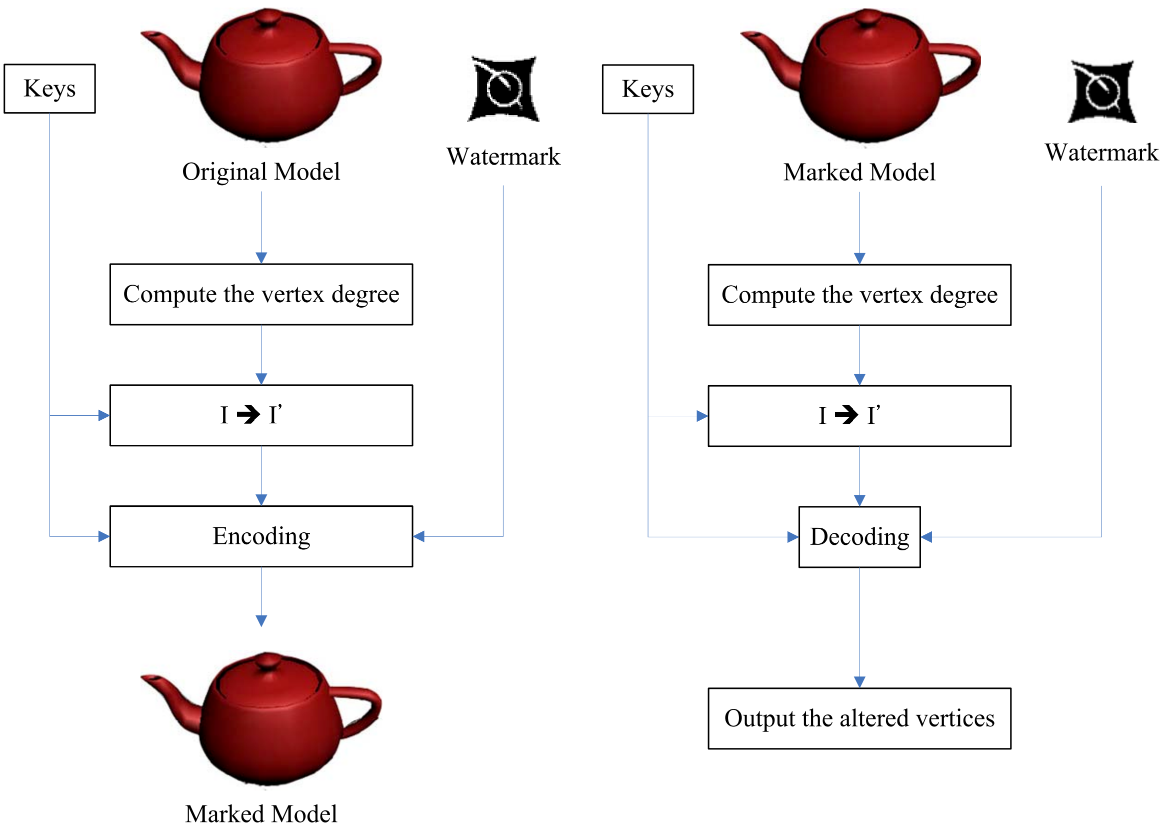
A 3D polygonal model can be presented as $M(V, C)$, where $V = \{v_1, v_2, v_3, \ldots, v_n\}$ is the set of vertices in M, and C is the connectivity between the vertices of V on M. The degree of a vertex is defined as the number of edges that are incident with the vertex. Hence, the degree of a vertex is the number of adjacent vertices of the vertex in 3D polygonal mode. Therefore, it can be used to stand for some of the connection relationships.

To precisely detect and locate the tampered vertices, we perturb all vertices to reach a predefined relationship. In the AAS, in order to bind the watermark to the position and the connectivity relationship of a vertex, the vertex's degree, coordinates x and y, and watermark are combined together with a hash function and embedded into this vertex's z coordinate. The VDS, on the other hand, roughly follows the same rationale, but the watermark is replaced by a key.

In practice, the hash function can be designed by adapting checksums, fingerprinting algorithms or cryptographic hash functions, such as MD5 or SHA hash functions. The choice is actually a trade-off between security level and computational complexity.

3.1. **The AAS.** In the AAS, to embed the watermark $W = \{w_1, w_2, w_3, \ldots, w_n\}$ into a 3D polygonal model $M(V, C)$, we first compute the vertex degree for all vertices in V. After perturbing the vertex indices, we combine the watermark with the position and the connection information of the vertices. We then perform the embedding one vertex after another according to the order of the new vertex indices. The watermarked model is generated after the procedures are complete.

In the decoding phase, with the watermarked model, we first compute the vertex degree. Then, we use the corresponding key to obtain the vertex indices. According to the vertex indices, secret keys and the original watermark, we verify whether the vertices have been maliciously altered. The encoding and decoding processes of the AAS are illustrated in Figure 1.



(a): The watermark encoding process. (b): The watermark decoding process.

FIGURE 1. The watermark encoding and decoding process.

3.1.1. *The Encoding Process.* The watermark embedding process is initiated by obtaining the vertex degree of the model. Then, the vertex index I is mapped to I' with a secret key K to enhance security. Afterward, the embedding of vertices can be performed one after one according to the order of the new vertex index I'. To bind the watermark to the position and the connectivity relationship for the vertex, the vertex's degree, its x and y coordinates, as well as the watermark are combined with a hash function and embedded into the vertex's z coordinate.

Two cases, $z \geq 0$ and $z < 0$, are discussed separately. Because the two cases are similar and symmetric to each other, we elaborate the case of $z \geq 0$. The four steps to embed a watermark value w_i to a vertex $v_i(x_i, y_i, z_i)$ when $z \geq 0$ are as follows:

(i) Take the key k_1 as the modulus and compute the residue of z_i using (1).

$$a_i = z_i \bmod k_1 \tag{1}$$

(ii) With the vertex's coordinates x_i and y_i, and its degree d_i as well as the watermark w_i, compute the hashing value m_i from a predefined hash function H by (2).

$$m_i = H(x_i, y_i, d_i, w_i) \tag{2}$$

(iii) Quantize m_i by using a key k_2 as in (3), where $0 \leq b_i < k_1$.

$$b_i = m_i / k_2 \tag{3}$$

(iv) Finally, slightly lower the value of z_i by a_i, such that the new value is a multiple of the key k_1, and then heighten the value by embedding b_i to it as illustrated in (4).

$$z_i' = z_i - a_i + b_i \tag{4}$$

To minimize the distortion of z_i, some subtle situations deserve to be further considered, and (4) can be modified by the following:

$$z_i' = \begin{cases} z_i - a_i + b_i - k_1, & \text{if } 0 \leq a_i < \frac{k_1}{2}, \frac{k_1}{2} \leq b_i < k_1, -k_1 \leq a_i - b_i < -\frac{k_1}{2} \\ z_i - a_i + b_i + k_1, & \text{if } \frac{k_1}{2} \leq a_i < k_1, 0 \leq b_i < \frac{k_1}{2}, \frac{k_1}{2} \leq a_i - b_i < k_1 \\ z_i - a_i + b_i, & \text{otherwise.} \end{cases} \tag{5}$$

With the modification in (5), the distortion of v_i can be reduced further.

The details of the watermark embedding process are shown in Algorithm 1.

Algorithm1 (The encoding algorithm of the AAS)

1. Compute the degree d_i for each vertex v_i;
2. The vertex index I is mapped to I' with a key K;
3. For each vertex $v_i(x_i, y_i, z_i)$ in the order of I'
   3.1. Compute a_i using (1);
   3.2. Compute m_i using (2);
   3.3. Compute b_i using (3);
   3.4. Compute z_i' using (4) or (5);
4. Output the marked model.

3.1.2. *The Decoding Process.* The watermarked model, the watermark and the corresponding keys K, k_1, k_2 are necessary to authenticate the model. The authentication is initiated by obtaining the vertex degree of the marked model. Then, the vertex index, which is the watermark embedding order, is retrieved by a mapping with the corresponding key K. We can then extract information embedded in the vertices one by one, based on the retrieved order. The discovery of a malicious vertex alteration can be determined by a comparison. The 3 steps to complete the detection on a vertex $v_i'(x_i', y_i', z_i')$ include:

(i) Extract the vertex's native information m_i about the position and connection relationship by (6).

$$m_i = H(x'_i, y'_i, d'_i, w_i) \tag{6}$$

(ii) Extract the pre-embedded information m'_i by performing (7).

$$m'_i = (z' \bmod k_1) \times k_2 \tag{7}$$

(iii) Compare these two values m_i and m'_i retrieved from (6) and (7). If $m_i \neq m'_i$, it means that the vertex has been maliciously altered.

After all the vertices in the model have been verified, the algorithm can output the tampered vertices. The details of the watermark decoding process are presented in Algorithm 2.

Algorithm2 (The decoding algorithm of the AAS)

---

1. Compute the degree d_i for each vertex v_i;
2. The vertex index I is mapped to I' with a given key K;
3. For each vertex $v_i(x_i, y_i, z_i)$ in the order of I'
   3.1. Compute $m_i = H(x_i, y_i, d_i, w_i)$;
   3.2. Compute $m'_i = (z \bmod k_1) \times k_2$;
   3.3. If $m_i \neq m'_i$ then mark v_i as altered;
4. Output the result.

3.2. **The VDS.** In the AAS, the information embedded into the z coordinate is similar to a message digested in cryptography. We can further adopt the concept of Keyed-hash message authentication code (HMAC) [22] to improve the AAS. The main improvement is that the embedded watermark in the AAS is replaced by a secret key in the VDS. In this manner, the watermark is no longer necessary, and remains able to authenticate the model and precisely detect malicious alterations.

The encoding and decoding processes of the VDS resemble those of the AAS. In the encoding process, for each vertex, we first compute the digest of the vertex's x, y coordinates and degree. Then, the digest is embedded into the vertex's z coordinate with the keys. The verification is completed by comparing the derived digest with that extracted from the z coordinate for each vertex in the decoding process. The details of the encoding and decoding processes are shown in Algorithm 3 and 4, respectively.

Algorithm3 (The encoding algorithm of the VDS)

---

1. Compute the degree d_i for each vertex v_i;
2. For each vertex $v_i(x_i, y_i, z_i)$
   2.1. Compute $m_i = H(x_i, y_i, d_i, K')$;
   2.2. Compute $z'_i = z_i - (z_i \bmod k_1) + m_i/k_2$;
3. Output the result.

Algorithm4 (The decoding algorithm of the VDS)

1. Compute the degree d_i for each vertex v_i;
2. For each vertex $v_i(x_i, y_i, z_i)$
   2.1. Compute $m_i = H(x_i, y_i, d_i, K')$;
   2.2. Compute $m'_i = (z \bmod k_1) \times k_2$;
   2.3. If $m_i \neq m'_i$ then mark v_i as altered;
3. Output the result.

4. **Analyses and Discussions.** We proposed two fragile watermarking schemes in the previous section: the AAS (the adaptive authentication scheme) and the VDS (the vertex digest scheme). Both authenticate the integrity of 3D models and not only detect, but also locate, any unauthorized modifications. The same as the majority of other proposed fragile watermarking schemes, the AAS is semi-public, because the watermark is necessary for the authentication process. Meanwhile, the VDS replaces the watermark with a key, and only some keys are needed to authenticate the integrity of a 3D model. Hence, the VDS is public.

In the AAS, the watermark information is embedded into the vertex in proper order. Therefore, during the decoding process, the traverse order of vertices in which the watermark is extracted should be the same as the order of the embedding process. With the VDS, however, the order of the vertices is no longer a concern, both in the encoding and decoding processes. The necessity of the traverse order is highly relative to the capability of robustness against vertex reordering in a scheme. Therefore, the VDS is robust against the attack of vertex reordering, while the AAS is not.

Until now, all the proposed fragile watermarking schemes we have referred to are of the category of region-level tamper localization. These watermarking schemes only locate the suspicious region, and false alarms may occur on some vertices, as illustrated in [13]. This is because the position information of neighboring vertices is taken into consideration when the predefined relationship is built for a mark vertex. Once the predefined relationship is detected to be destroyed on some mark vertex, all the neighboring vertices and the mark vertex itself are suspected. There is no way to distinguish the real tampered vertices from innocent vertices. Fragile watermarking schemes with vertex-level tamper localization capabilities are more attractive.

Both the AAS and VDS could be regarded as vertex-level tamper localization in the absence of some topology modifications. Owing to the fact that all vertices are adjusted to conform to the predefined relationship individually, any unauthorized modification to the vertices can be detected and located precisely. The degree is further considered in these schemes for preventing the cropping attack. Once some portion of a model is cut, vertices on the transection will be detected. Neither the degree nor the neighboring vertices of a vertex can absolutely represent the topology. As shown in Figure 2, the model is modified from (a) to (b). There are no obvious malicious modification except the edge (v_1, v_3) and (v_2, v_4) replaced by the edge (v_1, v_2) and (v_3, v_4) in the topology. Because the positions and degrees of all the vertices are the same in both cases, this malicious modification cannot be detected by the AAS or VDS. The scheme proposed by Chou and Tseng [13] also has the same predicament. If vertex v_0 is the mark vertex in their scheme, since the neighboring vertices of v_0 are the same in both cases, the barycenters computed by averaging the coordinates of neighbors are the same in both cases. Consequently, their scheme cannot detect this malicious alteration.

To resist this type of attacks, the AAS and VDS can be further modified by including the x and y coordinates of all neighboring vertices as the inputs of the hash function. However, the modified schemes fall into the class of region-level tamper localization. It is a trade-off. Such attacks that destroy the topology, while keeping the predefined relationship unchanged, are performed only after going deep into a model. We think this type of attack occurs rarely and can be ignored. Moreover, if the degree is removed from the hash function, both the AAS and VDS are suitable for the 3D models of point-sampled geometries, in which a 3D model is represented by dense vertices in the absence of topological information [23].

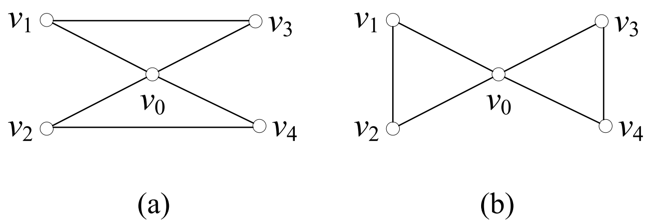


FIGURE 2. Two different topologies with the same vertices

From some viewpoints, certain innocent manipulations, such as affine transformations, which include rotation, translation, uniform scaling or a combination of the above, should not be detected as tampering. If such criterion is taken into account, some procedures could be adopted to recover these innocent manipulations. A model can be recovered from the affine transformations by utilizing the following information: (1) the three principal axes of principal component analysis (PCA) for the model, (2) the gravity center of the model, and (3) the diagonal length of the model's bounding box [24, 25]. To counter the manipulations of affine transformation on the watermarked model, we first perform model registration according to the aforementioned information in the decoding phase. In this way, both the AAS and VDS are resistant to affine transformations.

Distortion, which is used to evaluate the difference between the original and watermarked model, is one of the major considerations for a watermarking scheme. We adopt the definition introduced in [13] to calculate the distortion caused by the encoding process. The measurement of the distortion $D(M, M')$ is the average distortion of all vertices between the original model M and the watermarked model M' defined as:

$$D(M, M') = \frac{1}{|V|} \sum_{i=1}^{|V|} d(v_i, v_i'),$$

where $d(v, v')$ represents the Euclidean distance between the vertex v in the original model and its corresponding vertex v' in the watermarked model. Though almost all vertices of a model are perturbed after the encoding process in the AAS and VDS, the distortion is uniformly distributed and controllable. Therefore, the visualization quality of a marked model is fine.

Theorem 4.1. *In the AAS and VDS, the distance which a vertex will be perturbed in the encoding process is k_1 in maximum, and is $k_1/3$ on average. This value can be further decreased to $k_1/2$ as a maximum, and $k_1/6$ on average by introducing (5).*

Proof: In the AAS and VDS, all the moved vertices are perturbed only in their z coordinates. Therefore, only the z coordinate needs to be considered when we evaluate the distortion. By observing the encoding processes in the AAS and VDS, we see that the z coordinate of a vertex slightly decreases to be a multiple of k_1 and slightly increases to reach a new position z'. The movement is always less than k_1. Hence, it is clear that the maximum perturbing distance $|z - z'|$ is k_1.

To analyze the average case of $|z - z'|$, the positions of z and z' can be viewed as two independent random variables, X and Y, uniformly distributed in the interval between 0 and k_1. Therefore, to obtain the average of $|z - z'|$, we compute the expected value of $|X - Y|$:

$$\begin{aligned}
E(|X-Y|) &= \frac{1}{{k_1}^2}\int_0^{k_1}\int_0^{k_1}|x-y|dxdy \\
&= \frac{1}{{k_1}^2}\int_0^{k_1}\int_0^{y}(y-x)dxdy + \frac{1}{{k_1}^2}\int_0^{k_1}\int_y^{k_1}(x-y)dxdy \\
&= \frac{1}{{k_1}^2}\int_0^{k_1}\left(\left(yx-\tfrac{1}{2}x^2\right)\Big|_{x=0}^{x=y}\right)dy + \frac{1}{{k_1}^2}\int_0^{k_1}\left(\left(\tfrac{1}{2}x^2-yx\right)\Big|_{x=y}^{x=k_1}\right)dy \\
&= \frac{1}{{k_1}^2}\int_0^{k_1}\tfrac{1}{2}y^2dy + \frac{1}{{k_1}^2}\int_0^{k_1}\left(\tfrac{1}{2}y^2-k_1y+\tfrac{1}{2}{k_1}^2\right)dy \\
&= \frac{1}{{k_1}^2}\left(\left(\tfrac{1}{6}y^3\right)\Big|_{y=0}^{y=k_1} + \left(\tfrac{1}{6}y^3-\tfrac{1}{2}k_1y^2+\tfrac{1}{2}k_1^2y\right)\Big|_{y=0}^{y=k_1}\right) \\
&= \tfrac{1}{3}k_1
\end{aligned}$$

We conclude that the distance in which a vertex is perturbed in the encoding process is $k_1/3$ on average. By adopting (5), some special situations are taken into consideration separately, so that the distance between z and z' can be further reduced to $k_1/2$ in maximum. The average distance in which a vertex will be perturbed, by adopting (5), can be derived as demonstrated above, and the result is $k_1/6$. This completes the proof.

Thus, the distortion of a watermarked model $D(M, M')$ is controllable and is no more than $k_1/3$ or $k_1/6$ on average as a result of the adoption of (4) or (5), respectively. The smaller the key k_1 is, the less distortion will arise in the watermarked model. Under the limitation of machine precision, the distortion can be controlled and limited to a favorably small scale by choosing an appropriate key.

Our new schemes are compared with several proposed schemes, in terms of the criteria analyzed above. The results are listed in Table 1. In the scheme proposed by [13], fifty percent of the vertices on average having to modify their x coordinate to indicate whether they are marked or not. The marked vertices and the adjusting vertices should additionally modify their positions for different purposes. As a result, almost all the vertices will be moved in their scheme. Moreover, a marked vertex and its corresponding adjusting vertex, which are adjacent to each other, will be perturbed in the exactly opposite directions. This will lead to some perceptible abruptness in the watermarked model. Furthermore, because the watermark is divided and embedded in mark vertices, the traverse order is necessary for the reconstruction of the watermark in the decoding process.

TABLE 1. Comparisons of fragile watermarking schemes

Schemes/Criterion	C1	C2	C3	C4	C5	C6
[16]	Semi-public	No	Yes	Region	Long	No
[19]	Semi-public	No	Yes	Region	Long	Yes
[20]	Semi-public	Yes	No	Non	Short	Yes
[21]	Semi-public	Yes	No	Non	Medium	Yes
[13]	Public	Yes/No	No/Yes	Region	Short	Yes
The AAS	Semi-public	Yes	No	Vertex	Short	Yes
The VDS	Public	No	Yes	Vertex	Short	Yes

C1:Publicity, C2: Necessity of traverse order, C3:Robust to vertex reorder, C4:Tamper localization, C5:Key length, C6:Distortion controllability

TABLE 2. Experimental results of test models watermarked by the AAS and VDS with different k_1 values

Models	Vertices	Faces	k_1	Distortion	
				AAS	VDS
Apple	891	1704	0.01	0.003389	0.003328
			0.001	0.000349	0.000345
			0.0001	0.000037	0.000035
Beethoven	2655	5028	0.01	0.003367	0.003293
			0.001	0.000335	0.000328
			0.0001	0.000034	0.000034
Cow	3066	5804	0.01	0.003343	0.003303
			0.001	0.000335	0.000337
			0.0001	0.000033	0.000033
Triceratops	2832	5660	0.01	0.003267	0.003313
			0.001	0.000332	0.000333
			0.0001	0.000034	0.000033
Teapot	2022	3751	0.01	0.003381	0.003530
			0.001	0.000340	0.000364
			0.0001	0.000034	0.000035

5. **Experimental Results.** We implemented the AAS and VDS, using C++ programming language, and evaluated theses schemes on several 3D mesh models with various k_1 values. The information of models used in the experiments and the distortions of these models encoded by the AAS and VDS, with respect to various k_1 values are shown in

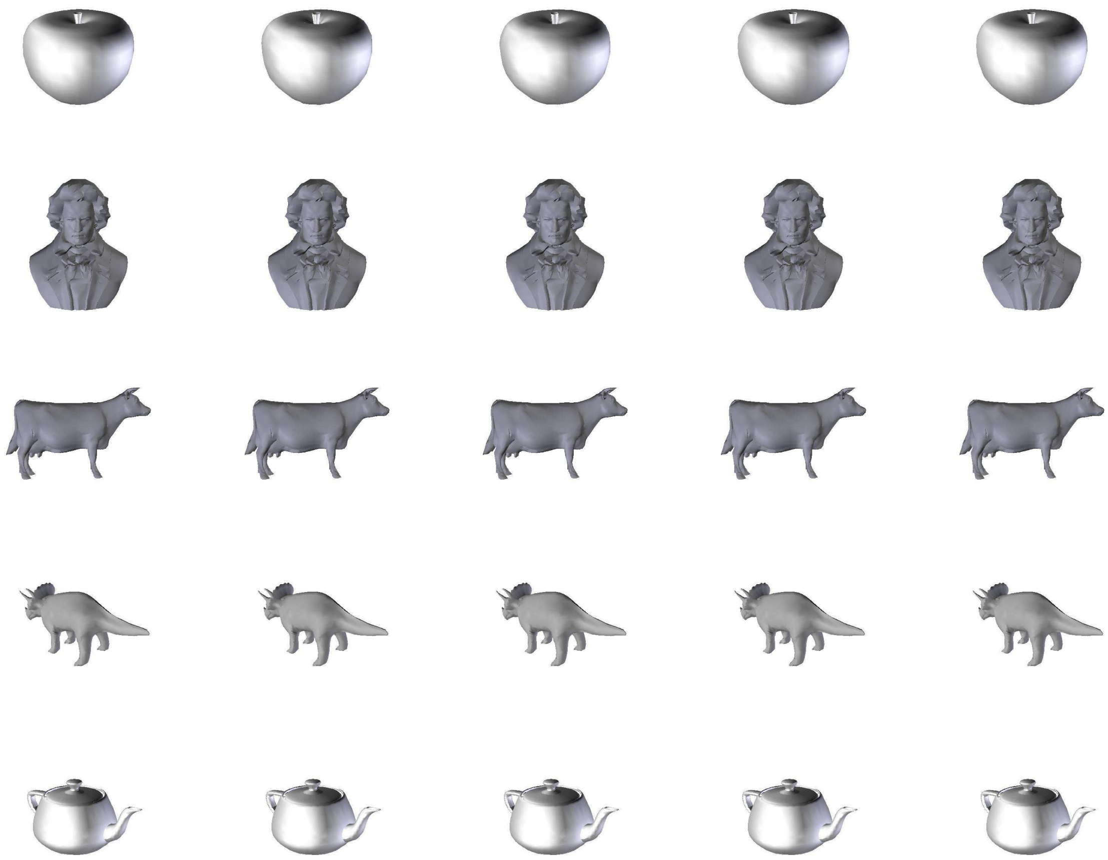


FIGURE 3. Experimental results. From left to right, the original models are shown in the first column. With the key $k_1 = 0.001$, the watermarked models produced by AAS and VDS are shown in the second and third columns, respectively. Finally the $4th$ and $5th$ columns show the watermarked models produced by the AAS and VDS with the key $k_1 = 0.0001$.

Table 2. As described in the previous section, the distortion is directly related to the key k_1, which is proven to be $k_1/3$ on average. As expected, the experimental results, as shown in Table 2, fully comply with the theoretical analysis.

Visual results of the watermarked models processed by the AAS and VDS, with respect to various k_1 values, are shown in Figure 3. The original models and the watermarked models are imperceptible on visualization in both schemes.

6. **Conclusions.** Two new fragile watermarking schemes for 3D mesh models, the AAS and VDS, have been proposed in this paper. The AAS is a semi-public scheme, while the VDS is a public scheme secure against the vertex reordering attack. Furthermore, both schemes can be easily modified to fit point-sampled geometry applications by removing the degree from the parameters of the hash function. Moreover, both schemes are resistant to affine transformations by performing model registration before the decoding process.

The distortions of both schemes are dependent on the key k_1. Theoretic analysis and experimental results give the same conclusion that the distortions are $k_1/3$ on average. Moreover, the experimental results go a step further to show the imperceptibility between the original and watermarked models. Hence, the distortions are compelling and are controllable by adopting feasible k_1 in both schemes.

Localization of malicious modification is taken into account in the AAS and VDS. In most cases, both fragile watermarking schemes achieve vertex-level tamper localization, which is superior to region-level tamper localization. The capability to identify the precise location of any malicious modifications is the first step towards tamper recovery. Tamper recovery is a promising field, worthy of further study.

Acknowledgment. This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the Grants NSC 96-2219-E-001-001 and NSC 96-2219-E-009-013. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 1996.
[2] B. Schneier, *Applied Cryptography*, 2nd ed. John Wiley & Sons, 1996.
[3] W. Stallings, *Cryptography and Network Security*, 4th ed. Prentice Hall, 2006.
[4] F. Zhang, X. Chen, Y. Mu, and W. Susilo, A new and efficient signature on commitment values, *International Journal of Network Security*, vol.7, no.1, pp.100-105, 2008.
[5] J. B. Feng, I. C. Lin, C. S. Tsai, and Y. P. Chu, Reversible watermarking: Current status and key issues, *International Journal of Network Security*, vol.2, no.3, pp.161-171, 2006.
[6] M. S. Hwang, C. C. Chang, and K. F. Hwang, Digital watermarking of images using neural network, *Journal of Electronic Imaging*, vol.9, no.4, pp.548-555, 2000.
[7] X. Kang, W. Zeng, and J. Huang, A multi-band wavelet watermarking scheme, *International Journal of Network Security*, vol.6, no.2, pp.121-126, 2008.
[8] S. D. Lin, Y. Kuo, and M. Yao, An image watermarking scheme with tamper detection and recovery, *International Journal of Innovative Computing Information and Control*, vol.3, no.6, pp.1379-1387, 2007.
[9] Z. M. Lu, C. H. Liu, and H. Wang, Image retrieval and content integrity verification based on multipurpose image watermarking scheme, *International Journal of Innovative Computing Information and Control*, vol.3, no.3, pp.621-630, 2007.
[10] S. S. Maniccam and N. Bourbakis, Lossless compression and information hiding in images, *Pattern Recognition*, vol.37, no.3, pp.475-486, 2004.
[11] G. Voyatzis and I. Pitas, Protecting digital-image copyrights: A framework, *IEEE Computer Graphics and Applications*, vol.19, no.1, pp.18-24, 1999.
[12] N. I. Wu and M. S. Hwang, Data hiding: Current status and key issues, *International Journal of Network Security*, vol.4, no.1, pp.1-9, 2007.
[13] 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.
[14] M. Luo, Digital watermarking of 3D models, *http://www-users.cs.york.ac.uk/∼ming*.
[15] B. Pfitzmann, Information hiding terminology, in *First International Workshop on Information Hiding*, pp.347-350, Berlin, Springer, LNCS 1174, 1996.
[16] 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.
[17] 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.

[18] F. Cayre, O. Devillers, F. Schmitt, and H. Maître, Watermarking 3D triangle meshes for authentication and integrity, INRIA, Research Report RR-5223, 2004.
[19] H. Y. S. Lin, H. Y. M. Liao, C. S. Lu, and J. C. Lin, Fragile watermarking for authenticating 3-D polygonal meshes, *IEEE Transaction on Multimedia*, vol.7, no.6, pp.997-1006, 2005.
[20] H. T. Wu and Y. M. Cheung, A fragile watermarking scheme for 3D meshes, in *The 7th ACM Workshop on Multimedia and Security (ACM2005)*, pp.117-123, 2005.
[21] H. T. Wu and Y. M. Cheung, A reversible data hiding approach to mesh authentication, in *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pp.774-777, 2005.
[22] NIST, The keyed-hash message authentication code (HMAC), FIPS PUB 198, 2002.
[23] M. Pauly, R. Keiser, and M. Gross, Multi-scale feature extraction on point-sampled surfaces, *Eurographics 2003, Computer Graphics Forum*, vol.22, no.3, pp.281-289, 2003.
[24] A. C. Rencher, *Methods of Multivariate Analysis*, 2nd ed. New York, Wiley, 2002.
[25] P. C. Wang and C. M. Wang, Reversible data hiding for point-sampled geometry, *Journal of Information Science and Engineering*, vol.23, no.6, pp.1889-1900, 2007.