Authentication of 3-D Polygonal Meshes

Hsueh-Yi Lin[1], Hong-Yuan Mark Liao[2], Chun-Shien Lu[2], and Ja-Chen Lin[1]

[1] Department of Computer and Information Science,
National Chiao-Tung University,
1001 Ta Hsueh Rd.,
Hsinchu 300, Taiwan
{HYLin, JCLin}@CIS.NCTU.edu.tw
[2] Institute of Information Science,
Academia Sinica,
Nankang, Taipei 115, Taiwan
{Liao, LCS}@IIS.Sinica.edu.tw

Abstract. Designing a powerful fragile watermarking technique for authenticating 3-D polygonal meshes is a very difficult task. Yeo and Yeung [17] were first to propose a fragile watermarking method to perform authentication of 3-D polygonal meshes. Although their method can authenticate the integrity of 3-D polygonal meshes, it is unable to distinguish malicious attacks from incidental data processings. In this paper, we propose a new fragile watermarking method which not only is able to detect malicious attacks, but also is immune to incidental data processings. During the process of watermark embedding, mesh parameterization techniques are employed to perturb the coordinates of invalid vertices while cautiously maintaining the appearance of the original model. Since the proposed embedding method is independent of the order of vertices, the hidden watermark is immune to some attacks, such as vertex reordering. In addition, the proposed method can be used to perform region-based tampering detection. The experimental results have shown that the proposed fragile watermarking scheme is indeed powerful.

1 Introduction

Transferring digitized media via the Internet has become very popular in recent years. Content providers who present or sell their products through networks are, however, faced with the copyright protection problem. In order to properly protect the rights of a content owner, it is desirable to develop a robust protection scheme that can prevent digital contents from being stolen or illegally distributed. From a user's point of view, after receiving a piece of digital content, he/she usually needs to verify the integrity of the content. As a result, there should be an authentication mechanism that can be used to perform the verification task. With the rapid advance of watermarking technologies in recent years, many investigators have devoted themselves to conducting research in this fast growing area. According to the objectives that a watermarking technique may achieve, two main-stream digital watermarking categories are: robust watermarking and fragile watermarking. While the former aims to achieve intellectual

T. Kalker et al. (Eds.): IWDW 2003, LNCS 2939, pp. 168–183, 2004.

property protection of digital contents, the latter attempts to authenticate the integrity of digital contents.

There are a great number of existing robust watermarking algorithms designed to protect 3-D graphic models [1,2,3], [8], [11,12,13], [16], [19]. Their common purpose is to provide a robust way to protect target contents when attacks are encountered. The existing fragile watermarking algorithms that are designed to authenticate 3-D graphic models are relatively few. In [5], Fornaro and Sanna proposed a public key approach to authenticating CSG models. In [17], Yeo and Yeung proposed a fragile watermarking algorithm for authenticating 3-D polygonal meshes. They embed a fragile watermark by iteratively perturbing vertex coordinates until a predefined hash function applied to each vertex matches the other predefined hash function applied to that vertex. Since their embedding algorithm relies heavily on the neighboring information of a vertex, it is unable to tolerate topological modifications, such as vertex reordering or polygonal simplification. In addition, particular attacks, such as floating-point truncation, applied to vertex coordinates might increase the false-alarm probability of tampering detection.

In this paper, we propose a new fragile watermarking algorithm for authenticating 3-D polygonal meshes. The proposed method not only is able to detect malicious attacks, but also is immune to the aforementioned unintentional data processings. In addition, the allowable range for alternating a vertex is explicitly defined so that the new scheme is able to tolerate reduction of floating-point precision. During the process of watermark embedding, the mesh parameterization technique is employed to perturb the coordinates of invalid vertices while cautiously maintaining the appearance of the original model. Since the proposed embedding method is independent of the order of vertices, the hidden watermark is immune to some vertex order-dependent attacks, such as vertex reordering.

The remainder of this paper is organized as follows. In Sec. 2, Yeo and Yeung's scheme for authenticating 3-D polygonal meshes is briefly reviewed. In Sec. 3, the proposed fragile watermarking method is described in detail. Experimental results are given in Sec. 4. Finally, conclusions are drawn in Sec. 5.

2 Yeo and Yeung's Approach and Its Drawbacks

In [17], Yeo and Yeung proposed a novel fragile watermarking algorithm which can be applied to authenticate 3-D polygonal meshes. In Yeo and Yeung's scheme [17], there are three major components, i.e., two predefined hash functions and an embedding process. For a given vertex, the vertex is identified as valid if and only if the values calculated by both hash functions are identical. Otherwise, the vertex is identified as invalid. During the authentication process, invalid vertices are considered as the set of vertices that has been tampered with. On the other hand, valid vertices indicate the set of vertices which has never been modified. In the embedding process, the coordinates of valid vertices are kept unchanged, but those of invalid vertices are iteratively perturbed until each of them becomes valid.

The first step in Yeo and Yeung's approach is to compute location indices. In this step, the first hash function is defined by a conversion function and associated with a given watermark pattern WM. The conversion function is used to convert a vertex coordinate $v = (v_x, v_y, v_z)$ into a location index $L = (L_x, L_y)$. The idea behind the conversion function is to map a three dimensional coordinate onto a two dimensional plane formed by a watermark pattern of dimension $WM\_X\_SIZE \times WM\_Y\_SIZE$. As a result, the location index L is used to point to a particular position in the watermark pattern. Then, the content of that particular position $WM(L)$ (either 0 or 1) is used for the purpose of comparison. Since the conversion function defined in [17] calculates the centroid of the neighboring vertices of a given vertex, the causality problem occurs. Furthermore, the traversal of vertices during the alternation of vertex coordinates must take causality into account so as to avoid error propagation.

The second step in Yeo and Yeung's approach is to compute value indices. In this step, the second hash function is related to a set of look-up tables, i.e., K_1, K_2, and K_3. These look-up tables, which are composed of sequences of bits, are generated and protected by an authentication key. Yeo and Yeung [17] proposed to convert each component of a vertex coordinate into an integer number so as to index into each of the look-up tables. The content of an indexed location is either 0 or 1. The three binary values derived from the three coordinates $p = (p_1, p_2, p_3)$ are then XOR processed to generate a final binary value. This binary value $K(p)$ is used as one of the components for deciding whether the current vertex is valid or not. If the vertex is not valid, then it is perturbed until it is valid. The amount of change that makes this vertex valid is the watermark embedded.

After establishing the above-mentioned two hash functions, the next step is to perturb the coordinates of all invalid vertices until they become valid. In [17], the authors proposed an iterative procedure which can gradually perturb an invalid vertex until both hash functions are matched. On the one hand, in order to maintain transparency, the embedding procedure must traverse in an orderly manner each vertex during the alteration of vertex coordinates. In addition, the ordering of vertices must be maintained during the watermark extraction process. Since the embedding process depends on the causality of the traversal of vertices, their method cannot tolerate an incidental modification, such as vertex reordering. This drawback to some extent limits the power of Yeo and Yeung's method. In this paper, we shall propose a new scheme that is more powerful than the existing fragile watermarking algorithms.

3 The Proposed Fragile Watermarking Method

In this section, we shall propose a new fragile watermarking scheme for authenticating 3-D polygonal meshes. In order to tackle the issues that were not handled by Yeo and Yeung [17], we employ the following concepts: (1) Each hash function can be designed so as to form a binary state space particularly helpful for defining the domain of allowable alternation for a given vertex. Accordingly, the domain of acceptable alternation for a given vertex can be defined as the intersection of the binary state spaces where the values of both hash functions match each other. (2) In order to resolve the causality problem, the conversion function used in the first hash function can be designed to simply perform the mapping from the 3-D space to a 2-D plane without considering the neighboring vertices of a vertex. Based on the above two concepts, we have designed a new scheme, which is shown in Fig. 1. With the new authentication scheme, malicious attacks applied to 3-D polygonal meshes can be easily distinguished from incidental ones. In what follows, we shall describe our authentication scheme in more detail.

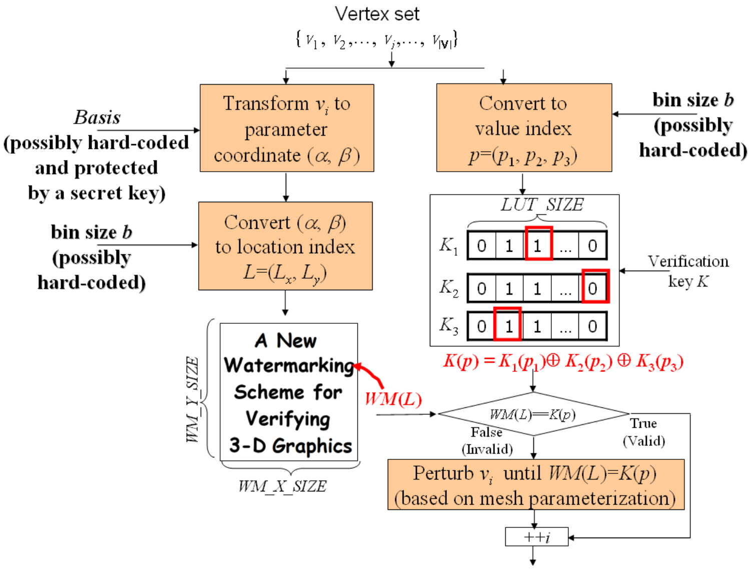

Vertex set
{ v_1, v_2,…, v_i,…, $v_{|V|}$}
Transform v_i to parameter coordinate (α, β)
Basis (possibly hard-coded and protected by a secret key)
Convert (α, β) to location index L=(L_x, L_y)
bin size b (possibly hard-coded)
A New Watermarking Scheme for Verifying 3-D Graphics
WM_Y_SIZE
WM_X_SIZE
$WM(L)$
Convert to value index p=(p_1, p_2, p_3)
bin size b (possibly hard-coded)
LUT_SIZE
K_1 0 1 1 ... 0
K_2 0 1 1 ... 0
K_3 0 1 1 ... 0
Verification key K
$K(p) = K_1(p_1) \oplus K_2(p_2) \oplus K_3(p_3)$
$WM(L)==K(p)$
False (Invalid)
True (Valid)
Perturb v_i until $WM(L)=K(p)$ (based on mesh parameterization)
++i

Fig. 1. The flowchart of the proposed authentication scheme for 3-D polygonal meshes.

3.1 Computing Location Indices

Since the conversion function used in the first hash function (the left hand side of Fig. 1) aims to calculate the location index that can be used to locate a particular bit in the watermark pattern, any functions that can transform a 3-D coordinate into a 2-D coordinate can serve this purpose. Therefore, it is possible to use some parameterization schemes to achieve the goal. As mentioned in the previous section, Yeo and Yeung did not use an analytical method to perturb invalid vertices. However, a systematic perturbation strategy is always preferable. Therefore, we propose to adopt the parameterization-based approach to make the vertex perturbation process analytic. For the purpose of clarity, we propose to split the location index computation process into two steps:

Step 1. Given a vertex coordinate v, the specified parameterization $S : R^3 \rightarrow R^2$ converts the vertex coordinate into a parameter coordinate. We propose to use so-called cylindrical parameterization [6] to perform the conversion task. The procedure involved in performing cylindrical parameterization is as follows [6]:

Given an oriented 3-D point, it is composed of a 3-D point m and its orientation n. A cylindrical parameterization process can be expressed as

$$S_{(m,n)}(v) \rightarrow (\alpha, \beta) = (\sqrt{\left\|v - m\right\|^2 - (n \cdot (v - m))^2}, n \cdot (v - m)), \qquad (1)$$

where (α, β) is the coordinate in the parameter domain. The range for each dimension of the parameter domain is $\alpha \in [0, \infty)$ and $\beta \in (-\infty, \infty)$, respectively.

Step 2. Convert the parameter coordinate formed in Step 1 into the so-called bin coordinate, i.e., the location index (L_x, L_y). This conversion can be accomplished by quantizing the parameter domain. In addition, a modulus operator is required to map them onto the dimension of a watermark pattern. In what follows, we shall describe how the parameter domains are quantized. Assume that the size of a 2-dimensional watermark pattern is $WM\_X\_SIZE \times WM\_Y\_SIZE$, the quantization formula for a cylindrical parameterization domain is as follows:

$$L = (L_x, L_y) = (\left\lfloor \frac{\alpha}{b} \right\rfloor \% WM\_X\_SIZE, \left\lfloor \frac{\beta}{b} \right\rfloor \% WM\_Y\_SIZE), \qquad (2)$$

where b is the quantization step for ordinary numeric values and % represents a modulus operator.

One thing to note is that the basis for cylindrical parameterization described in Step 1 can possibly be hard-coded into the algorithm so that detecting a watermark for the purpose of authentication can be realized as oblivious detection. A very important feature of the above design is that the quantized parameterization domain and the watermark pattern together form a binary state space. Such a state space is helpful for defining a legal domain of alternation for a given vertex. The state space corresponding to the cylindrical parameterization is illustrated in Fig. 2(a).

3.2 Computing Value Indices

Even though any functions for converting a floating-point number into an integer can be used to calculate value indices, the following conversion function was designed since it is able to form a binary state space. Assuming that the size of each look-up table is LUT_SIZE, the conversion function is formulated as

$$p = (p_1, p_2, p_3) = (\left\lfloor \frac{v_x}{b} \right\rfloor \% LUT\_SIZE, \left\lfloor \frac{v_y}{b} \right\rfloor \% LUT\_SIZE, \left\lfloor \frac{v_z}{b} \right\rfloor \% LUT\_SIZE), \qquad (3)$$

where b is the same quantization step as used to compute location indices.

As we have already mentioned, the quantization step b can be hard-coded into the implementation process. In addition, Fig. 2 reveals that the domain of acceptable alternation for a given vertex can be defined as the intersection of the binary state spaces where the values of both hash functions applied to that vertex match each other. Ideally, the largest acceptable displacement of alternation for a valid vertex is close to $\sqrt{3b^2}$ when the oriented point is chosen as $m(0,0,0)$ and $n(1,0,0)$.

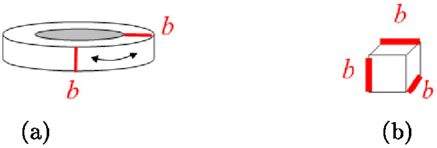


Fig. 2. The binary state space for a vertex: (a) the state space formed by cylindrical parameterization; (b) the state space formed by the conversion function for computing value indices.

3.3 Watermark Embedding

Since both hash functions have been well-designed to define the domain of acceptable alternation for a given vertex, the embedding procedure can focus on perturbing the coordinates of invalid vertices while maintaining transparency. In this paper, we apply a local mesh parameterization approach proposed in [10] for alternation of an invalid vertex. Our method is as follows: Given an invalid vertex v and its neighboring vertices in the counter-clockwise order $N(v) = \{v_1, v_2, \ldots, v_{|N(v)|}\}$, where $|N(v)|$ is the number of v's neighboring vertices, the proposed alternation procedure for an invalid vertex is divided into four steps, which can be explained with the help of Fig. 3. The details of the four steps are as follows:

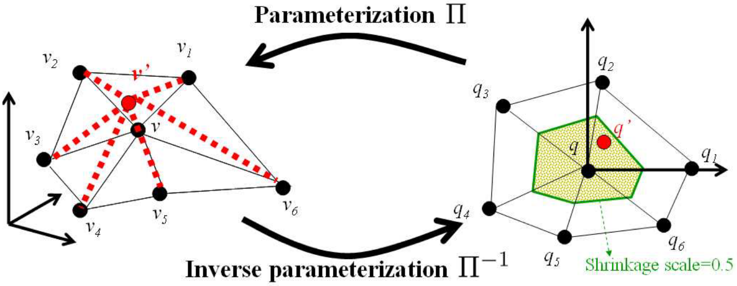


Fig. 3. The proposed alternation procedure for an invalid vertex.

Step 1. Transform the vertex coordinate v into the parameter coordinate q and its neighboring vertices $\{v_1, v_2, \ldots, v_{|N(v)|}\}$ to $\{q_1, q_2, \ldots, q_{|N(v)|}\}$, respectively, using arc-length parameterization [4]. Let $ang(a, b, c)$ be the angle formed by vectors $\overrightarrow{ba}$ and $\overrightarrow{bc}$. Then, the parameter coordinates are provided with the following properties:

$$\|q_k - q\| = \|v_k - v\|, \tag{4}$$

$$ang(q_k, q, q_{k+1}) = 2\pi \cdot ang(v_k, v, v_{k+1})/\theta, \tag{5}$$

where $\theta = \sum_{k=1}^{|N(v)|} ang(v_k, v, v_{k+1})$, $v_{|N(v)|+1} = v_1$, $q_{|N(v)|+1} = q_1$, and $k = 1, \ldots, |N(v)|$. If we set $q = (0, 0)$ and $q_1 = (\|v_k - v\|, 0)$, the parameter coordinates $q_2, q_3, \ldots, q_{|N(v)|}$ can be easily derived from Eqs. (4) and (5).

Step 2. Define an allowable region for alternating an invalid vertex in the parameter domain. Let the region be a shrunken ring whose origin is the parameter coordinate, q, and let the scale for shrinkage be 0.5. (Note that the reason for doing this is to avoid geometrical degeneracies, like triangle flipping, T-joints, etc.)

Step 3. Within the allowable region, find a new parameter coordinate q' satisfying the condition $WM(L(\Pi(q'))) = K(p(\Pi(q')))$. If there does not exist such a new parameter coordinate, alternation for the current invalid vertex is skipped, and $q' = q$ is assigned.

Step 4. Record the new vertex coordinate $v' = \Pi(q')$.

Note that after Step 1 is executed, the parameterization $\Pi : R^2 \to R^3$, also known as a bijective map [10], is established. In addition, Π^{-1} represents the inverse of the parameterization procedure. A bijective map can be realized by means of the well-known barycentric map [4]. Accordingly, the values of both hash functions applied to a parameter coordinate q' can be determined by $WM(L(\Pi(q')))$ and $K(p(\Pi(q')))$, respectively; on the other hand, the parameter coordinate q' can be transformed back into the vertex coordinate $v' = \Pi(q')$.

3.4 Analysis and Discussion

In this section, we shall conduct a thorough analysis of our authentication scheme for 3-D polygonal meshes. The watermarking parameters that can influence the quality of transparency and robustness are the shrinkage scale and bin size. On the other hand, we also know that the correlation value C can never reach 1. Therefore, we shall examine several crucial issues: (1) how to optimize the performance so that C can be very close to 1; (2) how to balance the competition between transparency and capacity using the shrinkage scale; and (3) how to guarantee the robustness of a hidden watermark. In what follows, we shall discuss these issues.

First of all, we aim to optimize the performance of our algorithm so that the watermark correlation value C can be very close to 1. In order to study this capacity issue, we make the following hypotheses: (1) the spacing between vertices will disable an invalid vertex from seeking a valid state; and (2) uniform parameterizations cannot tackle the irregularity of polygonal meshes, as illustrated in Fig. 4. In addition, the correlation of two watermarks is computed using Eq. (3) from [17]. To test the above two hypotheses, we picked five different models to generate analysis models with different mesh resolutions using a mesh resolution control algorithm described in [7]. Furthermore, for each model, we generated five analysis models corresponding to five different mesh resolutions. Fig. 5 shows the flat-shaded HIV model and its analysis models corresponding to five different mesh resolutions. In this analysis, we fixed the shrinkage scale as 0.5 and the bin size as 2. With varied mesh resolution levels, our fragile watermark was embedded into each model to test the effect of the mesh resolution on the watermark correlation value. In addition, we ran each test five times using different keys and reported the median value. Fig. 6(a) shows the effect of different mesh resolutions on the watermark correlation value. Note that the mesh resolution of zero in Fig. 6(a) indicates that the original models were not subjected to the mesh resolution control algorithm. Obviously, the curves shown in Fig. 6(a) confirm our two hypotheses. Furthermore, a polygonal mesh with higher mesh resolution would possess higher capacity for watermarking.

In order to investigate how the shrinkage scale can force a compromise between transparency and capacity, a suitable visual metric was needed to evaluate the difference between the original model and the watermarked model. In the literature [9], a Laplacian-based visual metric has frequently been used to capture human visual perceptibilities, such as smoothness. We, therefore, adopted this visual metric and measured the transparency as the Peak Signal to Noise Ratio $\mathrm{PSNR} = 20\log_{10}(\mathrm{peak}/diff)$, where peak means the longest edge of the object's bounding box and $diff$ is the Laplacian-based visual metric used in [9]. In this analysis, we picked five models that were at the fourth resolution. We chose the bin size and the shrinkage scale as 2 and 0.5, respectively. With various shrinkage scales, our fragile watermark was embedded into each model for transparency and capacity tests. In the same way, we ran each test five times using different keys and reported the median value. Figs. 6(c)-6(d) show the effects of different shrinkage scales on the watermark correlation value and PSNR value, respectively. From Figs. 6(c)-6(d), it is clear that the best choice of shrinkage scale is 0.5.

In order to demonstrate how robust our watermark is, we attacked the embedded watermark by means of randomization of vertex coordinates. To simulate such attacks, randomization of vertex coordinates was controlled by means of the noise strength, which is defined as the ratio of the largest displacement to the longest edge of the object's bounding box. In this analysis, we picked five models with the largest resolution level from the set of analysis models and fixed the shrinkage scale at 0.5. With various bin sizes, our watermark was embedded into each model and then attacked using different noise strengths in robustness tests.

In the same way, we ran each test five times using different keys and reported the median value. Fig. 6(b) shows the results of robustness tests using different bin sizes for the HIV-lv5 model. From these plots, it can be seen that a larger bin size can provide a hidden watermark with higher robustness. However, the drawback is that the false-alarm rate is increased as well.

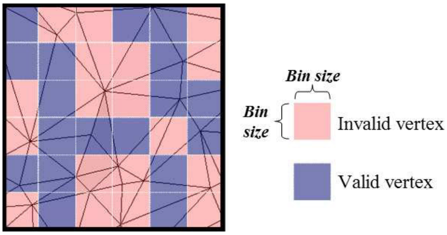


Fig. 4. Irregular polygonal mesh superimposed on uniform parameterization.

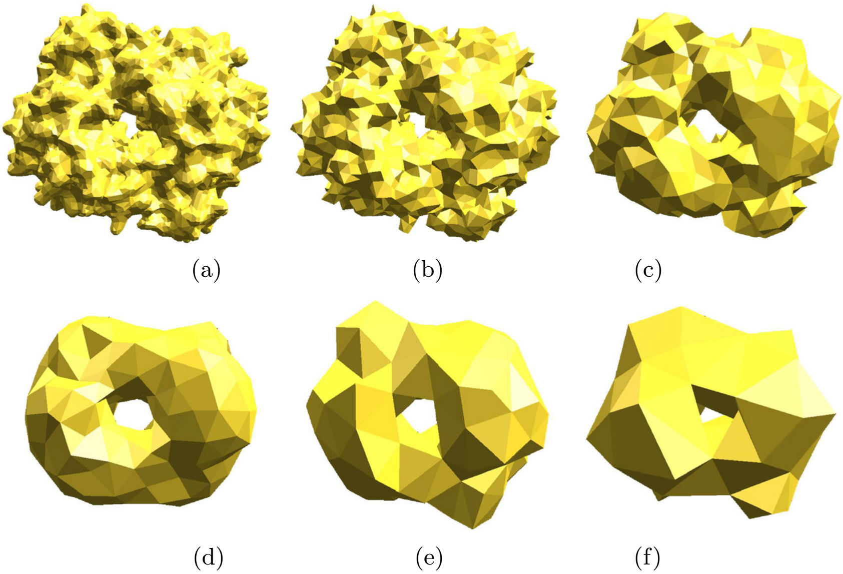
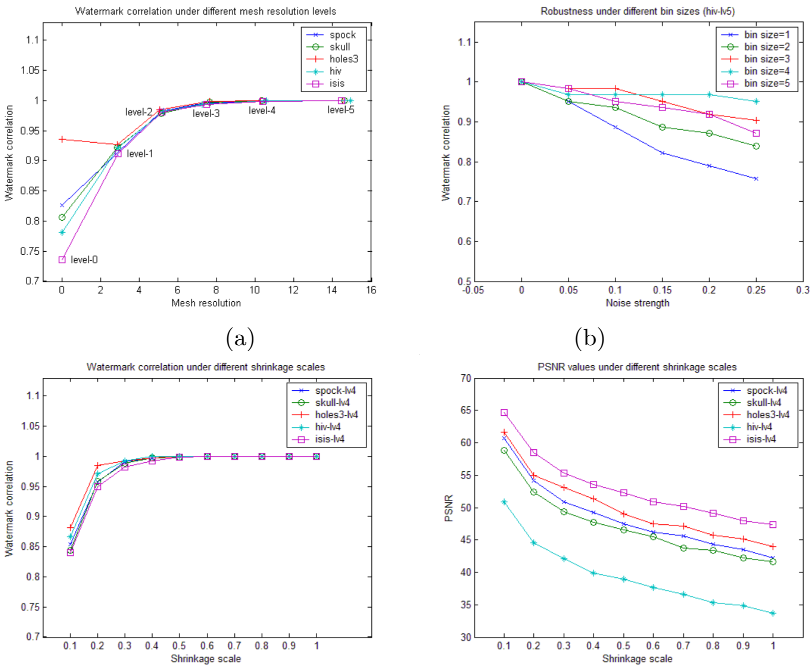


Fig. 5. Analysis models for the HIV protease surface model: (a) original HIV model; (b) HIV-lv1 model; (c) HIV-lv2 model; (d) HIV-lv3 model; (e) HIV-lv4 model; (f) HIV-lv5 model.



(a) (b)

(c) (d)

Fig. 6. Analysis on our authentication scheme for 3-D polygonal meshes: (a) effect of mesh resolution on the watermark correlation value; (b) robustness under different bin sizes for the HIV-lv5 model; (c) effect of shrinkage scale on the watermark correlation value; (d) effect of shrinkage scale on the trnsparency of our fragile watermark.

4 Experimental Results

A series of experiments were conducted to test the performance of the proposed fragile watermarking method. We shall start with parameter selection and then report quantitatively some experimental results. In addition, we shall present a set of visualization results that can demonstrate the power of the proposed method in distinguishing malicious attacks from incidental modifications.

4.1 Selecting Appropriate Parameters

We have reported in Sec. 3 that several parameters were needed during watermark embedding and detection. These parameters included a binary watermark pattern, a set of look-up tables, a basis for parameterization, and the degree of quantization. All of the parameters used in our experiments were set as follows. A binary watermark pattern with a size of 512×512 (as indicated in Fig. 7) was used in our experiments. That means, $WM\_X\_SIZE = WM\_Y\_SIZE = 512$. In addition, a set of look-up tables were generated and protected by one authentication key. The size of each table was 256. Therefore, $LUT\_SIZE = 256$.

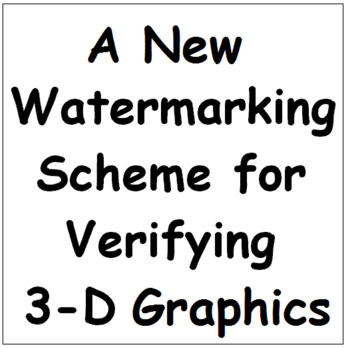


Fig. 7. The binary watermark pattern used in our experiments.

Table 1. A list of five triangulated meshes used in our experiments and their watermark correlation values detected using the proposed method.

Model	Number of Vertices/Faces	Watermark Correlation Value
dolphins	855/1692	1
spock	16386/32768	0.953558
mannequin	711/1418	0.998594
holes3	5884/11776	0.985214
HIV	9988/20000	0.900681

As to the basis for parameterization, since the 3-D vertex space is periodically aggregated into binary state spaces, its selection is not crucial to the proposed method. Therefore, we fixed the basis as $m(0, 0, 0)$ and $n(1, 0, 0)$ in the experiments. As for appropriate quantization steps, we selected them empirically. We assigned the ordinary numeric value, $b = 1$, in all the experiments.

4.2 Experimental Results on Authentication

The data set used in our experiments was a set of triangulated meshes, listed in Table 1. Each of them was watermarked using our fragile watermarking method presented in Sec. 3. The last column in Table 1 shows the watermark correlation values for the five different models. From the experimental results, it is easy to see that at least 90 percent of the vertices became valid after the proposed embedding method was applied.

The five test models were watermarked and tested to evaluate the robustness against reduction of floating-point precision. The results of this experiment are shown in Fig. 8, where the precision of a floating-point number is specified by a nonnegative decimal integer preceded by a period (.) and succeeded by a character f. It is clearly shown in Fig. 8 that the proposed method is very robust against reduction of floating-point precision up to ten to the minus three.

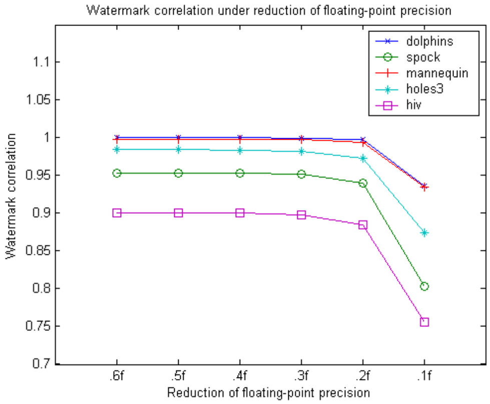


Fig. 8. Five test models were watermarked and tested to evaluate the robustness against reduction of floating-point precision.

4.3 Visualization of Authentication Results

Visualization is a good way to ”see” whether the proposed watermarking method is valid or not. Fig. 9 shows that the original and the watermarked Spock models were rendered as either wireframe or flat-shaded models, respectively. It can be seen that the watermarked model maintained high correlation with the original model, whether in a wireframe format or in a flat-shaded format.

The results of experiments on detecting malicious attacks are shown in Figs. 10-11. Fig. 10(a) shows that the watermarked Spock model was tampered with by stretching out Spock’s nose. Fig. 10(b) shows some detected potentially modified regions before the closing operator was applied. Note that approximately 50 percent of vertices on Spock’s nose were identified as invalid vertices, as shown in Fig. 10(b). Therefore, in order to amplify the effect of the authentication results, the morphological operators described in [15] were adopted so that the parts being tampered with in a model could be detected and highlighted. Fig. 10(c) shows the authentication results of Fig. 10(b) after some morphological operations were applied. Fig. 11 shows another example of malicious tampering, which could possibly occur in the real world. In this case, it is not obvious that the two dolphins were tampered with. Nevertheless, the proposed method still succeeded in malicious tampering detection. As shown in Fig. 11(d), among the two dolphins that were tampered with, one was translated, and the other one stretched out. Both attacks were detected and highlighted.

5 Conclusion

A new fragile watermarking scheme which can be applied to authenticate 3-D polygonal meshes has been presented in this paper.Watermarks are embedded using a local mesh parameterization technique and can be blindly extracted for

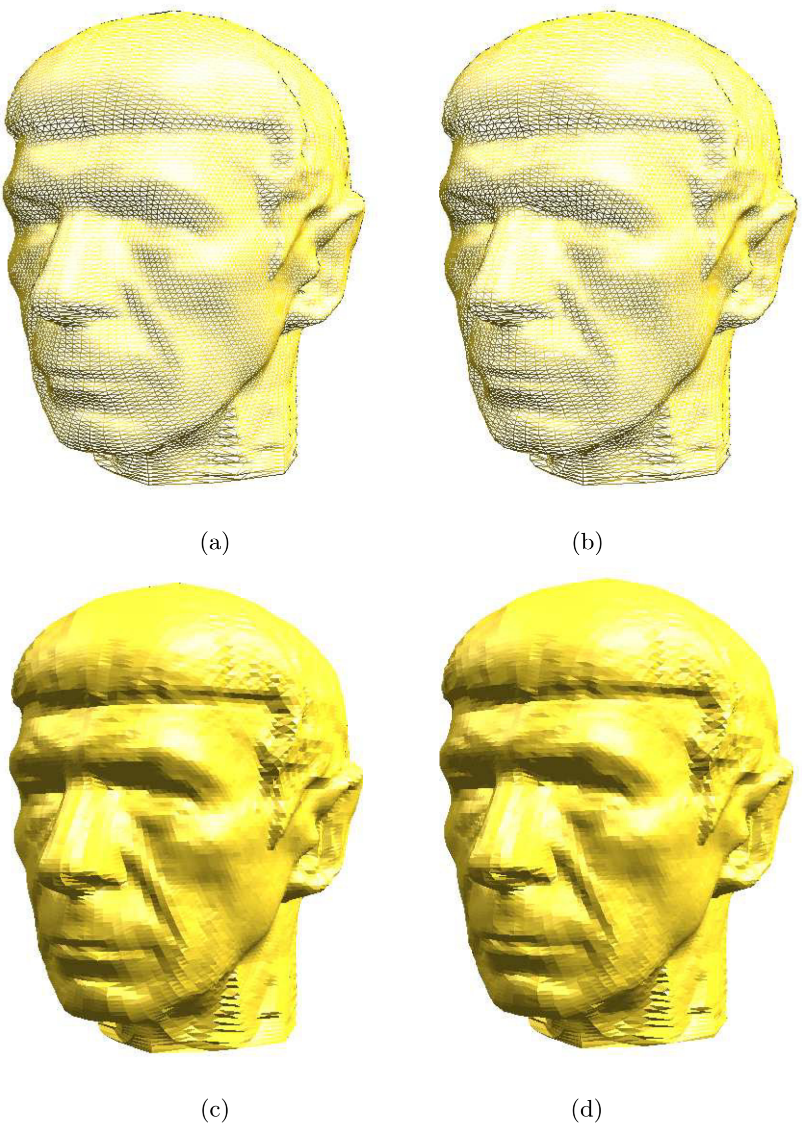


Fig. 9. Visualization of the transparency test: (a) the original Spock model rendered in a wireframe format; (b) the watermarked Spock model rendered in a wireframe format; (c) the original Spock model rendered in a flat-shaded form; (d) the watermarked Spock model rendered in a flat-shaded form.

authentication applications. The proposed scheme has three remarkable features: (1) the domain of allowable alternation for a vertex is explicitly defined by two well-designed hash functions; (2) region-based tampering detection is achieved by a vertex-order-independent embedding process; (3) fragile watermarking is

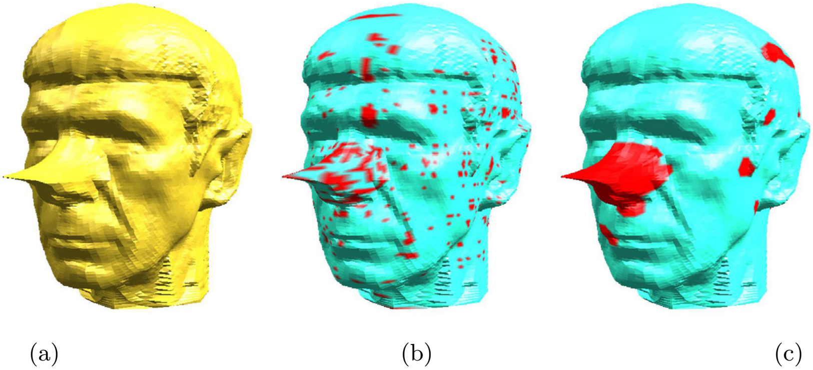


Fig. 10. Region-based tampering detection: (a) the watermarked Spock model was tampered with by stretching out its nose; (b) the detected potentially modified regions (before morphological operators were applied); (c) the detected modified regions after the morphological operators were applied.

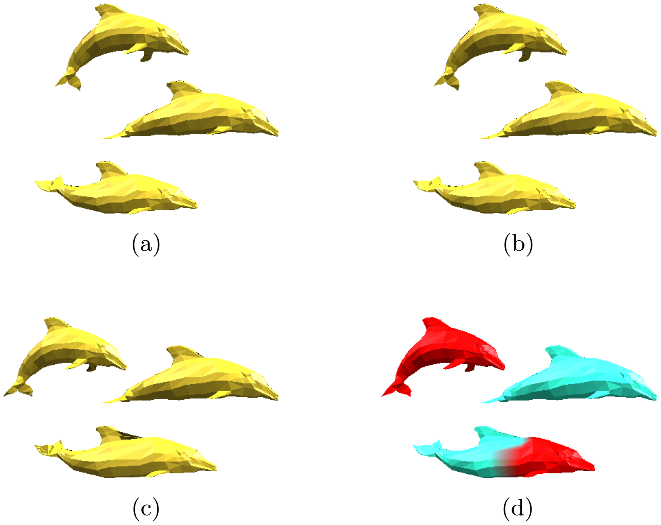


Fig. 11. Malicious tampering detection: (a) the original dolphins model; (b) the watermarked dolphins model; (c) a slightly modified dolphins model; (d) two out of the three dolphins have been tampered with. The maliciously modified dolphins were effectively detected.

achieved for the detection of malicious modification and tolerance of incidental manipulations. To the best of our knowledge, this is the first 3-D mesh authentication scheme that can detect malicious attacks involving incidental modifications.

Acknowledgements. This work was partially supported by DigiBits Interactive, Inc. Polygonal meshes used in this paper were provided courtesy of the University of Washington and Cyberware. For the use of the HIV protease surface model, we would like to thank Arthur Olson, The Scripps Research Institute. Part of our implementation used the triangle facet data structure and code of Henning Biermann and Denis Zorin at the Media Research Lab of the University of New York.

References

1. O. Benedens, Geometry-Based Watermarking of 3-D Models, *IEEE Computer Graphics and Applications*, Vol. 19, pp. 46–45, 1999.
2. F. Cayre and B. Macq, Data Hiding on 3-D Triangle Meshes, *IEEE. Trans. Image Processing*, Vol. 51, pp. 939–949, 2003.
3. F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, and H. Maître, Application of Spectral Decomposition to Compression and Watermarking of 3-D Triangle Mesh Geometry, *Signal Processing: Image Communication*, Vol. 18, pp. 309–319, 2003.
4. M. S. Floater, Parameterization and Smooth Approximation of Surface Triangulations, *Computer Aided Geometric Design*, Vol. 14, pp. 231–250, 1997.
5. C. Fornaro and A. Sanna, Public Key Watermarking for Authentication of CSG Models, *Computer-Aided Design*, Vol. 32, pp. 727–735, 2000.
6. A. Johnson and M. Hebert, Using Spin-Images for Efficient Multiple Model Recognition in Cluttered 3-D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21, pp. 433–449, 1999.
7. —, Control of Polygonal Mesh Resolution for 3-D Computer Vision, *Graphical Models and Image Processing*, Vol. 60, pp. 261–285, 1998.
8. S. Kanai, H. Date, and T. Kishinami, Digital Watermarking for 3-D Polygons Using Multiresolution Wavelet Decomposition, in *Proc. Sixth IFIP WG 5.2 GEO-6*, Tokyo, Japan, 1998, pp. 296–307.
9. Z. Karni and C. Gotsman, Spectral Compression of Mesh Geometry, in *Proc. SIGGRAPH*, New Orleans, Louisiana, 2000, pp. 279–286.
10. A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, MAPS: Multiresolution Adaptive Parameterization of Surfaces, in *Proc. SIGGRAPH*, Orlando, Florida, 1998, pp. 95–104.
11. R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, *IEEE J. Select. Areas in Commun.*, Vol. 16, pp. 551–560, 1998.
12. R. Ohbuchi, S. Takahashi, and T. Miyazawa, Watermarking 3-D Polygonal Meshes in the Mesh Spectral Domain, in *Proc. Graphics Interface*, Ontario, Canada, 2001, pp. 9–17.
13. E. Praun, H. Hoppe, and A. Finkelstein, Robust Mesh Watermarking, in *Proc. SIGGRAPH*, Los Angeles, CA, 1999, pp. 154–166.
14. E. Praun, W. Sweldens, and P. Schröder, Consistent Mesh Parameterizations, in *Proc. SIGGRAPH*, Los Angeles, CA, 2001, pp. 179–184.
15. C. Rössl, L. Kobbelt, and H. P. Seidel, Extraction of Feature Lines on Triangulated Surfaces Using Morphological Operators, in *Symposium Smart Graphics*, Stanford University, 2000.
16. K. Yin, Z. Pan, S. Jiaoying, and D. Zhang, Robust Mesh Watermarking Based on Multiresolution Processing, *Computers and Graphics*, Vol. 25, pp. 409–420, 2001.

17. B. L. Yeo and M. M. Yeung, Watermarking 3-D Objects for Verification, *IEEE Computer Graphics and Application*, Vol. 19, pp. 36–45, 1999.
18. M. M. Yeung and B. L. Yeo, An Invisible Watermarking Technique for Image Verification, in *Proc. Into'l Conf. Image Processing*, Piscataway, N.J., 1997, Vol. 2, pp. 680–683.
19. M. G. Wagner, Robust Watermarking of Polygonal Meshes, in *Proc. Geometric Modeling and Processing*, Hong Kong, China, 2000, pp. 201–208.
20. W. Sweldens and P. Schröder, Course 50: Digital Geometry Processing, SIGGRAPH'2001 Course Note, 2001.