Computer-Aided Design 40 (2008) 634–645

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

A numerically stable fragile watermarking scheme for authenticating 3D models

Wei-Bo Wang [a,b,*], Guo-Qin Zheng [a,b], Jun-Hai Yong [a,b], He-Jin Gu [c]

[a] School of Software, Tsinghua University, Beijing, PR China

[b] Key Laboratory for Information System Security, Ministry of Education of China, Beijing 100084, PR China

[c] Jiangxi Academy of Sciences, Nanchang 330029, PR China

ARTICLE INFO

Article history:
Received 4 January 2008
Accepted 18 March 2008

Keywords:
Fragile watermarking
Mesh authentication
Error prevention
Tamper detection

ABSTRACT

This paper analyzes the numerically instable problem in the current 3D fragile watermarking schemes. Some existing fragile watermarking schemes apply the floating-point arithmetic to embed the watermarks. However, these schemes fail to work properly due to the numerically instable problem, which is common in the floating-point arithmetic. This paper proposes a numerically stable fragile watermarking scheme. The scheme views the mantissa part of the floating-point number as an unsigned integer and operates on it by the bit XOR operator. Since there is no numerical problem in the bit operation, this scheme is numerically stable. The scheme can control the watermark strength through changing the embedding parameters. This paper further discusses selecting appropriate embedding parameters to achieve good performance in terms of the perceptual invisibility and the ability to detect unauthorized attacks on the 3D models. The experimental results show that the proposed public scheme could detect attacks such as adding noise, adding/deleting faces, inserting/removing vertices, etc. The comparisons with the existing fragile schemes show that this scheme is easier to implement and use.

1. Introduction

With the rapid development of internet technologies and the growth of multimedia contents, transferring digital media data via internet becomes more and more popular in recent years. These valuable data, however, are easily illegally distributed or tampered. Thus, it is desirable to develop a strong method to protect the copyright and the integrity of the digital data [10,11,29].

Watermarking has been viewed as a solution for the requirement above [1,2,8,16]. Compared with other possible solutions, such as encryption and digital signature, watermarking costs less and is computationally cheaper [34]. According to the application purpose, watermarking can be divided into the robust watermarking and the fragile watermarking. The robust watermarking focuses on protecting the ownership of digital data while the fragile watermarking intends to authenticate the integrity of digital data [10]. In general, the robust watermarking should tolerate heavy attacks and keep detectable in the extracting stage. On the other hand, the fragile watermarking is aimed to detect the slightest unauthorized attacks and to locate where the attacks happen. Both the robust and the fragile watermarking should be perceptual invisibility [1]. Another classification criterion divides watermarking into public and private according to whether the original media is needed during the extracting stage [10,11]. A private watermarking scheme references the original media to extract the watermarked information while a public scheme does not need the original media. In the fragile watermarking, the public schemes are preferred. It is because fragile schemes have to encrypt and transfer the original models for extracting the watermarks. And the encrypting and the transferring steps are costly [34]. Finally, according to the insertion domain, researchers classify the watermarking schemes into the spatial domain and the spectral domain.

Currently, there exist several watermarking schemes on different kinds of media, such as still images, video and audio [1–7,16]. 3D models are relatively new media forms on World Wide Web for many applications, such as CAD/CAM. Some watermarking schemes have been proposed to watermark the 3D models [8–15,17–21,26,27,29,34,37]. These schemes are primarily on meshes, which are considered as the "lowest common denominator" of 3D surface representations - it is easy to convert other representations to meshes [12]. Ohbuchi listed several embedding primitives in the spatial domain for 3D watermarking embedding [8], such as coordinates of a point, length of a line, etc. For the consideration of robustness, the 3D features, which are invariant to common transformation, are used to embed the watermarks. These features include the angle between polygons [41], the length of lines, the radius of curvatures, the distance to the mesh center [23,24,27,36,37], normal vectors [11,39,40], and the relative position of a vertex to its 1-ring neighbors [10,29,34,35]. Some schemes transform the

* Corresponding author at: School of Software, Tsinghua University, Beijing, PR China. Tel.: + 86 010 62778801.
E-mail address: wwb84nju@gmail.com (W.-B. Wang).

0010-4485/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2008.03.001

models into the spherical coordinate systems for robustness [26,31,42]. Compared with the spatial domain algorithms, the spectral domain algorithms are more robust and preserve the visual effect of the models better. It is because the inserted watermark bits are diluted in all the spatial parts of the models. Some commonly used spectral analysis tools include Laplacian basis functions [22,25,28,38], radial basis functions [33] and multi-resolutions such as wavelet [9,26,32]. The drawbacks of the spectral domain algorithms are that they are complicated and computationally expensive compared to the spatial domain algorithms.

In 3D fragile watermarking, Yeung and Yeo [10] slightly perturb the vertices such that a certain hash function of each vertex's coordinates matches another hash function applied to the centroid of its neighboring vertices. Lin et al. [29] apply two different hash functions on the coordinates of vertices and perturb the vertices until the two functions are matched. In [27], Wu and Cheung quantize the distance between the centroid of the mesh models and each surface to embed the watermarks. Chou and Tseng [34] apply multi-function and adjusting-vertex method to embed the watermarks. Despite their reasonable success, these schemes suffer from several drawbacks. Among them the numerically instable problem, to our best knowledge, has not been mentioned in any reported literature.

The remaining sections are organized as follows. Section 2 analyzes the problems encountered in these schemes, primarily the numerically instable problem. In Section 3 we propose our scheme, which is numerically stable and can handle the problems mentioned. The discussion of the performance and the experimental results are provided in Section 4. Finally, conclusions are given in Section 5.

2. Problems in previous work

In this section, we analyze the problems in the existing schemes, particularly the numerically instable problem, and give possible solutions to them. A watermarking algorithm, combined the solutions together, is proposed in Section 3.

Numerically Instable Problem: The numerically instable problem is accounting for the majority of computational errors in CAD systems [30]. Currently this problem also arises in some fragile 3D watermarking schemes. However, as we known no literature has reported this.

The fragile watermarking schemes must compare the extracted bits with the original ones in order to determine if the integrity of the model has been destroyed. If the two bits are not equal, the schemes can demonstrate unauthorized changes. Watermarking calculations are primarily based on the floating-point arithmetic [34] or on the integral arithmetic [10,27,29,37]. If the operations are based on the floating-point arithmetic, the schemes need to find a proper tolerance to equal for comparing. If the tolerance is too small, the bits to be compared will be determined as not equal, even if no attack has happened. However, if the tolerance is too large, the attacks will not be found since the bits are determined as equal. Unfortunately, the tolerances differ from one embedding operation to another, which means that every time the schemes embed a watermark bit, they have to calculate a new tolerance. This means that the number of calculating the tolerance equals the number of the embedding operations. It is expensive. Furthermore, these tolerances have to be attached to the model and transferred together. This increases the transferring cost remarkably since the number of tolerances equals the number of embedded vertices. It is unreasonable only using the maximum tolerance as a criterion. It will enlarge the tolerance at most cases in which some unauthorized changes may not be detected.

The method in [34] does not consider selecting proper tolerances to equal. When no tolerance to equal is given, in practice three methods are commonly used to compare two floating-point numbers. The first one is using a very small tolerance, such as 1e-10. The second one is splitting the decimal fracture of the two floating-point numbers and comparing them as integers. The third one is rounding-off the two floating-point numbers and comparing the rounded integers. However, even when no attack happens, the method in [34] fails to determine whether the original bits and the extracted bits are equal using these methods because of the numerically instable problem.

Two formulae and several parameters are used in [34] for embedding and extracting the watermarks. The formulae are

$$x' = \begin{cases} x - (|x - x^c| \bmod k^q) + \dfrac{k^q}{k^d} w & \text{if } x > x_c, \\ x + (|x - x^c| \bmod k^q) - \dfrac{k^q}{k^d} w & \text{otherwise,} \end{cases}$$

and

$$w' = (|x^c - x'| \bmod k^q) \frac{k^d}{k^q}.$$

In the formulae, x indicates one of the three coordinates of a vertex, x^c indicates the centroid of the neighboring vertices of x, k^q and k^d are the embedded parameters, w is the watermark bit before embedding and w' is the extracted bit. The two formulae could guarantee $w = w'$ based on the fact that $(a - b - (a - b) \bmod c) \bmod c = 0$, where $a \geq b$ and $a, b, c \in P$. The method in [34] applies the first formula twice to embed w and h into x_2 and x_3 of a vertex respectively. x_2 and x_3 are the second and the third parts of the coordinates of a vertex. The relationship between w and h is that $h = \mathit{hash\,function}(w)$. The relationship $h' = \mathit{hash\,function}(w')$ should still be kept because the formulae can guarantee that the extracted bits are equal to the original bits. However, this guarantee may not be held when x, w and other parameters are floating-point numbers because of the numerically instable problem. Table 1 gives the data of an example that violates this guarantee. In Table 1, x_2 and x_3 are the second and the third parts of the coordinates of the vertex. As the two embedded bits keep the relationship $h = h(w)$, the two extracted bits, w' and h' should still be equal. In Table 1, however, the two extracted bits are 2.7847 and 1.9359 when no attacks happen. Neither one of the three common used methods can be used to determine the two extracted bits equal. The method in [34] fails here since it does not give the tolerances to equal, and the further reason is the numerically instable problem.

The best way to deal with errors is to prevent them from happening [30]. Thus, selecting watermarking schemes based on the integral arithmetic is a cheap and easy way to settle the numerically instable problem. But, even if the schemes are based on the integral arithmetic, the floating-point numbers have to be converted into integers first because 3D models are mostly represented in the floating-point numbers. A common converting method is splitting the decimal fracture part of a floating-point number and the schemes in [10,27,29,37] apply this method. However, this method has drawbacks in practice. Several attacks, such as adding noise, may also only change the decimal fracture part of the coordinates of vertices. These attacks may not be detected since the modified parts have been discarded during the converting operation before detecting. Therefore, designing a proper converting method is important for improving the stability of 3D fragile watermarking algorithms. A possible solution is to view the mantissa part of the floating-point number as an unsigned integer since it does not lose any original information.

Causality Problem: This problem first arises in Yeo's scheme [10]. The method in [10] orderly perturbs the vertices until they match the predefined relationship with the centroid of their neighboring vertices. However, the latter perturbed vertices, which are the