A Sequential Quantization Strategy for Data Embedding and Integrity Verification

Yiu-ming Cheung, *Senior Member, IEEE*, and Hao-tian Wu, *Student Member, IEEE*

***Abstract*—Quantization-based embedding has been used for integrity verification in semi-fragile watermarking. However, some of the illegal modifications cannot be detected in the normal quantization-based methods, especially when the host values (i.e., the values chosen in a host signal for data embedding) are independent from each other. In this paper, a sequential quantization strategy (SQS) is proposed to make the modulation of a host value dependent on a certain number of the previous ones. Therefore, a balance between security improvement and tamper localization can be achieved for integrity verification. Furthermore, the proposed SQS is incorporated with a reversible data hiding mechanism. A new watermarking algorithm is then generated for mesh authentication. The experimental results show that the chance to detect illegal modifications is increased by adopting the SQS while the property of reversibility is achieved.**

***Index Terms*—Fragile watermarking, mesh authentication, reversible data hiding, sequential quantization strategy (SQS).**

I. Introduction

TO PREVENT the unauthorized modifications of multimedia data, such as digital images, video and audio clips, 3-D models, and so forth, one major challenge is to verify their authenticity and integrity. Traditionally, data authentication is performed by digital signature schemes like RSA and DSA [1] that append the file with a hash-based signature, which is encrypted with a private key and decrypted with the corresponding public key. In the cryptographic algorithms, a single bit error will lead the generated value to vary so that illegal modifications can be detected. However, extreme sensitivity to data alterations is not always suitable for multimedia data, where some information loss without noticeable quality degradation is tolerable, e.g., the lossy compressions like JPEG, MPEG, and MP3 for digital images, video and audio, respectively. Furthermore, the nonmalicious manipulations that preserve the perceptual effect of media content are often distinguished from the malicious ones in the common use. In addition, it is also desirable to estimate the strength of a tamper and localize its position from the practical point of view. Unfortunately, traditional cryptographic algorithms cannot meet the aforementioned requirements.

Manuscript received October 12, 2006; revised February 8, 2007. This work was supported by the Faculty Research Grant of Hong Kong Baptist University under Project FRG/06-07/II-07 and in part by the Research Grant Council of Hong Kong under Grants HKBU 2156/04E and HKBU 210306. This paper was recommended by Associate Editor Q. Sun.

The authors are with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (e-mail: ymc@comp.hkbu.edu.hk; htwu@comp.hkbu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2007.903553

As a branch of information hiding [2], digital watermarking [3] has been proposed for a variety of applications. In general, it can be modeled as a constrained communication problem with side information [4], [5]. In this paper, we concentrate on fragile watermarking that can be used for integrity verification of media content. Its procedure can be described as follows: Initially, a tamper-proof watermark (e.g., a string of bit values) is embedded into an original object $\mathbf{O}$, which is called *host signal* interchangeably (e.g., a digital image, video, audio, 3-D geometry, etc). The process of watermark embedding should be imperceptible, i.e., without introducing noticeable distortion to the watermarked object $\mathbf{O}'$. In authentication applications, the embedded watermark should be extractable from $\mathbf{O}'$ without any information regarding $\mathbf{O}$, while vulnerable to illegal changes of $\mathbf{O}'$. By comparing the extracted watermark with the embedded one, illegal modifications on $\mathbf{O}'$ can be detected. In addition, the capability of tamper localization may be achieved by exploiting the inherent properties of the host signal, e.g., see the block-wise algorithm [6] for digital images. As most of the existing data embedding algorithms introduce the irremovable distortion, it is desirable to recover the original object in the authentication process. Referred as reversible, lossless, distortion-free, or invertible data hiding [7], the property of reversibility is particularly preferred in the high-precision applications. It can be seen that fragile watermarking provides a promising approach to multimedia authentication.

In practice, fragile watermarking is performed by modifying each value, e.g., a pixel value in a digital image or a coefficient value in the transform domain, chosen from the host signal for data embedding. Such a value is called *host value* for short hereinafter. The existing embedding methods are mainly based on least significant bits (LSB) replacement (e.g., [6], [8]–[10]), look-up table (LUT) (e.g., [11], [12]), or quantization (e.g., [13]–[15]). A few reviews on fragile watermarking for image authentication have been reported (e.g., see [16], [17]). In the following, we will give a brief review along the aforementioned embedding methods on fragile watermarking of digital images and 3-D mesh models.

The LSB replacement is to replace the LSBs of the chosen host values, e.g., the pixel values of digital images in the checksums algorithm [8], by the watermark. The public-key cryptography can be combined with LSB replacement to equip the authentication with cryptographic strength (e.g., see [6], [10]). In that case, the embedded data will be sensitive to any alteration, thus suitable for strict authentication.

In the LUT-based embedding, a host value is modified according to one or more LUTs so that a bit value can be embedded. Compared with the LSB replacement, the watermark embedded based on LUT is more intensive so that the error dif-

fusion process is employed as shown in [11]. Further, the discrete cosine transform (DCT) coefficients after quantization can be modified to detect and localize the alterations of the watermarked image, e.g., see [12]. As a general embedding technique, LUT embedding is also used to embed a fragile watermark in a 3-D object [18]. By adopting the method in [18], Lin *et al.* [19] make the embedded watermark robust to vertex reordering by improving the mapping from vertex positions to location indices. Since the LUTs perform as secret keys in both of data embedding and extraction, the security of the LUT-based embedding relies on the difficulty of inferring them, as detailed in [20]–[22]. In a more recent work [23], a distortion compensated LUT embedding offers joint enhancement of security and robustness under the additive white Gaussian noise.

In the class of quantization-based embedding, a set of host values are modulated based on quantization to embed a semi-fragile watermark, i.e., the embedded watermark is fragile to illegal modifications, whereas robust to the nonmalicious processing by elaborately choosing the host values for modulation. As shown in [13], a semi-fragile watermarking algorithm accepting JPEG lossy compression, meanwhile rejecting malicious attacks, is proposed for digital images by exploiting two invariant properties of the DCT coefficients. In [14], a watermark is embedded by quantizing the coefficients in the Haar wavelet domain so that tamper detection in the localized spatial and frequency regions is possible. This capability helps to provide information on how the watermarked image is modified, therefore called as telltale tamper-proofing. In [15], an edge-based message digest is inserted into the coefficients in the SPIHT compression domain [24] using a sort of odd-even embedding, which is the simplest example of QIM [5], so that small distortions can be tolerated while malicious modifications are detected. Moreover, the quantization-based methods are also used for mesh authentication in [26]–[30] to allow some content-preserving manipulations while detecting the malicious ones. Besides, high information rate is achievable for 3-D triangle meshes, e.g., see [28].

Although quantization-based embedding has been used for integrity verification by modulating a set of host values to embed a semi-fragile watermark, some of the illegal modifications cannot be detected, especially when the values chosen for data embedding are independent from each other. In this paper, a sequential quantization strategy (SQS) is therefore proposed to detect illegal modifications more efficiently. With the SQS, the modulation of a host value is dependent on a certain number of the previous ones. Consequently, the change of a modulated host value may alter the data embedded in it and those modulated afterwards. Therefore, the chance to detect the malicious modifications can be increased so that the security of quantization-based embedding for integrity verification is improved. In particular, a balance between security improvement and tamper localization can be achieved by setting the maximum number of the previously modulated values on which one modulation depends. We analyze the condition of detecting the changes of the modulated host values in a special case when the security is emphasized. Also, the condition is compared with the one without adopting the proposed SQS. To make it more suitable for the authentication applications, a reversible data embedding algorithm is further incorporated with the SQS. By choosing the host values as proposed in [30], a new watermarking algorithm is then generated for polygonal meshes. The experimental results show that the integrity of the mesh content can be verified more efficiently while the original mesh model can be recovered from the watermarked one blindly.

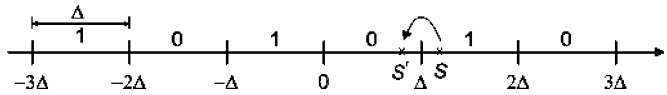


Fig. 1. With the quantization step-size Δ, a bit value $m \in \{0, 1\}$ can be embedded by modifying a scalar value S to the modulated one, denoted as S', within the quantization cell that is associated with m and nearest to S.

The rest of this paper is organized as follows. Section II presents an SQS to improve the security of quantization-based embedding for integrity verification. Then, the proposed SQS is further incorporated with a reversible data hiding mechanism in Section III. In Section IV, a new watermarking algorithm is generated for mesh authentication, and its performance is analyzed and compared with the previous work in Section V. Finally, we draw a conclusion in Section VI.

II. A Sequential Quantization Strategy for Data Embedding

A. Motivation

Quantization has been widely used for data embedding. To address how a bit value $m \in \{0, 1\}$ can be embedded by modulating a scalar value S, a function $B(\cdot)$ is defined with

$$B(i) = i - \left\lfloor \frac{i}{2} \right\rfloor \times 2 \tag{1}$$

where i is an integer and $\lfloor i/2 \rfloor$ represents the floor of $i/2$ so that the output $B(i)$ is a bit value of 0 or 1. For example, if $i = -3$, then $\lfloor i/2 \rfloor = -2$ and $B(-3) = 1$. As shown in Fig. 1, for any scalar value S, we can associate it with a bit value $B(\lfloor S/\Delta \rfloor)$ so that the range of the scalar values is quantized into a set of *quantization cells* with the step size Δ. Subsequently, a bit value m can be embedded by modulating a scalar value S to the modulated one, denoted as S', within a cell that is associated with m. Further, we require that this cell should be nearest to S to minimize the embedding effect. The bit value embedded as shown in Fig. 1 can be extracted by $B(\lfloor S'/\Delta \rfloor)$. If the modulated value S' is changed by δs, we use $S' + \delta s$ to denote the modified value. Then, the change of S' can be detected when $B(\lfloor (S' + \delta s)/\Delta \rfloor)$ is different from $B(\lfloor S'/\Delta \rfloor) = m$, i.e., $B(\lfloor (S' - \lfloor S'/\Delta \rfloor \times \Delta + \delta s)/\Delta \rfloor) = 1$. Otherwise, $B(\lfloor (S' + \delta s)/\Delta \rfloor)$ will be equal to m as long as $B(\lfloor (S' - \lfloor S'/\Delta \rfloor \times \Delta + \delta s)/\Delta \rfloor) = 0$. As shown in the first line of Fig. 2, where a bit value m is embedded in the same way as that in Fig. 1 to generate the modulated value S' at the position of "×," only the modifications causing $S' + \delta s$ in the shadowed cells can be detected by comparing $B(\lfloor (S' + \delta s)/\Delta \rfloor)$ with m.

To detect illegal modifications, the attributes of the host signal can be exploited to choose the host values that are dependent on

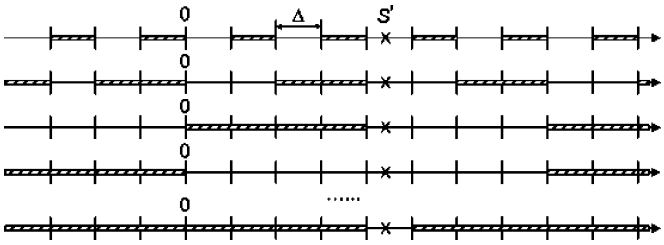


Fig. 2. Suppose a bit value m is embedded in a scalar value S based on odd-even embedding to generate the modulated value S' at the position of "×" in each line. The change of S', as denoted by δs, can be directly detected when $S'+\delta s$ falls into the shadowed cells in the first line. Suppose we further embed three bit values in $X_1+\lfloor S'/2\Delta\rfloor\times\Delta, X_2+\lfloor S'/4\Delta\rfloor\times\Delta, X_3+\lfloor S'/8\Delta\rfloor\times\Delta$ by modifying the scalar values X_1, X_2 and X_3 to X'_1, X'_2 and X'_3, respectively. If the modulated values X'_1, X'_2 and X'_3 are unchanged, the modifications causing $S'+\delta s$ in the shadowed cells as shown in the second to the fourth line can be detected, respectively. In the last line, the modification changing S' from the quantization cell in which it is to the shadowed cells can be detected if a bit value is embedded in $X_i+\lfloor S'/2^i\Delta\rfloor\times\Delta$ by modifying each scalar value X_i to X'_i for $i=1,2,3,\ldots$, and all of the modulated values $\{X'_1, X'_2, X'_3, \ldots\}$ are unchanged.

each other [21]. Although the chance to detect illegal modifications can be increased by choosing such kind of host values, there is still a probability that an illegal change cannot be detected. Therefore, we need to make the modulations of the host values dependent on each other so that illegal modifications can be detected more efficiently, especially when they are independent from each other, such as the pixel values in a digital image and the coordinates in 3-D geometry. In the following, a new quantization strategy will be presented for this purpose.

B. Sequential Quantization Strategy (SQS)

To make the modulation of a host value dependent on another one, one way is to add the previously modulated value, or its representation, to the other host value. In the following, we use $Q(x)$ to denote $\lfloor x/\Delta\rfloor\times\Delta$. Suppose that a bit value m is embedded by modulating a host value S with the quantization step-size Δ as shown in Fig. 1. If the modulated value S' is further changed by δs, the change can be detected by comparing $B(\lfloor(S'+\delta s)/\Delta\rfloor)$ with m only when $B(\lfloor(S'+\delta s)/\Delta\rfloor)\neq B(\lfloor S'/\Delta\rfloor)$. After we add $Q(S'/2)$ to another host value X_1, and embed a bit value m_1 in $X_1+Q(S'/2)$ by modifying X_1, δs can be further detected by comparing $B(\lfloor(X'_1+Q((S'+\delta s)/2))/\Delta\rfloor)$ with m_1 if the modulated value X'_1 is not changed and $B(\lfloor(S'+\delta s)/2\Delta\rfloor)\neq B(\lfloor S'/2\Delta\rfloor)$. We can further embed m_2 in $X_2+Q(S'/4)$ by modifying the host value X_2. If the modulated value X'_2 is not changed and $B(\lfloor(S'+\delta s)/4\Delta\rfloor)\neq B(\lfloor S'/4\Delta\rfloor)$, δs will be detected by comparing $B(\lfloor(X'_2+Q((S'+\delta s)/4))/\Delta\rfloor)$ with m_2. The chance to detect δs will be further increased if we embed m_3 in $X_3+Q(S'/8)$ by modifying the host value X_3. If the modulated values X'_1, X'_2, and X'_3 are unchanged, the modifications causing $S'+\delta s$ in the shadowed cells within the second to the fourth line in Fig. 2 can be detected, respectively. As shown in the last line in Fig. 2, the modification changing S' from the quantization cell in which it is to the shadowed cells will be detected if we embed a bit value m_i in $X_i+Q(S'/2^i)$ by modulating each host value X_i to X'_i for $i=1,2,3,\ldots$, and all of the modulated values $\{X'_1, X'_2, X'_3, \ldots\}$ are unchanged.

To approximate the property as shown in Fig. 2, a new strategy is proposed as follows: Given a host signal from which N host values $\mathbf{X}=\{X_1, X_2, \ldots, X_N\}$ are chosen for data embedding, a new signal $\mathbf{Y}=\{Y_1, Y_2, \ldots, Y_N\}$ can be generated from $\mathbf{X}$ by

$$\begin{cases} Y_1 = X_1 \\ Y_2 = X_2 + Q\left(\frac{X'_1}{2}\right) \\ Y_3 = X_3 + Q\left(\frac{X'_2}{2}\right) + Q(\frac{X'_1}{4}) \\ \ldots \\ Y_{D+1} = X_{D+1} + Q\left(\frac{X'_D}{2}\right) + \ldots + Q\left(\frac{X'_1}{2^D}\right) \\ \ldots \\ Y_{D+i} = X_{D+i} + Q\left(\frac{X'_{D+i-1}}{2}\right) + \ldots + Q\left(\frac{X'_i}{2^D}\right) \\ \ldots \\ Y_N = X_N + Q\left(\frac{X'_{N-1}}{2}\right) + \ldots + Q\left(\frac{X'_{N-D}}{2^D}\right) \end{cases}, \quad (2)$$

where X'_i with $i\in\{1,\ldots,N\}$ is the modulated value of the host value X_i: If we embed a bit value m_i by modulating Y_i to Y'_i, X_i should be modulated by $X'_i=Y'_i-(Y_i-X_i)$. The parameter D in (2) represents the maximum number of the previously modulated host values on which one modulation depends. In this way, a string of N bit values $\mathbf{M}=\{m_1, m_2, \ldots, m_N\}$ can be embedded in the generated signal $\mathbf{Y}$ by modulating the N host values $\{X_1, X_2, \ldots, X_N\}$ in a sequential way. So we call this strategy *Sequential Quantization Strategy* (SQS).

To extract the embedded data, the watermarked signal $\mathbf{Y}'=\{Y'_1, Y'_2, \ldots, Y'_N\}$ needs to be generated from the modulated host values $\mathbf{X}'=\{X'_1, X'_2, \ldots, X'_N\}$ by

$$\begin{cases} Y'_1 = X'_1 \\ Y'_2 = X'_2 + Q\left(\frac{X'_1}{2}\right) \\ Y'_3 = X'_3 + Q\left(\frac{X'_2}{2}\right) + Q\left(\frac{X'_1}{4}\right) \\ \ldots \\ Y'_{D+1} = X'_{D+1} + Q\left(\frac{X'_D}{2}\right) + \ldots + Q\left(\frac{X'_1}{2^D}\right) \\ \ldots \\ Y'_{D+i} = X'_{D+i} + Q\left(\frac{X'_{D+i-1}}{2}\right) + \ldots + Q\left(\frac{X'_i}{2^D}\right) \\ \ldots \\ Y'_N = X'_N + Q\left(\frac{X'_{N-1}}{2}\right) + \ldots + Q\left(\frac{X'_{N-D}}{2^D}\right). \end{cases} \quad (3)$$

If we embed a bit value m_i in Y_i as shown in Fig. 1, the embedded value m_i can be extracted by $B(\lfloor Y'_i/\Delta\rfloor)$. As shown in (3), the embedded value m_i is dependent on X'_i and at most the last D modulated host values at most before X'_i. In other words, the change of X'_i will affect the values of Y'_i and at most the next D ones after Y'_i. If we denote the modified value of Y'_k as $\hat{Y}_k$, the change of X'_i may be detected by comparing $B(\lfloor\hat{Y}_k/\Delta\rfloor)$ with m_k for at most $D+1$ times, i.e., $k\in\{i, i+1, \ldots, i+D\}$ if $i+D\leq N$. If the proposed SQS has not been used, the change of X'_i can only be detected by comparing $B(\lfloor X'_i/\Delta\rfloor)$ with m_i. That is, it becomes harder for an adversary to make the change of the modulated host values $\mathbf{X}'$ undetectable after adopting the SQS. The security of quantization-based embedding for integrity verification is improved in turn.

The capability of tamper localization has been taken into account in the proposed SQS by employing a parameter D in (2) to limit the range that a modification may affect, as shown in (3). To realize the tamper localization within the modulated host values $\mathbf{X}'$, the host values $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ should be ordered according to their neighborhood so that a host value X_i $i \in \{2, \ldots, N\}$, is always within the neighbors of the one prior to it. In this way, a balance between security improvement and tamper localization can be achieved by adjusting the parameter D. Anyway, the tamper can always be localized into the tampered block if the SQS has been used in a block-wise manner (e.g., see [6]). In that case, the set of host values are chosen from an individual block at each time while the block size can be adjusted to achieve a balance between security improvement and tamper localization.

When the security for integrity verification is emphasized, the parameter D should be assigned with $N-1$. Furthermore, the order of the host values $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ can be scrambled with a secret key so that it is much harder for an adversary to make the illegal changes of the modulated host values $\mathbf{X}' = \{X'_1, X'_2, \ldots, X'_N\}$ undetectable. In the following, a theorem will be introduced to analyze in which condition the changes of $\mathbf{X}'$ can be detected when the proposed SQS is adopted.

Theorem 1: By employing the proposed SQS with the parameter $D = N-1$, a bit value m_i is embedded in Y_i generated by (2) whereas it can be extracted from Y'_i in (3) by $B(\lfloor Y'_i/\Delta \rfloor)$ for $i = 1, 2, \ldots, N$. Suppose P out of the modulated host values $\mathbf{X}'$ have been changed and denoted as $\mathbf{X}'_{\mathbf{t}} = \{X'_{t_1}, X'_{t_2}, \ldots, X'_{t_P}\}$ accordingly with the index numbers $\{t_1, t_2, \ldots, t_P\} \subseteq \{1, 2, \ldots, N\}$. If we denote their changes as $\{\delta_1, \delta_2, \ldots, \delta_P\}$, the modifications will be detected if there exists an integer $i \in \{1, 2, \ldots, P\}$ so that

$$B\left(\left\lfloor \frac{X'_{t_i}+\delta_i}{2^n\Delta} \right\rfloor + \left\lfloor \frac{X'_{t_{(i-1)}}+\delta_{(i-1)}}{2^{n+t_i-t_{(i-1)}}\Delta} \right\rfloor + \ldots + \left\lfloor \frac{X'_{t_1}+\delta_1}{2^{n+t_i-t_1}\Delta} \right\rfloor\right) \neq B\left(\left\lfloor \frac{X'_{t_i}}{2^n\Delta} \right\rfloor + \left\lfloor \frac{X'_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta} \right\rfloor + \ldots + \left\lfloor \frac{X'_{t_1}}{2^{n+t_i-t_1}\Delta} \right\rfloor\right) \tag{4}$$

holds for an integer $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$, where t_{i+1} is defined as $N+1$ as $i = P$.

Proof: Suppose there exists an integer $i \in \{1, 2, \ldots, P\}$ so that (4) holds for an integer $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$, where $t_{P+1} = N+1$. Without loss of generalization, we take the case of $n = 0$. By denoting the jth modulated host value that has not been changed as X'_{u_j}, we then have $\{u_1, u_2, \ldots, u_{(t_i-i)}\} \cup \{t_1, t_2, \ldots, t_i\} = \{1, 2, \ldots, t_i\}$. From (3), it can be seen that the value extracted from Y'_{t_i} that is generated from the data extraction process is

$$B\left(\left\lfloor \frac{X'_{t_i}+\delta_i}{\Delta} \right\rfloor + \left\lfloor \frac{X'_{t_{(i-1)}}+\delta_{(i-1)}}{2^{t_i-t_{(i-1)}}\Delta} \right\rfloor + \ldots + \left\lfloor \frac{X'_{t_1}+\delta_1}{2^{t_i-t_1}\Delta} \right\rfloor + \sum_{j=1}^{t_i-i} \left\lfloor \frac{X'_{u_j}}{2^{t_i-u_j}\Delta} \right\rfloor\right)$$

whereas the embedded one m_{t_i} is equals to

$$B\left(\left\lfloor \frac{X'_{t_i}}{\Delta} \right\rfloor + \left\lfloor \frac{X'_{t_{(i-1)}}}{2^{t_i-t_{(i-1)}}\Delta} \right\rfloor + \ldots + \left\lfloor \frac{X'_{t_1}}{2^{t_i-t_1}\Delta} \right\rfloor + \sum_{j=1}^{t_i-i} \left\lfloor \frac{X'_{u_j}}{2^{t_i-u_j}\Delta} \right\rfloor\right).$$

Given (4), the value extract from Y'_{t_i} will be different from m_{t_i}. Therefore, the modifications can be detected by comparing the extracted values with the embedded ones. ■

When the SQS is employed, it can be seen that the changes of the modulated host values $\mathbf{X}'$ as described in Theorem 1 are possibly undetectable if for all $i \in \{1, 2, \ldots, P\}$ and $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$ with $t_{P+1} = N+1$

$$\begin{aligned} &B\left(\left\lfloor \frac{X'_{t_i}+\delta_i}{2^n\Delta} \right\rfloor + \left\lfloor \frac{X'_{t_{(i-1)}}+\delta_{(i-1)}}{2^{n+t_i-t_{(i-1)}}\Delta} \right\rfloor + \ldots + \left\lfloor \frac{X'_{t_1}+\delta_1}{2^{n+t_i-t_1}\Delta} \right\rfloor\right) \\ &= B\left(\left\lfloor \frac{X'_{t_i}}{2^n\Delta} \right\rfloor + \left\lfloor \frac{X'_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta} \right\rfloor + \ldots + \left\lfloor \frac{X'_{t_1}}{2^{n+t_i-t_1}\Delta} \right\rfloor\right). \end{aligned} \tag{5}$$

In contrast, provided that the host values $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ are modulated separately, the changes of $\mathbf{X}'$ in Theorem 1 are undetectable if for all $i \in \{1, 2, \ldots, P\}$

$$B\left(\left\lfloor \frac{X'_{t_i}+\delta_i}{\Delta} \right\rfloor\right) = B\left(\left\lfloor \frac{X'_{t_i}}{\Delta} \right\rfloor\right). \tag{6}$$

In (6), the change δ_i in

$$\left[2b\Delta - X'_{t_i} + \left\lfloor \frac{X'_{t_i}}{\Delta} \right\rfloor \times \Delta, (2b+1)\Delta - X'_{t_i} + \left\lfloor \frac{X'_{t_i}}{\Delta} \right\rfloor \times \Delta\right)$$

for any integer b is undetectable. However, in (5), only δ_i satisfying

$$\begin{aligned} &B\left(\left\lfloor \frac{X'_{t_i}+\delta_i}{2^n\Delta} \right\rfloor - \left\lfloor \frac{X'_{t_i}}{2^n\Delta} \right\rfloor\right) \\ &= B\left(\left\lfloor \frac{X'_{t_{(i-1)}}+\delta_{(i-1)}}{2^{n+t_i-t_{(i-1)}}\Delta} \right\rfloor - \left\lfloor \frac{X'_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta} \right\rfloor + \ldots + \left\lfloor \frac{X'_{t_1}+\delta_1}{2^{n+t_i-t_1}\Delta} \right\rfloor - \left\lfloor \frac{X'_{t_1}}{2^{n+t_i-t_1}\Delta} \right\rfloor\right) \end{aligned}$$

for all $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$ is undetectable. The condition in (5) is more difficult to be satisfied because we have to assign each number within $\{0, 1, \ldots, t_{i+1} - t_i - 1\}$ to n, especially when only a few modulated host values have been changed.

In particular, if there is only one modulated host value X'_i among $\mathbf{X}'$ having been changed by δ_i, the condition that the modification can be detected in Theorem 1 can be simplified to $B(\lfloor (X'_i + \delta_i)/(2^n\Delta) \rfloor) \neq B(\lfloor X'_i/(2^n\Delta) \rfloor)$ for an integer $n \in \{0, 1, \ldots, N-i\}$. If the host values are modulated separately, the modification will be detected if $B(\lfloor (X'_i + \delta_i)/\Delta \rfloor) \neq B(\lfloor X'_i/\Delta \rfloor)$. When i equals to N, the chance to detect δ_i is the same, despite whether the SQS is employed or not. If i is less than N, the value of Y'_k generated in (3) with $k \in \{i, i+1, \ldots, N\}$ might be altered by the modification, as denoted by

$\hat{Y}_k$. By comparing the extracted value $B(\lfloor \hat{Y}_k/\Delta \rfloor)$ with the embedded one m_k for $N-i+1$ times, the chance to detect δ_i will be increased. That is, the chance to detect illegal modifications has been increased by using the SQS.

Actually, the proposed SQS can be used to improve the security of any quantization-based embedding method for integrity verification of digital images, video or audio clips, and so forth. In the following section, a reversible data embedding algorithm will be introduced and incorporated with the SQS for the authentication applications.

III. A Reversible Data Embedding Algorithm

In this section, a reversible data embedding algorithm will be presented. For the sake of clarity, a string of bit values $\mathbf{M} = \{m_1, m_2, \ldots, m_N\}$ will be embedded into an original signal $\mathbf{V} = \{V_1, V_2, \ldots, V_N\}$ consisting of N scalar elements based on odd-even embedding, while the modulation information is further reserved in the watermarked signal $\mathbf{V}' = \{V_1', V_2', \ldots, V_N'\}$.

A. Data Embedding and Modulation Reservation

There are two parts of information that should be contained in the watermarked signal $\mathbf{V}'$. One part is the embedded data, i.e., $\mathbf{M}$, and the other part is the modulation information, which is defined as the difference between the original signal $\mathbf{V}$ and a special signal $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_N\}$ generated from $\mathbf{V}$, $\mathbf{M}$ and the quantization step-size Δ based on odd-even embedding. Further, another signal $\mathbf{E} = \{e_1, e_2, \ldots, e_N\}$ is defined with a parameter a to represent the modulation information by $e_i = (V_i - Z_i)/a$ for $i = 1, \ldots, N$. The detailed procedure to hide the two parts of information is as follows.

Step 1) For $i = 1, \ldots, N$, initialize the integer quotient U_i by $U_i = \lfloor V_i/\Delta \rfloor$. Then, $B(U_i)$ as defined in **(1)** and the remainder R_i are further calculated by

$$\begin{cases} B(U_i) = U_i - \left\lfloor \frac{U_i}{2} \right\rfloor \times 2 \\ R_i = V_i - U_i \times \Delta. \end{cases} \tag{7}$$

It can be seen from the definition of U_i that R_i is nonnegative.

Step 2) A bit value m_i is embedded by

$$Z_i = \begin{cases} U_i \times \Delta + \frac{\Delta}{2}, & \text{if } B(U_i) = m_i \\ U_i \times \Delta - \frac{\Delta}{2}, & \text{if } B(U_i) \neq m_i \& R_i \leq \frac{\Delta}{2} \\ U_i \times \Delta + \frac{3\Delta}{2}, & \text{if } B(U_i) \neq m_i \& R_i > \frac{\Delta}{2} \end{cases} \tag{8}$$

so that $Z_i = \lfloor Z_i/\Delta \rfloor \times \Delta + \Delta/2$ and $B(\lfloor Z_i/\Delta \rfloor) = m_i$. Subsequently, $e_i = (V_i - Z_i)/a$ is obtained.

Step 3) For $i = 1, \ldots, N$, e_i is further added to Z_i by

$$V_i' = Z_i + e_i = \left\lfloor \frac{Z_i}{\Delta} \right\rfloor \times \Delta + \frac{\Delta}{2} + e_i. \tag{9}$$

From (8), it can be seen that the difference between Z_i and V_i, denoted as γ_i, is within the range of $(-\Delta, \Delta]$. Since $e_i = \gamma_i/a$ for $i = 1, \ldots, N$, the one-dimensional signal $\mathbf{E} = \{e_1, e_2, \ldots, e_N\}$ will be distributed within $(-\Delta/a, \Delta/a]$ by choosing a positive value for the parameter a. Further, the parameter a can be assigned with a value greater than 2 so that $e_i \in (-\Delta/2, \Delta/2)$ for $i = 1, \ldots, N$. Thus, the adding of e_i in (9) will not change the embedded value m_i, i.e., $B(\lfloor V_i'/\Delta \rfloor) = B(\lfloor Z_i/\Delta \rfloor) = m_i$.

B. Data Extraction and Recovering the Original Signal

The values of the quantization step-size Δ and the parameter a used in the embedding process are required to extract the embedded data $\mathbf{M}$ and recover the original signal $\mathbf{V}$ from the watermarked one $\mathbf{V}'$. With the quantization step-size Δ, the embedded bit string $\mathbf{M} = \{m_1, m_2, \ldots, m_N\}$ can be extracted from the watermarked signal $\mathbf{V}' = \{V_1', V_2', \ldots, V_N'\}$ by

$$m_i = B\left(\left\lfloor \frac{V_i'}{\Delta} \right\rfloor \right). \tag{10}$$

In practice, the precision of the watermarked signal $\mathbf{V}'$ needs to be taken into account. Suppose $\mathbf{V}'$ is stored at the precision level of 10^{-m}, the round-off error that happens to an element V_i' in $\mathbf{V}'$ is within $(-5 \times 10^{-(m+1)}, 5 \times 10^{-(m+1)}]$. To ensure that the value of $\lfloor V_i'/\Delta \rfloor$ is not affected by the limited precision so that m_i can be correctly extracted, the following condition should be satisfied:

$$0 \leq \frac{\Delta}{2} + e_i \pm 5 \times 10^{-(m+1)} < \Delta \tag{11}$$

which is equivalent to $\Delta/2 - \Delta/a > 5 \times 10^{-(m+1)}$ as $e_i \in (-\Delta/a, \Delta/a]$. Given $a > 2$, the embedded data can be correctly extracted if

$$\Delta > \frac{a}{a-2} 10^{-m} \tag{12}$$

where 10^{-m} is the precision level of $\mathbf{V}'$.

To recover the original signal $\mathbf{V}$, the reserved modulation information $\mathbf{E}$ should be retrieved from the watermarked signal $\mathbf{V}'$. For $i = 1, \ldots, N$

$$e_i = V_i' - \left(\left\lfloor \frac{V_i'}{\Delta} \right\rfloor \times \Delta + \frac{\Delta}{2} \right) = V_i' - Z_i \tag{13}$$

where $Z_i = \lfloor V_i'/\Delta \rfloor \times \Delta + \Delta/2$ can be easily obtained from V_i' with the quantization step-size Δ. For $i = 1, \ldots, N$, an element V_i in the original signal $\mathbf{V}$ can be obtained by

$$V_i = Z_i + e_i \times a. \tag{14}$$

Due to the limited precision, the recovered signal often approximates to the original one. To overcome the round-off error, the precision of $\mathbf{V}'$ needs to be further improved. Hereinafter, we prove that $\mathbf{V}$ can be exactly recovered by storing every element in $\mathbf{V}'$ with one more decimal digit. Suppose $\mathbf{V}$ is stored at the precision level of 10^{-n}. $V_i - Z_i$ will be divided by the parameter a to generate the quotient, which is represented by e_i. To recover an element V_i in $\mathbf{V}$, e_i needs to be retrieved from V_i' in $\mathbf{V}'$, which is assumed at the precision level of $10^{-(n+1)}$. Due to the round-off error, the difference between e_i and the real value of the quotient is within $(-5 \times 10^{-(n+2)}, 5 \times 10^{-(n+2)}]$. Hence, the difference between $e_i \times a$ and $V_i - Z_i$ is within $(-5a \times 10^{-(n+2)}, 5a \times 10^{-(n+2)}]$. Therefore, the value of $e_i \times a$ can be exactly rounded to that of $V_i - Z_i$ at the precision of 10^{-n} if the parameter a is less than 10. Given (12), Z_i can be correctly obtained with the quantization step-size Δ so that an element V_i in $\mathbf{V}$ can be exactly recovered as shown in (14) for $i = 1$,

$\ldots, N$. As shown in (12), the range of the quantization step-size Δ is determined by the parameter a, which is within (2,10).

Since the precision improvement is required for the exact recovery of the original signal, the proposed algorithm is applicable to the media represented by numbers with decimal point, in which one more decimal digit can be appended without changing the data type. In particular, if there exists precision redundancy in the floating point numbers representing the original signal, the file size may be kept unchanged. Compared with trivially appending data to the original signal, the data embedded in the proposed algorithm can be made robust to some manipulations that preserve the watermarked signal. Besides, the distortion of the recovered signal is within a range determined by the parameter a even if the precision of the watermarked signal cannot be improved.

C. Incorporated With the Sequential Quantization Strategy

It is easy to incorporate the reversible algorithm with the SQS proposed in Section II to embed a string of bit values $\mathbf{M} = \{m_1, m_2, \ldots, m_N\}$ by modulating a set of N host values $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ chosen from a host signal. As shown in (2), a new signal $\mathbf{V}$, in replacement of $\mathbf{Y}$, can be generated from $\mathbf{X}$. Initially, V_1 is equal to X_1 and a bit value m_1 is embedded in V_1 as shown in Section III-A to generate the element V_1' in the watermarked signal $\mathbf{V}'$ with the quantization step-size Δ and the parameter a. Then, the host value X_1 is modulated to X_1' by $X_1' = V_1' - (V_1 - X_1)$. For $i = 2, \ldots, N$, V_i can be generated from X_i and the previous modulated host values as shown in (2). By embedding a bit value m_i in V_i, an element V_i' in the watermarked signal $\mathbf{V}'$ is generated and used to modulate the host value X_i to X_i' by $X_i' = V_i' - (V_i - X_i)$. Consequently, the modulated host values $\mathbf{X}' = \{X_1', X_2', \ldots, X_N'\}$ are obtained in a sequential way after a string of bit values $\mathbf{M}$ have been embedded.

In the data extraction process, the watermarked signal $\mathbf{V}' = \{V_1', V_2', \ldots, V_N'\}$ needs to be generated from the modulated host values $\mathbf{X}'$, in the same way as the generation of $\mathbf{Y}'$ in (3). For $i = 1, 2, \ldots, N$, m_i can be extracted by $B(\lfloor V_i'/\Delta \rfloor)$ while V_i can be obtained from V_i' with the quantization step-size Δ and the parameter a used in the embedding process, as shown in Section III-B. Then, the set of host values $\mathbf{X}$ are obtained by $X_i = V_i - (V_i' - X_i')$ so that the host signal can be recovered. The incorporation of the SQS and the reversible algorithm has the advantages of both reversibility and security improvement. To demonstrate its efficacy, we will implement it on 3-D polygonal meshes for authentication purpose in the following section.

IV. New Watermarking Algorithm for Mesh Authentication

With the development of three-dimensional (3-D) scanning, modeling and visualization techniques, more and more 3-D models have been used for geometry representation in 3-D games, video production, and so forth. Among all the types of 3-D models, polygonal meshes are considered as the basic representations of 3-D objects. Basically, a polygonal mesh consists of two parts of information, i.e., geometrical and topological. Given L vertices in a mesh, the mesh geometry can be represented by a set of vertex positions $\boldsymbol{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_L}\}$, where a vertex position $\mathbf{p_i}$ specifies the coordinates $\{p_{ix}, p_{iy}, p_{iz}\}$ in R^3. The mesh topology, i.e., the connectivity between vertices, specifies all the vertices in each polygon, such as the IndexedFaceSet in VRML format [25].

In the literature, a variety of watermarking algorithms (e.g., [18], [19], [26]–[36]) have been proposed to embed the watermarks into polygonal meshes for integrity verification or copyright protection. As the vertex coordinates in a polygonal mesh are represented by numbers with decimal point, the incorporation of the SQS and the reversible algorithm in Section III.C is applicable as well. Different from the traditional authentication algorithms, the robustness to the nonmalicious manipulations of translation and rotation can be achieved by choosing the appropriate host values for data embedding.

A. Watermark Embedding Process

Given a string of bit values, the task of watermark embedding is to embed each bit value in the distance from a vertex to the centroid of its traversed neighbors. A secret key K is used as the seed of pseudo-random number generator to scramble the vertex indices I and randomly choose a polygon in a polygonal mesh, respectively. The process of mesh traversal is ordered by the scrambled vertex indices I' as proposed in [30]: Among those vertices in the randomly chosen polygon, the one first indexed by I' is traversed at first. Among the vertices connected with the traversed one, the one that is first indexed by I' is subsequently traversed. Then, we always traverse the vertex that is first indexed by I' among the neighbors of the traversed ones (i.e., those vertices directly connected with the traversed ones) until all of them are traversed. For a traversed vertex with the position $\mathbf{p_i}$, let us use F_i to denote the number of its traversed neighbors. The set of its traversed neighbors can be represented by $(\mathbf{n_i^j})_{j=1}^{F_i}$, where $\mathbf{n_i^j}$ denotes the position of the jth traversed neighbor of $\mathbf{p_i}$. By using $\mathbf{n_c^i}$ to denote the centroid of the traversed neighbors, it can be calculated by

$$\mathbf{n_c^i} = \frac{1}{F_i} \sum_{j=1}^{F_i} \mathbf{n_i^j}. \tag{15}$$

Then, the Euclidean distance X_i from $\mathbf{n_c^i}$ to $\mathbf{p_i}$ is chosen to be modulated, which is defined by

$$X_i = \sqrt{(n_{cx}^i - p_{ix})^2 + (n_{cy}^i - p_{iy})^2 + (n_{cz}^i - p_{iz})^2} \tag{16}$$

where $\{n_{cx}^i, n_{cy}^i, n_{cz}^i\}$ and $\{p_{ix}, p_{iy}, p_{iz}\}$ are the coordinates of $\mathbf{n_c^i}$ and $\mathbf{p_i}$ in R^3, respectively. Given L vertices in a polygonal mesh, there are $L-1$ defined distances because only the first traversed vertex has no traversed neighbor. For consistency, we use N instead of $L-1$ to denote the number of the defined distances. Furthermore, for $i = 1, 2, \ldots, N$, we denote the position of the ith traversed vertex that has the traversed neighbors by $\mathbf{p_i}$, and the distance from it to the centroid of its traversed neighbors by X_i.

In [30], each of the distances $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ is modulated separately to embed a watermark. To utilize the incorporation of the SQS and the reversible algorithm, a new signal $\mathbf{V} = \{V_1, V_2, \ldots, V_N\}$ needs to be generated

from $\mathbf{X}$ as shown in (2) with the quantization step-size Δ. For $i = 1, 2, \ldots, N$, a bit value m_i is embedded in V_i with a parameter a to generate an element V_i' in the watermarked signal $\mathbf{V}'$, which contains the embedded data and the reserved modulation information. Then, the original distance X_i is modulated to X_i' by $X_i' = V_i' - (V_i - X_i)$. Subsequently, the modulated distance X_i' is used to adjust the vertex position $\mathbf{p_i}$ to a new one $\mathbf{g_i}$ by

$$\mathbf{g_i} = \mathbf{n_c^i} + (\mathbf{p_i} - \mathbf{n_c^i}) \times \frac{X_i'}{X_i}. \tag{17}$$

Given $X_i > \Delta/2$, the modulated distance X_i' will be positive so that the distance from $\mathbf{g_i}$ to $\mathbf{n_c^i}$ becomes X_i' after the adjustment. The watermarked mesh is generated after a string of bit values $\mathbf{M} = \{m_1, m_2, \ldots, m_N\}$ has been embedded. For the exact recovery of the original mesh, every coordinate in the watermarked mesh should be stored with one more decimal digit, i.e., at the precision level of $10^{-(n+1)}$ if the original mesh is at the level of 10^{-n}.

B. The Authentication Process

In the authentication process, the quantization step-size Δ and the parameter a used in the embedding process are required. Since the process of mesh traversal is ordered by the scrambled vertex indices I', the secret key K is needed to generate them. Therefore, the secret key K, the quantization step-size Δ, and the parameter a, which are independent from the original mesh, are used as the inputs of this process.

The watermark retrieval is performed as follows. Initially, the vertex indices I in the watermarked mesh are scrambled to generate the scrambled ones I' while a polygon is randomly chosen, both with the secret key K. Among the vertices in the randomly chosen polygon, the one first indexed by I' is traversed at first. During the mesh traversal ordered by I' as shown in the embedding process, the distance from a vertex to the centroid of its traversed neighbors can be calculated in the same way as that in (15) and (16). If the watermarked mesh is intact, the modulated distances $\mathbf{X}' = \{X_1', X_2', \ldots, X_N'\}$ can be obtained. Then, the watermarked signal $\mathbf{V}' = \{V_1', V_2', \ldots, V_N'\}$ can be generated from $\mathbf{X}'$ as shown in (3). With the quantization step-size Δ, a bit value m_i' can be extracted from an element V_i' in $\mathbf{V}'$ by

$$m_i' = B\left(\left\lfloor \frac{V_i'}{\Delta} \right\rfloor\right) \tag{18}$$

so that a string of bit values $\mathbf{M}' = \{m_1', m_2', \ldots, m_N'\}$ can be extracted. To detect the change of the watermarked mesh and estimate its strength, the extracted string $\mathbf{M}'$ is compared with the original one $\mathbf{M}$ by defining a numerical value H

$$H = \frac{1}{N}\sum_{i=1}^{N} C(m_i', m_i) \tag{19}$$

where $C(m_i', m_i)$ is equal to 1 if $m_i' = m_i$, and equal to 0 otherwise. Obviously, H will be less than 1 if the extracted values $\mathbf{M}'$ do not match with the original ones $\mathbf{M}$ exactly.

To recover the original mesh, the distances $\mathbf{X}$ defined in it need to be obtained. From the watermarked signal $\mathbf{V}'$, we can obtain the original one, i.e., $\mathbf{V}$, with the quantization step-size Δ and the parameter a as described in Section III-B. Therefore, the distances $\mathbf{X}$ can be obtained by $X_i = V_i - (V_i' - X_i')$ for $i = 1, 2, \ldots, N$. For each vertex that has been adjusted in the embedding process, its original position $\mathbf{p_i}$ can be restored by

$$\mathbf{p_i} = \mathbf{n_c^i} + (\mathbf{g_i} - \mathbf{n_c^i}) \times \frac{X_i}{X_i'} \tag{20}$$

where $\mathbf{g_i}$ is its position in the watermarked mesh and $\mathbf{n_c^i}$ is the centroid of its traversed neighbors.

When the vertex coordinates of the watermarked mesh are stored in the precision level of $10^{-(n+1)}$, the round-off error of each coordinate of $\mathbf{g_i}$ is within $(-5 \times 10^{-(n+2)}, 5 \times 10^{-(n+2)}]$. It can be seen from (15) that the error introduced to each coordinate of $\mathbf{n_c^i}$ is also within $(-5 \times 10^{-(n+2)}, 5 \times 10^{-(n+2)}]$. From (16), it can be seen that the total error introduced to the modulated distance X_i' is within $(-\sqrt{3} \times 10^{-(n+1)}, \sqrt{3} \times 10^{-(n+1)})$. To ensure that the embedded bit value m_i can be correctly extracted, it should be satisfied that

$$\sqrt{3} \times 10^{-(n+1)} < \frac{\Delta}{2} \pm \frac{\Delta}{a} < \Delta - \sqrt{3} \times 10^{-(n+1)} \tag{21}$$

i.e., $\Delta > 2\sqrt{3}a/a - 2 \times 10^{-(n+1)}$ when $a > 2$. Given (21), the error introduced to V_i in (14) is within $(-\sqrt{3}a \times 10^{-(n+1)}, \sqrt{3}a \times 10^{-(n+1)})$. Since $X_i = V_i - (V_i' - X_i')$, it can be seen from (3) that the difference between V_i and X_i is a multiple of Δ. Therefore, the error introduced to the recovered distance X_i is also within $(-\sqrt{3}a \times 10^{-(n+1)}, \sqrt{3}a \times 10^{-(n+1)})$. Since X_i' is calculated from $|\mathbf{g_i} - \mathbf{n_c^i}|$, the value of $\mathbf{g_i} - \mathbf{n_c^i}/X_i'$ in (20) is a unit vector in R^3. Therefore, the error introduced to each coordinate of the position vector $\mathbf{p_i}$ in (20) is within $(-(\sqrt{3}a + 0.5) \times 10^{-(n+1)}, (\sqrt{3}a + 0.5) \times 10^{-(n+1)})$. By setting the parameter a within $(2, 3\sqrt{3}/2)$, all the coordinates can be recovered at the precision level of 10^{-n}.

As a countermeasure to the modifications that may change the reserved modulation information, the following condition should be satisfied:

$$\frac{\Delta}{2} - \frac{\Delta}{a} < \frac{5 \times 10^{-(n+1)} - 5 \times 10^{-(n+2)}}{a} \tag{22}$$

i.e., $\Delta < 9 \times 10^{-(n+1)}/(a-2)$. Combined with (21), the quantization step size Δ should be chosen within the interval of $(2\sqrt{3}a \times 10^{-(n+1)}/(a-2), 9 \times 10^{-(n+1)}/(a-2))$ to make the embedded watermark resistant to the round-off error, whereas sensitive to the modifications changing the reserved modulation information. By choosing Δ and the parameter a as discussed, the recovered mesh content is regarded as the original one if the obtained value of H in (19) is 1.

V. Performance Analysis

The generated watermarking algorithm was performed on the mesh models listed in Table I by assigning $N - 1$ to the parameter D in (2). The watermark could be a string of randomly generated bit values, or the hash value generated from the attributes of a mesh model except for its geometry. Since the coordinates in the original mesh were at the precision level of 10^{-4},

TABLE I
POLYGONAL MESH MODELS USED IN THE EXPERIMENTS

3D mesh models	Number of patches	Number of vertices	Number of polygons	Capacity (bits)
fish	1	742	1408	741
teapot	5	1631	3080	1626
dog	48	7616	13176	7568
horse	31	10316	18359	10285
lion	313	20315	32094	20002

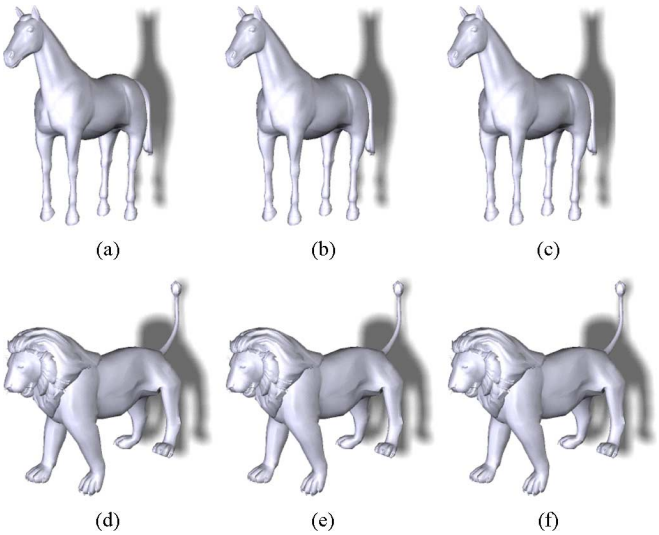


Fig. 3. By using the generated watermarking algorithm with the quantization step-size $\Delta = 0.0008$ and the parameter $a = 2.1$, 10285 and 20002 bits are embedded within the mesh models "horse" and "lion" to generate the watermarked ones, from which the original mesh models can be recovered, respectively. (a) Original mesh model of "horse." (b) Watermarked mesh model of "horse" with 10285 bits embedded. (c) Recovered mesh model of "horse." (d) Original mesh model of "lion." (e) Watermarked mesh model of "lion" with 20002 bits embedded. (f) Recovered mesh model of "lion."

the coordinates in the watermarked mesh were set at the precision level of 10^{-5} to guarantee the exact recovery. The parameter a should be set within the interval of $(2, 3\sqrt{3}/2)$, e.g., 2.1. Given $a = 2.1$, the quantization step-size Δ should be greater than $2\sqrt{3} \times 10^{-5}/(a-2) \simeq 7.27 \times 10^{-4}$ meanwhile less than $9 \times 10^{-5}/(a-2) = 9 \times 10^{-4}$. By setting the parameter a and the quantization step-size Δ at 2.1 and 0.0008 in the experiments, the pictures rendered from the mesh models "horse" and "lion" before and after the watermarking process are shown in Fig. 3.

A. Imperceptibility & Reversibility

To represent the distortion of the mesh content, the 3-D signal-to-noise ratio (3-D SNR) is defined as follows: Given L vertices in a polygonal mesh, the vertex positions can be represented by $\boldsymbol{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_L}\}$ while the vertex positions in the watermarked mesh are denoted as $\boldsymbol{G} = \{\mathbf{g_1}, \ldots, \mathbf{g_L}\}$. By using the mean square function $\mathrm{MS}(\mathbf{x})$, the 3-D SNR of the watermarked mesh is calculated by

$$\mathrm{SNR} = 10 \log_{10} \frac{\mathrm{MS}(\boldsymbol{P} - \overline{\boldsymbol{P}})}{\mathrm{MS}(\boldsymbol{G} - \boldsymbol{P})} \tag{23}$$

where $\overline{\boldsymbol{P}} = \{\overline{p}_x, \overline{p}_y, \overline{p}_z\}$ is the mean of P, i.e., the centroid of vertex positions. In the definition of (23), $\boldsymbol{P} - \overline{\boldsymbol{P}}$ is used to represent the signal, whereas $\boldsymbol{G} - \boldsymbol{P} = \boldsymbol{G} - \overline{\boldsymbol{P}} - (\boldsymbol{P} - \overline{\boldsymbol{P}})$ is the noise. In particular

$$\begin{cases} \mathrm{MS}(\boldsymbol{P} - \overline{\boldsymbol{P}}) = \dfrac{\sum_{i=1}^{L} (p_{ix} - \overline{p}_x)^2 + (p_{iy} - \overline{p}_y)^2 + (p_{iz} - \overline{p}_z)^2}{L} \\ \mathrm{MS}(\boldsymbol{G} - \boldsymbol{P}) = \dfrac{\sum_{i=1}^{L} (g_{ix} - p_{ix})^2 + (g_{iy} - p_{iy})^2 + (g_{iz} - p_{iz})^2}{L}. \end{cases} \tag{24}$$

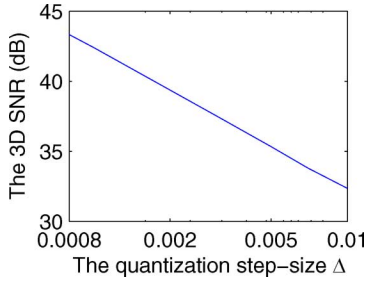


Fig. 4. By setting the parameter a at 2.1, the 3-D SNR of the watermarked mesh "teapot" varies with the quantization step-size Δ.

Similar to the former algorithm in [30], the impact of watermark embedding could be tuned by the quantization step-size Δ used in the experiments. If we assigned 0.01 to Δ, the calculated 3-D SNR of the watermarked mesh "teapot" was about 32.37 dB, whereas the SNR was 42.41 dB if 0.001 had been assigned to Δ instead. As shown in Fig. 4, the obtained 3-D SNR value of the watermarked mesh was decreased over the increase of the quantization step-size Δ. It can be seen that the SQS does not affect the imperceptibility of the quantization-based embedding. The capability of recovering the original mesh was enabled by storing every coordinate in the watermarked mesh with one more decimal digit. In the experiments, the file size of the watermarked mesh was not increased after improving the precision from 10^{-4} to 10^{-5}, when the coordinates were in floating point numbers. By setting the parameters a and Δ at 2.1 and 0.0008, the SNR values of the recovered meshes calculated in (23) were all *infinite*. Compared with the existing watermarking algorithms such as in [19] and [28], the advantage of the generated one for authentication applications is that the property of reversibility is achievable.

B. Data Hiding Capacity

Given L vertices in a mesh, the data hiding capacity using the generated watermarking algorithm is $L - 1$ bits, which is identical to that of the former algorithm in [30]. As shown in Table I, the capacity will be $L - W$ bits if there are L vertices and W separate patches in a polygonal mesh, for the first traversed vertex in each patch cannot be used to embed one bit value. It can be seen that the proposed SQS does not affect the data hiding capacity of quantization-based embedding.

C. Integrity Verification

To use the generated watermarking algorithm for mesh authentication, we test the sensitivity of the embedded watermark to illegal changes of the watermarked mesh, which was generated by setting Δ and a at 0.0008 and 2.1, respectively. Then, the watermarked meshes went through several geometrical modifications, such as modifying one vertex position by adding the

TABLE II
IN THE EXPERIMENTS, THE WATERMARKED MESH IS GENERATED BY ASSIGNING 0.0008 AND 2.1 TO THE QUANTIZATION STEP-SIZE Δ AND THE PARAMETER a, RESPECTIVELY. THE VALUES OF H ARE CALCULATED BY COMPARING THE EXTRACTED WATERMARKS WITH THE ORIGINAL ONE AFTER THE WATERMARKED MESH IS PROCESSED BY THE FOLLOWING MANIPULATIONS, RESPECTIVELY

Polygonal meshes models	Translation and rotation	Uniformly scaling	Modifying one vertex position	Moving two vertices oppositely	Adding Gaussian noise	Reducing one face	Extraction without the secret key
fish	1.0000	0.5236	0.9851	0.9716	0.8096	0.6869	0.4885
teapot	1.0000	0.4846	0.9956	0.9926	0.8296	0.5854	0.5295
dog	1.0000	0.4943	0.9990	0.9982	0.8196	0.4596	0.4922
horse	1.0000	0.5017	0.9996	0.9994	0.8137	0.6280	0.5051
lion	1.0000	0.5009	0.9997	0.9996	0.8213	0.8371	0.5077

vector $\{3\Delta, 3\Delta, 3\Delta\}$, and moving two vertices oppositely by adding the vector $\{2\Delta, 2\Delta, 2\Delta\}$ and $\{-2\Delta, -2\Delta, -2\Delta\}$. The global modifications were conducted by applying random permutations to every coordinate, according to the Gaussian noise with zero mean and a variance of $\Delta^2/16$. Translation, uniformly scaling and rotation were also applied to the watermarked geometry, respectively. Further, we made topological modifications by reducing one face from the watermarked mesh and performing the mesh traversal process without the secret key K.

After extracting a string of bit values from the modified geometry and comparing them with the original ones using (19), the obtained values of H are listed in Table II. From the experimental results, it can be seen that the content-preserving manipulations such as translation and rotation could be allowed, while other modifications were detected. However, the position and orientation of the watermarked mesh can also be authenticated by modulating the vertex coordinates directly, if necessary. By comparing the obtained H values with the NC values as shown in [30], it can be seen that illegal modifications can be detected more efficiently in the generated watermarking algorithm. Hence, the sensitivity of the embedded watermark to illegal modifications has been increased by employing the SQS. To achieve the capability of tamper localization, a polygonal mesh needs to be partitioned into several patches so that the generated watermarking algorithm can be performed on each of them.

D. Security

The security of the generated watermarking algorithm relies on the three facts: (1). The secret key K is used to scramble the vertex indices I and randomly choosing a polygon. Without knowing the secret key K, the mesh traversal must be exhasutively performed to ensure that the embedded watermark can be correctly extracted with the correct quantization step-size Δ. (2). By assigning a value slightly greater than 2 (e.g., 2.1) to the parameter a, the remainders left after dividing the modulated distances by Δ are distributed within $(0, \Delta)$. As a result, it is hard to accurately estimate the value of Δ from the watermarked mesh, especially when the technique of dither modulation [5] is also employed. (3). Compared with the former algorithm [30], what we have improved is the sensitivity of the embedded watermark to illegal modifications, making it even harder to construct a counterfeit mesh containing the same watermark without knowing the secret key K and the quantization step-size Δ.

VI. CONCLUSION

In this paper, a SQS has been proposed to modulate a set of host values chosen from a host signal for data embedding. The chance to detect illegal modifications has been increased by making the modulation of a host value dependent on a certain number of the previous ones. Furthermore, a balance between security improvement and tamper localization has been achieved for integrity verification. We have applied the incorporation of the SQS and a reversible data hiding mechanism to polygonal meshes by choosing the host values as proposed in [30]. The experimental results have demonstrated the efficacy of the generated watermarking algorithm for mesh authentication.

Therefore, the proposed SQS provides an efficient way to improve the security of quantization-based embedding for integrity verification meanwhile preserving the capability of tamper localization. The incorporation with the SQS does not affect the properties of the specific embedding algorithm (e.g., the property of reversibility), thus more suitable for the authentication applications.

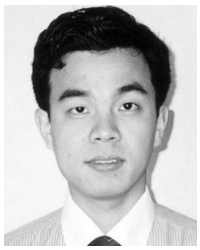
ACKNOWLEDGMENT

The authors would like to sincerely thank the Associate Editor and three anonymous reviewers for their insightful suggestions and valuable comments. The mesh models used in the experiments are from http://www.cs.unc.edu/~isenburg/asciicoder/ and http://pascal.leynaud.free.fr/3ds/.

REFERENCES

[1] A. Menezes, P. van Oorchot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.
[2] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Norwood, MA: Artech House, 2000.
[3] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. New York: Morgan Kaufmann, 2001.
[4] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, Jul. 1999.
[5] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.
[6] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1593–1601, Oct. 2001.
[7] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
[8] S. Walton, "Information authentication for a slippery new age," *Dr. Dobbs J.*, vol. 20, no. 4, pp. 18–26, 1995.

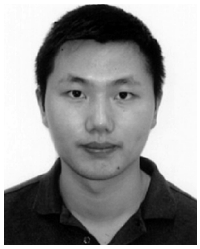
[9] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. IEEE ICIP*, 1994, vol. 2, pp. 86–90.
[10] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 585–595, Jun. 2002.
[11] M. Yeung and F. Mintzer, "Invisible watermarking for image verification," *J. Electron. Imaging*, vol. 7, no. 3, pp. 578–591, 1998.
[12] M. Wu and B. Liu, "Watermarking for image authentication," in *Proc. IEEE ICIP*, 1998, vol. 2, pp. 437–441.
[13] C.-Y. Lin and S.-F. Chang, "Semi-fragile watermarking for authenticating JPEG visual content," in *Proc. SPIE: Security Watermarking Multimedia Contents II*, 2000, vol. 3971, pp. 140–151.
[14] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proc. IEEE*, vol. 87, no. 7, pp. 1167–1180, Jul. 1999.
[15] L. Xie and G. Arce, "A class of authentication digital watermarks for secure multimedia communication," *IEEE Trans. Image Process.*, vol. 10, no. 11, pp. 1754–1764, Nov. 2001.
[16] E. Lin and E. Delp, "A review of fragile image watermarks," in *Proc. ACM Multimedia Security Workshop*, 1999, pp. 25–29.
[17] C. Rey and J. L. Dugelay, "A survey of watermarking algorithms for image authentication," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 6, pp. 613–621, 2002.
[18] B. L. Yeo and M. M. Yeung, "Watermarking 3-D objects for verification," *IEEE Comput. Graph. Appl.*, vol. 19, no. 1, pp. 36–45, Jan. 1999.
[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 Trans. Multimedia*, vol. 7, no. 6, pp. 997–1006, Dec. 2005.
[20] N. Memon, S. Shende, and P. Wong, "On the security of the Yueng-Mintzer authentication watermark," in *Proc. Final Program IS&T PICS 99*, 1999, pp. 301–306.
[21] M. Holliman and N. Memon, "Counterfeiting attacks for block-wise independent watermarking techniques," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 432–441, Mar. 2000.
[22] J. Fridrich, M. Goljan, and N. Memon, "Further attacks on Yeung–Mintzer fragile watermarking scheme," in *Proc. SPIE: Security Watermarking Multimedia Contents II*, 2000, vol. 3971, pp. 428–437.
[23] M. Wu, "Joint security and robustness enhancement for quantization based embedding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 831–841, Aug. 2003.
[24] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.
[25] *The Virtual Reality Modeling Language*, ISO/IEC DIS 14772-1 [Online]. Available: http://www.web3d.org/x3d/specifications/vrml/
[26] O. Benedens and C. Busch, "Toward blind detection of robust watermarks in polygonal models," *Proc. EUROGRAPHICS*, vol. 19, no. 3, 2000.
[27] F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 939–949, Apr. 2003.
[28] F. Cayre, O. Devillers, F. Schmitt, and H. Maitre, "Watermarking 3-D triangle meshes for authentication and integrity," INRIA, 2004, Res. Rep. RR-5223.
[29] H. T. Wu and Y. M. Cheung, "A reversible data hiding approach to mesh authentication," in *Proc. 2005 IEEE/WIC/ACM Int. Conf. Web Intelligence*, Compiegne, France, 2005, pp. 774–777.
[30] H. T. Wu and Y. M. Cheung, "A high-capacity data hiding method for polygonal meshes," presented at the 8th Inf. Hiding Workshop, 2006.
[31] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 551–560, Apr. 1998.
[32] O. Benedens, "Geometry-based watermarking of 3-D models," *IEEE Comput. Graph. Appl.*, vol. 19, no. 1, pp. 46–55, Jan. 1999.
[33] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proc. ACM SIGGRAPH 99*, 1999, pp. 69–76.
[34] E. Garcia and J. L. Dugelay, "Texture-based watermarking of 3-D video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 853–866, Aug. 2003.
[35] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3-D mesh objects," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 5, pp. 596–607, Oct. 2005.
[36] A. G. Bors, "Watermarking mesh-based representations of 3-D objects using local moments," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 687–701, Mar. 2006.



Yiu-ming Cheung (SM'06) received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2000.

Currently, he is an Associate Professor at the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong. His research interests include machine learning, information security, signal processing, pattern recognition, and data mining.

Dr. Cheung is the Founding and present Chair of the Computational Intelligence Chapter of IEEE Hong Kong Section. He is a senior member of ACM.



Hao-tian Wu (S'05) received the B.E. and M.E. degrees from Harbin Institute of Technology, Harbin, China, in 2002 and 2004, respectively. He is currently working toward the Ph.D. degree at the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

His research interests include fragile watermarking, multimedia security, 3-D geometry watermarking, and steganography.

Mr. Wu has been serving as secretary of the Computational Intelligence Chapter of IEEE Hong Kong section since 2005.