An Oblivious Watermarking for 3-D Polygonal Meshes Using Distribution of Vertex Norms

Jae-Won Cho, Rémy Prost, and Ho-Youl Jung

***Abstract*—Although it has been known that oblivious (or blind) watermarking schemes are less robust than nonoblivious ones, they are more useful for various applications where a host signal is not available in the watermark detection procedure. From a viewpoint of oblivious watermarking for a three-dimensional (3-D) polygonal mesh model, distortionless attacks, such as similarity transforms and vertex reordering, might be more serious than distortion attacks including adding noise, smoothing, simplification, remeshing, clipping, and so on. Clearly, it is required to develop an oblivious watermarking that is robust against distortionless as well as distortion attacks. In this paper, we propose two oblivious watermarking methods for 3-D polygonal mesh models, which modify the distribution of vertex norms according to the watermark bit to be embedded. One method is to shift the mean value of the distribution and another is to change its variance. Histogram mapping functions are introduced to modify the distribution. These mapping functions are devised to reduce the visibility of watermark as much as possible. Since the statistical features of vertex norms are invariant to the distortionless attacks, the proposed methods are robust against such attacks. In addition, our methods employ an oblivious watermark detection scheme, which can extract the watermark without referring to the cover mesh model. Through simulations, we demonstrate that the proposed approaches are remarkably robust against distortionless attacks. In addition, they are fairly robust against various distortion attacks.**

***Index Terms*—Distortion attack, distortionless attack, distribution of vertex norms, three-dimensional (3-D) polygonal meshes, watermarking.**

I. Introduction

With the remarkable growth of network technology such as the World Wide Web, digital media enables us to copy, modify, store, and distribute digital data without effort. As a result, it has become a new issue to research schemes for copyright protection. Traditional data protection techniques such as encryption are not adequate for copyright enforcement because the protection cannot be ensured after the data are decrypted. Watermarking provides a mechanism for copyright protection by embedding information, called a watermark, into host data [1]. Unlike encryption, digital watermarking does not restrict access to the host data but ensures the hidden data remain inviolate and recoverable. Note that so-called fragile or semifragile watermarking techniques have also been widely used for content authentication and tamper proofing [2]. Here, we address only watermarking technique for copyright protection, namely, robust watermarking.

Most previous research has focused on general types of multimedia data including text data, audio stream [3]–[5], still images [6]–[8], and video stream [9]. Recently, with the interest and requirement of three-dimensional (3-D) models such as Virtual Reality Modeling Language, computer-aided design (CAD), polygonal mesh models, and medical objects, several watermarking techniques for 3-D mesh models have been developed [1], [10]–[17], [25]–[27].

Since 3-D mesh watermarking techniques were introduced in [10], there have been several attempts to improve the performance in terms of transparency and robustness. Ohbuchi *et al.* [10] proposed three watermarking schemes: triangle similarity quadruple, tetrahedral volume ratio (TVR), and a visible mesh watermarking method. These schemes can be regarded as oblivious (or blind) techniques that can extract the watermark without reference to a cover mesh model, but they are not sufficiently robust against various attacks. For example, TVR is very vulnerable to remeshing, simplification, and reordering attacks. Beneden [11] proposed a watermark embedding method that modifies the local distribution of vertex directions from the center point of model. The method is robust against simplification attack because the local distribution is not sensitive to such operations. An extended scheme was also introduced in [12] to overcome a weakness to cropping attack. However, the method still requires preprocessing for reorientation during the process of watermark detection, as the local distribution essentially varies with the degree of rotation. Yu *et al.* [13], [14] proposed a vertex norm modification method that perturbs the distance between the vertices to the center of model according to the watermark bit to be embedded. It employs, before the modification, scrambling of vertices for the purpose of preserving the visual quality. Note that it is not an oblivious technique and also requires preprocessing such as registration and resampling. Some multiresolution based methods have also been introduced [15]–[17]. Kanai *et al.* [15] proposed a watermarking algorithm based on wavelet transform. Similar approaches, using Burt–Adelson style pyramid and mesh spectral

Manuscript received May 16, 2005; accepted March 1, 2006. This work was supported by the Korea Science and Engineering Foundation under KOSEF 000-B-105-898, by the Centre National de la Recherche Scientifique under CNRS, 14894, and by the Ministry of Information and Communications, Korea, under Information Technology Research Center Program 204-B-000-215. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. David J. Miller.

J.-W. Cho is with CREATIS, Centre National de la Recherche Scientifique, Inserm U630, INSA-UCB, Villeurbanne, France. He is also with the School of Electrical Engineering and Computer Science, Yeungnam University, Gyeongsan, Gyeongbuk, Korea (e-mail: ram56@yumail.ac.kr; cho@creatis.insa-lyon.fr).

R. Prost is with the CREATIS, Centre National de la Recherche Scientifique, Inserm U630, INSA-UCB, Villeurbanne, France (e-mail: remy.prost@creatis.insa-lyon.fr).

H.-Y. Jung is with the School of Electrical Engineering and Computer Science, Yeungnam University, Gyeongsan, Gyeongbuk, Korea (e-mail: hoyoul@yu.ac.kr).

Digital Object Identifier 10.1109/TSP.2006.882111

analysis, were also published in [16] and [17], respectively. The multiresolution techniques could achieve a high transparency of watermark but have not been used as an oblivious scheme since the connectivity information of vertices must be exactly known for multiresolution analysis in the watermark extraction process. Recently, there have been some trials that apply the spectral analysis based techniques directly to point-sampled geometry that is independent of vertex connectivity information [18], [19]. However, they are not oblivious scheme. Although it has been known that oblivious schemes are less robust than nonoblivious ones, they are more useful for various applications where a host signal is not available in the watermark detection procedure. For example, owner identification and copy control systems cannot refer to original data [20]–[22]. Furthermore, the use of nonoblivious watermarking can cause one to confuse the proof of ownership if an illegal user asserts that he is the copyright holder with a corrupt watermarked data as his original [23], [24]. In this paper, our interests focus on developing an oblivious watermarking.

Three-dimensional polygonal mesh models have serious difficulties for watermark embedding. While image data are represented by brightness (or amplitudes of RGB components in the case of color images) of pixels sampled over a regular grid in two dimension, 3-D polygonal models have no unique representation, i.e., no implicit order and connectivity of vertices [13], [14]. This creates synchronization problem during the watermark extraction, which makes it difficult to develop robust watermarking techniques. For this reason, most techniques developed for other types of multimedia are not effective for 3-D meshes. Furthermore, a variety of complex geometrical and topological operations could disturb the watermark extraction for assertion of ownership [14].

The geometrical attacks include adding noise, smoothing, and so on. Vertex reordering, simplification, and remeshing fall into the category of topological attacks. These attacks can be reclassified into two categories: distortion and distortionless attacks [27]. Distortion attacks include adding noise, simplification, smoothing, remeshing, clipping, and so on, which may cause visual deformation of the stego mesh model. Most conventional watermarking techniques of 3-D polygonal mesh models have been developed to be robust mainly against the distortion attacks [11], [12], [15]–[17], [26]. On the other hand, distortionless attacks include similarity transforms and vertex reordering. These attacks have been successfully overcome by some nonoblivious watermarking methods [1], [16], [26]. However, they might be more serious attacks to oblivious watermarking as they could fatally destroy the hidden watermark without any perceptual changes of stego mesh model. Clearly, it is required to develop an oblivious watermarking technique that is robust against distortionless as well as distortion attacks.

In this paper, we propose a statistical approach that modifies the distribution of vertex norms to hide watermark information into host 3-D meshes. In contrast with [11], we modify the distribution of vertex norms instead of normal distribution to hide watermark information. Similar to [11], we split the distribution into distinct sets called bins and embed one bit per bin. The distribution of vertex norms is modified by two methods. One is to shift the mean value of the distribution according to the watermark bit to be embedded and another to change its variance. A similar approach has been used to shift the mean value in our previous work [27], where a constant is added to vertex norms. Note that more sophisticated skills are introduced in this paper. In particular, histogram mapping functions are newly introduced and used for the purpose of elaborate modification. Since the statistical features are invariant to distortionless attacks and less sensitive to various distortion ones with local geometric alterations, robustness of watermark can be easily achieved. In addition, the proposed methods employ an oblivious watermark detection scheme.

The rest of this paper is organized as follows. In Section II, the main idea behind the statistical approach is introduced. In Sections III and IV, the proposed watermarking methods are described in detail, including their embedding and extracting procedures. Here, histogram mapping functions are also introduced to efficiently change the mean value and variance of the vector norm distribution. Section V shows the simulation results of the proposed against various distortion and distortionless attacks. Section VI concludes this paper.

II. Main Idea of the Proposed Watermarking Methods

In order to achieve robustness of watermark against distortionless attacks, it is very important to find a watermark carrier, also called primitive in [10], that can effectively preserve watermark from such attacks. For example, if vertices arranged in a certain order are used as the watermark carrier, the hidden watermark bit stream cannot be retrieved after vertex reordering. This is caused by the fact that 3-D polygonal meshes do not have implicit order and connectivity of vertices. For the same reason, preprocessing such as registration and resampling is required as in [13] and [14], or the robustness against distortionless attacks cannot be guaranteed [10], [15]–[17]. Clearly, statistical features can be promising watermark carriers as they are generally less sensitive to these kinds of attacks. Several features can be obtained directly from 3-D meshes, particularly, the distribution of vertex directions and distribution of vector norms. Distribution of vertex directions has been used as a watermark carrier in [11] and [12], where vertices are grouped into distinct sets according to their local direction and the distribution of vertex direction is altered in each set separately. The distribution does not change by vertex reordering operation, but it varies in essence with rotation operation. Thus, it requires reorientation processing before watermark detection [11], [12]. On the other hand, the distribution of vertex norms does not change either by vertex reordering or rotation operations. This is the reason why the distribution of vertex norms is used as a watermark carrier in this paper.

We propose two watermarking methods that embed a watermark into the 3-D mesh model by modifying the distribution of vertex norms. Figs. 1 and 2 show the main idea of each method, respectively. The first method is to make the mean value of vertex norms greater or smaller than a reference value according to a watermark bit that we want to insert. Assume that the vertex norms of cover meshes are mapped into the interval [0,1] and have a uniform distribution over the interval as shown in Fig. 1(a). In this figure, an arrow indicates the mean value of the vertex norms. To embed a watermark bit of $+1$, the distribution is modified so that its mean value is greater than a reference value as shown

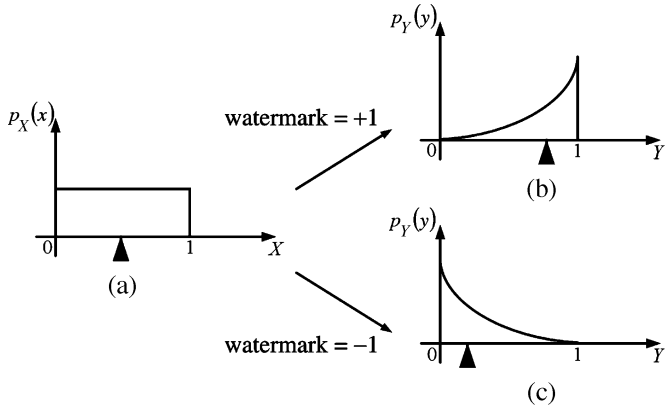


Fig. 1. Proposed watermarking method by shifting the mean of the distribution.

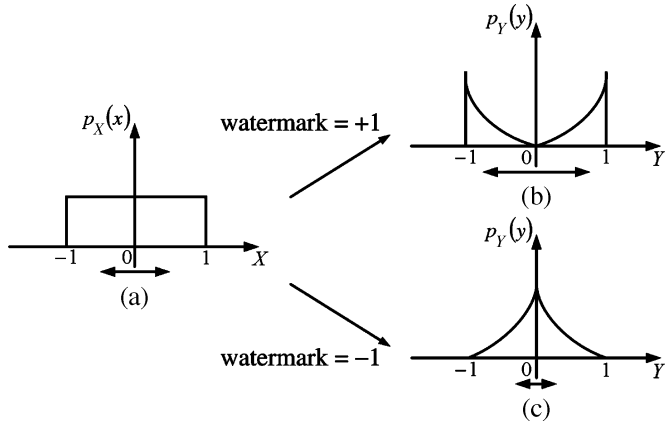


Fig. 2. Proposed watermarking method by changing the variance of the distribution.

in Fig. 1(b). To embed -1, the distribution is modified so that it is concentrated on the left side, and the mean value becomes smaller than a reference as shown in Fig. 1(c). The watermark extraction process is quite simple if the reference value is known. The hidden watermark bit can be easily retrieved by simple comparison of the reference with the mean value of vertex norms obtained from stego meshes. The second proposed method is to change the variance of vertex norms to be greater or smaller than a reference. Assume that the vertex norms are mapped into the interval [−1,1] and have a uniform distribution over the interval as shown in Fig. 2(a), where its standard deviation is indicated by a bidirectional arrow. To embed a watermark bit of $+1$, the distribution is modified to concentrate on both margins. This leads to increase the standard deviation as shown in Fig. 2(b). To embed -1, the distribution is altered to concentrate on the center so that the standard deviation becomes smaller than a reference deviation, as shown in Fig. 2(c). Similar to the first proposed, the watermark can be extracted by comparing the reference variance and variance taken from stego meshes.

Starting from the main idea of modifying the distribution of vertex norms, we introduce some techniques to enhance watermark capacity and transparency. The distribution is divided into distinct sections, hereafter referred to as bins, each of which is used as a watermark embedding unit to embed one bit of watermark. The number of watermark bits to be embedded can be

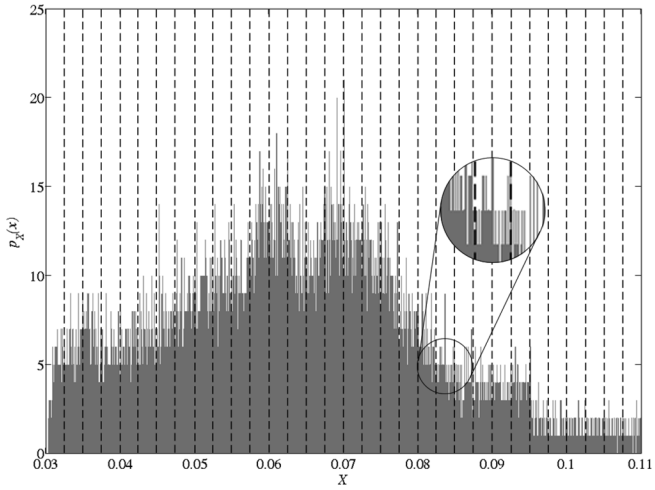


Fig. 3. Distribution of vertex norms obtained from the bunny model, where dashed vertical lines indicate the border of each bin.

properly selected by taking account the transparency. As an example, Fig. 3 shows the distribution of a bunny model, which is divided into bins by dashed vertical lines. It also shows that the distribution of each bin is close to uniform. In addition, we introduce histogram mapping functions that can effectively modify the distribution. The mapping functions are devised to reduce the visibility of the watermark as much as possible.

III. Proposed Watermarking Method I

This method embeds watermark information into 3-D polygonal mesh model by shifting the mean value of each bin according to assigned watermark bit. All of the vertex norms in each bin are modified by a histogram mapping function. Fig. 4 depicts the watermark embedding and extraction processes, which are described in detail in the following sections.

A. Watermark Embedding

First, Cartesian coordinates of a vertex $v_i = (x_i, y_i, z_i)$ on the cover mesh model $\mathbf{V}(v_i \in \mathbf{V})$ are converted into spherical coordinates $(\rho_i, \theta_i, \phi_i)$ by means of

$$\begin{aligned} \rho_i &= \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2} \\ \theta_i &= \tan^{-1} \frac{(y_i - y_g)}{(x_i - x_g)} \qquad \text{for } 0 \le i \le L-1 \\ \phi_i &= \cos^{-1} \frac{(z_i - z_g)}{\sqrt{(x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2}} \end{aligned} \quad (1)$$

where L is the number of the vertex, (x_g, y_g, z_g) is the center of gravity of the mesh model, and ρ_i is the ith vertex norm. The vertex norm represents the distance between each vertex and the center of gravity. The proposed method uses only vertex norms for watermarking and keeps the other two components θ_i and ϕ_i intact. Note that the distribution of vertex norms is invariant to vertex reordering and similarity transforms.

Secondly, vertex norms are divided into N distinct bins with equal range, according to their magnitude. Each bin is used independently to hide one bit of watermark. If every bin is processed for watermark embedding, we can insert at maximum N bits of

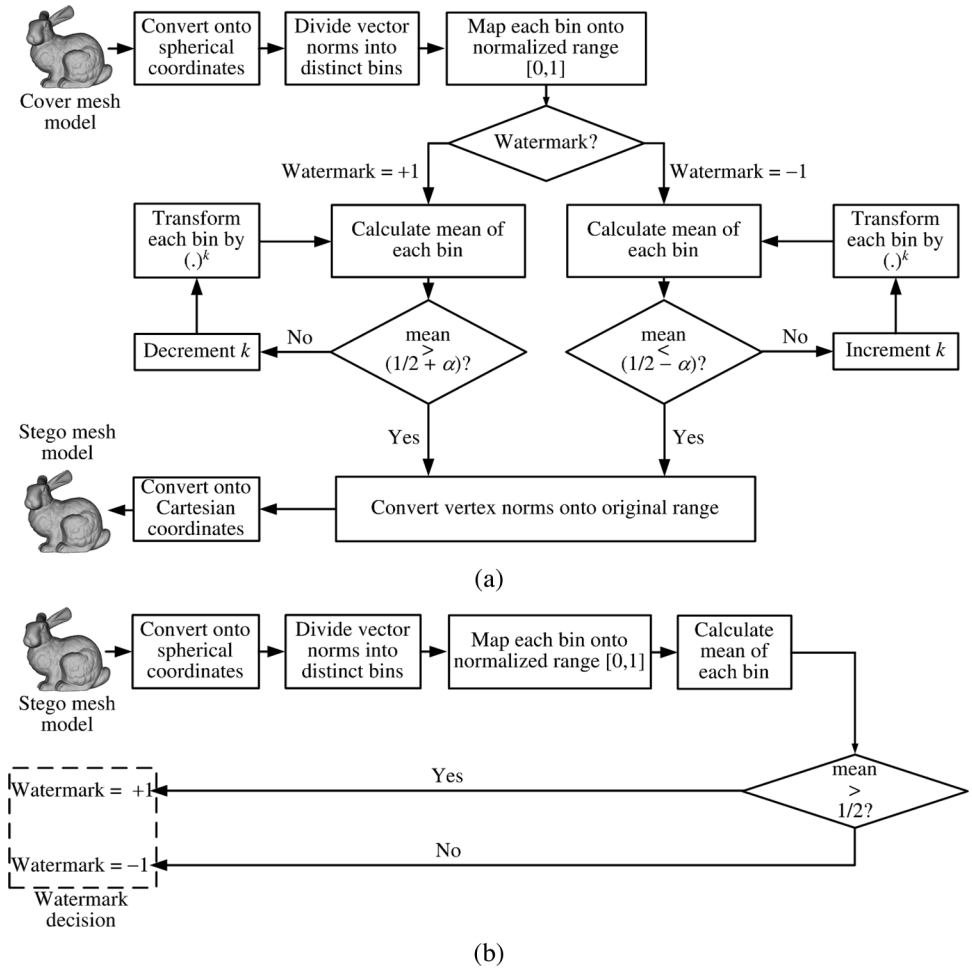


Fig. 4. Block diagrams of the (a) watermark embedding and (b) extraction for the proposed watermarking method shifting the mean value of vertex norms.

watermark. To classify the vertex norms into N bins, maximum and minimum vertex norms $\rho_{\max}$ and $\rho_{\min}$ are calculated in advance. The nth bin $\mathbf{B}_n$ is defined as follows:

$$\mathbf{B}_n = \left\{ \rho_{n,j} | \rho_{min} + \frac{\rho_{\max} - \rho_{\min}}{N} \cdot n < \rho_i < \rho_{\min} + \frac{\rho_{\max} - \rho_{\min}}{N} \cdot (n+1) \right\}$$
$$\text{for } 0 \le n \le N-1, 0 \le i \le L-1$$
$$0 \le j \le M_n - 1 \tag{2}$$

where M_n is the number of vertex norms belonging to the nth bin and $\rho_{n,j}$ is the jth vertex norm of the nth bin.

Thirdly, vertex norms belonging to the nth bin are mapped into the normalized range of [0,1] by

$$\widetilde{\rho}_{n,j} = \frac{\rho_{n,j} - \min_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\}}{\max_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\} - \min_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\}} \tag{3}$$

where $\widetilde{\rho}_{n,j}$ is the normalized, jth vertex norm of the nth bin. $\max_{\rho_{n,j}} \in \mathbf{B_n}\{\rho_{n,j}\}$ is the maximum vertex norm of the nth bin, and $\min_{\rho_{n,j}} \in \mathbf{B_n}\{\rho_{n,j}\}$ is the minimum vertex norm. Note that each bin now has a distribution very close to uniform over the unit interval, as mentioned in the previous section.

Before moving to the next step of watermark embedding, we consider a continuous random variable X with uniform distribution over the interval [0,1]. Clearly, the expectation of the random variable $E[X]$ is given by

$$E[X] = \int_0^1 x p_X(x) dx = \frac{1}{2} \tag{4}$$

where $p_X(x)$ is the probability density function of X. This expectation will be used as a reference value when moving the mean of each bin to a certain level in the next step. In our method, vertex norms in each bin are modified to shift the mean value. It is very important to assure that the modified vertex norms also exist within the range of each bin. Otherwise, vertex norms belonging to a certain bin could shift into neighbor bins, which may have a serious impact on the watermark extraction. We now propose a histogram mapping function, which can shift the mean to the desired level through modifying the value of vertex norms while staying within the proper range. The use of a mapping function is inspired from the histogram equalization techniques often used in image enhancement processing [29]. For a given continuous random variable X, the mapping function is defined as

$$Y = X^k \text{ for } 0 < k < \infty \text{ and } k \in \Re \tag{5}$$

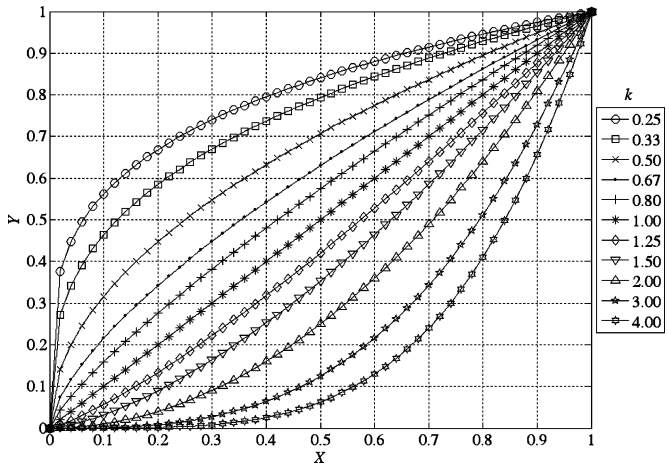


Fig. 5. Histogram mapping function, $Y = X^k$, for different parameters of k.

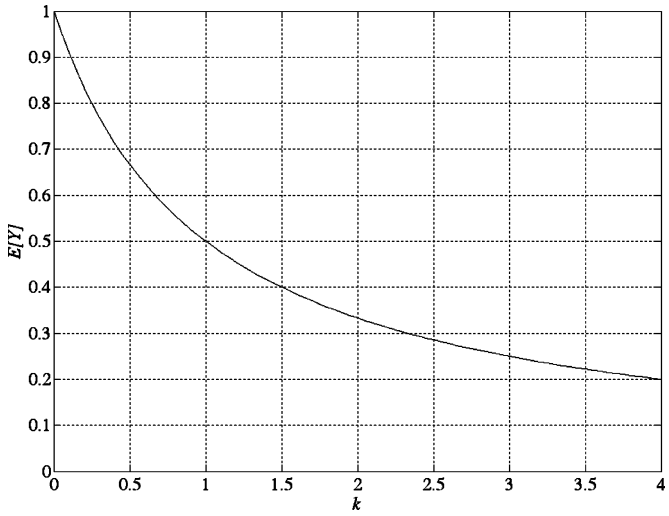


Fig. 6. Expectation of the output random variable via histogram mapping function with different k, assuming that the input random variable is uniformly distributed over unit range [0,1].

where Y is the transformed variable and the parameter k is a real value for $0 < k < \infty$. Fig. 5 shows curves of the mapping function for different values of k. When the parameter k is selected in the range $1 < k < \infty$, input variables are mapped into relatively small values. Moreover, increases in k decrease the value of the transformed variable. It means the reduction of mean value. On the other hand, the mean value increases for decreasing k when $0 < k < 1$. Expectation of output random variable $E[Y]$ is represented as

$$E[Y] = E[X^k] = \int_0^1 x^k p_X(x)dx = \frac{1}{k+1}. \tag{6}$$

Fig. 6 shows the expectation value of the output of the mapping function over k. The expectation value decreases monotonically with the parameter k. Therefore, we can easily adjust the mean value of the distribution by selecting a proper parameter. In particular, the mapping function not only guarantees to alter the variable within the limited range but also allows shifting of the mean value to the desired level.

The fourth step of the proposed watermark embedding is to shift the mean value of each bin via transforming vertex norms by the histogram mapping function. To embed a watermark bit of $+1(\omega_n = +1)$, vertex norms $\widetilde{\rho}_{n,j}$ are transformed in order to shift the mean of the distribution by a factor $\alpha(0 < \alpha < (1/2))$. Alternatively, to embed $\omega_n = -1$, vertex norms are transformed in order to shift the mean by a factor $-\alpha$. Then the mean of each bin $\widetilde{\mu}'_n$ is changed by

$$\widetilde{\mu}'_n = \begin{cases} \frac{1}{2} + \alpha \text{ if } \omega_n = +1 \\ \frac{1}{2} - \alpha \text{ if } \omega_n = -1 \end{cases} \tag{7}$$

where α is the strength factor that can control the robustness and the transparency of watermark. The exact parameter k_n can be found directly from (6)

$$k_n = \begin{cases} \frac{1-2\alpha}{1+2\alpha} \text{ if } \omega_n = +1 \\ \frac{1+2\alpha}{1-2\alpha} \text{ if } \omega_n = -1 \end{cases}. \tag{8}$$

Note that k_n exists in the range of]0,1[when the watermark bit is $+1$ and in the range of]1,∞[when the watermark bit is -1.

The real vertex norm distribution in each bin is neither continuous nor uniform. Then the parameter k_n cannot be calculated by (8). To overcome this difficulty, we use an iterative approach as follows.

For embedding $\omega_n = +1$ into the nth bin:

1) initialize the parameter k_n as 1;
2) transform normalized vertex norms by $\widetilde{\rho}'_{n,j} = (\widetilde{\rho}_{n,j})^{k_n}$;
3) calculate mean of transformed vertex norms through $\widetilde{\mu}'_n = (1/M_n)\sum_{j=0}^{M_n-1} \widetilde{\rho}'_{n,j}$;
4) if $\widetilde{\mu}'_n < (1/2) + \alpha$, decrease $k_n(k_n = k_n - \Delta k)$ and go back to 2);
5) replace normalized vertex norms with transformed norms using $\widetilde{\rho}_{n,j} = \widetilde{\rho}'_{n,j}$;
6) end.

For embedding $\omega_n = -1$ into the nth bin:

4) if $\widetilde{\mu}'_n > (1/2) - \alpha$, increase $k_n(k_n = k_n + \Delta k)$ and go back to 2).

Here, Δk indicates the step size for decrement or increment of parameter k_n, which is experimentally determined considering the tradeoff between the processing time and the precision error. The precision error means the distance between the desired mean ($(1/2)\pm\alpha$) and the actual modified one ($\widetilde{\mu}'_n$). Note that only step 4) is different according to the watermark bit to be embedded.

The fifth step is inverse processing of the third step. Transformed vertex norms of each bin are mapped onto the original range by

$$\rho'_{n,j} = \widetilde{\rho}'_{n,j} \cdot \left(\max_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\} - \min_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\} \right) + \min_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\} \tag{9}$$

where $\max_{\rho_{n,j} \in \mathbf{B_n}} \{\rho_{n,j}\}$ and $\min_{\rho_{n,j} \in \mathbf{B_n}} \{\rho_{n,j}\}$ are the same as those used in the third step.

Finally, the watermark embedding process is completed by combining all of the bins and converting the spherical coordinates to Cartesian coordinates. Let ρ'_i be a vertex norm in the

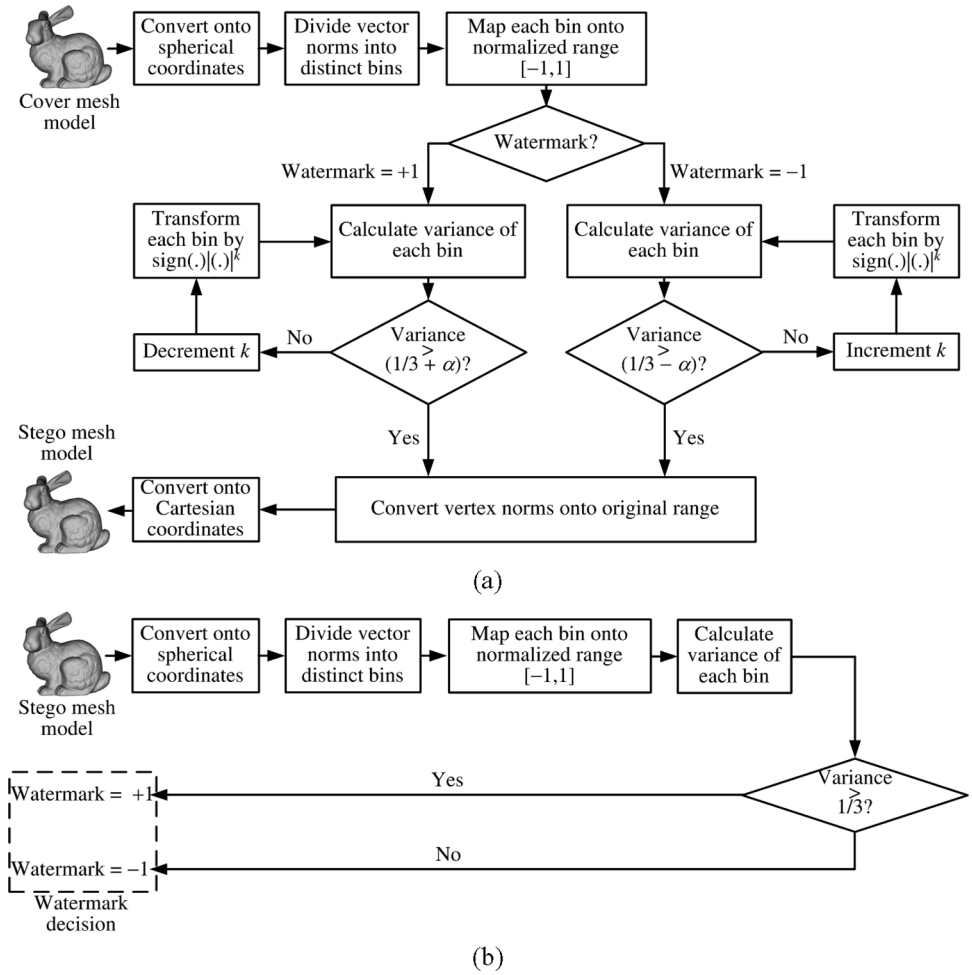


Fig. 7. Block diagrams of the (a) watermark embedding and (b) extraction for the proposed watermarking method changing the variance of vertex norms.

combined bin. A stego mesh model $\mathbf{V}'$ consisting of vertices $v_i' = (x_i', y_i', z_i')$ represented in Cartesian coordinate is obtained by

$$\begin{aligned} x_i' &= \rho_i' \cos\theta_i \sin\phi_i + x_g \\ y_i' &= \rho_i' \sin\theta_i \sin\phi_i + y_g \quad \text{for } 0 \le n \le L-1 \\ z_i' &= \rho_i' \cos\phi_i + z_g \end{aligned} \tag{10}$$

where θ_i, ϕ_i, and the center of gravity are the same as those calculated in the first step.

B. Watermark Extraction

The watermark extraction process is quite simple, as shown in Fig. 4(b). Similar to embedding process, the stego mesh model is first converted to spherical coordinates. After finding the maximum and minimum vertex norms, the vertex norms are classified into N bins and mapped onto the normalized range of [0,1]. Then, the mean of each bin $\widetilde{\mu}_n''$ is calculated and compared to the reference value 1/2. The watermark hidden in the nth bin ω_n'' is extracted by means of

$$\omega_n'' = \begin{cases} +1, \text{ if } \widetilde{\mu}_n'' > \frac{1}{2} \\ -1, \text{ if } \widetilde{\mu}_n'' < \frac{1}{2} \end{cases}. \tag{11}$$

Note that the watermark detection process does not require the original meshes.

IV. Proposed Watermarking Method II

In this method, the variance of vertex norm distribution is changed to hide one bit of watermark in each bin. Again, a histogram mapping function is introduced and applied. Both the watermark embedding and extraction processes of the method are quite similar to method I, as introduced in the previous section and shown in Fig. 7.

A. Watermark Embedding

As the first two steps and the last step of this watermark embedding process are identical to those mentioned in Section III-A, only the unique steps of this method are described in detail. Note that notations have not been changed from the previous section for the sake of simplicity.

In the first and second steps, vertices of cover meshes are represented in spherical coordinate, and vertex norms are divided into N bins such as $\rho_{n,j} \in B_n$ for $0 \le n \le N-1$ and $0 \le j \le M_n - 1$.

In the third step, vertex norms of each bin are mapped into a normalized range similar to method I. However, the range $[-1,1]$ is now mapped by

$$\widetilde{\rho}_{n,j} = 2 \cdot \frac{\left(\rho_{n,j} - \min_{\rho_{n,j} \in \mathbf{B}_n}\{\rho_{n,j}\}\right)}{\max_{\rho_{n,j} \in \mathbf{B}_n}\{\rho_{n,j}\} - \min_{\rho_{n,j} \in \mathbf{B}_n}\{\rho_{n,j}\}} - 1 \tag{12}$$

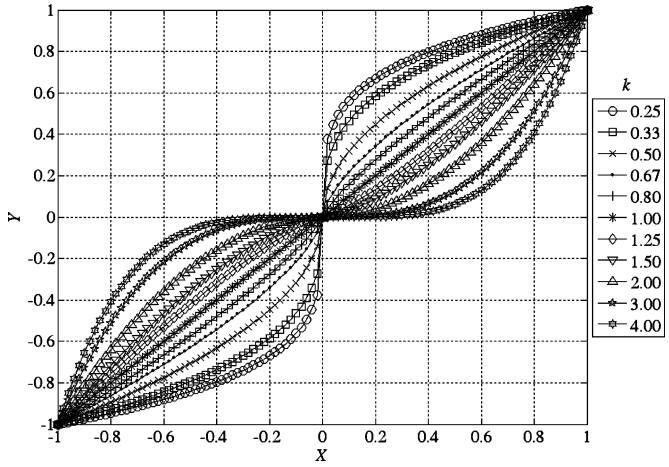


Fig. 8. Histogram mapping function $sign(X)|X|^k$ for different parameter of k.

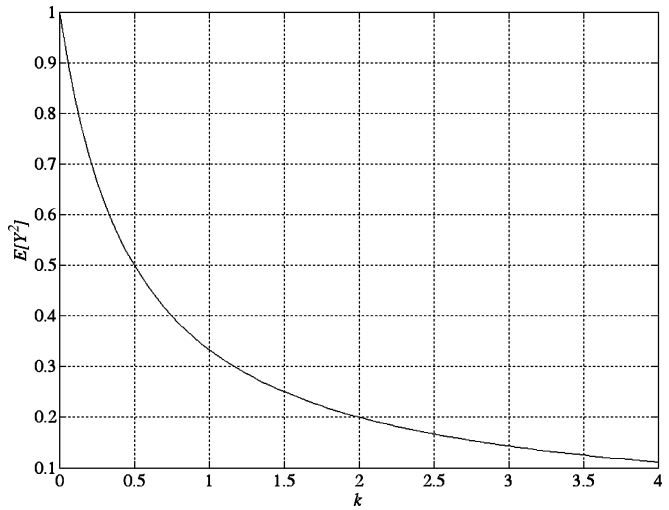


Fig. 9. Variance of the output random variable via histogram mapping function with different k, assuming that the input random variable is uniformly distributed over the normalized range $[-1,1]$.

where $\widetilde{\rho}_{n,j}$ is the jth vertex norm of the nth bin represented in the normalized range. Note that each bin has a nearly uniform distribution over the interval $[-1,1]$.

Now, let X be a continuous random variable with uniform distribution over $[-1,1]$. As X has a mean of zero, its variance is given by

$$E[X^2] = \int_{-1}^{1} x^2 p_X(x)dx = \frac{1}{3} \tag{13}$$

where $E[X^2]$ denotes the second moment of the random variable X. A variance of 1/3 will be used as a reference when changing the variance of each bin according to the watermark bit to be embedded. To change the variance, vertex norms in each bin should be modified within the normalized range of $[-1,1]$. For this purpose, we use a histogram mapping function, which can change the variance to the desired level by modifying vertex norms while staying within the specified range. For a given X, the mapping function is defined by

$$Y = \text{sign}(X)|X|^k \text{ for } 0 < k < \infty \text{ and } k \in \Re \tag{14}$$

where Y is the transformed variable and k is a real value for $0 < k < \infty$. Fig. 8 shows curves of the mapping function for different k. When the parameter is selected for $1 < k < \infty$, the input variable is transformed into an output variable with relatively small absolute value while maintaining its sign. Moreover, the absolute value of transformed variable becomes smaller as k increases. It means a reduction of the variance. On the other hand, variance increases for decreasing k on the range $0 < k < 1$. The variance of the output random variable $E[Y^2]$ is represented as

$$E[Y^2] = E\left[\left(\text{sign}(X)|X|^k\right)^2\right] = \int_{-1}^{1} |x|^{2k} p_X(x)dx = \frac{1}{2k+1}. \tag{15}$$

Fig. 9 shows the variance of the output random variable over k of the mapping function. Note that the variance decreases monotonically as k increases. Therefore, the variance of the distribution can easily be adjusted by selecting a proper parameter.

The fourth step of the watermark embedding process is to change variance of each bin via transforming vertex norms by the histogram mapping function. To embed $\omega_n = +1$, vertex norms $\widetilde{\rho}_{n,j}$ are transformed in order to change the variance of the distribution by a strength factor $\alpha(0 < \alpha < (1/3))$. To embed $\omega_n = -1$, vertex norms are transformed in order to change the variance by a factor $-\alpha$. Then the variance of each bin $\widetilde{\sigma}_n^{2'}$ is changed by

$$\widetilde{\sigma}_n^{2'} = \begin{cases} \frac{1}{3} + \alpha \text{ if } \omega_n = +1 \\ \frac{1}{3} - \alpha \text{ if } \omega_n = -1 \end{cases}. \tag{16}$$

The exact parameter can be found directly from (15)

$$k_n = \begin{cases} \frac{1-3\alpha}{1+3\alpha} \text{ if } \omega_n = +1 \\ \frac{1+3\alpha}{1-3\alpha} \text{ if } \omega_n = -1 \end{cases}. \tag{17}$$

Note that k_n exists in the range of]0,1[when the watermark bit is $+1$ and in the range of]1,∞[when the watermark bit is -1.

As mentioned in Section III-A, (17) is not useful for the real vertex norm distribution. Thus we use an iterative approach as follows.

For embedding $\omega_n = +1$ into the nth bin:

1) initialize the parameter k_n as 1;
2) transform normalized vertex norms by $\widetilde{\rho}_{n,j} = sign(\widetilde{\rho}_{n,j})|\widetilde{\rho}_{n,j}|^{k_n}$;
3) calculate the variance of transformed vertex norms through $\widetilde{\sigma}_n^{2'} = (1/M_n)\sum_{j=0}^{M_n-1} \widetilde{\rho}_{n,j}^{2}$;
4) if $\widetilde{\sigma}_n^{2'} < (1/3) + \alpha$, decrease $k_n (k_n = k_n - \Delta k)$ and go back to 2);
5) replace normalized vertex norms with transformed norms using $\widetilde{\rho}_{n,j} = \widetilde{\rho}'_{n,j}$;
6) end.

For embedding $\omega_n = -1$ into the nth bin:

4) if $\widetilde{\sigma}_n^{2'} > (1/3) - \alpha$, increase $k_n (k_n = k_n + \Delta k)$ and go back to 2).

The fifth step is to map each bin onto the original range using

$$\rho'_{n,j} = \frac{1}{2} \cdot (\widetilde{\rho}'_{n,j} + 1) \cdot \left(\max_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\} - \min_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\} \right) + \min_{\rho_{n,j} \in \mathbf{B}_n} \{\rho_{n,j}\}. \tag{18}$$

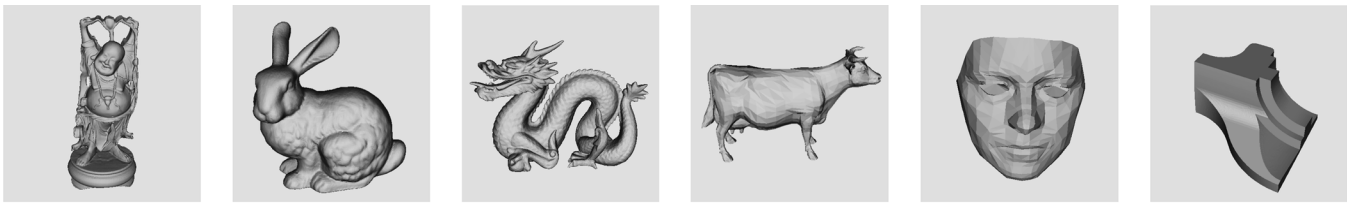


Fig. 10. Original mesh models (a) buddha, (b) bunny, (c) dragon, (d) cow, (e) face, and (f) fandisk.

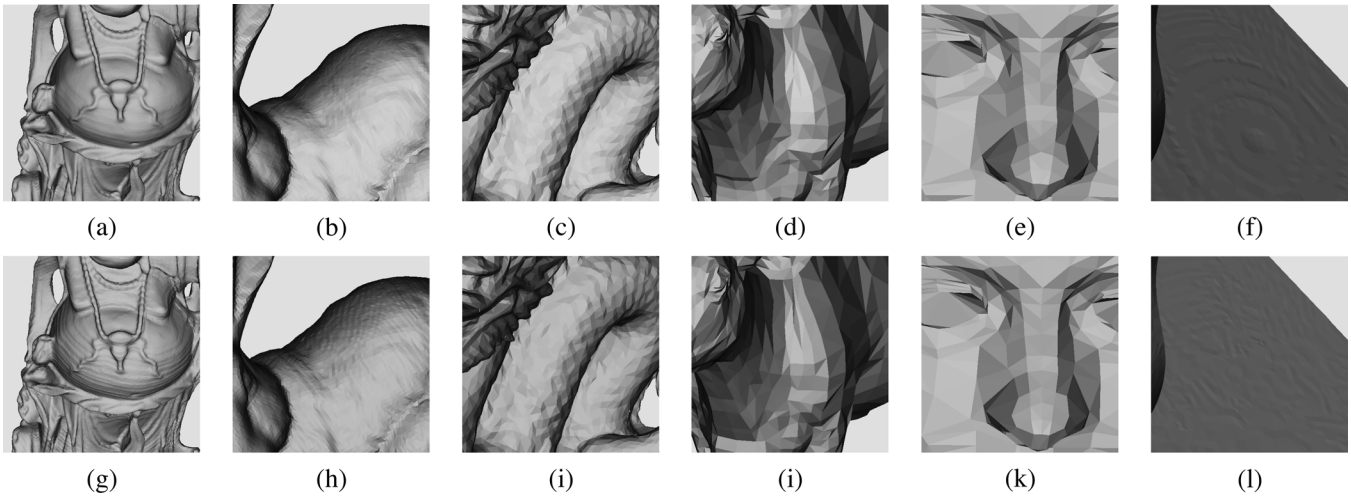


Fig. 11. Stego mesh models, where (a)–(f) are watermarked by method I and (g)–(l) by method II.

Finally, the watermark embedding process is completed by combining all of the bins and converting the spherical coordinates to Cartesian coordinates using (10).

B. Watermark Extraction

The watermark extraction process for this method is also quite simple, as illustrated in Fig. 7(b). The variance of each bin $\widetilde{\sigma}_n^{2''}$ is calculated and compared with the reference value 1/3. The watermark hidden in the nth bin ω_n'' is extracted by means of

$$\omega_n'' = \begin{cases} +1, \text{ if } \sigma_n^{2''} > \frac{1}{3} \\ -1, \text{ if } \sigma_n^{2''} < \frac{1}{3} \end{cases}. \tag{19}$$

Note that this watermark extraction process is performed without the cover mesh model.

V. Simulation Results

A. Baseline Simulations

Simulations were carried out on six 3-D triangular mesh models: a buddha (with 543 652 vertices and 1 087 716 cells), a bunny (with 35 947 vertices and 69 451 cells), a dragon (with 15 574 vertices and 29 999 cells), a cow (with 2903 vertices and 5804 cells), a face (with 539 vertices and 1042 cells), and a fandisk (with 6475 vertices and 12 946 cells) as shown in Fig. 10. The quality of the geometry of the 3-D mesh model was measured by Metro [28], which provides the forward and backward root mean square (rms) errors $e_f(\mathbf{V}, \mathbf{V}')$ and $e_b(\mathbf{V}', \mathbf{V})$, respectively. We used the maximum value between the two rms values, called maximum rms (mrms), as the quality measure. The mrms, denoted as $E(\mathbf{V}, \mathbf{V}')$, is calculated by

$$E(\mathbf{V}, \mathbf{V}') = \max\left\{e_f(\mathbf{V}, \mathbf{V}'), e_b(\mathbf{V}', \mathbf{V})\right\} \tag{20}$$

where

$$e_f(\mathbf{V}, \mathbf{V}') = \frac{1}{|\mathbf{V}|} \int_{\mathbf{V}} \left\{ \min_{v' \in \mathbf{V}'} \{\|v - v'\|\} \right\} dv$$

$$e_b(\mathbf{V}', \mathbf{V}) = \frac{1}{|\mathbf{V}'|} \int_{\mathbf{V}'} \left\{ \min_{v \in \mathbf{V}} \{\|v' - v\|\} \right\} dv'$$

and $\mathbf{V}$ and $\mathbf{V}'$ represent, respectively, the original and deformed meshes. The robustness of the watermark is measured in terms of correlation between the detected watermark and the designated one

$$\text{Corr} = \frac{\sum_{n=0}^{N-1} (\omega_n'' - \varpi'')(\omega_n - \varpi)}{\sqrt{\sum_{n=0}^{N-1} (\omega_n'' - \varpi'')^2 \times \sum_{n=0}^{N-1} (\omega_n - \varpi)^2}} \tag{21}$$

where ϖ indicates the average of the watermark and Corr is on the range of [−1,1].

In the simulations, we embedded 64 bits of watermark into a mesh model considering the tradeoff between the robustness and the transparency of watermark. Then, vertex norms were divided into 64 bins and one bit of watermark was hidden in each bin. For comparison of the two proposed methods, the strength factor of the watermark was determined experimentally so that both methods have very similar quality for each model in terms of mrms. Fig. 11 shows the stego mesh models, of which the performances are listed in Table I in terms of mrms and Corr

TABLE I
EVALUATION OF STEGO MESHES WHEN NO ATTACK

Method	Model	Strength factor	$E(\mathbf{V}, \mathbf{V}')$	$Corr$
Method I	buddha	0.03	0.38×10^{-4}	1.00
	bunny	0.03	0.40×10^{-4}	1.00
	dragon	0.04	0.45×10^{-4}	1.00
	cow	0.16	5.84×10^{-3}	1.00
	face	0.16	1.41×10^{-2}	0.43
	fandisk	0.01	7.32×10^{-4}	1.00
Method II	buddha	0.06	0.35×10^{-4}	1.00
	bunny	0.07	0.41×10^{-4}	1.00
	dragon	0.11	0.48×10^{-4}	1.00
	cow	0.26	5.91×10^{-3}	1.00
	face	0.28	1.27×10^{-2}	0.49
	fandisk	0.05	7.85×10^{-4}	1.00

TABLE II
THE NUMBER OF ITERATIONS AND THE PRECISION ERROR WHEN $\Delta k = 1 \times 10^{-3}$

Method	Model	Av. # of iteration (Av. operating time(ms))	Av. precision error
Method I	buddha	135(898.48)	5.38×10^{-3}
	bunny	154(83.09)	1.29×10^{-3}
	dragon	205(57.91)	1.60×10^{-3}
	cow	783(33.48)	6.46×10^{-4}
	face	761(9.69)	2.54×10^{-3}
	fandisk	111(9.55)	7.66×10^{-3}
Method II	buddha	275(2,728.63)	5.88×10^{-4}
	bunny	326(400.11)	1.22×10^{-4}
	dragon	529(297.45)	1.44×10^{-4}
	cow	12,487(865.30)	1.90×10^{-4}
	face	4,339(54.92)	1.02×10^{-3}
	fandisk	211(43.34)	2.46×10^{-3}

when no attack. Here, the strength factor of each watermark is also listed. The table shows that the statistical approach employed in the proposed cannot embed a watermark bit into every bin in the case of very small size models such as face. This is mainly caused by the fact that some of the bins are empty or do not contain enough vertices. For this reason, the proposed methods are not recommended to be applied to such small size models (approximately having under 2000 vertices). However, the hidden watermark can be extracted perfectly from all stego models except for the smallest size model. This means the proposed methods guarantee to hide a watermark bit into every bin for models with a sufficient number of vertices. From the viewpoint of watermark transparency, method I maintains better visual quality than method II. Some artifacts appear in smooth regions such as in the lower belly of buddha and the rump of the bunny as shown in Fig. 11(g) and (h). In particular, the artifacts are conspicuous in the flat regions of fandisk, even when small strength factor is applied. This is mainly due to the fact that every vertex is modified without considering local curvature of the models. It is also caused by discontinuities in the boundaries of neighbor bins when the distribution is modified. As results, the proposed methods are not applicable to CAD models with a flat region. Consequently, attack simulations have been performed with buddha, bunny, dragon, and cow.

In both our embedding processes, the step size for parameter k_n was used as $\Delta k = 0.001$. The step size was experimentally selected. Table II shows the average number of iterations per bin with its processing time in brackets. The experiments were realized on a Pentium IV 3.4 GHz processor. Average precision error between the desired and actual modified mean (or variance) is also listed. In this case, the execution time for watermark embedding was measured between 0.5 and 192.4 s, depending on the size of the model and the watermark strength factor.

B. Attack Simulations

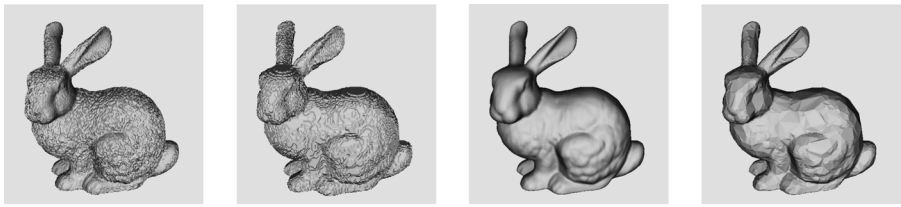
To evaluate the robustness of the watermark, various distortion and distortionless attacks were performed on the stego meshes. Each attack was applied with varying attack strengths. Distortion attacks including additive binary random noise, uniform quantization, smoothing, simplification, and subdivision were carried out. As examples, stego bunny models deformed by various distortion attacks are shown in Fig. 12.

For evaluating the resistance to noise attack, binary random noise was added to each vertex of stego model with three different error rates: 0.1%, 0.3%, and 0.5%. Here, the error rate represents the noise amplitude as a fraction of the maximum vertex norm of the object. We perform each noise attack five times using different random seeds and report the median as shown in Table III. The effect of the noise attack is shown in Fig. 12(a). The performance of method II decreases faster than that of method I as the error rate increases. Both methods are fairly resistant to the noise attacks under an error rate of 0.3, but good watermark detection cannot be expected for higher error rates. This is due to the fact that the additive noise essentially alters the distribution of vertex norms in the divided bins. In addition, more vertex norms exceed the range of each bin as the noise error rate increases. Similar tendency was observed in quantization and smoothing attacks. For such reasons, the robustness cannot be enhanced beyond a certain level even when the strength factor α increases. The robustness can also be improved by widening the size (width) of the bin, but the transparency of watermark and the number of bits to be embedded should be considered.

To evaluate the robustness against uniform quantization attacks, three different quantization rates are applied to stego meshes. Each coordinate of vertices is represented with 7, 8, and 9 bits. Table IV shows the robustness against the quantization attack. An example is shown in Fig. 12(b). Both methods are fairly robust up to 8 bit quantization. Similar to the case of noise attack, method II has relatively abrupt diminution of the robustness as the quantization step size increases.

Table V shows the performance of the watermarking schemes after smoothing attacks [30]. Three different pairs of iteration and relaxation were applied. An example of the attack is also shown in Fig. 12(c), where the effect can be seen in the rounded edges. The robustness depends on the smoothness of the original meshes. Buddha and bunny are relatively less sensitive to smoothing attacks.

To evaluate the robustness of our methods against simplification attacks, we utilized a simplification method [31].



(a) (b) (c) (d)

Fig. 12. Bunny model watermarked by method I and attacked by (a) adding binary noise with error ratio of 0.5%, (b) 7 bit/coordinate quantization, (c) smoothing with iteration of 50 and relaxation of 0.03, and (d) simplification with reducing 90.65% of vertices.

TABLE III
EVALUATION OF ROBUSTNESS AGAINST ADDITIVE BINARY NOISE ATTACKS

Method	Model	Error rate	$E(\mathbf{V},\mathbf{V}')$	$Corr$
Method I	buddha	0.1%	0.54×10^{-4}	0.94
		0.3%	1.26×10^{-4}	0.73
		0.5%	1.96×10^{-4}	0.41
	bunny	0.1%	0.59×10^{-4}	0.87
		0.3%	1.47×10^{-4}	0.51
		0.5%	2.38×10^{-4}	0.18
	dragon	0.1%	0.68×10^{-4}	1.00
		0.3%	1.49×10^{-4}	0.55
		0.5%	2.37×10^{-4}	0.21
	cow	0.1%	6.27×10^{-3}	0.97
		0.3%	8.95×10^{-3}	0.91
		0.5%	1.27×10^{-2}	0.41
Method II	buddha	0.1%	0.54×10^{-4}	1.00
		0.3%	1.26×10^{-4}	0.81
		0.5%	1.96×10^{-4}	−0.29
	bunny	0.1%	0.59×10^{-4}	1.00
		0.3%	1.47×10^{-4}	0.53
		0.5%	2.38×10^{-4}	−0.39
	dragon	0.1%	0.70×10^{-4}	1.00
		0.3%	1.49×10^{-4}	0.65
		0.5%	2.37×10^{-4}	−0.34
	cow	0.1%	6.30×10^{-3}	1.00
		0.3%	8.74×10^{-3}	0.50
		0.5%	1.29×10^{-2}	0.22

TABLE IV
EVALUATION OF ROBUSTNESS AGAINST UNIFORM QUANTIZATION ATTACKS

Method	Model	Quantization	$E(\mathbf{V},\mathbf{V}')$	$Corr$
Method I	buddha	9bits	0.64×10^{-4}	0.87
		8bits	1.21×10^{-4}	0.78
		7bits	2.42×10^{-4}	0.47
	bunny	9bits	0.69×10^{-4}	0.94
		8bits	1.26×10^{-4}	0.88
		7bits	2.46×10^{-4}	0.39
	dragon	9bits	0.75×10^{-4}	0.94
		8bits	1.30×10^{-4}	0.84
		7bits	2.48×10^{-4}	0.43
	cow	9bits	6.43×10^{-3}	0.91
		8bits	7.81×10^{-3}	0.94
		7bits	1.22×10^{-2}	0.51
Method II	buddha	9bits	0.64×10^{-4}	1.00
		8bits	1.21×10^{-4}	0.97
		7bits	2.42×10^{-4}	0.19
	bunny	9bits	0.73×10^{-4}	1.00
		8bits	1.28×10^{-4}	0.97
		7bits	2.46×10^{-4}	−0.01
	dragon	9bits	0.82×10^{-4}	1.00
		8bits	1.34×10^{-4}	1.00
		7bits	2.49×10^{-4}	0.72
	cow	9bits	6.56×10^{-3}	1.00
		8bits	8.07×10^{-3}	0.72
		7bits	1.23×10^{-2}	0.11

Watermarked models were simplified by various reduction ratios. Table VI demonstrates that the proposed methods are robust against simplification attacks. In addition, method II is more robust than method I. In this table, the percentage represents the number of vanished vertices as a fraction of total number of vertices. Fig. 12(d) shows an example of the simplification attack. Subdivision attacks were also carried out. Each triangle was uniformly divided into four cells. The performance is listed in Table VII. The results show that the proposed methods are robust against subdivision attacks, similar to attacks using simplification. The results demonstrate that the distribution of vertex norms is less sensitive to changes in the number of vertices. Clearly, this is an additional advantage of the statistical approach. However, clipping attack simulation shows that the proposed are very vulnerable to such attacks that cause severe alteration to the center of gravity of the model.

To evaluate the robustness of our methods against distortionless attacks, vertex reordering and similarity transforms were carried out. Vertex reordering attack was performed iteratively 100 times, also changing the seed of random number generator for each iteration. Similarity transforms were carried out with many combinations of rotation, uniform scaling, and translation factors. It is not necessary to tabulate watermark detection performance because both proposed perfectly extracted the hidden watermark information. As intended, the proposed watermarking methods are perfectly robust against distortionless attacks.

C. Parameters for Robustness

In this section, we analyze two parameters that can be adjusted to improve the robustness of the proposed methods. One is the watermark strength factor α; another is the size of bin. For these analysis, the bunny model and method I were used. Smoothing operation with iteration of 30 and relaxation of 0.03 was applied as an example attack. To analyze the effect of watermark strength factor, the bunny model was watermarked with

TABLE V
EVALUATION OF ROBUSTNESS AGAINST SMOOTHING ATTACKS

Method	Model	(# of iteration, relaxation)	$E(\mathbf{V},\mathbf{V}')$	$Corr$
Method I	buddha	(10,0.03)	0.32×10^{-4}	1.00
		(30,0.03)	0.33×10^{-4}	1.00
		(50,0.03)	0.36×10^{-4}	0.94
	bunny	(10,0.03)	0.41×10^{-4}	0.75
		(30,0.03)	0.80×10^{-4}	0.57
		(50,0.03)	1.21×10^{-4}	0.46
	dragon	(10,0.03)	0.76×10^{-4}	0.62
		(30,0.03)	1.85×10^{-4}	0.39
		(50,0.03)	2.88×10^{-4}	0.24
	cow	(10,0.03)	1.03×10^{-2}	0.69
		(30,0.03)	2.43×10^{-2}	0.23
		(50,0.03)	3.67×10^{-2}	0.17
Method II	buddha	(10,0.03)	0.31×10^{-4}	1.00
		(30,0.03)	0.31×10^{-4}	1.00
		(50,0.03)	0.33×10^{-4}	1.00
	bunny	(10,0.03)	0.42×10^{-4}	0.97
		(30,0.03)	0.81×10^{-4}	0.87
		(50,0.03)	1.22×10^{-4}	0.75
	dragon	(10,0.03)	0.79×10^{-4}	0.97
		(30,0.03)	1.89×10^{-4}	0.27
		(50,0.03)	2.92×10^{-4}	0.18
	cow	(10,0.03)	1.00×10^{-2}	0.42
		(30,0.03)	2.40×10^{-2}	0.11
		(50,0.03)	3.64×10^{-2}	0.11

varying the strength factor and underwent the smoothing attack. Here, 64 bits of watermark were embedded. Fig. 13(a) shows the correlation of watermark detection along different strength factors. Corresponding mrms of watermarked meshes is also plotted. It shows that the robustness can be improved to a certain limited level as the strength factor increases. However, the watermark transparency should be carefully considered, as mrms increases linearly. Fig. 13(b) shows the relationship between the size of bin and the correlation, where the strength factor is used as $\alpha = 0.03$. Note that the size of the bin is inversely proportional to the number of bins. This shows that the robustness can also be improved as the size of the bin increases. In other words, this means that the use of larger bins reduces the probability that vector norms exceed the corresponding bin when being attacked by smoothing operations. However, watermark transparency should be also carefully considered. Note that the use of larger bins limits the number of watermark bits.

D. ROC Analysis

The proposed methods were analyzed by a receiver operating characteristic (ROC) curve that represents the relation between probability of false positives P_{fp} and probability of false negatives P_{fn} by varying the decision threshold T_{Corr} for declaring the watermark present [1]. The probability density functions for P_{fp} and P_{fn} were measured experimentally with 100 correct and 100 wrong keys and approximated to Gaussian distribution. In these simulations, we used the same stego model of bunny as used in Section V-B. Fig. 14 shows the ROC curves when additive binary noise and simplification attacks are respectively applied into stego model of bunny. Equal error rate is also indicated in this figure. As shown in the figure, the proposed methods have fairly good performance in terms of watermark detection for both attacks.

TABLE VI
EVALUATION OF ROBUSTNESS AGAINST SIMPLIFICATION ATTACKS

Method	Model	Reduction ratio	$E(\mathbf{V},\mathbf{V}')$	$Corr$
Method I	buddha	30.02%	0.38×10^{-4}	1.00
		50.02%	0.39×10^{-4}	0.94
		70.03%	0.41×10^{-4}	0.84
		90.10%	1.05×10^{-4}	0.75
	bunny	32.11%	0.44×10^{-4}	0.94
		51.44%	0.52×10^{-4}	0.77
		70.79%	0.70×10^{-4}	0.58
		90.65%	3.44×10^{-4}	0.38
	dragon	30.36%	0.98×10^{-4}	0.76
		50.97%	1.91×10^{-4}	0.75
		63.74%	3.43×10^{-4}	0.44
		82.46%	7.47×10^{-4}	0.22
	cow	30.01%	1.02×10^{-2}	0.46
		44.54%	2.09×10^{-2}	0.46
		58.77%	4.31×10^{-2}	0.46
		75.41%	6.83×10^{-2}	0.17
Method II	buddha	30.02%	0.35×10^{-4}	1.00
		50.02%	0.35×10^{-4}	1.00
		70.46%	0.37×10^{-4}	1.00
		90.01%	0.92×10^{-4}	0.97
	bunny	32.10%	0.44×10^{-4}	1.00
		51.43%	0.52×10^{-4}	0.97
		70.78%	0.70×10^{-4}	0.94
		89.71%	3.35×10^{-4}	0.79
	dragon	30.42%	0.99×10^{-4}	1.00
		50.97%	1.96×10^{-4}	1.00
		63.81%	3.48×10^{-4}	0.71
		81.70%	8.52×10^{-4}	0.53
	cow	30.01%	1.08×10^{-2}	1.00
		43.62%	2.13×10^{-2}	0.51
		58.63%	2.13×10^{-2}	0.37
		76.34%	2.13×10^{-2}	0.25

TABLE VII
EVALUATION OF ROBUSTNESS AGAINST 1:4 SUBDIVISION ATTACKS

Method	Model	# of cells	$E(\mathbf{V},\mathbf{V}')$	$Corr$
Method I	buddha	4,350,864	0.38×10^{-4}	1.00
	bunny	277,804	0.34×10^{-4}	0.87
	dragon	119,996	2.80×10^{-4}	0.62
	cow	11,609	5.78×10^{-3}	0.58
Method II	buddha	4,350,864	0.35×10^{-4}	1.00
	bunny	277,804	0.41×10^{-4}	0.94
	dragon	119,996	2.82×10^{-4}	1.00
	cow	11,609	5.29×10^{-3}	0.61

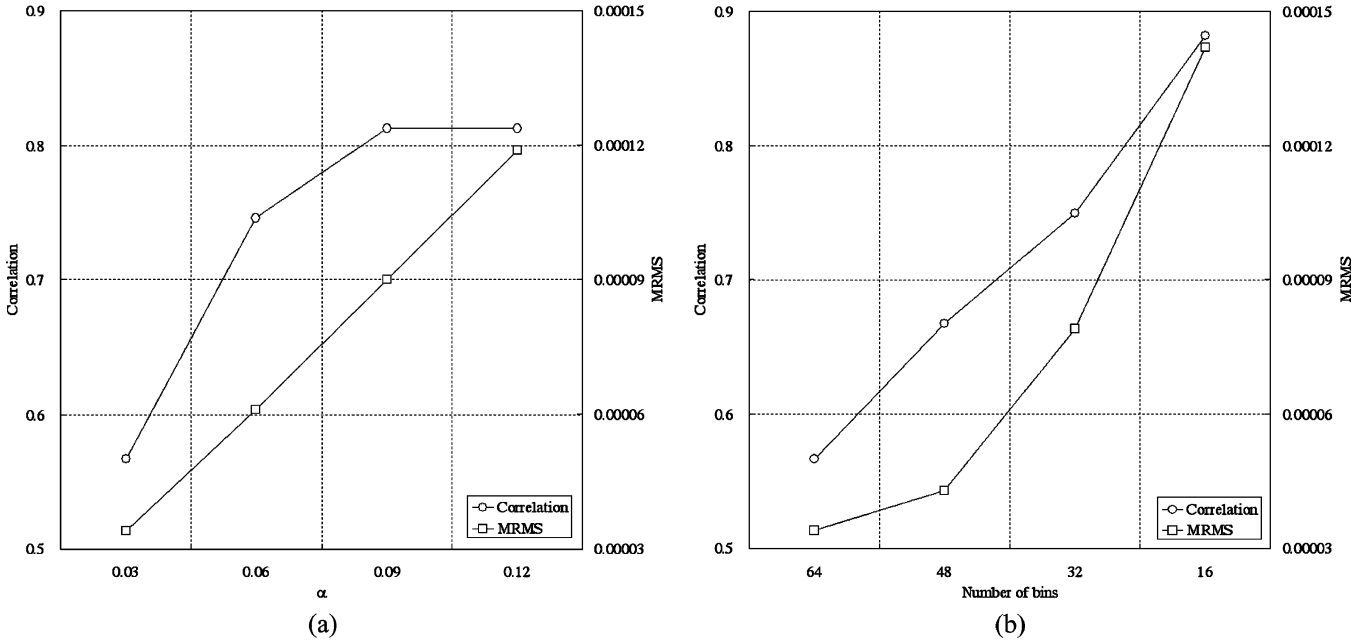


Fig. 13. Relationship (a) between the strength factor and the correlation and (b) between the number of bins and the correlation. As an example, a smoothing attack with iteration 30 and relaxation 0.03 is applied.

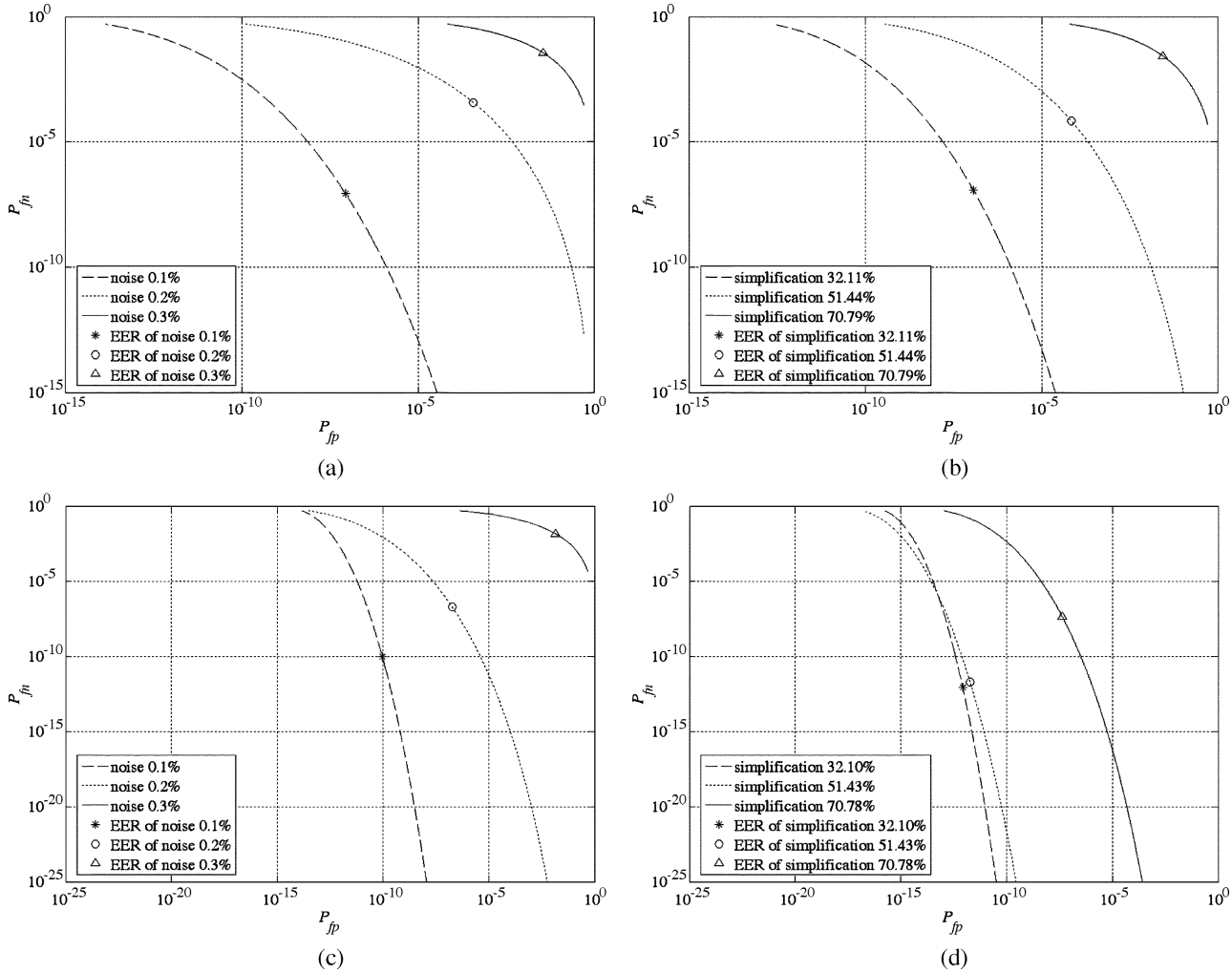


Fig. 14. ROC curves of bunny model (a) watermarked by method I and attacked by adding noise, (b) watemarked by method I and attacked by simplification, (c) watermarked by method II and attacked by adding noise, and (d) watemarked by method II and attacked by simplification.

VI. Conclusion

In this paper, we proposed two statistical watermarking methods for 3-D polygonal mesh models that modify the distribution of vertex norms via changing respectively the mean and the variance of each bin by histogram mapping function. Through the simulations, we proved that both proposed methods are perfectly robust against distortionless attack such as vertex reordering and similarity transforms. Moreover, they are fairly robust against various kinds of distortion attacks, in particular, simplification and subdivision operations. However, there are some drawbacks. Our proposals are not applicable to very small size models and CAD models with flat regions and are very vulnerable to clipping attacks that cause severe alteration to the center of gravity of the model. Nevertheless, the simulation results demonstrate a possible, oblivious watermarking method based on a statistical approach for 3-D polygonal mesh model.

Acknowledgment

The authors would like to thank the Computer Graphics Laboratory of Stanford University, Cyberware, Brian Curless, D. Terzopoulos, and the Viewpoint Animation Engineering of Sun Microsystems for allowing them to use the 3-D mesh models. The authors would like to thank the reviewers for their helpful comments, suggestions and related references. Their criticisms helped the authors to improve the quality of this paper.

References

[1] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proc. SIGGRAPH99*, Los Angeles, CA, Aug. 1999, pp. 49–56.

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

[3] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, no. 3/4, pp. 313–336, 1996.

[4] D. Gruhl and W. Bender, "Echo hiding," in *Proc. Inf. Hiding Workshop*, Cambridge, U.K., Dec. 1996, pp. 295–315.

[5] J. W. Cho, H. J. Park, Y. Huh, H. Y. Chung, and H. Y. Jung, "Echo watermarking in sub-band domain," in *Digital Watermarking, LNCS2939*, Mar. 2004, pp. 447–455.

[6] Y. S. Seo, S. H. Joo, and H. Y. Jung, "An efficient quantization watermarking on the lowest wavelet subband," *IEICE Trans. Fundamentals*, vol. E86-A, no. 8, pp. 2053–2055, Aug. 2003.

[7] I. J. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, pp. 1673–1687, Jun. 1997.

[8] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale and translation resilient public watermarking for images," in *Proc. SPIE*, San Jose, CA, Jan. 2000, vol. 3971, no. 2, pp. 90–98.

[9] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Process. (Special Issue on Copyright Protection and Control)*, vol. 66, no. 3, pp. 283–301, May 1998.

[10] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 551–560, May 1998.

[11] O. Beneden, "Geometry-based watermarking of 3D models," *IEEE Comput. Graph. Applicat.*, vol. 19, no. 1, pp. 46–55, Jan./Feb. 1999.

[12] S. H. Lee, T. S. Kim, B. J. Kim, S. G. Kwon, K. R. Kwon, and K. I. Lee, "3D polygonal meshes watermarking using normal vector distributions," in *IEEE Int. Conf. Multimedia Expo*, Jul. 6–9, 2003, vol. 3, pp. 105–108.

[13] Z. Yu, H. S. Ip, and L. F. Kwok, "A robust watermarking scheme for 3D triangular mesh models," *Pattern Recognit.*, vol. 36, no. 11, pp. 2603–2614, 2003.

[14] ——, "Robust watermarking of 3D polygonal models based on vertice scrambling," in *Comput. Graphics Int. 2003*, Tokyo, Japan, Jul. 9–11, 2003, pp. 254–257, 2003.

[15] S. Kanai, H. Date, and T. Kishinami, "Digital watermarking for 3D polygons using multiresolution wavelet decomposition," in *Proc. 6th IFIP*, Tokyo, Japan, Dec. 1998, pp. 296–307, WG 5.2, GEO-6.

[16] K. Yin, Z. Pan, J. Shi, and D. Zhang, "Robust mesh watermarking based on multiresolution processing," *Comput. Graph.*, vol. 25, pp. 409–420, 2001.

[17] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, "Watermarking 3D polygonal meshes in the mesh spectral domain," in *Proc. Graph. Interface*, Ottawa, ON, Canada, Jun. 2001, pp. 9–17.

[18] D. Cotting, T. Weyrich, M. Pauly, and M. Gross, "Robust watermarking of point-sampled geometry," in *IEEE Int. Conf. Shape Modeling Int. 2004*, 2004, pp. 233–242.

[19] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "Watermarking a 3D shape model defined as a point set," in *IEEE Int. Conf. Cyber Worlds 2004*, Tokyo, Japan, Nov. 2004, pp. 392–399.

[20] A. Kejariwal, "Watermarking," *IEEE Potentials*, pp. 37–40, Oct./Nov. 2003.

[21] P. H. W. Wong, O. C. Au, and Y. M. Yeung, "A novel blind multiple watermarking technique for images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 813–830, Aug. 2003.

[22] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Mateo, CA: Morgan Kaufman, 2001.

[23] S. Craver, N. Memon, B. L. Yeo, and M. Yeung, "Can invisible watermarks resolve rightful ownerships?," in *Proc. IS&T/SPIE Conf. Storage Retrieval Image Video Database V*, San Jose, CA, Feb. 13–14, 1997, vol. 3022, pp. 310–321.

[24] W. G. Kim, J. C. Lee, and W. D. Lee, "An image watermarking scheme with hidden signatures," in *Proc. IEEE Int. Conf. Image Process.*, Kobe, Japan, Oct. 24–28, 1999, vol. 2, pp. 206–210.

[25] M. G. Wagner, "Robust watermarking of polygonal meshes," in *Proc. Geometric Modeling Process. 2000*, Hong Kong, China, Apr. 10–12, 2000, pp. 201–208.

[26] J. Jian-qiu, D. Min-ya, B. Hu-jun, and P. Qun-sheng, "Watermarking on 3D mesh based on spherical wavelet transform," in *JZUS*, 2004, vol. 5, no. 3, pp. 251–258.

[27] J. W. Cho, M. S. Kim, R. Prost, H. Y. Chung, and H. Y. Jung, "Robust watermarking on polygonal meshes using distribution of vertex norms," in *Digital Watermarking (LNCS3304)*, Mar. 2005, pp. 283–293.

[28] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," in *Comput. Graph. Forum*, Jun. 1998, vol. 17, no. 2, pp. 167–174.

[29] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.

[30] D. A. Field, "Laplacian smoothing and delaunay triangulation," *Commun. Appl. Numer. Meth.*, vol. 4, pp. 709–712, 1988.

[31] W. J. Shroder, J. A. Zarge, and W. E. Loresen, "Decimation of triangle meshes," in *Proc. Siggraph'92*, 1992, pp. 65–70.

Jae-Won Cho received the B.S. and M.S. degrees from the Information and Communication Engineering Department, University of Yeungnam, Korea, in 2002 and 2004, respectively. He is currently pursuing the international cooperative Ph.D. degree between the Institut National des Sciences Appliquées de Lyon, France, and the University of Yeungnam.

His research interests include multimedia signal processing, digital watermarking, quality measurement for transmitted multimedia data, and compression of dynamic three-dimensional meshes.

Rémy Prost received the Ph.D. degree in electronics engineering from Lyon University, France, in 1977 and the Docteur es Sciences degree from the Institut National des Sciences Appliquées (INSA) de Lyon, France, in 1987.

He is currently a Professor in the Department of Electrical Engineering, INSA de Lyon, France. Both his teaching and research interests include digital signal processing, inverse problems, image data compression, multiresolution algorithms, wavelets, and mesh processing.

Ho-Youl Jung received the Ph.D. degree in electronics engineering from the Institut National des Sciences Appliquées de Lyon, France, in 1998.

He is currently an Associate Professor in the Department of Electrical Engineering and Computer Science, Yeungnam University, Korea. Both his teaching and research interests include digital signal processing, wavelets, watermarking, JPEG/JPEG-2000, quality measurement for transmitted multimedia data, and computer graphics (three-dimensional mesh processing).