Geometric Multiple Watermarking Scheme for Mobile 3D Content Based on Anonymous Buyer-Seller Watermarking Protocol

Suk-Hwan Lee[1], Seong-Geun Kwon[2*] and Ki-Ryong Kwon[3]
[1] Department of Information Security, Tongmyong University
535, Yongdangdong, Namug, Busan, Korea
[e-mail: skylee@tu.ac.kr]
[2] Department of Electronics Engineering, KyungIl University
50, Gamasil-Gil, Hayang-Eup, Gyeongsan-Si, GyeongBuk, Korea
[e-mail: sgkwon@kiu.ac.kr]
[3] Department of IT Convergene and Application Engineering, Pukyong National University
45, Yongso-ro, Busan, Korea
[e-mail: krkwon@pknu.ac.kr]
*Corresponding author: Seong-Geun Kwon

Received August 29, 2013; revised May 14, 2013; accepted January 24, 2014; published February 28, 2014

Abstract

This paper presents a multiple-watermarking scheme for copyright protection and the prevention of illegal copying of mobile 3D contents with low resolution. The proposed scheme embeds a unique watermark and a watermark certification authority (WCA) watermark into the spatial and encryption domains of a mobile 3D content based on the buyer–seller watermarking protocol. The seller generates a unique watermark and embeds it into the local maximum curvedness of a 3D object. After receiving the encrypted watermark from the WCA, the seller embeds it into the encrypted vertex data using an operator that satisfies the privacy homomorphic property. The proposed method was implemented using a mobile content tool, and the experimental results verify its capability in terms of copyright protection and the prevention of illegal copying.

Keywords: Mobile 3D Content, 3D Watermarking, Buyer-Seller Watermarking (BSW) Protocol, Copyright Protection, Curvedness

This work was supported by supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2011-0023118), under the framework of international cooperation program managed by National Research Foundation of Korea (2012K2A1A2032979) and the Busan Metropolitan City, Korea, under the 2013 Brain Busan 21 program grants.

http://dx.doi.org/10.3837/tiis.2014.02.010

1. Introduction

Along with the rapid propagation of high-performance smartphones and tablet PCs, mobile 3D engines for cross- or multi-platform use have recently interested. Typical mobile 3D engines include the SIO2 Engine [1], ShiVa3D [2], Unity3D [3], Corona SDK [4], and Unreal Engine 3 [5]. The security threat inherent to smartphones and the market expansion of mobile 3D content have produced the need for the development of security technology for mobile content. Digital rights management (DRM) for smartphone content, Authentec [6], BuyDRM [7], and Discretix [8] have been developed by supporting the PlayReady DRM [9] of Microsoft on the Android and iOS platforms. However, the latest mobile DRMs have focused on the secure transmission of video content on smartphones or tablet PCs, rather than on mobile 3D content.

With the proliferation of the 3D content industry, 3D watermarking schemes have been presented for copyright protection and piracy tracing, including a 3D graphic model [10]-[16], 3D CAD data [17], and 3D animation model [18],[19]. Most 3D watermarking systems use the coordinates of the vertices and the connectivity of the meshes for embedding a target [10]-[16]. Similarly, 3D hashing schemes have been presented for the authentication and copy detection of a 3D model [20]. However, these schemes are not suitable for securing mobile 3D content because they are more complex and can be applied to 3D models with a high resolution of huge vertices. Therefore, mobile 3D watermarking for mobile 3D content with low resolution is also needed.

Along with digital watermarking, cryptographic protocols have been researched to provide a contract process that is trustworthy to both buyers and sellers of e-commerce. One of these protocols is the buyer–seller watermarking (BSW) protocol combining an encryption technique with a watermarking technique [21]-[26]. Kwon et al. [27] presented a watermarking scheme for mobile 3D content based on Memon’s watermarking protocol [21]. However, this scheme is not robust to data rearrangements and data cropping, and does not consider the geometrical property of the vertex coordinate, which is a base component of a 3D object. Furthermore, Memon’s protocol used in this scheme has both an unbinding problem and a buyer anonymity problem. Lei et al. [22] presented an anonymous BSW protocol based on a public-key infrastructure (PKI) to solve the customer rights problem and unbinding problem. In the former problem, a malicious seller can forge a copyright violation to entrap a buyer. In the latter problem, a specific watermark cannot be combined to a specific contract or content in a trustworthy manner. It is therefore possible for a seller to embed a watermark embedded in the copied content into another contents. Above them, Katzenbeisser et al. [23] presented a secure embedding scheme in a BSW protocol to reduce large computations and bandwidth overhead using a homomorphic public-key cipher. Rial et al. [24] proposed a formal security definition for copyright protection protocols and analyzed an anonymous BSW protocol. The authors also [24] presented the design of a privacy-preserving BSW protocol.

In this paper, we introduce a multiple-watermarking scheme for a secure and trustworthy contract of mobile 3D content based on the anonymous BSW protocol. Our scheme can also be extended to other watermarking protocols [23],[24], and embeds multiple watermarks into the vertex data of mesh objects in both the spatial and encryption domains. A seller embeds a unique watermark of the copyright information into the curvedness of the vertices based on a non-uniform quantization, and encrypts an object with a unique watermark. The seller then embeds an encrypted watermark transmitted from the watermark certification authority (WCA) by combining it with the target encrypted data in the encryption domain, and finally transmits the multiple-watermarked and encrypted content to a buyer. The buyer obtains the multiple-watermarked content by decrypting it and plays the content in a mobile device. We implemented our scheme in a mobile device environment using the mobile 3D content tools of Khronos OpenGL ES 2.0 SDK [28], and evaluated its robustness and invisibility. Experimental results verify that our proposed scheme can protect the copyright and piracy violation of mobile 3D content properly because the peak signal-to-noise ratios (PSNRs) of multiple watermarks are above 36 dB and the watermarks are robust to vertex noising, data precision modification, simplification, rotation, scaling, and translation.

The rest of this paper is organized as follows. In section 2, we introduce the mobile 3D engine and BSW protocol. In section 3, we then explain the proposed multiple-watermarking scheme used for mobile 3D content based on the anonymous BSW protocol. We then describe the implementation and evaluation of our scheme in section 4. Finally, section 5 discusses the improvement of our scheme over previous schemes.

2. Related Work

In this section, we introduce the structure of mobile 3D content and the anonymous BSW protocol from [22] in view of a mobile 3D watermarking scheme.

2.1 Mobile 3D Content

Watermarking schemes for mobile 3D content can be presented according to the data structure of the content and the gaming scenario. A mobile 3D game engine is used to provide API rendering and animate 3D content in low-level devices such as mobile phones. Many of such engines have been developed over the past years [1]-[5].

The general structure of a mobile 3D engine for a game API is illustrated in **Fig. 1**. Most mobile devices use the arithmetic of fixed points instead of floating points for fewer arithmetic operations. The content designer loads the converted content, and designs and programs the game scenario using this content, porting it to the target device. There are three types of cases for embedding a watermark in a mobile 3D game engine, as shown in **Fig. 1**. In the first case, the watermark can be embedded into 1D, 2D, or 3D content. This case includes watermarking schemes for audio, images, video, and 3D models. However, many watermarks will be lost through a data conversion during mobile resource editing. In addition, it is difficult to extract a watermark in this case because most engines do not provide an inverse conversion from mobile data into the original content. In the second case, the watermark can be embedded into mobile data converted through mobile resource editing. The content designer can design various mobile games using watermarked mobile data. The watermark can be easily extracted from the data of a mobile game. In the third case, the watermark can be embedded during the process of data loading and programming. The content designer can design a watermarked mobile game program using mobile DRM. In this paper, based on the second and third cases, we introduce a watermarking scheme using mobile 3D data loaded by the game design.

Mobile 3D content is generally composed of hierarchical 3D scenes of each object. A *group* is a set of some scene nodes, and a *world* is a super group including all scenes. The world stores the information of the background and camera that is activated to the top group in a hierarchical structure of 3D objects. A *mesh* is the fundamental shape of an object that has a vertex buffer, index buffer, and shape appearance. A vertex buffer stores the information of the vertex coordinate, normal vector, and color, and an index buffer stores the index of vertices to represent a triangular strip. In addition, a shape appearance stores the animation track using a keyframe sequence and animation controller and material, polygon node and texture that can represent surface or skin information.

To embed the target, the watermarking system can select the vertex coordinate in the vertex buffer, the triangular strip arrangement in the index buffer, and the keytime/keyvalue in the keyframe sequence in the animation track and background image defined during 3D object rendering. Our scheme uses the vertex coordinate in the vertex buffer and the triangular strip in the index buffer because it is not easy for these to be changed by the external environment or through attacks.

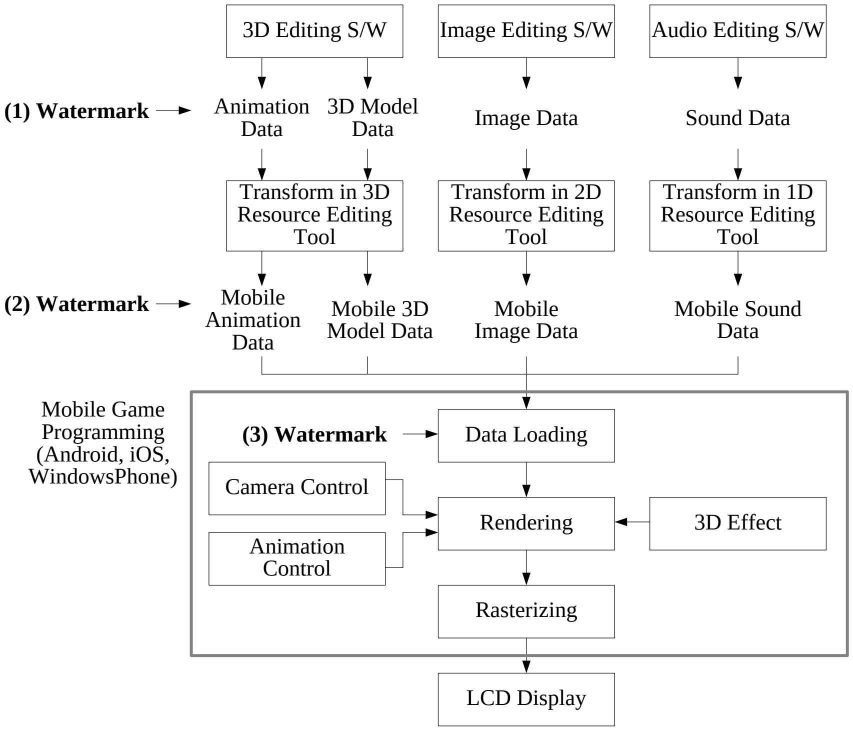


Fig. 1. General structure of mobile 3D engine and available embedding targets for watermark.

2.2 Anonymous Buyer-Seller Watermarking Protocol

For the copyright protection of mobile 3D content, we present multiple watermarkings based on the anonymous BSW protocol in [22] because it can solve both the customer rights and unbinding problems. While we can use other watermarking protocols that are potentially more secure than Lei's protocol, we are not focused on the design of a secure watermarking protocol, but rather on the use of a multiple-watermarking scheme based on a basic watermarking protocol. This protocol consists of a registration protocol, watermark-embedding protocol, and identification and arbitration protocols.

The registration protocol issues an anonymous certificate to the buyer. The buyer generates a random key pair (pk_B, sk_B) and transmits a public key pk_B to the certification authority (CA). The CA generates an anonymous certificate of pk_B and transmits it to the buyer. The CA also binds a certificate to the buyer information and guarantees that the binding information will not be opened except upon an arbiter (ARB) request.

The watermark embedding protocol is performed by the buyer, seller, and WCA, as illustrated in **Fig. 2**. The purchasing table, Tablex, stores the information produced by each contract. A unique watermark W^1 can be used as a key for searching the information in Tablex in conflicting violator.

The identification and arbitration protocols are performed by the seller, ARB, and WCA. When a conflict of suspect content Y occurs, the seller extracts W^1 in Y and selects a watermark W_k^1 with the maximum similarity after calculating the similarity between W^1 and all watermarks W_i^1 in Tablex. The seller then transmits both Y and the information connected to W_k^1, such as the certificate, signature, agreement, and encrypted watermark, to the ARB. The ARB verifies the certificate and signature and obtains a WCA watermark W^2 by requesting that the WCA decrypt $E_{pk^*}(W_k^2)$. The ARB then encrypts W^2 using pk^* and, verifies the similarity between $E_{pk^*}(W^2)$ and $E_{pk^*}(W_k^2)$, finally extracting W^2 in Y. Through this process, the ARB determines whether Y is pirated.

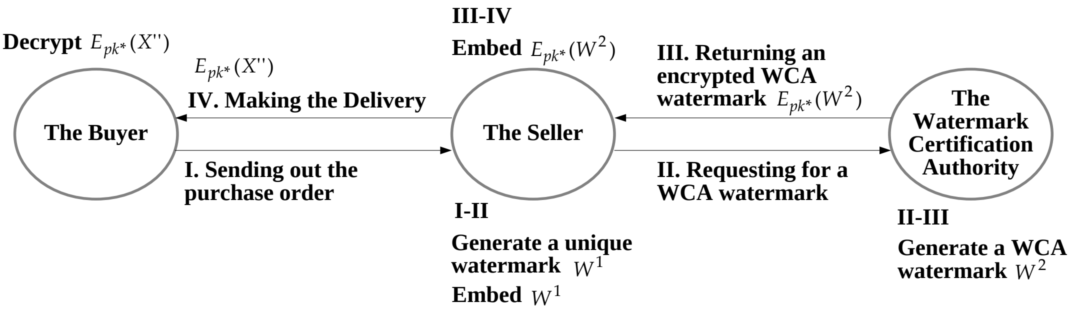

Decrypt $E_{pk^*}(X'')$
$E_{pk^*}(X'')$
IV. Making the Delivery
The Buyer
I. Sending out the purchase order
III-IV
Embed $E_{pk^*}(W^2)$
The Seller
I-II
Generate a unique watermark W^1
Embed W^1
III. Returning an encrypted WCA watermark $E_{pk^*}(W^2)$
II. Requesting for a WCA watermark
The Watermark Certification Authority
II-III
Generate a WCA watermark W^2

Fig. 2. The interrelation of the buyer, the seller and WCA in anonymous Buyer-Seller watermarking protocol [22].

2.3 Mesh Curvedness

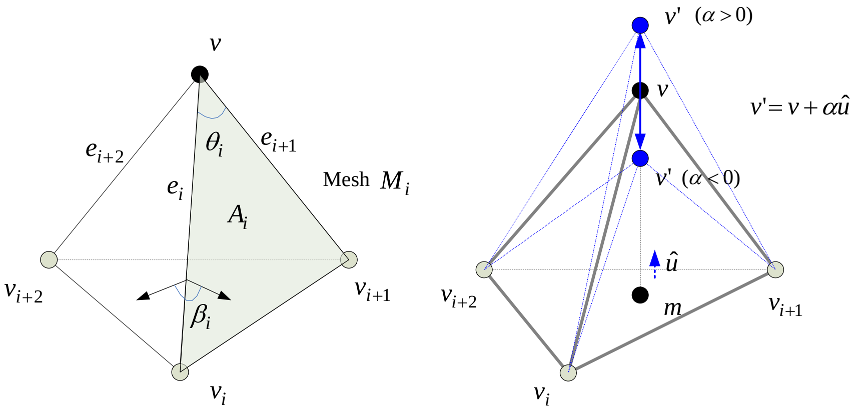
The geometric shape of any object can be defined by its curvedness, which is the measure of the intensity of the curvature shape. The curvedness has been widely used as shape features in 3D mesh segmentation [29] and 3D triangle optimization [30]. However, since it is very difficult for the curvedness to be changed into a specific value, it is not applied in 3D watermarking. The curvedness $C(\boldsymbol{v})$ of vertex $\boldsymbol{v}$ is defined by two principal curvatures, $k_{\max}(\boldsymbol{v})$ and $k_{\min}(\boldsymbol{v})$, of the surface at vertex $\boldsymbol{v}$ as follows.

$$C(\boldsymbol{v}) = \sqrt{\left(k_{\max}^2(\boldsymbol{v}) + k_{\min}^2(\boldsymbol{v})\right)/2} \tag{1}$$

where $k_{\max}(\boldsymbol{v})$ and $k_{\min}(\boldsymbol{v})$ are defined by $k_{\max}(\boldsymbol{v}) = H(\boldsymbol{v}) + \sqrt{H^2(\boldsymbol{v}) - K(\boldsymbol{v})}$, and $k_{\min}(\boldsymbol{v}) = H(\boldsymbol{v}) - \sqrt{H^2(\boldsymbol{v}) - K(\boldsymbol{v})}$ in which $H(\boldsymbol{v})$ and $K(\boldsymbol{v})$ are the mean curvature and Gaussian curvature, respectively, and can be calculated by

$$H(\boldsymbol{v}) = \frac{3}{4A}\sum e_i\beta_i, K(\boldsymbol{v}) = \frac{3}{A}(2\pi - \sum\theta_i) \tag{2}$$

when considering the mesh as a piecewise linear approximation of a smooth surface. Here, e_i is the Euclidean distance $\|\overrightarrow{\boldsymbol{v}\boldsymbol{v}_i}\|$ between vertex $\boldsymbol{v}$ and valence vertex $\boldsymbol{v}_i$, β_i is the dihedral angle between two neighboring valence meshes that include a line connected to $\boldsymbol{v}$ and $\boldsymbol{v}_i$, A is the sum of the areas of all valence meshes $M_i(i \in [1, N_M])$, and θ_i is the angle subtended in a valence mesh M_i at $\boldsymbol{v}$. **Fig. 3** illustrates the definition of the parameters used for computing two curvatures of vertex $\boldsymbol{v}$ in any tetrahedron.



(a) (b)

Fig. 3. (a) Parameters for computing the mean curvature and the Gaussian curvature of a vertex $\boldsymbol{v}$ and (b) the alteration of a vertex for movement of $C(\boldsymbol{v})$.

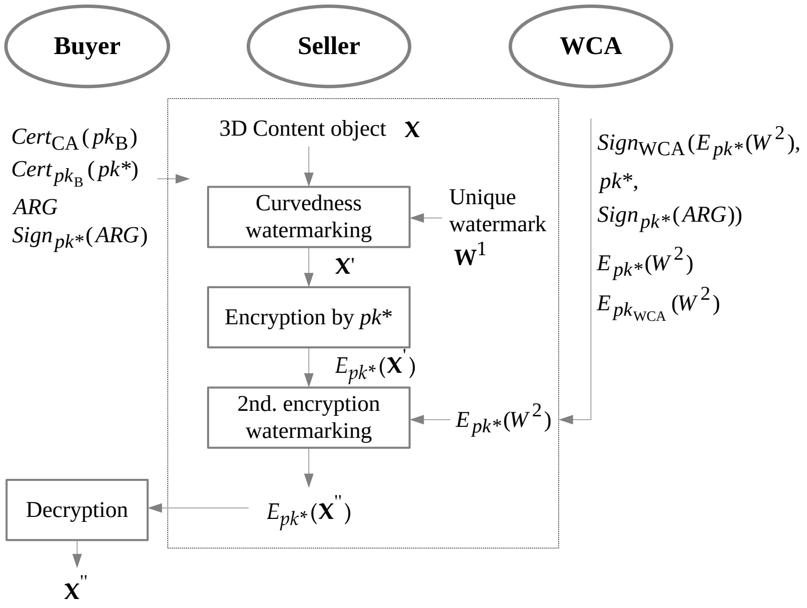

Buyer
Seller
WCA
$Cert_{\mathrm{CA}}(pk_{\mathrm{B}})$
$Cert_{pk_{\mathrm{B}}}(pk^*)$
ARG
$Sign_{pk^*}(ARG)$
3D Content object $\mathbf{X}$
Curvedness watermarking
Unique watermark $\mathbf{W}^1$
$\mathbf{X}'$
Encryption by pk^*
$E_{pk^*}(\mathbf{X}')$
2nd. encryption watermarking
$E_{pk^*}(W^2)$
$E_{pk^*}(\mathbf{X}'')$
Decryption
$\mathbf{X}''$
$Sign_{\mathrm{WCA}}(E_{pk^*}(W^2),$
$pk^*,$
$Sign_{pk^*}(ARG))$
$E_{pk^*}(W^2)$
$E_{pk_{\mathrm{WCA}}}(W^2)$

Fig. 4. The proposed multiple-watermarking process for mobile 3D content.

3. Proposed Mobile 3D Watermarking

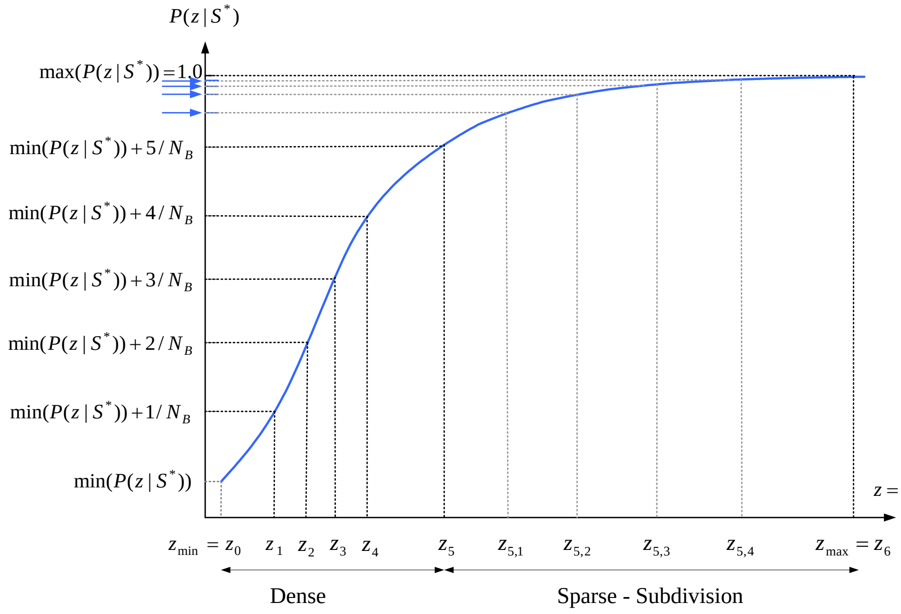
This paper presents multiple-watermarking schemes in the spatial and encryption domains of mobile 3D data for the copyright protection of mobile 3D content. As explained in section II, the embedding target is selected for vertices $\mathbf{V}$ in mesh object $\mathbf{X}$ of the mobile 3D content. In the proposed scheme, the seller generates a unique watermark W^1 of a random Gaussian bitstream, and obtains the first watermarked object $\mathbf{X}'$ by embedding W^1 into *the non-uniform quantization bins of curvedness* in $\mathbf{X}$. The seller then encrypts $\mathbf{X}'$, performs a linear combination $E_{pk^*}(W^2)$ using the encrypted normalized vertices of object $E_{pk^*}(\mathbf{X}')$, and obtains the final encrypted and watermarked object $E_{pk^*}(\mathbf{X}'')$. The buyer decrypts $E_{pk^*}(\mathbf{X}'')$ using a key sk^* and obtains the watermarked object, $\mathbf{X}''$. **Fig. 4** shows the multiple watermarking process for mobile 3D content.

3.2 Curvedness Watermarking

The *curvedness watermarking* scheme embeds a unique watermark W^1 with robustness into a geometry attack of scaling, rotation, and translation, and into a topology attack of connectivity. A unique watermark is used to protect the copyright of 3D objects. To consider the robustness, the watermark should be embedded into non-uniform quantized bins of curvedness. This is similar to quantization index modulation (QIM)-based watermarking in the DCT/DWT frequency domain for images and video. However, QIM-based watermarking cannot be applied to a 3D vector object. The curvedness has been widely used as the shape feature of a 3D object. A high curvedness in any vertex is very suitable for the embedding position because it corresponds to a high spatial frequency. Quantization-based watermarking for a 3D object has the following advantages: 1) noise robustness, i.e., the quantization step size can control the robustness, 2) shape preservation and adaptability, i.e., the watermark can be embedded adaptively to the curvedness of a 3D object while preserving the shape, 3) blind watermarking, i.e., the watermark can be detected without the host object, and 4) 3D compression flexibility, i.e., our scheme can be applied to the compression algorithm of a 3D object. This is an important factor of a mobile 3D object that requires a few vertices. A conventional scheme, which is based on the distribution of vertex distances, has noise robustness and blindness. However, it does not have shape preservation or adaptability, and lacks the flexibility of a compression algorithm.

3.1.1 Normalization and Reference curvedness

We normalize all vertices $\mathbf{V} = \{\boldsymbol{v}_k | k \in [1, N_V]\}$ in object $\mathbf{X}$, allowing the surface area $S(\mathbf{X})$ to be any target surface area S'. As the surface area is the sum of areas A_k in triangle meshes, all areas should be changed into $(S'/S(\mathbf{X}))A_k$ by uniformly rescaling the vertices that are connected to the meshes. Thus, all vertices $\boldsymbol{v}_k$ are normalized by the factor $\alpha = \sqrt{S'/S(\mathbf{X})}$. After performing the embedding process, the watermarked vertex $\mathbf{V}'$ can be obtained through inverse normalization.



(b)

Fig. 5. The reference curvedness according to cdf of curvedness.

The curvedness $C(\boldsymbol{v})$ will be moved to any target value with the reference curvedness according to the watermark bit. Therefore, the reference curvedness of the non-uniform quantization bin should be designed before the embedding process.

We design the non-uniform quantization bin using the cumulative distribution function (cdf) of the curvedness depending on the target surface area S'. From a number of experiments, we confirmed that the cdf of the curvedness is

$$P[C(\boldsymbol{v})] \approx 1 - e^{-C(\boldsymbol{v})/\alpha b}. \tag{3}$$

Let $C(\boldsymbol{v})$ be indicated by random variable z for simple notation, where z has a range of $[z_{\min}, z_{\max}] = [0, P^{-1}(1)]$. Let the set of reference curvedness values in non-uniform quantization bins be denoted as $\boldsymbol{R}$. We first divide cdf $P[z]$ into the initial N_B bins, and obtain random variables that are matched to each bin. Bin B_i can be defined by

$$B_i = [z_{i-1}, z_i), \text{ where } P(z_i) = P(z_{\min}) + \Delta \times i \text{ and } z_0 = z_{\min}, z_{N_B} = z_{\max}. \tag{4}$$

If the range of bins is larger than threshold ϵ, bin B_i will be sub-divided into two bins continuously until $z_i - z_{i-1} \leq \epsilon$. From the above process, we obtain a set of reference curvedness values $\boldsymbol{R}$ in non-uniform quantization bins $\boldsymbol{B}$.

$$\boldsymbol{R} = \{r_i = (z_i + z_{i-1})/2 \mid i \in [1, N_R], N_R \geq N_B\} \tag{5}$$

Fig. 5 shows an example of reference curvedness $\boldsymbol{R}$ and bins $\boldsymbol{B}$ of N_R=10, which are divided into initial $N_B = 6$ segments in $P[z]$. Since $P[z]$ is an inverse exponential function, the bin is dense at a low z_i, but sparse at a high z_i. Therefore, any bin can be sub-divided at a high z_i.

3.1.2 Bit Embedding

We find the local maxima of curvedness and embed watermark bits into them. The local maxima of curvedness $LM\big(C(\boldsymbol{v})\big)$ is defined as a curvedness that is both higher than the curvedness of the neighboring vertices and has distance $\|\overrightarrow{\boldsymbol{mv}}\|$ between vertex $\boldsymbol{v}$ and mesh center $\boldsymbol{m}$ higher than threshold Th. This indicates the shape feature of a 3D model. Th is 1.0, as determined experimentally. The number of watermark bits N_{W^1} is the same as the number of local maxima. If the curvedness $C(\boldsymbol{v})$ of $LM(C(\boldsymbol{v}))$ is included in bin B_i, $C(\boldsymbol{v})$ then moves into a range that is determined by both watermark bit w_k and the reference curvedness r_i of bin B_i.

$$C(\boldsymbol{v}') = \begin{cases} r_i < C(\boldsymbol{v}') < z_i, \ w_k = 1 \\ z_{i-1} < C(\boldsymbol{v}') < r_i, \ w_k = 0 \end{cases}, \text{ if } C(\boldsymbol{v}) \in B_i \tag{6}$$

Fig. 6 shows that $C(\boldsymbol{v})$ moves to a specific value on the basis of r_i according to the watermark bit.

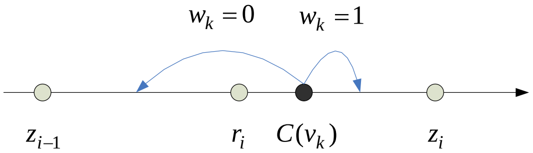


Fig. 6. Movement of the curvedness $C(\boldsymbol{v})$ that is included in a bin $B_i = [z_{i-1}, z_i)$.

It is actually very difficult to find vertex $\boldsymbol{v}' = C(C(\boldsymbol{v}'))^{-1}$ with the specific curvedness $C(\boldsymbol{v}')$ because of the various parameters for the mean and Gaussian curvatures. A generalized function for the inverse curvedness $C()^{-1}$ therefore cannot be modeled. We alter vertex $\boldsymbol{v}$ by controlling the moving intensity α such that the curvedness $C(\boldsymbol{v}')$ satisfies the embedding condition in the above equation.

$$\boldsymbol{v}' = \boldsymbol{v} + \alpha\hat{\boldsymbol{u}}, \hat{\boldsymbol{u}} = \overrightarrow{\boldsymbol{mv}}/\|\overrightarrow{\boldsymbol{mv}}\| \tag{7}$$

where $\hat{\boldsymbol{u}}$ is the unit vector between $\boldsymbol{v}$ and center $\boldsymbol{m}$ of the connected vertices, as shown in **Fig. 3(b)**. The available range $[\alpha_{\min}, \alpha_{\max}]$ of α can be determined based on the curved variation. To analyze the curved variation, we set α as

$$\alpha = \|\overrightarrow{\boldsymbol{mv}}\| \times scale, \tag{8}$$

and compute the curvedness as $scale$ varies from -1 to 1. From the experimental results, we verified that the curvedness is close to the second-order function on α and the available range $[\alpha_{\min}, \alpha_{\max}]$, as follows.

$$\alpha_{\min} = \begin{cases} \|\overrightarrow{\boldsymbol{mv}}\| \times scale_{\min}, \ \text{if } C(\boldsymbol{v} + (\|\overrightarrow{\boldsymbol{mv}}\| \times scale_{\min})\hat{\boldsymbol{u}}) > z_{i-1} \\ (-\boldsymbol{v} + C(z_{i-1})^{-1})/\hat{\boldsymbol{u}}, \ \text{if } C(\boldsymbol{v} + (\|\overrightarrow{\boldsymbol{mv}}\| \times scale_{\min})\hat{\boldsymbol{u}}) < z_{i-1} \end{cases} \tag{9}$$

$$\alpha_{\max} = (-\boldsymbol{v} + C(z_i)^{-1})/\hat{\boldsymbol{u}} \text{ for } C(\boldsymbol{v} + \alpha_{\max}\hat{\boldsymbol{u}}) = 1 \tag{10}$$

where $\frac{d(C(\boldsymbol{v}'))}{d(scale_{\min})} = 0$

We compute the moving intensity $\hat{\alpha}$ with the reference curvedness r_i within the available range, and then obtain the altered vertex $\boldsymbol{v}'$ according to watermark bit w_k.

$$\boldsymbol{v}' = \boldsymbol{v} + \alpha^* \hat{\boldsymbol{u}}, \text{ where } \alpha^* = \begin{cases} (\alpha_{\min} + \hat{\alpha})/2, & \text{if } w_k = 1 \\ (\alpha_{\max} + \hat{\alpha})/2, & \text{if } w_k = 0 \end{cases} \tag{11}$$

Through the above bit-embedding method, the seller can obtain the first watermarked object $\mathbf{X}' = \{\mathbf{V}'\}$ by embedding all bits of a unique watermark $\mathbf{W}^1$ in the selected bins $\mathbf{B}^*$. The computing method for inverse curvedness is fully explained in Appendix. The main process of curvedness watermarking is as follows.

1. Given vertices **V** in object **X**, normalize all vertices by the factor $\alpha = \sqrt{S'/S(\mathbf{X})}$.
2. Generate a unique watermark $\mathbf{W}^1$.
3. Compute curvedness $C(\boldsymbol{v})$ of the vertices, and obtain non-uniform quantization bins **B** and reference curvedness **R** in **B**.
4. Find the local maxima curvedness $LM\big(C(\boldsymbol{v})\big)$.
5. Compute the moving intensity α^* of the local maxima curvedness with **R** and $\mathbf{W}^1$, and then move its vertex by α^* in equation (11).
6. Generate a unique watermarked object $\mathbf{X}'$ through the above process.

3.3 Encryption Domain Watermarking

A WCA watermark W^2 is used to authenticate directly a 3D object in the encryption domain. W^2 is fragile to an attack. The encrypted watermark that WCA transmits should be embedded into the encrypted vector data while satisfying the privacy homomorphic property. This property can directly detect the watermark in the vector spatial domain without transforming a 3D object into the encryption domain. Complex 3D signal processing algorithms are very difficult to handle in the encryption domain. The encryption domain watermarking should therefore be integer based through the normalization of the 3D vector data. Our scheme has a higher PSNR and capacity than a conventional scheme in the encryption domain.

Given the encrypted WCA watermark $E_{pk^*}(\mathbf{W}^2) = \{E_{pk^*}(w_j^2)\}$, we embed $E_{pk^*}(\mathbf{W}^2)$ into encrypted vertiecs $E_{pk^*}(\mathbf{V}')$ of the first watermarked object $\mathbf{X}'$ by operator $\oplus$ with the bit transposing of addition and multiplication. Operator $\oplus$ satisfies both the homomorphic property in the encryption domain and the blinding property. Because of the homomorphic property, watermarking in the encryption domain has many limitations, making it fragile to geometric attacks. We convert the normalized vertex into the distance data for the rotation–scaling–translation (RST) robustness, and embed the encrypted watermark into these encrypted distance data.

All vertices $\mathbf{V}'$ can be separated into mean points $\mathbf{M}'$ and distances $\mathbf{D}'$ as follows.

$$\mathbf{V}' = \mathbf{M}' + \mathbf{D}' \text{ where } \mathbf{M}' = \{\left\|\overrightarrow{\boldsymbol{m}_k \boldsymbol{v}_k'}\right\| \,|k \in [1, N_V]\}, \mathbf{D}' = \{\boldsymbol{d}_k | k \in [1, N_V]\} \tag{12}$$

We collect vertices $\boldsymbol{v}'$ in object $\mathbf{X}'$, satisfying the following condition:

$$\boldsymbol{V}_W^2 = \{\boldsymbol{v}' | \, C(\boldsymbol{v}') \neq LM\big(C(\boldsymbol{v}')\big) \text{ and } \left\|\overrightarrow{\boldsymbol{mv}'}\right\| > Th\} = \boldsymbol{M}_W^2 + \boldsymbol{D}_W^2 \tag{13}$$

It reasons that local maxima curvedness is not included in the embedding target for the second watermark. The distance $\left\|\overrightarrow{\boldsymbol{mv}'}\right\|$ and Th are mentioned in section 3.1.2. When $|\mathbf{V}^2|$ is the cardinality of vertices, we set the number of encrypted watermarks N_{W^2} to $\lfloor |\mathbf{V}^2|/m \rfloor$, and embed watermark $\mathbf{W}^2$ m times repeatedly. The mean points and distances are represented as 32-bit fixed points, similar to a vertex coordinate.

We initialize the tth bits of the decimal points of the distances, and then generate the encrypted object $E_{pk^*}(\mathbf{X}') = E_{pk^*}(\mathbf{M}') + E_{pk^*}(\mathbf{D}')$ by encrypting a unique watermarked object $\mathbf{X}'$ using key pk^*. We embed $E_{pk^*}(\mathbf{W}^2)$ into the distances $E_{pk^*}(\boldsymbol{D}_W^2) \in E_{pk^*}(\mathbf{D}')$ using operator $\oplus$ m times repeatedly.

$$\begin{aligned} E_{pk^*}\big(\boldsymbol{d}_j''\big) \ni E_{pk^*}\big(\boldsymbol{D}_W^{2'}\big) &= E_{pk^*}(\boldsymbol{D}_W^2) \oplus E_{pk^*}(\mathbf{W}^2) \\ &= E_{pk^*}(\boldsymbol{D}_W^2) + E_{pk^*}(2^t) E_{pk^*}(\mathbf{W}^2) \end{aligned} \tag{14}$$

Since operator $\oplus$ has a homomorphic property in terms of addition and multiplication, the above equation can be rewritten as

$$E_{pk^*}\big(\boldsymbol{D}_W^{2'}\big) = E_{pk^*}(\boldsymbol{D}_W^2 + 2^t \mathbf{W}^2). \tag{15}$$

Looking at the spatial domain, we see that the watermarked vertex $\boldsymbol{V}_W^{2\prime}$ can be obtained by

$$\boldsymbol{V}_W^{2\prime} = \boldsymbol{M}_W^2 + \boldsymbol{D}_W^2 + 2^t \mathbf{W}^2. \tag{16}$$

Through the above method, the seller can obtain the final watermarked and encrypted object $E_{pk^*}(\mathbf{X}'')$ after embedding $E_{pk^*}(\mathbf{W}^2)$ in $E_{pk^*}(\mathbf{D})$, and transmit it to the buyer. The buyer can obtain the watermarked object $\mathbf{X}''$ after decrypting $E_{pk^*}(\mathbf{X}'')$ using a private key, sk^*. The main process of encryption domain watermarking is as follows.

1. Given vertices $\mathbf{V}'$ in a unique watermarked object $\mathbf{X}'$ and encrypted WCA watermark $E_{pk^*}(\mathbf{W}^2)$, find embedding target vertices $\boldsymbol{V}_W^2$ with a high distance in not local maxima curvedness.
2. Divide target vertices $\boldsymbol{V}_W^2$ into mean points $\boldsymbol{M}_W^2$ and distances $\boldsymbol{D}_W^2$, and initialize the tth bits of $\boldsymbol{D}_W^2$ decimal points.
3. Encrypt object $\mathbf{X}'$ using key pk^*, and embed $E_{pk^*}(\mathbf{W}^2)$ into the tth bits of $E_{pk^*}(\boldsymbol{D}_W^2)$ by operator $\oplus$.
4. Generate a final watermarked and encrypted object $E_{pk^*}(\mathbf{X}'')$ through the above process. Transmit $E_{pk^*}(\mathbf{X}'')$ to the buyer.

3.4 Watermark Extracting

When a pirated object $\mathbf{Y}$ occurs, the identification and arbitration protocols judge whether it is a pirated copy. In this case, the seller extracts a unique watermark $\mathbf{W}^1$ in $\mathbf{Y}$, and the ARB extracts $\mathbf{W}^2$ of the WCA in $\mathbf{Y}$. The extraction process is similar to the embedding process.

To extract a unique watermark $\mathbf{W}^1$, the proposed scheme normalizes the coordinates of vertices $\acute{\mathbf{V}}$ in object $\acute{\mathbf{Y}}$ and, obtains the local maxima curvedness $C(\acute{\boldsymbol{v}}_j)$, and then finds a non-uniform quantization bin B_i that includes $C(\acute{\boldsymbol{v}}_j)$, which is predefined. Watermark bit $\acute{w}_j^1$ can be extracted by comparing $C(\acute{\boldsymbol{v}}_j)$ with the reference curvedness r_i of B_i.

$$\acute{w}_j^1 = \begin{cases} 0, & C(\acute{\boldsymbol{v}}_j) < r_i \\ 1, & C(\acute{\boldsymbol{v}}_j) \geq r_i \end{cases} \tag{17}$$

Through this extracting method, all watermarks $\acute{\boldsymbol{W}}^1 = \{\acute{w}_j^1\}$ can be extracted from all of the local maximum curvedness.

The WCA watermark $\mathbf{W}^2$ can be extracted in the spatial domain based on the homomorphic property. We therefore obtain the embedded vertices $\acute{\boldsymbol{V}}_W^2 = \{\acute{\boldsymbol{v}}_k = \left\|\overrightarrow{\acute{\boldsymbol{m}}_k \acute{\boldsymbol{v}}_k'}\right\| + \acute{\boldsymbol{d}}_k\}$ and extract the bits of the second watermark

$$\acute{w}_k^2 = \left(\acute{\boldsymbol{d}}_k \& 2^t\right) \gg t - 1, \text{ for } k \in [1, mN_{W^2}] \tag{18}$$

where & is bitwise and >> is the right bit shift operator. After extracting $\acute{\boldsymbol{W}}^2$ m times, the WCA watermark $\acute{\boldsymbol{W}}^2$ can be obtained by $\acute{\boldsymbol{W}}^2 = \{\acute{w}_j^2\}$, $w_j^{2*} = \mathrm{INT}(\sum_{k=1}^{m}(\acute{w}_k^2/m) + 0.5)$.

4. Experimental Results and Analysis

This chapter describes the simulation of our proposed multiple-watermarking scheme based on the anonymous BSW protocol in a mobile 3D content API of Khronos OpenGL ES 2.0 SDK of PowerVR SGX [28], along with an evaluation of the robustness and invisibility. Most 3D model watermarking schemes [10]-[16] cannot be applied to mobile applications because they focus on a 3D model with high resolution. We compared our scheme with Kwon's scheme [27], which is suitable for mobile a 3D model with low resolution.

4.1 Mobile 3D Watermarking System

We simulated our mobile 3D multiple-watermarking scheme in the main message loop and render loop in PowerVR SGX, as illustrated in **Fig. 7**. From this figure, some functions defined in InitApp, InitInstance, Render, ReleaseView, QuitApp, and the Exit module will be called by the states of each module. The state of InitApp calls the InitApplication function, which is called in the main function at the initial time. It changes the parameters for the device settings before initializing the render API. The state of InitInstance calls the OsInitOS, OsDoInitAPI, and InitView functions. The OsInitOS function stores the instance handles and outputs the main window, the OsDoInitAPI function takes a window by generating the GL initialization, and the InitView function initializes the texture and vertex buffer when the rendering texture is changed. The Render state calls the RenderScene, ApiRenderComplete, and OsRenderComplete functions. The RenderScene function, which is the main rendering loop function, performs a view port, real rendering, and texture output, and is called at all frames. After rendering, the ApiRenderComplete and OsRenderComplete functions declare the completion of API and OS rendering. The ReleaseView state calls the ReleaseView function before completing the API rendering. This function releases the memory and resources taken from the InitView function. The QuitApp state calls the QuitApplication function before the program closes, which releases the memory and resources taken from the InitApplication function.

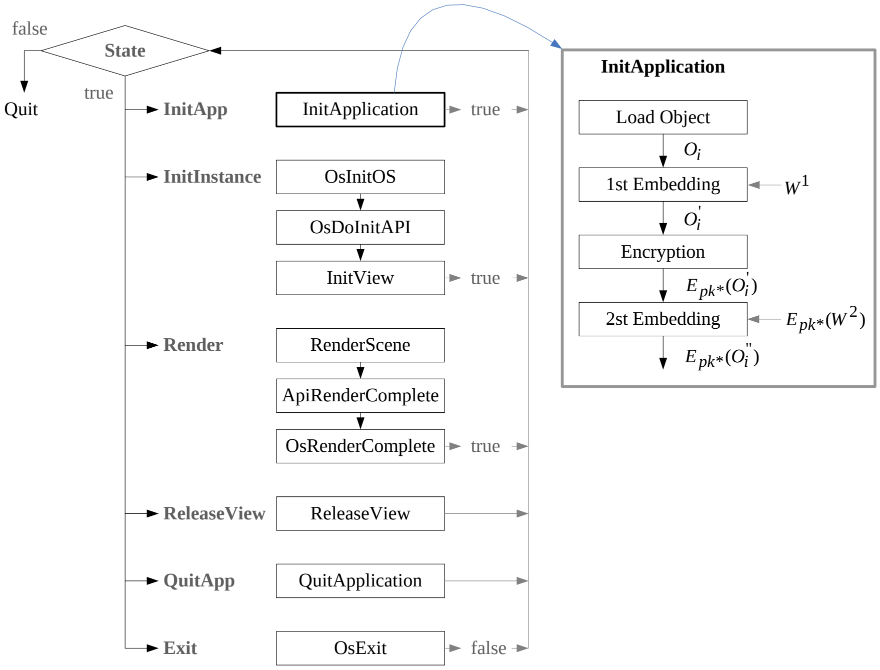


Fig. 7. The process of multiple-watermarking scheme in PowerVR SDK.

Table 1. Applications in each class

Model	Object	Polygon number	Vertex number	Unique watermark		WCA watermark	
				Bit number	**[%]**	**Bit number**	**[%]**
EvilSkull	Skull[1]	838	583	201	34.48	250	42.88
	Jaw[2]	346	175	38	21.71	50	28.57
	Total	*1,184*	*758*	*239*	*31.53*	350	46.17
Mouse	Head[3]	1,078	605	169	27.93	250	41.32
	Hand Left[4]	362	210	66	31.43	50	23.81
	Hand Right[5]	362	212	71	33.49	50	23.58
	Body[6]	332	189	51	26.98	50	26.46
	Object[7]	760	410	135	32.93	150	36.59
	Total	*2,894*	*1,626*	*492*	*30.26*	750	46.13
Polybump	Head Low[8]	276	153	48	31.37	50	32.68
Vase	Glass[9]	480	287	82	28.57	100	34.84
	Silver[10]	2352	1,335	361	27.04	550	41.20
	Total	*2,832*	*1,622*	*443*	*27.31*	750	46.24

In our experiment, we loaded a 3D object into the InitApplication function in a message loop and render loop of a mobile 3D content, and embedded multiple watermarks into each object, obtaining the encrypted content $E_{pk^*}(\mathbf{X}'')$, as illustrated in **Fig. 7**. Furthermore, we obtained the watermarked content $\mathbf{X}''$ after decrypting the encrypted content $E_{pk^*}(\mathbf{X}'')$ and displayed $\mathbf{X}''$ in the window in the RenderScene function.

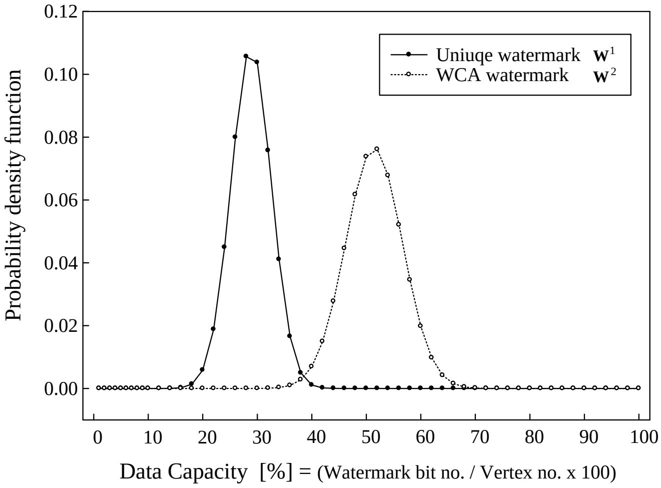
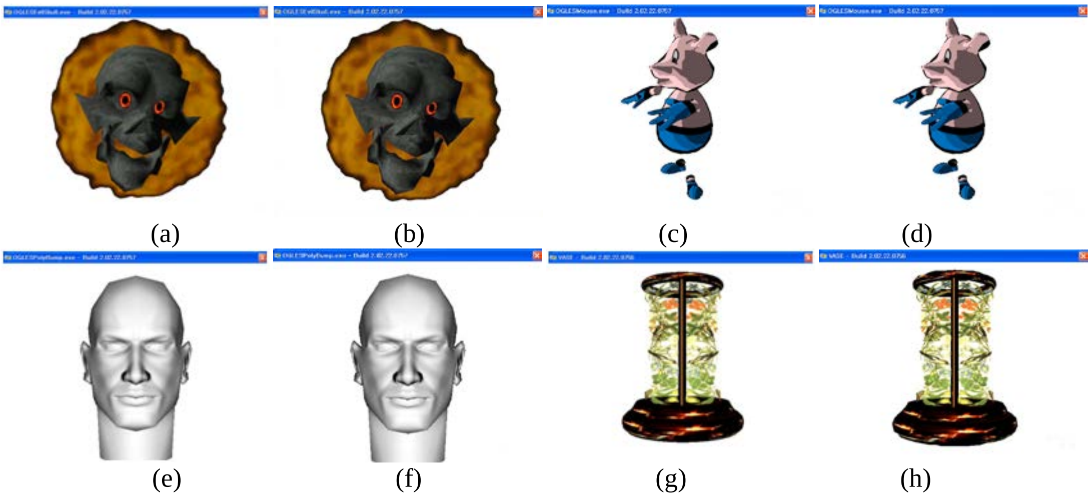


Fig. 8. Distribution of capacity of a unique watermark $\mathbf{W}^1$ and WCA watermark $\mathbf{W}^2$.


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

(e) (f) (g) (h)

Fig. 9. The first frame of mobile 3D model and watermarked 3D model; (a) EvilSkull and (b) watermarked EvilSkull, (c) Mouse and (d) watermarked Mouse, (e) Polybump and (f) watermarked Polybump, and (g) Vase and (h) watermarked Vase.

To evaluate the above system, we took mobile 3D data from EvilSkull, Mouse, PolybumpHead, and Vase models in PowerVR OpenGL ES SDK [28]. **Table 1** shows the objects in each of the models, and the polygon number, vertex number N_V, and bits of two watermarks, N_{W^1}, N_{W^2}, in each of the objects. We generated N_{W^1} and N_{W^2} bits with a random Gaussian distribution, and used them as a unique watermark $\mathbf{W}^1$ and WCA watermark $\mathbf{W}^2$.

We analyzed the capacity of the two watermarks, $\mathrm{Cap}(\mathbf{W}^1)$, $\mathrm{Cap}(\mathbf{W}^2)$, as

$$\mathrm{Cap}(\mathbf{W}^1) = \left|LM\big(C(\boldsymbol{v})\big)\right|/N_V = N_{W^1}/N_V \times 100[\%], \quad (19)$$
$$\mathrm{Cap}(\mathbf{W}^2) = N_{W^2}/N_V \times 100[\%] \quad (20)$$

which are the percentage of bits and number of vertices in an object. **Fig. 8** shows the distributions of two capacities, i.e., normal distributions of $N(28.88, 3.67^2)$ and $N(36.19, 8.53^2)$. **Fig. 9** shows both the original mobile 3D models and the watermarked models.

4.2 Invisibility Evaluation

Most 2D image watermarking schemes use the pixel-to-pixel PSNR for a quality evaluation. However, it is difficult to apply the PSNR to 3D model data for a 3D quality evaluation. In our experiment, we used the SNR of the vertex error distance to verify the invisibility. We therefore define SNR_i of vertex data $\mathbf{V} = \{\boldsymbol{v}_k | k \in [1, N]\}$ in object $\mathbf{X}$ by

$$\mathrm{SNR}_i = 10\log_{10}\left(\frac{R_{signal}^2}{R_{noise}^2}\right) = 20\log_{10}\left(\frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}|\boldsymbol{v}_k - \boldsymbol{M}|^2}}{\sqrt{\frac{1}{N}\sum_{k=1}^{N}|\boldsymbol{v}_k - \boldsymbol{v}_k'|^2}}\right). \quad (21)$$

where R_{signal} is the root mean square (RMS) of the distance difference $|\boldsymbol{v}_k - \boldsymbol{M}|$ between each of the vertices $\boldsymbol{v}_k$ and the mass point $\boldsymbol{M}$ of object $\mathbf{O}$, and R_{noise} is the RMS of the distance difference $|\boldsymbol{v}_k - \boldsymbol{v}_k'|$ between each of the vertices $\boldsymbol{v}_k$ and watermarked vertices $\boldsymbol{v}_k'$. **Fig. 10** shows the vertex SNR of each object, where the vertex SNRs of the two watermarked objects are about 40.50–73.75 dB in our scheme and 38.92–65.73 dB in Kwon's scheme. Our scheme has a higher vertex SNR than Kwon's scheme. The first frames of the mobile 3D models and watermarked models are shown in **Fig. 9**. These figures verify the invisibility of the watermark since there is little quality difference between the two models.

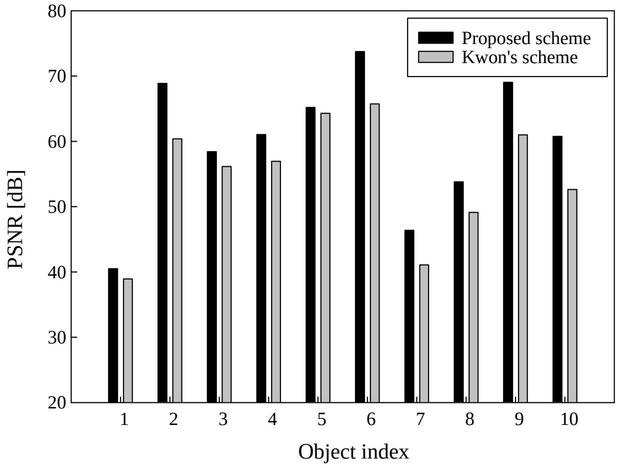


Fig. 10. Vertex PSNRs of proposed scheme and conventional scheme (Object index is the number of model in table 1).

4.3 Robustness Evaluation

We evaluated the robustness to certain attacks using the bit error rate (BER) of the extracted watermark, although the buyer cannot easily edit the mobile 3D content since he provides the executable code, but he can more easily duplicate it. The experimented attacks are an RST, additive noise, data precision change, cropping, simplification, and geometric modifier. **Fig.**

11 shows the average BERs extracted from the attacked objects. In most of the attacks, a unique watermark $\mathbf{W}^1$ is more robust than WCA watermark $\mathbf{W}^2$. It reasons that a WCA watermark must be embedded in the encryption domain by the operator with a limit of the homomorphic property.

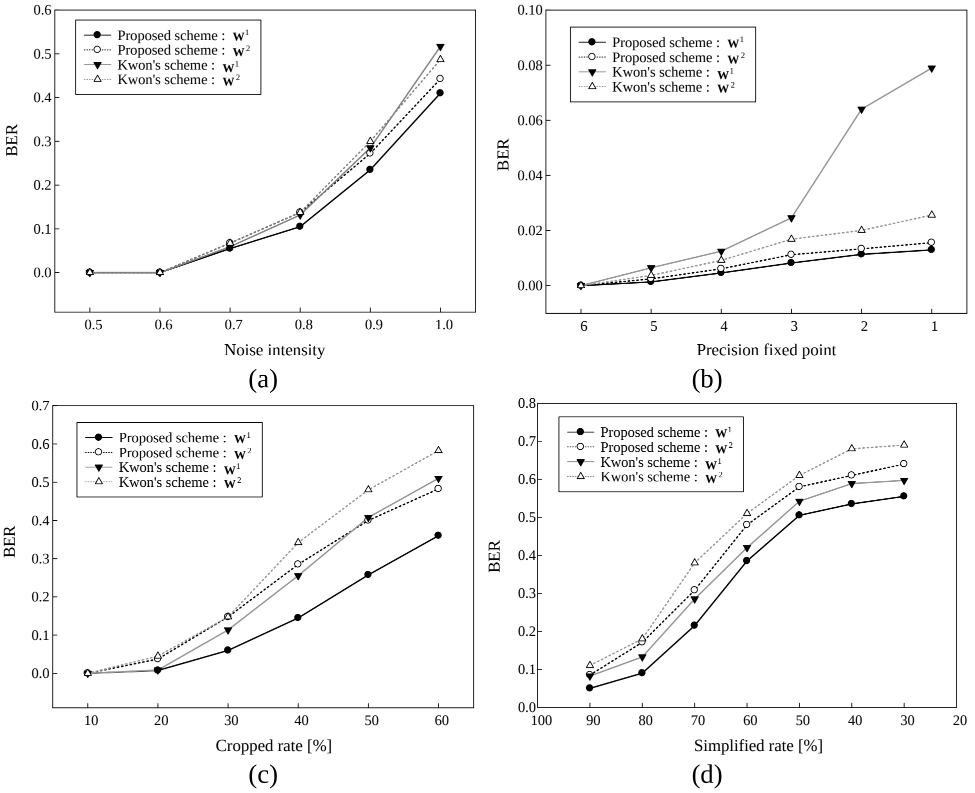


Fig. 11. BERs of the extracted watermark in (a) additive noising, (b) data precision changing, (c) data cropping, (d) simplification.

For the RST experiment, we rotated all objects into an arbitrary angle from [-100°, +100°], translated them into an arbitrary position in the position range [-50 50], and scaled them to an arbitrary factor in the scale range of [10 10]. Based on the experimental results, we verified that the two watermarks from our scheme are insensitive to RST attacks because of the curvedness for $\mathbf{W}^1$ and the distance for $\mathbf{W}^2$ in a normalized object, whereas $\mathbf{W}^1$ and $\mathbf{W}^2$ of Kwon's scheme are very sensitive to scaling and RST attacks, respectively.

For the additive noise experiment, we added uniform random noises to all vertices, $\boldsymbol{v}' = v \times (1 + \alpha \times uniform.\,())$, where α is the noise intensity of [0.5 1.0], and $uniform()$ is a uniform random function with [-0.5 0.5]. Fig. 11(a) shows the BERs of the extracted watermarks in objects that are noised at various intensities. All watermarks can be extracted without a bit error for $\alpha = 0.6$. For $\alpha = [0.7\ 1.0]$, the BERs of $\mathbf{W}^1$ and $\mathbf{W}^2$ in our scheme are about 0.004–0.107 and 0.000–0.044 less than those in Kwon's scheme, respectively.

In the change in data precision experiment, we reduced the decimal part in the fixed-point of the vertex data from 6 to 1. Fig. 11(b) shows the BERs of the extracted watermarks in objects with a changed precision. From this figure, we can see that the BERs of $\mathbf{W}^1$ in our scheme are about 0.005–0.066 less than those in Kwon's scheme, and the BERs of $\mathbf{W}^2$ in both schemes are similar. However, the BERs of two watermarks are low enough to be permitted.

For the data cropping experiment, we randomly cropped 10 to 60% of the vertices in the model object. **Fig. 11(c)** shows the BERs of the extracted watermarks in the cropped objects. The BERs of $\mathbf{W}^1$ in our scheme are about 0.001–0.170 less than the BERs of $\mathbf{W}^1$ in Kwon's scheme. In addition, the BERs of $\mathbf{W}^2$ in our scheme are about 0.001–0.100 less than the BERs of $\mathbf{W}^2$ in Kwon's scheme, whereas the BERs of $\mathbf{W}^2$s in both schemes are similar. In particular, more than 80% of $\mathbf{W}^1$ and $\mathbf{W}^2$ can be extracted even when 45% and 34% of the vertices are cropped. The proposed scheme can therefore extract most watermark bits with no error though cropping some vertices since the local maxima curvedness for $\mathbf{W}^1$ is widely distributed in the object, and $\mathbf{W}^2$ is embedded several times. However, if the number of embeddings increases, the embedding dataset will be increased and the invisibility will be poor.

A 3D model with a low resolution has a small number of vertices for simplification. If this model is simplified to any percentage, the shape of the model will approach the base shape. Nonetheless, we experimented with simplification to evaluate our scheme. The simplification changes the order of vertices and mesh properties, such as the area and angle, by reducing the number of vertices and meshes. Simplification therefore makes the watermarked vertices become lost or their indices altered. We store the curvedness of the watermarked vertices in order, and extract two watermarks after matching the stored curvedness with the computed curvedness from the simplified object.

For the simplification experiment, we simplified the objects for a 50 to 90% vertex percentage, based on the mesh optimization for multiple resolutions. The vertex percentage R [%] specifies the percentage of the vertex count N_R in a simplified object as the overall number N_V of vertices in the original model, $R = N_R/N_V \times 100[\%]$. When the watermarked vertices are lost, they are considered to error bits. **Fig. 11(d)** shows the BERs of watermarks extracted from simplified objects. When R is above 60%, the BERs approach 0.5, which indicates that the watermark is meaningless. Our scheme can extract more than 80% of $\mathbf{W}^1$ and greater than 70% of $\mathbf{W}^2$ up to R = 70%. However, Kwon's scheme can extract greater than 72% of $\mathbf{W}^1$ and more than 62% of $\mathbf{W}^2$ up to R = 70%.

From these results, we verified that our scheme is more robust to attacks than a conventional scheme.

5. Conclusion

We introduced a mobile multiple-watermarking scheme using the anonymous BSW protocol for the copyright protection of mobile 3D content. In the proposed scheme, the seller randomly selects one object from mobile 3D content and generates a unique watermark for copyrighting. The seller then determines the embedding target set using the local maxima curvedness in the selected object, and embeds a unique watermark into them using non-uniform quantization bins. After encrypting the watermarked content, the seller receives the encrypted watermark from the WCA and embeds this watermark using an operator in the encryption domain. This operator linearly combines the encrypted watermark with the encrypted vertex data while satisfying the homomorphic property. We implemented the proposed scheme using Khronos OpenGL ES 2.0 SDK of a mobile 3D content tool. The experimental results verify that the proposed scheme can protect the copyright of the seller, and is robust to certain attacks. Furthermore, we improved certain problems of a conventional scheme, i.e., the anonymity of the buyer, the unbinding, and the robustness of the watermark, by presenting an embedding scheme using the geometric property of the vertex data based on Lei's anonymous BSW protocol.

6. Appendix

6.1 Computation for inverse curvedness

Given curvedness $C(\boldsymbol{v}) = \sqrt{2H^2(\boldsymbol{v}) - K(\boldsymbol{v})} = z$ in equation (1), an inverse curvedness produces vertex $\boldsymbol{v} = C(z)^{-1}$ with curvedness z. However, it is very difficult to find $\boldsymbol{v}$ because of the parameters used for computing the mean and Gaussian curvatures, and is also difficult to model the inverse curvedness function. Thus, $C(z)^{-1}$ seems to be an unsolvable function. We therefore compute the inverse curvedness through an iteration process at discrete points.

Our scheme should find vertex $\boldsymbol{v}' = \boldsymbol{v} + \alpha\hat{\boldsymbol{u}}$ using the specific curvedness z by varying the moving intensity α from $\alpha_{\min}$ to $\alpha_{\max}$. Let the inverse curvedness of $C(\boldsymbol{v} + \alpha\hat{\boldsymbol{u}}) = z$ be $C(z)^{-1} = \boldsymbol{v} + \alpha\hat{\boldsymbol{u}} = \boldsymbol{v}'$, and $\alpha_{\min}$ and $\alpha_{\max}$ be denoted as A and B. We split the range of $[A, B]$ into N discrete points,

$$\alpha(k) = A + \frac{B-A}{N}k \text{ for } k \in [0, N] \tag{22}$$

and find $\alpha^*(k)$ in which the difference between z and $C(\boldsymbol{v} + \alpha(k)\hat{\boldsymbol{u}})$ is at minimum:

$$\alpha^*(k) = \arg\min_{k\in[0,N]} |z - C(\boldsymbol{v} + \alpha(k)\hat{\boldsymbol{u}})|, N = 100 \times \lfloor \|\overrightarrow{\boldsymbol{mv}}\| \rfloor \tag{23}$$

We re-split the range of $[\alpha^*(k) - \frac{B-A}{N}, \alpha^*(k) + \frac{B-A}{N}]$ into N discrete points and find $\alpha^*(k)$ from the above equation. This process is performed iteratively until $\alpha^*(k)$ converges to a value. Hence, $\boldsymbol{v} + \alpha^*(k)\hat{\boldsymbol{u}}$ is inverse curvedness $C(z)^{-1}$.

$$C(z)^{-1} \approx \boldsymbol{v} + \alpha^*(k)\hat{\boldsymbol{u}} = \boldsymbol{v}' \tag{24}$$

References

[1] SIO2 Engine, SIO2Interactive.com, http://sio2interactive.com/, Accessed to Sept. 2012.

[2] ShiVa3D, Stonetrip, http://www.stonetrip.com, Accessed to Sept. 2012.

[3] Unity3D, Unity Technologies, http://www.unity3d.com, Accessed to Sept. 2012.

[4] Corona DSK, Corona, http://www.coronasdk.com, Accessed to Sept. 2012.

[5] Unreal Engine 3 (UDK), Unreal Technology, http://www.udk.com/mobile, Accessed to Sept. 2012.

[6] Authentec, http://www.authentec.com/, Accessed to Sept. 2012.

[7] BuyDRM, http://www.buydrm.com/, Accessed to Sept. 2012.

[8] Discretix, http://www.discretix.com/, Accessed to Sept. 2012.

[9] Microsoft® PlayReady, https://www.microsoft.com/playready/, Accessed to Sept. 2012.

[10] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE Journal on Selected Areas in Communications*, vol. 16, issue 4, pp. 551-560, May 1998. Article (CrossRef Link)

[11] O. Benedens, "Geometry-based watermarking of 3D models," *IEEE Computer Graphics and Applications*, pp. 46-55, Jan./Feb. 1999. Article (CrossRef Link)

[12] S.-H. Lee and K.-R. Kwon, "A watermarking for 3D-mesh using the patch CEGIs," *Digital Signal Processing*, vol. 17, issue 2, pp. 396-413, March 2007. Article (CrossRef Link)

[13] S.-H. Lee and K.-R. Kwon, "Mesh watermarking based projection onto two convex sets," *Multimedia Systems*, vol. 13, no. 5-6, pp. 323-330, Feb. 2008. Article (CrossRef Link)

[14] J.M. Konstantinides, A. Mademlis, P. Daras, P.A. Mitkas, M.G. Strintzis, “Blind robust 3-D mesh watermarking based on oblate spheroidal harmonics,” *IEEE Transactions on Multimedia*, vol. 11, issue 1, pp. 23-38, Jan. 2009. Article (CrossRef Link)

[15] K. Kwangtaek, M. Barni, and H.Z. Tan, “Roughness-adaptive 3-D watermarking based on masking effect of surface roughness,” *IEEE Transactions on Information Forensics and Security*, vol. 5, issue 4, pp. 721-733, Dec. 2010. Article (CrossRef Link)

[16] L. Ming and A.G. Bors, “Surface-preserving robust watermarking of 3-D shapes,” *IEEE Transactions on Image Processing*, vol. 20, issue 10, pp. 2813-2826, Oct. 2011. Article (CrossRef Link)

[17] S.-H. Lee and K.-R. Kwon, “CAD drawing watermarking scheme,” *Digital Signal Processing*, vol. 20, issue 5, pp. 1379-1399, Sep. 2010. Article (CrossRef Link)

[18] Z. Sakr and N.D. Georganas, “Robust content-based MPEG-4 XMT scene structure authentication and multimedia content location,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 3, article 18, Aug. 2007. Article (CrossRef Link)

[19] S.-H. Lee and K.-R. Kwon, “VRML animated model watermarking scheme using geometry and interpolator nodes,” *Computer-Aided Design*, vol. 43, issue 8, pp. 1056-1073, Aug. 2011. Article (CrossRef Link)

[20] S.-H. Lee and K.-R. Kwon, “Robust 3D mesh model hashing based on feature object,” *Digital Signal Processing*, vol. 22, issue 5, pp. 744-759, Sep. 2012. Article (CrossRef Link)

[21] N. Memon and P.W. Wong, “A buyer-seller watermarking protocol,” *IEEE Transactions on Image Processing*, vol.10, no.4, pp. 643-649, April 2001. Article (CrossRef Link)

[22] C.L. Lei, P.L. Yu, P.L. Tasi, and M.W. Chan, “An efficient and anonymous buyer-seller watermarking protocol,” *IEEE Transactions on Image Processing*, vol.13, no.12, pp. 1618-1626, Dec. 2004. Article (CrossRef Link)

[23] S. Katzenbeisser, A. Lemma, M.U. Celik, M. Van der Veen, and M. Maas, “A buyer–seller watermarking protocol based on secure embedding,” *IEEE Transactions on Information Forensics and Security*, vol. 3 , issue 4, pp. 783-786, Dec. 2008. Article (CrossRef Link)

[24] A. Rial, Mina Deng, T. Bianchi, A. Piva, and B. Preneel, “A provably secure anonymous buyer–seller watermarking protocol,” *IEEE Transactions on Information Forensics and Security*, vol. 5, issue 4, pp. 920-931, Dec. 2008. Article (CrossRef Link)

[25] A. Rial, J. Balasch, and B. Preneel, “A privacy-preserving buyer–seller watermarking protocol based on priced oblivious transfer,” *IEEE Transactions on Information Forensics and Security*, vol. 6, issue 1, pp. 202-212, March 2011. Article (CrossRef Link)

[26] L. Quan, C. Zheng, and Z. Zude, “Research on secure buyer-seller watermarking protocol,” *Journal of Systems Engineering and Electronics*, vol. 19, issue 2, pp. 370-376, April 2008. Article (CrossRef Link)

[27] S.-G. Kwon, S.-H. Lee, K.-R. Kwon, E.-J. Lee, S.-Y. Ok and S.-H. Bae, “Mobile 3D game contents watermarking based on buyer-seller watermarking protocol,” *IEICE Transaction on Information and System*, vol. E91-D, no.7, pp. 2018-2026, July 2008. Article (CrossRef Link)

[28] PowerVR SGX, Khronos OpenGL ES 2.0 SDK, Imagination Technology, http://www.imgtec.com/, Accessed to Sept. 2012.

[29] A. Jagannathan and E. L. Miller, “Three-dimensional surface mesh segmentation using curvedness-based region growing approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2195-2204, Dec. 2007. Article (CrossRef Link)

[30] N. Dyn, K. Hormann, S.J. Kim and D. Levin, “Optimizing 3D triangulations using discrete curvature analysis,” *Math. Methods for Curves and Surfaces: Oslo 2000*, Vanderbilt University Press, pp. 135-146, 2001.
http://www.researchgate.net/publication/228909965_Optimizing_3D_triangulations_using_discrete_curvature_analysis/file/9fcfd50aa804db9277.pdf&sa=X&scisig=AAGBfm1rmavfjLLjhh1CiWCN8spsifJABA&oi=scholarr&ei=83niUqjGG4rQkwXHuoFI&ved=0CCYQgAMoADAA

Suk-Hwan Lee received a B.S., a M.S., and a Ph. D. degree in Electrical Engineering from Kyungpook National University, Korea in 1999, 2001, and 2004 respectively. He worked at Electronics and Telecommunications Research Institute in 2005. He is currently an associate professor in Department of Information Security at Tongmyong University, which he started in 2005. He works as an editor of Korea multimedia society journal and is a member of IEEE, IEEK, IEICE and also is an officer of IEEE R10 Changwon section. His research interests include multimedia signal processing, multimedia security, digital signal processing, bio security, and computer graphics.

Seong-Geun Kwon received the B.S., and M.S., and Ph.D degrees in Electronics Engineering in Kyungpook National University, Korea in 1996, 1998, and 2002, respectively. He worked at Mobile Communication Division of Samsung Electronics from 2002 to 2011 as a Senior Engineer and Project Manager. He is currently a assistance professor in department of Electronics Engineering at Kyungil University. His research interests include multimedia security, mobile VOD, mobile broadcasting (T-DMB, S-DMB, FLO, DVB-H), and watermarking.

Ki-Ryong Kwon received the B.S., M.S., and Ph.D. degrees in Electronics Engineering from Kyungpook National University in 1986, 1990, and 1994 respectively. He worked at Hyundai Motor Company from 1986-1988 and at Pusan University of Foreign Language form 1996-2006. He is currently a professor in Department of IT Convergence and Application Engineering at the Pukyong National University. He visited University of Minnesota at 2000-2001 and Colorado State University at 2011-2012 with visiting professor in USA. He is currently the General Affair and Organizing Committee/Vice President in Journal of Korea Multimedia Society. also is a director of IEEE R10 Changwon section. His current research interests are in the area of digital image processing, multimedia security and digital watermarking, wavelet theory.