

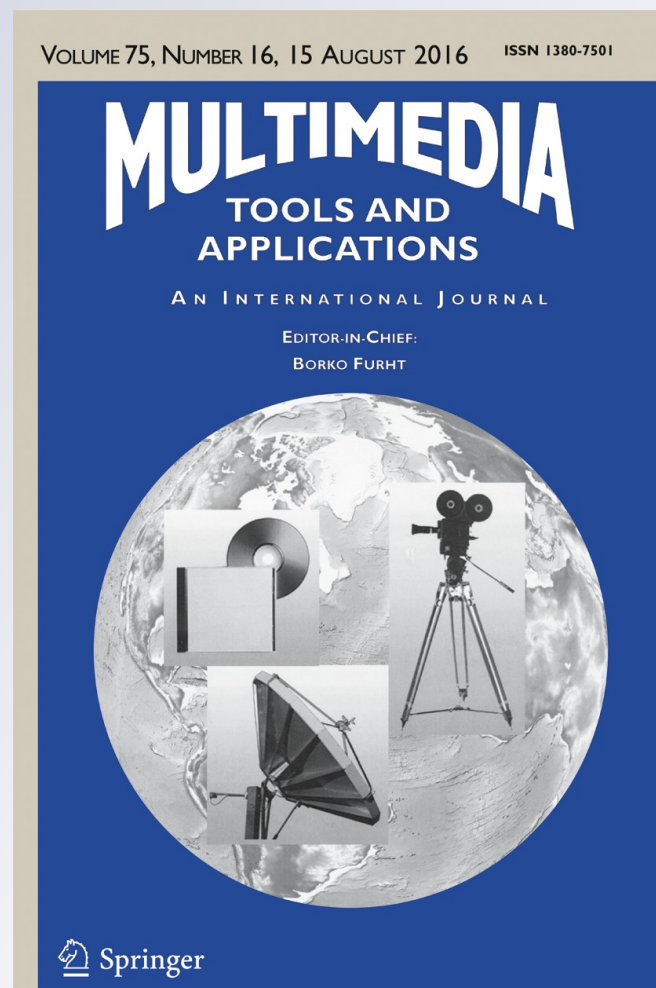
GPU-accelerated blind and robust 3D mesh watermarking by geometry image

Hung-Kuang Chen & Wei-Sung Chen

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501
Volume 75
Number 16

Multimed Tools Appl (2016)
75:10077-10096
DOI 10.1007/s11042-015-3062-y



Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

GPU-accelerated blind and robust 3D mesh watermarking by geometry image

Hung-Kuang Chen¹  · Wei-Sung Chen²

Received: 26 June 2015 / Revised: 18 September 2015 / Accepted: 3 November 2015 /

Published online: 14 November 2015

© Springer Science+Business Media New York 2015

Abstract The prevalence of cheap and powerful consumer level graphics accelerated hardware introduces a significant growth of 3D applications. In this paper, we have proposed a GPU-accelerated blind and robust watermarking approach to the 3D polygon meshes on the basis of the geometry image transform and image watermarking, which performs watermark embedding and detection on the basis of the geometry image derived from a spherical parametrisation of the input mesh with the help of massive-parallel processing power of the GPUs on the display card. The experimental results show that our approach is successful in at least two aspects. First, the watermark is robust, the embedded watermark survives from common geometric attacks, cropping, simplification, and re-meshing attacks. Second, with the help of parallel computations on the GPUs, the embedding and detection process is extremely fast.

Keywords Watermarking · 3D mesh · GPU · Geometry image

1 Introduction

Today, the development of information technologies and the prevalence of the internet have made it possible to publish, distribute, and purchase a variety of intellectual works such as

✉ Hung-Kuang Chen
hankchentw@gmail.com

Wei-Sung Chen
wschen213@gmail.com

¹ Electronic Engineering Department, National Chin-Yi University of Technology, Taichung, 41170, Taiwan

² Computer Science and Engineering Department, National Chung Hsing University, Taichung, 402, Taiwan

the books, music, photographs, movies, etc., in digital form. However, digital contents are also very likely to be either counterfeited or duplicated maliciously without any permission. These behaviours happened very frequently today, which greatly reduced the income of the author and the publisher and violate the laws of copyrights protection; consequently, discouraged the will of creation and publication and lead to the declination of our culture development. Hence, how to offer an appropriate protection mechanism over the intellectual properties, copyright ownership, and authentication of important documents has become a very important issue today.

To cope with this, the concept of digital watermarking was proposed [45] to provide protection of digital assets by *embedding* or *annotating* a specific message called the *watermark* into the protected content, called the *cover medium*, such as the audio/video stream, 2D image, 3D mesh, etc., through a dedicated procedure without interfering the use of this content. If the watermark is visible to the user, it is called a *perceptible* or *visible watermark*; otherwise, it is called an *imperceptible watermark*.

In occasions such as a law suit process, genuineness authentication, or communication verification, etc., a special detection process is usually applied to *extract* the imperceptible watermark to either identify the ownership or authorship of the host content or to indicate the modifications applied to the original content.

If the detection procedure can be performed without the need of the original host content, it is called a *blind detection* procedure; otherwise, it is called an *informed detection* procedure. During the transmission or distribution of the digital content, attacks such as noises, crop, or modifications might be applied. The extent to which the watermark can be extracted depends on the robustness of the watermarking mechanism.

A *robust watermarking* method must guarantee the success of watermark detection and identification after most of possible attacks; thus is suitable for copyright protection. On the other hand, a *fragile watermarking* must respond to any slightest modification to the content; hence is often used for authentication. From the intensive studies over the past two decades, versatile techniques of digital watermarking are developed for the protection of the contents of the literatures, music, 2D images, and videos [11, 22]

With the popularity of 3D graphics hardware and 3D TV, vast amount of 3D applications enlarged the demand of 3D models. Typical examples include the virtual reality systems, scientific or medical visualizations, ancient heritage preservations, geographic information systems, computer games, and 3D animations, etc.[2, 16, 25, 43]. However, digital watermarking of 3D contents introduces greater challenges, which makes the 3D digital watermarking methods relatively few and mostly irrelevant to current image watermarking methods [32].

In this paper, we present a novel blind and robust digital watermarking method for the 3D meshes using geometry image technique and spherical parametrization. The experimental results showed that the watermark is robust enough to survive from the attacks such as common geometric transforms, crop, re-sampling and simplification.

2 Related works

The increasing importance of 3D watermarking techniques has drawn much more attention to this field in these years. A great number of works have been lately proposed. Instead of giving a comprehensive survey as those done in [36, 49], we confined the review previous works to 3D mesh watermarking methods related to ours in this section. Suggested by most of previous literature, a major way to classify the 3D mesh watermarking techniques is to

categorize the techniques according to the domain in where the watermark embedding or detection is done. In accordance with this classification, current 3D mesh watermarking methods can be roughly classified either as a spatial or as a frequency domain technique.

2.1 Spatial domain techniques

A spatial domain technique embeds the watermark mainly into the geometric property of the host mesh by perturbing the geometric/topological structure of the mesh such as the vertex coordinates, vertex normal, edge lengths, triangle area, edge ratios, tetrahedral volume, etc. Methods of this kind are easy to implement but their watermarks are often fragile relative to the frequency domain techniques. To increase the robustness of the watermark, a more complicated and indirect embedding primitive and extra information other than the host message is usually required.

The earliest work on 3D mesh watermarking is proposed by Ohbuchi et al., which suggest using two geometric structure of the host mesh: the TSQ (Triangle Similarity Quadrature) and TVR (Tetrahedral Volume Ratio) [32]. Both techniques are blind. The TSQ technique is invariant to translation, rotation and uniform scaling and its watermark is resistant to resection and local deformation but fail to survive from coordinate randomization, re-meshing and other disturbances. The TVR technique is invariant to affine transformations but fails in projective transform, coordinate randomization, re-meshing, and other disturbances.

Benedens et al. proposed a more robust but private watermarking technique. They suggested the embedding the watermark by modifying the surface normals and their distributions. According to their report, their method is able to tolerate simplification attacks less than 36 percentages [4, 5].

Wagner et al. suggested embedding the watermark in the curvature normal of the target object [48]. Their method is affine invariant but is also vulnerable to the topology modifications.

Zafeiriou et al. suggested watermark embedding to the r-coordinate of a set of vertices within a certain range of angles in pseudo-random order [52]. The watermark is robust against both geometric transforms and mesh simplifications. However, it requires an extra alignment process to center and rotate the mesh to its initial orientation.

Cotting et al. suggested applying watermark embedding directly to the point clouds [10]. Following the same idea, Agarwal et al. devised another better technique that is robust against uniform affine transformations (rotation, scaling, and transformation), reordering, cropping, simplification, noise addition, remeshing, and progressive compression attacks [3].

Su et al. proposed three robust blind watermarking methods: namely, the OTC-W, OTP-W and Zero-W on the basis of Octree partitioning. They suggest applying PCA and Octree partition to the input mesh prior to watermarking. Their methods have the advantages of high embedding capacity and robustness against common geometric transformation and reordering attacks [8].

Wang et al. proposed another volume-based approach to 3D mesh watermarking, which suggested using an intrinsic 3D shape descriptor by jointly consider the analytic and continuous geometric volume moment as the watermarking primitive [50].

Luo and Bors [28] propose a new statistical approach for watermarking mesh representations of 3-D graphical objects, which employs a distance metric called the fast marching method (FMM). In their work, two embedding approaches varied by changing the mean and the variance of geodesic distance distributions are proposed. To ensure a minimal surface

distortion, the vertex displacements are performed according to the FMM during watermark embedding.

Singh et al. suggest embedding the watermark bits by altering the length of vertex normal. To increase the robustness, the pattern of watermark bits are repeatedly embedded into the cover object. Their algorithm is invariant to translation, rotation, uniform scaling and reordering point information of watermarked object [42].

The vertices of 3D mesh surface are categorized into flat, peak and deeper region. We first compare the perceivable distortion due to watermark insertion in the vertices of deep surfaces and the vertices belonging to either flat or peak surfaces.

According to the study of Garg et al., insertion of watermark in deeper surface have less distortion in comparison to watermark insertion in flat and peak surfaces. In view of the fact, their method begin with classifying the vertices into three mutually exclusive groups corresponding to the vertices of concave, flat, and protrusive areas. The watermark bits are embedded by relocating the selected vertices in accordance with their category [13].

To ensure optimal preservation of mesh surfaces, Bors et al. proposed another statistical approach to 3D watermarking on the basis of a new 3D surface preservation function metric considering the distance of a vertex displaced by watermarking and the Levenberg Marquardt optimization method during watermark embedding. Their method is statistical, blind, robust, and ensures minimal surface distortion [6].

Based on similar idea, Zhan et al. proposed another robust blind watermarking algorithm for 3D meshes considering vertex curvature so that both the robustness and visual masking are improved [53].

To enhance the radial distance-based framework to 3D mesh watermarking, Rolland et al. introduces the spread transform and perceptual shaping on the basis of roughness information and adopt the formulation of quadratic programming problem to minimizes the geometric distortion [37, 38]. In a later work, Rolland et al. study two conventional security mechanisms to highlight the limitations of this family of 3D watermarking systems with respect to security [39].

2.2 Frequency domain techniques

Prior to the embedment or detection of the watermark, a frequency domain technique usually requires a preprocessing stage to transform the host content into frequency domain signals. The frequency domain techniques usually have better robustness at the cost of additional transformation or analysis preprocessing overhead.

Kanai et al. applied wavelet-transform to the target object and embedded the watermark in the low-frequency part of the object [19].

Praun et al. suggest conversion of target object into multi-resolution format and embedding the watermark in the low frequency core [35]. Their approach claims to have the ability to against the simplification attacks.

Ohbuchi et al. used the eigenvectors of the matrix as the basis of the transformed domain and embedding the watermark in the low-frequency part of the target object [33].

Uccheddu et al. present another wavelet-based watermarking algorithm for 3D meshes. However, the host meshes are assumed semi-regular to permit wavelet decomposition. The method is robust against geometric transformations achieved by embedding the watermark in a normalized version of the host mesh obtained by PCA [46].

Abdallah et al. suggest using spectral conversion [1]. Since direct Laplacian spectral analysis requires a vast of calculations, they suggest partitioning the cover mesh into sub-meshes and apply the spectral conversion individually to each sub-mesh. Their approach is robust against the geometric transformations, adaptive random noise, mesh smoothing, mesh cropping, and combinations of these attacks.

Yang Liu et al. proposed a blind spectral two-way watermarking framework based on Dirichlet Manifold Harmonic Transform for parametrized surfaces [27]. It embeds watermarks into small surface patches without introducing discontinuity across the patch boundary and is robust against the majority of attacks.

2.3 Hardware accelerated techniques

Hardware assisted digital watermarking has been addressed of for years aiming for real-time performance and possible integrations with modern consumer products [14, 17, 20, 21, 24, 29, 30, 34, 40]. With the advent of powerful and low-cost GPUs, a number of approaches to image or video watermarking were also reported in recent years [7, 9, 26, 31, 47].

However, owing to the complexity of the representation, the hardware acceleration of watermarking approach to 3D meshes reveals more challenges and is relatively less studied [23, 41]. Among these works, Koller et al. proposed several approaches to the protection of 3D content in remote interactive rendering applications [23]. One of them is to provide protection of the 3D contents by encrypting the 3D model data with public-key encryption at creation time then performs on-chip decryption and rendering using GPUs. Shi et al. [41] proposed a set of OpenGL APIs for digital rights protection over commercial graphics data by integrating watermarking mechanism into the GPU.

2.4 Image-based techniques

Song and Cho proposed a robust watermarking approach to 3D meshes. They suggested acquiring range images from the cover mesh, i.e., the 3D model, by means of a set of virtual range scanners then applies any existing image watermarking method thereafter to the converted range image [44].

In a similar work [51], Ni et al. proposed another blind and robust method for 3D mesh watermarking, which applies Floater's parametrization [12] to convert the cover mesh into a geometry image [15] in the first place. The geometry image is partitioned into several sub-images of equal size. Normalization and 2D Haar-basis wavelet transform were applied to each sub-image prior to the insertion of the watermark bits to the low components of the wavelet coefficients using common LSB method. Their method is robust against common geometric attacks but lacks of evidences with regard to attacks such as cutting, re-meshing, simplification, etc.

3 Preliminaries

The input of our algorithm is assumed to be a manifold triangular mesh $M(V, T)$ defined in a 3D Cartesian coordinates. In which, V is a set of points defined in a Cartesian space \mathbf{R}^3 , or $V = \{v_i | v_i = (x_i, y_i, z_i) \in \mathbf{R}^3, i = 1, 2, \dots, n\}$, T is a set of triangles defined on V , or

$T = \{t_i | t_i = (v_a^i, v_b^i, v_c^i), v_a^i, v_b^i, v_c^i \in V, i = 1, 2, \dots, m\}$, and the cardinality of V is n , or $|V| = n$, and T is m , or $|T| = m$.

3.1 Topological notations

According to topology convention [13], an n -simplex is a topological entity consists of $n + 1$ vertices. Consequently, a 0-simplex is a vertex, a 1-simplex is an edge, and a 2-simplex is a triangle, and so forth. For an n -simplex s , the $(n - 1)$ -simplices in s are called the faces of s ; likewise, the edges of a triangle t are the faces of t , and the endpoints of an edge e are the faces of e . Let v be a vertex, a 0-simplex, and V be a set of vertices, 1-simplices, a number of operators are defined as follows.

- $[v_i]$: the set of adjoin edges connected to v , where $[v_i] = \{e_{ij} | e = v_i \bar{v}_j \in \mathbf{E}\}$ is called the star of v_i , denoted as $\mathbf{S}(v_i)$.
- $[[v_i]]$: the set of faces $f \in \mathbf{F}$ adjacent to v denoted as $\mathbf{R}(v_i)$.
- $[[v_i]] - v_i$ or ∂v : the boundary vertices of $\mathbf{S}(v_i)$ or commonly called the crown of v_i denoted as $\mathbf{C}(v_i)$.

3.2 Mesh parameterization

At a given point $x(u, v)$ in a triangle $T((u_i, v_i), (u_j, v_j), (u_k, v_k))$ defined in the parameter space Ω , the parametrisation of the point x in terms of its corresponding points p_i, p_j, p_k in the three dimensional Cartesian space \mathbf{R}^3 can be given by

$$\mathbf{x}(u, v) = \alpha \mathbf{p}_i + \beta \mathbf{p}_j + \gamma \mathbf{p}_k \quad (1)$$

where the triplet (α, β, γ) denotes the barycentric coordinates at the point $x(u, v)$ in the triangle [12].

3.3 Digital geometry image

In this context, we define a two dimensional texture image \mathbf{T} in a parameter space Ω as follows

$$\mathbf{T} = \{(R(u, v), G(u, v), B(u, v))\}, \quad (2)$$

where $R(u, v), G(u, v), B(u, v) : \Omega \rightarrow \mathbf{R}$ are functions of real values.

For a 2-manifold triangle mesh \mathbf{M} , we may create a geometry image \mathbf{I} from \mathbf{M} by re-sampling the mesh surface through any valid parametrisation process

$$\mathbf{x}(u, v) : (u, v) \in \Omega \rightarrow (x, y, z) \in \mathbf{R}^3 \quad (3)$$

by letting $R(u, v) = x$, $G(u, v) = y$, and $B(u, v) = z$.

3.4 Mesh registration, alignment, and normalization

We follow the work of Zafeiriou et al. [52], which suggests relocating the input mesh to its center of mass and align the mesh to its principle axis. In addition, to be resistant from scaling, the mesh is rescaled to fit a unit volume and the scaling factors are kept and sent along with the secret key.

In our system, we have assumed that the input mesh is a 2-manifold triangular mesh defined in a normalized Cartesian space \mathbf{R}^3 where $\forall v_i = (x_i, y_i, z_i) \in V$, $x_i, y_i, z_i = [0, 1] \in \mathbf{R}$ and that the watermark $\mathbf{W} = \{w_{i,j} | w_{i,j} = 0, 1\}$ is given as an $n \times m$ binary image }.

Prior to the embedding or detection process begins, either the input or the suspicious object is pre-processed to center and align to its center of mass and PCAs, respectively, as suggested by [18, 52].

To locate the mesh on its center of mass C , the calculation is given as follows.

1. Calculate the center of mass C :

$$C = \frac{\sum_{i=1}^n v_i}{n}, \quad (4)$$

2. Relocate all the vertices $v_i \in V$ with respect to the center of mass C :

$$v'_i = v_i - C, \quad (5)$$

If the model has undertaken a translation attack, its center of mass will be translated as well. By relocating the mesh to its new center of mass, we can restore it back to its original position. To ensure the rotation invariance, a common way is to rotate the input mesh so that its principle component is aligned with the Z-axis, where the principle component u of V is the eigenvector correspond to the largest eigenvalue derived from the covariance matrix C of V where C is given as follows:

$$C = \begin{bmatrix} \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i y_i & \sum_{i=0}^n x_i z_i \\ \sum_{i=0}^n y_i x_i & \sum_{i=0}^n y_i^2 & \sum_{i=0}^n y_i z_i \\ \sum_{i=0}^n z_i x_i & \sum_{i=0}^n z_i y_i & \sum_{i=0}^n z_i^2 \end{bmatrix}, \quad (6)$$

After the registration process, the input mesh is then normalized to a unit volume and the aspect ratio is then kept in the file.

3.5 Spherical coordinate transform

Given a vertex $v_i = (x_i, y_i, z_i) \in \mathbf{R}^3$, its spherical coordinate $s_i = (r_i, \theta_i, \phi_i)$ can be calculated as follows.

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \quad (7)$$

$$\theta_i = \tan^{-1} \frac{y_i}{x_i}, \quad (8)$$

$$\phi_i = \cos^{-1} \frac{z_i}{r_i}. \quad (9)$$

On the other hand, to convert a given spherical coordinate representation $s_i = (r_i, \theta_i, \phi_i)$ back to its corresponding Cartesian coordinates, one may proceed as follows.

$$x' = r \cos \theta \sin \phi, \quad (10)$$

$$y' = r \sin \theta \cos \phi, \quad (11)$$

$$z' = r \cos \phi. \quad (12)$$

4 The geometry image based 3D watermarking

As usual, our geometry image-based 3D watermarking system is composed of two subsystems: i.e., the watermark embedding subsystem and the watermark detection subsystem. A block diagram of our system is shown in Fig. 1. Prior to watermark embedding and detection, the input mesh, either the input mesh or the suspicious mesh, is relocated to its center

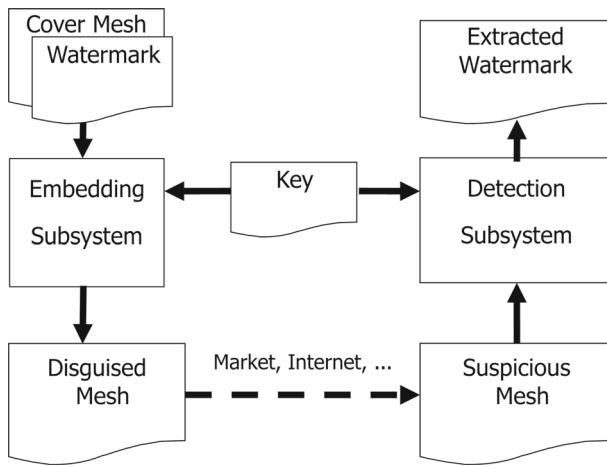


Fig. 1 The block diagram of the proposed watermarking system

of mass and aligned to its principle axis, transformed into the spherical coordinates, and converted into a geometry image.

After the geometry image is generated, the watermark data is then sequenced according to the labelling sequence generated from a secret key. By applying the embedding procedures running on the pixel shaders, the watermark data is then inserted to the pixels of the geometry image corresponding to the labelling sequence into the geometry image. Consequently, a protected mesh can then be derived by converting the geometry image back to a 3D mesh in Cartesian space. Likewise, the suspicious mesh is centered, aligned, and transformed into spherical coordinates, then converted to a geometry image, then a labelling sequence used for watermark bits extraction implemented on the pixel shaders is generated from a secret key.

Note that, any valid mesh parametrization can be applied to generate the geometry image. In our application, we use a simple spherical parametrization for its simplicity and easiness for implementation.

5 The watermark embedding subsystem

The watermark embedding of our approach is summarized as follows. We have assumed that our cover mesh has been converted into an geometry image $G = \{p_{i,j}\}$ of dimension $w \times h$ and that each pixel in G has three color components $p_{i,j} = \{r_{i,j}, g_{i,j}, b_{i,j}\}$, $r_{i,j}, g_{i,j}, b_{i,j} = [0, 1]$. The watermark bits are placed both randomly and redundantly on an image $W = \{w_{i,j}\}$ of the same dimension as the geometry image. To control the embedding and prevent from embedding interferences, a mask image $L = \{l_{i,j} = \{0, 1\}\}$ of the same dimension as the geometry image, i.e., $w \times h$, is created from a given private key. These three images are then transferred to the GPUs as texture images then the embedding code is carried out in parallel by the pixel shaders. Afterwards, the resulting geometry image is used to generate the watermarked mesh. The complete process is illustrated in Fig. 2.

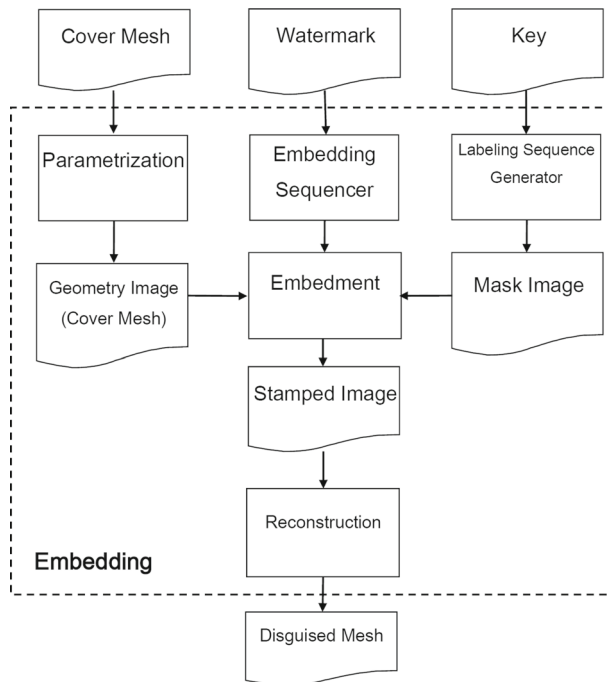


Fig. 2 The embedding subsystem

5.1 Embedding sequencing

To ensure that all the watermark codes are embedded, we may assumed that number of the watermark bits $|\mathbf{W}|$ is much less than the number of pixels of the geometry image \mathbf{G} such that each watermark bit $w_i \in \mathbf{W}$ can be embedded into one or more pixels of \mathbf{G} .

The embedding sequence is randomized by using a mask image $\mathbf{L} = \{l_{i,j} = \{0, 1\}\}$, such as the one depicted in Fig. 3, generated randomly from a secret key $K = RRRRRRRFF$

Fig. 3 The texture pattern of labelling sequence (Key = 9841530012)



of nine digits. In which, the R digits as well as the F digits respectively are the random seed number used for the mask image generation and a reserved number for watermark dependent data such as the number of faces when the watermark is a 3D mesh. Note that the generated mask image are of the same dimension as that of the geometry image G .

After the mask image is generated, it is then transferred along with the geometry image as textures to the shaders of the GPU for watermark embedding operation.

5.2 Watermark embedding

Inspired by [52], two different operators, i.e., \oplus and \ominus , are applied to embed watermark bits to a color component of pixel $p_{i,j}$, say $g_{i,j}$, of the geometry image G of the cover mesh \mathbf{M} according to its shape of local region. If the target is located in a protrusive region, an operator \oplus is used to embed a piece of watermark data w as follows.

$$\begin{aligned} \oplus : g'_{i,j} &= g_{i,j} \oplus w, \\ g'_{i,j} &= \frac{|g_{i,j} \times 10^a|}{10^a} + (w \times 10^{-a}), \quad \text{if } g_{i,j} \geq 0 \\ g'_{i,j} &= \frac{|g_{i,j} \times 10^a|}{10^a} - (w \times 10^{-a}), \quad \text{if } g_{i,j} < 0 \end{aligned} \quad (13)$$

On the other hand, If the target is located in a concave region, an alternative operator \ominus given as follows is applied.

$$\begin{aligned} \ominus : g'_{i,j} &= g_{i,j} \ominus w, \\ g'_{i,j} &= \frac{|g_{i,j} \times 10^a|}{10^a} - (w \times 10^{-a}), \quad \text{if } g_{i,j} \geq 0 \\ g'_{i,j} &= \frac{|g_{i,j} \times 10^a|}{10^a} + (w \times 10^{-a}), \quad \text{if } g_{i,j} < 0 \end{aligned} \quad (14)$$

To help determining whether the shape of the neighbourhood of a given vertex is protrusive or concave, the common 8-neighbor operator $N_8(g_{i,j})$ for any pixel $g_{i,j} \in G$ is defined as follows.

$$N_8(g_{i,j}) = \begin{cases} 1, & \frac{1}{8} \sum_{g \in 8\text{-neighbors}} r(g) < r(g_{i,j}) \\ -1, & \frac{1}{8} \sum_{g \in 8\text{-neighbors}} r(g) > r(g_{i,j}) \end{cases} \quad (15)$$

In which, $r(v)$ represents the magnitude of vertex v 's radius in its spherical coordinate representation.

Given a geometry image $\mathbf{G} = \{g_{i,j}\}$, the embedding of the watermark bits $\mathbf{W} = \{w_{i,j}\}$ is performed according to the corresponding pixel values of the mask image $\mathbf{L} = \{l_{i,j}\}$ and the result of the neighbourhood operator $H(g_{i,j})$ as follows.

$$g'_{i,j} = \begin{cases} g_{i,j}, & \text{if } l_{i,j} = 0. \\ g_{i,j} \oplus w, & \text{if } l_{i,j} = 1 \wedge N_8(g_{i,j}) = 1. \\ g_{i,j} \ominus w, & \text{if } l_{i,j} = 1 \wedge N_8(g_{i,j}) = -1. \end{cases} \quad (16)$$

6 The watermark detection subsystem

Similar to the watermark embedding, the suspicious mesh is centered, aligned and converted to a geometry image, and a labelling image which determines the location of watermark bits is generated from a given secured private key prior to watermark detection. The process is also carried out on the pixel shaders as illustrated in Fig. 4.

After the publication or distribution of the watermarked model, at times, we may find a suspiciously counterfeit over the market. In situations like this, the input mesh could

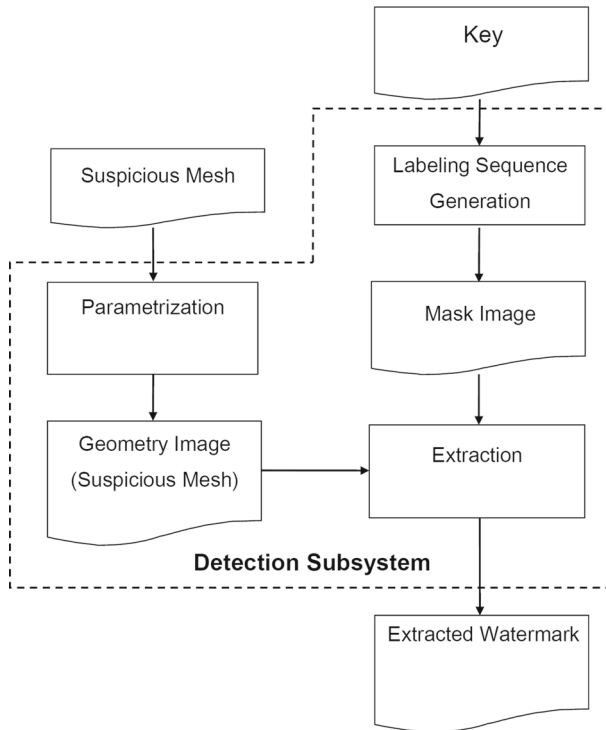


Fig. 4 The watermark detection subsystem

have been transformed or attacked. Therefore, we have to relocate and rescale the model then transform it to the spherical coordinate and built a corresponding geometry image by following the same procedures we have used in watermark embedding.

At times when the model is cropped, a part of geometry image is reconstructed afterward. To cope with this, we use a set of pixels $D = \{d_{i,j} | d_{i,j} = -1, 0, 1, 2\}$ to classify the signals presented in the pixels of the geometry image, i.e., $G' = \{g'_{i,j}\}$, of suspicious mesh as follows.

$$d_{i,j} = \begin{cases} -1, & \text{if } l_{i,j} = 1, H_8(g_{i,j}) = -1; \\ 0, & \text{if } l_{i,j} = 0; \\ 1, & \text{if } l_{i,j} = 1, H_8(g_{i,j}) = 1; \\ 2, & \text{if } l_{i,j} = 1, g_{i,j} = 0. \end{cases} \quad (17)$$

Note that, the case $d_{i,j} = 2$ indicates that the pixel $g'_{i,j} \in G'$ has been cropped or lost and the watermark bits are available only for $d_{i,j} = 1, -1$.

On the basis of D , the watermark bits are detected by the pixel shaders as follows.

If $d_{i,j} = 1$,

$$w_{i,j} = \begin{cases} g'_{i,j} \times 10^a - \left\lfloor g'_{i,j} \times 10^a \right\rfloor, & \text{if } g'_{i,j} \geq 0; \\ g'_{i,j} \times 10^a + \left\lceil g'_{i,j} \times 10^a \right\rceil, & \text{if } g'_{i,j} < 0. \end{cases} \quad (18)$$

Table 1 The test results from the embedding of a tetrahexahedron mesh as the watermark into the Manface, Egea, and Torso meshes with secret key=352124520

Mesh	MSE	SNR	NC
Manface	0.00001254	76.00946045 dB	1.00000024
Egea	0.00001934	73.74573517 dB	1.00000006
Torso	0.00000959	75.58837128 dB dB	0.99999976

If $d_{i,j} = -1$,

$$w_{i,j} = \begin{cases} g'_{i,j} \times 10^a + \left\lfloor g'_{i,j} \times 10^a \right\rfloor, & \text{if } g'_{i,j} \geq 0; \\ g'_{i,j} \times 10^a - \left\lfloor g'_{i,j} \times 10^a \right\rfloor, & \text{if } g'_{i,j} < 0; \end{cases} \quad (19)$$

Owing to the numeric errors introduced by floating point processing, the extracted values of the watermark data may require corrections. To ensure correctness, the watermark is embedded repetitively into the geometry image. For this purpose, the last two digits of the private key are devoted to the division of extracted watermark data so that they can be classified into multiple data arrays of the same size. From these arrays, the finale result are then computed by majority vote and clustering.

Furthermore, the embedding capacity of our algorithm depends on the dimension of geometry image in use. For a 512×512 image, the maximum payload is about $3 * 512 * 512 * w$ bits where w represents the number of bits embedded into a color channel of a pixel. In this paper, we only consider embedding the watermark bits into the g-channel of the pixels selected by the masking sequence generated by a private key.

7 Experimental results

In this paper, three metrics, i.e., the mean square error(MSE), signal-to-noise ratio(SNR), and normalized correlation(NC) are applied to evaluate the results of our experiments. Let

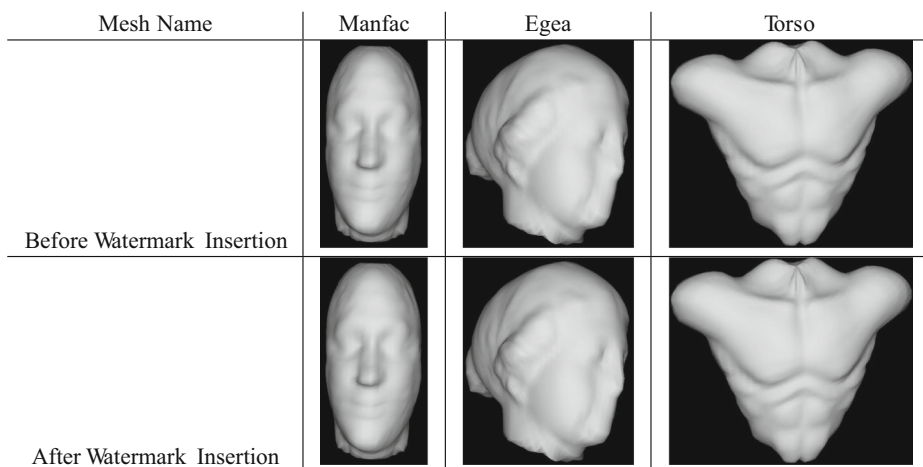


Fig. 5 The cover meshes and their watermarked meshes

$(v = x_i, y_i, z_i)$ be a vertex of the input model and $(v' = x_i, y_i, z_i)$ be the corresponding vertex of the watermark embedded model. An estimation of the mean square error between the two models is given as follows.

$$MSE = \sum_{i=0}^{N-1} (x'_i - x_i)^2 + (y'_i - y_i)^2 + (z'_i - z_i)^2 \quad (20)$$

The extent of variation between the two models can be further evaluation by an estimation of the SNR of the watermarked mesh. Generally speaking, for the human naked eyes, the visual differences between the two models are insignificant if SNR value is greater than 30dB. The SNR of the watermarked mesh compared to its original mesh is calculated as follows.

$$SNR = 10 \log_{10} \frac{\sum_{i=0}^{N-1} (x_i^2 + y_i^2 + z_i^2)}{MSE} \quad (21)$$

On the other hand, the normalized correlation (NC) given as follows can be used to estimate the degree of resemblance of the two models.

$$NC = \frac{\sum_{i=0}^{N-1} (x_i \times x'_i + y_i \times y'_i + z_i \times z'_i)}{\sum_{i=0}^{N-1} x_i^2 + y_i^2 + z_i^2} \quad (22)$$

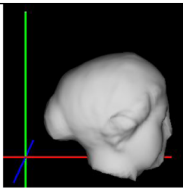
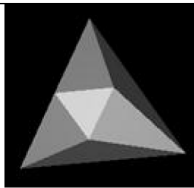
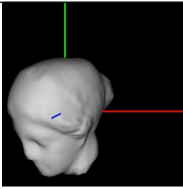
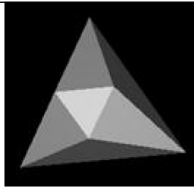
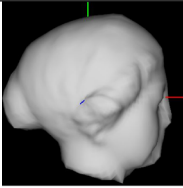

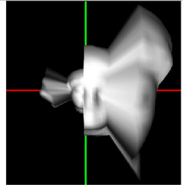
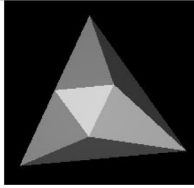
Transformation Attack	Attacked Mesh	Extracted Watermark
translation		
rotation		
uniform scaling		
nonuniform scaling		

Fig. 6 The results of common affine transformation attacks

If $NC = 1$, the two meshes are identical. Smaller value of NC means greater deviation is presented between the two meshes in comparison.

To verify our system, we have tested three meshes from public domain, i.e., the Man-face, Egea, and Torso meshes. The distortion of resulting watermarked mesh is evaluated with regard to geometric metrics including Root Mean Square Error(RMS), Signal-to-Noise Ratio(SNR), and Normalized Cross Relation(NCC) and the rendered images respectively are shown in Table 1 and Fig. 5.

To evaluate the robustness of the watermark, we have simulated the common transformation attacks, noise, cropping, simplification and reordering attacks and the results are presented respectively in Figs. 6, 7, 8, 9, and 10.

The results presented in Fig. 6 show that the watermark detected by the system is robust against common affine transformations such as translation, rotation, uniform and non-uniform scaling.

Simulating noise attack, we use two random variables: one for vertex selection and the other for scaling the displacement of selected vertex along the direction of vertex normal. Part of the results is shown in Fig. 7. For the results with 15 % and 33 % noise, the new system is able to detect the complete watermark but for mesh with 50 %, we found one face of the watermark mesh is lost.

To perform simplification attack, we have adopted the mesh simplification module from Meshlab®, the results are presented in Fig. 8, in which, about 25 %, 55 %, and 68 % vertices of the watermarked mesh are decimated.

In Fig. 8, the detected watermark is complete(100 % detected) when 25 % and 55 % of the vertices of the watermarked mesh are decimated. Even if 68 % of the vertices are decimated, we still can detect almost the complete watermark while only one of its faces is lost.

To evaluate the effects of crop, we have applied two types of crop tests: regular crop and random crop. The results are presented in the Fig. 9. Since we may repetitively embed the watermark into the cover mesh, the new system can successfully detect the complete

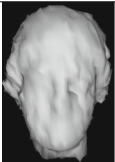

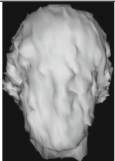



Random Noise	Attacked Mesh	Extracted Watermark	Error Rate	Authentication
15%			16.82%	success
33%			34.00%	success
50%			48.78%	success(one face lost)

Fig. 7 The results of noise attacks

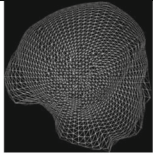

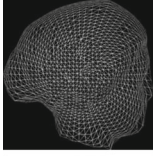

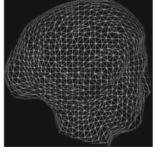

Vertex Decimation	Attacked Mesh	Detected Watermark	Error Rate	Authentication
25%			31.26%	success
55%			42.66%	success
68%			65.56%	success(one face lost)

Fig. 8 The results of simplification attacks

watermark when the size of watermark is small enough to fit in the remainder of the cover mesh.

The results shown in Fig. 9 indicated that our new system is capable of detect the complete watermark even if the mesh is 70 % cropped. For random crop(50 %), only one face of the watermark is lost, the new system can detect almost complete watermark.

The re-meshing attack in our experiment is performed by mid-point subdivision over the disguised mesh. Figure 10 shows the results of applying one, two, and three passes of subdivision; in all these cases, our system can detect complete watermark information.

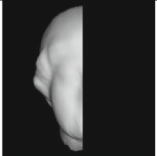

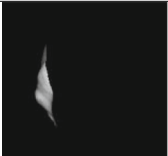



Cropping	Attacked Mesh	Extracted Watermark	Error Rate	Authentication
50%			46.99%	success
70%			67.89%	success
50%(random)			56.42%	success(one face lost)

Fig. 9 The results for mesh cropping attacks

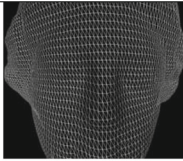

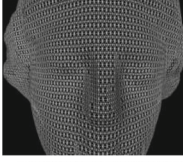

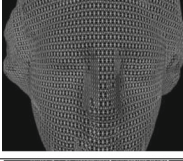

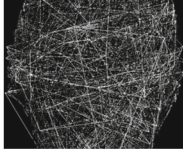

Remeshing	Attacked Mesh	Extracted Watermark	Error Rate	Authentication
subdivision once			1.10%	success
subdivision twice			1.10%	success
subdivision thrice			1.10%	success
random reordering			1.10%	success

Fig. 10 The results of re-meshing and vertex reordering attacks

According to the results presented above, we may conclude that our system is successful in the perspectives of watermark robustness. Furthermore, with the help of GPU computation, the system is highly computational efficient.

8 Concluding remarks and future works

In this paper, we have proposed a novel blind and robust 3D mesh watermarking approach enabling real-time high-speed watermark embedding and detection through GPU computations by transforming the input cover mesh to a geometry image. From the experimental results, it is evident to conclude that our approach is robust against common geometric transforms, noise addition, simplification, crop, and re-meshing attacks.

In addition, the execution times for the embedding and extraction of watermark in all cases are usually less than a second, which complete its execution after a click on the embedding or detection button. Hence, we did not present this part of results in this paper.

Moreover, this approach is also beneficial due to its blindness in watermark extraction. Furthermore, since the watermark embedding and detection of our system are implemented on GPU, it is obviously more computation efficient than almost all the other approaches to 3D mesh watermarking.

However, in this paper, we only adopt a very generic approach to embedding or detection by bit replacement which is vulnerable to quantization attacks. In the near future, we will try

some more scalable and robust spatial and transformed domain techniques from 2D image watermarking for further improvements on the robustness.

References

1. Abdallah EE, Hamza AB, Bhattacharya P (2007) Spectral graph-theoretic approach to 3d mesh watermarking. In: GI '07: Proceedings of Graphics Interface 2007, pp. 327–334. ACM, New York, NY, USA. doi:[10.1145/1268517.1268570](https://doi.org/10.1145/1268517.1268570)
2. Ackerman M (1998) The visible human project. Proc IEEE 86(3):504–511. doi:[10.1109/5.662875](https://doi.org/10.1109/5.662875)
3. Agarwal P, Prabhakaran B (2009) Robust blind watermarking of point-sampled geometry. Trans Info For Sec 4(1):36–48. doi:[10.1109/TIFS.2008.2011081](https://doi.org/10.1109/TIFS.2008.2011081)
4. Benedens O (1999) Geometry-based watermarking of 3d models. IEEE Comput Graph Appl 19(1):46–55. doi:[10.1109/38.736468](https://doi.org/10.1109/38.736468)
5. Benedens O (2003) Robust watermarking and affine registration of 3d meshes. In: IH '02: Revised Papers from the 5th International Workshop on Information Hiding. Springer-Verlag, London, UK, pp 177–195
6. Bors AG, Luo M (2013) Optimized 3d watermarking for minimal surface distortion. Image Processing IEEE Transactions on 22(5):1822–1835
7. Brunton A, Zhao J (2005) Real-time video watermarking on programmable graphics hardware. In: Electrical and Computer Engineering, 2005. Canadian Conference on, pp. 1312–1315. doi:[10.1109/CCECE.2005.1557218](https://doi.org/10.1109/CCECE.2005.1557218)
8. Cai S, Shen X (2011) Octree-based robust watermarking for 3d model. Journal of Multimedia 6(1):83–90
9. Cano ECG, Bassem R, Sabourin R (2013) A parallel watermarking application on a g. Ingenio Magno 3(1):6–15
10. Cotting D, Weyrich T, Pauly M, Gross M (2004) Robust watermarking of point-sampled geometry. In: SMI '04: Proceedings of the Shape Modeling International 2004, pp. 233–242. IEEE Computer Society, Washington, DC, USA. doi:[10.1109/SMI.2004.51](https://doi.org/10.1109/SMI.2004.51)
11. Cox I, Miller ML, Bloom JA (2002) Digital watermarking. Morgan Kaufmann Publishers Inc., San Francisco
12. Floater M, Hormann K, Ks G (2006) A general construction of barycentric coordinates over convex polygons. Adv Comput Math 24(1–4):311–331. doi:[10.1007/s10444-004-7611-6](https://doi.org/10.1007/s10444-004-7611-6)
13. Garg H, Agarwal S (2013) A secure image based watermarking for 3d polygon mesh. SCIENCE AND TECHNOLOGY 16(4):287–303
14. Garimella A, Satyanarayana M, Murugesh P, Niranjan U (2004) Asic for digital color image watermarking. In: Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th, pp. 292–296. IEEE
15. Gu X, Gortler SJ, Hoppe H (2002) Geometry images. ACM Trans Graph 21(3):355–361. doi:[10.1145/566654.566589](https://doi.org/10.1145/566654.566589)
16. Hoashi K, Uemukai T, Matsumoto K, Takishima Y (2009) Constructing a landmark identification system for geo-tagged photographs based on web data analysis. In: ICME'09: Proceedings of the 2009 IEEE international conference on multimedia and expo. IEEE Press, Piscataway, NJ, pp 606–609
17. Jeong YJ, Moon KS, Kim JN (2008) Implementation of real time video watermark embedder based on haar wavelet transform using fpga. In: Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on, vol. 3, pp. 63–66. IEEE
18. Kalivas A, Tefas A, Pitas I (2003) Watermarking of 3d models using principal component analysis. In: ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo. IEEE Computer Society, Washington, pp 637–640
19. Kanai S, Date H, Kishinami T et al. (1998) Digital watermarking for 3d polygons using multiresolution wavelet decomposition. In: Proc. Sixth IFIP WG 5.2 GEO-6, vol. 5, pp. 296–307
20. Karthigaikumar P, Baskaran K (2010) Hardware implementation of invisible image watermarking algorithm using secured binary image authentication technique. International Journal of Electronic Security and Digital Forensics 3(4):333–354

21. Karthigaikumar P, Baskaran K (2011) Fpga and asic implementation of robust invisible binary image watermarking algorithm using connectivity preserving criteria. *Microelectron J* 42(1):82–88
22. Katzenbeisser S, Petitcolas FA (2000) *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, Inc, Norwood
23. Koller D, Levoy M (2005) Protecting 3d graphics content. *Commun ACM* 48(6):74–80
24. Kougianos E, Mohanty SP, Mahapatra RN (2009) Hardware assisted watermarking for multimedia. *Comput Electr Eng* 35(2):339–358. doi:[10.1016/j.compeleceng.2008.06.002](https://doi.org/10.1016/j.compeleceng.2008.06.002). Circuits and Systems for Real-Time Security and Copyright Protection of Multimedia. <http://www.sciencedirect.com/science/article/pii/S004579060800061X>
25. Levoy M, Pulli K, Curless B, Rusinkiewicz S, Koller D, Pereira L, Ginzton M, Anderson S, Davis J, Ginsberg J, Shade J, Fulk D (2000) The digital michelangelo project: 3d scanning of large statues. In: *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 131–144. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. doi:[10.1145/344779.344849](https://doi.org/10.1145/344779.344849)
26. Lin C, Zhao L, Yang J (2011) A high performance image authentication algorithm on gpu with cuda. *International Journal of Intelligent Systems and Applications (IJISA)* 3(2):52
27. Liu Y, Prabhakaran B, Guo X (2012) Spectral watermarking for parameterized surfaces. *IEEE Transactions on Information Forensics and Security* 7(5):1459–1471
28. Luo M, Bors A (2011) Surface-preserving robust watermarking of 3-d shapes. *IEEE Trans Image Process* 20(10):2813–2826. doi:[10.1109/TIP.2011.2142004](https://doi.org/10.1109/TIP.2011.2142004)
29. Maity SP, Banerjee A, Kundu MK (2004) An image-in-image communication scheme and vlsi implementation using fpga. In: *India Annual Conference, 2004. Proceedings of the IEEE INDICON 2004*. First, pp. 6–11. IEEE
30. Maity SP, Kundu MK, Maity S (2009) Dual purpose fwt domain spread spectrum image watermarking in real time. *Comput Electr Eng* 35(2):415–433
31. Mohanty S, Pati N, Kougianos E (2007) A watermarking co-processor for new generation graphics processing units. In: *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on*, pp. 1–2. doi:[10.1109/ICCE.2007.341552](https://doi.org/10.1109/ICCE.2007.341552)
32. Ohbuchi R, Masuda H, Aono M (1997) Watermarking three-dimensional polygonal models. In: *MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia*, pp. 261–272. ACM, New York, NY, USA. doi:[10.1145/266180.266377](https://doi.org/10.1145/266180.266377)
33. Ohbuchi R, Takahashi S, Miyazawa T, Mukaiyama A (2001) Watermarking 3d polygonal meshes in the mesh spectral domain. In: *GRIN'01: No description on Graphics interface 2001*. Canadian Information Processing Society, Toronto, pp 9–17
34. Petitjean G, Dugelay JL, Gabriele S, Rey C, Nicolai J (2002) Towards real-time video watermarking for system-on-chip. In: *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, vol. 1, pp. 597–600. IEEE
35. Praun E, Hoppe H, Finkelstein A (1999) Robust mesh watermarking. In: *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 49–56. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. doi:[10.1145/311535.311540](https://doi.org/10.1145/311535.311540)
36. Rolland-Névière X (2014) Robust 3d watermarking. Université de Nice-Sophia Antipolis, Ph.D. thesis
37. Rolland-Névière X, Doërr G, Alliez P (2014) Spread transform and roughness-based shaping to improve 3d watermarking based on quadratic programming. In: *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 4777–4781. IEEE
38. Rolland-Névière X, Doërr G, Alliez P (2014) Triangle surface mesh watermarking based on a constrained optimization framework. *Information Forensics and Security. IEEE Transactions on* 9(9):1491–1501
39. Rolland-Névière X, Doërr G, Alliez P (2015) Security analysis of radial-based 3d watermarking systems
40. Roy SD, Li X, Shoshan Y, Fish A, Yadid-Pecht O (2013) Hardware implementation of a digital watermarking system for video authentication. *Circuits and Systems for Video Technology. IEEE Transactions on* 23(2):289–301
41. Shi W, Lee HHS, Yoo RM, Boldyreva A (2006) A digital rights enabled graphics processing system. In: *Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics Hardware, GH '06*, pp. 17–26. ACM, New York, NY, USA. doi:[10.1145/1283900.1283903](https://doi.org/10.1145/1283900.1283903)
42. Singh LK, Chaudhry D, Varshney G (2012) A novel approach of 3d object watermarking algorithm using vertex normal, vol 60

43. Snaveley06 N, Seitz S. M, Szeliski R (2006) Photo tourism: exploring photo collections in 3d. In: SIGGRAPH '06: ACM SIGGRAPH 2006 Papers, pp. 835–846. ACM, New York, NY, USA. doi:[10.1145/1179352.1141964](https://doi.org/10.1145/1179352.1141964)
44. Song HS, Cho N. I (2004) Digital watermarking of 3d geometry. In: Intelligent Signal Processing and Communication Systems, 2004. ISPACS 2004. Proceedings of 2004 International Symposium on, pp. 272–277. doi:[10.1109/ISPACS.2004.1439058](https://doi.org/10.1109/ISPACS.2004.1439058)
45. Tirkel AZ, Rankin GA, van Schyndel RM, Ho WJ, Mee NRA, Osborn CF (1993) Electronic water mark. In: DICTA'93: Proceedings of the 2nd Conference on Digital Image Computing: Techniques and Applications. Australian Pattern Recognition Society, Sydney, pp 666–672
46. Uccheddu F, Corsini M, Barni M (2004) Wavelet-based blind watermarking of 3d models. In: MM&Sec '04: Proceedings of the 2004 workshop on Multimedia and security, pp. 143–154. ACM, New York, NY, USA. doi:[10.1145/1022431.1022456](https://doi.org/10.1145/1022431.1022456)
47. Vihari P, Mishra M (2012) Image authentication algorithm on gpu. In: Communication Systems and Network Technologies (CSNT), 2012 International Conference on, pp. 874–878. doi:[10.1109/CSNT.2012.188](https://doi.org/10.1109/CSNT.2012.188)
48. Wagner MG (2000) Robust watermarking of polygonal meshes. In: GMP '00: Proceedings of the Geometric Modeling and Processing 2000. IEEE Computer Society, Washington, p 201
49. Wang K, Lavoue G, Denis F, Baskurt A (2008) A comprehensive survey on three-dimensional mesh watermarking. IEEE Transactions on Multimedia 10(8):1513–1527. doi:[10.1109/TMM.2008.2007350](https://doi.org/10.1109/TMM.2008.2007350)
50. Wang K, Lavoué G, Denis F, Baskurt A (2011) Robust and blind mesh watermarking based on volume moments. Computers & Graphics 35(1):1–19
51. Yi-qiang N, Bo L, Hong-bin Z (2007) A blind watermarking of 3d triangular meshes using geometry image. In: Computer Graphics, Imaging and Visualisation, 2007. CGIV '07, pp. 335–340. doi:[10.1109/CGIV.2007.3](https://doi.org/10.1109/CGIV.2007.3)
52. Zafeiriou S, Tefas A, Pitas I (2005) Blind robust watermarking schemes for copyright protection of 3d mesh objects. IEEE Trans Vis Comput Graph 11(5):596–607. doi:[10.1109/TVCG.2005.71](https://doi.org/10.1109/TVCG.2005.71)
53. Zhan Yz, Li Yt, Wang Xy, Qian Y (2014) A blind watermarking algorithm for 3d mesh models based on vertex curvature. Journal of Zhejiang University SCIENCE C 15(5):351–362



Hung-Kuang Chen received his Ph.D. degrees in computer science of electronic engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 1995 and 2006. From August 1995 to July 2002, he served as a Lecturer at the Department of Electronic Engineering of Lung-Hwa University of Science and technology, Taoyuan, Taiwan. From August 2002 to January 2007, he was a faculty of the Department of Information and Design of Asia University. Since February 2007, he served as an associate professor in the electronic engineering department of the National Chin-Yi University of Technology, Taichung, Taiwan. His research interests cover the computer graphics, virtual reality, and parallel computing.



Wei-Sung Chen received his B.S. and M.S. degrees respectively from the Department of Information Engineering and the Department of Digital Media Design in Asia University, Taichung, Taiwan, in 2006 and 2009. Since 2010, he is a Ph.D. candidate of the department of Computer Science and Engineering at National Chung-Hsing University. His research interests covers Non-Photo-Realistic Rendering, digital watermarking, Steganography and Virtual Reality.