# Progressive Watermarking on 3D Meshes

Hung-Kuang Chen
Dept. of Electronic Engineering
National Chin-Yi University of Technology
Taichung, Taiwan 40109
Email: hankchentw@gmail.com

Yung-Hung Chen
Dept. of Information and Design
Asia University
Taichung, Taiwan
Email: pencialbox@gmail.com

***Abstract*—Traditional 3D watermarking mostly applies off-line decode, which usually requires a complete 3D mesh. For a set of level-of-detail(LOD) meshes, the watermark has to be robust enough to survive from LOD generation algorithms that applies mesh simplification; otherwise, they may have to embed the watermark into each LOD mesh individually. To cope with such problems, we proposed a novel 3D mesh watermarking approach on the basis of the progressive mesh and the discrete wavelet transform. The new approach begins with mutiresolution processing of the cover object and the watermark respectively by a progressive mesh encoder and a Haar-based discrete wavelet transform encoder. By embedding the transformed sequence of the watermark image into the vertex split sequence of the progressive mesh, the watermark and the cover object are transmitted synchronously such that the progressive decode of the cover object and the extraction and decoding of the watermark can be performed on-the-fly with the transmission. Once if the transmission is interrupted, it is still possible to verify the watermark for it is progressively displayed along with the cover object.**

## I. Introduction

The rapid advancements in information technology have brought us into the digital era. Cultural products such as the newspaper, magazines, paintings, photographs, music, video, etc., in our daily life have gradually evolved into digital forms which nowadays are published, distributed, and purchased through various kinds of digital networks. Despite these commercial products, a great portion of official documents are also distributed through digital networks. The conveniences brought by the easiness of duplication and modification in digital form have significantly lowered the costs of manufacture and distribution on one hand. However, they also introduce the problems such as software/product piracy and document counterfeiting on the other hand.

To prevent from such illegal uses, digital watermarking has been proven to be an effective means. By embedding a special secret message, called the watermark, in the content, called the cover object, one may claim his ownership or copyrights over the controversial contents by extracting such secretly hided watermark from it during the law-suit processes. Since such approach is simple and effective, it has been widely studied and applied for the protection/authentication of digital text, image, audio, and video contents in last decade [1], [2]. Recently, with the great improvements in 3D hardware accelerators, vast amount of new applications now make use of a lots of 3D contents. Applications in Virtual Reality systems, scientific/medical visualizations, ancient heritage preservations, geographic information systems, 3D animations, and computer games usually provide a great amount of 3D digital contents [3]–[6]. In comparison with the image watermarking techniques, approaches to digital watermarking on 3D contents are relatively few.

Since most graphics processors usually accept only the polygonal meshes, most 3D digital contents must be converted to polygonal mesh form prior to the rendering process. Watermarking on polygonal meshes has at least the advantage of universally applicable to all kinds of digital 3D contents. To detect the watermark, previous methods mostly require complete information of the cover object. In this paper, we proposed a new approach that is able to detect the embedded watermark signal and verify the transmitted content progressively during the transmission of the cover object.

## II. Related Works

Currently, most digital watermarking methods deal with 1D or 2D contents such as audio, 2D images, and video, etc. By contrast, methods for 3D contents are relatively few. We only introduce the part of works related to our method in this section. For a more comprehensive survey, please refer to the survey papers [1], [2]. Previous literatures suggested several classification criteria to the digital watermarking methods:

- Blindness: a method is called a blind watermarking method if the extraction of the embedded message needs no additional information other than the cover medium itself; otherwise, the method is called a non-blind watermarking method.
- Robustness: a method is claim to be a robust watermarking method if it is able to extract the embedded message successfully from a series intentional/unintentional attacks; opposed to this, if the extraction of the embedded message is very sensitive to the determent of the cover medium, then this method is called a fragile watermarking method.
- Space of Embedment: similar to previous watermarking methods for 2D images, according to the space of the embedment, the watermarking methods for 3D meshes may be classified into spatial domain or transform-domain techniques.

Spatial domain techniques embed the watermark to the spatial information such as the vertex coordinates, vertex normals, and the topology (vertex connectivity) of the cover object. A number of previous works of such kind are illustrated as follows.

Ohbuchi et al. made use of the geometric structure of the cover object in combination with TSQ (Triangle Similarity Quadrature) and TVR(Tetrahedral Volume Ratio embedding) techniques to embed watermark to the vertex coordinates [7]. Benedens etl al. suggested embedding the watermark by modification of surface normal and its distribution [8], [9]. Wagner et al. suggest embedding the watermark in the curvature normal of the target object [10]. Zafeiriou et. al. proposed another method by embedding the watermark data to the r-coordinate of a set of vertices within a certain range of angles in pseudo-random order [11]. The method is robust against both geometric transforms and mesh simplifications. However, It require an alignment process to translate and rotate the disguised mesh to its initial pose. Cotting proposed directly applies watermark embedding to the point clouds without the need to derive consistent connectivity information [12]. On the basis of such concept, Agarwal et. al. proposed another approach that is robust against the attacks such as noise addition and cropping [13]. Their method claim to be robust against uniform affine transformations (rotation, scaling, and transformation), reordering, cropping, simplification, noise addition attacks, remeshing, and progressive compression. The bit-encoding scheme achieves 4 b/point, while maintaining the imperceptibility of the watermark with low distortions. The estimated time complexity is $O(nlogn)$, where $n$ is the number of 3-D points.

The transform domain approaches start with transforming the cover object into a set of basis and associated coefficients followed by embedding the watermark to the coefficients. Kanai et al. applied wavelet-transform to the target object and embedded the watermark in the low-frequency part of the object [14]. Praun et al. suggest conversion of target object into multi-resolution format and embedding the watermark in the low frequency core [15]. 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 [16]. Uccheddu et. al. present another wavelet-based watermarking algorithm for 3D meshes. However, the host meshes are assumed semi-regular to permit a 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 [17]. Abdallah et. al. suggest using spectral conversion [18]. Since direct Laplacian spectral analysis requires a vast of calculations, they suggest partitioning the cover mesh into submeshes and apply the spectral conversion individually to each submesh. Their approach is robust against the geometric transformations, adaptive random noise, mesh smoothing, mesh cropping, and combinations of these attacks. In additon to these, a few more recent approach based on converting the mesh into a spectral geometry image [19] and mesh segmentation with multiple principal plane analysis are reported [20].

When the protected content is transmitted through the network, it is usually converted to a progressive format and transmitted progressively so that it can be viewed or verified progressively over the other end of the communication link. Since the progressive transmission is frequent used, watermark authentication in progressive transmission is important. However, previous methods cannot extract the watermark in accordance to the progress transmission. Since progressive transmission is frequently used, there is an increasing need of the mechanism of watermark authentication in progressive transmission.

We address on this issue and propose a novel approach to providing progressive watermark authentication of progressively transmitted 3D objects. The new method provides progressive watermark authentication of progressively transmitted 3D objects and is robust enough to extract visually distinguishable watermark after common/malicious attacks such as the coordinate transform, cropping, or simplification. Furthermore, the extracted watermark appears progressively as the cover object has been progressively transmitted.

## III. THE PROGRESSIVE WATERMARKING ON PROGRESSIVE MESH

As we have illustrated in Fig. 1, a possible framework of our new watermark authentication system comprises four modules: the preprocessing, embedding, extraction, and authentication modules.

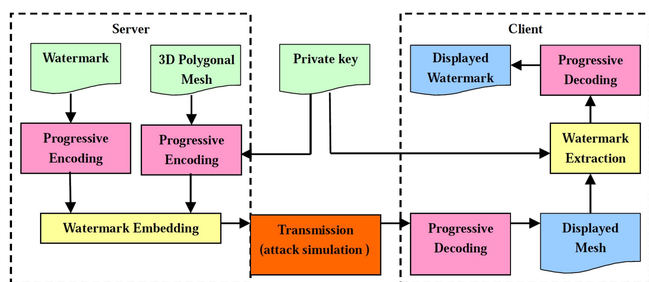



Fig. 1. A possible framework of our new progressive mesh watermarking method.

In our system, we have adopted the approach suggested by Wagner et. al. to embed the watermark into the local curvature normals of the vertices [10]. To facilitate the discussion of our approach, we have made the following assumptions.

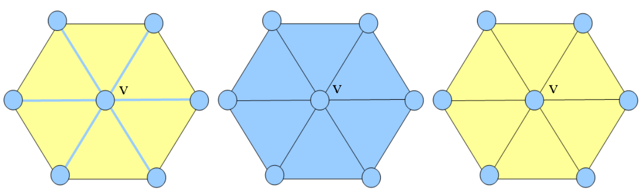
### A. Terminologies and Basic Assumptions

We have assumed that the cover object is a polygonal mesh $\mathbf{M}(\mathbf{V}, \mathbf{E}, \mathbf{F})$ consisting of a set of vertices $\mathbf{V} = \{v_j | v_j = (x_j, y_j, z_j) \in \mathbf{R}^3\}$ in a 3D Cartesian space, a set of edges $\mathbf{E} = \{(v_i, v_j) | v_i, v_j \in \mathbf{V}\}$, and a set of triangular faces $\mathbf{F} = \{(v_i, v_j, v_k) | v_i, v_j, v_k \in \mathbf{V}\}$. 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.

Furthermore, 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.

- $\lceil v_i \rceil$: the set of adjoint edges connected to $v$, where $\lceil v_i \rceil = \{e_{ij} | e = \bar{v_i v_j} \in \mathbf{E}\}$ is called the star of $v_i$, denoted as $\mathbf{S}(v_i)$.
- $\lceil\lceil v_i \rceil\rceil$: the set of faces $f \in \mathbf{F}$ adjacent to $v$ denoted as $\mathbf{R}(v_i)$.
- $\lfloor\lceil v_i \rceil\rfloor - v_i$ or $\partial v$: the boundary vertices of $\mathbf{S}(v_i)$ or commonly called the crown of $v_i$ denoted as $\mathbf{C}(v_i)$.

An example is shown in the Fig. 2 for better understanding of these terminologies.




(a) $\lceil v_i \rceil$ (b) $\lceil\lceil v_i \rceil\rceil$ (c) $\lfloor\lceil v_i \rceil\rfloor - v_i$ or $\partial v$

Fig. 2. Some terminologies associated with the neighborhood of a vertex

Finally, we assume that the watermark $\mathbf{W} = \{w_i | w_i$ is a b-bit binary code, $1 \leq i \leq m, i \in \mathbf{N}\}$.

### B. Embedding Sequencing

Furthermore, to ensure that all the watermark codes are embedded, we may assumed that a function exists that maps from a set of watermark indices $\mathbf{I} = \{1, 2, \cdots, m\}$ to a set of vertex indices $\mathbf{J} = \{1, 2, 3, \cdots, |\mathbf{V}|\}$ and at least a vertex index $j$ correspond to $i$ for each code of the input watermark $w_i \in \mathbf{W}$, or $j = f(i)$.

This mapping can be stochastic. With a secret key $k$, we may have $f = h \circ g$ where $h : \mathbf{I} \to \mathbf{K} \wedge g : \mathbf{K} \to \mathbf{J}$ and applies a one-to-one mapping stochastic function $S : \mathbf{K} \to \mathbf{J}$ to randomize the mapping. Note that we must satisfies at least a one-to-one correspondence to embed all the watermark codes, which implies that the cardinality of the input watermark $\mathbf{W}$ must be lower than that of the vertex set $\mathbf{V}$, i.e., $|\mathbf{W}| \leq |\mathbf{V}|$.

A possible transformation invariant design of such functions have been describe earlier [10], in which, $h : \mathbf{I} \to \mathbf{S}$ accepts an integer $i$ as the index of vertex $v_i \in \mathbf{V}$ and returns a transform invariant normal $n_i$ of $v_i$ in spherical coordinate $S$ and $g : \mathbf{S} \to \mathbf{J}$ returns an integer $j$ as the index to the watermark code $w_j$ in the stream of $\mathbf{W}$.

### C. Progressive Encodement

In our approach the process start with a discrete wavelet transform of the input watermark and a generic edge-collapse-based mesh simplification over the cover object [21]. We begin by a brief introduction to our generic polygonal mesh simplification algorithm followed by the discussion of the iterative Haar basis wavelet encoder [21]. In both cases, the derived outputs comprise a base part and a series of refinement records sequenced from lower to higher frequencies.

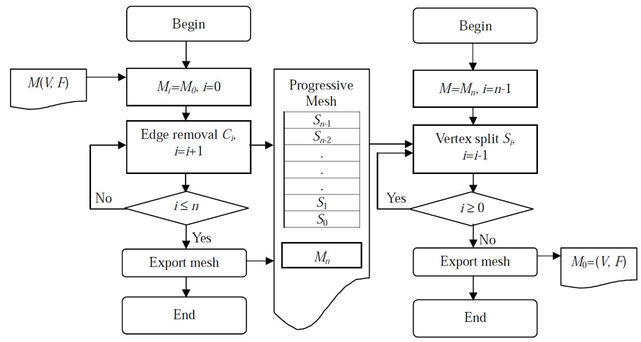



Fig. 3. The conversion and restoration of a progressive mesh.

### D. Progressive Mesh Encoder

For a 3D polygonal mesh, we adopt the progressive mesh (PM) suggested by Hoppe et al. [22]; hence, the cover object, a polygonal mesh $M_0$, is converted to a base mesh $M_n$ and a series of refinement operations, the vertex splits $\mathbf{S} = \{s_{n-1}, s_{n-2}, ..., s_0\}$, after $n$ edge collapses applied to $M_0$.

The conversion and restoration to and from the original mesh are illustrated by the flow diagram shown in Fig. 3.

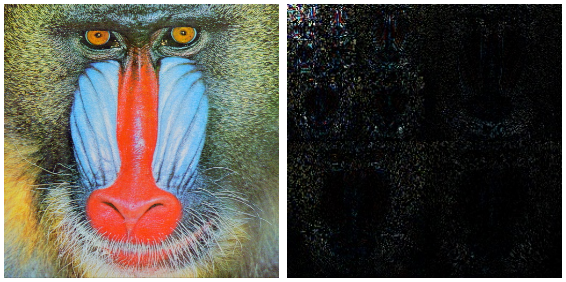
We have devised our simplification algorithm on the basis of a fast edge collapse-based simplification algorithm [23], which performs the simplification by selecting the lowest cost edge collapse among the star of a vertex. To simplify the encoding and decoding process, the overall greedy selection is not applied. In order to guarantee a consistent output, the simplifications are performed under the following restrictions:

- Consistent manifold property: each edge collapse performed must guarantee a manifold output. Hence, if two rings $\mathbf{R}(v_i)$ and $\mathbf{R}(v_j)$ are merged after the collapse of edge $e_{ij} = (v_i, v_j)$, the resulting ring $\mathbf{R}(v_i, j)$ must be manifold.
- Independency: the simplified region should be excluded from further simplification to avoid repeated embedding.

### E. 2D-DWT Encoder

At present, there are a number of ways to do 2D image wavelet transform [10]. However, deciding which is optimal is rather difficult. Since the design of a DWT basis is not in the focus of this paper, we simply implemented a basic iterative Haar wavelet encoder $\mathbf{W}_{n\times n}$ to process the 2D image $\mathbf{I}_{n\times n}$ [21]. This process is repeated until the desired lowest resolution image is derived. An example is shown in Fig. 4, in which the baboon image is at first rescaled to $256 \times 256$ pixels then iteratively transformed with out wavelet encoder $\mathbf{W}$ seven times.

$$\mathbf{W} = \begin{bmatrix} \mathbf{A}_{\frac{n}{2}\times n} \\ \hline \mathbf{D}_{\frac{n}{2}\times n} \end{bmatrix}$$



(a) Baboon (b) DWT encoded Baboon

Fig. 4. An example of DWT image transform.

Note that $\mathbf{A}$ and $\mathbf{D}$ are the sum and difference filters of two neighboring pixels, respectively. With $\mathbf{W}_{n\times n}$ we may derive a lower resolution image and three sets of refinement coefficients as follows.

$$\mathbf{WIW^T} = \begin{bmatrix} \mathbf{AIA^T} & \mathbf{AIG^T} \\ \mathbf{GIA^T} & \mathbf{GIG^T} \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{V} \\ \mathbf{H} & \mathbf{D} \end{bmatrix}$$

### *F. The Watermark Embedding*

We assume that the watermark $\mathbf{W}$ and a function $w_j = f(v_i)$, $w_j \in \mathbf{W} \wedge v_i \in \mathbf{V}$ exists that help us establish the correspondence between the vertices of the input mesh and the data codes of the watermark such that we can identify for each vertex which watermark code was embedded and vice versa. Let the watermark be transformed to $n+1$ layers, in our implementation, we streamed the watermark as follows.

$$\mathbf{B}_n \rightarrow (\mathbf{v},\mathbf{h},\mathbf{d})_n \rightarrow (\mathbf{v},\mathbf{h},\mathbf{d})_{n-1} \rightarrow \cdots \rightarrow (\mathbf{v},\mathbf{h},\mathbf{d})_0$$

The transformed image is the sequenced in this manner and embedded into the vertex split stream of the progressive mesh from low to high frequencies. The process of such embedment is illustrated in Fig. 5, in which the stream of watermark codes are embedded level-by-level to the vertex split stream. Thus, even if the transmission is interrupted, the extracted low frequency part of the watermark still allows visual judgment. Furthermore, by embedding the stream of the watermark into the stream of progressive mesh, the extracted image will be clear from the vague to the effect of gradually.

Similar to Wagner's approach, we will embed one watermark code for each vertex in its relative scale $l$ . Given a vertex $v_i$ and its vertex ring $S_i$ , we can calculate the average normal vector $n_i$ of the vectors from $v_i$ to each vertex on the crown of $v_i$ as follows.

$$n_i = \frac{1}{|\mathbf{R}(v_i)|} \sum_{\forall v_j \in \mathbf{R}(v_i)} (v_j - v_i) \tag{1}$$

By summing up the normals, we can find the average length of the normals $\bar{n}$ by

$$\bar{n} = \sum_{\forall v_i \in \mathbf{V}} n_i \tag{2}$$

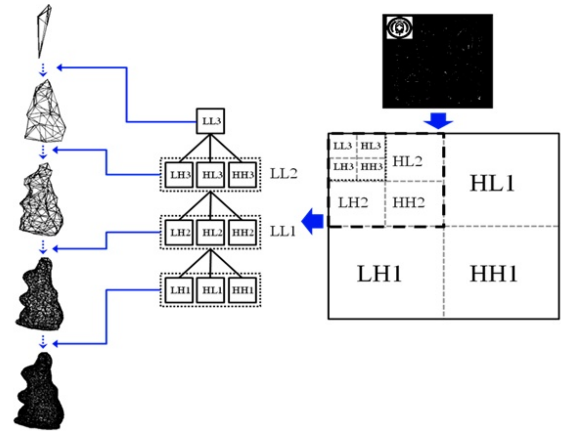



Fig. 5. An illustration of the embedment of DWT transformed watermark into the vertex split stream of a progressive mesh.

Then, for each we compute a relative scale $l_i$ of the normal $n_i$ with respect to $\bar{n}$ by letting where $c$ is a heuristic constant.

$$l_i = c \times \frac{|n_i|}{\bar{n}} \tag{3}$$

On the basis of these calculations, the embedding function $P$ given as follows.

$$l'_i = P(l_i, w_j) = \left|\frac{l_i}{2^b}\right| \times 2^b + w_j, b \geq \log_2 w_j \tag{4}$$

Thus, the normal of $v_i$ after embedding $w$ is

$$n'_i = l'_i \times \frac{\bar{n}}{c} \times \frac{n_i}{|n_i|}. \tag{5}$$

To reflect the adjusted relative scale $l'_i$ the new position of $v_i$ is calculated as follows.

$$n'_i = \frac{1}{|\mathbf{R}(v_i)|} \times \sum_{\forall v_j \in \mathbf{R}(v_i)} (v_j - v_i) = \frac{\sum_{\forall v_j \in \mathbf{R}(v_i)} v_j}{|\mathbf{R}(v_i)|} - v'_i$$

$$\Rightarrow v'_i = \frac{\sum_{\forall v_j \in \mathbf{R}(v_i)} v_j}{|\mathbf{R}(v_i)|} - n'_i \tag{6}$$

Note that we must satisfies at least a one-to-one correspondence to embed all the watermark codes, which implies that the cardinality of the input watermark $\mathbf{W}$ must be lower than that of the vertex set $\mathbf{V}$, i.e., $|\mathbf{W}| \leq |\mathbf{V}|$.

Furthermore, to avoid the embedding dependency problem, we will not embed watermark into the crown of a vertex if its relative scale has embedded with a watermark code. Furthermore, to reduce the influence of embedded watermark, the heuristic constant must be chosen carefully to satisfy

$$\frac{2^{\lceil \log_2 w_j \rceil + 1}}{2^{\lceil \log_2 l_i \rceil + 1}} \geq \epsilon$$

where $\epsilon$ is the lower bound of data loss.

### G. Watermark Extraction

To extract the water $w_i$ from a vertex $v_i$, we must have the original average length $L$ and the constant $c$. By calculating the average normal vector $n'_i$ for each vertex, we can find

$$l'_i = c \times \frac{|n'_i|}{\bar{n}} \tag{7}$$

Because $\lfloor \frac{l_i}{2^b} \rfloor = \lfloor \frac{l'_i}{2^b} \rfloor$, by substituting this result into Eq. 4, we have

$$w_j = l'_i - \lfloor \frac{l_i}{2^b} \rfloor \times 2^b. \tag{8}$$

### H. Watermark Authentication

After the extraction of the embedded watermark, it is necessary to provide a means to verify or compare the extracted watermark with the original one. In this paper, we have adopted the normalized correlation (NC) [24] as the way to compare the extracted watermark with the original watermark data, which gives a quantified measurement of similarity for the certification of the watermark. It's advantages are obvious: fault tolerance and objectivity. The formula is given as follows:

$$NC = \frac{\sum_{\forall w_i \in \mathbf{W}} \sum_{\forall w_j \in \mathbf{W}'} w_i \times w'_j}{\sum_{\forall w_i \in \mathbf{W}} \sum_{\forall w_j \in \mathbf{W}'} {w_i}^2} \tag{9}$$

When the extracted watermark perfectly matches the original watermark, the NC value is equal to 1. As more inconsistencies are found between the extracted and the original watermarks, the NC value will be reduced. Hence, we may evaluate the extracted watermark quality through the NC value.

### I. Distortion Measurement

*1) The extracted watermark:* For two 3D meshes or 2D images, to judge the extent of their differences merely by visual perception is not convincing. To provide a modest assessment for the verification of the extracted watermark, a common way is to calculate the noise ratio compared with the original watermark (peak to noise ratios, PSNR). In general, the noise ratio is defined as

$$PSNR = 10 - \log_{10} \frac{255^2}{MSE} dB, \tag{10}$$

$$MSE = \frac{1}{m^2} \times \frac{1}{\sum_{i=1}^{m} \sum_{j=1}^{m} (\alpha_{ij} - \beta_{ij})^2} \tag{11}$$

In Eq.11, MSE stands for the the mean square error of two $m \times m$ gray-scale digital images, $\alpha_{ij}$ and $\beta_{ij}$ respectively represents the pixel of the original watermark images and the pixel of the extracted watermark located on position $(i, j)$. Note that, larger PSNR value means greater similarity.

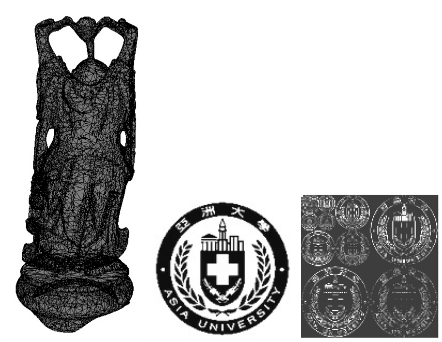
*2) The Disguised Mesh:* To measure the distortion between the disguised mesh and the original one, we have adopted the Metro developed by the Visual and Computer Graphics Laboratory. With Metro, a variety of comparative measurements can be derived: the NC, the PSNR, and the Hausdorff distance [25]. The PSNR values can be calculated by modifying Equation 10 as follows.

$$PSNR = 10 - \log_{10} \frac{B^2}{MSE} dB \tag{12}$$

In Eq.12, B is the bounding box diagonal length of the cover mesh. Note that, for a 3D mes, the MSE value is the mean square distance between corresponding pairs of vertices between the original and the disguised mesh.

## IV. Experimental results

In the experiments, we will evaluate the impact of the watermark embedding from two perspectives: the distortion of the cover mesh and the quality of the extracted watermark. We begin by examining the distortion of the cover mesh under various conditions of embedment by varying the embedding parameter b and the spectrum of embedment. Then evaluate the impact of the embedding parameters to the quality of watermark. In addition, we have also verified the robustness of the watermark under affine transformation attacks. The watermark image used in the experiments is rescaled to a $2^k \times 2^k$ gray image. After applying wavelet transform to this image, the resulting stream comprises $2^{2k}$ codes. Assuming the progressive mesh to be embedded contains $n$ vertex splits, a straightforward sequential embedding by letting $j = f(i) = i$ mod $2^{2k}$ which requires $n \leq 2^{2k}$. The models as well as the watermark image used in our experiment are shown in Fig. 6. The cover objects, the polygonal meshes, are retrieved from the Stanford 3D Scanning Repository.




(a) (b) (c)

Fig. 6. The cover mesh and the watermark image:(a) the cover mesh (Happy Buddha);(b)the watermark; the watermark image after applying DWT transform.

### A. The Impact of Watermark Accuracy

After the watermark is embedded, the model will more or less suffer from a certain degree of distortion. According to Table I, as more bits are embedded in each primitive, the PSNR value of the cover mesh increases, i.e., the extent of distortion becomes more significant. On the other hand, as more bits are embedded into the cover mesh, the quality of extracted watermark is getting better. That is to say, the larger the value of $b$ is, the watermark becomes more robust against the attacks at the cost of greater the model distortion.

TABLE I
THE IMPACT OF CODE SIZE B TO THE QUALITY OF THE WATERMARK AND THE DISGUISED MESH.

| Code Size b | 8 BIT | 12 BIT | 16 BIT |
|---|---|---|---|
| Disguised Mesh(PSNR) | 93.56171 | 69.696938 | 44.997974 |
| Disguised Mesh(NC) | 0.999991 | 0.999898 | 0.995793 |
| Watermark(PSNR) | 7.212543 | 38.979134 | 53.081692 |
| Watermark(NC) | 0.647436 | 0.998296 | 0.998356 |
| Hausdorff | 0.000117 | 0.001252 | 0.006401 |

### B. The Impact of Embedding location

In the experiments we set $b$ to 12 bits and embed the watermark in the 30,737 vertex split records. The embedded watermark is a discrete wavelet transformed 2D image ($2^7 \times 2^7$) that comprises $2^{14} = 16,384$ records. The embedment to the vertex split sequences are started from the base mesh to the last record, which corresponds to low-frequency to high-frequency wavelet coefficients. In this study, we also evaluated the impact of the embedment in various frequency locations of the model to the distortion of the cover mesh and the quality of the extracted watermark. The results are presented in Table II. As can be seen from Table II, embedding the watermark in low-frequency part of the cover mesh improves the quality of the extracted watermark but causes greater damage to the cover mesh.

TABLE II
THE DEGREE OF DISTORTION IN PSNR OF THE COVER MESH AFTER THE EMBEDMENT OF WATERMARK

| V-Split Sequence | 0 | 2031 | 4031 | 8030 | 12030 |
|---|---|---|---|---|---|
| Distortion (PSNR) | 71.35 | 75.25 | 76.91 | 79.38 | 81.36 |

### C. Geometric Transformation Attacks

By relocating the center and re-aligned the principle axis of the mesh, our approach, as can be seen from Fig 7, is able to resist geometric transformations such as rotation, scaling, and translation.

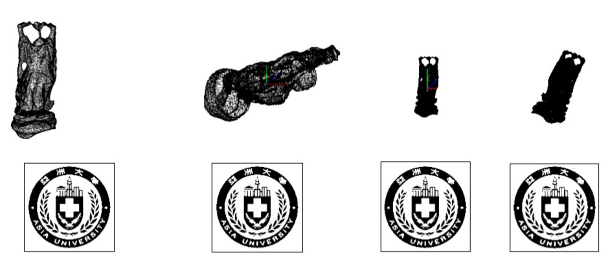


Fig. 7. The extracted watermark after the rotation, scaling, and translation attacks.

### D. Progressive Decoding

The main feature or our method is the capability of decoding and displaying the watermark during the transmission of the cover object. Fig. 8 shows the decoded watermark during the transmission of the cover mesh. From Fig. 8, it is evident that our method is able to decode and display the watermark during the transmission and decoding of the progressive mesh of the disguised model. We may also find that the resolution of the displayed watermark changes its resolution in accordance with the amount of decoded vertex splits records of the progressive mesh.

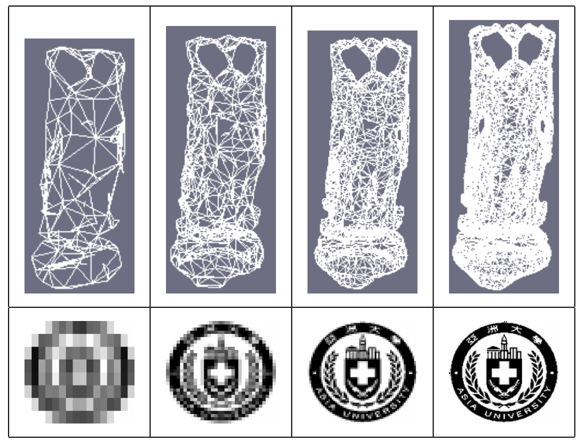


Fig. 8. The extracted watermark during the transmission of the cover mesh.

## V. CONCLUSION

In this paper, we have proposed a novel approach to digital watermarking on the polygonal meshes which embeds a Haar Discrete wavelet transformed multi-resolution watermark image into the vertex split sequence of the cove object's progressive mesh. The new approach is quite satisfactory and achieves at least the following contributions:

- *Online progressive display of watermark*: according to our knowledge, our approach is the first watermarking approach to progressive transmission decoding of watermark. Instead of traditional offline decoding after the transmission of the watermark, it is possible to do online authentication during the transmission of a 3D mesh with our approach.
- *Robust against geometric transforms*: as we have described earlier in the experiment section, the new approach is also robust against the common geometric transforms by relocation of the center and the alignment of the principle axis.
- *Survive from transmission interruption*: since the decode of the watermark is carried out on-the-fly with the transmission and decoding of the progressive mesh of the cover mesh, the watermark is still detectable if the transmission is accidentally interrupted.

In summary, our new approach has successfully achieves the goal of our design and has at least three advantages against the previous approaches to 3D mesh watermarking. The proposed approach is novel and especially useful in the circumstances where online verification and authentication are needed. In addition, it is also interesting to note that, with our new approach, we can classify the information into several classes according to its importance and hide these information to the vertex slit stream accordingly, which might be useful for hiding information of different extent of secrecy.

Somehow, in this paper, we only follow the spatial technique proposed by [10] to embed the watermark, which is not robust enough against mesh simplification attack. It will be interesting to develop a robust progressive watermarking approach by transformed domain techniques or a fragile watermarking approach for the purpose of authentication and verification.

## ACKNOWLEDGMENT

The authors would like to thank the Stanford 3D Scanning Repository for providing us all the test meshes and the support by National Science Council of the Republic of China under the contract NSC-98-2221-E-167-015.

## REFERENCES

[1] S. Katzenbeisser and F. A. Petitcolas, Eds., *Information Hiding Techniques for Steganography and Digital Watermarking*. Norwood, MA, USA: Artech House, Inc., 2000.

[2] I. Cox, M. L. Miller, and J. A. Bloom, *Digital watermarking*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[3] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The digital michelangelo project: 3d scanning of large statues," in *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144.

[4] N. Snavely06, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM, 2006, pp. 835–846.

[5] K. Hoashi, T. Uemukai, K. Matsumoto, and Y. Takishima, "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*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 606–609.

[6] M. Ackerman, "The visible human project," *Proceedings of the IEEE*, vol. 86, no. 3, pp. 504 –511, mar 1998.

[7] R. Ohbuchi, H. Masuda, and M. Aono, "Watermaking three-dimensional polygonal models," in *MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia*. New York, NY, USA: ACM, 1997, pp. 261–272.

[8] O. Benedens, "Geometry-based watermarking of 3d models," *IEEE Comput. Graph. Appl.*, vol. 19, no. 1, pp. 46–55, 1999.

[9] ——, "Robust watermarking and affine registration of 3d meshes," in *IH '02: Revised Papers from the 5th International Workshop on Information Hiding*. London, UK: Springer-Verlag, 2003, pp. 177–195.

[10] M. G. Wagner, "Robust watermarking of polygonal meshes," in *GMP '00: Proceedings of the Geometric Modeling and Processing 2000*. Washington, DC, USA: IEEE Computer Society, 2000, p. 201.

[11] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3d mesh objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 5, pp. 596–607, 2005.

[12] D. Cotting, T. Weyrich, M. Pauly, and M. Gross, "Robust watermarking of point-sampled geometry," in *SMI '04: Proceedings of the Shape Modeling International 2004*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 233–242.

[13] P. Agarwal and B. Prabhakaran, "Robust blind watermarking of point-sampled geometry," *Trans. Info. For. Sec.*, vol. 4, no. 1, pp. 36–48, 2009.

[14] S. KANAI, H. DATE, T. KISHINAMI, and S. K. H. Date, "Digital watermarking for 3d polygons using multiresolution wavelet decomposition," in *Proc. Sixth IFIP WG 5.2 GEO-6*, 1998, pp. 296–307.

[15] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 49–56.

[16] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, "Watermarking 3d polygonal meshes in the mesh spectral domain," in *GRIN'01: No description on Graphics interface 2001*. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2001, pp. 9–17.

[17] F. Uccheddu, M. Corsini, and M. Barni, "Wavelet-based blind watermarking of 3d models," in *MM&Sec '04: Proceedings of the 2004 workshop on Multimedia and security*. New York, NY, USA: ACM, 2004, pp. 143–154.

[18] E. E. Abdallah, A. B. Hamza, and P. Bhattacharya, "Spectral graph-theoretic approach to 3d mesh watermarking," in *GI '07: Proceedings of Graphics Interface 2007*. New York, NY, USA: ACM, 2007, pp. 327–334.

[19] Y. He, B.-S. Chew, D. Wang, C.-H. Hoi, and L.-P. Chau, "Streaming 3d meshes using spectral geometry images," in *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*. New York, NY, USA: ACM, 2009, pp. 431–440.

[20] S.-C. Cheng, C.-T. Kuo, and D.-C. Wu, "A novel 3d mesh compression using mesh segmentation with multiple principal plane analysis," *Pattern Recogn.*, vol. 43, no. 1, pp. 267–279, 2010.

[21] E. J. Stollnitz, T. D. Derose, and D. H. Salesin, *Wavelets for computer graphics: theory and applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[22] H. Hoppe, "Progressive meshes," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1996, pp. 99–108.

[23] H.-K. Chen, C.-S. Fahn, J. J. P. Tsai, R.-M. Chen, and M.-B. Lin, "Generating high-quality discrete lod meshes for 3d computer games in linear time," *Multimedia Syst.*, vol. 11, no. 5, pp. 480–494, 2006.

[24] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58–68, 1999.

[25] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces," Paris, France, France, Tech. Rep., 1996.