Technologies for 3D Model Watermarking: A Survey

Chang-Min Chou[1, 2] and Din-Chang Tseng[1]

[1] Institute of Computer Science and Information Engineering, National Central University, Taiwan 320.
[2] Department of Electronic Engineering, Ching Yun University, Taiwan 320.

Summary
Due to the explosive growth of Internet and the development of digital content designing and processing techniques, many valuable materials can be represented in digital forms for exhibition and access via Internet. Due to the characteristics of easy duplication and modification of digital contents, it is necessary to develop a variety of watermarking techniques for various protection purposes such as ownership claiming and authentication. In this survey paper, we examine 3D model watermarking technologies developed over the last decade. We classify various algorithms into two classes: robust watermarking and fragile watermarking. We describe main ideas behind each class, and compare the advantages and disadvantages of the algorithms in each class. Finally, we address some trends in the 3D model watermarking technology development.

Key words:
3D watermarking, 3D mesh, Digital signature, Copyright protection, information security

1. Introduction

In recent years, 3D graphic models have become more accessible to general end users due to the usage of advanced scanning devices and the virtual-reality modeling language (*VRML*) for graphic description. Moreover, due to the explosive growth of Internet and the development of digital content designing and processing techniques, many valuable materials can be represented in digital forms for exhibition and access via Internet. Due to the characteristics of easy duplication and modification of digital contents, it is necessary to develop a variety of digital signature or watermarking techniques for various protection purposes such as model authentication and ownership claiming. Digital signatures [1, 2] are designed for the receiver of electronic documents to verify the identity of the sender and to check the originality of the documents. The watermarking schemes are usually designed for the sender to check the copyright ownership (robust watermarking) or for the receiver to verify the authentication of the received media (fragile watermarking). The main difference between digital signature and watermarking techniques is that the former attaches a small piece of information (the digital signature) transmitted with the original documents whereas the latter embeds invisible information (the watermarks) in the original media.

Encryption techniques [3, 4] can be symmetric or asymmetric [5]. Generally speaking, asymmetric methods have better security, but are very computationally expensive when comparing to symmetric methods. A compensation scheme is to encrypt the digital contents by a symmetric method and then encrypt the "symmetric key" by an asymmetric method. A typical digital signature scheme extracts a "message digest" from the original document and then encrypts the message digest by an asymmetric method. The message digest can represent the original document in such a way that two different message digests will be generated for two different documents even if they have only one bit of difference. The encrypted message digest is attached to the original document and transmitted to the receiver. Although all data types can be treated as binary data files and encrypted by digital signature schemes, directly applying digital signature for large multimedia data sets is still costly and computationally expensive [6]. Instead of using digital signature for all kinds of electronic documents, researchers develop various watermarking schemes for various multimedia data types such as audio, images, video, and 3D models. Watermarks can be invisibly/inaudibly embedded in these media by altering some of their lower significant bits. Watermarking schemes usually don't need any complex computation (at most *FFT* or wavelet transform [7] is used); thus they can be performed in a fast and low cost way comparing to the digital signature schemes.

According to the application purposes, watermarking techniques can be classified into robust and fragile schemes. Robust watermarking is usually designed for ownership claiming while fragile watermarking is used for digital content authentication and verification. The design goal of robust watermarking is to make the embedded watermarks remain detectable after being attacked. In contrast, the requirements of fragile watermarking are to detect the slightest unauthorized modifications and locate the changed regions.

According to the extraction strategies of watermarks, watermarking schemes can be classified into private and public. A private watermarking scheme needs the original

Manuscript received February 5, 2007
Manuscript revised February 25, 2007

model and watermarks to extract the embedded watermarks while a public (or called blind) scheme can extract watermarks in the absence of the original model and the watermarks. The word "public" used in watermarking schemes is different from that represented in cryptography schemes. In addition, a "semi-public" watermarking scheme does not need the original model in the watermark extraction stage, but the original watermarks are necessary for comparing with the extracted watermarks. From the viewpoints of technique development and applications, all fragile watermarking schemes are public. It is nonsense and inefficient to use a private fragile watermarking because if we have the original model base, we can search the corresponding model and directly compare with the marked model.

2. Robust Watermarking for 3D Models

Most early 3D watermarking algorithms were designed as robust watermarking scheme for ownership claiming and copyright protection. In addition, some researchers tried to hide data into 3D models by using 3D robust watermarking techniques [8, 9]. Ohbuchi *et al*. [10] proposed three requirements for 3D robust watermark embedding: unobtrusive, robust, and space efficient. Unobtrusive means the embedding must not interfere with the intended use of a model, such as viewing. Robust means the embedded watermarks should remain detectable after being maliciously attacked. Space efficient means an embedding method should be able to embed sufficient amount of information into models.

Initially, Ohbuchi *et al*. [10] proposed a large variety of techniques for embedding data in 3D polygonal models. In 3D models, there are two attributes that can be used for watermark embedding: the geometry (coordinate) information and the topology (connectivity) information. Ohbuchi *et al*. argued that geometry is the best data type in a 3D geometrical model for embedding. They presented various fundamental methods for embedding data into a polygonal model. We here introduce a triangle similarity quadruple (*TSQ*) embedding scheme proposed in [10] as an illustration.

The *TSQ* algorithm uses a dimensionless quantity pair such as $\{b/a, h/c\}$ in Fig. 1 to define a set of similar triangles. The algorithm uses a quadruple of adjacent triangles that share edges in the configuration depicted in Fig. 2 as a Macro-Embedding-Primitive (*MEP*). Each MEP stores a quadruple of values {Marker, Subscript, Data1, Data2}. A marker is a special value (in this case $\{b/a, h/c\}$) that identifies *MEP*s. In Fig. 2, the triangle marked *M* stores a Marker, *S* stores a Subscript, and *D*1 and *D*2 store data values Data1 and Data2. While each *MEP* is formed by topology, a set of *MEP*s are arranged by the quantity of subscript. The *TSQ* algorithm embeds a message according to the following steps.

(1) Traverse the input triangular mesh to find a set of four triangles to be used as an *MEP*. In doing so, avoid vertices that have already been used for the watermark, or triangles that are unfit for stable embedding, such as triangles whose dimensionless quantities are too small.

(2) Embed the marker value in the center triangle of the *MEP* by slightly changing the coordinates of vertices v_1, v_2, and v_4 such that the quantity pair $\{e_{14}/e_{24}, h_4/e_{12}\} = \{b/a, h/c\}$ (refer to Fig. 2). e_{ij} represents the edge between v_i and v_j.

(3) Embed a subscript and two data symbols in the remaining three triangles of the *MEP* by slightly changing the coordinates of vertices v_0, v_3, and v_5. Subscript is embedded in the pair $\{e_{02}/e_{01}, h_0/e_{12}\}$, and two data symbols are embedded in the pairs $\{e_{13}/e_{34}, h_3/e_{14}\}$ and $\{e_{45}/e_{25}, h_5/e_{24}\}$. For each of the three triangles, the algorithm first modifies the ratio h_i/e_{ij} by changing only h_i , and then modifies the ratio e_{ij}/e_{kl} while keeping the height h_i constant.

(4) Repeat (1) to (3) until all the data symbols of the message are embedded.

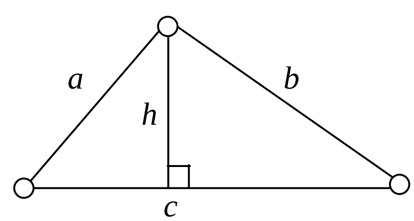

Fig. 1. Examples of dimensionless quantities that define a set of similar triangles.

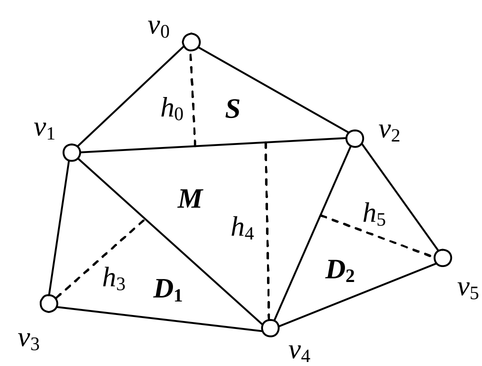

Fig. 2. A macro-embedding-primitive. In the figure, v_i are vertices, e_{ij} are lengths of the edges, and h_i are heights of the triangles.

The *TSQ* extraction algorithm is a public scheme since it does not require the original 3D model for extraction. The quantity pair $\{b/a, h/c\}$ is the key to identify marker triangles. Watermarks embedded by the *TSQ* algorithm remain detectable against translation, rotation, and uniform-scaling transformations of the marked 3D models. However, the watermarks will be destroyed by randomization of coordinates, or by topological alteration such as re-meshing, smoothing, and simplification operations.

Later, Ohbuchi *et al*. [11] proposed a robust watermarking method that adds watermark into a 3D polygonal mesh in the mesh's spectral domain. The algorithm computes spectra of the mesh by using eigenvalue decomposition of a Laplacian matrix derived only from connectivity of the mesh. Mesh spectra can be obtained by projecting coordinates of vertices onto the set of eigenvectors. A watermark is embedded by modifying the magnitude of the spectra. Based on the previous results, Ohbuchi *et al*. [12] proposed a robust watermarking algorithm which embedded watermark into mesh spectral domain with major improvements in computational efficiency and attack resiliency for 3D polygonal meshes.

Benedens [13] presented the fundamentals of geometry based watermarking. He proposed a watermarking algorithm that modifies normal distribution to invisibly store information in the model's geometry. The embedding scheme is robust against mesh simplifications. The highest rate of robustness achieved was resistance to a simplification that reduced the model to 36 percent of its original number of faces. One drawback of the algorithm is the large amount of a priori data needed before watermark retrieval. For private watermarks, this is tolerable.

Praun *et al*. [14] proposed a sophisticated robust mesh watermarking scheme. They first constructed a set of scalar basis functions over the mesh vertices using multi-resolution analysis [15] and then perturbed vertices along the direction of the surface normal weighted by the basis functions. Their watermarking scheme is resistant to common mesh attacks such as translation, rotation, scaling, cropping, smoothing, simplification, and re-sampling operations.

Zafeiriou *et al*. [16] proposed two methods suitable for blind 3D mesh object watermarking applications. Their first method is robust against 3D rotation, translation, and uniform scaling. The second one is robust against both geometric and mesh simplification attacks. Both algorithms are based on principal component analysis, and thus both algorithms will fail against cropping attack in that it can cause severe alteration to the principal object axis.

3. Fragile Watermarking for 3D Models

Fragile watermarking techniques for still images have been widely studied and investigated in recent years [17, 18]. On the other hand, fragile watermarking for 3D models got relatively less notice. There are two major functions in 3D fragile watermarking: integrity checking and changed region locating. Moreover, a good fragile watermarking scheme should be invariant to translation, rotation, and uniformly scaling operations. We think these operations do not change the integrity of the original model and should not be treated as malicious attacks. Possible applications of public fragile watermarking include demonstrating a digital material having not been changed (or having been changed) in an official situation (*e.g.*, in a court) and to confirm the received digital material having not been changed at the receiver end. The functions of the 3D fragile watermarking scheme are similar to that of digital signature. Both schemes can check the originality of the received 3D models. Digital signature can verify the identity of the sender while the 3D fragile watermarking can locate the changed regions. Moreover, for 3D models, the 3D fragile watermarking is superior to the digital signature scheme in its easy implementation and faster computation. A comparison of these two schemes is summarized in Table 1.

Table 1. A comparison of fragile watermarking and digital signature applied in 3D models.

Functions	Digital signature	Fragile watermarking
Verify the *ID* of the sender	Yes	No
Check the integrity of the documents	Yes	Yes
Locate the changed regions	No	Yes
Implementation cost	High	Low
Computation speed	Slow	Fast
Application fields	General	Different methods for different media

Yeo and Yeung [19] are the pioneers of discussing 3D fragile watermarking. They computed two indices for every vertex: the location index and the value index. For each vertex, the location index is calculated by a hash function according to the coordinates of it and its neighboring vertices, while the value index is calculated by another hash function based on the coordinates of it. Then they slightly perturbed every vertex to make these two indices being equal. Unauthorized modifications will be detected in the watermark extraction phase by checking the difference between these two indices. The scheme is both public and fragile, but there are two problems left in their scheme: the causality problem and the convergence problem.

Most 3D fragile watermarking schemes follow the concepts proposed by Yeo and Yeung [19]: slightly perturbing the positions of a subset of vertices on a mesh model to keep them in some predefined relationship with their neighboring vertices. Two problems frequently arise in the embedding stage: the causality problem and the convergence problem. The causality problem arises while the neighboring relationship of a former processed vertex is influenced by the perturbing of its latter processed neighboring vertices. The convergence problem means that the original model has been heavily distorted before some vertices reach the predefined relationship.

The causality problem: The location index of a former processed vertex will be changed by the perturbing of latter processed neighboring vertices. As one example illustrated in Fig. 3, suppose we process vertices in the order v_0, v_1, v_2, …, v_n, we first process and perturb v_0 to make its value index (calculated from its coordinates) being equal to its location index (calculated from the coordinates of it and its neighboring vertices). However, the location index of v_0 will be changed when we process and perturb v_1 to adjust its value index. The same situation occurred when the following vertices are processed. To avoid the causality problem, we need to perturb vertices in a predefined order (v_0, v_1, v_2, …, v_n) and constrain the calculation of the location index for each vertex only to involve the vertices that have been processed. For example, in Fig. 3, the calculation of the location index of v_0 should only involve itself. The calculation of the location index of v_1 should only involve v_0 and v_1. The calculation of the location index of v_2 can only involve v_0, v_1 and v_2, and so on. This constraint can avoid the causality problem. A drawback of this constraint is that the capability for detecting modification will be ruined if the predefined traverse order has been changed or became untraceable. For example, a simple model simplification (vertex decimation) or vertex re-numbering operation may totally disable the capability of the scheme.

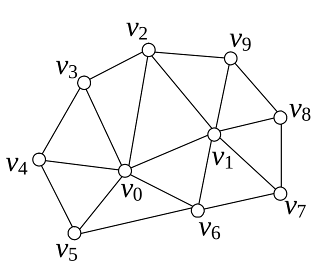

Fig. 3. A causality problem example.

The convergence problem: In the vertex perturbing stage, it requires perturbing the vertices of the whole model to make the value index and the location index of every vertex being equal. In practice, some vertices may need to be perturbed more and more to satisfy the requirement as one example illustrated in Fig. 4. In the figure, v_0 has to be perturbed to a position that heavily distorts the original structure to make its two indices being equal. It is possible that the requirement is met only when the model has been heavily distorted or the requirement can never be met. Another disadvantage comes from the convergence problem is that the user can not control the distortion induced by the perturbing process.

Fornaro and Sanna [20] proposed a public key approach for authentication of constructive solid geometry (*CSG*) models. They computed watermarks from a model followed by a watermark encryption operation and then attached the encrypted watermarks to the solids or comments of the *CSG* models. The advantage of the method is that no modification on the original 3D model is needed and this characteristic is sometimes very important to some “artistic” or “technical” models. The drawback of this scheme is that it can only tell if a model has been modified, but can not locate the modified regions. To locate modifications is an important issue in 3D fragile watermarking.

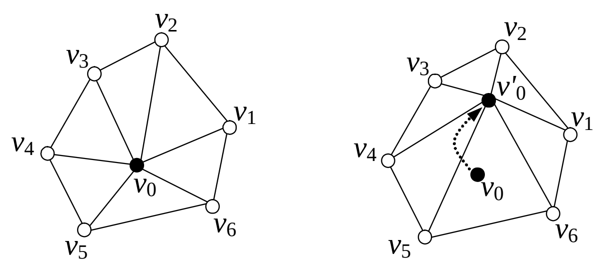

Fig. 4. A causality problem example.

Benedens and Busch [21] proposed a system consisting of three watermarking algorithms: one named vertex flood algorithm (*VFA*) suitable for embedding fragile public readable watermarks with high capacity and offering a way of model authentication; one realizing affine invariant watermarks, named affine invariant embedding (*AIE*); and a third one, named normal bin encoding (*NBE*) algorithm, realizing watermarks with robustness against more complex operations, most noticeably polygon reduction. Their watermarking systems achieved robustness against randomization of vertices, mesh altering (re-meshing), and polygon simplification operations.

Lin *et al*. [22] proposed a scheme similar to Yeo and Yeung’s method [19]. Their method eliminates the causality problem by applying two different hash functions on the vertex coordinates, without considering the neighboring vertices of a vertex. In the embedding stage, they slightly perturbed every vertex making these two hash function values being equal; however, the convergence problem still occurs in this scheme. To avoid heavy distortion due to vertex perturbing, they set a threshold and simply skip the vertices that could not meet the requirement under the threshold. This causes some embedding “holes” which will cause false-alarm in the watermark detection stage.

Wu and Cheung [23] proposed a fragile watermarking scheme for authenticating 3D mesh models. The watermark embedded by the method is invariant to translation, rotation, and uniformly scaling, but sensitive to other operations. The main idea of the method is to keep the ratio between the distance from the mesh center to each surface face and a quantization step remaining the same after the model is translated, rotated, or uniformly scaled. There are two major drawbacks in this scheme. Firstly, it is a semi-public watermarking scheme since the original watermark is needed in the detecting stage to authenticate the watermarked model. In fragile watermarking, a pure public watermarking scheme is preferred since we don't want to pay extra cost for encrypting the original watermark. Secondly, it fails in locating the changed regions since the center position of the mesh will be changed once any vertex has been changed. However, their concept of invariant geometry transformation is worth further studying since the transformation does not affect the integrity of the mesh.

Chou and Tseng [24] proposed a multi-function vertex embedding method and an adjusting-vertex method to overcome the causality and convergence problems. They tried to keep the mark vertices maintaining a predefined relationship with their neighboring vertices. Vertices that have not been changed will satisfy the relationship. On the other hand, vertex that does not satisfy the relationship indicates that at least one of its neighboring vertices or the vertex itself (or both) has been changed. The union of all mark vertices and their neighboring vertices should cover the whole model; thus any change on the model can be detected. Fig. 5 shows how their scheme detects modifications and locates the suspicious modified regions. In Fig. 5 (a), v is a vertex that has been changed, and w_i, $i = 0 \sim 5$, are mark vertices. In Fig. 5 (b), two mark vertices, w_1 and w_4, have detected the change, whereas other four mark vertices don't. Fig. 5 (c) shows that these two mark vertices and their neighboring vertices are set to be suspicious vertices (linked by dashed lines, shown as gray nodes). Then parts of suspicious vertices are released by the undetected mark vertices (changed from gray to non-gray nodes), as shown in Fig. 5 (d). Lastly, the scheme can induce the possible modified vertices: v and/or w_1. For some artistic or technical models, it is very important to control the distortion ratio caused by watermark embedding. Their method can control the average distortion by the keys used in watermark embedding. Their approach is a pure public scheme and can locate the changed regions, but the scheme is not geometry transformation invariant.

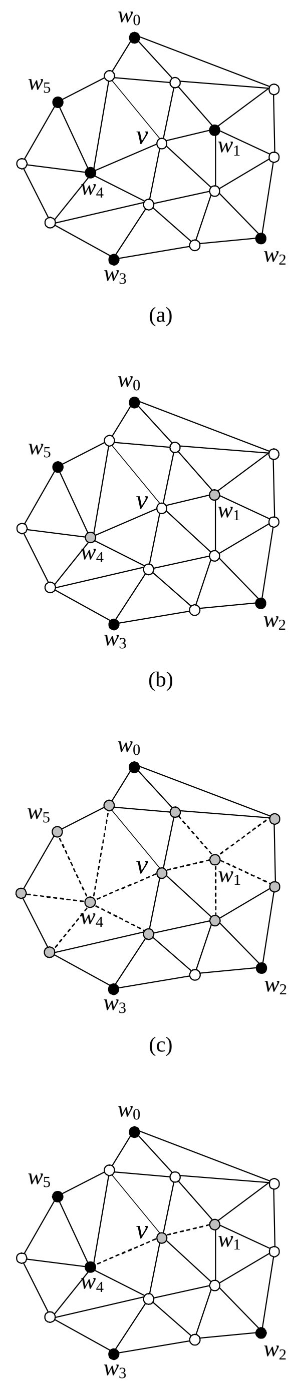

Fig. 5. A mesh with one vertex being changed. (a) v is a vertex that has been changed and w_i, $i = 0 \sim 5$, are mark vertices. (b) Two mark vertices, w_1 and w_4, have detected the change. Black nodes denote the mark vertices, white nodes denote the non-mark vertices, and gray nodes denote the suspicious vertices. (c) These two mark vertices and their neighboring vertices are set to be suspicious vertices (linked by dashed lines). (d) Parts of suspicious vertices are released by the undetected mark vertices (changed from gray to non-gray nodes).

4. Performance Evaluation

For robust watermarking schemes, the embedded watermarks should resist against various malicious attacks. The possible attacks include cropping, smoothing, simplification, noising, re-meshing, vertex re-ordering, translation, rotation, and scaling (uniformly or non-uniformly) operations. A good robust watermarking scheme should resist against as many attacks as possible. For fragile watermarking schemes, the extraction algorithm should be a public scheme, and it should detect and locate the changed regions.

Both robust and fragile watermarking schemes should control the distortion caused by the watermark embedding. Only a few researches discussed about the topic of distortion control. In 3D models, the distance between two surfaces X and Y can be defined by L^2 measurement [25],

$$d(X,Y) = \sqrt{\frac{1}{area(X)} \int_{x \in X} d(x,Y)^2 \, dx} \quad , \qquad (1)$$

where $d(x, Y)$ is the Euclidean distance from a point x on X to the closest point on Y. We modify the definition of L^2 measurement to $d(M, M')$ for representing the average distortion of all vertices in a model,

$$d(M,M') = \frac{1}{|M|} \sum_{i=1}^{|M|} \left| v_i - v_i' \right| \quad , \qquad (2)$$

where M and M' are the original and marked models, respectively. The L^2 measurement should be treated as one measurement of performance evaluation criteria for 3D model watermarking algorithms.

5. Conclusion

In this paper, we performed a survey on current 3D model watermarking techniques by classifying major algorithms into classes, describing main ideas behind each algorithm, and comparing their strength and weakness.

The major function of robust watermarking is for ownership claiming. The design goal of robust watermarking is to make the embedded watermarks remain detectable after being attacked. There three requirements for 3D robust watermark embedding: unobtrusive, robust, and space efficient. Unobtrusive means the embedding must not interfere with the intended use of a model. Robust means the embedded watermarks should remain detectable after being maliciously attacked. Space efficient means an embedding method should be able to embed sufficient amount of information into models. The major function of fragile watermarking is for digital content authentication and verification. The design goal of fragile watermarking is to detect the slightest unauthorized modifications and locate the changed regions. There are two major functions in 3D fragile watermarking: integrity checking and changed region locating.

Future watermark embedding schemes could concentrate on distortion control and the development of performance evaluation criteria. Moreover, future fragile watermarking scheme should be invariant to translation, rotation, and uniformly scaling operations. These operations do not change the integrity of the original model and should not be treated as malicious attacks.

References

[1] U. Maurer, “New approaches to digital evidence,” Proceedings of the IEEE, Vol.92, Issue 6, pp.933-947, 2004.

[2] T. O. Wells, “Electronic and digital signatures: in search of a standard,” IT Professional, Vol.2, Issue 3, pp.24-30, 2000.

[3] T. Beth and D. Gollman, “Algorithm engineering for public key algorithms,” IEEE Journal on Selected Areas in Communications, Vol.7, Issue 4, pp.458-466, 1989.

[4] W. Diffie and M. E. Hellman, “Privacy and authentication: An introduction to cryptography,” Proceedings of the IEEE, Vol.67, Issue 3, pp. 397-427, 1979.

[5] W. Diffie and M. E. Hellman, “New directions in cryptography,” IEEE Trans. Information Theory, Vol.22, Issue 6, pp.644-654, 1976.

[6] L. Xie and G. R. Arce, “A class of authentication digital watermarks for secure multimedia communication,” IEEE Trans. Image Processing, Vol.10, No.11, pp.1754-1764, 2001.

[7] M. Barni, F. Bartolini, and A. Piva, “Improved wavelet-based watermarking through pixel-wise masking,” IEEE Trans. Image Processing, Vol.10, No.5, pp.783-791, 2001.

[8] F. Cayre and B. Macq, “Data hiding on 3-D triangle meshes,” IEEE Trans. Signal Processing, Vol.51, No.4, pp.939-949, 2003.

[9] R. Ohbuchi, H. Masuda, and M. Aono, “Data embedding algorithms for geometrical and non-geometrical targets in three-dimensional polygonal models ,” Computer Communications, Vol.21, Issue 15, pp.1344-1354, 1998.

[10] R. Ohbuchi, H. Masuda, and M. Aono, “Watermarking three-dimensional polygonal models through geometric and topological modifications,” IEEE Journal on Selected Areas in Communication, Vol.16, No.4, pp.551-560, 1998.

[11] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, “Watermarking 3D polygonal meshes in the mesh spectral domain,” in Proc. The Graphics Interface, Ottawa, Ontario, June 7-9, 2001, pp.9-17.

[12] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, “A frequency-domain approach to watermarking 3D shapes,” Computer Graphics Forum, Vol.21, No.3, pp.373-382, 2002.

[13] O. Benedens, “Geometry-based watermarking of 3-D models,” IEEE Computer Graphics and Applications, Vol.19, No.1, pp.46-55, 1999.

[14] E. Praun, H. Hoppe, and A. Finkelstein, “Robust mesh watermarking,” in ACM SIGGRAPH, Los Angeles, CA, Aug. 8-13, 1999, pp.49-56.

[15] H. Hoppe, “Progressive meshes,” in ACM SIGGRAPH, New Orleans, LA, Aug. 4-9, 1996, pp.99-108.

[16] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," IEEE Trans. Visualization and Computer Graphics, Vol.11, Issue 5, pp.596-607, 2005.
[17] M.-S. Hsieh, D.-C. Tseng, and Y.-H. Huang, "Hiding digital watermarks using multiresolution wavelet transform," IEEE Trans. Industrial Electronics, Vol.48, No.5, pp.875-882, 2001.
[18] M.-S. Hsieh and D.-C. Tseng, "Perceptual digital watermarking for image authentication in electronic commerce," Journal of Electronic Commerce Research, Vol.4, pp.157-170, 2004.
[19] B.-L. Yeo and M.-M. Yeung, "Watermarking 3D objects for verification," IEEE Computer Graphics and Applications, Vol.19, No.1, pp.36-45, 1999.
[20] C. Fornaro and A. Sanna, "Public key watermarking for authentication of CSG models," Computer-Aided Design, Vol.32, No.12, pp.727-735, 2000.
[21] O. Benedens and C. Busch, "Towards blind detection of robust watermarks in polygonal models," Computer Graphics Forum, Vol.19, No.3, pp.C199-208, 2000.
[22] H.-Y. Lin, H.-Y. Liao, C.-S. Lu, and J.-C. Lin, "Fragile watermarking for authenticating 3-D polygonal meshes," IEEE Trans. Multimedia, Vol.7, No.6, pp.997-1006, 2005.
[23] H.-T. Wu and Y.-M. Cheung, "A fragile watermarking scheme for 3D meshes," in Proc. The 7th Workshop on Multimedia and Security, New York, Aug. 1-2, 2005, pp.117-124.
[24] C.-M. Chou and D.-C. Tseng, "A public fragile watermarking scheme for 3D model authentication," Computer-Aided Design, Vol.38, No.11, pp.1154-1165, 2006.
[25] A. Khodakovsky, P. Schro¨der, and W. Sweldens, "Progressive geometry compression," in ACM SIGGRAPH, New Orleans, LA, July 23-28, 2000, pp. 271-278.

Chang-Min Chou received his M.S. degree in Computer Engineering and Computer Science from *California State University, Long Beach*, USA, in June 1992. He is currently a Ph.D. student in the Department of Computer Science and Information Engineering at *National Central University*, Chung-li, Taiwan. He has been a faculty in the Department of Electronic Engineering at *Ching Yun University*, Chung-li, Taiwan since 1994. His current research interests include algorithms, image processing, and 3D model processing; especially in the topics of watermarking and progressive compression and transmission for 3D models.

Din-Chang Tseng received his Ph.D. degree in Information Engineering from *National Chiao Tung University*, Hsinchu, Taiwan, in June 1988. He has been a professor in the Department of Computer Science and Information Engineering at *National Central University*, Chung-li, Taiwan since 1996. He is a member of *The IEEE, The Chinese Image Processing and Pattern Recognition Society, and Chinese Geographic Information Association*. His current research interests include computer vision, image processing, and virtual reality; especially in the topics: computer vision system for advanced safety vehicle, wavelet-based image processing, facial expression detection and animation, and view-dependent multiresolution terrain modeling.