

Watermarking MPEG-4 2D Mesh Animation with Time-Series Analysis*

SHIH-HSUAN YANG, CHIN-KUEN LIANG AND CHIN-YUN HSIEH

Department of Computer Science and Information Engineering

National Taipei University of Technology

Taipei, 106 Taiwan

E-mail: shyang@ntut.edu.tw

Former watermarking techniques have been largely developed for application to natural videos. Important applications, including virtual reality, computer games, cartoons, and movies, involve another category of visual data: computer animation. In this study, we developed robust watermarking techniques to protect MPEG-4 2D mesh animation against copyright infringement. Three time-series analysis tools, the discrete cosine transform (DCT), singular spectrum analysis (SSA), and discrete wavelet transform (DWT), are employed to analyze the motion characteristics of mesh animation. The watermark is cast upon the important motion components by employing the spread-spectrum principle. During watermark extraction, a spatial-domain least-squares registration technique is used to restore the distorted mesh. Each watermark bit is then detected by a hard decision with the aid of cryptographically secure keys. We have tested the proposed system against a variety of attacks, including affine transformations, temporal smoothing, spectral enhancement and attenuation, and additive random noise. The time-series analysis tools studied here are evaluated in terms of their robustness to attacks and the required computational complexity.

Keywords: copyright protection, computer animation, discrete cosine transform (DCT), discrete wavelet transform (DWT), MPEG-4, singular spectrum analysis (SSA), time-series analysis, watermarking

1. INTRODUCTION

Digital multimedia have proliferated with the rapid development of the PC and Internet. Digital technology makes possible error-free reproduction and efficient dissemination of multimedia contents. This, in turn, makes it imperative to protect intellectual property rights (IPR). Without adequate IPR protection, the revenues of content creators will be seriously jeopardized. The watermark, an indelible and identifiable code in the host media, has been proposed as an alternative to the conventional encryption mechanism. The ownership of a copyright can be identified by examining the claimed watermark [1]. A watermarking system for copyright protection should satisfy two conflicting requirements: transparency and robustness. A watermark is *transparent* if the

Received May 11, 2004; revised July 20, 2004; accepted September 1, 2004.

Communicated by Ja-Ling Wu.

* This work was supported in part by National Science Council, Taiwan, R.O.C., under Contract No. NSC 90-2213-E-027-012. A preliminary version of this paper was presented in Information Security Conference (ISC2004), 2004.

introduced perceptual degradation is invisible. A watermark is *robust* if reasonable processing of the host media and attacks on the watermark cannot defeat its detection. The added complexity should also be limited, especially for real-time applications.

During the last decade, watermarking has been a very active research area. Various watermarking techniques have been developed for application to natural images, videos, and audio [2]. Besides natural images and videos, computer graphics and animations have been widely applied in virtual reality, computer games, cartoons, and even movies. However, conventional image and video watermarking techniques are inappropriate for graphic data because they have rather different data representations. Natural images and videos are described by the ‘pixels’ located on a rectangular grid, while synthetic graphics are characterized by a ‘polygon model’ composed of geometry, topology, and texture information. Watermarking of polygon models was pioneered by Ohbuchi *et al.* [3], who proposed to imprint watermarks on geometric or topological primitives. Benedens [4] proposed a robust watermark imposed on the surface normals of a polygon model. Praun *et al.* [5] developed a general spread-spectrum mesh watermarking method based on the concept of the progressive mesh [6]. In [7, 8], a spectral analysis derived from vertex connectivity of 3D polygonal meshes was employed to modulate the resulting spectral coefficients. Yin *et al.* [9] proposed to use the Burt-Adelson pyramid to obtain a multiresolution representation of a mesh and embed watermark information in the coefficients of a suitable scale. Murotani and Sugihara [10] employed the singular spectrum analysis to extract spatial frequencies from a vertex series, and a watermark was cast upon the singular values.

Consider a role-playing video game simulating an NBA all-star basketball game. In this case, the most valuable information would be the motion of an animated model (e.g. Michael Jordan’s air stride) rather than the static model. However, the aforementioned techniques protect only the static polygon models, and few researches have been conducted on the watermarking of graphics animation. Kim *et al.* [11] proposed a temporal-domain multiresolution watermarking scheme for articulated figures. Each frame of an articulated figure consists of several body segments specified by position and orientation information, and its motion is given by three-dimensional rigid transformations. The watermark is placed on the displacement vector through multiresolution analysis of rigid transformations. Yang *et al.* [12] realized a robust digital watermarking scheme for MPEG-4 2D mesh animation. The watermark is spread over salient feature motions that are obtained from multiresolution analysis of the MPEG-4 2D mesh. The system was evaluated against some common types of processing of mesh animation.

This paper extends the framework in [12] to explore more time-series analysis (transformation) tools for MPEG-4 2D mesh watermarking. In addition to the prestige of the MPEG standards, the mesh is chosen as the watermarking target because it is regarded as the “lowest common denominator” of surface representations [5]. The proposed method can be generalized to any mesh-based animated graphics. Watermarking in the transform-domain is generally preferable to its original-domain counterpart since an appropriate transformation manifests the perceptual characteristics of video and animation. Furthermore, the spread spectrum principle can be naturally integrated into transform-domain methods [13, 14]. Three mainstream transformation tools, the discrete cosine transform (DCT), discrete wavelet transform (DWT), and singular spectrum analysis (SSA), are evaluated here. These transformations are used to identify significant spectral

components of the mesh movement upon which the watermark is placed. We use a mesh registration technique to combat geometric distortions. Each watermark bit is detected by a hard decision with respect to the undisturbed coefficient, and the retrieved watermark is verified by means of a similarity test. The rest of the paper is organized as follows. In section 2, we review the MPEG-4 2D dynamic mesh and the employed time-series analysis tools. Section 3 introduces the proposed watermarking system. Section 4 presents simulation results and analyses. Some concluding remarks are given at the last section.

2. TIME-SERIES ANALYSIS

2.1 MPEG-4 2D Dynamic Mesh Coding

The MPEG-4 standard [15] encompasses object-based synthetic animation as opposed to the former MPEG-1 and MPEG-2 coding standards that are strictly restricted to frame-based natural video. Because of its versatility, the MPEG-4 standard defines a number of tools, object types, and profiles that can be tailored to a target application. For example, the MPEG-4 animated 2D mesh object type contains the coding tools for representing 2D animation. One frame of an MPEG-4 animated 2D mesh is shown in Fig. 1, which consists of three objects: the 2D mesh, the still texture of the fish, and the attached ‘MPEG-4’ banner. Within a visual sequence that contains multiple visual object planes, the texture and banner objects need to be coded only once. The mesh animation, such as a swimming fish, can be emulated by the motion of the mesh vertexes (called a 2D dynamic mesh), with the texture warped onto the mesh.

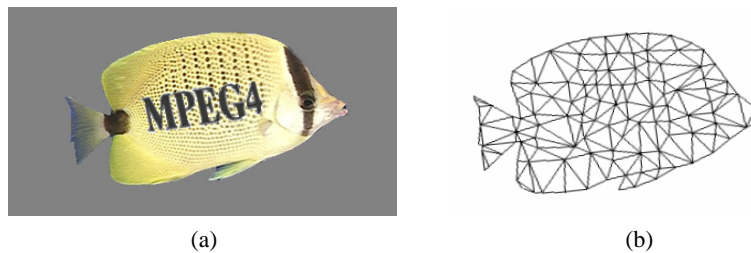


Fig. 1. (a) A frame of an MPEG-4 animated 2D mesh; (b) The associated MOP.

An MPEG-4 2D dynamic mesh consists of a temporal sequence of triangular meshes that tessellate a 2D visual object plane into triangular patches. All the mesh object planes (MOPs) of a mesh object have the same topology with time-varying node point locations. The geometry of the first MOP (I-MOP) is fully coded. The subsequent predictive-coded MOPs (P-MOPs) record only the motion vectors of the node points relative to the previous MOP. All the node points and motion vectors are specified in half-pixel resolution. In our approach, we extract the feature motion of MPEG-4 2D dynamic meshes with the time-series analysis tools explained below.

2.2 Discrete Cosine Transform (DCT)

The time-series analysis tool, also called transformation, plays an essential role in signal processing. The transform coefficients manifest themselves by means of temporal or spectral characteristics that facilitate the divide-and-conquer strategy. Desired properties of transformation include orthogonality, decorrelation, energy compaction, and the existence of fast algorithms. Restricting transformations to be linear and orthogonal simplifies the design and implementation of a transformation-based system.

Linear orthogonal transformation can be viewed as a linear projection of a signal onto an orthogonal basis. Among the various eligible orthogonal basis functions, DCT is widely adopted owing to its excellent energy-compaction capability. For highly correlated data (with a correlation coefficient $\rho > 0.7$), DCT approaches the performance of the Karhunen-Loeve transform (KLT), the theoretically optimal but computationally intractable transformation. Additionally, fast algorithms similar to fast Fourier transform (FFT) exist for DCT [16]. An N -point DCT of an input signal $\mathbf{x} = \{x_0, x_1, \dots, x_{N-1}\}$ is computed as follows:

$$y_n = \sqrt{\frac{2}{N}} e(n) \sum_{k=0}^{N-1} x_k \cos \frac{(2k+1)n\pi}{2N}, \quad n = 0, 1, \dots, N-1, \quad (1)$$

where $\mathbf{y} = \{y_0, y_1, \dots, y_{N-1}\}$ is the corresponding DCT coefficients and

$$e(n) = \begin{cases} 1, & \text{if } n \neq 0 \\ 1/\sqrt{2}, & \text{if } n = 0 \end{cases}. \quad (2)$$

The first DCT coefficient, $y_0 = \sqrt{\frac{1}{N}} \sum_{k=0}^{N-1} x_k$, represents a scaled average value of the input signal and, thus, is often termed the DC value. Since the cosine functions of different harmonics are orthogonal to each other, the inverse DCT is similarly obtained as follows:

$$x_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} e(n) y_n \cos \frac{(2k+1)n\pi}{2N}, \quad k = 0, 1, \dots, N-1. \quad (3)$$

2.3 Singular Spectrum Analysis (SSA)

Singular spectrum analysis (SSA) is a relatively novel time-series analysis tool [17]. We construct an $M \times N$ augmented data matrix \mathbf{A} from a sequence $\mathbf{x} = \{x_0, x_1, x_2, \dots\}$ as follows:

$$\mathbf{A} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{N-1} \\ x_1 & x_2 & \cdots & x_N \\ \vdots & \vdots & \ddots & \vdots \\ x_{M-1} & x_M & \cdots & x_{M+N-2} \end{bmatrix}. \quad (4)$$

The matrix \mathbf{A} is a Hankel matrix that contains lagged copies of \mathbf{x} . A singular value decomposition (SVD) on \mathbf{A} yields

$$\mathbf{A} = \mathbf{U} \times \mathbf{S} \times \mathbf{V}^T, \quad (5)$$

where $\mathbf{U} = [\mathbf{u}_0 \ \mathbf{u}_1 \ \dots \ \mathbf{u}_{M-1}]_{M \times M}$ and $\mathbf{V} = [\mathbf{v}_0 \ \mathbf{v}_1 \ \dots \ \mathbf{v}_{N-1}]_{N \times N}$ are orthogonal matrices. \mathbf{S} is the $M \times N$ singular-value matrix with d nonzero diagonal entries. These nonzero (and positive) entries, s_0, s_1, \dots, s_{d-1} , called singular values, are arranged in a nonincreasing order. The augmented data matrix \mathbf{A} can, therefore, be expressed in the spectral decomposition form

$$\mathbf{A} = \sum_{i=0}^{d-1} s_i \mathbf{u}_i \mathbf{v}_i^T = s_0 \mathbf{A}_0 + s_1 \mathbf{A}_1 + \dots + s_{d-1} \mathbf{A}_{d-1}, \quad (6)$$

where $\mathbf{A}_i = \mathbf{u}_i \mathbf{v}_i^T$ is the i^{th} basis matrix. SSA is signal dependent; it produces different bases for every data segment of length $M + N - 1$. Since \mathbf{U} and \mathbf{V} are orthogonal matrices, the singular values can be recovered from a distorted version of \mathbf{A} , denoted as $\tilde{\mathbf{A}}$, as follows:

$$\tilde{\mathbf{S}} = \mathbf{U}^T \times \tilde{\mathbf{A}} \times \mathbf{V}. \quad (7)$$

Compared to a fixed block transform such as DCT, SSA is more adaptive to signal's nature but requires more computations. Furthermore, the decomposition matrices \mathbf{U} and \mathbf{V} need to be stored for the reconstruction of singular values.

2.4 Discrete Wavelet Transform (DWT)

DCT and other fixed block transformations can capture only the stationary features of signals. SSA achieves adaptiveness with time-varying bases and, thus, exhibits considerably higher complexity. To resolve this dilemma, wavelet analysis reveals both the transient and stationary characteristics of a signal under a multiresolution framework [18]. The filter-bank structure of DWT is shown in Fig. 2. The two filters $h_0[n]$ (lowpass) and $h_1[n]$ (highpass) form a quadrature-mirror filter (QMF) pair. One-level decomposition generates two subbands, L1 (lowpass) and H1 (highpass). Subsequent decompositions are performed along the lowpass subbands; thus, an r -level decomposition generates $r + 1$ subbands, H1, H2, ..., H- r , and L- r . The L- r subband stands for the base (approximation) component while the H subbands stand for details at various scales. The signal can be perfectly synthesized from the constituent subbands through upsampling and interpolative filtering using the synthesis filters $g_0[n]$ and $g_1[n]$.

Desirable properties of wavelet filters include good time-frequency localization, orthogonality, regularity, symmetry, and low implementation complexity. In this paper, we consider six popular wavelet bases: Lazy, Haar (D2), orthogonal D4 and D8, and biorthogonal 5/3 and 9/7 wavelets. The analysis filters $h_0[n]$ and $h_1[n]$ of these wavelets are listed in Table 1. The Lazy wavelet basically performs only even-odd sample separation. The Haar wavelet simply takes the sum (for lowpass filtering) and difference (for

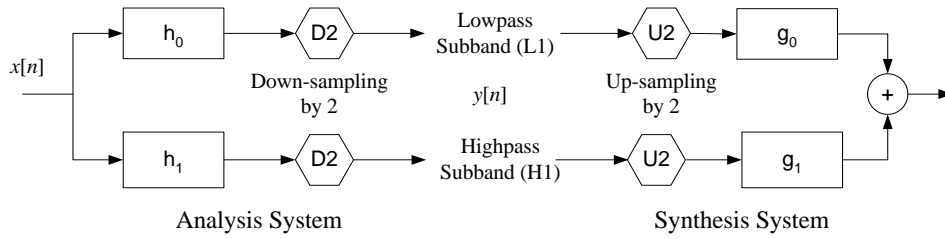


Fig. 2. Filter bank structure of wavelet transform.

Table 1. Analysis filters of the wavelets under study. The negative indexes apply only to biorthogonal 5/3 and 9/7 wavelets.

Filter Index		0	1 (-1)	2 (-2)	3 (-3)	4 (-4)	5	6	7
Lazy	$h_0[n]$	1							
	$h_1[n]$	1							
Haar	$h_0[n]$	$1/\sqrt{2}$	$1/\sqrt{2}$						
	$h_1[n]$	$1/\sqrt{2}$	$-1/\sqrt{2}$						
D4	$h_0[n]$	0.48296	0.83652	0.22414	-0.12941				
	$h_1[n]$	0.12941	0.22414	-0.83652	0.48296				
D8	$h_0[n]$	0.23038	0.71485	0.63088	-0.02798	-0.18703	0.03084	0.03288	-0.01060
	$h_1[n]$	0.01060	0.03288	-0.03084	-0.18703	0.02798	0.63088	-0.71485	0.23038
5/3	$h_0[n]$	$6\sqrt{2}/8$	$2\sqrt{2}/8$	$-\sqrt{2}/8$					
	$h_1[n]$	$1/\sqrt{2}$	$-1/2\sqrt{2}$						
9/7	$h_0[n]$	0.85267	0.37740	-0.11062	-0.02385	0.03783			
	$h_1[n]$	0.78849	-0.41809	-0.04069	0.06454				

highpass filtering). D4 and D8 are the Daubechies orthonormal wavelets with compact support and the maximal number of vanishing moments [19]. The 5/3 and 9/7 biorthogonal filters are the most compression-efficient DWT kernels and have been adopted in the JPEG-2000 standard [20].

3. WATERMARKING PROCESS

3.1 Time-Series Analysis of MPEG-4 2D Dynamic Mesh

Consider an MPEG-4 2D dynamic mesh that consists of m MOPs with n vertexes on each MOP. The node points corresponding to the same vertex across MOPs are gathered to produce a temporal sequence. The i^{th} vertex sequence ($0 \leq i \leq n-1$) is denoted as

$$\mathbf{p}^{(i)} = \{(p_{j,h}^{(i)}, p_{j,v}^{(i)}) : j = 0, 1, \dots, m-1\},$$

where the subscript j is the MOP index, while h and v stand for the horizontal and vertical components, respectively. The employed transformation is independently performed on each of the horizontal and vertical vertex sequences. The proposed watermark resides

in the DC component for DCT, the largest singular value for SSA, and the approximation subband for DWT. All these watermark-bearing coefficients are generally called *important components* in this paper, since they occupy an overwhelming majority of the signal's energy.

In the case of DCT, the vertex sequence is divided into blocks of length N ($N = 8$ in this paper). Hence, $\left\lfloor \frac{m}{8} \right\rfloor$ segments are obtained for each vertex sequence ($\lfloor \cdot \rfloor$ denotes the floor function). For a vertex segment

$$\mathbf{x} = \{(x_{0,h}, x_{0,v}), (x_{1,h}, x_{1,v}), (x_{N-1,h}, x_{N-1,v})\},$$

DCT generates the spectral coefficients

$$\mathbf{y} = \{(y_{0,h}, y_{0,v}), (y_{1,h}, y_{1,v}), (y_{N-1,h}, y_{N-1,v})\},$$

where $y_{0,h} = \sqrt{\frac{1}{N}} \sum_{k=0}^{N-1} x_{k,h}$ and $y_{0,v} = \sqrt{\frac{1}{N}} \sum_{k=0}^{N-1} x_{k,v}$ are the DC components. To give a simple concrete example to illustrate the basic idea, consider the vertex segment

$$\mathbf{x} = \{(60, 40), (66, 44), (70, 48), (72, 44), (74, 40), (72, 36), (72, 32), (70, 36)\}.$$

The resulting DCT coefficients would be

$$\mathbf{y} = \{(196.58, 113.14), (-8.15, 10.67), (-8.16, -5.23), (-20.4, -6.50), (-1.41, 0), (-0.86, -1.94), (-1.21, 2.16), (0.84, 0.08)\}.$$

The watermark is to be embedded in the DC component 196.58 or 113.14.

SSA is realized through time-shift segments of the vertex sequence. In this paper, we choose $M = N = 8$, which establishes a transformation window of length $M + N - 1 = 15$. Hence, $\left\lfloor \frac{m}{15} \right\rfloor$ non-overlapping segments are obtained for each vertex sequence. Continue with the previous example and take $M = N = 4$ for simplicity. The resulting augmented data matrices (for horizontal and vertical components, respectively) are

$$\mathbf{A}_h = \begin{bmatrix} 60 & 66 & 70 & 72 \\ 66 & 70 & 72 & 74 \\ 70 & 72 & 74 & 72 \\ 72 & 74 & 72 & 72 \end{bmatrix} \text{ and } \mathbf{A}_v = \begin{bmatrix} 40 & 44 & 48 & 44 \\ 44 & 48 & 44 & 40 \\ 48 & 44 & 40 & 36 \\ 44 & 40 & 36 & 32 \end{bmatrix}.$$

The singular value decomposition generates the singular values $\{282.26, 7.49, 1.74, 0.50\}$ for \mathbf{A}_h and $\{168.55, 11.22, 2.85, 0.19\}$ for \mathbf{A}_v . The largest singular values 282.26 and 168.55 are subject to watermarking.

In the case of DWT, an r -level decomposition generates $(r + 1)$ subbands. For purposes of fairness, we take $r = 3$ in this paper. Proceed with the previous example with the

Haar transform. The first-level decomposition generates two subbands:

$$\{\text{L1 : H1}\} = \frac{1}{\sqrt{2}} \{(126, 84), (142, 92), (146, 76), (142, 68) : (-6, -4), (-2, 4), (2, 4), (2, -4)\}.$$

The second-level decomposition transforms the L1 subband into

$$\{\text{L2 : H2}\} = \{(134, 88), (144, 72) : (-8, -4), (2, 4)\},$$

and the third-level decomposition transforms the L2 subband into

$$\{\text{L3 : H3}\} = \frac{1}{\sqrt{2}} \{(278, 160) : (-10, 16)\}.$$

The approximation coefficients $\frac{1}{\sqrt{2}}(278, 160) = (196.58, 113.14)$ are eligible for subsequent watermarking.

An objective distortion measure is required for the design of a watermarking system. In this paper, we use the mean square motion difference (MSMD) to characterize the animation difference between two dynamic meshes [12]. Denote the mesh motion vector as $\mathbf{s}_j^{(i)} = (s_{j,h}^{(i)}, s_{j,v}^{(i)})$, where

$$(s_{j,h}^{(i)}, s_{j,v}^{(i)}) = (p_{j+1,h}^{(i)} - p_{j,h}^{(i)}, p_{j+1,v}^{(i)} - p_{j,v}^{(i)}). \quad (8)$$

The MSMD between two dynamic meshes is calculated as follows:

$$\text{MSMD} = \frac{1}{n(m-1)} \sum_{i=0}^{n-1} \sum_{j=0}^{m-2} [(s_{j,h}^{(i)} - t_{j,h}^{(i)})^2 + (s_{j,v}^{(i)} - t_{j,v}^{(i)})^2], \quad (9)$$

where $\{\mathbf{s}_j^{(i)}\}$ and $\{\mathbf{t}_j^{(i)}\}$ are the corresponding mesh motion vectors under comparison.

3.2 Watermark Embedding

Watermarks should be placed upon perceptually significant components to ensure sufficient robustness [13]. High-frequency or small-scale components pertaining to local and intricate motion details are, thus, inadequate for labeling. On the other hand, most researches reported in the literature excluded the DC components and approximation subband to avoid visual artifacts. This restriction is not firm as conjectured; it has been discovered that the DC or approximation coefficients possess a larger watermarking capacity than the other coefficients [21]. Different from [12], where the watermark was embedded in the largest-scale detail subband, we directly distribute the watermark over the important components (DC coefficients, largest singular values, or approximation coefficients). These important components of graphics animation correspond to long-term movement of a vertex sequence. Since the average value of the corresponding basis functions is generally nonzero, labeling the important components results in a dis-

placement for a segment of node points. As will be shown in the next section, this strategy increases robustness without sacrificing transparency.

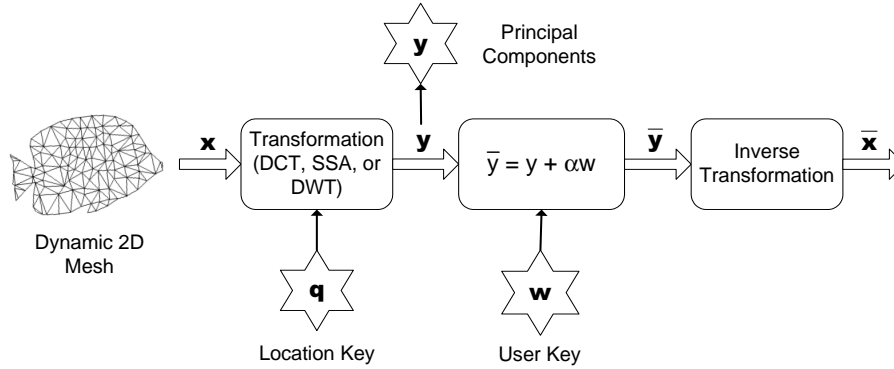


Fig. 3. Watermark embedding process.

The proposed watermark embedding process is depicted in Fig. 3. A secret key associated with a user's identification produces a K -bit watermark sequence $\mathbf{w} = (w_0, w_1, \dots, w_{K-1})$, $w_j \in \{1, -1\}$. We choose \mathbf{w} as an M-sequence, which possesses good correlation properties. Another secret key determines the watermarking locations $\mathbf{q} = (q_0, q_1, \dots, q_{K-1})$ among the eligible coefficients. Depending on the polarity of w_j , a positive or negative value is added to one directional component (horizontal or vertical, chosen at random and recorded in \mathbf{q}) of the eligible coefficient located at q_j , denoted as y_{q_j} . More specifically, y_{q_j} is modulated to be \bar{y}_{q_j} as

$$\bar{y}_{q_j} = y_{q_j} + \alpha w_j. \quad (10)$$

The parameter α ($\alpha > 0$) denotes the watermark strength, and its value determines the tradeoff between transparency and robustness. Proceed with the DCT example in the last section. Suppose that $\alpha = 1$, $w_0 = -1$, and q_0 determines the DC coefficient, 196.58, to be marked. This coefficient is modified as follows:

$$\bar{y}_{q_0} = 196.58 - \alpha = 195.58.$$

Taking the corresponding inverse transformation for DCT and DWT to restore the 2D dynamic mesh completes the watermarking process. In the case of SSA, the watermarked sequence $\bar{\mathbf{x}}$ is obtained by averaging the off-diagonal entries of \mathbf{A} in Eq. (4) [10]. To facilitate later watermark verification, the content holder should securely store the cryptographic keys that generate \mathbf{q} and \mathbf{w} , and the labeled important components \mathbf{y} .

3.3 Watermark Extraction and Verification

The watermark extraction and verification process is depicted in Fig. 4. Although geometric transformations (translation, rotation, and scaling) hardly affect the perceptual

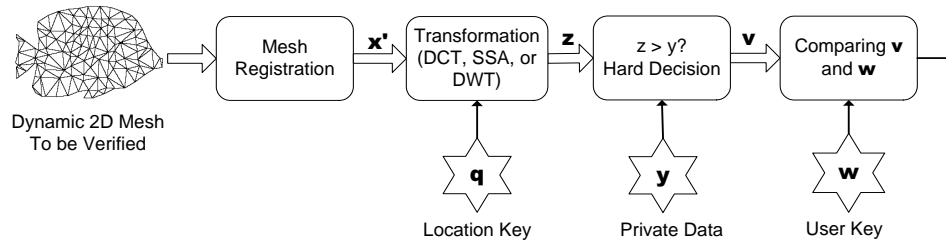


Fig. 4. Watermark verification process.

quality of an image, they defeat many watermarking schemes by breaking the synchronization between the watermarked and original data. In this paper, we employ the mesh registration technique [22] to solve this problem. Assume that an affine linear transformation model $T : (h, v) \rightarrow (h', v')$ is specified in the homogeneous coordinates as follows:

$$\begin{bmatrix} h' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma \cos \theta & -\gamma \sin \theta & t_h \\ \gamma \sin \theta & \gamma \cos \theta & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h \\ v \\ 1 \end{bmatrix}, \quad (11)$$

where the node points (h, v) and (h', v') are drawn from the original unwatermarked I-MOP and the test I-MOP, respectively. The four parameters γ (scaling factor), θ (rotation angle), t_h , and t_v (displacements) can be obtained from the least-squares approximation. Depending on whether the original unwatermarked media are required for detection, a watermark detector is referred to as informed or blind. The proposed method is *semi-blind* in the sense that only partial information of the original mesh sequence (I-MOP and the important components to be labeled) is needed for watermark detection. Under these circumstances, the content owner or an authorized third party can verify copyright infringement with reasonable complexity.

After mesh registration is performed, the dynamic mesh under suspicion takes transformation as in watermark embedding. SSA is the exception; it restores the singular values using Eq. (7) with additional decomposition matrices \mathbf{U} and \mathbf{V} . Denote by z_{q_j} the watermark-bearing coefficient at location q_j . The watermark bit v_j is extracted according to the following hard-decision rule:

$$v_j = \begin{cases} 1, & \text{if } z_{q_j} > y_{q_j} \\ -1, & \text{if } z_{q_j} \leq y_{q_j} \end{cases}. \quad (12)$$

Continue with the numerical example in the previous section. If the extracted coefficient at the watermarking positions satisfies

$$z_{q_0} < 196.58,$$

then the watermark bit w_0 is correctly deciphered. The similarity between the K -bit pat-

tern $\mathbf{v} = (v_0, v_1, \dots, v_{K-1})$ and the original watermark \mathbf{w} is measured based on the bit error rate (BER):

$$\text{BER} = \frac{|\{v_j : v_j \neq w_j\}|}{K}, \quad (13)$$

where the numerator denotes the number of bits in which \mathbf{v} and \mathbf{w} differ. An appropriate threshold for BER should be determined for reliable watermark verification.

4. SIMULATION RESULTS

Evaluation of the proposed watermarking framework was conducted based on the following four MPEG-4 2D dynamic meshes:

1. Mesh12 (Fig. 5). A fish dilates and rotates clockwise with constant angular velocity. The MOP is of the Delaunay type with 120 node points and 187 triangles.
2. Bream (Fig. 6). This simulates a swimming fish turning from the side view to the front view. The MOP is of the Delaunay type with 165 node points and 270 triangles.
3. Akiyo (Fig. 7). A natural video of a news report is tessellated to obtain a Delaunay mesh composed of 210 node points and 375 triangles. The talking (and nodding) head demonstrates evident movement around the mouse.
4. Mesh06 (Fig. 8). This is a rectangular uniform mesh with 120 node points and 198 triangles. The mesh dilates and rotates clockwise, and then shrinks and rotates counterclockwise. This process repeats itself with accelerated velocity.

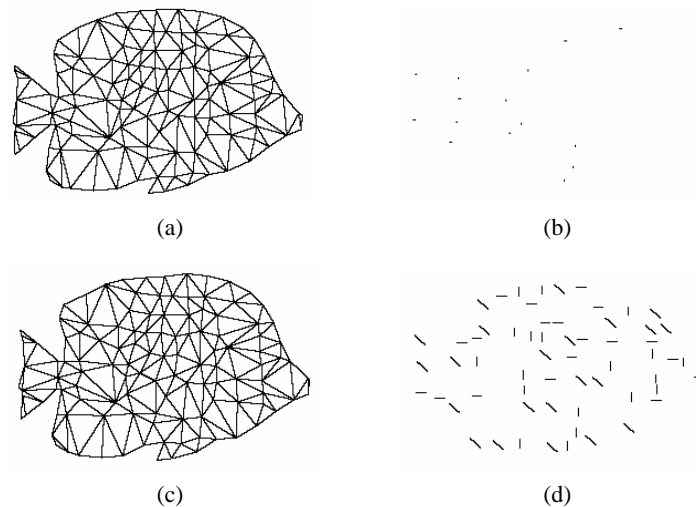


Fig. 5. (a) A watermarked MOP of Mesh12 (28th MOP, SSA, MSMD = 0.005); (b) Difference between the watermarked and original meshes; (c) Watermarked mesh attacked by random pixel noise with strength = 2 (MSMD = 3.5622); (d) Difference between the attacked and watermarked meshes. The difference is magnified by a factor of 5 to obtain a clearer view.

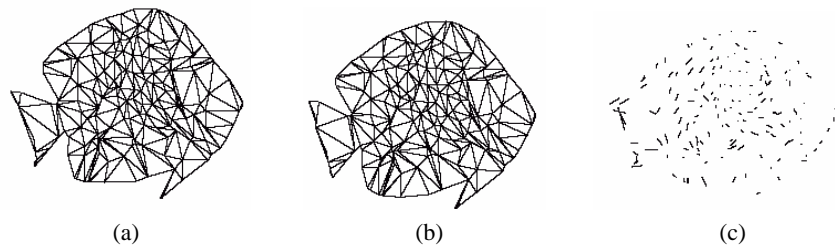


Fig. 6. (a) A watermarked MOP of Bream (MSMD = 0.005); (b) Watermarked mesh attacked by temporal smoothing S5 (MSMD = 3.8437); (c) Difference between the attacked and watermarked meshes. The difference is magnified by a factor of 5 to obtain a clearer view.

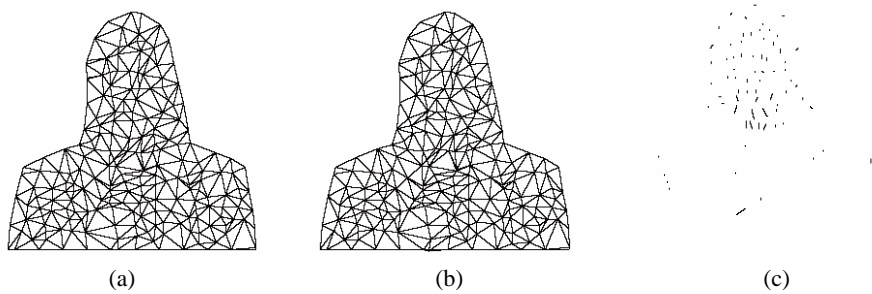


Fig. 7. (a) A watermarked MOP of Akiyo (MSMD = 0.005); (b) Watermarked mesh attacked by spectral enhancement (MSMD = 0.5733); (c) Difference between the attacked and watermarked meshes. The difference is magnified by a factor of 5 to obtain a clearer view.

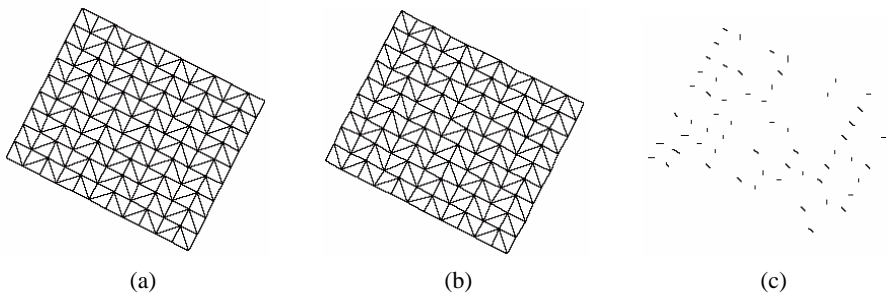


Fig. 8. (a) A watermarked MOP of Mesh06 (MSMD = 0.005); (b) Watermarked mesh attacked by random pixel noise with strength = 1 (MSMD = 0.9873); (c) Difference between the attacked and watermarked meshes. The difference is magnified by a factor of 5 to obtain a clearer view.

We adapted all the test meshes to 128 MOPs, and a binary signature 127 bits long ($K = 127$) was embedded in each mesh. To make a fair comparison, we adjusted the watermark strength α so that all the watermarked meshes with any of the described transformations had the same MSMD ($= 0.005$) as the original mesh. Note that the introduced distortion watermark was unnoticeable, as is shown in Fig. 5 (b), which plots the enlarged difference of node points between the watermarked and original meshes.

The proposed system is evaluated here against the following attacks commonly encountered in graphics animation. Some of them were first introduced in [11].

- Affine linear transformations. Two cases will be examined.
 - (a) Scaling by a factor of 1.3, followed by a translation of (12, 0), followed by a rotation of 15 degrees.
 - (b) Scaling by a factor of 0.7, followed by a translation of (− 11, − 9), followed by a rotation of 65 degrees.

The mesh registration technique introduced in section 3.3 can restore most of these transformations. However, the embedded watermark deteriorates due to inherent quantization noise, especially when shrinking (by a scaling factor < 1) is involved.
- Temporal smoothing. Lowpass filtering of each of the vertex sequences with two different impulse responses, S3: {1/4, 1/2, 1/4} and S5: {1/5, 1/5, 1/5, 1/5, 1/5}. One example of the S5 operation applied to the Bream mesh is shown in Fig. 6.
- Spectral enhancement (SE) and spectral attenuation (SA). The 3-level DWT is adopted with the 9/7 filter, and the H1 and H2 subbands are multiplied by factors of 2 and 0.6, respectively.
- Random noise. 50% of the node points are subjected to additive noise, in the horizontal or vertical directions, or both. Two levels of noise strength S , 1 pixel ($S = 1$) and 2 pixels ($S = 2$), will be examined. One such resulting MOP with $S = 2$ for Mesh12 is shown in Fig. 5. Noise strength equal to 2 or a larger value introduces intolerable visual degradation. Another example with $S = 1$ is shown in Fig. 8. Random noise can be regarded as short-term jamming added to the dynamic mesh.

We summarize the BER performance against the attacks in Tables 2 to 5. The benefits of each time-series analysis tool can be deduced by comparing it with the Lazy wavelet, which basically results in no transformation. All the nontrivial transformations demonstrate notable enhancement in robustness. Among the four test meshes, Mesh06 is the most difficult to protect because its motion contains more high-frequency components. When compared with [12], where watermarking information was embedded in detail subbands, the proposed approximation subband watermarking shows significantly better robustness. In Table 6, we list the number of false-negative detections out of the 32 attacks with a BER identification threshold equal to 0.20. This threshold ensures excellent false-negative performance for SSA, Haar, 5/3, and 9/7 wavelets (with approximation subband embedding). The corresponding false-positive probability of random watermarks is

$$\sum_{i=0}^{25} \binom{127}{i} \left(\frac{1}{2}\right)^{127} = 1.56 \times 10^{-12}. \quad (14)$$

(Since $26/127 = 0.205 > 0.2$, at most 25 out of 127 bits can have different degrees of parity for a fake watermark.) This identification threshold (0.20) is, therefore, adequate to avoid harming an innocent user.

To investigate the computational efficiency of the proposed watermarking method, we measured the average processing time required for watermark embedding and

Table 2. BER results for Mesh12. The second value for DWT corresponds to the results of watermarking in the detail subband [12]. (MSMD: S3 = 0.0041, S5 = 0.0074, SE = 0.0411, SA = 0.0060, RN1 = 0.8931, RN2 = 3.5503)

Attacks	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Affine (a)	0	0	0.079 0.110	0.008 0	0	0	0 0.008	0.008 0.008
Affine (b)	0.094	0.102	0.189 0.079	0.039 0.024	0.047 0.102	0.031 0.079	0.055 0.047	0.055 0.031
S3	0	0	0.362 0.386	0	0	0	0	0
S5	0	0	0.409 0.425	0 0.008	0 0.094	0 0.008	0 0.008	0
Spectral Enhancement	0	0	0	0	0	0	0	0
Spectral Attenuation	0	0	0	0	0	0	0	0
Random Noise S = 1 (RN1)	0.008	0	0.157 0.165	0 0.071	0.016 0.165	0.008 0.031	0.016 0.102	0.024 0.079
Random Noise S = 2 (RN2)	0.150	0.055	0.134 0.236	0.142 0.291	0.024 0.386	0.071 0.220	0.063 0.299	0.126 0.244

Table 3. BER results for Bream. The second value for DWT corresponds to the results of watermarking in the detail subband [12]. (MSMD: S3 = 1.8523, S5 = 3.8492, SE = 3.8598, SA = 0.6615, RN1 = 0.8912, RN2 = 3.5680)

Attacks	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Affine (a)	0	0	0.024 0.016	0	0	0	0 0.008	0
Affine (b)	0.047	0.024	0.118 0.134	0.024 0.016	0	0.008 0	0 0.024	0 0.016
S3	0	0	0.315 0.331	0 0.031	0 0.024	0 0.024	0 0.063	0 0.047
S5	0.039	0	0.346 0.409	0.008 0.244	0 0.315	0 0.173	0.008 0.378	0.008 0.331
Spectral Enhancement	0.134	0.173	0.150 0.173	0.181 0.134	0.126 0.150	0.134 0.150	0.039	0.102 0
Spectral Attenuation	0.157	0.039	0.189 0.213	0.047 0.079	0.008 0.110	0 0.063	0	0
Random Noise S = 1 (RN1)	0	0	0.157 0.150	0 0.047	0.008 0.071	0 0.024	0 0.079	0 0.118
Random Noise S = 2 (RN2)	0.150	0.055	0.197 0.118	0.047 0.197	0.071 0.165	0.047 0.173	0.008 0.299	0.008 0.291

Table 4. BER results for Akiyo. The second value for DWT corresponds to the results of watermarking in the detail subband [12]. (MSMD: S3 = 0.2094, S5 = 0.5397, SE = 0.5733, SA = 0.1003, RN1 = 0.8908, RN2 = 3.5710.

Attacks	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Affine (a)	0.016	0	0.047 0.063	0.039 0.016	0.016 0.008	0.008 0	0.008 0.016	0.008 0
Affine (b)	0.024	0.094	0.087 0.087	0.016 0.079	0.008 0.016	0.024 0.016	0.008 0.039	0.008 0.008
S3	0	0	0.394 0.409	0 0.008	0 0.008	0 0.008	0	0
S5	0	0	0.402 0.425	0 0.008	0.008 0.134	0 0.071	0 0.047	0 0.024
Spectral Enhancement	0	0	0.008 0.016	0 0.047	0 0.039	0 0.024	0	0
Spectral Attenuation	0	0	0.031 0.008	0 0.024	0 0.024	0 0.016	0	0
Random Noise S = 1 (RN1)	0	0	0.150 0.157	0 0.031	0 0.031	0 0.024	0 0.102	0 0.008
Random Noise S = 2 (RN2)	0.087	0.071	0.181 0.244	0.039 0.189	0.055 0.291	0.024 0.134	0 0.134	0.024 0.126

Table 5. BER results for Mesh06. The second value for DWT corresponds to the results of watermarking in the detail subband [12]. (MSMD: S3 = 0.2649, S5 = 0.8330, SE = 1.6016, SA = 0.2836, RN1 = 0.9873, RN2 = 2.1040)

Attacks	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Affine (a)	0.079	0.071	0.488 0.465	0.063 0.236	0.370 0.386	0.386 0.252	0.047 0.118	0.047 0.126
Affine (b)	0.220	0.197	0.402 0.394	0.165 0.260	0.528 0.339	0.465 0.299	0.165 0.205	0.173 0.236
S3	0.087	0.063	0.496 0.496	0.063 0.260	0.394 0.449	0.323 0.260	0.047 0.252	0.047 0.244
S5	0.102	0.102	0.480 0.465	0.110 0.276	0.331 0.370	0.354 0.276	0.087 0.323	0.102 0.315
Spectral Enhancement	0.142	0.228	0.433 0.472	0.181 0.276	0.457 0.331	0.496 0.402	0.094 0.126	0.276 0.071
Spectral Attenuation	0.079	0.063	0.496 0.480	0.071 0.283	0.252 0.268	0.299 0.228	0.047 0.126	0.047 0.087
Random Noise S = 1 (RN1)	0.079	0.063	0.307 0.354	0.055 0.268	0.472 0.425	0.394 0.370	0.055 0.323	0.071 0.283
Random Noise S = 2 (RN2)	0.228	0.157	0.457 0.362	0.134 0.331	0.449 0.496	0.535 0.417	0.197 0.346	0.157 0.441

Table 6. Number of false-negative detections out of the 32 attacks when BER was set to be 0.20. The corresponding false-positive probability for random watermarks is 1.56×10^{-12} . L and H stand for approximation and detail subband embedding, respectively.

DCT	SSA	Lazy		Haar		D4		D8		5/3		9/7	
		L	H	L	H	L	H	L	H	L	H	L	H
2	1	14	17	0	10	8	11	8	9	0	8	1	8

Table 7. Average processing time in ms.

(a) Mesh12.

	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Embedding	0.235	34.06	14.82	20.01	23.22	31.56	17.27	34.94
Extraction	3.585	37.64	13.10	13.59	16.03	22.10	12.65	25.85

(b) Bream.

	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Embedding	0.390	36.56	20.23	27.98	31.95	44.78	24.51	47.76
Extraction	5.550	43.34	15.99	20.28	22.84	32.42	20.20	34.42

(c) Akiyo.

	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Embedding	0.625	28.71	25.73	35.08	40.17	54.88	29.83	60.90
Extraction	4.455	34.70	19.31	25.38	28.04	38.49	24.17	43.16

(d) Mesh06.

	DCT	SSA	Lazy	Haar	D4	D8	5/3	9/7
Embedding	0.150	36.02	13.90	19.43	22.50	31.25	16.39	34.94
Extraction	3.660	40.01	12.71	14.55	15.72	23.49	14.67	25.05

watermark verification, and the results are given in Table 7. All the simulations were conducted on a Pentium-IV 1.6 GHz PC with 256 MB of DRAM. The elapsed time was averaged over 200 experiments to achieve statistical reliability. The SSA and the 9/7 filter were the most time-consuming transformations among all of them. Among all the transformations, DCT was distinguishably faster because only a simple averaging operation for partial watermark-bearing coefficients is needed. The time required for least-squares mesh registration is included in the watermark extraction process. Recall that each of the test 2D dynamic meshes contained 128 MOPs. It follows from the results shown in Table 7 that no more than 0.5 ms/MOP was required for watermark embedding or watermark identification, even with the most complicated mesh (Akiyo) and filter (SSA or 9/7). Since the animated mesh was rendered at 30Hz or 33.3 ms/MOP, the computational load of the proposed scheme is very light and is suitable for real-time applications.

5. CONCLUSIONS

We have explored several time-series analysis tools, including DCT, SSA, and DWT, for watermarking MPEG-4 2D mesh animation. These transformations can efficiently extract the important motions of an MPEG-4 2D dynamic mesh. An invisible binary watermark is cast upon the important motion components based on the spread-spectrum principle. A simple hard-decision detector along with a spatial-domain registration technique is used for watermark retrieval. We have evaluated the proposed system against a variety of attacks. Of the various time-series analysis tools under study, SSA and approximation subband embedding with Haar, 5/3, or 9/7 wavelets achieved the best robustness performance, while DCT (with only DC value modifications) showed moderate robustness with the lowest complexity. The experimental results verify the usefulness of the proposed transformation-based framework for copyright protection.

REFERENCES

1. I. J. Cox, M. L. Miller, and A. J. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2002.
2. G. C. Langelaar, I. Setywan, and R. L. Lagendijk, "Watermarking digital image and video data," *IEEE Signal Processing Magazine*, Vol. 17, 2000, pp. 20-46.
3. R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three dimensional polygonal models through geometry and topological modifications," *IEEE Journal on Selected Areas in Communications*, Vol. 16, 1998, pp. 551-560.
4. O. Benedens, "Geometry-based watermarking of 3D models," *IEEE Computer Graphics and Applications*, Vol. 19, 1999, pp. 46-55.
5. E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proceedings of SIGGRAPH '99*, 1999, pp. 49-56.
6. H. Hoppe, "Progressive meshes," in *Proceedings of SIGGRAPH '96*, 1996, pp. 99-108.
7. R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, "Watermarking 3D polygonal meshes in the mesh spectral domain," in *Proceedings of Graphics Interface*, 2001, pp. 9-17.
8. R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes," *Computer Graphics Forum*, Vol. 21, 2002, pp. 373-382.
9. K. Yin, Z. Pan, J. Shi, and D. Zhang, "Robust mesh watermarking based on multiresolution processing," *Computers and Graphics*, Vol. 25, 2001, pp. 409-420.
10. K. Murotani and K. Sugihara, "Watermarking 3D polygonal meshes using the singular spectrum analysis," in *Proceedings of Mathematics of Surfaces: 10th IMA International Conference*, 2003, pp. 85-98.
11. T. H. Kim, J. Lee, and S. Y. Shin, "Robust motion watermarking based on multiresolution analysis," in *Proceedings of EUROGRAPHICS*, Vol. 19, 2000, pp. 189-198.
12. S. H. Yang, C. Y. Liao, and C. Y. Hsieh, "Multiresolution watermarking for MPEG-4 2D mesh animation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, Vol. E87-A, 2004, pp. 879-886.

13. I. J. Cox, J. Killian, F. T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, Vol. 6, 1997, pp. 1673-1687.
14. W. Zhu, Z. Xiong, and Y. Q. Zhang, "Multiresolution watermarking for images and video," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, 1999, pp. 545-550.
15. ISO/IEC 14496-2, Information Technology – Coding of Audio-Visual Objects, Part 2: Visual.
16. K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithm, Advantages, and Applications*, Academic Press, 1990.
17. J. B. Elsner and A. A. Tsonis, *Singular Spectrum Analysis: A New Tool in Time Series Analysis*, Kluwer, 1996.
18. E. J. Stollnitz, T. D. DeRose, and D. H. Sallesin, *Wavelets for Computer Graphics: Theory and Applications*, Morgan Kaufmann Publishers, 1996.
19. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, CBMS series, 1992.
20. B. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000," *IEEE Signal Processing Magazine*, Vol. 18, 2001, pp. 22-35.
21. J. Huang, Y. Q. Shi, and Y. Shi, "Embedding image watermarks in DC components," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, 2000, pp. 974-979.
22. Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Image and Vision Computing*, Vol. 10, 1992, pp. 145-155.



Shih-Hsuan Yang (楊士萱) received the B.S. degree in Electrical Engineering from the National Taiwan University in 1987. He obtained the M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from the University of Michigan, Ann Arbor, in 1990 and 1994, respectively. Since 1994, he has been a faculty member of the National Taipei University of Technology, Taiwan. He is currently an Associate Professor of Computer Science and Information Engineering. His major research interests include image and video coding, multimedia transmission, data hiding, and information theory.



Chin-Kuen Liang (梁晉坤) received the M.S. degree in Computer Science and Information Engineering from the National Taipei University of Technology in 2004. His interest is software programming. His previous work experience included window user interface programming and IC card driver applications.



Chin-Yun Hsieh (謝金雲) is an Associate Professor of Computer Science and Information Engineering at the National Taipei University of Technology. Dr. Hsieh received a diploma in Electrical Engineering from the National Taipei Institute of Technology, Taiwan, in 1979, the M.S. degree in Computer Science from the University of Mississippi, Oxford, Mississippi, in 1984, and the Ph.D. degree in Computer Science from the University of Oklahoma, Norman, in 1991. His research interests include software engineering, distributed and parallel computing, and Internet applications.