Journal of Electronic Imaging 13(1), 209–216 (January 2004).

Comparative evaluation of semifragile watermarking algorithms

Özgür Ekici
University of Ottawa
School of Information Technology and Engineering
Ottawa, Canada

Bülent Sankur
Barış Coşkun
Umut Naci
Mahmut Akcay
Boğaziçi University
Department of Electrical and Electronics Engineering
Bebek, Istanbul, Turkey

Abstract. *Semifragile watermarking techniques aim to prevent tampering and fraudulent use of modified images. A semifragile watermark monitors the integrity of the content of the image but not its numerical representation. Therefore, the watermark is designed so that the integrity is proven if the content of the image has not been tampered with, despite some mild processing on the image. However, if parts of the image are replaced with the wrong key or are heavily processed, the watermark information should indicate evidence of forgery. We compare the performance of eight semifragile watermarking algorithms in terms of their miss probability under forgery attack, and in terms of false alarm probability under nonmalicious signal processing operations that preserve the content and quality of the image. We propose desiderata for semifragile watermarking algorithms and indicate the promising algorithms among existing ones.* © *2004 SPIE and IS&T.* [DOI: 10.1117/1.1633285]

1 Introduction

One prominent application of watermarking technology is the monitoring of the integrity of the multimedia documents. The specific interest in semifragile watermarking algorithms arises from the multitude of practical and commercial applications, where content needs to be strictly protected, but the exact representation during exchange and storage need not be guaranteed. The alterations on the documents can occur unintentionally or can be implanted intentionally. The so-called unintentional or innocent alterations typically arise from such diverse facts as bit errors during transmission and storage, or signal processing operations such as filtering, contrast enhancement, sharpening, and compression. Intentional or malicious alterations, on the other hand, are assumed to be due to an explicit forgery attempt by a pirate with the explicit purpose of changing the contents of a document. The main distinction then, is whether the content is altered as in malicious and intentional attacks or whether only the representation, but not the content, of the document is altered, as occurs in unintentional, nonmalicious cases. The line of demarcation between these two attacks categories is, however, not always clear-cut, as it depends very much on the application domain. A case in point is histogram equalization, where, if the representation of the lighting condition is considered informative, it then becomes a malicious manipulation; alternately, it remains simply a well-intentioned contrast enhancement.

There have been a significant number of watermarking algorithms aimed at tamper detection. One group of techniques,[1,2] called fragile watermarking algorithms, function as a strict tamper detection tool, in that they are intolerant of even a one-bit alteration. They are alternatively called cryptographic message digests, and can only validate original copies. On the other hand, semifragile tamper detection methods are designed to monitor changes in the content. In other words, they are capable, in principle, to differentiate between the innocent and malicious attack categories. Ideally, a semifragile tamper detector would gloss over innocent alterations on the image due, for example, to postproduction editing, mild compression, filtering, or contrast enhancement, but it should give an alarm whenever a content change occurs. Semifragile watermarking methods validate image content, but not its representation, and are thus judiciously made robust against allowable alterations, while being sensitive to nonpermitted modifications. Of course, if the signal processing operations are taken far enough, as in the case of high compression rates, they start changing not only the representation, but the content of the document as well, and they should then qualify as a mali-

Paper 02080 received Aug. 6, 2002; revised manuscript received Mar. 3, 2003; accepted for publication Jul. 1, 2003. This paper is revision of a paper presented at the SPIE conference on Multimedia Systems and Applications IV, Aug. 2001, Denver, Colorado. The paper presented there appears (unrefereed) in SPIE Proceedings Vol. 4518.
1017-9909/2004/$15.00 © 2004 SPIE and IS&T.

cious attack. The breakpoint between an innocent and a malicious attack is not well defined, and it depends on the application domain and document type. One can use, however, the reliability of the detected watermark as an authenticity measure of the document, and reach, for example, a tampered or nontampered decision by thresholding it.

There are several possible classifications of the tamper-detection and content-authentication watermarking techniques.[3,4]

- Visually authenticated (semi)fragile watermarking, where typically a thumbnail or a visual pattern is hidden in the image, and tamper detection is based on the visual assessment of perceived differences by an operator, as in Marvel *et al.*[5,6]
- Statistically authenticated (semi)fragile watermarking, where an estimate of tampering likelihood is obtained based on the correlation coefficient or measured mismatch between the inserted and recovered authentication sequences. In these techniques, the performance can typically be given in terms of probability of correct watermark sequence detection as a function of false alarm rate.[7–11]

Self-embedding techniques for proof of authenticity and image protection as in Fridrich and Goljan,[12] Lan and Tewfik,[13] and in the case of video in Robie and Mersereau.[14]

Proposed algorithms can be self authenticating or independently authenticating.

- Self-authenticating algorithms are based on the validation of a robust hash, which was embedded and which is to be extracted again from the test image itself, as in the methods of Lin and Chang,[10,15] Fridrich,[8,16–18] Xie, Arce, and Graverman,[19] and, to some extent, Hung, Cheng, and Chen.[20]
- Independently authenticated algorithms receive validation based on an image-independent authentication sequence, as in the methods of Eggers,[7,21] Fridrich,[16] Lin, Podilchuk, and Delp,[9] Queluz and Lamy,[22,23] Kundur and Hatzinakos,[24,25] Lan, Mansour, and Tewfik,[26] and Gwo, Lu, and Liao.[27]

We plan to present a comparative assessment of the statistically authenticated, oblivious, semifragile watermarking techniques.[28] This excludes visually verified algorithms, such as Marvel, Hartwig, and Boncelet,[6] and Yeung and Mintzer,[2] or nonoblivious techniques as in Xie, Arce, and Graverman.[19] We measure the performance in terms of probability of miss when there is a forgery attack and in terms of probability of false alarm when there is no forgery, but the image is subjected to various mild signal-processing operations. Obviously, one desires to achieve low probability of miss $\{P_M\}$ when there is a forgery attack, and at the same time, low probability of false alarm $\{P_F\}$ when one deals with allowed signal processing operations. Thus, the work aims to test and compare the algorithms on the basis of their claim to semifragility, that is, resistance to mild signal-processing operations and their specificity to detect forgery. The forgery attack we experiment with is the substitution attack, which substitutes the semifragilely watermarked document with its original unwatermarked version. In Sec. 2, we give the rationale of considering the substitution attack as representative of malicious attacks.

The outline of the work is as follows. Section 2 describes the types of attacks that should rightly cause an alarm and the set of attacks against which the tamper-detection watermark should resist. The comparison method of semifragile techniques is also given. Section 3 describes the semifragile watermarking algorithms with brief descriptions of their insertion, extraction, and verification schemes. Section 4 provides comparative results and conclusions.

2 Methods for Comparison

The semifragile watermarking methods should be moderately robust to differentiate between malicious and nonmalicious attacks. However, the line of demarcation between the benign and malicious attacks is application and document dependent. In this study, we select the following list of manipulations based partly on examples in the literature and partly somewhat subjectively. We expect that the semifragile watermarking algorithms should not give false alarms against these permissible alterations:

- mild compression, for example up to 70% JPEG
- histogram equalization (uniform distribution)
- sharpening (unsharp masking filter with coefficients $[-1-1-1;\ -1A+8,-1;\ -1-1-1]$, where $A=1$)
- low-pass filtering within a support of 3×3 (equal weight coefficients equal to 1/9)
- median filtering within a support of 3×3
- additive Gaussian noise down to a signal to noise ratio of 35 dB
- salt-and-pepper noise, up to 1% (value set to 255 and 0, respectively)
- random bit errors in transmission and storage of the image in raw format, with a 0.001 probability of bit errors.

As pointed out in Sec. 1, it is arguable when and if these signal manipulations do not constitute an authenticity threat. Our aim, however, is first to prepare a list of algorithms with measured robustness and temper-sensitivity properties. Then, given an application scenario with specific robustness and tamper-sensitivity properties, in principle it should be possible to select the adequate algorithm.

The main content-altering manipulations that must generate tamper alarm, hence, the nonpermissible alterations, are the following:

- image forgeries intended to remove, substitute, or insert objects in the scene
- image manipulations that modify the geometry of objects such as their rotation, flipping, translation, and scaling or image manipulations that change the appearance of objects such as color, shade, shadow manipulation, etc.
- changes in the scene background, for example, change of the time of day or changes in background texture such as forest, ocean, etc.

- cropping.

All manipulations that affect the geometry of the image such as flipping, rotation, cropping, etc., suffer from desynchronization and are thus automatically detectable by the algorithms considered. Consequently, manipulations such as mild cropping, or interpolation that could have been considered as innocent operations in certain circumstances, are eliminated. On the other hand, the manipulations of object insertion and deletion, and scene background changes, are tantamount to substitution. It follows then that the variety of forgery attacks considered before can be collapsed to substitution attacks only. Furthermore, we implement the substitution attack by replacing the watermarked image portion with its original version. There are two reasons for this specific choice. One is that the substitution of an image block with the original version would be the most difficult to detect, hence the most challenging attack. The second reason is that it is impractical to conceive and implement literally thousands of forgery attacks using commercial image-processing tools, and it would be very difficult to gauge and normalize the severity of the attacks. Thus, substitutions make it feasible to streamline forgery attacks.

We used ten different images of size 512×512, each watermarked ten times with different keys. The performance was measured in terms of the false alarm rate P_F, which is the probability that image blocks indicate tampering in the absence of any malicious attack, and in terms of the miss probability P_M, which is the percentage of maliciously attacked images that do not generate any tampering alarm. Notice that the substitution attack was block-based, and hence coincided with the grid structure of the algorithms. Random placements of the substituted blocks astride the grid structure of the algorithm would simply weaken the level of attack within the original blocks of the algorithms.

The eight semifragile watermarking methods that we tested are listed in Table 1. As shown in Table 1, different watermarking algorithms are conceived to protect image regions of different size. For example, some methods check for tampering on a pixel basis, others act on the basis of 8×8, 16×16, or 64×64 blocks, while still others are designed to monitor row/column triples.

To achieve a normalized basis of comparison among the algorithms, we did two things: insertion strength normalization and footprint normalization. First, the insertion strength was standardized and all the tested algorithms were tuned to achieve two levels of document-to-watermark ratio, namely, 38- and 41-dB peak signal-to-noise ratio (PSNR). The necessity of any watermarked image to possess PSNR above the lower limit of 38 dB was suggested in Kutter and Petitcolas.[29] Second, given the different footprint sizes on which the algorithms could detect evidence of tampering, we converted the footprint size to a standard 64×64 block. In fact, we evaluated the P_M and P_F scores in two different ways, which we denote as native results and integrated results.

Native results. The performance figures are calculated on the basis of the image block size as proposed in the original work, hence called the native size. Thus, for example, the statistics for the Lin-Chang, Eggers-Girod methods were first collected on the basis of 8×8 blocks, for Lin-Podilchuk-Delp on the basis of 16×16 blocks, for Lan-Mansour-Tewfik and Fridrich methods on the basis of 64×64 blocks, and for the Queluz method on the basis 3×128 pixel lines.

Integrated results. To be able to compare performances over regions of the same size, the miss and false alarm rates of the prior native blocks are integrated to 64×64-sized image blocks. It would have been desirable to compare algorithms on various sized block attacks, let's say from 4×4 on to larger sizes. However, not all algorithms could function at all sizes, hence the only reasonable size at which all algorithms could compete was 64×64. To convert the performance figures of an algorithm that was conceived for native blocks, (e.g., 8×8 subblocks or 128-pixel columns) to P_M and P_F figures for the 64×64 blocks, we calculated the probability of exceeding a threshold in the absence of attacks. We note that almost all semifragile methods give occasionally false alarms, even in the absence of any attack. Thus a 64×64 region is declared as tampered if a sufficient number of its native blocks are found tampered. This critical threshold, that is, the number of false-alarming native blocks, was set at the 1% level for the 64×64 block in the absence of any attack.

In summary, the performance of the algorithms was measured on both their native sizes and the larger size 64×64. While no one algorithm was favored, the eight methods investigated and their characteristics are listed in Table 1.

3 Semifragile Watermarking Algorithms

In this section, we briefly describe the semifragile watermarking schemes tested. For each algorithm, we outline the generation of the authentication sequence, the insertion, extraction, and verification procedures. In the sequel, we express the image at pixel location (x,y) as $I(x,y)$, while to denote pixels of a particular block b, we use the notation $I_b(x,y)$. The block discrete cosine transform (DCT) coefficients are indicated by $C_b(p,q)$.

3.1 *Lin-Chang Algorithm*

Lin and Chang's algorithm[10,15,30] is conceived to tolerate, in particular, JPEG-style compression of the watermarked image. It is based on two properties of the DCT coefficient quantization, namely, 1. order invariance, where the order relation of DCT coefficient pairs remains unaltered after JPEG processing, if not set equal; and 2. coefficient invariance, where if a coefficient is quantized to an integer multiple of the step size, its value is not changed after JPEG compression with a smaller step size. A parallel algorithm is that of Hung, Cheng, and Chen,[20] which uses the block vector quantization (VQ) indices for authentication data. This algorithm, however, results in a large payload of VQ coefficients, and its performance turns out to be rather poor.

Authentication data. The authentication data consists of the ordinal relationship of three pairs of DCT coefficients chosen from 8×8 blocks pair-wise coupled according to a random mapping. These coefficients are selected from a

Table 1 Characteristics of the semifragile watermarking methods as given in their original paper.

Algorithm	Insertion domain and method of insertion	Size of control area	Authentication information/method
Lin-Chang	DCT coefficients and JPEG-50 quantization	Pair of 8×8 blocks	Ordinal relation of randomly chosen three DCT coefficients in a block
Lin-Podilchuk Delp	Spatial and additive mixing	16×16 blocks	Inverse DCT of random noise pattern planted in the mid (one-third) band of the transform block coefficients
Eggers-Girod	DCT coefficients and binary QIM	8×8 blocks	Scalar Costa scheme: odd/even dithered quantization of DCT coefficients
Fridrich	Spatial and additive mixing	64×64 blocks	Robust hash of the block, obtained by quantized projections of the block onto 30 smoothed random bases, acting as seeds for random noise and a separate low-frequency sequence
Kundur-Hatzinakos	QIM of Haar wavelet coefficients	4×4 blocks	Authentication sequence inserted via odd/even quantization of the four-level wavelet coefficients
Gwo-Lu-Liao	QIM of the block mean of wavelet coefficients	4×4 blocks	Authentication sequence inserted via odd/even quantization of the average value of selected groups of wavelet coefficients
Queluz	Spatial and adaptive quantization of row/column projections	Triad of image rows columns	Authentication sequence inserted via odd/even quantization of projections of column (row) triples onto random bases
Lan-Mansour-Tewfik	DCT coefficients and Hadamard projections and quantization	Group of (typically 64) of 8×8 blocks.	Authentication sequence inserted via odd/even quantization of projections of DCT coefficient vectors onto Hadamard bases

predetermined low-frequency path in the respective DCT blocks. Their ordinal relationship, which forms the authentication data, remains invariant under JPEG compression. An authentication bit is considered as 1 if the DCT coefficient in a selected path is greater than its paired partner; otherwise, the authentication bit is 0. In other words, for the coupled blocks, denoted as 1 and 2, we have the authentication information at the (p,q) coefficient:

$$\phi(p,q)=\begin{cases}1 & C_1(p,q)-C_2(p,q)\geqslant 0\\ 0 & C_1(p,q)-C_2(p,q)<0\end{cases}.$$

Insertion method. The six bits extracted from a block pair are inserted in alternate DCT coefficients of the same block pair by forcing these coefficients to odd or even multiples of a JPEG50 quantization step size. If the bit to be inserted complies with the least significant bit of the carrier DCT coefficient divided by the quantization step size, then no change is made, otherwise, the coefficient is incremented by a one-step size amount. Since the embedding distortion may cause a change in the authentication bits, this procedure should be iterated a few times (typically three) before the DCT coefficients are stabilized.

Extraction and verification method. The extraction scheme is a replica of the insertion scheme, in that both the authentication bits are regenerated and the least significant bit (LSB) bits of the modulated DCT carrier coefficients are read off. An 8×8 block is declared nontampered if at least five of the six inserted bits are verified. Otherwise, the block is considered as tampered.

3.2 *Lin-Podilchuk-Delp Algorithm*

Authentication sequence. The authentication sequence is a pseudo-random zero-mean, unit-variance Gaussian noise sequence. Its seed is controlled by a key but is otherwise independent of the document.

Insertion method. The Gaussian authentication sequence is placed in the upper triangular positions (excluding DC component) of an empty DCT matrix. Subsequently, the inverse DCT of the matrix is calculated and the resulting 16×16 spatial pattern is mixed additively with the image DCT block at a given strength:

$$I_b'(p,q)=I_b(p,q)+\gamma W_b(p,q).$$

Finally, the block-wise inverse DCT yields the semifragilely watermarked image.

Extraction and verification method. The watermark in the image is estimated by suppressing image spectral components in every block while enhancing the presence of the watermark. To this effect, horizontal (column-wise) and vertical (row-wise) differences are calculated both for the test image and the spatial watermark pattern. These difference vectors from the horizontal (Δ_{Col}) and vertical (Δ_{Row}) sets are concatenated to form two sets, one derived from the test image,

$$I_b^*=\{\Delta_{\mathrm{Col}}[I_b(x,y)]|\Delta_{\mathrm{Row}}[I_b(x,y)]\},$$

Table 2 Performance analysis of Lin-Chang's semifragile watermarking on the basis 8×8 block pairs. Overall, 20480 blocks were tested.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	11.2%	0.0%	44.3%	88.6%	39.2%	66.1%	0.4%	0.0%	0.0%
41	11.3%	0.0%	48.9%	89.1%	42.9%	69.4%	8.7%	0.1%	0.0%

and the other from the watermark spatial pattern,

$$W_b^* = \{\Delta_{\mathrm{Col}}[W_b(x,y)] | \Delta_{\mathrm{Row}}[W_b(x,y)]\},$$

where | denotes the concatenation operation. The verification is based on the correlation of the extracted data with the differenced version of the original watermark pattern, that is:

$$\rho = \frac{\langle I_b^*, W_b^* \rangle}{[\langle I_b^*, I_b^* \rangle \langle W_b^*, W_b^* \rangle]^{1/2}} \geqslant T_b .$$

3.3 *Egger-Girod Algorithm*

Authentication data. The authentication data consist of a random binary sequence $\{d_n\}$ embedded with a secret dither sequence $\{k_n\}$.

Insertion method. The authentication message is embedded in cover image coefficients by a dithered quantization rule $Q_\Delta\{\}$, where Δ is the step size corresponding to the strength of insertion, referred to also as the scalar Costa's scheme (SCS). The embedding is randomized by a pseudo-random dither sequence $k_n \in (0,1]$. The cover data selected for watermark insertion are the second through eighth coefficients in the zigzag order of the 8×8 block DCT coefficients. The embedding rule for the n'th element can be written as

$$a_n = \Delta\left(\frac{d_n}{2} + k_n\right) \quad s_n = x_n + \alpha(Q_\Delta\{x_n - a_n\} + a_n - x_n),$$

where $Q_\Delta\{.\}$ indicates scalar uniform quantization with step size Δ. This embedding scheme is controlled by two parameters: the quantization step size Δ and the scale factor α. Both parameters can be jointly tuned to achieve a good trade-off between the embedding distortion and detection reliability for a given noise variance of an additive white Gaussian noise (AWGN) attack.

Extraction and verification method. Watermark extraction consists of observing the quantization residual. The residual should be in the $(-\Delta/2, \Delta/2)$ interval for a 0 authentication bit, and its absolute value should be in the $(\Delta/2, \Delta)$ interval for 1. The tampering decision is based on the likelihood test that determines whether the watermark sequence was embedded with key k_n, or was not embedded with that specific key.

3.4 *Fridrich's Algorithm*

Authentication data. The authentication bits are generated as a robust visual hash of the 64×64 image blocks. Each block is projected onto M (30) basis vectors, and their inner product is quantized to 1 bit. The quantization threshold is adjusted to make equal the occurrence of ones and zeroes. The basis vectors themselves are obtained by smoothing 2-D arrays of uniform random numbers. The authentication sequence is thus tied intimately to the image content. For robustness against low-pass filtering, a separate pattern obtained by a geometric sequence of real numbers with factor α, is obtained.

Insertion method. For each image block, a set of M random noise patterns of size 64×64 are generated. The seeds for the random number generator are each different, and obtained by the concatenation of the block projection bit onto the i'th ($i = 1...M$) basis, the block identity number, and a random key. These M 64×64 random patterns are summed and scaled to form a spread-spectrum signal, and they are made DC free and mixed additively to the middle

Table 3 Performance of Lin-Podilchuk-Delp's semifragile watermarking on the basis 16×16 blocks. Overall, 81,920 blocks are tested, where the threshold is $T_b = 0.1$. The insertion gain is set to $\gamma = 5$ and to $\gamma = 3.5$ to achieve 38- and 41-dB PSNR, respectively.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	6.5%	6.7%	52.9%	5.6%	18.3%	8.3%	7.5%	7.1%	7.2%
41	6.4%	14.3%	69.1%	11.0%	36.5%	15.5%	15.9%	14.2%	14.1%

Table 4 Performance analysis of Eggers-Girod's semifragile watermarking on the basis 102,400 8 $\times$8 pixel blocks. Quantization step size $\Delta=26$ and $\Delta=35$ for 41 and 38 dB, respectively ($\alpha=0.8$). Seven authentication bits on the second through the seventh DCT coefficients were embedded per block.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	26.7%	0.0%	31.6%	61.1%	35.2%	62.5%	0.4%	0.1%	1.9%
41	24.9%	0.0%	42.3%	59.7%	37.3%	67.1%	8.6%	0.2%	2.5%

one-third of the block DCT coefficients. In addition, after the block has been rendered zero-mean and fixed variance, its first 300 DCT coefficients are perturbed to conform to a given pattern of 1 and 0 indices. These indices are obtained via the modulo operation of the DCT coefficients with the geometrical sequence.

Extraction and verification method. The received image is divided into blocks of the same size, and the spread-spectrum signal is regenerated in the same way as in the insertion stage. This spread-spectrum signal is correlated with the middle third of DCT coefficients and compared with a threshold, which is adjusted to render the number of ones and zeros equal, as in the insertion stage. The tampering decision is based on the probability of obtaining k correct symbols out of M, that is $P_{\text{tamper}}=C(M,k)2^{-k}$, where $C(.)$ denotes the combinatorial function. In our experiments, we took $M=30$ while $k=22$ to satisfy $P_f<0.01$. For low frequency, a weighted correlation between attained indices of DCT coefficients and a watermark pattern is calculated.

3.5 *Kundur-Hatzinakos Algorithm*

Authentication sequence. The authentication data is a random sequence independent of the image content, called the tamper authentication function (TAF).

Insertion method. A four-level discrete wavelet transform (DWT) of the image is taken using Haar bases. The authentication bits are inserted in the wavelet coefficients by quantification to even or odd multiples of a step size, according to the polarity of the TAF sequence bit. More specifically, in a given band and a given position, only one of the randomly chosen horizontal, vertical, or diagonal components is marked. The quantization step size is given by $\Delta=2^l$, where l denotes the resolution level, $l=0$ being the original image. The decision to map the wavelet coefficients to odd or even multiples of the quantization coefficient is randomized via a key.

Extraction and verification method. This stage mimics the insertion stage, in that the DWT of the test image is calculated, and the coefficients within which the bits were embedded are searched. The odd or even quantization state of the coefficients is estimated to obtain the hidden authentication sequence, which is to be compared with the regenerated sequence. A pixel at the 0 level (original image) is declared as tampered if the corresponding pixel in the fourth level appears tampered, or if the fourth level coefficient passes the test, the coefficients in the second and third levels both fail. The high miss probability of a 4×4 block in a substitution attack is due to the fact that the fourth level pixel bit will not match 50% of the time, while the third and second level pixel bits will not match one eighth of the time, thus in total will check for error 62.5% of the time, or alternatively will miss 37.5% of the time.

3.6 *Queluz Algorithm*[31]

The following information is based on Queluz and Lamy's research.[11,22,23,31]

Table 5 Performance of Fridrich's semifragile watermarking on 64$\times$64 pixel blocks with insertion strength adjusted to $\gamma=0.7$ and to $\gamma=0.6$ in spread spectrum watermarking; $\alpha=0.062$ and $\alpha=0.045$ in low-frequency watermarking to attain 38- and 41-dB PSNR, respectively. A block is declared tampered if P_{tamper} is above 1% in the spread spectrum algorithm or the correlation value is less than 32% in a low-frequency algorithm. A block is not declared as tampered if it passes in any of the algorithms. Overall, 6400 blocks were tested.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	0.6%	1.1%	43%	20.0%	11.4%	3.1%	1.1%	5.0%	1.9%
41	1.0%	1.6%	62%	21.0%	19.5%	5.5%	2.5%	25.8%	2.5%

Table 6 Performance of Kundur-Hatzinakos semifragile watermarking, where the quantization step sizes Δ are taken as factors of 4, 8, and 16, respectively, for levels 2, 3, and 4 of the decomposition. These step sizes are adjusted to attain 41- and 38-dB PSNR. Overall, 1,310,720 4×4 blocks were tested.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	37.6%	0.1%	37.2%	50.6%	25.4%	57.4%	15.9%	13.3%	0.1%
41	37.3%	0.1%	38.7%	53.3%	31.3%	59.0%	20.7%	14.9%	0.1%

Authentication data. The authentication data consists of a random sequence, independent of the image, inserted in quantized projections of rows and columns.

Insertion method. Nonoverlapping image columns or rows are considered in groups of three. Each triad is projected onto three random basis functions, resulting in projection values P_1, P_2, and P_3. We have observed that using the low-pass filtered version of these arrays improves the performance slightly. The resulting inner products are rank ordered as $P_{[1]}\leqslant P_{[2]}\leqslant P_{[3]}$ and a quantization step size Δ is calculated based on the span of these projections, which is $|P_{[3]}-P_{[1]}|$. The median projection $P_{[2]}$ is then quantized to odd or even multiples of the Δ step size according to the bit to be inserted.

Extraction and verification method. For each triad of lines, the embedded bit is extracted by computing the position of the median projection $P_{[2]}$ to the nearest boundary: if the parity of this boundary is even, a 0 is extracted; otherwise, a 1 is extracted. It is also required that the median projection be within a percentage of distance to its nearest boundary. A triad of lines is deemed tampered if its quantization state does not check the authentication bit. We take two rows (columns) concatenated at a time from the 64×64 blocks, resulting in 128-pixel-long test lines.

3.7 *Lan-Mansour-Tewfik Algorithm*

Authentication data. The data consists of a binary watermark message that instruments the odd/even quantization of feature vectors.

Insertion method. DCT transform of the 8×8 blocks of the image is first taken. The DCT coefficients of the same order, say all (i,j) coefficients, from the blocks of the image are collected into 64-long vectors using Hilbert scanning paths. Actually, the blocks are visited in a Hilbert scan, the (i,j)'th DCT coefficients form a string, whose length equals the number of blocks in the image (e.g., 4096 in an 512×512 image), and then they are further partitioned into smaller subvectors of size 64. The subvectors become, in effect, the ensemble of DCT coefficients of the same order over a neighborhood, due to Hilbert scanning. These DCT subvectors are projected onto columns of the Hadamard matrix, and finally these projections are quantized to odd and even multiples of a step size.

Extraction method. The extraction process is the replica of the insertion method followed by reading off of the odd/even quantization state of the projections. A tampering decision is based on the number of projections that does not satisfy the predetermined quantization state of the projections.

3.8 *Gwo-Lu-Liao Algorithm*

There are a number of algorithms, which in the quest for robustness, embed the information in the average value of blocks, and redistribute the change in the mean to the pixels.[27,32,34] We tested the algorithm in Ref. 27, which was described in more detail.

Authentication sequence. The authentication data is a random sequence independent of the image content.

Insertion method. Similar to the Kundur-Hatzinakos algorithm,[25] first a four-level DWT of the image is taken. Then the average value of a number of wavelet coefficients in one of the HH, LH, or HL bands is calculated. The 0 or

Table 7 Performance of Queluz's semifragile watermarking on basis 128-pixel line triad projections. Overall, 20 line triads per 64×64 block are considered, and the quantization step size is $\Delta=2.5$ and $\Delta=3.7$, respectively, to achieve 41- and 38-dB PSNR. Overall, 128,000 triads were tested.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	50.0%	0.9%	7.7%	45.9%	6.1%	39.2%	1.1%	1.3%	1.1%
41	49.9%	1.1%	11.9%	50.3%	16.7%	41.6%	1.1%	1.2%	1.1%

Table 8 Performance of Lan-Mansour-Tewfik's semifragile watermarking on the basis 64 8×8 pixel blocks, corresponding to a 64×64 pixel block. The DCT coefficients, rank-ordered according to the JPEG quantization table, are grouped into 64-D subvectors. DCT coefficients 4 through 13 in zigzag order are marked with odd/even quantization. The step size is $\Delta=26$. Overall, 6400 blocks are tested.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	16.9%	1.4%	83.4%	49.7%	82.6%	82.7%	85.5%	76.1%	76.8%

1 watermark bit value is imposed onto the block average using odd/even quantization step sizes. These step sizes are obtained from distributing the change to the block pixels, which effects this odd or even quantization of the block mean. The quantization step size is taken on the basis of wavelet frequency masking visibility coefficients, which depend on the level and orientation. Only levels 2 through 4 are used for embedding.

Extraction and verification method. This DWT of the test image is calculated and the block means are computed and checked to see if they verify the odd/even quantization condition. The decision fusion is done on level-2 pixels, which correspond to 4×4 blocks of the original image.

4 Experimental Results

Eight semifragile watermarking methods have been described in Sec. 3 and their performance figures have been given according to their native block sizes in Tables 2 through 9. For a comparative assessment, their watermark-to-document ratio, as given by the PSNR, was set to 38 and 41 dB. Sample error images resulting from semifragile watermarking at the 38-dB level are given in Fig. 1. In these figures, for the sake of visibility, the error signal has been multiplied by a factor of 10 and it has been put on a pedestal of 128. To normalize the size of the tamper-control region, we have used an integration scheme to convert the native block size to a fixed control size of 64×64. As it was explained in Sec. 1, this block size was imposed, in that it was the minimum effective dimension at which some algorithms[8] could work properly. On the other hand, 1/64th of a 512×512 image appears as a reasonable tamper verification size. Notice that for each algorithm native size, the integration is done with a sliding window in steps of its native size, e.g., in steps of 8×8 pixels for the Lin-Chang algorithm, and so forth.

The integration scheme uses the tamper and no-tamper decisions of the blocks with native sizes, and integrates them into a tamper/no-tamper decision for the 64×64-sized block. We first determine the threshold level for each algorithm, which guarantees lower than 1% false alarm rate under the no attack case. Recall that all algorithms, when a verification test is applied, give rise to some level of false alarm even in the absence of any attack. More specifically, we have found experimentally the following false alarm thresholds, as in Table 10.

In Tables 11 and 12 the miss and false alarm probability results are given for the 64×64 regions.

The following comments can be made on the integrated performance of semifragile algorithms, that is, on 64×64 blocks.

- Lin-Chang's algorithm: As expected the algorithm performs very well in the presence of JPEG compression, but otherwise it is very fragile against signal-processing attacks.
- Lin-Podilchuk-Delp's algorithm: It is robust against almost all nonmalicious signal-processing operations, except for smoothing. We have observed that the use of Wiener filtering, to enhance the watermark signal, slightly improves the performance.
- Eggers-Girod's algorithm: This algorithm provides a statistically meaningful measure of tamper probability. It proves very robust against JPEG as it withstands (does not give false alarm) down to the JPEG quality factor of 30. It can withstand some signal-processing

Table 9 Performance of Gwo-Lu-Liao's semifragile watermarking on the basis of Haar functions. The quantization step sizes Δ are proportional to the visibility threshold values for all levels. Level 1 is not watermarked. Overall, 1,310,720 4×4 blocks were tested.

Forgery attack Probability of miss P_m		Signal-processing attacks Probability of false alarm P_f							
dB	Substitution	No attack	Smooth	Sharpen	1% S and P	Histog. equaliz.	35-dB AWGN	JPEG 70	Random errors
38	38.2%	1.0%	12.5%	41.2%	2.9%	41.3%	0.5%	0.5%	0.8%
41	26.2%	3.6%	28.7%	62.7%	33.6%	63.1%	3.0%	2.5%	4.2%

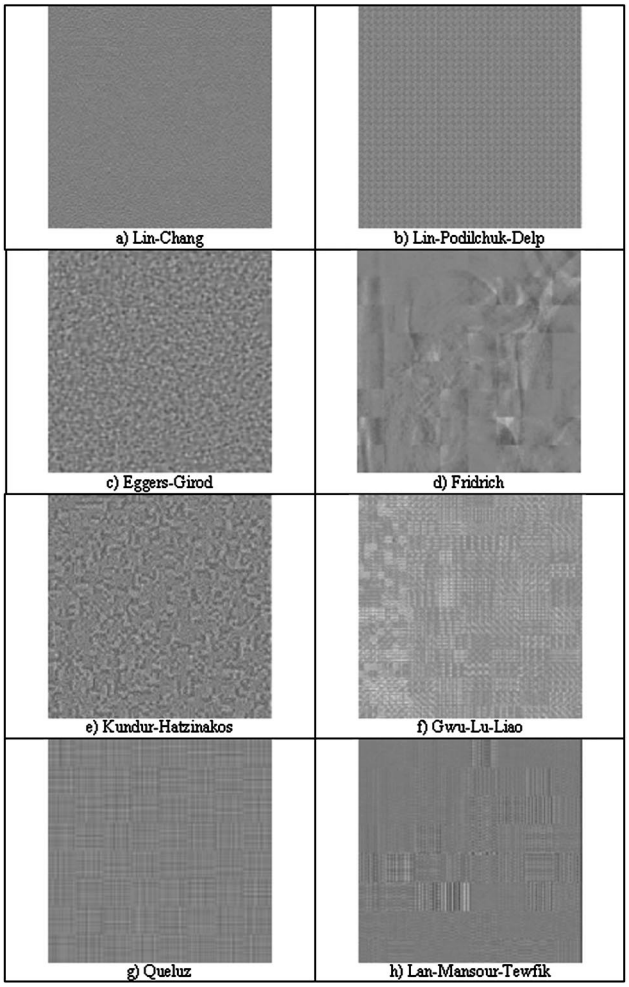

Fig. 1 Illustration of the watermarking residuals (watermarked image minus the original one).

operations but, for example, it is very sensitive to histogram equalization and smoothing. The visual distortion becomes, however, unacceptable just below the 38-dB document-to-watermark ratio.

- Fridrich's algorithm: This algorithm also provides a statistically meaningful measure of tamper probability. It is reasonably robust against signal-processing attacks. Its computational load is high relative to other algorithms.
- Kundur-Hatzinakos' algorithm: It has good JPEG performance, but otherwise it is weak against most signal-processing operations. In Kundur's algorithm, it becomes rather difficult to set a threshold for integrated 64×64 performance, since the distributions of the number of faulty subblocks under the forgery and signal-processing attacks are largely overlapping. This algorithm yields some notion on the scale of the attack.
- Queluz's algorithm: It is a surprisingly robust algorithm against many signal-processing operations, for example, with JPEG as it can withstand down to quality factor 10. It is sensitive to histogram equalization and salt-and-pepper noise. We have observed that smoothing the random bases improves the performance slightly.
- Lan-Mansour-Tewfik's algorithm: This technique was very fragile against all signal-processing operations due to the fact that the projections of DCT coefficients onto Hadamard columns were sometimes very small. Better results might be achieved by the dithered quantization. The visual distortion becomes very annoying even at 38 dB.
- Gwo-Lu-Liao's algorithm: This algorithm follows more or less the performance pattern of Kundur's algorithm, though in some items, notably, in the probability of miss, it is inferior.

Table 10 Experimental 1% false alarm thresholds to compute the 64×64 integrated performance.

Lin-Chang	A 64×64 block is tampered if seven or more of the 64 8×8 blocks fail.
Lin-Podilchuk-Delp	A 64×64 block containing 16 16×16 blocks is tampered if five or more fail.
Eggers-Girod	A 64×64 block is tampered if the likelihood ratio, integrated over $8\times64=512$ DCT coefficients, exceeds 0.5.
Fridrich	A 64×64 block is tampered if eight or more of its hash bits extracted differ from the embedded ones in the spread spectrum algorithm and/or if weighted correlation of the DCT indices and watermark pattern falls below 32%.
Kundur-Hatzinakos	A 64×64 block is tampered if 80 or more of the 256 4×4 blocks at resolution level zero fail.
Gwo-Lu-Liao	A 64×64 block is tampered if 80 or more of the 256 4×4 blocks at resolution level zero fail.
Queluz	A 64×64 block is tampered if four or more of the tested 20 row or column triads fail.
Lan-Mansour-Tewfik	A 64×64 block is tampered if three or more of the ten authentication bits embedded in projections are wrong.

One can note that, as far as miss probability is concerned, all of them perform better than 1% (except Gwo-Lu-Liao's). On the other hand, they tend to differ widely in the false alarm rate among signal-processing attack types. We can prepare a scorecard for the algorithms based on their robustness against the signal-processing manipulations. As it was done in the case of Stirmark,[33] a score of 1 is given whenever the false alarm rate is below 5%, and 0 otherwise. It appears then that, based on the previous P_F and P_M thresholds, the Fridrich, Eggers-Girod, and Lin-Podilchuk-Delp algorithms obtain the highest scores.

5 Conclusions

The comparative simulation experiments reveal that almost all algorithms (except Gwo-Lu-Liao and Lan-Mansour-Tewfik) do well on detecting the substitution forgery, and have also low false-alarm probability under the no-attack situation. On the other hand, it is interesting to observe they are all more or less sensitive to image smoothing opera-

Table 11 False alarm and miss probabilities on the basis of 64×64 pixel blocks. A block is declared tampered if the number of its native blocks that appears tampered exceeds a critical threshold. Smoothing scores are the average of the 3×3 median and convolution operations. **PSNR**=41 dB.

Semifragile method	Forgery attack P_{miss}	Signal-processing attacks P_f						
		No attack	Smooth	Histog. equal.	S and P 1%	AWGN 35 dB	JPEG 70	Sharpen
Chang	0.0%	0.0%	100%	99%	100%	32.3%	0.0%	100.0%
Delp	0.1%	2.3%	54.5%	3.4%	6.5%	2.7%	2.4%	0.3%
Eggers	0.0%	0.0%	41.4	91%	2.6%	0.0%	0.0%	65.6%
Fridrich	1.0%	1.6%	62.0%	5.5%	19.5%	2.5%	25.8%	21%
Kundur	0.1%	0.0%	77.7%	99.5%	51.9%	10.0%	2.9%	98.1%
Queluz	0.01%	0.01%	27.8%	94.3%	42.7%	0.01%	0.01%	100%
Liao	8.7%	3.0%	34.3%	80.7%	43.3%	1.7%	1.5%	79.9%

tions. Since the successful performance of an algorithm depends entirely on the context of its application, the following remarks can be made on the merit points. The merit points we consider are: the claim to semifragility, that is, resisting selected signal operations; and giving statistically sound figures of tamper probability.

The tamper indication must be statistically sound. In this respect, the Fridrich, Eggers-Girod, and Lin-Podilchuk-Delp (the latter under Gaussian assumption) algorithms output statistically sound tamper probability, while the others indicate tampering only on the basis of the count of bit errors between the test and actual sequences.

The tamper detection block size should be flexible, as the attack size can vary from a line consisting of a few tens of pixels long to the entire image plane itself. In this respect, Fridrich's and Lan-Mansour-Tewfik's algorithms are the least flexible, as they cannot function reliably for sizes below 64×64. Also, the Lin-Podilchuk-Delp algorithm necessitates 16×16 block size for operation, which is larger than the minimum sizes of all the rest.

There is another aspect to the block size, which is the ease with which the tamper indication of smaller blocks can be integrated to the tamper indication of a larger block, if we suspect that an image region larger than the native size of the algorithm has been affected. Thus if the native size is 8×8, one would like to predict the tamper likelihood over a larger region, say 32×32 or 64×64. When integrating scores, the sporadic native block alarms not due to malicious attack tend to be averaged out, while consistent evidences in the small blocks give a more reliable indication. In this respect, the Eggers-Girod algorithm provides the most straightforward formula for integration, as it simply sums the tamper/no-tamper likelihood of the coefficients subtended by the block. The Lin-Podilchuk-Delp algorithm can also be easily run over different sized blocks.

Semifragility, that is, robustness against selected signal-processing attacks, could be a desirable aspect, while we quickly point out that when and where a signal manipulation is considered as malicious or innocent depends on the application context. When algorithms are compared with respect to their robustness against signal-processing operations, using the scoring method suggested at the end of Sec. 4, one can conclude that the Eggers-Girod, Fridrich, Lin-Podilchuk-Delp algorithms perform uniformly well to a reasonable extent.

Notice that there could be other merit criteria, such as an autocorrecting capability, the ease with which a method can be attacked, etc.

In conclusion, we provide a comparative performance analysis of semifragile algorithms in detecting forgery at-

Table 12 False alarm and miss probabilities on the basis of 64×64 pixel blocks. A block is declared tampered if the number of its *n* subblocks that appears tampered exceeds a critical threshold. Smoothing scores are the average of the 3×3 median and convolution. **PSNR**=38 dB.

Semifragile method	Forgery attack P_{miss}	Signal-processing attacks P_f						
		No attack	Smooth	Histog. equal.	S and P 1%	AWGN 35 dB	JPEG 70	Sharpen
Chang	0.0%	0.0%	100%	100%	100%	0.0%	0.0%	100.0%
Delp	0.2%	0.2%	37.1%	0.5%	1.2%	0.5%	0.3%	0.3%
Eggers	0.0%	0.0%	25.3%	87.3%	0.9%	0.0%	0.0%	75.0%
Fridrich	0.6%	1.1%	43%	3.1%	11.4%	1.1%	5.0%	20.0%
Kundur	0.1%	0.0%	68.2%	98.9%	31.7%	3.5%	0.6%	95.9%
Queluz	0.01%	0.01%	15.5%	87.6%	7.8%	0.01%	0.01%	99.5%
Liao	16.1%	0.2%	12.9%	59.4%	2.0%	0.3%	0.1%	57.7%
Tewfik	16.9%	1.4%	83.5%	82.7%	82.6%	85.5%	76.1	81.4%

tempts in the guise of substitution attack and in being semifragile, that is, not giving false alarms in the face of selected mild signal-processing operations. While in this comparison no specific application context was envisaged, the simulation results would hopefully guide a designer to select the algorithms according to specific fragility and robustness characteristics, as dictated by the application context.

References

1. G. Coatrieux, B. Sankur, and H. Maitre, ''Strict integrity control of biomedical images,'' *Proc. SPIE* **4314**, 229–240 (2001).
2. M. M. Yeung and F. C. Mintzer, ''Invisible watermarking for image verification,'' *J. Electron. Imaging* **7**(3), 578–591 (July 1998).
3. E. T. Lin and E. J. Delp, ''A review of fragile image watermarks,'' *Proc. Multimedia and Security Workshop (ACM Multimedia '99) Multimedia Contents*, Orlando, pp. 25–29, October 1999.
4. J. Fridrich, ''Methods for tamper detection in digital images,'' *Proc. ACM Workshop on Multimedia and Security*, pp. 19–23, Orlando, FL, October 30–31, 1999.
5. L. M. Marvel, C. Boncelet, Jr., and C. T. Retter, ''Spread spectrum image steganography,'' *IEEE Trans. Image Process.* **8**(8), 1075–1083 (Aug. 1999).
6. L. M. Marvel, G. W. Hartwig, Jr., and C. Boncelet, Jr., ''Compression compatible fragile and semifragile tamper detection,'' *Proc. SPIE* **3971**, 131–139 (2000).
7. J. J. Eggers, J. K. Su, and B. Girod, ''A blind watermarking scheme based on structured codebooks,'' IEEE Conf. Secure Images and Image Authentication, London, April 10, 2000.
8. J. Fridrich, ''Robust digital watermarking based on key-dependent basis functions,'' *2nd Information Hiding Workshop*, Portland, Oregon, April 15–27, 1998.
9. E. T. Lin, C. I. Podilchuk, and E. J. Delp, ''Detection of image alterations using semi-fragile watermarks,'' *Proc. SPIE* **3971**, 152–163 (2000).
10. C. Y. Lin and S. F. Chang, ''Semifragile watermarking for authentication JPEG visual content,'' *Proc. SPIE* **3971**, 140–151 (2000).
11. M. P. Queluz, ''Content-based integrity protection of digital images,'' in *Security and Watermarking of Multimedia Contents*, P. W. Wong and E. J. Delp, Eds., *Proc. SPIE* **3657**, 85–93 (1999).
12. J. Fridrich and M. Goljan, ''Protection of digital images using self embedding,'' *Symp. Content Security and Data Hiding in Digital Media*, New Jersey Institute of Technology, May 14, 1999.
13. T.-H. Lan and A. H. Tewfik, ''Fraud detection and self embedding,'' *ACM Multimedia* **2**, 33–36 (1999).
14. D. L. Robie and R. M. Mersereau, ''Video error correction using steganography,'' *Appl. Signal Processing* **2002**(2), 164–173 (Feb. 2002).
15. C.-Y. Lin and S.-F. Chang, ''A robust image authentication algorithm surviving JPEG compression,'' *Proc. SPIE* **3312**, 296–307 (1997).
16. J. Fridrich, ''Combining low frequency and spread spectrum watermarking,'' *Proc. SPIE* **3456**, 2–12 (1998).
17. J. Fridrich, ''Security of fragile authentication watermarks with localization,'' *Proc. SPIE* **4675**, 691–700 (2002).
18. J. Fridrich, ''A hybrid watermark for tamper detection in digital images,'' *ISSPA'99 Conf.*, pp. 301–304, Brisbane, Australia, August 22–25, 1999.
19. L. Xie, G. R. Arce, and R. F. Graverman, ''Approximate message authentication codes,'' *IEEE Trans. Multimedia* **3**, 242–252 (June 2001).
20. K. L. Hung, C. C. Cheng, and T. S. Chen, ''Secure discrete cosine transform based technique for recoverable tamper proofing,'' *Opt. Eng.* **40**(9), 1950–1958 (Sep. 2001).
21. J. J. Eggers and B. Girod, ''Blind watermarking applied to image authentication,'' ICASSP'2001: Intern. Conference on Acoustics, Speech and Signal Processing, Salt Lake City, USA, May 7–11, 2001.
22. M. P. Queluz and P. Lamy, ''Spatial watermark for image verification,'' *Proc. SPIE* **3971**, 120–130 (2000).
23. M. P. Queluz, ''Spatial watermark for image content authentication,'' *J. Electron. Imaging* **11**(2), 275–285 (April 2002).
24. D. Kundur and D. Hatzinakos, ''Towards a telltale watermarking technique for tamper-proofing,'' *Proc. IEEE Int. Conf. On Image Processing*, **2**,409–413 (Oct. 1998).
25. D. Kundur and D. Hatzinakos, ''Digital watermarking for telltale tamper-proofing and authentication,'' *Proc. IEEE Special Issue on Identification and Protection of Multimedia Information* **87**(7), 1167–1180 (July 1999).
26. T. H. Lan, M. F. Mansour, and A. H. Tewfik, ''Robust high capacity data embedding,'' *ICASSP 2001*, Utah, April 2001.
27. G.-W. Yu, C.-S. Lu, and H.-Y. M. Liao, ''Mean quantization-based fragile watermarking for image authentication,'' *Opt. Eng.* **40**(7), 1396–1408 (2001).
28. O. Ekici, B. Coşkun, U. Naci, and B. Sankur, ''Comparative assessment of semifragile watermarking methods,'' *Proc. SPIE* **4518**, 177–188 (2001).
29. M. Kutter and F. A. P. Petitcolas, ''Fair evaluation methods for image watermarking systems,'' *J. Electron. Imaging* **9**(4), 445–455 (Oct. 2000).
30. C. Y. Lin and S. F. Chang, ''A robust image authentication method distinguishing JPEG compression from malicious manipulation,'' *IEEE Trans. Circuits Syst. Video Technol.* **11**(2), 153–168 (2001).
31. M. P. Queluz, ''Authentication of digital images and video: generic models and a new contribution,'' *Signal Process. Image Commun.* **216**(5), 461–475 (2001).
32. H. Bassali, J. Chugani, S. Agarwal, A. Aggarwal, and P. Dubey, ''Compression tolerant watermarking for image verification,'' *Proc. ICIP'2000: International Conf. on Image Processing*, Vancouver, Canada, Sep. 10–13, 2000.
33. F. A. P. Petitcolas, ''Watermarking schemes evaluation,'' *IEEE Signal Processing* **17**(5), 58–64 (Sep. 2000).
34. G.-J. Yu, C.-S. Lu, H.-Y. M. Liao, and J.-P. Sheu, ''Mean quantization blind watermarking for image authentication,'' *Proc. 7th IEEE Int. Conf. on Image Processing*, Vol. III, pp. 706–709, Vancouver, BC, Canada, Sept. 10–13, 2000.

Özgür Ekici received his BS degree in electrical engineering from Boğaziçi University, Istanbul, Turkey, in 2001, and MS degree from university of Ottawa, School of Information Technology and Engineering, Canada, in 2003. His areas of current interest include coding theory and digital transmission techniques, in particular for mobile communication channels.

Bülent Sankur has received his BS degree in electrical engineering at Robert College, Istanbul, and completed his MSc and PhD degrees at Rensselaer Polytechnic Institute, New York. He has been active at Boğaziçi University in the Department of Electric and Electronic Engineering in establishing curricula and laboratories, and guiding research in the areas of digital signal processing, image and video compression, and multimedia systems. He was the chairman of the International Telecommunications Conference and the technical co-chairman of ICASSP'2000. He has held visiting positions at the University of Ottawa, Canada: Istanbul Technical University: Technical University of Delft, The Netherlands: and Ecole Nationale Supérieure des Telecommunications, France.

Barış Coşkun received his BS degree in 2001 and is presently pursuing his MSc studies in electrical engineering at Boğaziçi University, Istanbul, Turkey. His areas of current interest include multimedia signal processing and multimedia networking.

Umut Naci has graduated from Boğaziçi University, Department of Electrical and Electronics Engineering, Istanbul in 2001 and is currently pursuing his master's degree in the same university. At the same time, he is a research engineer at GVZ Speech Technologies Company. His research interest are speaker verification systems, embedded speech recognition technologies and real time image processing.

Mahmut Akcay graduated from Bilkent University, Department of Physics, in 2000, and completed his master's degree at the Université de Claude-Bernard, Lyon, France. He is presently pursuing his doctoral studies in a joint program between Boğaziçi University (Turkey) and the Université de Claude-Bernard, Laboratoire LIRIS. His research interest are in the areas of multimedia signal processing and multimedia networking.