Combined Hashing/Watermarking Method for Image Authentication

Vlado Kitanovski, Dimitar Taskovski, and Sofija Bogdanova

***Abstract*—**In this paper we present a combined hashing/watermarking method for image authentication. A robust image hash, invariant to legitimate modifications, but fragile to illegitimate modifications is generated from the local image characteristics. To increase security of the system the watermark is generated using the image hash as a key. Quantized Index Modulation of DCT coefficients is used for watermark embedding. Watermark detection is performed without use of the original image. Experimental results demonstrate the effectiveness of the presented method in terms of robustness and fragility.

***Keywords*—**authentication, blind watermarking, image hash, semi-fragile watermarking

I. Introduction

A digital image watermark is a perceptually invisible message embedded in the image. This embedding is done by an encoder using a secret key. The watermark can carry information about the owner or recipient of the image, the image itself or some additional information (image caption, image date etc.). The watermarked image may undergo many possible changes by users or attackers: unintentional modifications and malicious attacks that aim to disable watermark detection. Ideally, the watermark must resist modifications and attacks as long as they result in images that are perceptually similar. The watermark detector should decide whether a watermark is present in the image or not. If the original image is used in making the decision, the detector's efficiency is increased and this system is called a *private* or *non-oblivious watermarking system.* Private watermarking systems are usually expensive in terms of storage. In contrast, the detectors in *public* or *blind watermarking systems* have no access to the original image. Blind watermarking systems are more practical but the detector is less efficient than in private watermarking systems.

Watermarks can be broadly classified into two types: *robust* and *fragile* (or *semi-fragile*) watermarks. Robust watermarks are generally used for copyright protection or owner identification because they are robust to many kinds of image processing operations. Fragile or semi-fragile watermarks are mainly used for image authentication and integrity verification because they are fragile to certain modifications. Semi-fragile watermarks are designed so that they can survive unintentional and legitimate modifications, but they become undetectable after malicious and illegitimate modifications, such as adding, replacing or removing objects from the image [1].

Many digital watermarking schemes have been proposed for image authentication. A nice overview of several semi-fragile methods is presented in [2]. A blind watermarking method for this application is essential, so that the watermark *W* depends on a secret key *k* and on the image *I*. It is important that the dependence on the key be sensitive, while the dependence on the image be robust [3]:

1. *W(k,I)* is uncorrelated with *W(k,I')* whenever images *I* and *I'* are dissimilar;
2. *W(k,I)* is strongly correlated with *W(k,I')* whenever *I* and *I'* are similar (*I'* is the image *I* after lossy compression or some spatial domain attacks);
3. *W(k,I)* is uncorrelated with *W(k',I)* for *k≠k'*.

Requirements 1-3 could be satisfied if a robust image hash function is used for watermark generation. An image hash function [4] produces a bit-string (the image hash) that is the same (or almost the same) for all images I' that are perceptually similar to I, while at the same time, two completely different images produce two uncorrelated hash strings.

From the security aspect, if multiple images are marked using the same key, blind watermarking schemes present security weaknesses [5]. At the same time, copy attacks [6] can seriously compromise the integrity of the system. In order to combat these attacks and achieve stronger security, it is desirable to use image-dependent keys [7].

In this paper we present a blind watermarking method for image authentication. Using a secret key we generate a robust image hash from local image characteristics. This image hash is invariant with respect to legitimate modifications that do not change the visual content of the image, but is fragile with respect to illegitimate modifications. The watermark is generated using the image hash as a key in order to increase security. Owing to this relation between the watermark and the image hash, it is possible for the detector to differentiate

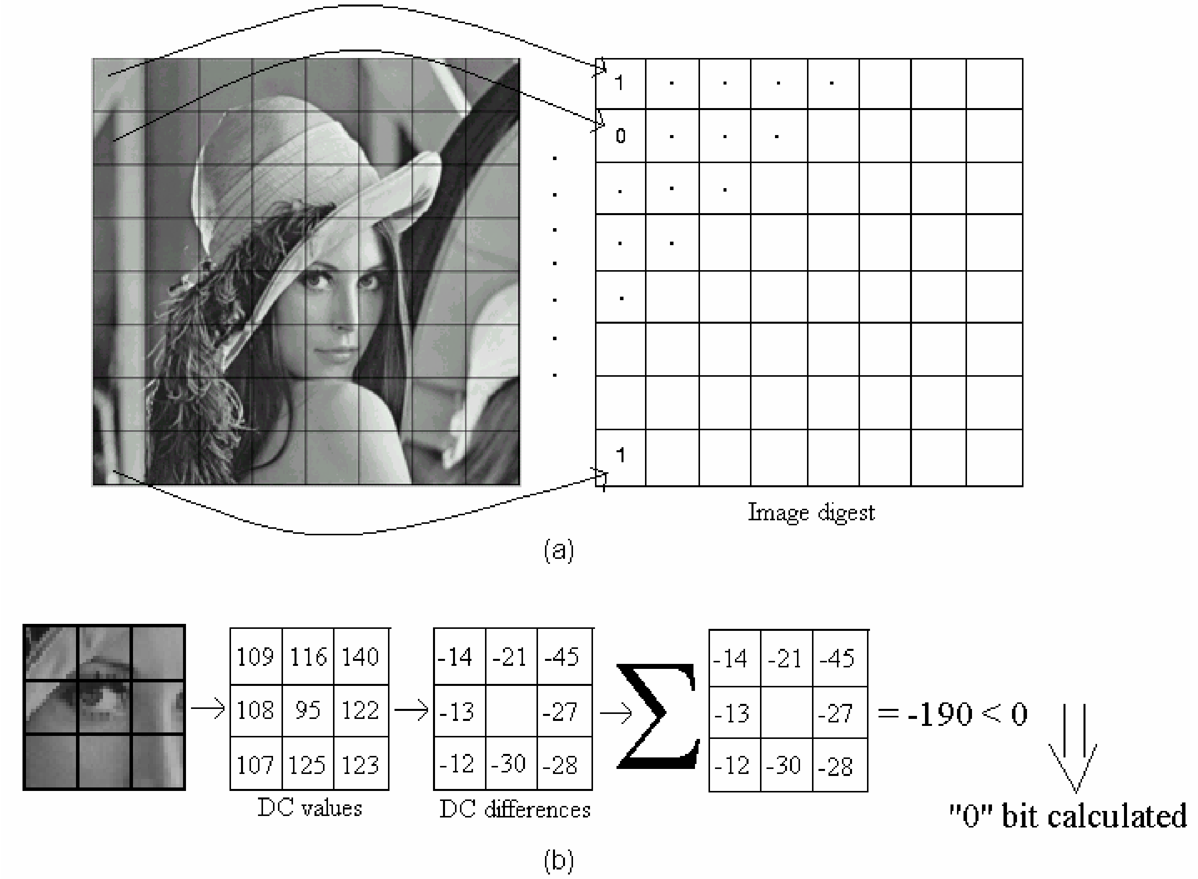
Manuscript received 17 February, 2006.

Vlado Kitanovski is with the Faculty of Electrical Engineering, Ss. Cyril and Methodius University - Skopje, R. Macedonia (phone: +389-2-3099107; fax: +389-2-3064262; e-mail: vlade@etf.ukim.edu.mk).

Dimitar Taskovski is with the Faculty of Electrical Engineering, Ss. Cyril and Methodius University - Skopje, R. Macedonia (e-mail: dtaskov@etf.ukim.edu.mk).

Sofija Bogdanova is with the Faculty of Electrical Engineering, Ss. Cyril and Methodius University - Skopje, R. Macedonia (e-mail: sofija@etf.ukim.edu.mk).

among legitimate and illegitimate modifications of the image.



109	116	140
108	95	122
107	125	123

-14	-21	-45
-13		-27
-12	-30	-28

Fig. 1 (a) Calculating image digest from the image. (b) Calculating the image digest bit for the middle block

For example, if the image is tampered, the watermark detector will compute an image hash that is similar (not identical) to the hash of the original image, but this difference will lead to wrong watermark regeneration in the detector and with high probability, negative authentication. Quantized Index Modulation of DCT coefficients is used to embed the watermark. Watermark detection is performed without using the original image. We present experimental results relating to robustness against some common image processing operations and fragility to tampering.

II. Watermarking method

A. Watermark generation

For the purposes of selective authentication deployed by a blind watermarking system, the watermark should be content dependent. So, it is calculated from an image hash that is invariant with respect to common processing operations.

First, a bit string called the image digest is calculated from the local image characteristics [8]. The image is divided into $M\times M$ blocks, and differences between the block's DC values are used to form the image digest. The use of DC values provides the invariance of the computed digest to high frequency modifications. The length of the image digest is the number of $M\times M$ blocks in the image, that is, one bit is extracted from every block. More specifically, one bit for the image digest, h_i, is derived from the i-th block as follows:

$$s_i = \sum_{j=1}^{8}(DC_i - DC_j) \geq 0$$

$$h_i = \begin{cases} 1, & s_i \geq 0 \\ 0, & s_i < 0 \end{cases} \qquad (1)$$

where j indexes the eight blocks that are neighbors to the i-th block. This is shown on Fig. 1.

The image digest is then coded using a secret key k to obtain an image hash. The watermark W is obtained by additional bit-sensitive-like coding of the image hash, as shown on Fig. 2. We perform this coding by using the image hash as a key to set the state of a uniform pseudo-random generator that generates the watermark W. This additional coding implies that two different image hashes, different even in a single bit, will be

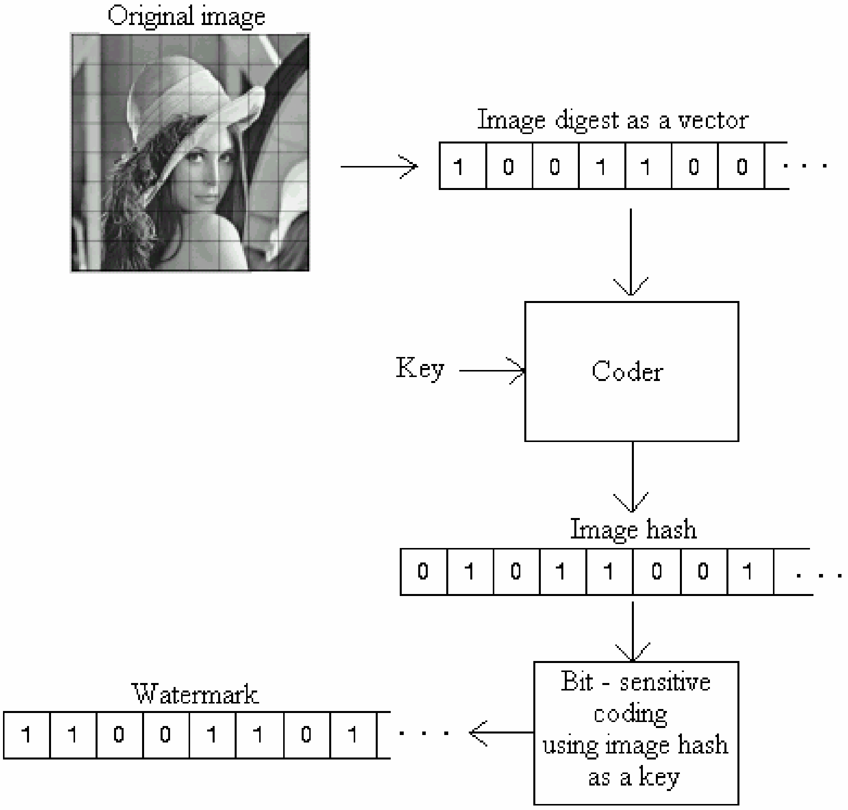


Fig. 2 Watermark generation

coded into two different and statistically independent watermarks. As described in section II.C, this coding enables the detector to differentiate between legitimate and illegitimate modifications of the image.

B. Watermark embedding

The generated watermark is embedded in the low frequency components of the *P*×*P* DCT blocks obtained after applying a block-based DCT transform to the host image. Only one watermark bit is embedded in each DCT block. This embedding uses a simple form of the QIM [9]: every bit of the watermark is used to select a quantizer that will quantize specific DCT samples in each block. There are two quantizers: one that modulates the "ones" and another one that modulates the "zeros". These quantizers have same quantization steps, δ, and their levels of reconstruction are shifted one from another for $\delta/2$ (Fig. 3). The quantization step δ is maximized among all choices satisfies certain criteria for perceptual transparency.

The process of watermark embedding causes distortion to the host image that is not equally perceptible in all parts of the image. It is well known that the human eye is less sensitive to modifications in those areas where brightness is high or low than in areas with mid-range luminance. Also, the human eye is less sensitive to modifications in the highly textured areas than in the relatively flat areas. We use these two characteristics of the human visual system to identify the image areas where this modification can be easily hidden. In the relatively more sensitive blocks, the energy of the watermark should be relatively small, and moreover a small quantization step δ_1 is used. In the less sensitive areas the energy of the watermark can increase so a larger quantization step δ_2 is used. The sensitivity of the block is determined by computing the block's mean value and variance and setting appropriate thresholds.

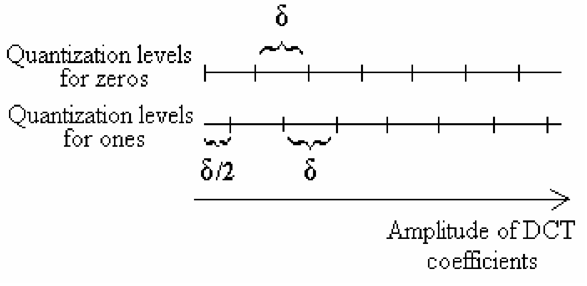

δ
Quantization levels for zeros
Quantization levels for ones
δ/2
δ
Amplitude of DCT coefficients

Fig. 3 Quantization levels of DCT coefficients

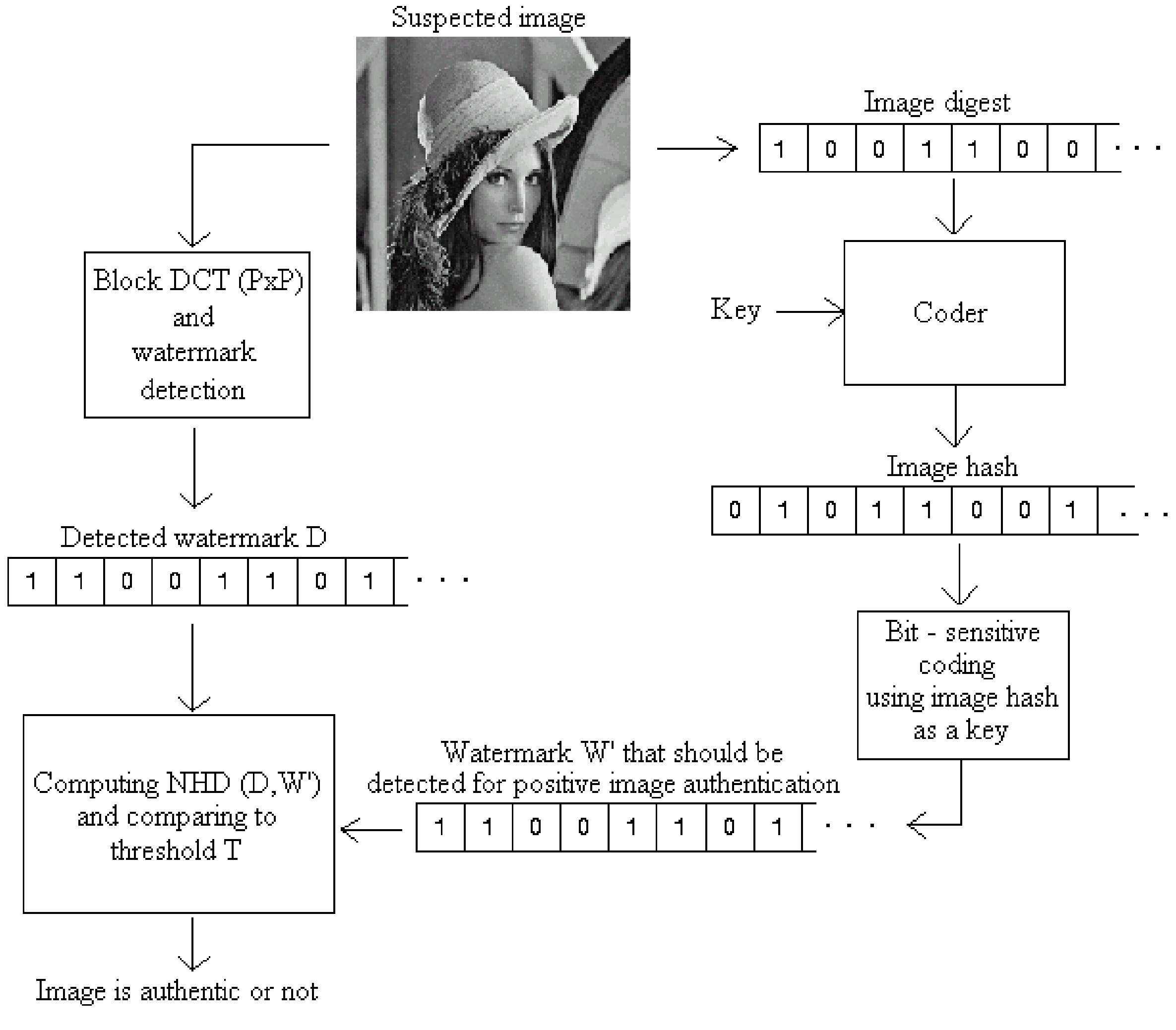

Suspected image
Image digest
1 0 0 1 1 0 0 . . .
Block DCT (PxP) and watermark detection
Key
Coder
Image hash
0 1 0 1 1 0 0 1 . . .
Detected watermark D
1 1 0 0 1 1 0 1 . . .
Bit - sensitive coding using image hash as a key
Watermark W' that should be detected for positive image authentication
Computing NHD (D,W') and comparing to threshold T
1 1 0 0 1 1 0 1 . . .
Image is authentic or not

Fig. 4 Watermark detection

To increase the robustness of the embedded bit within a block, a total of n DCT coefficients are quantized in each block. These coefficients belong to the low-frequency part of the block. All n coefficients are used to embed only one watermark bit. This means that one watermark bit is encoded with an n-bit sequence. We denote the coded bit-sequences for “0” and “1” by K_0 and K_1, respectively. K_0 and K_1 are chosen such that K_1 is the complement of K_0. For increased security, a small amount of noise is added to the quantized DCT coefficients, with a tendency to reduce distortion.

C. *Watermark detection*

Since the watermark is not known to the detector, it must be regenerated from the suspected image. It is important to calculate the watermark correctly. Using the secret key k, the image hash and the watermark W' are generated by detector in the same way as in the embedding process. If the watermarked image hasn’t been modified, the computed image hash will be same as the original image hash thus leading to correct watermark calculation. Also, if the watermarked image has been unintentionally modified (for instance, compressed or filtered), the computed image hash will be same as the original image hash due to the invariance of the image hash to the common image processing operations, and the watermark will be calculated correctly. However, if the watermarked image has been tampered with, the computed image hash will be different from the original and the regenerated watermark W' will be different and uncorrelated with the embedded watermark W.

Watermark detection begins with block-based DCT transform of the suspected image (Fig. 4). The detector computes the mean value and the variance for each block in order to determine the correct quantization step (δ_1 or δ_2). The detector also knows both quantizers, K_0, K_1 and the n DCT coefficients for extraction of the watermark bit within a block. Every coefficient from the selected n is quantized with the two quantizers. The quantizer that generates lower distortion determines the extracted bit from that coefficient. In this

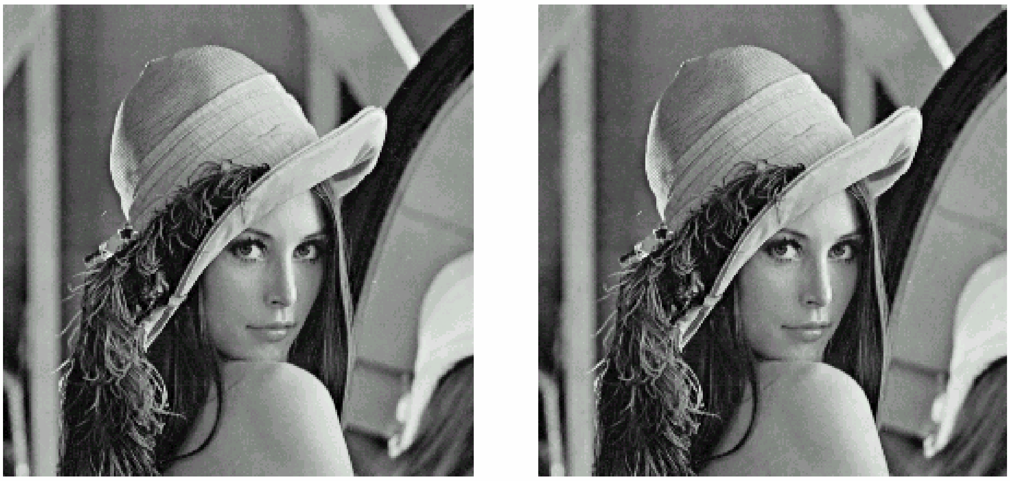


Fig. 5 Original image (left) and watermarked image (right)

TABLE I
VALUES OF P_P FOR DIFFERENT VALUES FOR T AND N. BLOCK DIMENSIONS ARE 48×48

N	121 (512×512)			180 (720×576)			352 (1024×768)		
T	0.15	0.20	0.25	0.15	0.20	0.25	0.15	0.20	0.25
P_p	10^{-16}	10^{-12}	10^{-8}	10^{-23}	10^{-17}	10^{-12}	10^{-44}	10^{-31}	10^{-22}

manner, an n-bit sequence S is extracted from each block. Decision about the embedded bit in the i-th block is made by comparing S_i with K_0 and K_1. If the Hamming distance $d_H(S_i, K_0)$ is smaller than $d_H(S_i, K_1)$ then a "0" bit is decided for the i-th block, and if $d_H(S_i, K_1) < d_H(S_i, K_0)$ then "1" is decided.After processing all blocks, a detected N-bit watermark D is obtained.

Generally, the detected watermark D is different from W (or W'). There are few reasons for this: first, if the watermarked image has been unintentionally modified, some changes to the DCT coefficients occur, and some of the watermark bits will be incorrectly recovered; and second, even if the watermarked image is brought to the detector right after the embedding, there is a probability that some of the bits will be incorrectly recovered. This is due to the distortion that is made during the watermark embedding which modifies the variance within a block and can cause incorrect quantizer selection in the detector. Therefore, a measure for similarity is required to decide for the presence of the watermark W' (which we presume is same as W). As a measure for similarity between D and W' we use normalized Hamming distance (*NHD*). The value of *NHD* is compared to a threshold T to decide whether image is authentic (that is, if the authentication watermark is present). The threshold is the maximum *NHD* that results in positive detection of W'. The selection of the value for T is restricted from the false positive probability P_p. If the suspected image is not watermarked, then W' is independent of D and there is a probability that some of their bits will be identical. P_p can be precisely calculated if we assume that D is independent from W' for a unwatermarked image. In that case, the variable k, which is number of equal bits between W' and D, has a binomial distribution $P_k(k,N,p)$, where $p = 0.5$ and N depends of the image dimensions and block size for watermark embedding. P_p equals the cumulative sum:

$$P_p = F_k(T,N) = \sum_{k=0}^{T} P_k(k,N,p) \qquad (2)$$

According to (2), different values of P_p are given in Table I for some typical image dimensions and 48×48 blocks.

III. EXPERIMENTAL RESULTS

To evaluate the performance of the presented method, several standard 512×512 images were used for watermark embedding and detection. All images are divided into 64 blocks of size 64×64 for image hash computing. Then, the watermark is computed and embedded bit-by-bit in 48×48 blocks. Table II shows the PSNR values for the watermarked images. Although, PSNR is not a very effective prediction of perceptual quality, there is no doubt that high PSNR ensures excellent quality of the watermarked images. An example of the perceptual transparency of the watermark is shown in Fig 5. To check the robustness of our method, we performed several attacks on the watermarked image, including JPEG compression and spatial domain attacks. To check the ability of tamper detection we performed several experiments of random block substitutions from the test images.

In all experiments, we choose T to be 0.20 which guarantees

TABLE II
NHD VALUES AFTER WATERMARK DETECTION IN THE ATTACKED WATERMARKED IMAGES

Image	Lenna	barbara	boat	goldhill	Baboon	peppers
PSNR (dB) of the watermarked image	49.38	48.51	45.79	49.81	38.26	45.58
Unattacked image	0	0	0	0	0	0.008
min. *NHD* when using 500 wrong keys	0.380	0.384	0.377	0.372	0.384	0.376
JPEG 100%	0	0	0	0	0	0.008
JPEG 80%	0	0	0	0.008	0.024	0
JPEG 50%	0	0.008	0.008	0	0.041	0
JPEG 30%	0.041	0.025	0.041	0.049	0.091	0.008
average 3×3 filter	0.016	0.049	0.041	0.024	0.140	0.041
Wiener 3×3 filtering	0	0.033	0.008	0.016	0.041	0.016
Median 3×3 filtering	0	0.066	0.041	0.041	0.190	0.033
Gaussian blur with radius of one pixel	0.066	0.181	0.140	0.124	0.446	0.107
Horizontal motion blur, 4 pixels	0.082	0.124	0.115	0.115	0.496	0.107
unsharp contrast enhancement filter	0.074	0.289	0.173	0.190	0.512	0.264
Gaussian noise, PSNR = 35 dB	0	0	0.008	0	0.008	0.008
Gaussian noise, PSNR = 31 dB	0.074	0.024	0.066	0.090	0.049	0.058
Salt & Pepper noise, PSNR = 31 dB	0.082	0.058	0.099	0.124	0.058	0.049
Speckle noise, PSNR = 31 dB	0.107	0.008	0.066	0.107	0.099	0.066
Brightness +15%	0.016	0.024	0.008	0.041	0.454	0.016
Contrast +15%	0.033	0.082	0.049	0.132	0.446	0.107

low P_p of 10^{-12}. On the other hand, we let 20% of the recovered watermark to be incorrect due to common image processing operations; in these cases, authentication was still positive.

A. Detection in the original watermarked image

If no attacks are applied to the watermarked image, the *NHD* value of detection should ideally be zero. In our case there is a small probability of incorrect quantization step selection in the detector, so *NHD* values can be different from zero, as it is case for the peppers image. The results for all images are shown in Table II.

B. Detection with a wrong key

If a wrong key is used for image hash computation in the detector, *NHD* value should be above the threshold resulting in negative watermark detection or negative image authentication. The minimum of 500 *NHD* values obtained after detection with 500 wrong random generated keys for every image are shown in Table II.

C. JPEG compression attacks

In this experiment we performed JPEG compression on the watermarked images with different quality factors (*QF*): *QF*=100%, 80%, 50% and 30%. *NHD* values are given in Table II. From these results we can see that the presented method is robust to JPEG compression. For cases that *QF* is larger than 80%, the extracted watermarks are almost identical with the embedded ones. For all other cases, *NHD* remains below the threshold.

D. Spatial domain attacks

In this experiment we performed standard spatial attacks including average filtering, Wiener filtering, median filtering, blurring, sharpening, brightness/contrast alterations and attacks with additive and multiplicative noise. The results are shown in Table II. For all images, except for Baboon image, the presented method is robust to most standard types of lowpass filtering, additive and multiplicative noise down to PSNR of 31 dB, brightness and contrast alterations up to 15%. High *NHD* values for the Baboon image are caused by wrong image hash computation for the attacked image. This image hash differs from the original image hash in a single bit, which leads to completely different generated watermarks in the detector and therefore high *NHD* value.

E. Tamper detection

The performance of the presented method for tamper detection was measured in terms of miss probability P_M, which is the percentage of maliciously attacked images that do not generate any tampering alarm. Malicious attacks were simulated by random block substitution between images. Several block dimensions were used. The results given in Table III show that our method detects tampered areas with dimensions 128×128 or bigger with high probability.

IV. CONCLUSION

In this paper we presented a blind watermarking method for image authentication. The watermark is generated using a robust image hash as a key in order to increase security. Compared to the semifragile watermarking methods described in [2], experimental results for our method show larger PSNR for the watermarked image, as well as increased robustness to: spatial linear and non-linear filtering, attacks with additvie or multiplicative noise and brightness/contrast alterations. As further work, we envision improvements of this method in order to decrease the miss probability for smaller tampered areas, and also enabling localization, i.e. the ability to localize the tampered area.

TABLE III
VALUES OF P_M FOR TAMPERED AREAS WITH DIFFERENT DIMENSIONS

Dimension of the tampered area	64×64	96×96	128×128
P_M [%]	36.67	13.33	0

REFERENCES

[1] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers Inc., San Francisco, 2001.

[2] O. Ekici, B. Sankur, B. Coskun, U. Naci, and M. Akcay, "Comparative evaluation of semifragile watermarking algorithms," *Journal of Electronic Imaging* , vol 13, No. 1, pp. 209– 216 January 2004.

[3] J. Fridrich, "Visual hash for oblivious watermarking," *Proc. of IS&T/SPIE's 12th Sym. on Electronic Imaging*, Vol. 3971, USA, Jan 2000.

[4] R. Venkatesan, S. M. Koon, M. H. Jakubowski and P. Moulin, "Robust image hashing," *IEEE Proc. ICIP*, Vol. 3, pp. 664 - 666, Sep 2000.

[5] M. Holliman, N. Memon, and M. Yeung, "On the need for image dependent keys in watermarking," in *Proc. 2nd Workshop on Multimedia,* Newark, NJ, 1999.

[6] M. Kutter, S. Voloshinovskij, and A. Herrigel, "The watermark copy attack," *Proc. SPIE*, vol. 3657, pp. 226–239, 1999.

[7] J. Cannons, and P. Moulin, "Design and statistical analysis of a hash-aided image watermarking system," *IEEE Trans. Image Processing,* vol. 13, No. 10, pp. 1393-1408, Oct. 2004

[8] D. K. Roberts, "Security camera video authentication", *10th IEEE Digital Signal Processing Workshop*, Pine Mountain, Georgia, USA, October 13-16, 2002.

[9] B. Chen and W. Wornell, "Quantization Index Modulation: A class of provably good methods for digital watermarking and information embedding", *IEEE Trans. Information Theory,* vol. 47, No. 4, May 20