CONTENT-ADAPTIVE SEMI-FRAGILE IMAGE AUTHENTICATION BASED ON JPEG2000 COMPRESSION

Mathias Schlauweg and Erika Müller

Institute of Communications Engineering,
Faculty of Computer Science and Electrical Engineering, University of Rostock,
Richard-Wagner Str. 31, 18119, Rostock, Germany

ABSTRACT

In this paper, a new system is presented for authenticating image content using digital watermark embedding in the DWT-domain of JPEG2000. A semi-fragile signature, generated from the host signal, is embedded into the image for content verification. Generation as well as embedding of the signature is adapted to the image content for performance improvement. The system is tested extensively and performance results are compared to those of methods proposed by other authors. We show that our new system outperforms the methods compared to. Our semi-fragile authentication is robust against non-malicious modifications, such as lossy compression, noise, image blurring and sharpening, changes of luminance and contrast as well as scaling. But in contrast to other methods, our authentication is secure at the same time, which we proof by different forgery attacks.

Index Terms — Semi-fragile image authentication, digital watermarking, JPEG2000, texture-based segmentation

1. INTRODUCTION

The change from classical analogue to digital photography led to several advantages and new applications, in the last years. Images and video can be generated easier and with higher quality than ever before. Distribution and duplication of digital content is possible without loss of quality. No special knowledge or expensive tools are required for post-processing multimedia data.

But these innovations can also yield unpleasant disadvantages. Images can be manipulated very easily. For example, every year, there is a spectacular image content manipulation revealed in any famous print media. The repertoire reaches from correction of small blemish to dramatization of war reporting or political campaigns. Hence, images and video are in a credibility crisis.

There are two countermeasures for exposing image content manipulations. In a first strategy, also known as *digital forensic*, experts are consulted looking for traces of image processing. For example, during image capture every camera produces its own fingerprints (e.g., inconsistencies in lightning, chromatic aberration, special noise pattern). If image objects are deleted or mixed from different images, these traces can catch expert's eye.

Another way of content verification is the embedding of *digital watermarks*. For embedding, the multimedia signal is slightly changed. At the verification side, these signal changes can be detected and thus the embedded information can be retrieved. By checking the correctness of the extracted watermark a user can infer easily if the image has been tampered with. This strategy is advantageous over the forensic approach, because no experts are needed and images can be used according to customs.

The aim is to allow admissible manipulations such as lossy compression or image enhancement, but to reject malicious manipulations that change the visual content.

In this paper, we present a secure authentication system based on digital watermarking, which is robust against a wide range of non-malicious image processing operations. In section 2, we formulate requirements that an authentication system should fulfil. In sections 3, we describe our new system in detail. Simulation results and comparisons with methods by other authors are given in section 4. Finally, section 5 concludes the work.

2. IMAGE AUTHENTICATION

As mentioned above, an authentication watermark should be robust against non-malicious image processing but fragile against image content attacks. This requirement is known as *semi-fragility*. For example, the watermark should be robust against lossy compression, noise addition, change of image-size, or image enhancement.

At the other hand, operations that must be detected are, for example, cropping, deleting, or merging (copy/paste) of image objects and operations that strongly affect perceptual image quality.

Next to semi-fragility, there are further important requirements for an authentication system. First, the watermark has to be generated depending on the content of the host image. This should avoid an attacker copying a valid

watermark to a manipulated image. Second, the content-dependent watermark should be signed using asymmetric encryption. A forger must be prevented from generating a new valid, signed watermark for a manipulated image. Third, for security reasons and for the purpose of easier applicability, during verification, the original image should not be needed. This property is known as *obliviousness*. Hence, everybody should be able to verify the integrity of an image.

Comparative overviews of different semi-fragile image authentication methods can be found in [1]-[4].

In this paper, we compare our new system with those methods given in [1] and results from other methods listed in there. These comparisons represent the current state-of-the-art concerning the above-claimed requirements.

3. PROPOSED AUTHENTICATION SYSTEM

The proposed system in this paper combines ideas from approaches in [9] and [10] extended by further normalizations and adaptations. The overall framework of our authentication watermarking system is demonstrated in Fig. 1.

First, the image is hashed to generate a content-dependent watermark. For this hashing not the gray-value pixels are used but semi-fragile features extracted from image content. Afterwards, the hash-value is signed and encoded using forward error correction. The resulting watermark is embedded within the image. At verification side, the watermark can be extracted, decrypted, and compared with the hash generated from the received image without any extra information from the embedding site.

All single steps of this framework are described in the following subsections in detail.

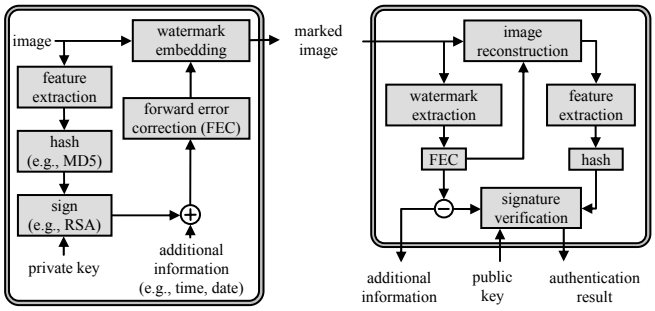

Fig. 1. Digital watermarking system for image authentication

3.1. Watermark generation and embedding

Our new authentication system is based on quantization of the coefficients of the host image in the *discrete wavelet domain* (DWT). It is directly integrated in the process of a JPEG2000 compression.

3.1.1. Construction of a secure image-dependent hash

If $x := \{x_j \in \mathbb{R} : 1 \le j \le J\}$ are the coefficients of an image in DWT-domain and q_j is a quantized value using quantizer $Q(\bullet)$ and step-size Δ, then $\hat{x}_j = Q^{-1}(q_j)$ is the reconstructed value of q_j, as in Eq. (1) and Eq. (2).

$$q_j = Q(x_j) = \operatorname{sign}(x_j)\left\lfloor \frac{|x_j|}{\Delta} \right\rfloor \tag{1}$$

$$\hat{x}_j = Q^{-1}(q) = \begin{cases} 0 & q = 0 \\ \operatorname{sign}(q)(|q| + 0{,}5)\Delta & q \neq 0 \end{cases} \tag{2}$$

In numerous simulations, we found out that if we quantize and, afterwards, hash all coefficients $x := \{x_n \in \mathbb{R} : 1 \le n \le N\}$ of the LL_4-subband of the DWT-decomposition a secure and robust image-dependent hash-value can be constructed.

As long as the quantized coefficients $\hat{x}$ after changes due to image processing operations or attacks remain within the range $[\Delta\mathbb{Z}; \Delta(\mathbb{Z}+1))$ they yield the same hash-value during verification. If a forger moves just one single LL_4-coefficient out of its quantization interval this manipulation can be detected and alarm is raised.

A digital signature is generated from the hash-value by the use of asymmetric encryption (e.g., *RSA*). We use a key-length of 512 bits. Additionally to the hash-value also time, date, etc. can be integrated to make the shot unique.

Afterwards, the signature is encoded using forward error correction. We apply *convolutional coding* with a code rate r = 1/2. Hence, the watermark $w := \{w_n \in \pm 1 : 1 \le n \le N\}$ to be embedded has a length of 1024 bits.

3.1.2. Signature embedding by quantization

For our semi-fragile authentication approach it is sufficient not to embed the signature watermark as robust as possible but as robust as necessary. That means, if an image processing operation or an attack yields a different hash-value during verification it doesn't matter if the signature can be extracted correctly. Signature and hash-value don't match, and hence, verification fails.

For that reason, we embed the data within the same host signal locations the signature is generated from, using scalar *dither modulation* [6]. Hence, the embedding locations are secured by the hash process in turn.

Since JPEG2000 applies quantization with dead-zone, our watermark embedding is adapted to this dead-zone as in Eq. (3), where y is the watermarked host signal.

$$y_n = \begin{cases} x_n & -\Delta/4 \le x_n \le \Delta/4 \\ & \text{and } w_n = +1 \\ \Delta \cdot \left(\operatorname{sign}(x_n) \left\lfloor \frac{|x_n|}{\Delta} \right\rfloor + w_n \cdot \Delta/4 \right) & \text{otherwise} \end{cases} \tag{3}$$

Data is embedded by quantizing every LL_4-coefficient to a closest quantization lattice point of one of two subsets of lattices $\Lambda_{w_n} = \Delta\mathbb{Z} + w_n\Delta/4$. In Fig. 2, these lattice points are marked by either × or ○.

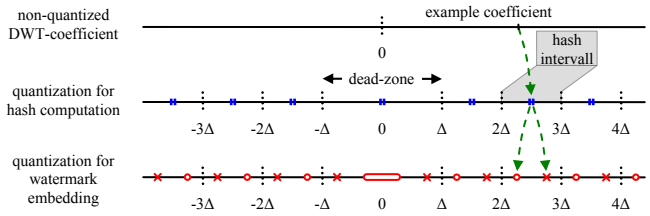

Fig. 2. Example: quantization for hash interval computation and watermark embedding using the modified dither modulation with dead-zone

3.1.3. Image size normalization

To tolerate scaling of host image size signature generation as well as embedding take place at a fixed size of 512x512 pixels. Since *tiling* is a basic part of JPEG2000 compression, we separate larger images into sub-regions of this size, and hence, get a standard conform JPEG2000 file stream.

In this way, JPEG2000 decoding with successive scaling, as shown in Fig. 3, doesn't affect image authenticity. The image can be verified as long as changes caused by image scaling don't effect hash-value computation.

On the other hand, if scaling is a consequence of partial JPEG2000 file stream decoding (*image size transcoding*) the integrity can be verified as long as the marked LL_4-subband is available.

Fig. 3. Progressive image decoding with lower resolution (left); image decoding and scaling afterwards (right)

3.1.4. Luminance and contrast normalization

Since we use LL-subband coefficients for signature generation as well as embedding, the host image has to be normalized prior watermarking to allow luminance and contrast adjustment operations.

For that reason, in a first step, the host signal is normalized to the mean pixel luminance (subtraction of gray-value pixel mean). In a second step, the image is normalized to contrast. As in Eq. (4), a factor g is computed from the pixel values of image $I := \{I_j \in \mathbb{N} : 0 \le I_j \le 255, 1 \le j \le J\}$. Prior to hashing and signature embedding, all host signal values are divided by factor g, where the same process takes place during signature verification.

$$g = \frac{1}{256}\left(\frac{1}{J}\sum_{j=1}^{J} I_j^{\,2}\right)^{1/2} \tag{4}$$

A contrast change, now, becomes a scaling of factor g, and hence, can be reversed similar to the normalization proposed by Pérez-González *et al*. in [7].

Further, we embed g as a second watermark in the HL_4-, LH_4-, and HH_4-coefficients using the same strategy as described in subsection 3.1.2. Thereby, g is represented by 32 bits and encoded using *repeat-accumulate coding* with a code rate of $r = 1/96$. The resulting 3072 bits are embedded using a small step-size, whereby there occur no further perceptual embedding distortions.

3.2. Watermark extraction and image reconstruction

During content integrity verification the embedded signature has to be compared (i.e., after extraction and decoding) with the hash-value generated from the received image.

The watermark is extracted by nearest neighbor quantization to one of the two quantizer subsets.

3.2.1. Hash interval error correction

As a consequence of data embedding at the same locations used for quantization-based hashing the overall robustness of hash-value computation is degraded.

As mentioned before, the hash-value remains constant as long as the quantized LL_4-coefficients don't leave the interval $[\Delta\mathbb{Z}; \Delta(\mathbb{Z}+1))$. But, as shown in Fig. 2, due to embedding the coefficients are moved to the lower or upper half of the quantization interval, respectively. Hence, even image processing operations changing the LL_4-coefficients more than $\Delta/4$ yield the verification to fail.

To solve this problem we extended the watermark bit error correction as follows. If $\hat{w} = [-1; +1]$ denotes the watermark extracted from the received host signal $\hat{y} \in \mathbb{R}$ and $\tilde{w} = \{-1, +1\}$ is the corrected watermark after FEC-decoding, then Eq. (5) can be applied to correct the hash intervals.

$$q_n = \operatorname{sign}(\hat{y}_n) \cdot \left\lfloor \frac{|\hat{y}_n|}{\Delta} + \frac{\hat{w}_n - \tilde{w}_n}{8} \right\rfloor \tag{5}$$

As demonstrated in Fig. 4, the hash interval is expanded to the range $[\Delta(\mathbb{Z}-1/4); \Delta(\mathbb{Z}+3/4))$ or $[\Delta(\mathbb{Z}+1/4); \Delta(\mathbb{Z}+5/4))$, respectively, depending on the watermark bit at the appropriate location. Hence, despite data embedding the coefficients can be changed up to $\Delta/2$ without affecting images authenticity. That way, the overall robustness is gained by a factor of two.

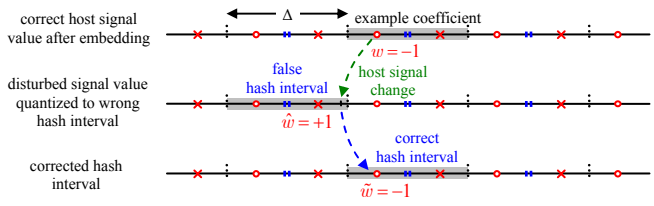

Fig. 4. Example: reconstruction of hash interval by combining hash-value quantization and watermark bit error correction

3.2.2. Watermark removal for image quality enhancement

A watermark bit is embedded in the lower or the upper half of the hash interval by moving the host signal coefficient to the points $\Delta(\mathbb{Z}+1/4)$ or $\Delta(\mathbb{Z}+3/4)$, respectively. Since the distribution of the coefficients of a DWT-transformed image in each hash interval can be approximated by a uniform distribution, this position is not optimal. To reduce this noise, which is higher than simple Δ-quantization noise, the verification algorithm can move the quantized LL_4-coefficients back to the centre of the hash intervals, $\Delta(\mathbb{Z}+1/2)$, after the watermark bits are extracted. This removes the embedded watermark and enhances image quality. Simulations have shown that the *peak-signal-to-noise-ratio* (PSNR) can be raised by approximately 2.5dB due to this removal.

3.3. Adaptation of step-size Δ based on image content

The choice of embedding strength (step-size Δ), and hence, the robustness of the hash as well as the signature are limited by the visual perception of embedding induced distortions. As shown in Fig. 5, if the same step-size is used for all LL_4-coefficients watermark embedding is not optimal.

The *human visual system* is less sensitive to changes in textured regions than in smooth regions of an image. That means, the choice of embedding strength is mainly limited by the visual perception of distortions in homogenous regions such as the cloud-free sky in the example image.

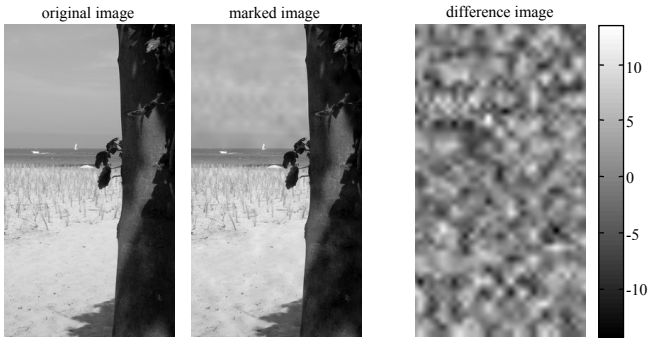

Fig. 5 Example: image distortions caused by signature generation and embedding using the same step-size Δ = 8 for all LL_4-subband coefficients

To improve the performance of our authentication system we use different step-sizes. We separate the image into homogenous regions and stronger textured regions. For signature generation and embedding within the LL_4-coefficients representing the former regions we use step-size Δ_1. For all the rest we use Δ_2.

As a consequence of using a larger step-size for textured regions the overall robustness against non-malicious image processing can be improved, as we show in section 4.

3.3.1. Texture-based image region separation

Based on the ideas in [8], we developed a new texture-based image region separation. We separate an image into less and stronger textured regions using the DWT-coefficients. As visualized in Fig. 6, except for the LL_3-subband, all coefficients of the third DWT-decomposition level are compared to a threshold τ. Afterwards, the three matrices are added and 2x2 block-wise averaged. Finally, the known morphologic operations *closing* and *erosion* are applied to refine the separation. The resulting matrix $F \in \mathbb{R}$ we call feature mask, in this paper.

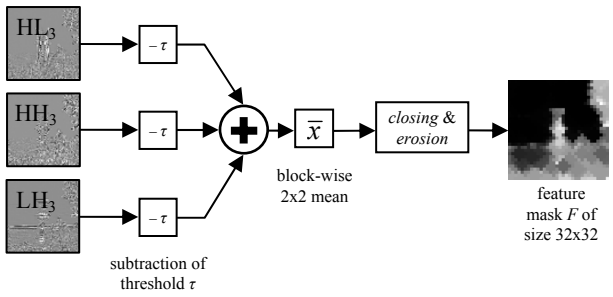

Fig. 6 Texture-based feature mask generation

Compared to the original image, homogenous regions yield negative values, whereas for stronger textured regions feature F is positive. Hence, during watermark embedding, we apply Δ_1 for all locations where $F < 0$, otherwise, we apply Δ_2, if $F \geq 0$.

3.3.2. Errors due to feature mask changes

Because our authentication system is oblivious, the feature mask has to be calculated in the same way from the received signal during verification. Slight changes due to image processing can yield errors. Even if no manipulation has been applied discrepancies can occur, for example, because the embedding process itself has influence on the parameter calculation and the separation feature mask.

As shown in Fig. 7, suppose a coefficient is quantized to the highlighted point o of lattice Λ′ (using Δ_1) during embedding. If afterwards the separation feature mask changes for this location, then Λ″ would be used during extraction, where Λ′ and Λ″ denote the two quantization lattices that each consist of sub-lattices Λ_{-1} and Λ_{+1} marked with × and o. Since Λ″ at this location covers Λ′ with a point o, as well, no error occurs. But if the point × right beside it has been used, it would be falsely decided to a point o in lattice Λ″. In this case, a bit substitution error would occur.

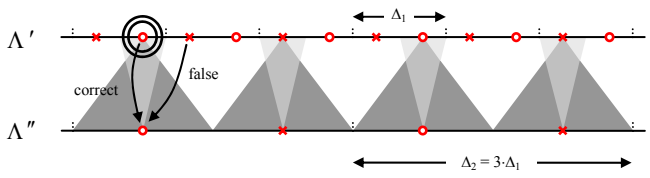

Fig. 7. Errors during watermark extraction due to choice of wrong quantization lattice Λ_2 instead of Λ_1

3.4. Soft image separation during FEC-decoding

The most often applied technique to circumvent discrepancies between the separation feature masks during embedding and extraction is to form a gap around the separation threshold. In other words, the used feature is pre-distorted to leave a margin. As a consequence, the image quality is degraded. Furthermore, there are separation approaches where it is computational infeasible to project the pre-distortion back onto the host image.

To solve this problem without applying pre-distortion, we propose to integrate commonly used hard region separation into an overall soft processing framework, as in Fig. 8.

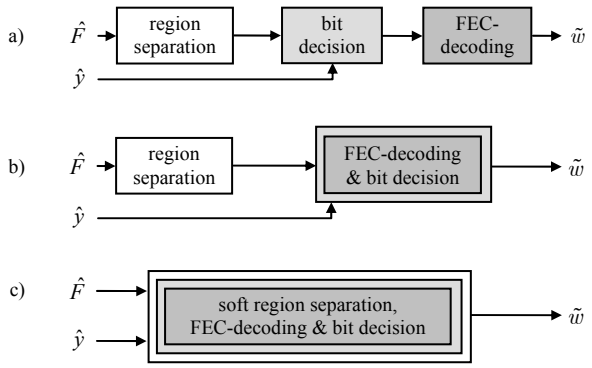

Fig. 8 Overall hard processing a), hard region separation with soft bit decoding b), overall soft processing c), where $\hat{F}$ = separation feature during extraction, $\hat{y}$ = received host signal, $\tilde{w}$ = corrected watermark

We use the separation feature $\hat{F}$ computed from the received image to weight the extracted watermark signal during FEC-decoding. For that, we exploit two properties as described in the following subsections.

3.4.1. Property I - certainty of separation decision

We use the certainty of how close the texture feature is to the selected feature threshold τ. If the feature is close to the decision threshold ($\hat{F}$ tending to zero), it is uncertain which quantization lattice has to be used during extraction. In this case, the certainty tends to zero. If the feature is far from the threshold and it is sure which lattice was chosen during embedding, then the certainty is high.

3.4.2. Property II - lattice point coverage

By simulations we found out that it is advantageous to apply odd ratio (i.e., Δ_2/Δ_1 = 3, 5, 7, …) for the step-sizes during watermark embedding. In this case, the points of lattices Λ' and Λ'' show more "covers" than for even ratio Δ_2/Δ_1.

Further, we learned that if we choose ratio Δ_2/Δ_1 to be 3, 7, 11, etc., then lattice Λ'' has to be inverted, resulting in lattice $\overline{\Lambda}''$. That means, all bits to be embedded at locations where $F \geq 0$ have to be inverted, first. Likewise, after extraction, all bits received using lattice Λ'' have to be inverted.

For attacks, such as lossy compression, noise adding or filtering, the distortion of the quantized signal can be expected to be Gaussian distributed. Since the variance of this distribution is the same for both lattices Λ' and Λ'', the following probability density functions $pdf(\bullet)$ can be expected. As can be seen in Fig. 9, there are spaces at lattice Λ'' where it is unlikely that a signal sample $\hat{w}_2$ is located.

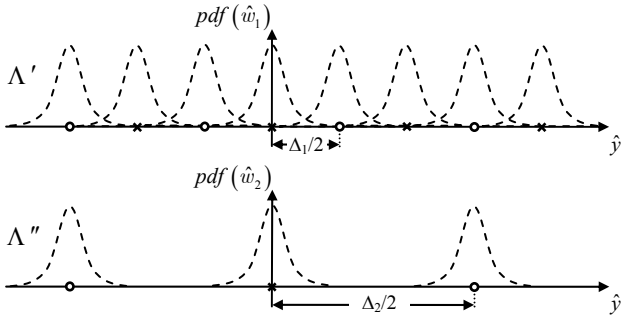

Fig. 9 Probability density function of the disturbed host signal $\hat{y}$, superimposed for all possible quantization lattice points (schematic representation)

If the feature is close to the decision threshold and the signal sample is somewhere in the space where $pdf(\hat{w}_2)$ is small, it is more likely that the sample was originally embedded using lattice Λ'.

3.4.3. Host signal weighting using both properties

During decoding, we separate the received host signal into two sub-signals $\hat{w}_1 = Q'(\hat{y})$ and $\hat{w}_2 = Q''(\hat{y})$, where $Q'(\bullet)$ denotes the quantizer used for lattice Λ' and $Q''(\bullet)$ denotes the quantizer for lattice Λ''. Afterwards, $\hat{w}_1$ and $\hat{w}_2$ are coupled to the certainty-of-separation-decision as well as lattice-point-coverage-property using the two weighting functions $f_1(\hat{F})$ and $f_2(\hat{F})$.

$$f_1(\hat{F}) = \begin{cases} 1 & , -\infty < \hat{F} < -\alpha \\ \frac{1}{2}\left(1+\cos\left(\frac{\hat{F}+\alpha}{\alpha}\cdot\frac{\pi}{2}\right)\right) & , -\alpha \leq \hat{F} < +\alpha \\ 0 & , +\alpha \leq \hat{F} < +\infty \end{cases} \quad (6)$$

$$f_2(\hat{F}) = \frac{\Delta_2}{\Delta_1}\cdot\begin{cases} 0 & , -\infty < \hat{F} < -\alpha \\ \frac{1}{2}\left(1-\cos\left(\frac{\hat{F}+\alpha}{\alpha}\cdot\frac{\pi}{2}\right)\right)\cdot\beta(\hat{F},\hat{w}_2) & , -\alpha \leq \hat{F} < +\alpha \\ 1 & , +\alpha \leq \hat{F} < +\infty \end{cases} \quad (7)$$

$$\beta(\hat{F},\hat{w}_2) = 1-\sin\left(\frac{\hat{F}+\alpha}{\alpha}\cdot\cos\left(\hat{w}_2\cdot\frac{\pi}{2}\right)\right) \quad (8)$$

By applying Eq. (9), finally, the two sub-signals are joint resulting in watermark signal $\tilde{w}$, which is the input to the soft-decision FEC-decoder (e.g., *Viterbi algorithm*).

(a) (b) (c)

Fig. 10. Example: (a) marked image using non-adaptive embedding, where $\Delta_1 = \Delta_2 = 6$, resulting in PSNR = 40.89 dB, (b) marked image using texture-based step-size adaptation, where $\Delta_1 = 3$ and $\Delta_2 = 9$, resulting in similar PSNR = 40.98 dB, and (c) contrast-enhanced difference of (b) to original image

$$\tilde{w} = \frac{\hat{w}_1 \cdot f_1\left(\hat{F}\right) + \hat{w}_2 \cdot f_2\left(\hat{F}\right)}{2} \qquad (9)$$

3.5. Perceptual image quality after signature embedding

The choice of embedding strength (step-size Δ), and hence, the robustness of the hash as well as the signature are limited by the visual perception of embedding induced distortions. To demonstrate these distortions, in Fig. 10, marked images are shown using non-adaptive as well as adaptive embedding. As can be seen by smaller visual distortions at the difference of the middle image to the original, for homogenous regions we use a smaller step-size than for stronger textured regions. Although the PSNR-values are similar for the left and middle image, distortions cannot be seen for the adaptively marked image in the middle.

By the use of subjective tests and simulations, we found out that choosing $\Delta_1 = 3$ and $\Delta_2/\Delta_1 = 3$ yield the best compromise between perceptual image quality and robustness.

4. EXPERIMENTAL RESULTS

4.1. Robustness against non-malicious manipulations

For all simulations we used a set of 52 different gray-scale images of size 512x512 pixels. For LL_4-coefficient hashing we applied the known *message-digest algorithm 5* (MD5) yielding a hash-value of length 128 bits. Further, we used RSA for signing the hash (512 bits key length). Finally, the signature was FEC-encoded using convolutional coding and a code-rate $r = 1/2$. Hence, 1024 bits were embedded within the LL_4-subband (32x32 coefficients) of every host image.

4.1.1. Comparison of performance

To compare the performance of our system with those of methods by other authors we use the results collected by Ekici *et al*. in their image authentication overview paper [1].

Results of robustness simulation are given for seven image authentication methods, where the watermark embedding induced image distortion was fixed to PSNR = 41 dB.

The authors tested the robustness (*false positive ratio*) against a set of following signal processing attacks:

- Smooth - low pass filtering using a 3x3 filter mask
- Sharpen - edge enhancement using 3x3 unsharp masking
- S and P - salt-pepper-noise (1%)
- Histogram equalization
- AWGN - Gaussian noise ($\sigma = 4{,}5 \rightarrow$ PSNR ~ 35 dB)
- JPEG 70 - lossy JPEG compression (QF = 70)
- Random file stream bit errors (0.1%)
- No attack - verification of marked, undisturbed image

Further, Ekici *et al*. simulated the *false negative ratio* (P_{miss}) by verifying the non-marked original images.

Table 1. False alarm and miss probabilities for comparison of performance of our approach with results of other authentication methods as given in [1], where embedding induced PSNR = 41dB

Semi-fragile 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	Random errors
Chang *et al.*	0,0 %	0,0 %	100 %	99,0 %	100 %	32,3 %	0,0 %	100 %	0,0 %
Delp *et al.*	0,1 %	2,3 %	54,5 %	3,4 %	6,5 %	2,7 %	2,4 %	0,3 %	14,1 %
Eggers *et al.*	0,0 %	0,0 %	41,4 %	91,0 %	2,6 %	0,0 %	0,0 %	65,6 %	2,5 %
Fridrich	1,0 %	1,6 %	62,0 %	5,5 %	19,5 %	2,5 %	25,8 %	21,0 %	2,5 %
Kundur *et al.*	0,1 %	0,0 %	77,7 %	99,5 %	51,9 %	10,0 %	2,9 %	98,1 %	0,1 %
Queluz	0,01 %	0,01 %	27,8 %	94,3 %	42,7 %	0,01 %	0,01 %	100 %	1,1 %
Liao *et al.*	8,7 %	3,0 %	34,3 %	80,7 %	43,3 %	1,7 %	1,5 %	79,9 %	4,2 %
Schlauweg *et al.*	0,0 %	0,0 %	0,0 %	100 %	100 %	0,0 %	0,0 %	43,7 %	0,0 %

The results in Table 1 show that our authentication system (Schlauweg *et al*.) performs better in most cases. We reach better false positive ratios except for histogram equalization and salt-pepper-noise. However, since 1%-salt-pepper-noise yields visual image degradations, we rate this operation to the group of malicious manipulations, anyhow. Histogram

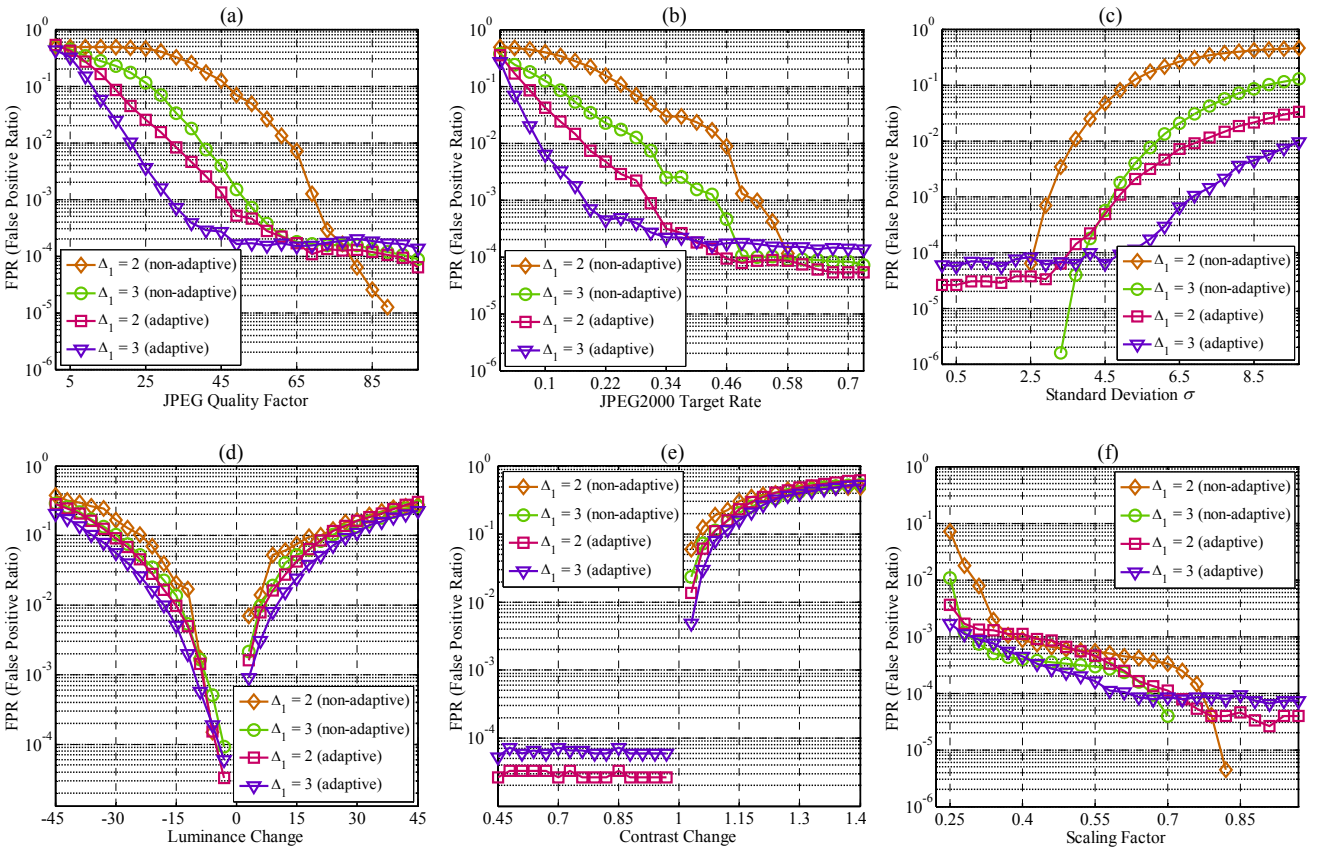

Fig. 11. Results of simulations for the overall enhanced, adaptive image authentication system - robustness against: (a) JPEG compression, (b) JPEG2000 compression, (c) additive Gaussian noise, (d) luminance change, (e) contrast change, and (f) scaling of image size. Parameters: for non-adaptive embedding $\Delta_2/\Delta_1 = 1$, for adaptive embedding $\Delta_2/\Delta_1 = 3$, $\tau = 1.5$, $\alpha = 5$.

equalization, on the other hand, is a known image enhancement operation. But, although our system is able to handle luminance or chrominance changes, it fails for the complex non-linear equalization of image histogram.

We think that for applicability of an authentication system it is important that the system is secure. Hence, we highlight that P_{miss} (forgery attack) is zero for our system.

4.1.2. Comparison of non-adaptive and adaptive embedding

To demonstrate how our new authentication system benefits from texture-based adaptive hashing/embedding, in Fig. 11, we show results of further robustness tests. It can be seen that for similar visual perception of embedding induced distortions the robustness against non-malicious image processing could be improved.

4.2. Security of the proposed authentication system

In addition to high robustness against non-malicious image processing, it is important that alarm is raised during verification if fraudulent attacks are applied to the marked image.

For example, to simulate a copy/paste-attack for a large number of marked images, we randomly exchanged two pixel blocks within each image. Afterwards, these images were verified. The aim is to reach a false negative ratio tending to one. The results of this test are shown in Fig. 12.

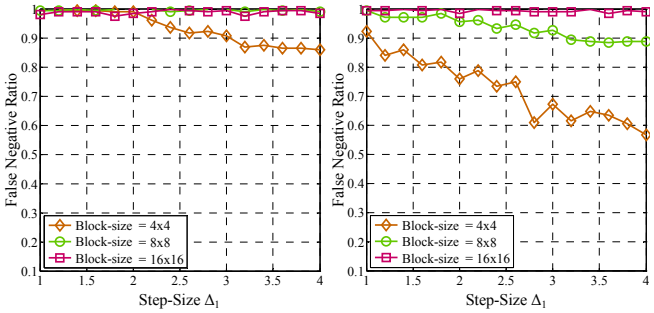

Fig. 12 Results of simulations of pixel block-exchanging attack, where the difference between both exchanged pixel blocks is PSNR < 15 dB (left) and PSNR < 25 dB (right)

Since some pixel blocks exchanged against each other can be similar resulting in the same hash-value during verification, the determined false negative ratio is not always one. So, we separated the blocks exchanged during this test into two groups. All pairs of blocks yielding a strong differ-

(a) (b) (c)

Fig. 13. Example for an always detectable manipulation using the proposed authentication system: (a) watermarked image with $\Delta_1 = 3$ and $\Delta_2/\Delta_1 = 3$; (b) manipulated image - object of size 6x9 pixel removed; (c) contrast-enhanced difference image

ence (PSNR < 15 dB) were assigned to the first group. Pairs yielding a higher PSNR were assigned to the second group.

As can be seen, the usage of $\Delta_1 = 3$ and $\Delta_2/\Delta_1 = 3$ yields a secure authentication system in terms of copying and pasting or deleting image objects of big and small size. For example, the filigree 6x9 pixel-sized attack shown in Fig. 13 is always securely detectable using our image authentication. Hence, our system is not only robust against a range of allowed image processing operations but secure against manipulating attacks at the same time.

5. CONCLUSION

In this paper, we described the embedding of a digital watermark for image authentication. During JPEG2000 compression, a semi-fragile signature was generated from image content and embedded by quantization of the coefficients in the DWT-domain. Generation as well as embedding of the signature is adapted to the image content for performance improvement. For that, we presented a soft-decoding texture-based image region separation and used different step-sizes for less and stronger textured image regions to exploit the texture masking properties of the human visual system. Our image authentication is tested extensively and performance results are compared to those of methods proposed by other authors. We showed that our new system outperforms these methods. Our semi-fragile authentication is robust against non-malicious modifications, such as lossy compression, noise, image blurring and sharpening, changes of luminance or contrast as well as scaling. But in contrast to other methods, our authentication is secure at the same time, which we proofed by different forgery attacks.

6. REFERENCES

[1] Ö. Ekici, B. Sankur, B, Coşkun, U. Naci, and M. Akcay, "Comparative evaluation of semifragile watermarking algorithms," *Journal of Electronic Imaging*, vol.13 (1), pp. 209-216, Jan. 2004.

[2] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When seeing isn't believing," *IEEE Transaction on Signal Processing*, vol.21, pp. 40-49, March 2004.

[3] C. Rey and J.-L- Dugelay, "A survey of watermarking algorithms for image authentication," *EURASIP Journal of Applied Signal Processing*, vol.6, pp. 613-621, March 2002.

[4] Q. Sun and S. F. Chang, "A secure and robust digital signature scheme for JPEG2000 image authentication," *IEEE Transactions on Multimedia*, vol.7 (3), pp. 480-494, June 2005.

[5] Z. Zhang, G. Qui, Q. Sun, X. Lin, Z. Ni, and Y. Q. Shi, "A unified authentication framework for JPEG2000," *In Proc. of IEEE International Conference on Multimedia and Expo*, vol.2, pp. 915-918, June 2004.

[6] B. Chen and G. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol.47 (4), pp. 1423-1443, May 2001.

[7] F. Pérez-González, C. Mosquera, M. Barni, and A. Abrardo, "Rational Dither Modulation: A novel data hiding method robust to value-metric scaling attacks," *In Proc. of IEEE Workshop on Multimedia Signal Processing*, pp. 139-142, Sept. 2004.

[8] D. Huang, L. Jiufen, H. Jiwu, and L. Hongmei, "A DWT-based image watermarking algorithm," *In Proc. of IEEE International Conference on Multimedia and EXPO*, pp. 313-316, Aug. 2001.

[9] M. Schlauweg, D. Pröfrock, and E. Müller, "JPEG2000-based secure image authentication," *In Proc. of ACM Multimedia and Security Workshop*, Geneva, Switzerland, pp. 62-67, Sept. 2006.

[10] M. Schlauweg, D. Pröfrock, and E. Müller, "Avoiding hard decisions in adaptive watermarking," *In Proc. IEEE International Conference on Image Processing*, Dallas, USA, pp. 453-456, Sept. 2007.