Image Authentication Using Robust Image Hashing with Localization and Self-Recovery

Ammar M. Hassan,
Ayoub Al-Hamadi, Bernd Michaelis
IESK
Otto-von-Guericke-Unviersity
Magdeburg, Germany
{Ammar, Al-Hamadi}@ovgu.de

Yassin M. Y. Hasan
Computer Sc. and Info. Dept.
Taibah University
Madinah, KSA
ymyhasan@aun.edu.eg

Mohamed A. A. Wahab
Electrical Engineering Dept.
Minia University
Minia, Egypt

ABSTRACT

The rapid growth of efficient tools, which generate and edit digital images demands effective methods for assuring integrity of images. A semi-fragile block-based image authentication technique is proposed which can not only localize the alteration detections but also recover the missing contents. The proposed technique distinguishes content-preserving manipulations from the content alterations using secure image hashing instead of cryptographic hashing. The original image is divided into large blocks (sub-images) which are also divided into 8×8 blocks. Secure image hashing is utilized to generate the sub-image hash (signature) which may slightly change when the content-preserving manipulations are applied. Furthermore, the sub-image code is generated using the JPEG compression scheme. Then, two sub-image hash copies and the sub-image code are embedded into relatively-distant sub-images using a doubly linked chain which prevents the vector quantization attack. The hash and code bits are robustly embedded in chosen discrete cosine transform (DCT) coefficients exploiting a property of DCT coefficients which is invariant before and after JPEG compression. The experimental results show that the proposed technique can successfully both localize and compensate the content alterations. Furthermore, it can effectively thwart many attacks such as vector quantization attacks.

Keywords

Cryptographic hashing, image authentication, image hashing, watermarking.

1. INTRODUCTION

The current advances in information technology, the widespread multimedia applications and wireless services require efficient methods for guaranteeing privacy, security, protection and integrity of the assorted multimedia data categories. Since many recently developed devices and efficient software products offer consumers worldwide capabilities of flexibly creating, manipulating, and exchanging multimedia data, considerable efforts and contributions have been lately made on digital watermarking that inserts a piece of information (the watermark) into multimedia (host/cover) data for many purposes such as [Has04, Has07, Won01]: image authentication, copyright protection, fingerprinting, broadcast monitoring and data hiding.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

For example, in medical archiving and e-commerce, we strongly desire to be sure that the images are genuine and in the news reporting, it is important that the image truthfully reflects the real view at the time of capture [Lan99, Won01]. For image authentication purposes, it is required that the watermarking algorithm is blind, secure and so sensitive that slight modifications to the image content are detected and precisely localized [Lin99,Yeu97]. Fragile [Won98, Bar02, Cel02], semi-fragile [Eki04, Lin00, Lin01a, Lin07, Mae06], self-recovery/embedding [Fri99b, Lin01b, Lue08, Wan08] watermarking schemes have recently been presented for image authentication.

Fragile image authentication schemes are so sensitive to pixel changes where their watermarks are easily damaged even in case of harmless changes in the image data due to content-preserving manipulations that do not affect the content [Lin99]. Hence, fragile image authentication is applicable and of interest only in case of lossless environment, i.e., coding, storage, transmission (of the watermarked image). The fundamental objective of the attacker facing such fragile watermark is to keep a watermark that makes

his/her altered or completely forged image, "pass" the verification test as authentic [Has04, Has07, Lue08]. Block-based fragile/semi-fragile image authentication schemes provide attack localization but they are vulnerable to vector quantization (VQ) attacks [Hol00], relying on that the watermark embedding/verification processes are run on independent blocks. Once an attacker has a table of authenticated blocks (with the same security parameters), he/she can use the best-authenticated approximation of an un-authenticated block without having the verification process detecting his/her alterations. This type of attack principally differs from the attacks against copyright protection and information hiding where the attacker may mainly want to significantly distort or remove the watermark with imperceptible alterations in the image [Kut00, Kir02].

Various global and block-based (sized down to pixel-wise) fragile image authentication methods have been developed. A simple fragile scheme simply replaces the least significant bits (LSBs) of the image of interest with the checksum (i.e., modulo-2 addition) bits of a long word of some most significant bits (MSBs) [Lin99]. In [Yeu97], the use of a user-defined color look-up tables (LUTs) guided pixel-wise adjustment to embed the watermark is proposed. Wong's block-based method [Won97] and its public-key modified versions [Won98, Won01] replace the LSBs of each block with a signature of its MSBs, with the image size, image index and/or block index, xor-ed with its corresponding watermark block.

On the other hand, semi-fragile image authentication techniques embed watermarks so robustly to survive (to some, application dependent, extend) various kinds of typical image processing manipulations such as lossy compression as long as the image contents are preserved. At the same time, embedded watermarks must detect malicious alterations such as deleting or adding an object. In many semi-fragile schemes, the relations between pairs of discrete cosine transform (DCT) coefficients in a block are used as the block signature. Then, the signatures (watermarks) are robustly embedded in low frequency coefficients [Lin00, Lin01a]. In [Mae06], the authors introduce two methods to generate the signatures using the discrete wavelet transform (DWT). In the first method, random values are added to the difference between two coefficients before the difference is encoded to generate the signature bit. The second method proposes the use of a multiple nonuniform quantizer to encode the coefficient difference in each pair. Lin et. al. use the differences of DCT coefficients as signatures and modify other DCT coefficients to match the signatures [Lin07].

Furthermore, to not only localize altered regions but also compensate for the damage, self-recovery/ embedding image authentication techniques have been presented that embed an image approximation into the image itself in a fragile [Lue08, Wan08] or semi-fragile [Has07, Fri99a] way using various techniques.

An original self-recovery/embedding image authentication technique based on JPEG compression has been introduced in [Fri99b]. A JPEG compressed version of each block B is inserted into the LSBs of the block $B+\vec{P}$, where $\vec{P}$ is a vector of length approximately 1/3 of the image size, with a randomly chosen direction. The algorithm limitations and possible attacks are addressed in [Fri99c, Lue08]. In [Lin01b], Lin and Chang have proposed an algorithm using quantized coefficients of the DCT of the image blocks as a watermark and modifying the coefficients differences to match the quantized coefficients (watermark). The attacker can easy defeat the verification process applying the same algorithm into a fake image. Instead of using a JPEG compression version as an image approximation, Wang and Tsai have used fractal codes of a ROI (region of interest), which is chosen as the important object in the host image [Wan08]. On the other hand, Lue et al. proposed a technique that uses a halftone version of the host image as an approximation image [Lue08].

In this paper, image hashing technology, which will be described in the next sections in details is utilized to generate the sub-image signature. A code of the approximated sub-image is computed using the principals of JPEG compression. Then, the sub-image signature copies and the sub-image code are robustly embedded into DCT coefficients of two relatively-distant sub-images making a doubly linked chain.

The remainder of this paper is organized as follows: cryptographic hashing, which is mostly used to generate image/block signature in fragile algorithms, and image hashing, which we adopt to generate the proposed signatures, are described in Section 2. In Section 3, existing image hashing schemes are presented. The proposed technique is introduced in Section 4. Experimental results are shown in Section 5. In Section 6, the conclusion is presented.

2. CRYPTOGRAPHIC HASHING AND IMAGE HASHING

The cryptographic hash functions such as MD4, MD5, and SHA [Men01, Sch96] map the input data to a short fixed length string. For the hash function H_c and the input data d, it should be easy to compute the hash $h_c=H_c(d)$. For this type of functions, called one-way-functions, it is too hard to estimate the input data d from the hash h_c. Hash functions have, at least, the following additional properties [Men01, Sch96]:

- Given the hash h_c, it is computationally infeasible to find an input which hashes to that output, i.e. it is hard to find d such that $H_c(d)=h_c$.
- Given the data d, it is hard to find another input data d_0 which hashes to the same output, i.e. it is hard to find $H_c(d)=H_c(d_0)$.
- It is computationally infeasible to find any two inputs d_0 and d_1 which have the same output (i.e., satisfying collision resistance).

It is clear that the cryptographic hash is so sensitive to changes in the input data where small changes, even a single bit, dramatically change (~50%) the output. To secure the hash, it may be encrypted by an encryption algorithm. The cryptographic hash is mostly used for digital signatures and fragile image authentication.

On the other hand, the image (visual) hash function H maps the input image (or sub-image) to an output $h=H(I)$ that is invariant under perceptually insignificant image changes with the following main properties[Fri00, Mih01, Swa06, Ven00, Tan08]:

- It is hard to find two different images having the same or very close hash value(s) (collision resistance).
- Given h, perceptual changes to an image I lead to a different hash $H(I') \neq H(I)$.
- The hash is key dependent, for security reasons, so that different keys give significantly different hash values.

The main difference between image (visual) and cryptographic hashing is that image hashing accepts perceptually insignificant changes in the input image with small hash changes; but small changes in the input data lead to very significant changes in the cryptographic hash.

3. IMAGE HASHING SCHEMES

In [Fri00], The image hash is generated by projecting the input image onto patterns which are generated using a zero-mean uniform distributed key random generator. The resulting hash is resilient to many normal operations but it is not collision free [Swa06]. Venkatesan et al. have introduced an image hashing algorithm that uses the discrete wavelet transform (DWT) of an image. Statistics of each subband block are calculated, randomly quantized and encoded to generate the final hash value [Ven00]. Unfortunately, the algorithm does not work well for object insertion. In [Mih01], the DWT is employed to capture the image hash based on threshoding and iterative filtering. Swaminathan et al. [Swa06] have exploited the Fourier-Mellin transform to generate image features. In the polar coordinate, the summation of image values alone angle axis at equal distant points for a specific radius is an image feature. The image features for radii are represented as the image hash.

In [Tan08], a robust image hash algorithm uses a non-negative matrix factorization (NMF) scheme for generating the image hash. First, the image undergoes preprocessing as a sequence of image resizing, color space conversion and low-pass filtering. The preprocessed image is then divided into unequal blocks. Next, each block is rescaled to a fixed size and put as a vector in a matrix that is undergone NMF. The elements of the NMF coefficient matrix are quantized and encoded to generate the image hash. We use this algorithm to generate the sub-image signature in our proposed image authentication technique. So, we describe it in more details in the rest of this section. The scheme is composed of the following four main steps:

First step: Image preprocessing

a- The image is resized to $q \times q$ using bi-linear interpolation.
b- The color space of $q \times q$ image is converted to YC_bC_r.
c- The Y plane is passed through a low-pass filter.

Second step: Building the secondary image

a- The preprocessed image U is randomly divided into t strips, and each strip is again divided into t blocks with varied sizes, resulting in $t^2=N_b$ blocks in total.
b- Each block is resized to $k \times k$ using bi-linear interpolation.
c- Each $k \times k$ block is stacked to construct a $k^2 \times 1$ vector v.
d- Each vector v is used as a column in a pseudo-random order to form the $m \times n$ matrix V, where $m=k^2$. V is called the secondary image.

Third step: Data reduction

a- V undergoes NMV giving the coefficient matrix C (see the appendix).
b- C entries are quantized to generate a binary matrix C^b as follow:

$$c^b_{l,j} = \begin{cases} 0, & c_{l,j} \leq c_{l,j+1} \\ 1, & c_{l,j} > c_{l,j+1} \end{cases} \quad (1)$$

where $c_{l,j}$ denotes the entry of C in the l^{th} row and the j^{th} column , and $c_{l,n+1} = c_{l,1}$.

Final step: Hash security

a- C^b entries are concatenated to form a binary string.
b- The binary string is interleaved using a key to produce a key-dependent image hash h.

4. PROPOSED TECHNIQUE

Image hashing is employed to generate the sub-images' hashes (signatures) which are used to check the authenticity of an image. Two signature copies of each sub-image are robustly embedded into two

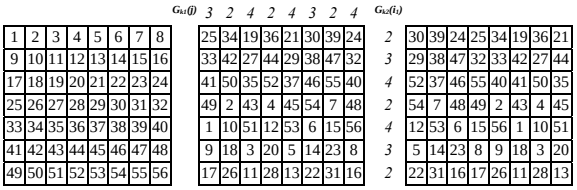

Figure 1. Example of the proposed scheme for choosing relatively-distant sub-images. (a) Original sub-images. (b) Sub-images after $G_{k1}(j)$ column-wise circular shifts. (c) Sub-images after $G_{k2}(i_1)$ row-wise circular shifts.

relatively distant sub-images which are pseudo-randomly chosen using a doubly linked chain in low frequency DCT coefficients. In the proposed technique, the image of interest is divided into sub-images. The sub-image hash (signature) is computed using the secure image hashing algorithm [Tan08] and the sub-image code, which represents the approximated sub-image is generated using the JPEG compression principles. Then, the sub-image hash copies and the sub-image code are robustly inserted into relatively distant sub-images. In the next subsections, the embedding and the verification processes are described in details.

Embedding Process

The original $M \times N$ image I is divided into $m' \times n'$ sub-images as follows:

$$I = \left\{ SI_{1,1}, SI_{1,2}, ..., SI_{2,1}, SI_{2,2}, ..., SI_{\lceil M/m' \rceil, \lceil N/n' \rceil} \right\} \quad (2)$$

where m' mod 8=0 and n' mod 8=0, $\lceil x \rceil$ is the floor of x. For each sub-image $SI_{i,j}$, the sub-image hash $h_{i,j}$ is computed using the secure image hashing algorithm [Tan08] such that:

$$h_{i,j} = H(SI_{i,j}) \quad (3)$$

To compute the sub-image code $C^S_{i,j}$, the sub-image $SI_{i,j}$ is resized to 8×8. Then, the resized sub-image is undergone the DCT. The DCT coefficients are quantized using the quantization table which corresponds to 50% quality JPEG compression. Then, the quantized coefficients are encoded using a fixed bit allocation table to generate the sub-image code.

Each sub-image is divided into 8×8 blocks as follow:

$$SI_{i,j} = \left\{ b^{i,j}_{1,1}, b^{i,j}_{1,2}, ..., b^{i,j}_{m'/8, n'/8} \right\} \quad (4)$$

Two hash copies and the code of each sub-image are robustly inserted and spread into two relatively distant sub-images generating a doubly linked chain.

In the color images, we use the Y channel of the YC_bC_r color space for embedding the hash copies and codes. The choice of the two relatively distant sub-images depends on the sub-image index and it is controlled by secret keys as follows:

$$\begin{aligned} i_1 &= (i + G_{k1}(j)) \bmod M_s + 1, G_{k1}(j) \in [[M_s/3], [2M_s/3]] \\ j_1 &= (j + G_{k2}(i_1)) \bmod N_s + 1, G_{k2}(i_1) \in [[N_s/3], [2N_s/3]] \\ i_2 &= (i + G_{k3}(j)) \bmod M_s + 1, G_{k3}(j) \in [[M_s/3], [2M_s/3]] \\ j_2 &= (j + G_{k4}(i_2)) \bmod N_s + 1, G_{k4}(i_2) \in [[N_s/3], [2N_s/3]] \end{aligned} \quad (5)$$

where G_{k1}, G_{k2}, G_{k3} and G_{k4} are key seed random generators with keys *k1, k2, k3* and *k4*. *Ms* and *Ns* are the number of sub-images per column and row, respectively.

Fig. 1 illustrates an example of the indices of the first relatively distant sub-image for each sub-image after $G_{k1}(j)$ column-wise circular shifts followed by $G_{k2}(i_1)$ row-wise circular shifts.

Then, each block of distant sub-images is transformed to the frequency domain using the DCT. We robustly embed two hash copies of the sub-image and the sub-image code (sub-image approximation) into the two relatively distant sub-images. One copy is embedded into the first distant sub-image blocks and the other copy into the second distant sub-image blocks. Furthermore, we divide the sub-image code into two groups which are embedded into the two distance sub-images. For embedding a bit of a sub-image hash copy or a bit of a sub-image code, we use a proved theorem given in [Lin00]. The theorem explains that if a DCT coefficient is quantized by $Q_{qf}(v)$ (*qf* refers to the compression quality factor), this coefficient can be reconstructed after JPEG compression with *qf1>qf*. Depending on this theorem, we can embed a bit into a DCT coefficient using an arbitrary quantization step and we can also recover this bit even if JPEG compression is applied with a quality factor greater than the quality factor, which is used in the embedding operation. Therefore, if we arrange the DCT coefficients of a block in zigzag order , the chosen coefficient of the block $b^{i,j}_{u,v}$, which has an index (u,v) in the sub-image $SI_{i,j}$, is modified as follows :

$$mb^{i,j}_{u,v}(l) = \begin{cases} \left[\dfrac{b^{i,j}_{u,v}(l)}{Q_m(l)} \right] Q_m(l) , \left[\dfrac{b^{i,j}_{u,v}(l)}{Q_m(l)} \right] \bmod 2 = h_{i,j}(t) \\ \left[\dfrac{b^{i,j}_{u,v}(l)}{Q_m(l)} + sign\left(\dfrac{b^{i,j}_{u,v}(l)}{Q_m(l)} - \left[\dfrac{b^{i,j}_{u,v}(l)}{Q_m(l)} \right] \right) \right] Q_m(l) , else \end{cases} \quad (6)$$

where $mb^{i,j}_{u,v}$ is the modified block, Q_m is the specific quantization table, $[x]$ is the round of x, l is the chosen middle frequency coefficient index, t is the hash index, sign(x) is equal 1 if x is a positive value and it is -1 if x is a negative value. Using (6), we can embed the bits of the hash and also the bits of the code into low frequency coefficients, which have pre-specific indices. A sub-image hash copy and the first group of the code are embedded into chosen coefficients of the

first distant sub image blocks. This operation is repeated for embedding another copy of the sub-image hash and the second group of the code into other chosen coefficients of the second distant sub-image blocks. After embedding the hashes and codes of all sub-images, the DCT coefficients are converted back to pixel integer domain. There is a possibility for losing some embedded bits by the rounding and truncation which are used for converting to the pixel domain. Therefore, we use an iteration procedure to assure the embedded bits are exactly extracted from the authenticated image.

Verification Process

In the verification process, the alterations that may occur on an authenticated image are not only detected and localized but also repaired. In the verification process, the test image I' is divided into sub-images and each sub-image hash $h'_{i,j}$ is computed. For each sub-image $SI'_{i,j}$, the corresponding distant sub-images indices are computed using (5). The embedded hash copy $he'_{i,j}$ and the first group of the sub-image code are extracted from the first distant sub-image $SI'_{i1,j1}$. A bit is extracted as follows:

$$he^1_{i,j}(t) = \left\lfloor \frac{b^{'i1,j1}_{u,v}(l)}{Q_m(l)} \right\rfloor \bmod 2 \qquad (7)$$

where $b^{'i1,j1}_{u,v}$ is the block, which has an index (u,v), in the sub-image $SI'_{i1,j1}$, and $Q_m(l)$ is the quantization step. The other hash copy $he^2_{i,j}$ and the second group of the code are extracted by the same method from the second distant sub-image $SI'_{i2,j2}$. The two groups of the code are combined together to be the extracted code $C^e_{i,j}$ of the sub-image $SI'_{i,j}$. To evaluate the match of hashes, the normalized Hamming distance is used which is defined as:

$$d(h1,h2) = \frac{1}{L}\sum_{t=1}^{L} |h1(t) - h2(t)| \qquad (8)$$

where L is the length of the hash string. For each sub-image, we compute the normalized Hamming distance between the computed and extracted hashes. The status of the sub-image $ST_{i,j}$ (altered sub-image or not) is evaluated as follows:

$$ST_{i,j} = \begin{cases} 0, d(h'_{i,j}, he^1_{i,j}) > T \quad and \quad d(h'_{i,j}, he^2_{i,j}) > T \\ 1, else \end{cases} \qquad (9)$$

where T is a threshold, $ST_{i,j}$=0 if the sub-image $SI'_{i,j}$ is considered as an altered sub-image, otherwise $ST_{i,j}$=1. For each altered sub-image, the approximated original sub-image can be recovered if the two distance sub-images of the concerned sub-image are not altered. Therefore, the reconstructed sub-image $SI^c_{i,j}$ is rebuilt as follows:

$$SI^c_{i,j} = \begin{cases} dec(C^e_{i,j}),\ ST_{i,j} = 0 \, and \ ST_{i1,j1} = ST_{i2,j2} = 1 \\ LOST, ST_{i,j} = 0 \, and \, (ST_{i1,j1} = 0 \, or \, ST_{i2,j2} = 0) \\ SI'_{i,j}, ST_{i,j} = 1 \end{cases} \qquad (10)$$

where *LOST* is a sub-image that is marked as a lost sub-image, *dec* is a sub-image decoding method.

5. EXPERIMENTAL RESULTS

To examine the robustness of the proposed technique, we consider the performance of it to JPEG compression and additive noise. The proposed system has been tested using 50 512×512 images. We firstly study the effects of JPEG compression with a range of quality factors. Then, the additive Gaussian noise effects are addressed. The size of the used sub-image is 32×32. To calculate the sub-image hash, the parameters are r=8, t=2 and k= 16. Thus, the hash length is 32 bits. In the robustness tests of the proposed technique, the quantization table of 50% quality JPEG compression is used as a predefined quantization table Q_m. The chosen coefficients' indices of the first distant sub-image blocks are {(1,4),(4,1)} for embedding the hash copy and {(2,3),(3,2)} for embedding the first group of the code. For the second distant sub-image blocks, the chosen coefficients' indices are {(2,4),(4,2)} for the second hash copy and {(1,3),(3,1)} for the second group of the code.

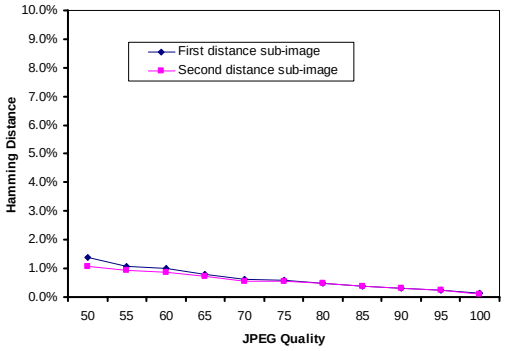

Figure 2. Average Hamming distance between the computed and extracted hashes for various JPEG compression quality factors.

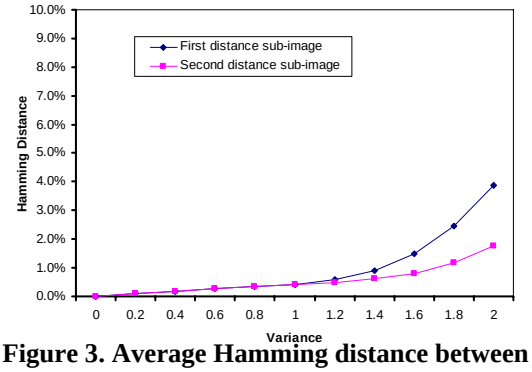

Figure 3. Average Hamming distance between the computed and extracted hashes for various Gaussian noise variances.

Fig. 2 illustrates that the average Hamming distance between the computed sub-image hash and extracted hashes recovered from the first distant sub-image and the second distant sub-image, respectively for various JPEG compression quality factors.

The effects of the additive zero-mean Gaussian noise have also been tested. Fig. 3 shows the average Hamming distance between the computed sub-image hash and extracted hashes which are extracted from the first distant sub-image and the second distant sub-image, respectively for various noise variances. From these figures, we observe that the normalized Hamming distance values are less than 9%. Thus, we can use this value as a threshold *T*.

To validate the proposed technique, we test it to check its capability of detecting local malicious manipulations mixed with JPEG compression. Fig. 4 is the original image and the approximated image, which represents the codes of all sub-images is shown in Fig. 5. The correlation coefficient between the original (grayscale version) and approximated images is 0.9198 and the peak signal to noise ratio PSNR of the approximated image relative to original image is 26.07 dB. The original image is authenticated using the proposed technique with the used quantization table Q_m of 70% quality JPEG compression to yield the image of Fig. 6. The correlation coefficient between the original and authenticated images is 0.9982 and the PSNR of the authenticated image relative to the original image is 42.47 dB. The authenticated image is altered by a local malicious attack. Then, it is undergone 80% quality JPEG compression to yield the image of Fig. 7. In Fig. 8, the proposed technique efficiently detects and localizes the content alterations. The proposed technique can not only localize the alteration detection but also successfully recover the missing contents as shown in Fig. 9.

6. CONCLUSION

A self-recovery semi-fragile image authentication technique is proposed which uses secure image hashing with improved localization. Using image hashing in the proposed technique to generate the signatures gives the proposed technique the capability to be robust against the normal operations such as JPEG compression and additive noise. To thwart the vector quantization attack, two sub-image hash copies and the sub-image code are securely embedded into two relatively distant sub-images. The experiment results explain

Figure 5. Approximated image, correlation coefficient=0.9198, PSNR=26.07dB.

Figure 6. Authenticated image, correlation coefficient=0.9982, PSNR=42.47dB.

Figure 4. Original image.

Figure 7. Altered version of the authenticated image.

that the proposed technique successfully distinguishes the normal manipulations such as JPEG compression from malicious operations and precisely localizes the alteration detections. Moreover, the proposed technique can successfully compensate the missing contents.

Figure 8. Verification result marking the altered regions.

Figure 9. Reconstructed image.

7. ACKNOWLEDGMENTS

This work is supported by Forschungspraemie (BMBF-Förderung, FKZ: 03FPB00213) and Transregional Collaborative Research Centre SFB/TRR 62 ”Companion-Technologyfor Cognitive Technical Systems” funded by the German Research Foundation (DFG).

8. APPENDIX

Non-negative matrix factorization NMF

In NMF, a non-negative matrix V is factorized into two matrices, B and C:

$$V \approx B\,C \tag{11}$$

where B and C are called the base matrix and the coefficient matrix respectively. The factors C and B must be non-negative.

If the size of V is $m \times n$, the sizes of B and C are $m \times r$ and $r \times n$, respectively. If r is chosen as less than m and n, NMF may be used for dimensionality reduction.

To compute B and C, the following updating rules are applied [Tan08]:

$$B_{i,l} \leftarrow B_{i,l} \frac{\sum_{j=1}^{n} C_{l,j} V_{i,j} / (BC)_{i,j}}{\sum_{j=1}^{n} C_{l,j}} \tag{12}$$

$$C_{l,j} \leftarrow B_{l,j} \frac{\sum_{i=1}^{m} B_{j,l} V_{i,j} / (BC)_{i,j}}{\sum_{i=1}^{m} B_{i,l}} \tag{13}$$

where i =1,2,...,m ; j=1,2,...,n; l=1,2,...,r.

9. REFERENCES

[Bar02] Barreto, P. S. L. Kim, M. H. Y. and Rijmen, V. : Toward a secure public-key blockwise fragile authentication watermarking, IEE Proc. Vision, Image & signal Proc., vol.149, no.2, pp.57-62, 2002.

[Cel02] Celik, M. U. Sharma, G. Saber, E. and Tekalp, A. M. : Hierarchical watermarking for secure image authentication with localization, IEEE Trans. on Image Proc., vol.11, no.6, June 2002.

[Eki04] Ekici, O. Sankur, B. Coskun,B. Naci, U. and Akcay, M. : Comparative evaluation of semi-fragile watermarking algorithms, Journal of Electronic Imaging, vol.13, no.1, pp.209-216, Jan. 2004.

[Fri99a] Fridrich, J. : Methods for tamper detection in digital images, in Proc. ACM Workshop on Multimedia and Security, Orlando, 1999, pp.19-23.

[Fri99b] Fridrich, J. and Goljan, M. : Images with self-correction capabilities, in Proc. ICIP'99, Kobe, Japan, 1999.

[Fri99c] Fridrich, J. and Goljan, M. : Protection of digital images using self embedding, Symp. Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, May 14, 1999.

[Fri00] Fridrich, J. and Goljan M. : Robust hash functions for digital watermarking, in Proc. IEEE Int. Conf. Information Technology: Coding Computing, Mar. 2000, pp. 178–183.

[Has04] Hasan, Y. M. Y. and Hassan, A. M. : Fragile blockwise image authentication thwarting

vector quantization attack, in Proc. IEEE ISSPIT'04, Rome, Italy, 2004.

[Has07] Hasan, Y. M. and Hassan, A. M. : Tamper detection with self-correction hybrid spatial-dct domains image authentication technique, in Proc. IEEE ISSPIT'07,Cairo, Egypt, 2007.

[Hol00] Holliman, M. and Memon, N. : Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes, IEEE Trans. on Image Processing, vol. 9, no. 3, pp.432-441, March 2000.

[Kir02] Kirovski, D. and Petitcolas, F. A. P. : Blind pattern matching attack on watermarking systems, IEEE Trans. on Signal Processing, pp.1-9, 2002.

[Kut00] Kutter, M. Voloshynovskiy, S. and Herrigl, A. : The watermark copy attack, in Proc. SPIE Elect. Imaging, San Jose, USA, Jan. 23-28, 2000.

[Lan99] Lan, T. and Tewfik, A. H. : Fraud detection and self embedding, in Proc. ACM Multimedia'99, Orlando, FA, 1999.

[Lin99] Lin, E. and Delp, E. : A review of fragile image watermarks, in Proc. Of the ACM Multimedia and Security Workshop, 1999, pp. 25-29.

[lin00] Lin, C. and Chang, S. : Semi-fragile watermarking for authenticating JPEG visual content, SPIE Security and Watermarking of Multimedia Contents II EI '00, SanJose, CA, Jan. 2000.

[Lin01a] Lin, C. and Chang, S. : A robust image authentication method distinguishing JPEG compression from malicious manipulation, IEEE Trans. On Circuits and Systems of Video Technology, vol. 11, no. 2, Feb. 2001.

[Lin01b] Lin, C. and Chang, S. F. : SARI: Self-authentication and recovery image watermarking system, Proceedings of the ninth ACM Conference on Multimedia, Ottawa, Canada ,2001.

[Lin07] Lin, C. Su, T. and Hsieh, W. : Semi-fragile watermarking scheme for authentication of JPEG images, Tamkang Journal of Science and Engineering, vol. 10, no. 1, pp. 57-66, 2007.

[Lue08] Lue, H. Lu, Z. Chu, S. and Pan, J. : Self embedding watermarking scheme using halftone image, IEICE Trans. Inf.&Syst., vol.E91-D, no.1, Jan.2008.

[Mae06] Maeno, K. Sun, Q. Chang, S. and Suto, M. : New semi-fragile authentication watermarking techniques using random bias and nonuniform quantization, IEEE Trans. On Multimedia, vol. 8, no. 1, Feb. 2006.

[Men01] Menezes, A. J. Orschot, P. C. and Vanstone, S. A. : Handbook of Applied Cryptography, CRC Press, 2001.

[Mih01]Mihçak, M. K. and Venkatesan, R. : New iterative geometric methods for robust perceptual image hashing, in Proc. ACM Workshop Securityand Privacy in Digital Rights Management, Philadelphia, PA, Nov. 2001.

[Sch96] Schneier, B. : Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, USA, 1996.

[Swa06] Swaminathan, A. Mao, Y. and Wu, M. : Robust and secure image hashing, IEEE Trans. On Information Forensics and Security, vol. 1, no. 2, June 2006.

[Tan08] Tang, Z. Wang, S. Zhang, X. Wei, W. and Su, S. : Robust image hashing for tamper detection using non-negative matrix factorization, Journal of Ubiquitous Convergence and Technology, vol. 2, no. 1, may 2008.

[Ven00] Venkatesan, R. Koon, S. M. Jakubowski, M. H. and Moulin, P. : Robust image hashing, in Proc. IEEE Int. Conf. Image Processing, Vancouver,BC, Canada, Sep. 2000, vol. 3, pp. 664–666.

[Wan08] Wang, S. and Tsai, S. : Automatic image authentication and recovery using fractal code embedding and image inpainting, Journal of the Pattern Recognition Society, vol. 41, pp. 701 – 712, 2008.

[Won97] Wong, P. W. : A watermark for image integrity and ownership verification, in Proc. IS & TPIC, Portland, OR, USA, May 1997.

[Won98] Wong, P. W. : A public key watermark for image verification and authentication, in Proc. ICIP, NY, USA, Oct.4-7, 1998, pp.425–429.

[Won01] Wong, P.W. and Memon, N. : Secret and public key image watermarking schemes for image authentication and ownership verification, IEEE Trans. on Image Processing, vol. 10, pp. 1593-1601. Oct. 2001.

[Yeu97] Yeung, M. and Mintzer, F. : An invisible watermarking technique for image verification, in Proc. ICIP'97, Santa Barbara, CA, USA, 1997.