International Journal of Computer Science and Applications,
©Technomathematics Research Foundation
Vol. 12, No. 2, pp. 99 – 119, 2015

SECURE IMAGE VERIFICATION IN JOINTED FREQUENCY DOMAINS

RELU-LAURENTIU TATARU

Faculty of Electronics, Telecommunications and Information Technology,
POLITEHNICA University of Bucharest, 1-3, Iuliu Maniu Bvd., Bucharest 6, Romania
Email: tataru.relu@yahoo.com

The paper illustrates an image hashing scheme and an image verification system based on jointed hashing and watermarking algorithms. We propose an image hashing scheme based on a set of robust features extracted from jointed DCT-DWT frequency domains and enciphered using an efficient chaotic system. The proposed hashing algorithm and a rotation-scale-translation (RST) invariant watermarking scheme are used to design a robust verification system for digital images. A robustness investigation is conducted using compression, noise addition, filtering and geometric transforms. The Checkmark benchmark is used to achieve most of the attacks. Our experiments reveal the performances of the hashing scheme and also the success of the proposed verification system against common image processing attacks.

Keywords: image hashing; digital watermarking; DCT; DWT; Fourier Transform.

1. Introduction

Increasing popularity of mobile devices (smartphones, tablets and other gadgets) with high quality cameras and development of high resolution digital cameras raised the number of digital images published over the Internet. An important contribution to this evolution was due to fast development of social networks and services dedicated to media sharing. All these services increased the number of images released in the online environment. This evolution raised the need of addressing some important issues such as: ownership protection, content authentication, fast image indexing and retrieval.

Digital watermarking was proposed as a first solution for these issues. Watermarking techniques are used to embed binary signatures (watermarks) modifying directly the content of the image. Watermarked images are identified detecting the presence of the watermark inside the image. However, many watermarking methods do not solve the problem of content identification. Most of the embedded watermarks are independent of the image content. An alternative of these problems was provided by the concept of image hashing.

Perceptual image hashing was used in the last years to solve ownership disputes, authentication and image retrieval problems. An image hashing scheme usually provides a sequence of values defining the visual characteristics of the image. The result is an image fingerprint usually protected with cryptographic techniques. Compared with conventional hash functions from cryptography, perceptual hash functions designed for images tolerate those modifications which do not affect the content of the image (i.e. compression, filtering, noise addition etc.). Thus, images with different representations, but with the same visual content, provide the same or very close hash values.

In the current work we propose a new algorithm which includes a perceptual hashing scheme for digital images secured with chaotic sequences. The interest was to find a suitable image representation space providing robust features to create an image hash for serving authentication of digital images, copyright protection and easy image database management. Our idea was to combine efficiently existent spaces in frequency domain for feature extraction. We also propose a verification system that exploits the advantages provided by our hashing scheme and the features of a robust watermarking scheme. The proposed system is designed using two schemes that eliminate each other's flaws.

This paper is organized as follows: Section 2 describes the design principles and properties of image hashing regarding some state of the art principles, Section 3 presents the construction of the image hashing scheme, Section 4 illustrates the integration of the proposed hashing scheme in a hash based watermarking scheme, Section 5 points out the simulations results, Section 6 presents some practical applications of the proposed hashing scheme and verification system and Section 7 concludes the work.

2. Image Hashing – design principles and properties

2.1. *Design Principles*

It is widely accepted that the basic components of perceptual image hashing are image pre-processing, feature extraction, feature post-processing and randomization.

By image pre-processing a new scaled version of the digital image is obtained. This is usually achieved by reducing the representation space of the original image without losing the significant features of the content. The result is usually a scaled version of the input image, facilitating operations with low computational cost.

Feature extraction is the next phase in the construction of the image hash. This is an important stage because the feature space influences directly the robustness of the hash function. Depending on the application type of the image hashing scheme, a certain domain for feature extraction could be imposed. Robust algorithms are usually relied on frequency transforms for feature extraction. A scheme resilient to JPEG compression may use Discrete Cosine Transform (DCT) for the feature extraction stage. In [Yu et al. (2010)], the authors proposed a scheme based on the statistical modeling of DCT coefficients as a Gaussian distribution. The authors assert that invariance of DCT coefficients achieves robustness against attacks such as JPEG compression, filtering, scaling, brightness adjustment, histogram equalization and even small angle rotations. In [Fridrich et Gojan (2000)], the authors illustrate the advantage of considering low frequencies in DCT domain for feature extraction. Their reasoning is based on the properties of low frequency DCT coefficients which preserve the significant information of the image. Any modification in these frequencies is noticeable on the host image. Other transforms are also used in achieving perceptual image hashing. Guo and Dimitros proposed in [Guo et Dimitros (2007)] the extraction of a robust feature set by using Discrete Wavelet Transform (DWT) followed by Radon transform. The hash value is generated using a probabilistic quantization. This hash value is resilient to image compression, filtering, scaling and rotations. The authors assert good results even for image tampering. When high robustness against geometric attacks is required, the feature set may be extracted using transforms such as Discrete Fourier Transform or Mellin Fourier Transform. Swaminathan et al. propose in their work an image hashing algorithm based on Mellin Fourier Transform. They claim to obtain good results against rotation operations up to 10^{o} and 20% cropping. This class of methods usually performs well against this type of attacks. However, they may be less robust against other common attacks such as noise addition.

The feature extraction process could be also realized in other transform domains such as Singular Value Decomposition (SVD) [Kozat et al. (2004)] or Fast Johnson-Lindenstrauss Transform (FJLT) [Lv et Wang (2008)].

The feature set extracted from a transform domain is generally built to assure the goals of the image hashing scheme.

Post-processing stage is usually a compression of the previously extracted features. A feature reduction technique is usually applied in this purpose in order to obtain a final binary feature set. This step is commonly realized using one of the following techniques: random projection of the feature set in another space, direct compression of the feature set, feature set quantization, clustering or by computing a cryptographic hash of the feature set.

Randomization is the last step in achieving the final perceptual hash value of the image. This step is mandatory and assures the unpredictability of the hash value obtained for each digital image in the presence of the secret key.

2.2. *Properties of image hashing*

The final hash algorithm should provide the following features:

- **one-way** – the hash produced from an image makes impossible the recovery of the input image; Moreover, given the original image and the corresponding hash value, the recovery of the secret key is infeasible;
- **collision-free** – perceptually different images provide different hash values with a high probability;
- **key-dependence** – the hash value is highly dependent on the secret key;
- **robustness** – the hash obtained from an image should be invariant to common image processing modifications i.e. noise addition, compression, geometric transforms, filtering, brightness and contrast adjustment;

The use of the hash value in verifying an image with a pair is resumed to the direct comparison of the two binary hashes. Few or zero differences between the hash values validate the authenticity of one image with respect to the other.

A goal of this paper is to propose a robust hash function which respects both design principles and general features of image hashing algorithms. The proposed algorithm is potentially capable of solving copyright disputes, authenticating similar images and retrieving the image content from large image databases.

3. Image Hashing In Jointed Frequency Domain

As most of image hashing algorithms, our scheme computes a global set of features from a digital image. A feature set is used to compute a perceptual hash value. The feature set is enciphered using a chaotic system. The novelty of the proposed algorithm is given by the feature set construction and the use of a proven secure chaotic system for the feature set encryption. A description of the proposed image hashing scheme is illustrated in the following subsections.

3.1. *Image Pre-Processing*

A color digital image is converted to grayscale and resized to a default size mxm. The resizing procedure allows fast operations on the grayscale image. Comparing to the original input image, the content of the new image is not changed.

For the feature extraction step, the grayscale image is converted in frequency domain. This is the most significant stage in computing a robust feature set. A feature set built in frequency domain provides good robustness to certain classes of attacks.

3.2. *Feature Extraction*

For the feature extraction step, the grayscale image is converted in frequency domain. This is the most significant stage in computing a robust feature set. A feature set built in frequency domain provides good robustness to certain classes of attacks.

Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) are jointly used for extracting the global feature set. The first level DWT decomposition assures the separation of the information from the grayscale image in frequency sub-bands LL_1 (low-low), HL_1 (high-low), LH_1 (low-high) and HH_1 (high-high). The LL_1 sub-band carries most of the information from the grayscale image. For this reason, we consider the LL_1 sub-band in the feature generation process. A n –level decomposition is performed, considering LL_{n-1} sub-band ($n \geq 2$) at each iteration. At the n -level decomposition, the LL_{n-1} sub-band is obtained. This sub-band provides a matrix preserving most of the correlations from the original grayscale image. The DCT transform is applied for this sub-band on blocks of size kxk . The Wavelet distribution from the LL_n sub-band is changed at the block level, and the new distribution follows the properties of DCT transform. Most significant frequencies are positioned in the top-left corner of the block, and the less significant frequencies are grouped in the bottom-right, according to the DCT distribution. The first term of each DCT block, i.e. the (0,0) frequency, integrates the most important part of information from the block. This frequency, also called DC term, is extracted from each DCT block.

3.3. *Feature Post-Processing*

A feature vector containing the DC's of all DCT blocks computed from the LL_n sub-band is obtained at the previous step. At the current step, we apply a binarization technique for the feature vector. This is achieved by comparing each component of the feature set with the global mean of the feature set. Binary 0 is used to represent DC values under the mean and binary 1 is used to represent DC values above the mean. Thus, we obtain a binary fingerprint of the digital image.

3.4. *Feature Randomization*

This step is mandatory in order to assure the confidence of the binary feature set. The security of the feature set is obtained by direct enciphering with a recently proposed chaotic system. The chaotic generator proposed by [Vlad et al. (2013)] is used as a stream cipher. This generator is based on tent-map and the running-key principle. The tent-map has the following formula:

$$x_{n+1} = f(\mathrm{x_n}) = \begin{cases} \dfrac{x_n}{a}, & 0 \le x_n \le a \\ \dfrac{1-x_n}{1-a}, & a < x_n \le 1 \end{cases} \tag{1}$$

where $a \in (0,1) \backslash \{0.5\}$ is the control parameter of tent-map, $\mathrm{x_n}$ is the n^{th} value of the chaotic sequence generated using the tent-map and x_0 is the initial value from $(0,1)$ range. Binary sequences Z_i are generated using the X_i real value sequences generated with the tent-map and binarization threshold c.

$$z_{i,j} = \begin{cases} 0, & 0 \le x_{i,j} \le c \\ 1, & c < x_{i,j} \le 1 \end{cases} \tag{2}$$

where $z_{i,j}$ is the j^{th} element of the chaotic binary sequence Z_i and $x_{i,j}$ is the j^{th} element of the chaotic non-binary sequence X_i.

A running-key procedure is applied for typical Z_i binary sequences in order to obtain binary i.i.d. sequences compatible with the fair coin model.

The enciphering key used for the proposed image hashing algorithm is a binary sequence based on five additions of typical sequences Z_i, as shown in equation 3.

$$Y = \sum_{i=0}^{4} Z_i \bmod 2 \tag{3}$$

According to [Vlad et al. 2013], the binarization threshold was considered equal to the control parameter $(c = a)$. The security of the method was theoretically and experimentally proven for a control parameter a in the range $(0.39,\, 0.61) \backslash \{0.5\}$ for 5 modulo 2 additions.

The secret key K of the system is given by the initial values of each non-binary sequence X_i and the control parameter:

$$K = \left(x_{00} \,\|\, x_{01} \,\|\, x_{02} \,\|\, x_{03} \,\|\, x_{04} \,\|\, a\right) \tag{4}$$

Note: As already suggested in [Vlad et al. (2013)], the additions number could be increased, extending the range of the control parameter a. This result leads to a larger selection of the secret key. The construction principle of the chaotic system used to generate the pseudo-random key to encipher the feature set is presented in Figure 1.

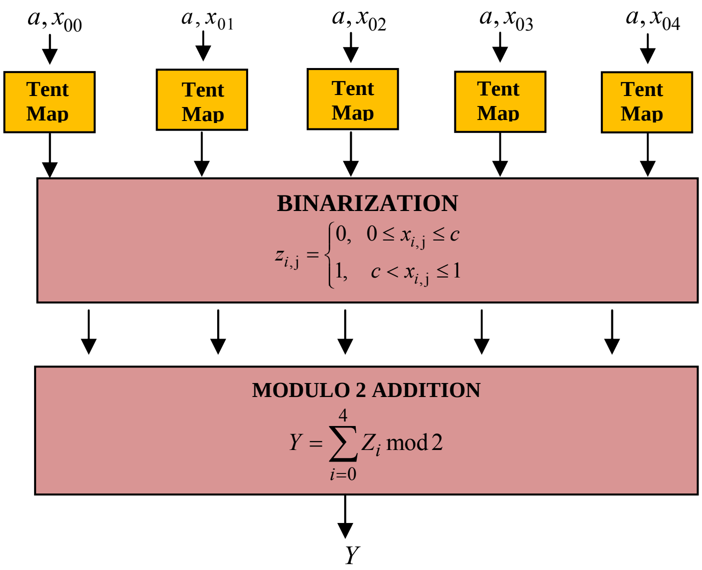


Fig. 1. Chaotic System.

The step-by-step construction of the perceptual image hashing scheme according to the description is next presented:

1. The input color image is transformed to grayscale.

2. The grayscale image is resized to a standard mxm dimension using the bicubic interpolation.

3. A n –level DWT Haar decomposition is applied on the grayscale image.

4. LL_n sub-band is divided in non-overlapping kxk blocks and DCT transform is applied on each block.

5. DC coefficients are extracted from all DCT blocks and the vector V containing the features of the image is built. The length of the feature vector V is given by the formula:

$$l = \left(\frac{m}{k \cdot 2^n}\right)^2 \tag{5}$$

6. The mean value m_{dc} of the feature vector is computed.

7. The feature vector V is binarized and a new binary feature vector W is obtained according to the formula:

$$w_i = \begin{cases} 0, & v_i < m_{dc} \\ 1, & v_i \geq m_{dc} \end{cases} \tag{6}$$

where $V = (v_i)_{i=1...l}$ and $W = (w_i)_{i=1...l}$

8. A pseudo-random sequence $Y = (y_i)_{i=1...l}$ is generated using the chaotic system presented in Figure 1, with the secret key K.

9. The binary feature vector W is enciphered using the pseudo-random sequence Y and the final hash value $H = (h_i)_{i=1...l}$ is obtained, where: $h_i = x_i \otimes y_i$, $i = 1...l$

At the end of all steps, a hash value with l – bits length is obtained. The proposed system is illustrated in Fig. 2.

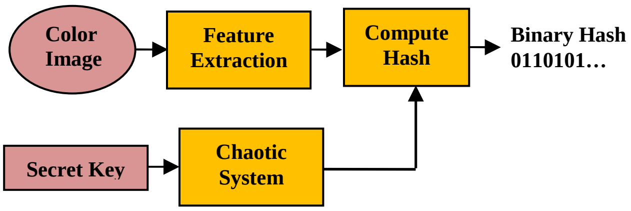


Fig. 2. Proposed Hashing System.

4. Integrating the hashing scheme in watermarking applications

4.1. *Jointed watermarking and hashing scheme*

Our claim is that the proposed hashing scheme (HS) can be individually used in certain applications which do not require the resistance of the scheme to important geometric transforms. The domain features used to build the image signature is theoretically robust to compressions, noise addition and filtering, but is not resilient to important geometric transforms such as rotation. Our idea was to combine the proposed hashing scheme with a Rotation-Scale-Translation (RST) invariant watermarking method. Most of the watermarking schemes robust to geometric attacks are built in Fourier and Mellin Fourier domain [Poljicak et al. (2011)][Kim et al. 2004][Solachidis et Pitas (2001)]. We decided to use the watermarking method presented in [Poljicak et al. (2011)] because of its effectiveness and robustness against geometric attacks.

The idea behind our method is illustrated in Figure 3. The proposed hashing scheme assures the extraction of approximately the same hash values from the original image and its watermarked version as long as the watermarking method does not degrade considerably the image content. This property, referred as transparency or fidelity for most of the watermarking methods, is a mandatory request for an efficient watermarking scheme. Such capability is also provided by [Poljicak et al. (2011)] where the authors claim to obtain a very good level of the Peak Signal-to-Noise Ratio (PSNR) between original and watermarked image. Thus, the use of the hybrid method is highly-functional.

This jointed method is referred in the following sections as HWS and it is designed to work with an original verification system presented in Section 4.3. A short description of the watermarking scheme (WS) [Poljicak et al. (2011)], is illustrated below.

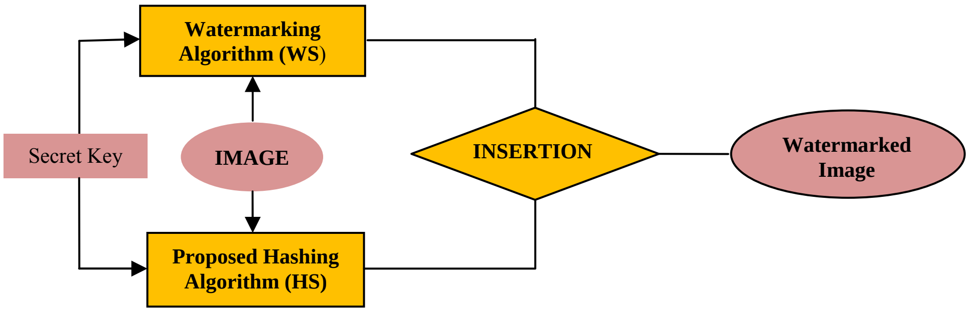


Fig. 3. Jointed Hashing-Watermarking Scheme (HWS).

4.2. *RST Invariant Watermarking (WS)*

4.2.1 Watermark Embedding

1. HS scheme is used to compute the perceptual hash of the image and to provide the hash value as watermark for WS. The hash value is the binary vector H of length l .
2. The color image is transformed from RGB space to YCbCr color space.
3. Fourier transform is applied to the luminance component (Y).
4. Low frequency magnitude coefficients of the transform are moved to center.
5. According to the implementation radius r , elements of the watermark matrix are calculated according to Eq. 7:

$$W(\mathrm{x_i}, y_i) = H(\mathrm{j})\left[\frac{1}{9}\sum_{s=-1}^{1}\sum_{t=-1}^{1} M(\mathrm{x_i}+\mathrm{s}, y_i + t)\right] \quad (7)$$

where $W(\mathrm{x_i}, y_i)$ are elements of the watermark matrix, $H(\mathrm{j})$ is the j'th element of the hash vector, M is the matrix of magnitudes provided by the cover image and $(\mathrm{x_i}, y_i)$ coordinates on the circle of radius r around the center of the cover image.

6. The watermark matrix is embedded into the magnitude coefficients using Eq. 8:

$$M_w(\mathrm{x,y}) = \mathrm{M(x,y)} + \alpha \cdot \mathrm{W(x,y)} \quad (8)$$

where M is the magnitude in the cover image, $(\mathrm{x,y})$ pair denotes the position of the magnitude in the image, W is the watermark matrix, α is the strength factor, M_w is the magnitude of the watermarked image;

7. The magnitude coefficients are combined with the corresponding phase components and transformed back to spatial domain.
8. The YCbCr to RGB conversion is applied and the watermarked image is obtained.

4.2.2 Watermark Detection

1. The HS is applied to the watermarked image to compute the hash value if the hash value of the reference image is not available.
2. The watermarked image is scaled to the standard dimension and transformed to Fourier domain.
3. Low frequency magnitude coefficients of the transform are moved to center.
4. A blind iterative search of the watermark is started in a predefined interval of radiuses $[r_{\min}, \mathrm{r}_{\max}]$.
5. Each extracted vector is then resized to length l' according to the formula:

$$l' = \mathrm{r}_{\max} \cdot \pi \tag{9}$$

6. The values of each vector are normalized to interval $[0,1]$.
7. The cross covariance between each vector and the reference hash (or extracted hash if reference hash misses) is computed. The investigated image is considered to be watermarked if the cross covariance exceeds a predefined threshold wTh.

4.3 Proposed Verification System

The final verification system provides a dual decision concerning the relation between a reference and a target image.

Firstly, the hash value of the targeted image is computed using HS with the corresponding secret key. The hash value is computed and compared with the reference hash stored independently and a former decision is obtained. Also, the presence or absence of the watermark is detected using WS and a latter decision is provided. Whenever a positive decision is achieved, the targeted image is considered to be the same or a modified version of the image providing the reference hash value. When both decisions are negative, the target image and the reference image providing the reference hash are considered to be independent.

The design of the verification system is given in Figure 4 and the decision making strategy uses the OR logical operation to provide the final result, as presented in Table 1.

Note: Only the presences of the target image and reference hash are required.

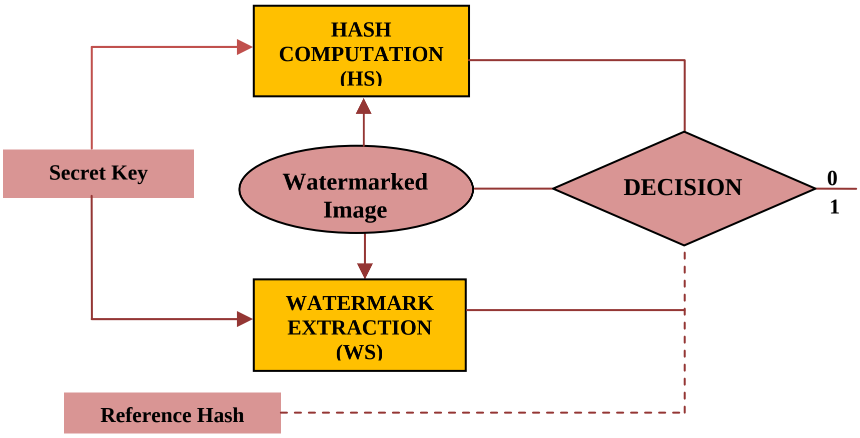


Fig. 4. Verification System for HWS.

Table 1. Verification System's Decision.

Case	HS	WS	Decision
1	1	1	1
2	1	0	1
3	0	1	1
4	0	0	0

As it can be observed from Table 1, in the first case, both HS and WS provide a positive answer i.e. the hash computed using HS and the reference hash are similar and the presence of the watermark is detected by WS. This case conducts to a situation when the computed hash may be compared directly with the extracted watermark. For this kind of situations, there is no need for the reference hash to establish the presence of the watermark. Only the failure of one of the schemes would require the presence of the reference hash to verify adequately the targeted image, which conducts to cases 2 and 3.

In case 2, the extracted watermark is different from both computed hash and reference hash. However, the extracted hash value and the reference hash are similar. Thus the detection of the watermark was perturbed by a transformation that didn't distort HS. In this case, the success of the verification system is only assured by HS. A false response of the verification system is provided when HS fails and a collision occurs i.e. a similar hash value is obtained for a perceptually different image. A collision resistance analysis for HS is presented in Section 5.

In case 3, HS was distorted by a perturbation applied to the image. This perturbation did not affect WS. The use of the reference hash assures the detection of the watermark, thus the success of the verification system. Our verification system performs well in this case if and only if the decision of WS is correct. Details about the accuracy of WS are included in [Poljicak et al. (2011)].

Case 4 brings up two sub-cases: the former is when both schemes fail, which leads to the fail of the verification system. The latter is when the verification system performs well and there is no link between the target image and the image providing the reference hash.

A final value obtained for our verification system in the process of verifying a reference image with a target image is provided by the formula illustrated below:

$$d = sign\left[\max\left(\frac{hTh}{h} - 1, \frac{w}{wTh} - 1\right)\right] \tag{10}$$

where h is the value of the Bit-Error-Rate (BER) between the hashes of the reference image and the target image, hTh is the BER threshold value for HS, w is the cross covariance value obtained between the reference hash value i.e. the inserted watermark and the extracted watermark and wTh is the cross covariance threshold value for WS.

According to Table 1, a positive d confirms the similarity between the reference and target image. The confidence in a correct positive verification of the target image is increased when the maximum value of the formula is decided between two positive values.

5. Experimental Results

5.1. *Simulations Description and Results*

Our simulations are illustrated for the hashing scheme (HS), for the watermarking scheme (WS) and for the hybrid hash based watermarking scheme (HWS).

5.1.1 Simulation of HS

Several parameters were tested during the simulations for HS. This parameters were the dimension of the resized image (M), the n – level of the DWT transform and the block size kxk of the DCT transform. In this paper we illustrate the performances of the proposed algorithm for the following parameters: $M = 256$, $n = 2$, $k = 4$, $l = 256$

The investigation of the proposed scheme was performed for hash values with constant binary lengths $l = 256$. All feature sets were encrypted using the 256–bit length chaotic sequences generated according to [Vlad et al. 2013]. Each binary error from the binary hash contributes to the final error with the value $\frac{1}{l}$ (i. e. $err \approx 0.0039$).

For our purposes we used two different databases with resized images between 512x384 and 1024x1024. The investigation of the proposed algorithm was conducted using the following databases: Uncompressed Color Image Database (UCID), and Break Our Steganographic System (BOSS) database, both databases with color images. 1000 uncompressed images with different formats (tif and bmp) were randomly chosen from each database to create our testing set containing 2000 images. Images from BOSS database were converted from CR2 (Cannon Raw file format) in bmp format.

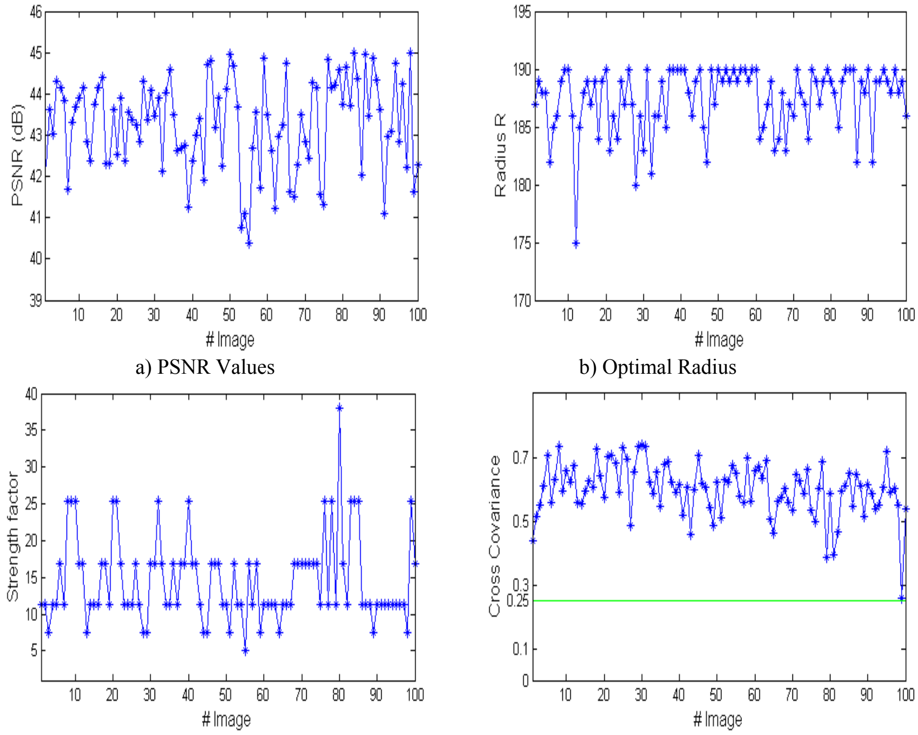
To define the similarity between the reference hash and the target hash, we used the Bit Error Rate (BER) as a measure of number of differences. The BER value for two hashes is given by the ratio between the number of erroneous bits and the total number of bits. A perfect similarity equates with a 0 BER value and two completely different images should provide a BER value close to 0.5 (not similar).

5.1.2 Simulation of WS

The watermarking method from [Poljicak et al. (2011)] was tested using 100 images to confirm the performances claimed by the authors. Images were selected consecutively from BOSS database, and resized to 512x512. All watermarks inserted into the images had a 256-bit length. The watermark length is slightly different from the original paper, where the standard watermark length was 200. This difference was introduced in order to quantify our idea of using the hash value of the image as watermark.

In terms of transparency, WS proved its effectiveness even for a 256-bit watermark length. This is proved by the PSNR values, calculated between the original and watermarked versions for the images used in our simulations. The evolution of the PSNR for 100 images is illustrated in Figure 5a. The watermarked versions of the images were obtained using optimized values for radius r and strength factor α. All corresponding values of the radius and the strength factor for all100 images are illustrated in Figure 5b and Figure 5c.

Cross covariance calculated between the inserted and extracted watermark was used to decide the presence or the absence of the watermark. The threshold value 0.25 obtained by the authors in [Poljicak et al. (2011)] proved to be enough to separate the watermarked from unmarked images, as it could be seen in Figure 5d.



a) PSNR Values

b) Optimal Radius

c) Strength factor α

d) Cross Covariance

Fig. 5. Performances of WS.

5.1.3 Simulation of HWS

Our experiments were conducted as follows. The hashes of 100 cover images and the hashes of their watermarked versions were computed and compared. The main purpose was to investigate the impact of the watermark embedding in the hash computation process and to test HWS's reliability. Bit-Error-Rate (BER) between hashes of cover and watermarked images was used to quantify the similarity between hashes of image pairs.

The results obtained for 100 images are presented in Figure 6.

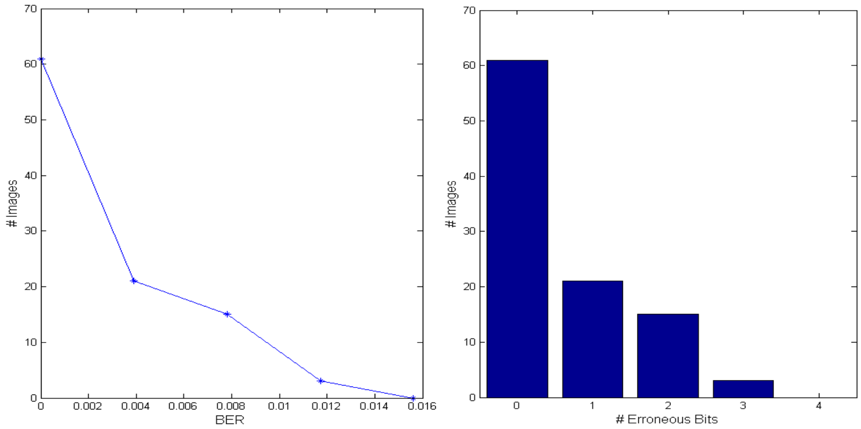


Fig. 6. a) BER between hashes. b) Erroneous bits between hashes.

The maximum number of erroneous bits calculated among 100 images was 3 from a total of 256 bits of the hash value and this result was obtained for only 3 images. Figure 6a illustrates very small values of the BER between hashes of cover-watermarked pairs. This result supports our claim that very similar hash values are obtained from cover-watermarked image pairs. Such results confirm the functionality of the proposed hash based watermarking scheme, thus the correctness of our verification system.

5.2. ***Robustness against JPEG compression***

5.2.1 Robustness of HS against JPEG compression

Our tests for the proposed hash function aimed primarily the resilience of the method to the JPEG compression with different quality factors$\left(Q=10,\ 20...100\right)$. A number of 1000 uncompressed digital images from each database were compressed with different quality factors, from 10 to 100. All hash values computed from uncompressed images were compared with hash values calculated for the corresponding JPEG image compressed with quality factor Q. Our results, illustrated in Table 2 and Table 3, prove the robustness of the proposed method at compressions down to very low quality factors. A DWT 2-level decomposition of the image jointed with the DCT 4x4 decomposition provides good results in terms of robustness against JPEG compression for both image sets even for a small HS threshold *i.e* 0.05

Table 2. JPEG compression results for BOSS database.

Break Our Steganographic System Database – BOSS(1000 images)				
BER (B) Quality (Q)	< 0.05	< 0.10	< 0.15	≥ 0.15
10%	91%	99%	100%	0%
20%	99%	100%	100%	0%
30%	100%	100%	100%	0%
40%	100%	100%	100%	0%
50%	100%	100%	100%	0%
60%	100%	100%	100%	0%
70%	100%	100%	100%	0%
80%	100%	100%	100%	0%
90%	100%	100%	100%	0%
100%	100%	100%	100%	0%

Table 3. JPEG compression results for UCID database.

Uncompressed Colour Image Database – UCID (1000 images)				
BER (B) Quality (Q)	< 0.05	< 0.10	< 0.15	≥ 0.15
10%	91%	99%	100%	0%
20%	98%	100%	100%	0%
30%	99%	100%	100%	0%
40%	100%	100%	100%	0%
50%	100%	100%	100%	0%
60%	100%	100%	100%	0%
70%	100%	100%	100%	0%
80%	100%	100%	100%	0%
90%	100%	100%	100%	0%
100%	100%	100%	100%	0%

5.2.2 Robustness of the WS against JPEG compression

One of the claims from [Poljicak et al. (2011)] is that the watermarking method is resilient to JPEG compression with a good percentage for high values of the quality factor. We also tested this feature using our image set with 100 images. The results for three quality factors (90, 50, 10) are illustrated in Figure 7. All results obtained for sets of 100 images at 10 quality factors are also presented in Table 4. As it can be seen, our image set proved the robustness of WS for high quality factors. However, the WS proved to be vulnerable at low and medium quality factors.

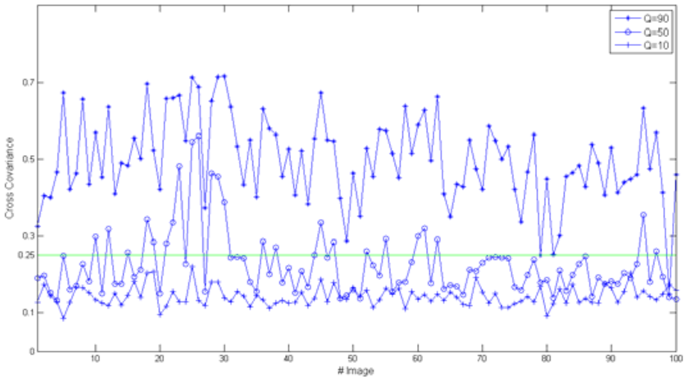


Fig. 7. WS's robustness against JPEG compression for three quality factors.

Table 4. WS's accuracy against JPEG compression for ten quality factors.

Quality Factor	**Total image number: 100** **(** $wTh = 0.25$ **)**	
	Detected Correctly	**Undetected**
100%	100	0
90%	99	1
80%	89	11
70%	63	37
60%	43	57
50%	24	76
40%	17	83
30%	7	93
20%	0	100
10%	0	100

5.2.3 Robustness of the HWS to JPEG compression

Observing the performances separately for both schemes, it can be easily seen that the only use of the response provided by the hashing scheme allows a much better association between the uncompressed image and its JPEG version. Differently from experiments illustrated in 5.2.1, the comparison of hashes is done between the hash provided by the watermarked uncompressed image and its compressed version. 999 from 1000 image pairs (100 watermarked uncompressed images with 10 JPEG compressed images for each image, corresponding to 10 quality factors: 10, 20…100) were correctly identified by the HWS considering the threshold value of the hash scheme $hTh = 0.15$.

The results for our image set at a 10%, 50% and 90% JPEG compression rate are presented in Figure 8. The complete results concerning the accuracy of the HWS method can be seen in Table 5.

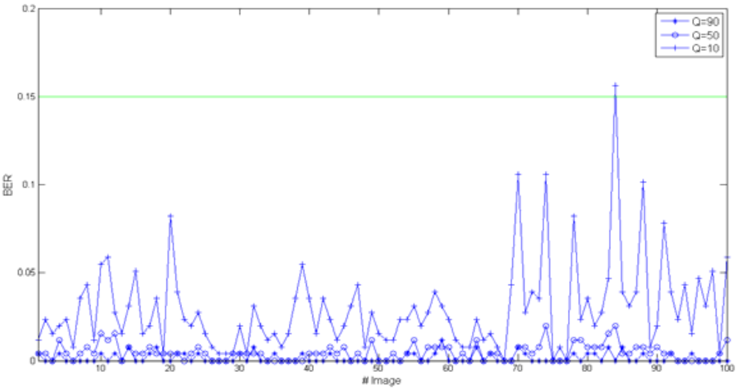


Fig. 8. HWS's robustness against JPEG compression for three quality factors.

Table 5. HWS's accuracy against JPEG compression for ten quality factors.

Quality Factor	**Total image number: 100** ($hTh = 0.15$)	
	Identified Correctly	**Unidentified**
100%	100	0
90%	100	0
80%	100	0
70%	100	0
60%	100	0
50%	100	0
40%	100	0
30%	100	0
20%	100	0
10%	99	1

5.3. *Robustness against other image processing attacks*

The hashing algorithm (HS) exploits the advantage of transform domain and also provides some robustness against common image processing attacks such as filtering, noise addition and some geometric transforms. The watermarking scheme (WS) is robust against geometric transforms, filtering and noise addition, but quite vulnerable to compressions with low and medium quality factors.

The jointed use of the HS and WS in HWS provides better results in terms of robustness against common image processing attacks, both theoretically and experimentally.

In order to prove our claims, we illustrate some of our results in Table 6. Several attacks were applied on the Lena image stored in BMP format with size 512x512. Most of the attacks were conducted using the Checkmark framework [Pereira et al. 2001].

The BER value calculated between the hashes was used as metric to measure the similarity between the original and attacked images and the cross covariance was used to detect the presence of the watermark.

The proposed hashing scheme performed well under filtering, noise addition and some geometric attacks for the Lena image and also for 100 images randomly chosen from the BOSS database. However, the method proved to be vulnerable against geometric manipulations such as rotations greater than 2° and also for important cropping attacks. For all this attacks, the WS performed well, with a high accuracy. The situations were WS generally fails (JPEG and JPEG 2000 compressions with medium and low quality factors), are well decided by HS. As it can be seen for the Lena image, in all situations where one of the schemes failed, the final decision provided by the verification system was correct.

Table 6. Robustness of HS, WS and HWS against common image processing attacks.

Attack	HS	Detection	WS	Detection	HWS	Detection	Final
Name	(BER)		(CC)		(sign)		Decision
Gaussian Noise(1) (2)	0	√	0.67	√	1	√	√
Hard Thresholding(1) (2)	0.0039	√	0.36	√	1	√	√
Soft Thresholding(1) (2)	0.0039	√	0.36	√	1	√	√
Wiener Filtering	0.0039	√	0.39	√	1	√	√
Median Filtering(1) (2)	0.0078	√	0.30	√	1	√	√
Sharpening(1)	0.0160	√	0.69	√	1	√	√
Stirmak(1) (2)	0.0156	√	0.15	x	1	√	√
JPEG compression 10(1)	0.0117	√	0.15	x	1	√	√
Wavelet compression 10(1)	0.0117	√	0.17	x	1	√	√
Denoising with Remodulation(1) (2)	0	√	0.45	√	1	√	√
Sample Down(1) (2)	0.0117	√	0.15	√	1	√	√
Template Remove	0.0039	√	0.34	√	1	√	√
NullLineRemove(1) (2)	0.0039	√	0.52	√	1	√	√
Scale 75%	0	√	0.69	√	1	√	√
Scale 150%	0	√	0.68	√	1	√	√
Scale 200%	0	√	0.68	√	1	√	√
Rotation -60°	0.5039	x	0.45	√	1	√	√
Rotation -30°	0.4336	x	0.55	√	1	√	√
Rotation 30°	0.4297	x	0.52	√	1	√	√
Rotation 60°	0.4219	x	0.42	√	1	√	√
Central Crop 25%	0.2461	x	0.64	√	1	√	√
Central Crop 50%	0.3516	x	0.48	√	1	√	√

(1) This attack was conducted using Checkmark Benchmark. (2) Worst result is provided for this attack

5.4. *Robustness against malicious attacks of the hashing scheme*

An attacker may perform two types of malicious attacks on HS. The former implies the counterfeiting of both digital image and hash value. A second type of manipulation is by direct modification of the image content, while retaining the hash value of the image. The first class of attacks is unfeasible for the proposed scheme due to the secrecy of the enciphering key of the chaotic system. The resilience of the proposed algorithm against this class of attacks is given by the strength of the chaotic system and the secrecy of the key. For the latter class of attacks, the image may be maliciously distorted using the following techniques: object addition, object removal and object altering. Our block based hashing scheme is less sensitive to local modifications. Changing small parts of the image is reflected by the DC coefficients obtained for the DCT transform applied for the LL sub-band. However, these changes are not fully reflected in the binary feature vector. This is because the averaging procedure used to collect the feature set is not always sensitive to this type of modification. The use of a threshold very close to 0 may assure a partial robustness against this class of attacks.

A feasible solution for images containing text elements (letters, numbers, visible watermarks etc.) is using character identification techniques. All extracted text elements may be hashed using a robust cryptographic hash function. This hash value is concatenated to the perceptual hash value and a final hash is built. The use of SHA-256 as cryptographic hash function assures a 512 bit length of the final hash value.

The proposed verification system is not designed to detect malicious attacks. Generally, the malicious attacks slightly perturb the content of the image, hence the watermark can still be detected and the image will be authenticated.

5.5. *Collision Resistance of the hashing scheme*

A perceptual hashing scheme should provide different hashes for dissimilar images. The proposed algorithm complies with this requirement and provides different hash values for different images. In order to illustrate the collision resistance property of the proposed image hashing scheme an example is illustrated in Figure 9 for images AudiA4_1.jpg and AudiA4_2.jpg (source: www.autovit.ro) The BER value 0.5078 calculated for the images presented in Figure 9, indicates the total difference between the hashes of the two distinct images.

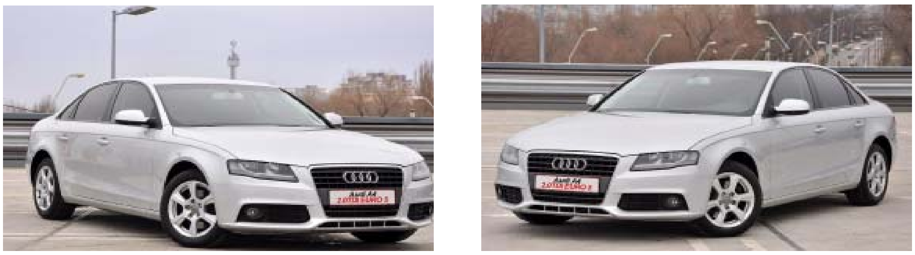


Fig. 9. a) AudiA4_1.jpg b) AudiA4_2.jpg
BER = 0.5078

However, the BER value of dissimilar images is not always close to 0.5. In Figure 10 we illustrate the discriminative capability of the proposed algorithm, by computing the probability density function of BER values for dissimilar images. This result was obtained for 1000 image pairs, randomly extracted from the test databases. The BER values calculated between perceptual hashes of distinct images have a Gaussian distribution, with the mean 0.4763, which is close to the theoretical value 0.5.

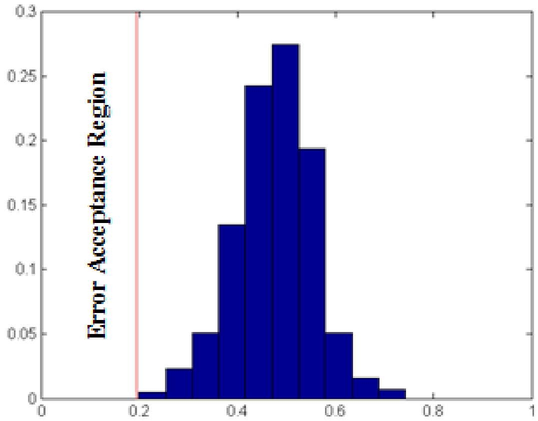


Fig. 10. Probability density function for 1000 dissimilar image pairs.

The minimum value (i. e. 0.1992) is far enough from most of the values obtained for Lena image and its attacked versions using Checkmark Benchmark. A BER threshold value fixed at 0.15 assures very good performances for the perceptual hashing method.

5.6. *Why jointed frequencies perform better?*

A jointed frequency domain proved to be better for computing the perceptual hash from a digital image than a single transform domain. This statement is supported mostly in terms of collision resistance. The only use of the DCT transform for computing the DC coefficients that provide the feature vector tends to decrease the BER values obtained between hashes of dissimilar images. The jointed work of the two transforms catches better the details from each image, providing a much more sensitive feature vector. This sensitivity is enough to obtain different hashes for perceptually different images, maintaining certain robustness against common image processing manipulations.

6. Practical applications of HS and HWS

The goal of the proposed image hashing system alone was to cover the following three topics: image authentication, image retrieval and copyright protection. In all three cases the reference image is required to compute and store the reference hash value. For each target image the hash value is computed using the same secret key as for the reference image. Two hashes are computed at the bit level in order to determine the similarity level.

The hashing verification system is built according to Figure 11. After comparing the two hashes at bit level, a BER value is computed. Depending on the sensitivity of the application integrating the image hashing scheme, different threshold value hTh of the BER should be chosen. BER values above the threshold hTh outputs a binary 0 (non-authentic image) and BER values under the threshold hTh outputs 1 (authentic image). A BER value should be close to 0 when very high sensitivity is required (e.g. authentication of tampered images) and the hTh may be increased when the application is not very restrictive (e.g. applications with content identification such as TinEye, Google Images).

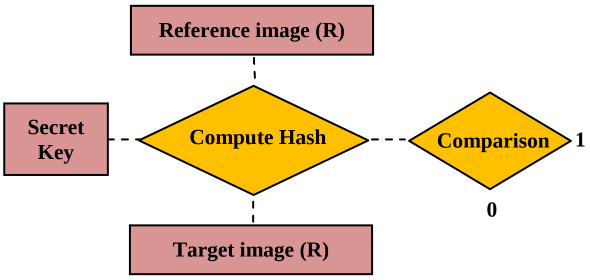


Fig. 11. Hash based verification system.

The proposed hash based watermarking scheme (HWS) can be used as well as other watermarking applications, with a good robustness against the majority of image processing attacks. Hash computation is an additional feature of HWS comparing to existent watermarking applications and its first goal is to increase the capabilities of the classical watermarking application and secondly to enhance the confidence in the answer provided by that watermarking application.

7. Conclusions and future works

In this paper we investigated the concept of image hashing in frequency domain secured with the aid of chaotic sequences and we proposed an original verification system based on both hashing and watermarking concepts. We have seen that the hash value computed using feature points from jointed DWT and DCT transforms can be inserted as watermark into a digital image using a RST invariant watermarking scheme. The watermarking scheme was chosen deliberately to overpass the drawbacks of the proposed hashing system in our attempt to verify a digital image against image processing attacks.

Both, the proposed hashing scheme and the verification system may be applicable for image authentication and copyright protection for applications with certain purposes. In part of future research, we will concentrate on an alternative approach which is more robust against tampering attacks.

Acknowledgments

The author thanks Professor Adriana Vlad for the guidance concerning the use of the chaotic system from this work and for all technical advices and discussions about chaos based cryptography. The author also thanks Professor Ante Poljicak for providing the code of the watermarking method.

References

Bas, P., Filler, T., Pevny, T. (2011): Break our steganographic system – the ins and outs of organizing BOSS, *Proc. of Information Hiding Conference*, Prague.

Fridrich, J.; Goljan, M. (2000): Robust hash functions for digital watermarking, Proc. IEEE Int. Conf. Information Technology: Coding Computing, pp. 178-183.

Furon, T., Bas, P. (2008): Broken arrows, EURASIP Journal on Information Security.

Guo, X. X.; Dimitrios, H. (2007): Content based image via wavelet and radon transform, Proc. Of the 8th Pacific Rim Conference on Multimedia, Hongkong, China, vol. 4810, pp. 755-764.

Kim, B.-S.; Choi, J.-G.; Park, K.-H. (2004): RST-Resistant Image Watermarking Using Invariant Centroid and Reordered Fourier-Mellin Transform, Digital Watermarking Lecture Notes in Computer Science, Vol. 2939, pp. 370-381.

Kozat, S. S.; Mihcak, K.; Venkatesan, R. (2004): Robust perceptual image hashing via matrix invariances, Proc. IEEE Conf. on Image Processing, pp. 3443-3446.

Lv, X.; Wang, Z. J. (2008): Fast Johnson-Lindenstrauss Transform for Robust and Secure Image Hashing, Proc. of the IEEE 10th Workshopon Multimedia Signal Processing (MMSP), pp: 725-729.

Mihcak, M. K.; Venkatesan, R. (2005): New iterative geometric methods for robust perceptual image hashing, Proc. ACM Workshop Security and Privacy in Digital Rights Management, Philadelphia.

Monga, V.; Evans B. (2004): Robust perceptual image hashing using feature points, Proc. IEEE Int. Conf. Image Processing, Singapore, pp. 677-680.

Pereira, S.; Voloshynovskiy, S.; Madueno, M.; Marchand-Maillet, S.; Pun T. (2001): Second generation benchmarking and application oriented evaluation, Information Hiding Workshop, Pitsburgh, PA, USA.

Poljicak, A.; Mandic, L.; Agic, D. (2011): Discrete Fourier transform–based watermarking method with an optimal implementation radius, J. Electron. Imaging, vol. 20, no. 3.

Schaefer, G.; Stich, M. (2004): UCID – An uncompressed colour image database, Proc. SPIE, Storage and Retreval Methods and Applications for Multimedia, pp. 472-480, San Jose, U.K.

Solachidis, V.; Pitas, I. (2001): Circularly symmetric watermark embedding in 2-D DFT domain, IEEE Transactions on Image Processing, pp. 1741 – 1753.

Swaminathan, A.; Mao, Y.; Wu, M. (2004): Image hashing resilient to geometric and filtering operations, Proc. IEEE Workshop on Multimedia Signal Processing, Siena, Italy.

Tataru, R.-L. (2014): Image Hashing Secured With Chaotic Sequences, FedCSIS, pp. 735-740, Warsaw, Poland.

Vlad, A.; Luca, A.; Hodea O.; Tataru, R. (2013): Generating chaotic secure sequences using tent map and a running-key approach, Proc. of The Romanian Academy, Series A, vol. 14, pp. 292-302.

Weng, L.; Preneel, B. (2007): A secure perceptual hash algorithm for image content authentication, Proc. Of IEEE International Conference on Signal Processing and Communications.

Yu, F.-X.; Lei, Y.-Q; Wang, Y.-G. ; Lu, Z.-M. (2010): Robust image hashing based on invariance of DCT coefficients, JIH-MSP, vol.1, pp.286-291.