2

Perceptual Image Hashing

Azhar Hadmi[1], William Puech[1], Brahim Ait Es Said[2]
and Abdellah Ait Ouahman[2]
[1]*University of Montpellier II, CNRS UMR 5506-LIRMM*
[2]*University of Cadi Ayyad, ETRI Team*
[1]*France*
[2]*Morocco*

1. Introduction

With the fast advancement of computer, multimedia and network technologies, the amount of multimedia information that is conveyed, broadcast or browsed via digital devices has grown exponentially. Simultaneously, digital forgery and unauthorized use have reached a significant level that makes multimedia authentication and security very challenging and demanding. The ability to detect changes in multimedia data has been very important for many applications, especially for journalistic photography, medical or artwork image databases. This has spurred interest in developing more robust algorithms and techniques to allow to check safety of exchanged multimedia data confidentiality, authenticity and integrity. Confidentiality means that the exchange between encrypted multimedia data entities, which without decryption key, is unintelligible. Confidentiality is achieved mainly through encryption schemes, either secret key or public key. Authentication is an another crucial issue of multimedia data protection, it makes possible to trace the author of the multimedia data and allow to determine if an original multimedia data content was altered in any way from the time of its recording. Integrity allows degradation detection of multimedia and helps make sure that the received multimedia data has not been modified by a third party for malicious reasons. Many attempts have been noted to secure multimedia data from illegal use by different techniques fields such as encryption field, watermarking field and perceptual image hashing field. The field of encryption is becoming very important in the present era in which information security is of the utmost concern to provide end-to-end security. Multimedia data encryption has applications in internet communication, multimedia systems, medical imaging, telemedicine, military communication, etc. Although we may use the traditional cryptosystems to encrypt multimedia data directly, it is not a good idea for two reasons. The first reason is that the multimedia data size is almost always much great. Therefore, the traditional cryptosystems need much more time to directly encrypt the multimedia data. The other problem is that the decrypted multimedia data must be equal to the original multimedia data. However, this requirement is not necessary for image/video data. Due to the characteristic of human perception, a decrypted multimedia containing small distortion is usually acceptable. Deciding upon what level of security is needed is harder than it looks. To identify an optimal security level, the cost of the multimedia information to be protected and the cost of the protection itself are to be compared carefully. At present, many available image encryption algorithms have been proposed (Ozturk & Ibrahim, 2005;

Puech et al., 2007; Rodrigues et al., 2006). In some algorithms, the secret-key and algorithm cannot be separated effectively. This does not satisfy the requirements of the modern cryptographic mechanism and are prone to various attacks. In recent years, the image encryption has been developed to overcome above disadvantages as discussed in (Furht et al., 2004; Stinson, 2002). The other field to secure multimedia data is the watermarking field. Watermarking schemes have been developed for protecting intellectual property rights, which embed imperceptible signal, called watermark, carrying copyright information into a multimedia data *i.e.* image to form the watermarked image. The embedded watermark should be robust against malicious attacks so that it can be correctly extracted to show the ownership of the host multimedia data whenever necessary (Bender et al., 1996; Memon & Wong, 1998). A fragile or semi-fragile watermark detects changes of the host multimedia data such that it can provide some form of guarantee that the multimedia data has not been tampered with and is originated from the right source. In addition, a fragile watermarking scheme should be able to identify which portions of the watermarked multimedia data are authentic and which are corrupted; if unauthenticated portions are detected, it should be able to restore it (Cox et al., 2002). Watermarking has been widely adopted in many applications that require copyright protection, copy control, image authentication and broadcast monitoring (Cox et al., 2000). Watermarking can be used in copyright check or content authentication for individual images, but is not suitable when a large scale search is required. Furthermore, data embedding inevitably cause slight distortion to the host multimedia data (Wang & Zhang, 2007) and change its content. Recently, researchers in the field of security/authentication of multimedia data have introduced a technique inspired from the cryptographic hash functions to authenticate multimedia data called the *Perceptual hash functions* or *Perceptual image hashing* in case of image applications. It should be noted that the objective of a cryptographic hash function and a perceptual image hash function are not exactly the same. For example, there is no robustness or tamper localization requirement in case of a cryptographic hash function (Ahmed & Siyal, 2006). Traditionally, data integrity issues are addressed by cryptographic hashes or message authentication functions, such as MD5 (Rivest, 1992) and SHA series (NIST, 2008), which are sensitive to every bits of the input message. As a result, the message integrity can be validated when every bit of the message are unchanged (Menezes et al., 1996). This sensitivity to every bit is not suitable for multimedia data, since the information it carries is mostly retained even when the multimedia has undergone various content preserving operations. Therefore, bit-by-bit verification is no longer a suitable method for multimedia data authentication. A rough classification of content-preserving and content-changing manipulations is given in Table 1 (Han & Chu, 2010). Robust perceptual image hashing methods have recently been proposed as primitives to overcome the above problems and have constituted the core of a challenging developing research area to academia as well as the multimedia industry. Perceptual Image hashing functions extract certain features from image and calculate a hash value based on these features. Such functions have been proposed to establish the "perceptual equality" of image content. Image authentication is performed by comparing the hash values of the original image and the image to be authenticated. Perceptual hashes are expected to be able to survive on acceptable content-preserving manipulations and reject malicious manipulations. In recent years, there has been a growing body of research on perceptual image hashing that is increasingly receiving attention in the literature. Perceptual image hashing system generally consists of four pipeline stages: the *Transformation* stage, the *Feature extraction* stage, the *Quantization* stage and the *Compression and Encryption* stage as shown in Figure 1. The *Quantization* stage in a perceptual image hashing system is very

important to enhance robustness properties and increase randomness to minimize collision probabilities in a perceptual image hashing system. This step is very difficult especially if it is followed by the *Compression and Encryption* stage because we do not know the behavior of the extracted continuous features after content-preserving/content-changing manipulations (manipulations examples are given in Table 1). For this reason, in most proposed perceptual image hashing schemes, the *Compression and Encryption* stage is ignored.

Content-preserving manipulations	Content-changing manipulations
- Transmission errors	- Removing image objects
- Noise addition	- Moving of image elements or changing their positions
- Compression and quantization	- Adding new objects
- Resolution reduction	- Changes of image characteristics: color, textures, structure, etc.
- Scaling	- Changes of the image background: day time or location
- Rotation	- Changes of light conditions: shadow manipulations etc.
- Cropping	
- γ Distortion	
- Changes of brightness hue and saturation	
- Contrast adjustment	

Table 1. Content-preserving and content-changing manipulations.

In this chapter we analyze the importance of the *Quantization* stage problem in a perceptual image hashing pipeline. This chapter is arranged as follows. In Section 2, a classification of perceptual image hashing methods is presented followed by an overview of the unifying framework for perceptual image hashing. Then, the basic metrics and important requirements of a perceptual image hashing function wherein a formulation of the perceptual image hashing problem is given. Then, perceptual hash verification measures are presented followed by an overview of recent published schemes proposed in the literature. In Section 3, we present the quantization problem in perceptual image hashing systems, then we discuss the different quantization techniques used for more robustness of a perceptual image hashing scheme where we show their advantages and their limitations. In Section 4, a new approach of analysis of the quantization stage is presented based on the theoretical study presented in Section 3 and it is followed by a presentation and discussion of some obtained experimental results. Finally, Section 5 offers a discussion on the issues addressed and identifies future

research directions. The objective of the latter section is to present prospects and challenges in the context of perceptual image hashing.

2. Perceptual image hashing

In this Section, we give a classification of different perceptual image hashing techniques followed by the presentation of perceptual image hashing framework and basic requirements related to perceptual image hashing are discussed. Furthermore, related work is reviewed and the challenging problems that are not yet resolved are identified.

2.1 Perceptual image hashing methods classification

Most of the existing image hashing studies mainly focus on the feature extraction stage and use them during authentication, which can roughly be classified into the four following categories (Zhu et al., 2010), (Han & Chu, 2010):

- *Statistic-based schemes* (Khelifi & Jiang, 2010; Schneider & Chang, 1996; Venkatesan et al., 2000): This group of schemes extracts hash features by calculating the images statistics in the spacial domain, such as mean, variance, higher moments of image blocks and histogram.
- *Relation-based schemes* (Lin & Chang, 2001; Lu & Liao, 2003): This category of approaches extracts hash features by making use of some invariant relationships of the coefficients of discrete cosine transform (DCT) or wavelet transform (DWT).
- *Coarse-representation-based schemes* (Fridrich & Goljan, 2000; Kozat et al., 2004; Mihçak & R.Venkatesan, 2001; Swaminathan et al., 2006): In this category of methods, the perceptual hashes are calculated by making use of coarse information of the whole image, such as the spatial distribution of significant wavelet coefficients, the low-frequency coefficients of Fourier transform, and so on.
- *Low level feature-based schemes* (Bhattacharjee & Kutter, 1998; Monga & Evans, 2006): The hashes are extracted by detecting the salient image feature points. These methods first perform the DCT or DWT transform on the original image, and then directly make use of the coefficients to generate final hash values. However, these hash values are very sensitive to global as well as local distortions that do not cause perceptually significant changes to the images.

2.2 Perceptual image hashing framework

A perceptual image hashing system, as shown in Fig. 1, generally consists of four pipeline stages: the *Transformation* stage, the *Feature extraction* stage, the *Quantization* stage and the *Compression and Encryption* stage.

In the *Transformation* stage, the input image undergoes spacial and/or frequency transformation to make all extracted features depend the the values of image pixels or the image frequency coefficients. In the *Feature Extraction* stage, the perceptual image hashing system extracts the image features from the input image to generate the continuous hash vector. Then, the continuous perceptual hash vector is quantized into the discrete hash vector in the *Quantization* stage. The third stage converts the discrete hash vector into the binary perceptual hash string. Finally, the binary perceptual hash string is compressed and encrypted into a short and a final perceptual hash in the *Compression and Encryption* stage (Figure 1).

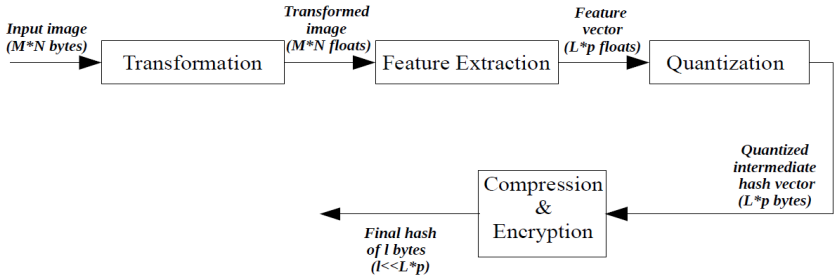


Fig. 1. Four pipeline stages of a perceptual image hashing system.

2.2.1 Transformation stage

In the *Transformation* stage, the input image of size $M \times N$ *bytes* undergoes spatial transformations such as color transformation, smoothing, affine transformations, etc. or frequency transformations such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), etc. When the DWT transformation is applied, most perceptual image hashing schemes take into account just the LL subband because it is a coarse version of the original image and contains all the perceptually information. The principal aim of those transformations is to make all extracted features, in the *Feature Extraction* stage, depend upon the values of image pixels or its frequency coefficients in the frequency space.

2.2.2 Feature Extraction stage

In the *Feature Extraction* stage, the image hashing system extracts the image features from the transformed image to generate the feature vector of L features where $L << M \times N$. Note that each feature can contain p elements of type $float$ which means that we get $L \times p$ $floats$ at this stage. It is still an open question, however, which mappings (if any) from DCT/DWT coefficients preserve the essential information about an image for hashing and/or mark embedding applications. We can at this stage add another features selection as shown in Fig. 2, so only the most pertinent features are selected which are statistically more resistant against a specific allowed manipulation like addition of noise and image rotation, etc. The selected features can be presented as an intermediate hash vector of $K \times p$ $floats$, where $K < L$.

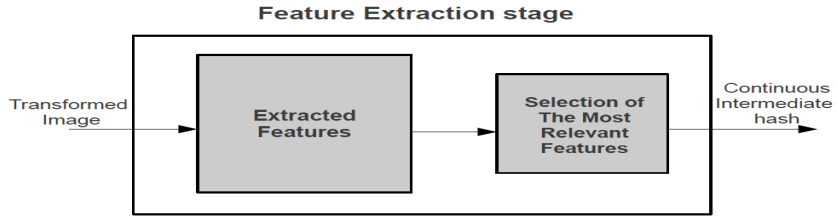


Fig. 2. Selection of the most relevant features in the Feature Extraction stage.

2.2.3 Quantization stage

In the next stage, the *Quantization* stage, we get a quantized intermediate perceptual hash vector which contains $L \times p$ elements of type *byte*. Uniform quantization can be applied to quantize each component of the continuous perceptual hash vector. Adaptive

quantization (Mihçak & R.Venkatesan, 2001) is another quantization type which is is the most famous quantization scheme in the field of image hashing. The difference between the two quantization schemes is that the partition of uniform quantization is based on the interval length of the hash values, whereas the partition of adaptive quantization is based on the probability density function (pdf) of the hash values. This kind of quantization is detailed in Section 3.

2.2.4 Compression and Encryption stage

Compression and Encryption stage is the final step of a perceptual image hashing system, the binary intermediate perceptual hash string is compressed and encrypted into a short perceptual hash of fixed size of l *bytes*, where $l << L \times p$, which presents the final perceptual hash that allows image verification and authentication at the receiver. This stage can be ensured by cryptographic hash functions i.e. SHA series which generate the final hash of fixed size (hash of 160 bits in case SHA-1).

In the next section, we give the most important requirements that a perceptual image hashing must achieve and show how they conflict with each other.

2.3 Metrics and important requirements of a perceptual image hashing

Perceptual hash functions can be categorized into two categories: unkeyed perceptual hash functions and keyed perceptual hash functions. An unkeyed perceptual hash function $H(x)$ generates a hash value h from an arbitrary input x (that is $h = H(x)$). A keyed perceptual hash function generates a hash value h from an arbitrary input x and a secret key k (that is $h = H(x;k)$). The design of efficient robust perceptual image hashing techniques is a very challenging problem that should address the compromise between various conflicting requirements. Let P denote probability. Let $H()$ denote a perceptual hash function which takes one image as input and produces a binary string of length l. Let I denote a particular image and I_{ident} denote a modified version of this image which is "perceptually similar" to I. Let I_{diff} denote an image that is "perceptually different" from I. Let h_1 and h_2 denote hash values of the original image I and the perceptually different image I_{diff} from I. $\{0/1\}^l$ represents binary strings of length l. Then the four desirable properties of a perceptual image hashing function are identified as follows:

- Equal distribution (unpredictability) of hash values:

$$P(H(I) = h_1) \approx \frac{1}{2^l}, \forall h_1 \in \{0,1\}^l \tag{1}$$

- Pairwise independence for perceptually different images I and I_{diff}:

$$P(H(I) = h_1 | H(I_{diff}) = h_2) \approx P(H(I_{ident}) = h_1), \qquad \forall h_1, h_2 \in \{0,1\}^l \tag{2}$$

- Invariance for perceptually similar images I and I_{ident}:

$$P(H(I) = H(I_{ident})) \geq 1 - \theta_1, \qquad for\ a\ given\ \theta_1 \approx 0 \tag{3}$$

- Distinction of perceptually different images I and I_{diff}:

$$P(H(I) \neq H(I_{diff})) \geq 1 - \theta_2, \qquad for\ a\ given\ \theta_2 \approx 0 \tag{4}$$

To meet property in equation (3), most perceptual hash functions try to extract features of images which are invariant under insignificant global modifications such as compression or enhancement. Equation (4) means that, given an image I, it should be nearly impossible for an adversary to construct a perceptually different image I_{diff} such that $H(I) = H(I_{diff})$. This property can be hard to achieve because the features used by published perceptual hash functions are publicly known (Kerckhoffs, 1883; Mihçak & R.Venkatesan, 2001). Also, it makes property in equation (3) be neglected in favor of property in equation (4). Likewise for perfect unpredictability, an equal distribution (equation (1)) of the hash values is needed. This would deter achieving the property in equation (3) (Monga, 2005). Depending on the application, perceptual hash functions have to achieve these conflicting properties to some extent and/or facilitate trade-offs. From a practical point of view, both robustness and security are important. Lack of robustness (equation (3)) renders an image hash useless as explained above, while security (equations (1),(4)) means that it is extremely difficult for an adversary to modify the essential content of an image yet keep the hash value unchanged. Thus, trade-offs must be sought, and this usually forms the central issue of perceptual image hashing research.

2.4 Perceptual hash verification

Perceptual image hashing system calculates hashes for similar images that must be equal. Referring to the image space as shown in Figure 3, let I denote an image, and X denote the set of images I_{ident} that are modified from I by means of content-preserving manipulations and are defined to be perceptually similar to I. Let Y contains all other images I_{diff} that are irrelevant to I and its perceptually similar versions. I_{diff} are the results of content-changing manipulations. Consequently, $\{I\} \cup X \cup Y$ forms an entire image space. Let h, h_{ident} and h_{diff} denote hash values of the original image I, the perceptually similar image I_{ident} from I and the perceptually different image I_{diff} from I respectively. In robust and secure perceptual image, the following properties are required when Encryption and Compression stage is applied in a perceptual image hashing system: $\boldsymbol{h = h_{ident}}$ **for all identical images** $\boldsymbol{I_{ident} \in X}$ and $\boldsymbol{h \neq h_{ident}}$ **for all different images** $\boldsymbol{I_{diff} \in Y}$ (Figure 3). Since the requirement of bit-by-bit hashes equality is usually hard to achieve, most of the proposed schemes compute distances and similarities between perceptual hashes. The most often used are the Bit Error Rate (BER), the Hamming distance and the Peak of Cross Correlation (PCC). The first two measure the distance between two hash values, whereas the latter measures the similarity between two hash values. Using theses measures, the sender determines the threshold τ. The proper selection of τ is very important as it defines the boundary between content-preserving and content-changing manipulations.

Let $d(.,.)$ indicates the used measure *i.e.* a normalized Hamming distance function. Let h, h_{ident} and h_{diff} denote hash values of the original image I, the perceptually similar image I_{ident} from I and the perceptually different image I_{diff} from I respectively. The error-resilience of multimedia data hashing is defined as follows. I_{ident} is successfully identified to be perceptually similar to I if $d(h, h_{ident}) \leqslant \tau$ holds. In other words if two images are perceptually similar, their corresponding hashes need to be highly correlated. If $d(h, h_{diff}) \gg \tau$, then I_{diff} is identified as modified from I by means of content-changing manipulations. Overall, the main theme of perceptual image hashing is to develop a robust perceptual image hash function that can identify perceptually similar multimedia contents and reject content-changing manipulations.

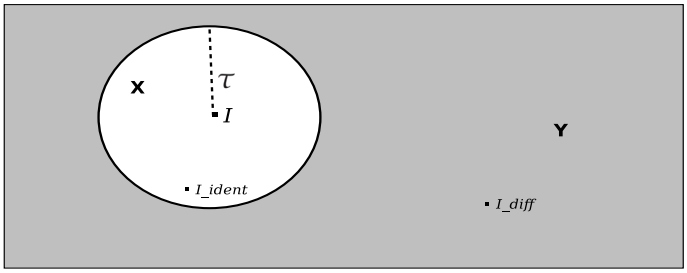


Fig. 3. The image space $\{I\} \cup X \cup Y$ formed by an image $\{I\}$, its perceptually similar versions set X and its modified version set Y.

2.5 Review of some related work on perceptual image hashing techniques

In recent years, there has been a growing body of research on perceptual image hashing that is increasingly receiving attention in the literature. Most of these existing papers focus on studies of the feature extraction stage because they believe that extracting a set of robust features that resist, and to stay relatively constant, content-preserving manipulations and at the same time should detect content-changing manipulations is the most important objective in perceptual image hashing system. Few papers address perceptual image hashing system security. In (Fridrich, 2000), the extraction of the hash is based on the projection of image coefficients onto filtered pseudo-random patterns. The final perceptual hash is used for generating a pseudo-random watermark sequences, that depend sensitively on a secret key yet continuously on the image, for authentication and integrity verification of still images. In (Venkatesan et al., 2000), a perceptual image hashing technique based on statistics computed from randomized rectangles in the discrete wavelet domain (DWT) is presented. Averages or variances of the rectangles are then calculated and quantized with randomized rounding to obtain the hash in the form of a binary string. The quantized statistics are then sent to an error-correcting decoder to generate the final hash value. Statistical properties of wavelet subbands are generally robust against attacks, but they are only loosely related to the image contents therefore rather insensitive to tampering. This method has been shown to be robust against common image manipulations and geometric attacks. The proposed method in (Schneider & Chang, 1996) is using the intensity histogram to sign the image. Since the global histogram does not contain any spatial information, the authors divide the image into blocks, which can have variable sizes, and compute the intensity histogram for each block separately. This allows some spatial information to be incorporated into the signature. The method in (Fridrich & Goljan, 2000) is based on the observation of the low frequency DCT coefficient. If a low frequency DCT coefficient of an image is small in absolute value, it cannot be made large without causing visible changes to the image. Similarly, if the absolute value of a low frequency coefficient is large, it cannot change it to a small value without influencing the image significantly. To make the procedure dependent on a key, the DCT modes are replaced with DC-free random smooth patterns generated from a secret key. Other researchers have used others techniques to perform image perceptual hashing. Authors in (Swaminathan et al., 2006) used Fourier-Mellin transform for perceptual image hashing applications. Using Fourier-Mellin transform's scale invariant property, the magnitudes of the Fourier transform coefficients were randomly weighted and summed. However, since Fourier transform did not offer localized frequency information, this method was not able to detect malicious local modifications. In a more recent development, a perceptual image hashing

scheme based Radon Transform is proposed in (Lei et al., 2011) where the authors perform Radon Transform on the image and calculate the moment features which are invariant to translation and scaling in the projection space. Then Discrete Fourier Transform (DFT) is applied on the moment features to resist rotation. Finally, the magnitude of the significant DFT coefficients is normalized and quantized as the final perceptual image hash. The proposed method can tolerate almost all the typical image processing manipulations, including JPEG compression, geometric distortion, blur, addition of noise and enhancement. The Radon transform was first used in (Lefebvre et al., 2002), and further expanded in (Seo et al., 2004). Authors in (Guo & Hatzinakos, 2007) propose a perceptual image hashing scheme based on the combination of discrete wavelet transform (DWT) and the Radon Transform. Taking the advantages of the frequency localization property of DWT and shift/rotation invariant property of the Radon transform, the algorithm can effectively detect malicious local changes, and at the same time, be robust against content-preserving modifications. Obtained features derived from the Radon Transform are then quantized by the probabilistic quantization (Mihçak & Venkatesan, 2001) to form the final perceptual hash.

In this Section, we have presented some reviews of different schemes proposed in the field of perceptual image hashing. In Section 3, we develop the quantization problem in perceptual image hashing and we present some approaches to address this problem which surely have limitations in practice.

3. Quantization problem in perceptual image hashing

3.1 Problem statement

The goal of the quantization stage, in the perceptual image hashing system, is to discretize the continuous intermediate hash vector (continuous features) into a discrete intermediate hash vector (discrete features). This step is very important to enhance robustness properties and increase randomness to minimize collision probabilities of a perceptual image hashing system. Quantization is the conventional way to achieve this goal. The quantization step is difficult because we do not know how the values in the continuous intermediate hash drop after content-preserving (non-malicious) manipulations in each quantization interval Q. This difficulty of an efficient quantization increases more when it is followed by an encryption and compression stage *i.e.* SHA-1, because the discrete intermediate hash vectors must be quantized in a correct way for all perceptual similar images. For this reason this stage is ignored in most schemes presented in the literature. To understand the quantization problem statement, let us suppose that the incidental distortion introduced by content-preserving manipulations can be modeled as noise whose maximum absolute magnitude is denoted as B, which means that the maximum range of additive noise is B. Suppose that the original scalar value $x_l \in \mathbb{R}$ for $l \in \{1, ..., L\}$ of the continuous intermediate hash is bounded to a finite interval $[-A, A]$. Furthermore, suppose that we wish to obtain a quantized message $q(x_l)$ of x_l in P quantization points given by the set $\tau = \{\tau_1, ..., \tau_P\}$. The points are uniformly spaced such that $Q = \tau_j - \tau_{j-1} = 2A/(P-1)$ for $j \in \{1, ..., P\}$. Now suppose $x_l \in [\tau_j, \tau_{j+1})$, then it will be quantized as τ_j. However, when this value is corrupted after noise addition, the distorted value could drop in the previous quantization interval $[\tau_{j-1}, \tau_j)$ or in the next interval $[\tau_{j+1}, \tau_{j+2})$ and it will be quantized as τ_{j-1} or τ_{j+1}, respectively, and the quantized x_l value will not remain unchanged as τ_j before and after noise addition. Thus, the noise corruption will cause a different quantization result and automatically cause different perceptual hashes (Hadmi et al., 2010). Figure 4 shows the distribution of the original DWT

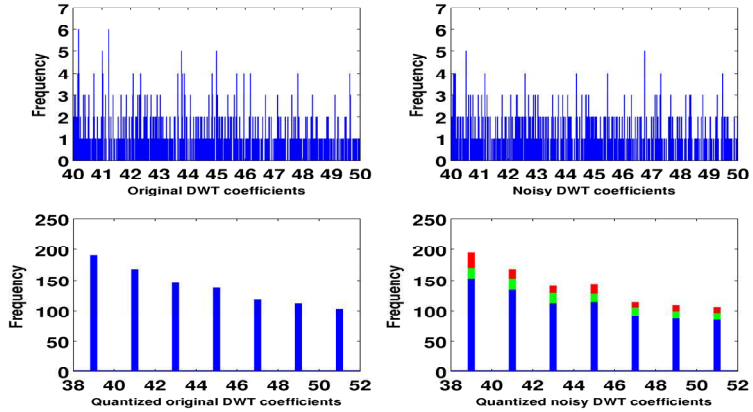


Fig. 4. The influence of additive Gaussian noise on the quantization ($Q = 2$) of the original DWT LL-subband coefficients and their noisy version in the interval $[40, 50]$. In green: DWT LL-subband quantized coefficients that dropped from the right neighboring quantization interval. In red: DWT LL-subband quantized coefficients that dropped from the left neighboring quantization interval.

LL-subband (level 3) coefficients, of Lena image sized 1024×1024, in the interval $[40, 50]$ and their noisy version, in the same interval $[40, 50]$, by an additive Gaussian noise of standard deviation equals $\sigma = 1$. When applying a Gaussian noise with $\sigma = 1$, the noisy image remains visually the same than the original image however it causes changes on extracted features distribution as we can see in Figure 4. This causes errors in the quantization step because the quantized features do not remain unchanged after noise addition as shown in Figure 4. To avoid such cases, many quantization schemes have been proposed in the literature. Authors in (Sun & Chang, 2005) proposes an error correction coding (ECC) to correct errors of extracted features caused by corruption from additive noise to get the same quantization result before and after additive noise. In their work, they assume that the quantization step $Q > 4B$, which is not always true at the practical point of view, and they push the points away from the quantization decision boundaries and create a margin of at least $Q/4$ so that original x_l value when later contaminated will not exceed the quantization decision boundaries. The illustration of the concept of error correction is illustrated in Figure 5. The original feature P is quantized in nQ before adding noise, but after adding noise there is also a possibility that the noisy feature value could drop at the range $[(n-1)Q, (n-0.5)Q)[$ and will quantized as $(n-1)Q$. As a solution to this, Authors propose to add or subtract $0.25Q$ to remain the features at the range $[(n-0.5)Q, (n+0.5)Q)]$ and then remain the quantized value the same as the original quantized value nQ even after adding noise.

Other similar work based on this approach has recently been proposed (Ahmed et al., 2010) where the authors calculate and record a vector of 4-bits called "Perturbation information". This additional transmitted information has the same dimension of the extracted features. It is used at the receiver's end to adjust the intermediate hash during the image verification stage before performing quantization. Therefore, the information carried in the "Perturbation information" helps to make a decision to positively authenticate an image or not. Their theoretical analysis is more general than in (Sun & Chang, 2005) from a practical point of view. One main disadvantage of such schemes is that vectors used to correct errors of extracted

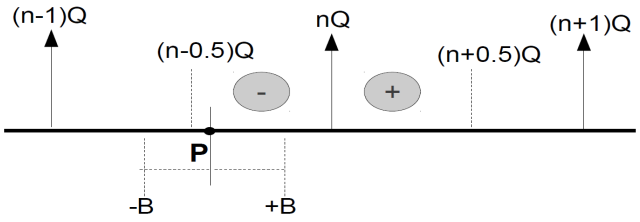


Fig. 5. Illustration on the concept of error correction in Sun's scheme (Sun & Chang, 2005).

features need to be transmitted or stored beside the image and the final hash as shown in Figures 6 and 7.

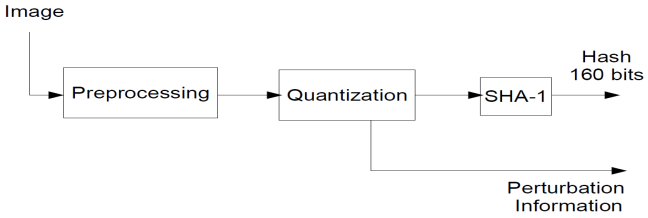


Fig. 6. Hash generation module with quantization in Fawad's scheme (Ahmed et al., 2010).

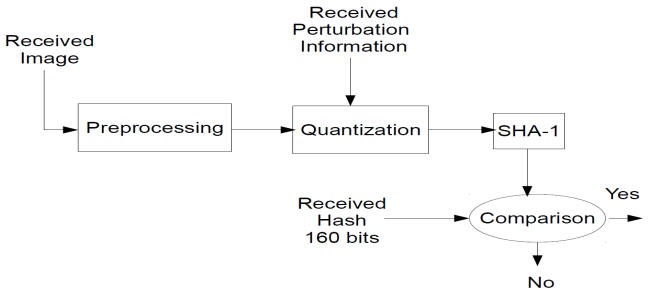


Fig. 7. Image verification module with quantization in Fawad's scheme (Ahmed et al., 2010).

Another quantization scheme which is widely applied in perceptual image hashing (Swaminathan et al., 2006), (Zhu et al., 2010) proposed by (Mihçak & Venkatesan, 2001) called *Adaptive Quantization* or *Probabilistic Quantization* in (Monga, 2005). Its property is that it takes into account to the distribution of the input data. The quantization intervals $Q = \tau_j - \tau_{j-1}$ for $j \in \{1, ..., P\}$ are designed so that $\int_{\tau_{j-1}}^{\tau_j} p_X(x)\,dx = 1/P$, where P is the number of quantization levels and $p_X(.)$ is the pdf of the input data X. The central points $\{C_j\}$ are defined so as to make $\int_{\tau_{j-1}}^{C_j} p_X(x)\,dx = \int_{C_j}^{\tau_j} p_X(x)\,dx = 1/(2P)$. Around each τ_j, a randomization interval $[A_j, B_j]$ is introduced such that $\int_{A_j}^{\tau_j} p_X(x)\,dx = \int_{\tau_j}^{B_j} p_X(x)\,dx = r/P$, where $r \leq 1/2$. The randomization interval is symmetric around τ_j for all j in terms of distribution p_X. The natural constraint must be respected $C_j \leq A_j$ and $B_j \leq C_{j+1}$. The overall quantization rule is then

given by:

$$q(x_l) = \begin{cases} j-1 & \text{w.p.} \quad 1 & \text{if } C_j \leq x_l < A_j, \\ j-1 & \text{w.p.} \quad \left(\frac{P}{2r}\int_{x_l}^{B_j} p_X(t)\,dt\right) & \text{if } A_j \leq x_l < B_j, \\ j & \text{w.p.} \quad \left(\frac{P}{2r}\int_{A_j}^{x_l} p_X(t)\,dt\right) & \text{if } A_j \leq x_l < B_j, \\ j & \text{w.p.} \quad 1 & \text{if } B_j \leq x_l < C_{j+1}. \end{cases} \tag{5}$$

where w.p. stands for "with probability".

The discrete scheme of *Adaptive Quantization* has recently been developed by (Zhu et al., 2010) to make it applicable in practice.

3.2 Theoretical analysis

In this section, we analyze statically the behavior of the extracted features under additive uniform noise, Section 3.2.1 and Gaussian noise, Section 3.2.2, as well as the probability of a false quantization for these selected features. The main goal of this analysis is to give a theoretical behavior of the extracted image features to be hashed against content-preserving /content-changing manipulations, that are simulated by an additive noise, that may undergo an image (Hadmi et al., 2011).

3.2.1 Case of an additive uniform noise

To analyze the influence of an additive noise on perceptual image hashing robustness, we have decided to lead a statical analysis of the quantization problem. The idea is to compute the length of the quantization interval Q for a noise whose maximum absolute magnitude is B, which represents the content-preserving manipulations, and a previously fixed probability that a value in this interval drops out, that is denoted as P_{drop}.

To address this problem, we have started by developing the convolution product between two distributions defined as follows:

- Let $P_\rho(x)$ denote the extracted feature distribution limited to an interval $[a, b]$ of length $\rho = b - a$. $P_\rho(x)$ is given by:

$$P_\rho(x) = \begin{cases} \frac{1}{\rho} & \text{for } x \in [a, b], \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

- Let $P_B(x)$ denote the probability density function of the continuous uniform noise, which presents content-preserving manipulations, in the interval $B = [-\frac{B}{2}, \frac{B}{2}]$, with $B < \rho$. $P_B(x)$ is expressed as:

$$P_B(x) = \begin{cases} \frac{1}{B} & \text{for } x \in [-\frac{B}{2}, \frac{B}{2}], \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The convolution product $h(x)$ of $P_\rho(x)$ by $P_B(x)$ is:

$$h(x) = \int_{-\infty}^{+\infty} P_\rho(y) P_B(x-y)\, dy = \int_a^b \frac{1}{\rho} P_B(x-y)\, dy \tag{8}$$

Finally, we get the convolution product $h(x)$ (equation (9)) expressed as:

$$h(x) = \begin{cases} 0 & \text{for } x \leq a - \frac{B}{2}, \\ \frac{1}{\rho B}\left(x + \frac{B}{2} - a\right) & \text{for } x \in \left]a - \frac{B}{2}, a + \frac{B}{2}\right], \\ \frac{1}{\rho} & \text{for } x \in \left]a + \frac{B}{2}, b - \frac{B}{2}\right], \\ \frac{1}{\rho B}\left(-x + \frac{B}{2} + b\right) & \text{for } x \in \left]b - \frac{B}{2}, b + \frac{B}{2}\right], \\ 0 & \text{for } x > b + \frac{B}{2}. \end{cases} \tag{9}$$

An example of $h(x)$ is presented in Figure 8, with $B < \frac{\rho}{2}$.

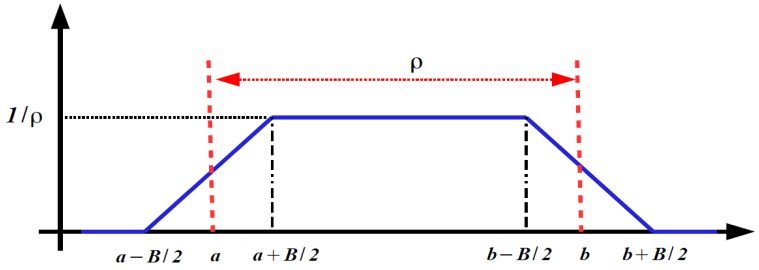


Fig. 8. Convolution product of $P_\rho(x)$ by $P_B(x)$.

Suppose that y presents an extracted feature which is in the interval $[a, b]$ and let P_{drop} be the probability that y drops out from $[a, b]$ because of the adding noise B. Thus, $P_{drop}(y)$ is calculated and expressed as follows (Equation 10):

$$\begin{aligned} P_{drop}(y) &= P(y \notin [a,b]) \\ &= \int_{a-\frac{B}{2}}^{a} h(x)\, dx + \int_{b}^{\pm\frac{B}{2}} h(x)\, dx \\ &= \frac{B}{4\rho} \end{aligned} \tag{10}$$

Equation (10) allows us to get an information of the extracted features behavior after adding noise. For example, for a uniform noise of length $B = 4.10^{-2}$, if we want to have $P_{drop} = 10^{-3}$, then the length of the quantization interval ρ that must be chosen is: $\rho = 10$.

To make a comparison between the theoretical probability that extracted features drop out from the quantization interval given by Equation 10 and the experimental probability, we

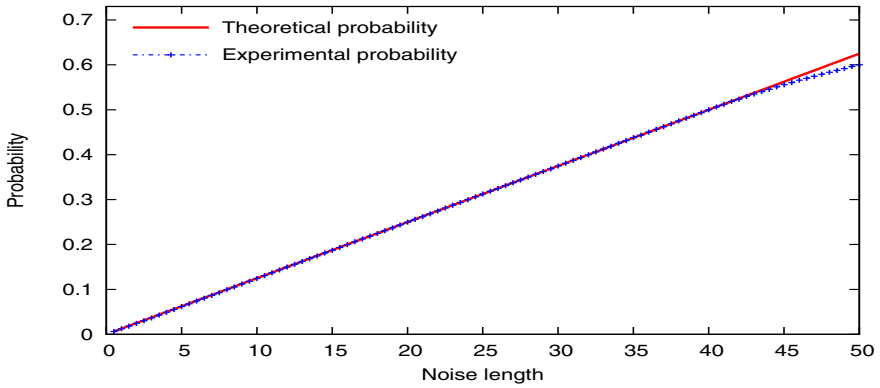


Fig. 9. Comparison between the theoretical and the experimental probabilities that extracted features drop out from the quantization interval for various noise lengths.

applied continuous uniform noise of different lengths from $B = 0$ to $B = 50$ on the same $N = 10000$ samples in the interval $\Delta = [-10, 10]$, and then we calculated the probability P_{drop} for each noise length. We note that the experimental results presented in Figure 9 coincide with the theoretical results calculated from Equation 10 for all noise lengths until $B = 44$. Some divergences are observed after this noise length which can be considered as content-changing (malicious) manipulations.

The same analysis can be performed for other noise distributions such as Gaussian distribution or triangular distribution. Thus, by just modeling the content-preserving manipulations by the aforementioned distributions, we can precisely obtain the probability from which the extracted features will drop from a fixed quantization interval to its neighboring intervals. Alternately, we can beforehand fix the maximum range of additive noise that we judge to be a content-preserving manipulation and the probability that extracted features change of quantization interval. This will allow us to fix the length of the appropriate quantization interval which respects to this probability.

3.2.2 Case of an additive Gaussian noise

Figure 10 shows an example of an original image of size 512×512 and their noisy versions with many levels of additive Gaussian noise controlled by its standard deviation σ. Note that the applied additive Gaussian noise is 0-mean, and changing its standard deviation σ allows us to increase or decrease its level.

To evaluate the perceptual similarity between the original and their modified versions, we can based on the perceptual aspect provided by the Human Visual System (HVS), on the method of the Structural SIMilarity (SSIM)[1] (Wang et al., 2004), or on the method of Peak Signal to Noise Ratio (PSNR). Table 2 gives the SSIM and PSNR values for noisy images obtained by applying different standard deviation values σ of the additive Gaussian noise. The quality of the Gaussian noisy images is compared to the original image and they are classified into four

[1] SSIM is a classical measure well correlated to the Human Visual System.The SSIM values are real positive numbers lower or equal to 1. Stronger is the degradation and lower is the SSIM measure. A SSIM value of 1 means that the image is not degraded.

(a) Original image (b) σ=1 (c) σ=5

(d) σ=10 (e) σ=11 (f) σ=14

(g) σ=15 (h) σ=20 (i) σ=25

(j) σ=30 (k) σ=35 (l) σ=40

Fig. 10. Original image and their noisy versions with different additive Gaussian noise parametrized with different standard deviations σ.

categories: very similar, similar, different and very different. The changed images qualified very similar and similar (Figures 10(b), 10(c), 10(d), 10(e)) must have the same perceptual hash of the original image noted by I_{ident}. Other cases of images *i.e.* images qualified as different or very different Figures (10(f), 10(g), 10(h), 10(i), 10(j), 10(k), 10(l)) from the original image must have a different perceptual hash noted by I_{diff} as presented in Table 2.

Standard deviation σ	SSIM	PSNR (dB)	Image quality	Perceptual hash
1	0.997	47.79	very Similar	I_{ident}
5	0.946	34.15	Similar	I_{ident}
10	0.828	28.16	Similar	I_{ident}
11	0.802	27.32	Similar	I_{ident}
14	0.728	25.25	Different	I_{diff}
15	0.704	24.70	Different	I_{diff}
20	0.600	22.24	Different	I_{diff}
25	0.517	20.36	Different	I_{diff}
30	0.450	18.86	very Different	I_{diff}
35	0.397	17.59	very Different	I_{diff}
40	0.354	16.50	very Different	I_{diff}

Table 2. SSIM and PSNR values for noisy images obtained by applying different standard deviation values σ of the additive Gaussian noise.

In the case of σ=1, the noisy image remains visually the same as the original image and it has high values of SSIM ($SSIM = 0.997$) and PSNR ($PSNR = 47.79$). For σ=5, σ=10 and σ=11, the changes in the noisy images are very small and we can consider that the noisy images are still similar to the original image. In the case of $\sigma = 5, 10, 11$, the SSIM values remain smaller than 80% and the PSNR values remain larger than $27db$. When the level of the additive Gaussian noise increases, the noisy images are perceptually different from the original image as it is shown in Figure 10 for σ=14,...,40 and both the SSIM and PSNR values degrade. We can fix the threshold of the additive Gaussian noise that holds a good content in the sense of human perception fixed at σ=11 as it is justified in term of the SSIM and PSNR values. We fixed the degradation to a SSIM value of 80% and the PSNR value at $27db$ to consider a noisy image similar to the original image. The threshold of the SSIM and PSNR values is justified in terms of the subjective measure based on the HVS for many tests that we have done for a large database of grayscale images as we can see in Figure 10.

To address theoretically the influence of an additive Gaussian noise whose 0-mean and standard deviation σ on an uniform distribution of features limited in an interval $[a, b]$, we compute the convolution product between the distribution of the extracted features and the distribution of the additive Gaussian noise defined as follows:

- Let $P_\rho(x)$ denote the extracted feature distribution limited to an interval $[a, b]$ of length $\rho = b$ - a. $P_\rho(x)$ is given by:

$$P_\rho(x) = \begin{cases} \frac{1}{\rho} & \text{for } x \in [a, b], \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

- Let $P_\sigma(x)$ denote the probability density function of the Gaussian noise whose 0-mean and standard deviation σ, which presents content-preserving manipulations. $P_\sigma(x)$ is expressed as:

$$P_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{12}$$

The convolution product $h(x)$ of $P_\rho(x)$ by $P_\sigma(x)$ is:

$$\begin{aligned} h(x) &= \int_{-\infty}^{+\infty} P_\rho(y) P_\sigma(x-y)\, dy \\ &= \frac{1}{\rho}\Big(\int_{-\infty}^{x-a} P_\sigma(y)\, dy - \int_{-\infty}^{x-b} P_\sigma(y)\, dy \Big) \\ &= \frac{1}{\rho}\Big(\int_{-\infty}^{x-a} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}\, dy - \int_{-\infty}^{x-b} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}\, dy \Big) \\ &= \frac{1}{\rho}\Big(\int_{-\infty}^{\frac{x-a}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, dy - \int_{-\infty}^{\frac{x-b}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, dy \Big) \\ &= \frac{1}{2\rho}\Big[erf\Big(\frac{x-a}{\sqrt{2}\sigma}\Big) - erf\Big(\frac{x-b}{\sqrt{2}\sigma}\Big) \Big] \end{aligned} \tag{13}$$

with $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$.

The convolution product $h(x)$ models the behavior of the original features after adding the Gaussian noise in each quantization interval. Figure 11 shows a normalized uniform distribution of 10000 features belonging in the interval $[10, 20]$ before and after the quantization stage where the quantization step Q=10. All these features are quantized to the value 15 as shown in Figure 11. Figure 12 presents the normalized distribution of the noisy features after adding a Gaussian noise with 0-mean and standard deviation σ=2. This distribution coincides exactly with the theoretical results given by Equation 13. As shown in Figure 12, the noisy features are quantized and spread in 3 quantization intervals and are quantized to three values: 5, 15 and 25. The 5 quantized value presents the quantized value to the left neighbor quantization interval and the 25 presents the quantized value to the right neighbor quantization interval. Statistically, for the same experiment settings we have 8% of features drop to the left neighbor quantization interval and 8% of features drop to the right neighbor quantization interval. For the other experiments settings, we always have a symmetric percentage of features drop in the left and right neighbor quantization interval.

4. Experimental results

4.1 Experimental analysis protocol

In this section, we describe the quantization analysis protocol for perceptual image hashing based on statistical invariance of extracted block mean features. The aim is to find agreement between the density of the additive Gaussian noise, the size of the image block and the quantization step size that must be taken to ensure a good level of image hashing robustness. As shown in Figure 13, the original input image I of size $N \times M$ pixels is split to non

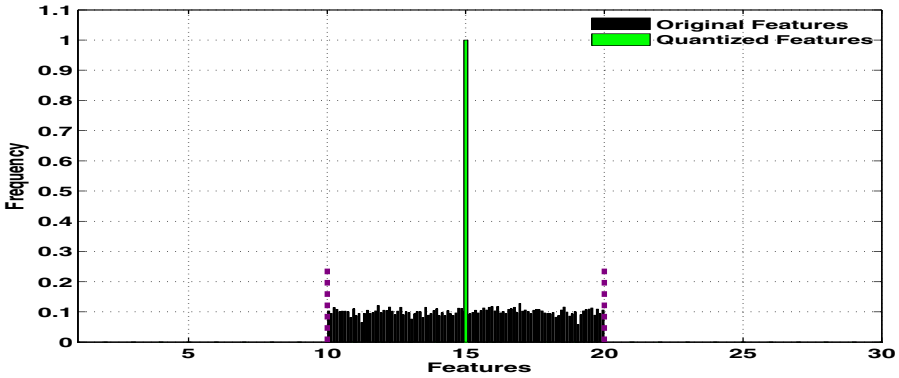


Fig. 11. 10000 original features uniformly distributed in one quantization interval $[10, 20]$ before quantization (black) and after uniform quantization (green) where the quantization step Q=10.

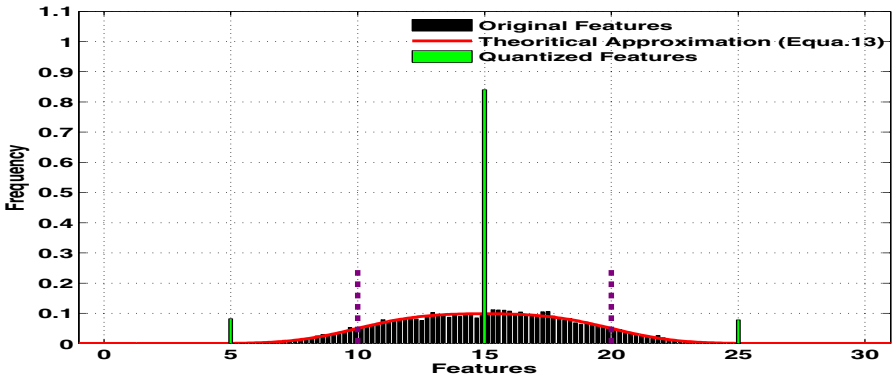


Fig. 12. 10000 noisy features after adding Gaussian noise whose 0-mean and standard deviation $\sigma = 2$ before quantization (black) and after uniform quantization (green) where the quantization step Q=10.

overlapping blocks of size $q \times p$ pixels that we note by $B_{i,j}$, where $i \in \{1, 2, \ldots, \frac{N}{q}\}$ and $j \in \{1, 2, \ldots, \frac{M}{p}\}$. The float mean value $m_{i,j}$ of each block $B_{i,j}$ is computed and stored in a one dimensional vector that we note by $\boldsymbol{V_m(k)}$, where $k \in \{1, 2, \ldots, \frac{N}{q} \times \frac{M}{q}\}$. Quantization step is the conventional way to descretize the continuous vector V_m. For a given quantization size step Q, the quantized vector $\boldsymbol{V'_m(k)}$ of $\boldsymbol{V_m(k)}$ is given by the floor operation:

$$V'_m(k) = \lfloor \frac{V_m(k)}{Q} \rfloor \times Q + \frac{Q}{2} \tag{14}$$

where k=$\{1, 2, \ldots, \frac{N}{q} \times \frac{M}{q}\}$.

The distribution $Dist_I$ of the quantized vector $\boldsymbol{V'_m}$ is then calculated and stored as a reference enabling us to make a comparison with distributions of other candidate images for verification of their integrity with the original image.

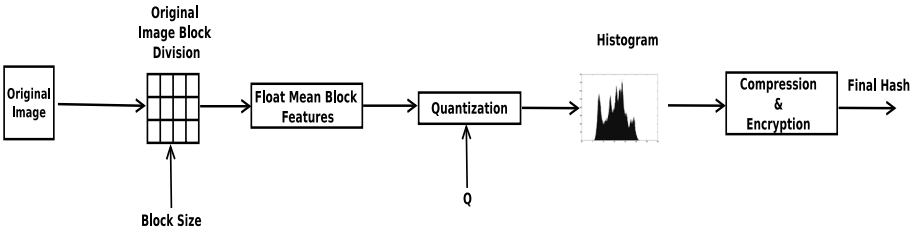


Fig. 13. Proposed quantization analysis protocol for perceptual image hashing based image block mean.

The image hashing system assumes that the original image I may be sent over a network consisting of possibly untrusted nodes. During the untrusted communication the original image could be manipulated for malicious purposes. Therefor, the received image $\bar{I}$ may undergo non-malicious operations like JPEG compression, etc. or malicious tampering. The final perceptual hash of I should be used to authenticate its received version $\bar{I}$. In the case of non-malicious operations, the original feature vector and the received one should differ by a small Euclidean distance which makes quantization control easier, and by a large Euclidean distance in the case of content-changing manipulations. This allows to have different results after the quantization step. Note, that even if the feature vector undergo small changes under small additive noise may cause false authentication of the received image $\bar{I}$ where it has to be considered similar to I. The received image $\bar{I}$, that we simulate like the original image plus a Gaussian noise with 0-mean and a standard deviation σ, will undergo the same steps than the original image (Fig.13) which allows to get the distribution $Dist_{\bar{I}}$ of $\bar{\boldsymbol{V}}'_{\boldsymbol{m}}(\boldsymbol{k})$. Let $\boldsymbol{V_m(k)}$ be the mean of an original image block of size $q \times p$ pixels noted by $p_{i,j}$. By the same way, we note by $\bar{\boldsymbol{V}}'_{\boldsymbol{m}}(\boldsymbol{k})$ the mean of noisy image block noted by $p'_{i,j}$. $\bar{\boldsymbol{V}}'_{\boldsymbol{m}}(\boldsymbol{k})$ can be expressed as function of $\boldsymbol{V_m(k)}$ as follow:

$$\begin{aligned}
\bar{V}'_m(k) &= \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} p'_{i,j} \\
&= \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} (p_{i,j} + n_{i,j}) \\
&= \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} p_{i,j} + \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} n_{i,j} \\
&= V_m(k) + \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} n_{i,j}
\end{aligned} \tag{15}$$

where $n_{i,j}$ is a Gaussian noise belongs to $\mathcal{N}_{0,\sigma}$ and $k \in \{1, 2, \ldots, \frac{N}{q} \times \frac{M}{q}\}$.

The term "$\frac{1}{p\times q}\sum_{i=1}^{p}\sum_{j=1}^{q} n_{i,j}$" in Equation 15 belongs to Gaussian distribution with 0-mean and standard deviation $\frac{\sigma}{\sqrt{p\times q}}$.

$\bar{V}'_m$ is the discrete vector which contains the quantized values of the computed means of the received image blocks. The comparison between $Dist_I$ and $Dist_{\bar{I}}$ allows us to get the information about the percentage of stable features that stayed fix after the additive Gaussian noise, the percentage of the features that moved to the left neighbor quantization interval and the percentage of the features that moved to the right neighbor quantization interval. This information of the features behavior is very useful, it allows us to take into account the percentage of the stable features that resist to non-malicious operations, simulated by an additive Gaussian noise. Also, it allows us to control the parameters of blocks size division and quantization step size to achieve an aimed level of the image hashing system robustness against a given level of additive noise. Selected features will then be hashed in the step of "Compression and Encryption" as shown in Figure 1. The "Compression and Encryption" stage is achieved by the cryptographic hash function SHA-1 generating a final hash of 160-bits with height level of security.

4.2 Experimental analysis of the quantization problem in a perceptual image hashing system

In the experiments of the proposed scheme, the features are the means of different image block sizes. The computed image block are sized: 4×4, 8×8 and 16×16. Then after, they are quantized by different quantization step sizes: Q=1, Q=4 and Q=16. In other words, for each given quantization step size, we tested different image block sizes against different levels of the additive Gaussian noise. The experiments are tested for a large database of grayscale images of size 512×512. Figure 3 shows the variation of mean distribution for different image block sizes and different levels of additive Gaussian noise in the case of quantization step size $Q = 4$ applied for the image Figure 10(a). In the case of the quantization step size Q=4 and standard deviation σ=1 (Figure 10(b)) (Table 3), we observe that unstable mean block features decrease when we increase the block size. We note also that the percent of stable mean block features is significant even in the case of block size equals to 4×4 (Table 4). When the standard deviation in the additive Gaussian noise increase (case of σ=5 shown in Table 3) while keeping the visual contents of the noisy image the same as the original image 10(a), the percentage of the stable mean block features decrease compared to the case of σ=1. When the visual contents of the noisy/attacked image changes Figure 10(l) than the original one (case of σ=40), we observe that a little of mean block features remain stable for all the block size that we tested as shown in Table 3.

The obtained numerical results in Table 4 present the percentage of features that have not moved and remain stable under different additive Gaussian noise and also those that drop from the left neighbor quantization interval or from the right neighbor quantization interval for each size of image block. As we can observe in Table 4, the percentage of stable features that remain fixed after adding Gaussian noise decreases when the level of the noise increases. For the same level of noise, the percentage of stable features increase when the the image block size increase. Thus, if we set the quantization step size to $Q = 1$, we can take into account the percentage of stable features that resist against tolerable level of the additive Gaussian noise. For example, if we fix the quantization step size Q equals to value 1 and we

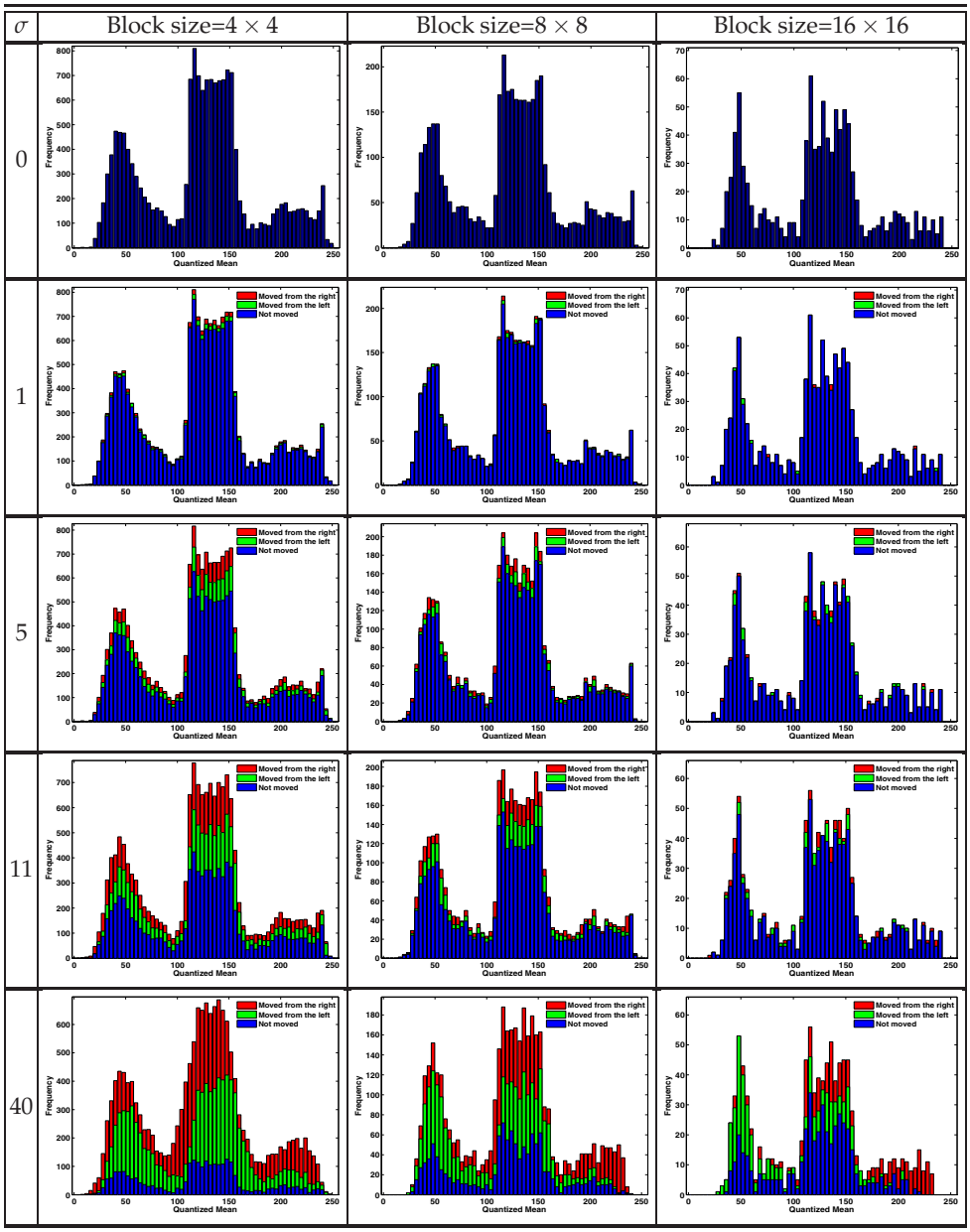
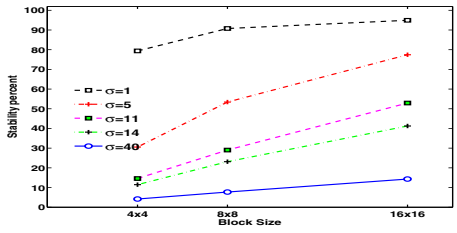


Table 3. Variation of mean distribution for different image block sizes and different levels of additive Gaussian noise in the case of quantization step size $Q = 4$.

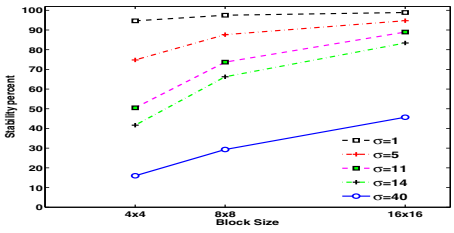
consider that an image which undergos a tolerable manipulations equivalent to an additive Gaussian noise whose a standard deviation equal to $\sigma = 5$, we choose a compromise between

the percentage of stable features and the size of the blocks image decomposition. For the block size equal 4×4 we have to take into account the maximum percent of stable features $\approx$ 30% and if the block size equals 8×8, we take into account the maximum percent of stable features $\approx$ 54%. The highest percentage of stable features $\approx$ 77% can be taken if we applied a 16×16 in the preprocessing image treatment. We tested our experiment on a large database of grayscale images of size 512×512 and we observed that these values presented in Table 4 can be obtained approximatively for others images of the same settings of image blocks decomposition and Gaussian noise addition, also we noted that the percentages of features that moved from the left and those moved from the right approximately equals which coincides with the theoretical study presented in Section 3.2.2. Same remarks of the approximately equalities of the percentages that moved from the left and the right are observed in the cases of Q=4 and Q=16 than in the case of the quantization step size Q=1. These obtained numerical values are almost approximately fixed in the same settings parameters in the block image decomposition and the level of Gaussian noise addition because we tested our experiments on large database grayscale images. These values are obtained for the grayscale image shown in Figure 10(a) and can be obtained for any other grayscale image.

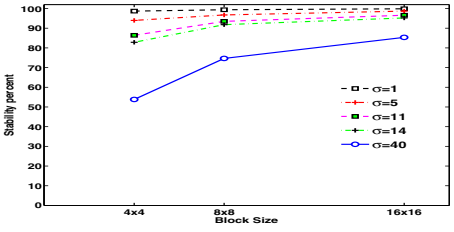
Based on the numerical results presented in Table 4, Figure 14 shows the percentage of the features that remain stable under the additive Gaussian noise for different image blocks decomposition. As we remark, to get a high percentage of of stable features, we have two possibilities: either we apply great size of image block decomposition or the original image undergos small additive Gaussian noise.



(a) Case of quantization step size Q=1.



(b) Case of quantization step size Q=4.



(c) Case of quantization step size Q=16.

Fig. 14. Stability percent of mean features for a fixed quantization step size for different block sizes: (a) Case of quantization step sizes $Q = 1$, (b) Case of quantization step size $Q = 4$ and (c) Case of quantization step size $Q = 16$.

Q	Block Size	σ	**(%)** Not Moved	**(%)** Moved from the Right	**(%)** Moved from the Left
1	4×4	1	79.4128	10.4004	10.1868
		5	30.5237	34.8633	34.6130
		11	14.5569	42.5842	42.8589
		14	11.4258	43.0176	45.5566
		40	4.1382	47.5281	48.3337
	8×8	1	**90.7471**	4.5410	4.7119
		5	53.4180	23.3643	23.2178
		11	29.0283	35.0586	35.9131
		14	23.0957	36.3037	40.6006
		40	7.6660	46.5332	45.8008
	16×16	1	**94.9219**	2.1484	2.9297
		5	77.4414	11.8164	10.7422
		11	52.9297	23.5352	23.5352
		14	41.2109	27.2461	31.5430
		40	14.2578	43.5547	42.1875
4	4×4	1	**94.6960**	2.7710	2.5330
		5	74.7864	12.8540	12.3596
		11	50.4456	24.6826	24.8718
		14	41.6382	28.4851	29.8767
		40	15.9119	41.8457	42.2424
	8×8	1	**97.5098**	1.2939	1.1963
		5	87.6221	6.0547	6.3232
		11	73.7061	12.7930	13.5010
		14	66.2598	15.4297	18.3105
		40	29.2725	35.8154	34.9121
	16×16	1	**98.9258**	0.5859	0.4883
		5	**94.7266**	3.0273	2.2461
		11	88.9648	4.9805	6.0547
		14	83.3984	7.7148	8.8867
		40	45.7031	28.5156	25.7812
16	4×4	1	**98.6694**	0.6714	0.6592
		5	**93.9575**	3.0273	3.0151
		11	86.3953	6.6162	6.9885
		14	82.8918	8.0811	9.0271
		40	53.8086	22.5220	23.6694
	8×8	1	**99.4141**	0.2686	0.3174
		5	**96.7529**	1.5625	1.6846
		11	**93.5059**	3.0518	3.4424
		14	**91.7969**	3.5645	4.6387
		40	74.6826	12.5244	12.7930
	16×16	1	**99.9023**	0.0000	0.0977
		5	**98.7305**	0.7812	0.4883
		11	**96.5820**	1.3672	2.0508
		14	**95.2148**	1.9531	2.8320
		40	85.3516	6.9336	7.7148

Table 4. Numerical results for different levels of the additive Gaussian noise and image block size in the case of the quantization step sizes $Q = 1$, $Q = 4$ and $Q = 16$.

5. Conclusion

In this chapter, we introduced the main aim of the perceptual image hashing field in image security. We presented the important merits and requirements of a perceptual image hash function used for authentication wherein a formulation of the perceptual image hashing problem was given. We dedicated a section to presenting an overview of recent techniques that are used for perceptual image hashing. After, we presented the different quantization techniques used for more robustness of a perceptual image hashing scheme showing their advantages and their limitations. Finally, we presented a theoretical model describing the behavior of the extracted image features to be hashed against content-preserving/content-changing manipulations. In the presented analysis, we simulated the manipulations that may undergo the original image by an additive Gaussian noise. We tested the presented model by several experiments to demonstrate the effectiveness of the proposed theoretical model giving practical analysis for robust perceptual image hashing. The presented model is applied on image hashing based on statistical invariance of mean block features. The obtained results confirms the theoretical study presented in Section 3.2. Some approximations must be done to improve results. The same study can be generalized for other features in block-based image hashing scheme like DCT domain features, DWT domain features, etc.

6. References

Ahmed, F. & Siyal, M. Y. (2006). A secure and robust wavelet-based hashing scheme for image authentication, *in* T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua & L.-T. Chia (eds), *Advances in Multimedia Modeling*, Vol. 4352 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62.

Ahmed, F., Siyal, M. Y. & Abbas, V. U. (2010). A secure and robust hash-based scheme for image authentication, *Signal Processing* 90: 1456–1470.

Bender, W., Gruhl, D., Morimoto, N. & Lu, A. (1996). Techniques for data hiding, *IBM Systems Journal* 35(3-4): 313–336.

Bhattacharjee, S. K. & Kutter, M. (1998). Compression tolerant image authentication, *Proceedings of the IEEE International Conference on Image Processing (ICIP (1))*, pp. 435–439.

Cox, I. J., Miller, M. L. & Bloom, J. A. (2000). Watermarking applications and their properties, *Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'00)*, IEEE Computer Society, Las Vegas, NV, USA, pp. 6–10.

Cox, I. J., Miller, M. L. & Bloom, J. A. (2002). *Digital watermarking*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Fridrich, J. (2000). Visual hash for oblivious watermarking, *SPIE Photonic West Electronic Imaging 2000, Security and Watermarking of Multimedia Contents*, Vol. 3971, SPIE, San Jose, California, pp. 286–294.

Fridrich, J. & Goljan, M. (2000). Robust hash functions for digital watermarking, *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'00)*, IEEE Computer Society, Washington, DC, USA, pp. 178–183.

Furht, B., Socek, D. & Eskicioglu, A. M. (2004). Fundamentals of multimedia encryption techniques, *IN MULTIMEDIA SECURITY HANDBOOK*, CRC Press, pp. 93–131.

Guo, X. C. & Hatzinakos, D. (2007). Content based image hashing via wavelet and radon transform, *Proceedings of the multimedia 8th Pacific Rim conference on Advances*

in multimedia information processing, PCM'07, Springer-Verlag, Berlin, Heidelberg, pp. 755–764.

Hadmi, A., Puech, W., AitEssaid, B. & Aitouahman, A. (2010). Analysis of the robustness of wavelet-based perceptual signatures, *IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA'10)*, Paris, France, pp. 112–117.

Hadmi, A., Puech, W., AitEssaid, B. & Aitouahman, A. (2011). Statistical analysis of the quantization stage of robust perceptual image hashing, *IEEE 3rd European Workshop on Visual Information Processing (EUVIP'11)*, Paris, France.

Han, S. H. & Chu, C. H. (2010). Content-based image authentication: current status, issues, and challenges, *International Journal of Information Security* 9: 19–32.

Kerckhoffs, A. (1883). La cryptographie militaire, *Journal des sciences militaires* 9(1): 5–38.

Khelifi, F. & Jiang, J. (2010). Perceptual image hashing based on virtual watermark detection, *IEEE Transactions on Image Processing* 19: 981–994.

Kozat, S. S., Venkatesan, R. & Mihçak, M. K. (2004). Robust perceptual image hashing via matrix invariants, *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, pp. 3443–3446.

Lefebvre, F., Macq, B. & Legat, J. D. (2002). Rash: Radon soft hash algorithm, *Proceedings of the European Signal Processing Conference (EUSIPCO'02)*, Toulouse, France.

Lei, Y., Wang, Y. & Huang, J. (2011). Robust image hash in radon transform domain for authentication, *Signal Processing: Image Communication* 26: 280–288.

Lin, C. Y. & Chang, S. F. (2001). A robust image authentication method distinguishing jpeg compression from malicious manipulation, *IEEE Transactions on Circuits and Systems for Video Technology* 11(2): 153–168.

Lu, C. S. & Liao, H. Y. M. (2003). Structural digital signature for image authentication: an incidental distortion resistant scheme, *IEEE Transactions on Multimedia* 5(2): 161–173.

Memon, N. & Wong, P. W. (1998). Protecting digital media content, *Communication ACM* 41: 35–43.

Menezes, A. J., Vanstone, S. A. & Oorschot, P. C. V. (1996). *Handbook of Applied Cryptography*, 1st edn, CRC Press, Inc., Boca Raton, FL, USA.

Mihçak, M. K. & R.Venkatesan (2001). New iterative geometric methods for robust perceptual image hashing, *Digital Rights Management Workshop*, pp. 13–21.

Mihçak, M. K. & Venkatesan, R. (2001). A perceptual audio hashing algorithm: A tool for robust audio identification and information hiding, *Proceedings of the 4th International Workshop on Information Hiding*, IHW '01, Springer-Verlag, London, UK, UK, pp. 51–65.

Monga, V. (2005). *Perceptually Based Methods for Robust Image Hashing*, Phd dissertation, University of Texas at Austin.

Monga, V. & Evans, B. L. (2006). Perceptual image hashing via feature points: Performance evaluation and trade-offs, *IEEE Transactions on Image Processing* 15(11): 3452–3465.

NIST (2008). FIPS PUB 180-3, Federal Information Processing Standard (FIPS), Secure Hash Standard (SHS), Publication 180-3, *Technical report*, National Institute of Standards and Technology, Department of Commerce.

Ozturk, I. & Ibrahim, S. (2005). Analysis and comparison of image encryption algorithms, *Education Technology and Training & Geoscience and Remote Sensing* 3: 803–806.

Puech, W., Rodrigues, J. M. & Develay-Morice, J. E. (2007). A new fast reversible method for image safe transfer, *Journal of Real-Time Image Processing* 2(1): 55–65.

Rivest, R. L. (1992). The MD5 Message-Digest Algorithm, *Technical Report RFC 1321*, Internet Engineering Task Force (IETF).

Rodrigues, J. M., Puech, W. & Bors, A. G. (2006). Selective encryption of human skin in jpeg images, *Proceedings of the IEEE International Conference on Image Processing (ICIP'06)*, pp. 1981–1984.

Schneider, M. & Chang, S. F. (1996). A robust content based digital signature for image authentication, *Proceedings of the IEEE International Conference on Image Processing (ICIP'96)*, Vol. 3, pp. 227–230.

Seo, J. S., Haitsma, J., Kalker, T. & Yoo, C. D. (2004). A robust image fingerprinting system using the radon transform, *Signal Processing: Image Communication* 19(4): 325–339.

Stinson, D. (2002). *Cryptography: Theory and Practice*, 2nd edn, Chapman & Hall, CRC.

Sun, Q. & Chang, S. F. (2005). A robust and secure media signature scheme for jpeg images, *VLSI Signal Processing* 41(3): 305–317.

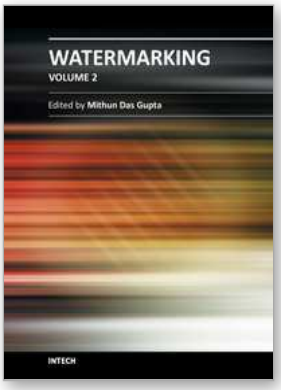
Swaminathan, A., Mao, Y. & Wu, M. (2006). Robust and secure image hashing, *IEEE Transactions on Information Forensics and Security* 1(2): 215–230.

Venkatesan, R., Koon, S. M., Jakubowski, M. H. & Moulin, P. (2000). Robust image hashing, *Proceedings of the IEEE International Conference on Image Processing (ICIP'00)*, pp. 664–666.

Wang, S. Z. & Zhang, X. P. (2007). Recent development of perceptual image hashing, *Journal of Shanghai University* 11: 323–331.

Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity, *IEEE Tansactions on Image Processing* 13(4): 600–612.

Zhu, G., Huang, J., Kwong, S. & Yang, J. (2010). Fragility analysis of adaptive quantization-based image hashing, *IEEE Transactions on Information Forensics and Security* 5: 133–147.



Watermarking - Volume 2

Edited by Dr. Mithun Das Gupta

ISBN 978-953-51-0619-7
Hard cover, 276 pages
Publisher InTech
Published online 16, May, 2012
Published in print edition May, 2012

This collection of books brings some of the latest developments in the field of watermarking. Researchers from varied background and expertise propose a remarkable collection of chapters to render this work an important piece of scientific research. The chapters deal with a gamut of fields where watermarking can be used to encode copyright information. The work also presents a wide array of algorithms ranging from intelligent bit replacement to more traditional methods like ICA. The current work is split into two books. Book one is more traditional in its approach dealing mostly with image watermarking applications. Book two deals with audio watermarking and describes an array of chapters on performance analysis of algorithms.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Azhar Hadmi, William Puech, Brahim Ait Es Said and Abdellah Ait Ouahman (2012). Perceptual Image Hashing, Watermarking - Volume 2, Dr. Mithun Das Gupta (Ed.), ISBN: 978-953-51-0619-7, InTech, Available from: http://www.intechopen.com/books/watermarking-volume-2/perceptual-image-hashing

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821