Signal Processing: *Image Communication* ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A robust and secure perceptual hashing system based on a quantization step analysis

Azhar Hadmi [a,b], William Puech [a,*], Brahim Ait Es Said [b], Abdellah Ait Ouahman [b]

[a] *LIRMM Laboratory, UMR 5506 CNRS, University of Montpellier II, 161, rue Ada, 34095 Montpellier cedex 05, France*
[b] *Team of Telecommunications and Network, University of Cadi Ayyad, Faculty of Science Semlalia, 40000 Marrakech, Morocco*

ARTICLE INFO

Keywords:
Robust hashing
Quantization
Multimedia security
Content identification
Image hashing
Image authentication

ABSTRACT

Perceptual hashing is conventionally used for content identification and authentication. It has applications in database content search, watermarking and image retrieval. Most countermeasures proposed in the literature generally focus on the feature extraction stage to get robust features to authenticate the image, but few studies address the perceptual hashing security achieved by a cryptographic module. When a cryptographic module is employed [1], additional information must be sent to adjust the quantization step. In the perceptual hashing field, we believe that a perceptual hashing system must be robust, secure and generate a final perceptual hash of fixed length. This kind of system should send only the final perceptual hash to the receiver via a secure channel without sending any additional information that would increase the storage space cost and decrease the security. For all of these reasons, in this paper, we propose a theoretical analysis of full perceptual hashing systems that use a quantization module followed by a crypto-compression module. The proposed theoretical analysis is based on a study of the behavior of the extracted features in response to content-preserving/content-changing manipulations that are modeled by Gaussian noise. We then introduce a proposed perceptual hashing scheme based on this theoretical analysis. Finally, several experiments are conducted to validate our approach, by applying Gaussian noise, JPEG compression and low-pass filtering.

1. Introduction

With the fast progress in 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 is 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 check the safety of exchanged multimedia data confidentiality, authenticity and integrity. To ensure confidentiality, multimedia data should stay unintelligible without a decryption key. This is achieved mainly through encryption secret key or public key. Authentication is an another crucial multimedia data protection issue. It makes it possible to trace the author of multimedia data and to determine if the original multimedia data content was altered in any way from the time of its recording. Integrity allows multimedia degradation detection and helps to make sure that the received multimedia data has not been modified by a third party for malicious reasons. In the area of multimedia security, two types of approaches have been proposed to satisfy those requirements in recent years: fragile watermarking and perceptual hashing. The main advantage of fragile watermarking is the ability to detect changes in the host multimedia data. Thus, it can

[*] Corresponding author. Tel.: +33 467418685; fax: +33 467418500.
E-mail address: william.puech@lirmm.fr (W. Puech).

0923-5965/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.image.2012.11.009

provide some form of guarantee that the multimedia data has not been tampered with and has originated from the right source. Watermarking can be used in copyright checking or content authentication for individual images, but is not suitable when a large scale search is required. Furthermore, data embedding inevitably causes slight distortion in the host multimedia data [2]. The main advantage of perceptual hashing schemes is that the multimedia data is not altered and not degraded at all. Perceptual hashing schemes are inspired from the cryptographic hash functions to authenticate multimedia. Traditionally, data integrity issues are addressed by cryptographic hashes or message authentication functions such as MD5 [3] and SHA series [4], which are sensitive to each bit of the input message. Consequently, the message integrity is validated when each bit of the message is unchanged [5]. The sensitivity of cryptographic hashes is not suitable for multimedia data, since the information it carries is mostly retained even when the multimedia data has undergone various content preserving operations. Perceptual hashing methods have recently been proposed as primitives to overcome the above problems and have constituted the core of a challenging developing research area for academic and multimedia industry stakeholders. Perceptual hashing functions extract features from images and calculate a hash value based on these features. Such functions have been proposed to establish the "perceptual equality" of the image content. The performance of a perceptual hashing function primarily consists of robustness, discrimination and security. Robustness means the perceptual hashing function always generates the same perceptual hash values for perceptually similar images. Discrimination means that different imageinputs must result in totally different hash values. The security of a perceptual hashing function means that it is impossible for an adversary to keep the same perceptual hash value when the image content is perceptually modified. Image authentication is performed by comparing the hash value of the original image and the image to be authenticated. Perceptual hashes are expected to be able to survive acceptable content-preserving manipulations and reject malicious manipulations.

In this paper, we develop a theoretical analysis of full perceptual hashing systems that use a quantization module followed by a crypto-compression module. From this analysis, we propose to combine the extraction of robust visual features with a cryptographic hash function to result in a robust and secure perceptual hashing procedure. In Section 2, we provide an overview of perceptual hashing systems. In Section 3, we present a theoretical analysis of the quantization problem in perceptual hashing systems. Section 4 presents the quantization analysis protocol based on the statistical invariance of the extracted features. Section 5 presents our proposed perceptual hashing method based on a theoretical analysis of the quantization problem in perceptual hashing systems. In Section 6, several experimental results are presented that validate our proposed approaches. Section 7 finally concludes this paper with some perspectives.

2. Overview of perceptual hashing

2.1. Perceptual hashing framework

A perceptual 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 *crypto-compression* stage.

In the *transformation* stage, the input image undergoes spatial and/or frequency transformation. Those transformations make all extracted features depend on image pixel values of or image frequency coefficients. In the *feature extraction* stage, the perceptual hashing system extracts the image features from the transformed image to generate a continuous intermediate hash vector. Then the continuous intermediate hash vector is quantized into the discrete hash vector in the *quantization* stage to form an intermediate binary perceptual hash vector. Finally, the intermediate binary perceptual hash vector is compressed and encrypted into a short and a final perceptual hash at the *crypto-compression* stage.

2.1.1. Transformation stage

In the *transformation* stage, the input image of size $M \times N$ bytes undergoes spatial transformations such as

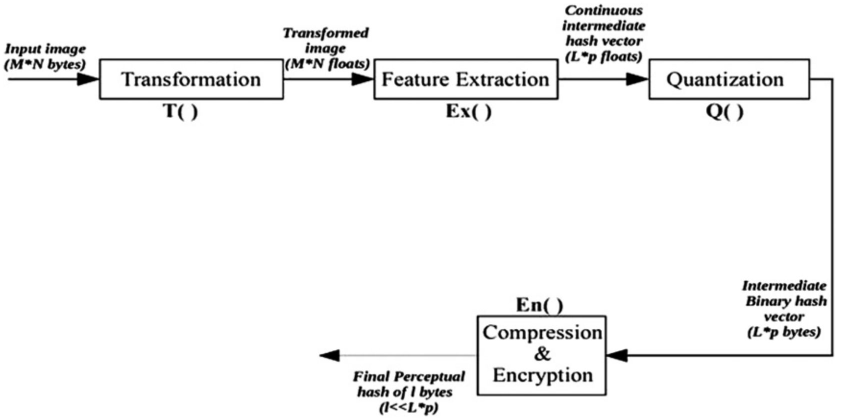


Fig. 1. Four pipeline stages of a perceptual hashing system.

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009

color transformation, smoothing, affine transformations, or frequency transformations involving the discrete cosine transform (DCT) or discrete wavelet transform (DWT). When a DWT is applied, most perceptual hashing schemes take just the LL subband into account because it is a coarse version of the original image and contains all of the perceptually information. The principal aim of these transformations is to make all extracted features depend upon the image pixel values (in case of spacial transformation) or the their frequency coefficients (in case of frequency transformation).

2.1.2. Feature extraction stage

In the *feature extraction* stage, the perceptual hashing system extracts the image features from the transformed image to generate the feature vector of L features, where $L \ll M \times N$. Note that each feature can contain p elements of type *float*, which means that we get $L \times p$ *floats* at this stage. However, which mappings (if any) from DCT/DWT coefficients preserve the essential information about an image for hashing and/or mark embedding applications is still an open question. We can add another features selections at this stage, as shown in Fig. 2, then only the most pertinent features are selected.These are statistically more resistant against a specific allowed manipulation like the addition of noise, JPEG compression and filtering. The selected features can be presented as an intermediate hash vector of $K \times p$ *floats*, where $K < L$. Note that the visual features selected are usually publicly known and can therefore be modified. This might threaten security, as the hash value could be adjusted maliciously to match that of another image.

2.1.3. Quantization stage

In the *quantization* stage, we get a quantized intermediate perceptual hash vector which contains $K \times p$ elements of type *byte*. Uniform quantization can be applied to quantize each component of the continuous perceptual hash vector. Adaptive quantization [6] is another quantization type which 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.

2.1.4. Compression and Encryption stage

The *compression and encryption* stage is the final step of a perceptual hashing system which guarantees both the system security and the fixed length of the final perceptual hash. The binary intermediate perceptual hash vector is compressed and encrypted into a short perceptual hash of fixed size of l bytes, where $l \ll K \times p$, 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 that generate the final hash with a fixed size (hash of 160 bits in case of SHA-1).

2.2. Metrics and important requirements of perceptual 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 hashing techniques is very challenging and should involve a trade-off between various conflicting requirements. Let P denote probability and $H()$ denote a perceptual hash function which takes one image as input and produces a binary string of length l, as presented in Fig. 1. 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_I and $h_{I_{diff}}$ 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 sought after properties of a perceptual hashing function are identified as follows:

- Equal distribution (unpredictability) of hash values

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

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

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

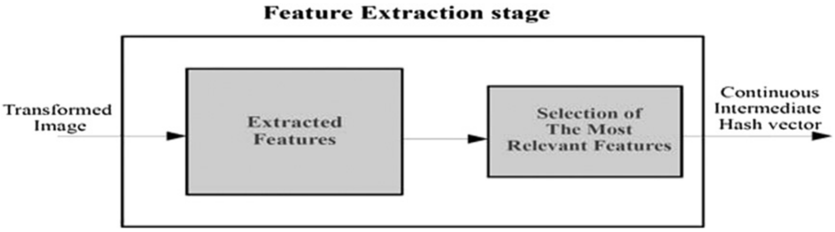

Feature Extraction stage
Transformed Image
Extracted Features
Selection of The Most Relevant Features
Continuous Intermediate Hash vector

Fig. 2. Selection of the most relevant features in the feature extraction stage.

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

$$P(H(I)=H(I_{ident}))\geq 1-\theta \quad \text{for a given } \theta\approx 0 \tag{3}$$

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

$$P(H(I)\neq H(I_{diff}))\geq 1-\theta \quad \text{for a given } \theta\approx 0 \tag{4}$$

To fulfil the property in Eq. (3), most perceptual hash functions try to extract features of images which are invariant under insignificant global modifications such as compression or enhancement. Eq. (4) means that, given an image I, it is almost impossible for an adversary to create a perceptually different image I_{diff} with $H(I_{diff})=H(I)$. This property can be hard to achieve because the features used by published perceptual hash functions are publicly known [7,8] and can therefore be modified. This might threaten security. Thus, the property in Eq. (3) will have to be neglected in favor of the property in Eq. (4). Likewise for perfect unpredictability, an equal hash value distribution (Eq. (1)) is needed. This would deter fulfillment of the property in Eq. (3) [9]. Note that requirements (1), (2) and (4) are the basic requirements of cryptographic hash functions, whereas requirement (3) focuses entirely on the robustness property of perceptual hashing functions. 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. A lack of robustness (Eq. (3)) renders an image hash useless, as explained above, while security (Eqs. (1) and (4)) means that it is extremely difficult for an adversary to modify the essential content of an image yet keep the hash values unchanged. Thus, trade-offs must be sought, and this is usually the central issue of perceptual hashing research.

2.3. *Content-preserving manipulations vs content-changing manipulations*

The main concept behind perceptual hashing for authentication is to extract image features based on human perception and then to encrypt and compress them to form the final perceptual hash that can be used for authentication. Typically, some applications may need to apply some non-malicious manipulations to enhance the original image quality such as low-pass filtering, minor contrast enhancement, or even image cropping, changing the size, or other operations. Some applications may also require lossy compression to meet the bandwidth or storage space resource constraints. Content-preserving manipulations only change the pixel values, which results in different levels of visual distortion in the image, but the image contents, which carry the same visual information to the receiver, are still preserved. On the other hand, malicious/content-changing manipulations consist of changing the content of the original image (captions, faces, etc.) to a new one, which carries different visual information to the receiver. One typical example of malicious modification is replacing some parts of the image with different contents for malicious use. Perceptual hashing for authentication use is expected to be able to survive acceptable content-preserving manipulations and reject other malicious manipulations.

2.4. *Review of some perceptual hashing schemes*

In recent years, a growing body of research on perceptual hashing has been receiving increased attention in the literature. Most current perceptual hashing studies mainly focus on extracting robust visual features at the feature extraction stage and then using them during the authentication step. The authors believe that robustness is ensured by extracting a set of robust visual features that withstand (or stay relatively constant) content-preserving manipulations, which is the most important objective (property of Eq. (3)). At the same time, those extracted features should detect content-changing manipulations. Few papers address perceptual hashing system security guaranteed by the use of an encryption module during the generation of the final perceptual hash (property of Eq. (4)). When an encryption module is missed in a perceptual hashing system, security properties are threatened. Those methods can roughly be classified in the four following categories [10,11]:

- *Statistic-based schemes* [12–14]: This group of schemes extracts image features by calculating the image statistics in the spatial domain such as mean, variance, higher moments of image blocks and histogram.
- *Relation-based schemes* [15,16]: This category involves approaches to extract image features by making use of some invariant relationships of the coefficients of discrete cosine transform (DCT) or wavelet transform (DWT).
- *Coarse-representation-based schemes* [17,7,18,19]: In this category, the perceptual hash is 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* [20,21]: The image features 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 a final hash value. However, the perceptual hash value is very sensitive to global as well as local distortions that do not cause perceptually significant image changes.

In [22], the hash extraction is based on the projection of image coefficients onto filtered pseudo-random patterns. The final perceptual hash is used for generating pseudo-random watermark sequences, that depend closely on a secret key throughout the image, for authentication and integrity verification of still images.

In [14], a perceptual 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. The statistical properties of wavelet subbands are generally robust against attacks, but they are only loosely related to the image contents and are therefore rather insensitive to tampering. This method has been shown to be robust against common image manipulations and geometric attacks.

The proposed method in [13] uses 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 [17] is based on 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 significantly influencing the image. 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 other techniques to perform image perceptual hashing. The authors in [19] used a Fourier–Mellin transform for perceptual hashing applications. Using the Fourier–Mellin transform scale invariant property, the magnitudes of the Fourier transform coefficients were randomly weighted and summed. However, since the 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 hashing scheme based Radon transform is proposed in [23] 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 the 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 typical image processing manipulations, including JPEG compression, geometric distortion, blur, the addition of noise and enhancement. The Radon transform was first used in [24], and further expanded in [25].

Authors in [26] propose a perceptual hashing scheme based on a combination of the discrete wavelet transform (DWT) and the Radon transform. Taking the advantages of the frequency localization property of DWT and the shift/rotation invariant property of the Radon transform, the algorithm can effectively detect malicious local changes, while also being robust against content-preserving modifications. The obtained features derived from the Radon transform are then quantized by adaptive quantization [6] to form the final perceptual hash.

In [27], perceptual image hashing based on the wave atom transform is presented. The original image is decomposed into multiscale coefficients with tilings using the wave atom transform. The perceptual hash is then extracted from the mean and variance of tilings in the third scale band. The coefficients in the third scale band change slightly when the image content is not altered visually, but the coefficients vary significantly when malicious tampering takes place. This property is very useful in a perceptual image hashing system.

The authors in [28] propose a histogram-based perceptual image hashing function using the resistance of two statistical features: image histogram in shape and mean value. The relative magnitudes of the histogram values are computed in two (or a few) different bins for hashing. The scaling invariance of the histogram shape and the independence of the histogram to the pixel position can thus be fully applied for different content-preserving geometric distortions. The perceptual hash value is robust to various common image processing and geometric deformation operations since the histogram is extracted from a low-frequency image component. The proposed hashing scheme can be used for practical applications, e.g. for searching content-preserving copies from the same source. The proposed method is highly robust against perceptually insignificant attacks. However, the fragility to malicious attacks is a drawback. An improvement of this method is proposed in [29], where the authors propose an improved histogram-based image hashing scheme using a *K*-means algorithm, which obtains a better fragility result than in [30].

The common aspect between all of the above mentioned schemes is that they do not really take the *crypto-compression* stage into account. They are only satisfied by extracting visual features by several image processing techniques and quantizing them to generate the final perceptual hash using, in some cases, a secret key to enhance the system security. When the crypto-compression stage is missed, security properties are threatened. As the features used by published perceptual hash functions are publicly known, they can therefore be modified and adjusted maliciously to match that of another image. Referring to the image space shown in Fig. 3, and based on the pipeline presented in Fig. 1, we obtain the following equations when the crypto-compression stage is ignored:

- Invariance for perceptually similar images I and I_{ident}
$$P(Ex(T(I)) = Ex(T(I_{ident}))) \geq 1-\theta \quad \text{for a given } \theta \approx 0 \qquad (5)$$

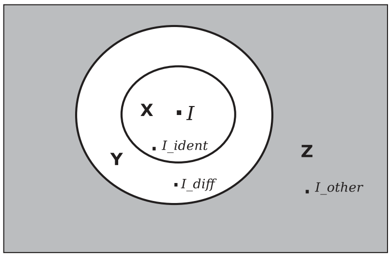


Fig. 3. The image space $\{I\} \cup X \cup Y \cup Z$ formed by an image $\{I\}$, its perceptually similar version set X, its modified version set Y and all other image set Z.

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009

- Invariance for perceptually different images I and I_{diff}

$$P(Ex(T(I))=Ex(T(I_{diff})))\geq 1-\theta_1 \quad \text{for a given } \theta_1\approx 0 \tag{6}$$

- Distinction of different images I_{other} from I

$$P(Ex(T(I))\neq Ex(T(I_{other})))\geq 1-\theta \quad \text{for a given } \theta\approx 0 \tag{7}$$

where $P()$, $Ex()$, $T()$ stand for probability, extraction stage, and transformation stage, respectively, as presented in Fig. 1.

From a practical standpoint, there is no guarantee of having the natural constraint $\theta<\theta_1$ and, moreover, $\theta-\theta_1 \backsimeq 0$ is realistic. Consequently, the feature extraction step, to generate a multimedia perceptual hash, is not enough to achieve a secure perceptual hashing system.

Another problem in such cases is that the size of the final perceptual hash has no fixed length and is usually greater (~ Mbytes instead of few bits).

In the next section, we present the quantization problem in perceptual hashing schemes, and then we give an overview of some perceptual hashing schemes that take the crypto-compression stage into account.

3. Theoretical analysis of the quantization problem in perceptual hashing

3.1. Problem statement

As explained in Section 2.1.3, the quantization stage in a perceptual hashing system involves discretizing the continuous intermediate hash vector (continuous features) into a discrete intermediate hash vector (discrete features). This step is very important to decrease the data size in order to compress and encrypt it. It also enhances the robustness properties by minimizing the collision probabilities of a perceptual hashing system. Quantization is the conventional way to achieve this goal. The quantization step is a difficult process because it is not known how the values in the continuous intermediate hash drop after content-preserving (non-malicious) manipulations in each quantization interval of size Q. This difficulty of achieving efficient quantization increases more when it is followed by a crypto-compression stage, i.e. SHA-1, because the discrete intermediate hash vectors must be quantized in a correct way for all similar perceptual images. For this reason, this stage is ignored in most perceptual hashing 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 additive noise range is B. Suppose that the original scalar values $x_l \in \mathbb{R}$ for $l \in \{1,\ldots,L\}$ of the continuous intermediate hash are bounded to a finite interval $[-A,A]$. Furthermore, suppose that we wish to obtain a quantized message $q(x_l)$ of x_l in N quantization points given by the set $\tau=\{\tau_1,\ldots,\tau_N\}$. The points are uniformly spaced such that $Q=\tau_j-\tau_{j-1}=2A/(N-1)$ for $j\in\{1,\ldots,N\}$. 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, noise corruption will cause a different quantization result and automatically cause different perceptual hashes [31]. Fig. 4 shows the distribution of the original DWT LL-subband (level 3) coefficients, of a Lena 512 × 512 sized image, in the interval [40,50] and their noisy version, in the same interval [40,50], by an additive Gaussian noise of standard deviation equal to $\sigma=1$. When

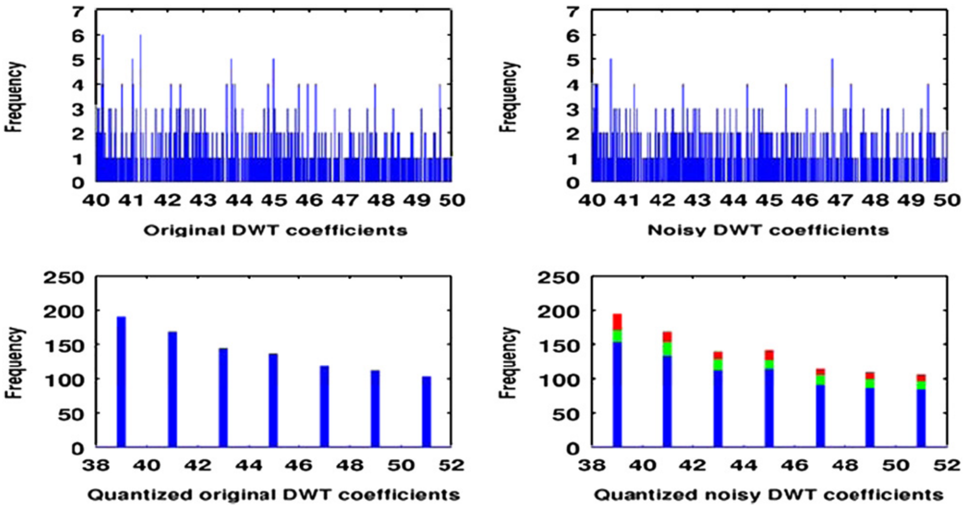


Fig. 4. The influence of additive Gaussian noise on quantization ($Q=2$) of the original DWT LL-subband coefficients and a noisy version in the interval [40,50]. In green: DWT LL-subband quantized coefficients that dropped from the right neighboring quantization interval. In red: DWT LL-subband quantized coefficients that dropped from the left neighboring quantization interval. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

a Gaussian noise with $\sigma = 1$ is applied, the noisy image remains perceptually identical to the original image, but it causes changes in the extracted feature distribution as we can see in Fig. 4. This causes errors in the quantization step because the quantized features do not remain unchanged after noise addition.

3.2. *Proposed quantization techniques to solve the quantization problem*

To avoid the quantization problem discussed in Section 3.1 and feature instability after minor changes in the original image, various quantization schemes have been proposed in the literature. Authors in [32] propose an error correction coding (ECC) to correct errors in extracted features caused by corruption from additive noise to get the same quantization result before and after noise addition. In their work, they assume that the quantization step size Q verifies: $Q > 4B$, where B is the maximum range of additive noise, and then they push the original feature points away from the quantization decision boundaries and create a margin of at least $Q/4$. Thus, the original x_l value when later contaminated will not exceed the quantization decision boundaries. Suppose that the original feature value is located at the point nQ, then no matter how this value is corrupted, the distorted value will still be in the range $[(n-0.5)Q,(n+0.5)Q)]$, and the quantized feature will remain unchanged as nQ before and after noise addition. However, if the original feature P (Fig. 5) drops in the range $[(n-0.5)Q,nQ]$, its quantized value is still nQ before adding noise, but there is also a possibility that the noisy feature could drop in the range $[(n-1)Q,(n-0.5)Q)]$ (point P' in Fig. 5) and will be quantized as $(n-1)Q$ after adding noise. Thus, noise corruption will cause a different quantization result. As a solution to the quantization problem, authors propose an error correction coding (ECC) procedure based on the quantizer output: the quotient F (integer rounding) and remainder R given by

$$F = \left[\frac{x_l}{Q}\right], \quad R = x_l - F \times Q \tag{8}$$

In the authentication procedure, add the value $0.25Q$ if $R < 0$ and subtract the value $0.25Q$ if $R \geq 0$, to keep the features in the range $[(n-0.5)Q,(n+0.5)Q)]$ and then keep the quantized value the same as the original quantized value nQ even after adding noise. The error correction concept is illustrated in Fig. 5. Note that the vector R has the same length as the vector that contains the extracted features and needs to be transmitted to adjust the quantization on the receiver side which is too costly in storage space. It is also hypothesized that $Q > 4B$ is not always true from a practical point of view.

Other similar work based on this approach has recently been proposed [1], where the authors calculate and record a vector of 4-bits called “perturbation information”. This additional transmitted information has the same dimension as 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 in the “perturbation information” helps to make a decision to positively authenticate an image or not. Their theoretical analysis is more general than in [32] from a practical standpoint. One main disadvantage of such scheme is that the vectors used to correct errors in extracted features need to be transmitted or stored beside the image and the final hash which is too costly in storage space. Their proposed schemes are illustrated in Figs. 6 and 7.

Another quantization scheme which is widely applied in perceptual hashing [19,10] proposed by [6] is called *adaptive quantization* or *probabilistic quantization* in [9]. Its key feature is that it takes the distribution of the input data into account. The quantization intervals $Q = \tau_j - \tau_{j-1}$ for $j \in \{1, \ldots, N\}$ are designed so that $\int_{\tau_{j-1}}^{\tau_j} p_X(x)\,dx = 1/N$, where N is the number of quantization levels and $p_X(.)$ is the pdf of the input data X. The central points $\{C_j\}$ are

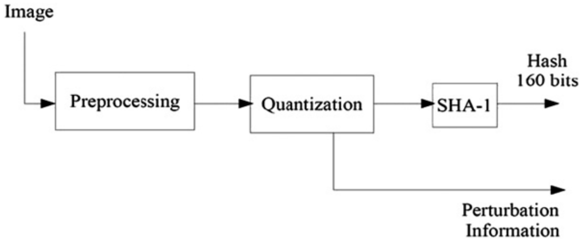


Fig. 6. Hash generation module with quantization in the Fawad et al. scheme [1].

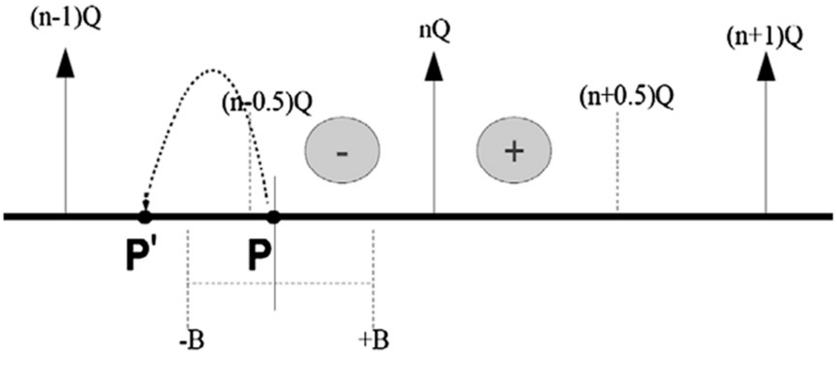


Fig. 5. Illustration on the concept of error correction in the Sun et al. scheme [32].

defined so as to make $\int_{\tau_{j-1}}^{C_j} p_X(x)\,dx = \int_{C_j}^{\tau_j} p_X(x)\,dx = 1/(2N)$. Around each τ_j, a randomization interval $[A_j,B_j]$ is introduced such that $\int_{A_j}^{\tau_j} p_X(x)\,dx = \int_{\tau_j}^{B_j} p_X(x)\,dx = r/N$, where $r \le 1/2$. The randomization interval is symmetric around τ_j for all j in terms of distribution p_X. The natural constraint must be respected $C_j \le A_j$ and $B_j \le C_{j+1}$. The overall quantization rule is then given by

$$q(x_l) = \begin{cases} j-1 \text{ w.p. } 1 & \text{if } C_j \le x_l < A_j \\ j-1 \text{ w.p. } \left(\dfrac{N}{2r}\int_{x_l}^{B_j} p_X(t)\,dt\right) & \text{if } A_j \le x_l < B_j \\ j \text{ w.p. } \left(\dfrac{N}{2r}\int_{A_j}^{x_l} p_X(t)\,dt\right) & \text{if } A_j \le x_l < B_j \\ j \text{ w.p. } 1 & \text{if } B_j \le x_l < C_{j+1} \end{cases} \tag{9}$$

where w.p. stands for “with probability”.

When the feature x_l belongs the range $[A_i,B_i[$, then it has two quantization possibilities:

(1) $q(x_l) = j-1$ if $(N/2r)\int_{x_l}^{B_j} p_X(t)\,dt > (N/2r)\int_{A_j}^{x_l} p_X(t)\,dt$ and
(2) $q(x_l) = j$ if $(N/2r)\int_{A_j}^{x_l} p_X(t)\,dt > (N/2r)\int_{x_l}^{B_j} p_X(t)\,dt$.

The discrete scheme of *adaptive quantization* was recently developed by [10] to make it practically applicable.

3.3. Theoretical analysis of the quantization problem

In this section, we statically analyze the behavior of the extracted features under additive Gaussian noise, as well as the probability of 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 additive noise and that may affect an image [33].

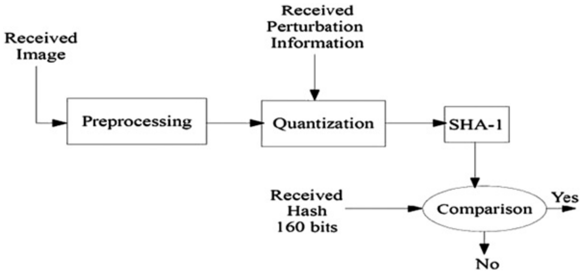


Fig. 7. Image verification module with quantization in the Fawad et al. scheme [1].

To theoretically assess the influence of additive Gaussian noise whose 0-mean and standard deviation σ on a uniform distribution of features within a limited 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

$$P_\rho(x) = \begin{cases} \dfrac{1}{\rho} & \text{for } x \in [a,b] \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- 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

$$P_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \tag{11}$$

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

$$\begin{aligned} h(x) &= \int_{-\infty}^{+\infty} P_\rho(y) P_\sigma(x-y)\,dy \\ &= \frac{1}{\rho}\left(\int_{-\infty}^{x-a} P_\sigma(y)\,dy - \int_{-\infty}^{x-b} P_\sigma(y)\,dy\right) \\ &= \frac{1}{\rho}\left(\int_{-\infty}^{x-a} \frac{1}{\sigma\sqrt{2\pi}} e^{-y^2/2\sigma^2}\,dy - \int_{-\infty}^{x-b} \frac{1}{\sigma\sqrt{2\pi}} e^{-y^2/2\sigma^2}\,dy\right) \\ &= \frac{1}{2\rho}\left[erf\left(\frac{x-a}{\sqrt{2}\sigma}\right) - erf\left(\frac{x-b}{\sqrt{2}\sigma}\right)\right] \end{aligned} \tag{12}$$

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

The convolution product $h(x)$ models the behavior of the original features after adding Gaussian noise in each

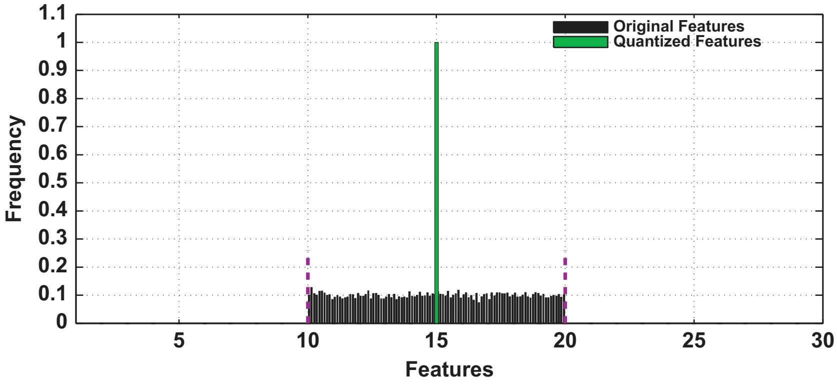


Fig. 8. About 10 000 original features uniformly distributed in one quantization interval [10,20] before quantization (black) and after uniform quantization (green), where the quantization step $Q=10$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009

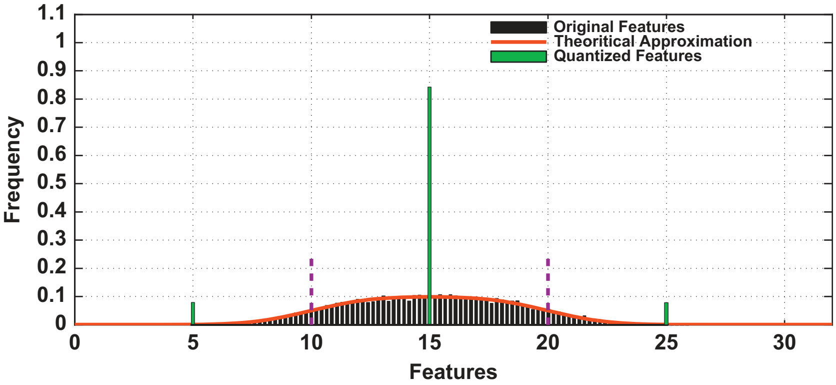


Fig. 9. About 10 000 noisy features after adding Gaussian noise whose 0-mean and standard deviation $\sigma=2$ before quantization (black) and after uniform quantization (green), where the quantization step $Q=10$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

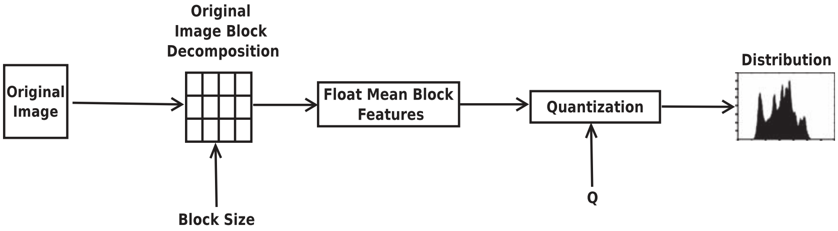


Fig. 10. Proposed quantization analysis protocol for a perceptual hashing based image block mean.

quantization interval. Fig. 8 shows a normalized uniform distribution of 10 000 features belonging in the quantization interval [10,20], before and after the quantization stage, where the quantization step $Q=10$. All of those features are quantized to the value 15, as shown in Fig. 8. Fig. 9 presents the normalized distribution of the noisy features after adding Gaussian noise with 0-mean and standard deviation $\sigma=2$. This distribution roughly coincides with the theoretical results given by Eq. (12). As shown in Fig. 9, the noisy features are spread in three quantization intervals and are quantized to three values: 5, 15 and 25. The numerical value 5 presents the quantized value in the left neighboring quantization interval that presents 8% of the total features. The numerical value 25 presents the quantized value in the right neighboring quantization interval that presents 8% of the total features. For the other experimental settings, we always have a symmetric percentage of features dropped in the left and right neighboring quantization interval. Thus, for a given set of visual features, we can estimate the percentage of features that go beyond the left and right of each quantization interval based on information about the quantization step size and the density of the tolerated noise.

4. A new perceptual hashing system taking the quantization stage into account

In this section, we describe the quantization analysis protocol for perceptual hashing based on statistical invariance of the extracted block mean features. The aim is to obtain 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 Fig. 10, the original input image I of size $N\times M$ pixels is split to non-overlapping blocks of size $q\times p$ pixels that we denote by $B_{i,j}$. Let $p_{i,j}$ be the pixels in $B_{i,j}$, where $i\in\{1,\ldots,(N/q)\}$ and $j\in\{1,\ldots,(M/p)\}$. The float mean value $m_{i,j}$ of each block $B_{i,j}$ is computed and stored in a one-dimensional vector that we denote by $V_m(k)$, where $k\in\{1,\ldots,(N/q)\times(M/q)\}$. A quantization step is the conventional way to discretize the continuous vector V_m. For a given quantization step Q, the quantized vector $V'_m(k)$ of $V_m(k)$ is given by the floor operation

$$V'_m(k)=\left\lfloor\frac{V_m(k)}{Q}\right\rfloor\times Q \tag{13}$$

where $k=\{1,\ldots,(N/q)\times(M/p)\}$.

The distribution $Dist_I$ of the quantized vector 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 respect to the original image.

The image hashing system assumes that the original image I may be sent over a network possibly consisting of untrusted nodes. During the untrusted communication, the original image could be manipulated for malicious purposes. Therefore, the received image $\bar{I}$ may undergo non-malicious operations like JPEG compression or malicious tampering. The final perceptual hash of I should be used to authenticate the received image $\bar{I}$. In the case of non-malicious operations, the original feature vector and the received one should only differ by a small distance, which makes quantization control easier, and by a large distance in the case of content-changing manipulations. The large distance difference between the original feature vector and the received one could give different results after the quantization step. Note that even if the feature vector undergoes small changes under small additive noise, it may cause false authentication of the received image $\overline{I}$ while it has to be considered similar to I. Therefore, the received image $\overline{I}$, which is formed by the original image plus Gaussian noise with 0-mean and a standard deviation σ, will undergo the same steps as the original image (Fig. 10). This process allows us to get the quantized values of the computed means of the received image block $\overline{B}_{i,j}$ containing the pixels $p'_{i,j}$. $\overline{V}'_m(k)$ can be expressed as a function of $V_m(k)$ as follows:

$$\begin{aligned}\overline{V}'_m(k) &= \frac{1}{p\times q}\sum_{i=1}^{p}\sum_{j=1}^{q}p'_{i,j}\\ &= \frac{1}{p\times q}\sum_{i=1}^{p}\sum_{j=1}^{q}(p_{i,j}+n_{i,j})\\ &= \frac{1}{p\times q}\sum_{i=1}^{p}\sum_{j=1}^{q}p_{i,j}+\frac{1}{p\times q}\sum_{i=1}^{p}\sum_{j=1}^{q}n_{i,j}\\ &= V_m(k)+\frac{1}{p\times q}\sum_{i=1}^{p}\sum_{j=1}^{q}n_{i,j} \qquad (14)\end{aligned}$$

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

The term $1/(p\times q)\sum_{i=1}^{p}\sum_{j=1}^{q}n_{i,j}$ is a Gaussian distribution with 0-mean and standard deviation $\sigma/\sqrt{p\times q}$. Thus, Eq. (14) can be written as follows:

$$\overline{V}'_m(k)=V_m(k)+\frac{\sqrt{p\times q}}{\sigma\sqrt{2\pi}}e^{-(p\times q\times k^2)/2\sigma^2} \qquad (15)$$

The comparison between $Dist_I$ and $Dist_{\overline{I}}$ allows us to get information about the percentage of stable features that have not dropped out the quantization interval after Gaussian noise addition, the percentage of the features that moved to the left neighboring quantization interval and the percentage of the features that moved to the right neighboring quantization interval. This information on the feature behavior is very useful, it allows us to take into account the percentage of the stable features that resist non-malicious operations, simulated by additive Gaussian noise. It allows us to control the parameters of the size to split the image into blocks and quantization step size to achieve an aimed level of the image hashing system robustness against a given additive noise level. Features selected from the stable features will then be hashed in the *crypto-compression* stage (Fig. 1). The crypto-compression stage is achieved by the cryptographic hash function SHA-1 generating a final hash of 160-bits with a high level of security.

5. Proposed perceptual hashing scheme robust to the quantization stage

In this section, we present our proposed perceptual hashing scheme that is robust to the quantization stage. Based on a theoretical study of the quantization problem in perceptual hashing systems, we propose to add new modules to the standard perceptual hashing system presented in Fig. 1. The block diagram of the hash generation module is shown in Fig. 11. Various steps involved in the hash generation process are as follows:

1. Let the input image be represented by I of dimension $N\times M$ pixels. Image I is split into non-overlapping blocks of dimension $q\times p$ pixels. This gives a total of $(N/q)\times(M/p)$ blocks. Each block is represented by B_i, where $i=1,\ldots,(N/q)\times(M/p)$.
2. Let $B_i(x_k,y_k)$ represent the gray value of a pixel at spatial location (x_k,y_k) in the block B_i, where $k=1,\ldots,q\times p$. Let the mean of each block be represented by m_i, where i is the block index. Each m_i is calculated as follows:

$$m_i=\frac{1}{q\times p}\sum_{k=1}^{q\times p}B_i(x_k,y_k) \qquad (16)$$

All of the computed continuous means m_i present features extracted from the transformed image in the *feature extraction* stage. Thus, they should be quantized during the *quantization* stage to form the quantized intermediate perceptual hash vector with a specific

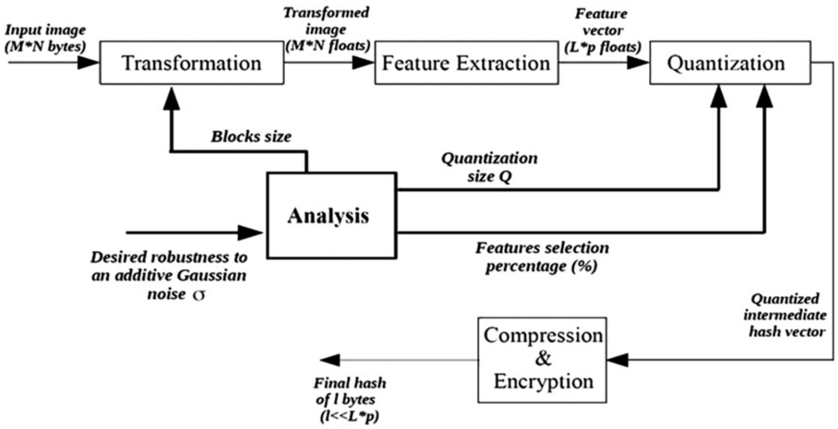

Input image
(M*N bytes)
Transformation
Transformed image (M*N floats)
Feature Extraction
Feature vector (L*p floats)
Quantization
Blocks size
Quantization size Q
Analysis
Desired robustness to an additive Gaussian noise σ
Features selection percentage (%)
Quantized intermediate hash vector
Compression & Encryption
Final hash of l bytes (l<<L*p)

Fig. 11. Proposed perceptual hashing scheme robust to the quantization stage.

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009

quantization step Q. The uniform quantization technique is used in our scheme. Let m_i' be the element of the quantized intermediate perceptual hash vector of a specific index i. m_i' is calculated as follows:

$$m_i' = \left[\frac{m_i}{Q}\right] \times Q \tag{17}$$

3. The sender determines the information about the desired robustness to the additive Gaussian noise, σ. Based on this information, the *analysis* module gives the appropriate percentage of the extracted features that must be selected for a chosen quantization step size and image block decomposition size. For the proposed method, the features are randomly selected taking into account the desired robustness.
4. The quantized intermediate perceptual hash vector is compressed and encrypted by the cryptographic hash function SHA-1. Consequently, the obtained final perceptual hash is 160 bits in size.

6. Experimental results

In the experiments of the proposed perceptual hashing scheme, the grayscale images are split into blocks

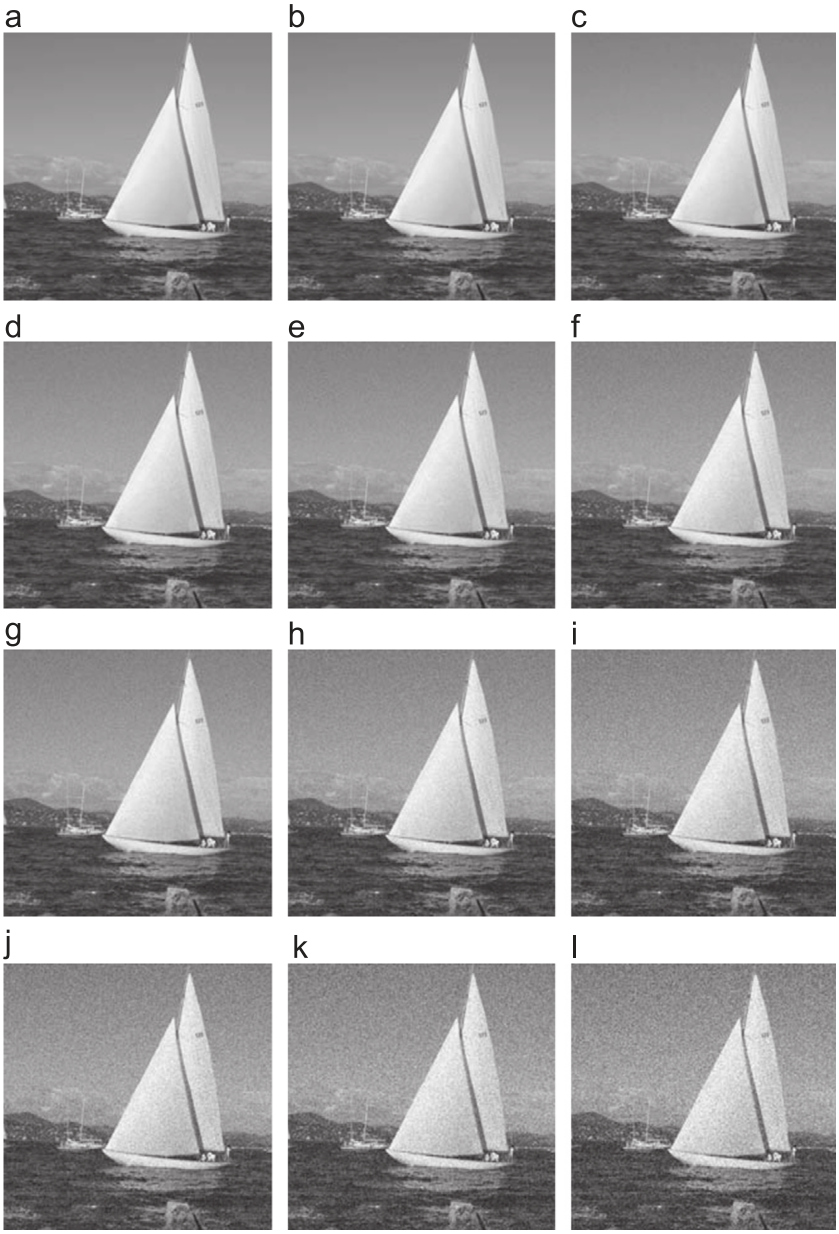


Fig. 12. Original image and noisy versions with different additive Gaussian noise parameterized with different standard deviations σ: (a) original image, (b) $\sigma = 1$, (c) $\sigma = 5$, (d) $\sigma = 10$, (e) $\sigma = 11$, (f) $\sigma = 14$, (g) $\sigma = 15$, (h) $\sigma = 20$, (i) $\sigma = 25$, (j) $\sigma = 30$, (k) $\sigma = 35$ and (l) $\sigma = 40$.

of size: 4×4, 8×8 and 16×16 pixels. The extracted features are continuous means of the image blocks of different sizes. Then the continuous block means 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 additive Gaussian noise levels. The experiments were carried out on a database of 100 grayscale images of size 512×512. Fig. 12 shows an example of an original image of size 512×512 and their noisy versions with many additive Gaussian noise levels 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 the Gaussian noise density. After the similarity evaluation presented in Section 6.1, in Section 6.2 we develop the selection of stable features. In Section 6.3, we present the robustness evaluation of our proposed method, and finally, in Section 6.4, we compare our scheme with other methods.

6.1. Similarity evaluation

An evaluation of the perceptual similarity between the original and the modified versions can be based on the perceptual aspect provided by the human visual system (HVS), on the method of the Structural SIMilarity (SSIM) [34], or on the peak signal to noise ratio (PSNR) method. Table 1 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 the images are classified into four categories: very similar, similar, different and very different. The changed images ranked as very similar and similar (Fig. 12(b)–(e)) must have the same perceptual hash of the original image denoted by $h(I_{ident})$. In the case of $\sigma=1$, the noisy image remains visually the same as the original image and it has high SSIM ($SSIM=0.997$) and PSNR ($PSNR=47.79$ db) values. In the cases of $\sigma=5$, $\sigma=10$ and $\sigma=11$, changes in the noisy images are very small and we can consider that they are still similar to the original image. The SSIM values also remain higher than 80% and the PSNR values remain higher than 27 db. Other images qualified as different or very different (Fig. 12(f)–(l)) from the original image must have a different perceptual hash denoted by $h(I_{diff})$, as presented in Table 1. When the additive Gaussian noise level increases, the noisy images are perceptually different from the original image, as shown in Fig. 12, for $\sigma=14,\dots,40$ and both the SSIM and PSNR values degrade. In order to keep a good visual content for the human visual system (HVS), we propose to set the threshold of the additive Gaussian noise at $\sigma=11$. This threshold value is justified in terms of the SSIM and PSNR values. We fixed the degradation to a SSIM value of 80% and the PSNR value at 27 db 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 conducted on 100 grayscale images of 512×512 size.

Table 1
SSIM and PSNR values for noisy images obtained by applying different standard deviation values σ of the additive Gaussian noise.

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

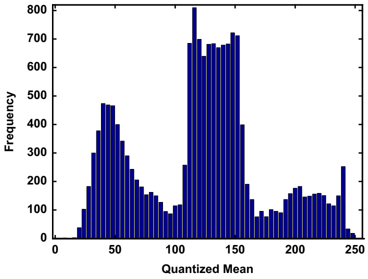
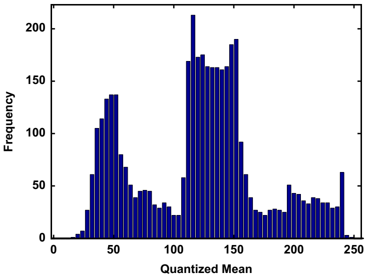
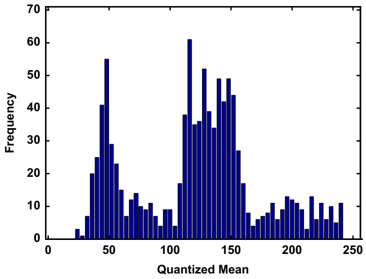
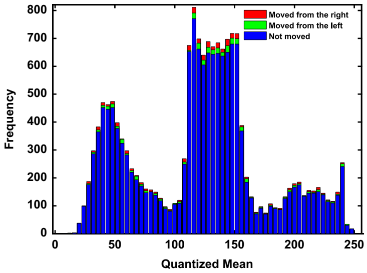
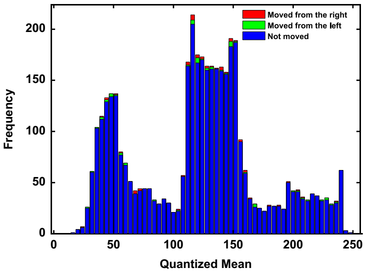
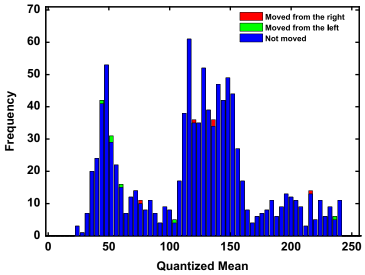
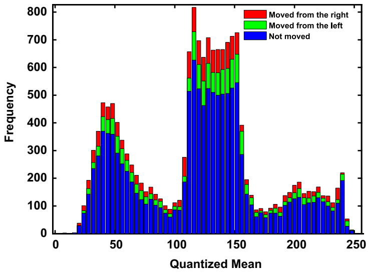
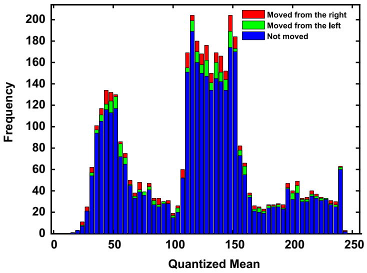
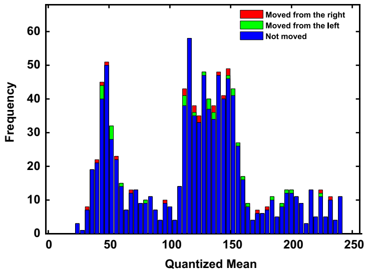
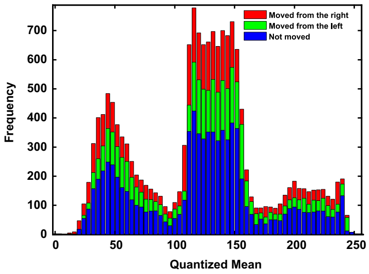
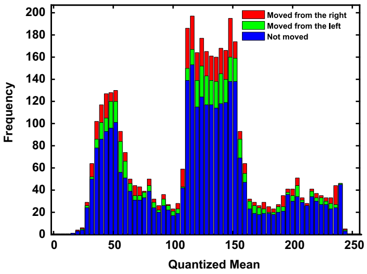
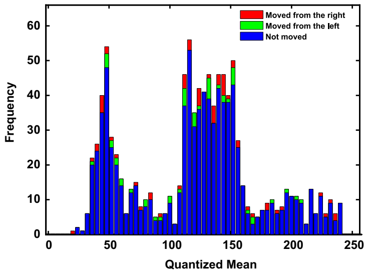
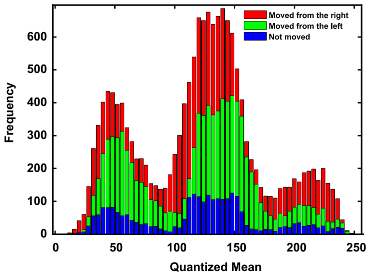
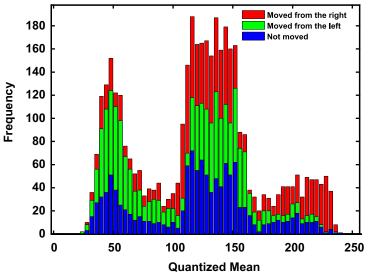
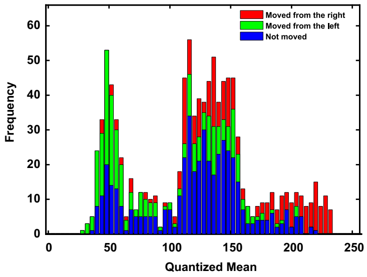
6.2. Stable feature selection

Table 2 shows the variation in mean distribution for different image block sizes and different additive Gaussian noise levels in the case of quantization step size $Q=4$ applied on the image in Fig. 12(a).

In the case of the quantization step size $Q=4$ and standard deviation $\sigma=1$ (Fig. 12(b)) (Table 2), we observe that unstable mean block features decrease when we increase the block size. We also note that the percentage of stable mean block features is significant even in the case of a block size of 4×4 (Table 3). When the standard deviation of the additive Gaussian noise increases (case of $\sigma=5$ shown in Table 2) while keeping the visual contents of the noisy image the same as the original image (Fig. 12(a)), the percentage of the stable mean block features decreases compared to the case when $\sigma=1$. When the visual contents of the noisy image changes, as shown in Fig. 12(l) in the case of $\sigma=40$, we observe that some of mean block features remain stable for all the block sizes that we tested, as shown in Table 2.

The obtained results shown in Table 3, for several quantization size values, present the percentage of features that have not moved and remain stable under different additive Gaussian noise levels and also those that drop from the left neighboring quantization interval or from the right neighboring quantization interval for each image block size. As noted in Table 3, the percentage of stable features that remain stable after adding Gaussian noise decreases when Gaussian noise level increases. For the same additive Gaussian noise level, the percentage of stable features increases when the image block size increases. Thus, if we set the quantization size at $Q=1$, we can take into account the percentage of stable features that withstand a tolerable additive Gaussian noise level for a given block image decomposition size. In the case of an image block size of 4×4 and $\sigma=5$, we have to take into account the maximum percentage of stable features $\approx30\%$, and if the block size is 8×8, we take the maximum percentage of stable features $\approx54\%$ into account. The highest percentage of stable features $\approx77\%$ can be taken if we apply a 16×16 block size in the preprocessing image treatment. We tested our experiment on a large

Table 2
Variation in the mean distribution for different image block sizes and different additive Gaussian noise levels in the case of quantization step size Q=4.

σ	Block size=4 × 4	Block size=8 × 8	Block size=16 × 16
0			
1			
5			
11			
40			

database of grayscale images of 512 × 512 size and we observed that the feature stability values presented in Table 3 can be roughly obtained for other images under the same image block decomposition settings, and additive Gaussian noise level, we also noted that the percentages of features that moved from the left and those that

Table 3
Numerical results for different additive Gaussian noise levels and image block sizes in the case of the quantization step sizes $Q=1$, $Q=4$ and $Q=16$.

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

moved from the right are approximately equal, which coincides with the theoretical study presented in Section 3.3. These numerical values are obtained for the grayscale image shown in Fig. 12(a) and are almost approximately the same for any other grayscale image.

Based on the numerical results presented in Table 3, Fig. 13 shows the percentage of features that remain stable against the additive Gaussian noise for different image block decomposition sizes. Based on the visual human system (HVS), we can determine the desired robustness to the tolerated additive Gaussian noise given by σ, and then set the parameters of the quantization step and the size of the image block decomposition to get the appropriate percentage of stable features.

6.3. *Robustness evaluation*

6.3.1. *Gaussian noise*

In previous works [1,31], it was found that the perceptual signatures generated from LL-subband coefficients are sensitive to additive Gaussian noise even if it is small and insignificant. Consequently, the generated perceptual signatures are unstable due to this instability of some coefficients that are close to the quantization boundaries. Those unstable coefficients have a high probability of changing their states (quantization interval) after the Gaussian noise addition.

In this section, we present an experimental comparison between the new proposed method, presented in this paper, and our previous work [31] in order to show that robustness improvements by adopting the new proposed perceptual hashing method are increased.

Consider the original squirrel image shown in Fig. 14(a) of 512×512 pixel size. In Fig. 14(b), a slightly Gaussian noisy version of the squirrel image is shown in which the additive Gaussian noise with $\sigma=5$ is perceptually insignificant. While in Fig. 14(b), the additive Gaussian noise of a standard deviation $\sigma=15$ is more significant. For a good perceptual hashing robustness and discrimination level, we must have: $H(I)=H(I')$ and $H(I)\neq H(I'')$.

By the approach proposed in this paper, we seek to improve the perceptual hashing robustness. For that purpose, I and I' are transformed in the DWT domain. Then we take the 4th LL-subband original and noisy coefficients that are stored in matrix C_o and C_n, respectively. C_o and C_n of 32×32 size are split into non-overlapping blocks of dimension 4×4. This gives a total of 8×8 blocks in each matrix. Then the float mean value of each block is computed and stored in a one-dimensional vector V_m for the original image I, and V'_m for the Gaussian noisy image I'. V_m and V'_m present the continuous intermediate hash vectors of I and I', respectively, that contain 64 float elements. Fig. 15(a) and (b) shows distributions of the continuous intermediate hash vectors V_m and V'_m, respectively. Note that after the Gaussian noise addition to the original image, the distribution of continuous intermediate hash changes and later causes false quantization. To get the intermediate binary hash, we propose to quantize the continuous intermediate hash by a quantization step size $Q=16$. Let $\overline{V}_m$ and $\overline{V}'_m$ denote the intermediate binary hashes of V_m and V'_m, respectively. Fig. 15(c) shows the variation in the quantized features after the Gaussian noise addition with $\sigma=5$. After applying the cryptographic hash function SHA-1, we get two different perceptual hashes of 160-bits for two images classified as perceptually the same

$$H(I)=En(\overline{V}_m)$$
$$=A6D7131C\ B29A6861\ 2B8F5EBF\ 1D656E06\ AF203423$$

$$H(I')=En(\overline{V}'_m)$$
$$=B0B69CED\ 2E6B6B3\ 929C451C\ C6B2E6A0\ 774DC15F$$

While we always get the natural result for perceptually different images I and I''

$$H(I'')=En(\overline{V}''_m)$$
$$=F5B6F749\ 469964B\ 905197DA\ C4F49696\ BB223F89$$
$$\neq H(I)$$

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009

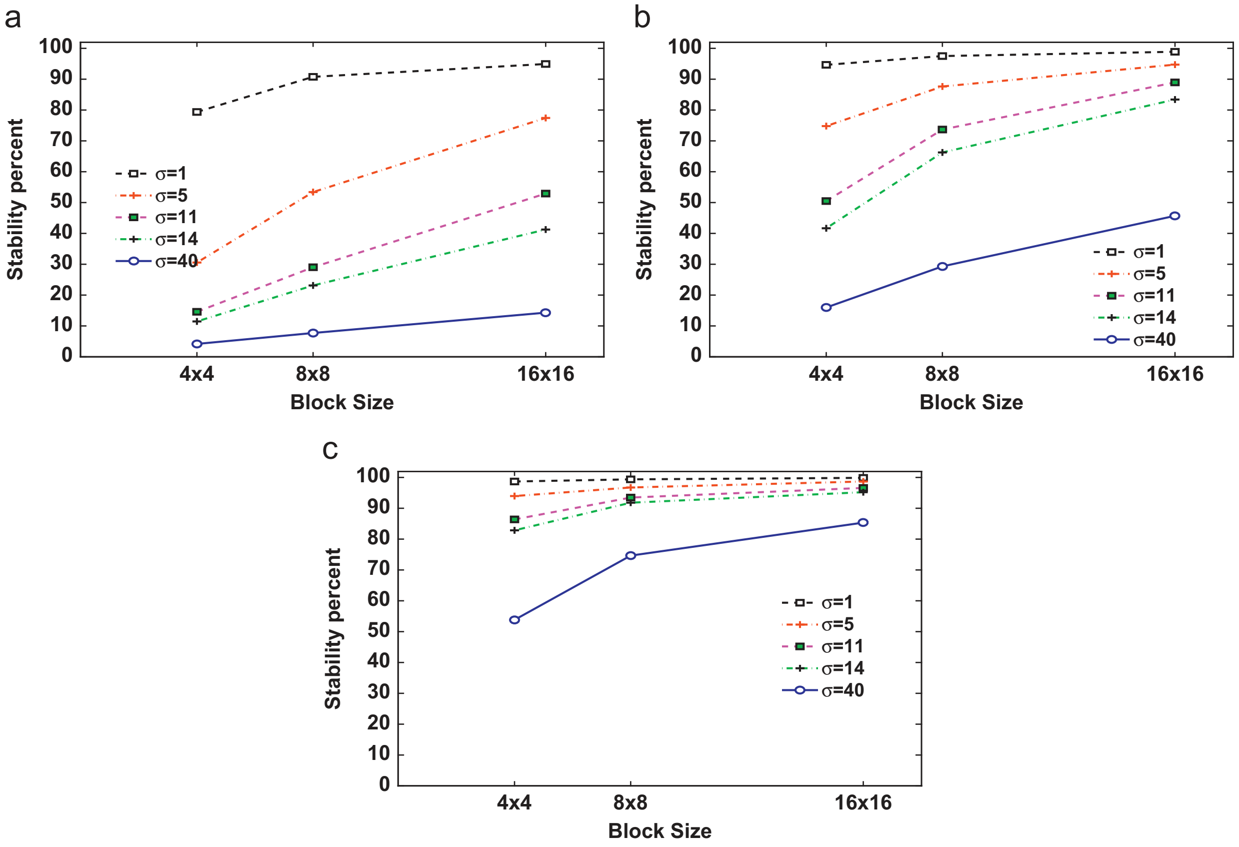


Fig. 13. Stability percentage of mean features for a set quantization step size for different block sizes: (a) Case of quantization step size $Q=1$, (b) Case of quantization step size $Q=4$ and (c) Case of quantization step size $Q=16$.

To avoid such a case, we propose to make random selections from the intermediate binary hash $\overline{V}'_m$ for several times. That allows us to minimize the probability of selecting false quantized features. In each random selection, we randomly select a range of perceptual features and then encrypt them using the advanced encryption standard (AES). Based on the statistical results presented in Table 3, we can positively authenticate the image or not.

In our experiments, we made five random selections of 10 features from the total of 64 features. Referring to the statistical results presented in Table 3, in case of $Q=16$, $block\ size = 4\times4$ and $\sigma=5$, we have statically about 93.95% of stable features. Thus, the probability of a false feature quantization $\simeq 0.1$. This information is very useful, it allow us to know *a priori* the number of stable features. Therefore, in each random selection of 10 features, we have around one feature that might be false quantized as experimentally justified in Table 4. In each previously random selection, we made five random sub-selections of five features. In each sub-selection, we apply the AES algorithm on the five randomly selected features and record only the 8 first bits of the AES output given in hexadecimal format. Table 4 summaries the obtained results.

As shown in Table 4, we get 15 correct perceptual signatures from all the 25 random sub-selections of five features. If we increase the number of random selections and/or the random sub-selection, we get a high number of perceptual signatures that we denote δ. Note that $\delta \simeq$ (*Number of random selections* $\times$ *Number of random sub-selection*)/2, for perceptually similar images I and I_{ident}, and $\delta \simeq 0$, for perceptually different images I and I_{diff}. Based on the threshold δ, we can authenticate, positively or not, the received image.

6.3.2. JPEG compression

Fig. 16(a) and (b) shows the JPEG compressed images of the original squirrel image shown in Fig. 14(a) with quality factors $QF=70\%$ and $QF=10\%$, respectively. In the case of $QF=70\%$, the changes in the JPEG compressed image $I_{QF=70}$ are perceptually insignificant. While they are more significant in $I_{QF=10}$ in case of $QF=10\%$. For a robust perceptual image hashing system, we should have: $H(I)=H(I_{QF=70})$ and $H(I)\neq H(I_{QF=10})$.

With a conventional perceptual image hashing method, we cannot positively authenticate the image $I_{QF=70}$ due to the quantization problem. Fig. 17 shows the variation in the quantized feature after applying JPEG compression with $QF=70\%$ on the original image, where we observe some unstable features that dropped to the neighboring quantization intervals. The features are the image mean blocks of 16×16 size quantized by a quantization step of size $Q=16$. As done in Section 6.3.1, we

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009

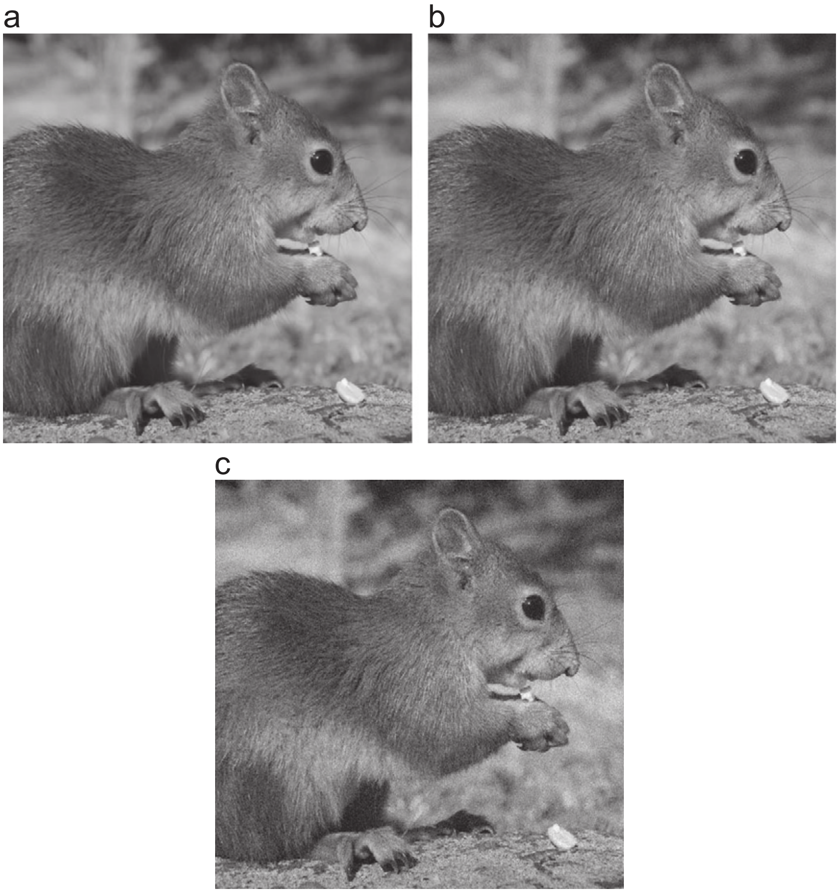


Fig. 14. Original squirrel image and its Gaussian noisy versions: (a) Original squirrel image I, (b) Gaussian noisy ($\sigma = 5$) squirrel image I' and (c) Gaussian noisy ($\sigma = 15$) squirrel image I''.

propose to make random selections from the intermediate binary hash for several times. In each random selection, we randomly select a range of perceptual features and then encrypt and compress them. The goal of these random selections is to minimize the probability of selecting false quantized features. With the proposed approach, based on the threshold δ, we can authenticate the received image even after JPEG compression.

6.3.3. *Low-pass filtering*

Fig. 18(a) and (b) shows the filtered images of the original squirrel image of Fig. 14(a) with a low-pass filter of size [3,3] and [10,10], respectively. In the case of a low-pass filter of size [3,3], the changes in the filtered image $I_{[3,3]}$ are perceptually insignificant. While they are more significant in $I_{[10,10]}$ in case of a low-pass filter of size [10,10]. For a robust perceptual image hashing system, we should have: $H(I) = H(I_{[3,3]})$ and $H(I) \neq H(I_{[10,10]})$.

As shown in Fig. 19, the low-pass filter of size [3,3] causes errors in the quantization step while keeping the visual content of the filtered image (Fig. 18(a)) the same as the original one (Fig. 14(a)). Making random selections allows us to minimize the probability of selecting false quantized features, as detailed in Section 6.3.1. In conclusion, with the proposed approach, based on the threshold δ, we can also authenticate the received image even after image filtering.

6.4. *Comparison with other methods*

With our proposed method, to authenticate the received image, the sender determines the threshold δ and sends the perceptual hash of 160 bits. The proper selection of δ is very important as it defines the boundary between non-malicious manipulations and malicious tampering. δ is computed from the number of random selections and the number of random sub-selections made on the binary intermediate hash. By comparison with the method proposed in [32], the sender needs to send additional information to adjust the quantization of the binary intermediate hash. This additional information has the same dimension as the extracted features and needs to be transmitted or stored beside the image and the final hash. That is too costly in storage space. Our proposed approach allows us to generate a secure perceptual hash of 160 bits and ensures the robustness without wasting the storage space. Note that using the adaptive quantization [6], instead the uniform quantization, increases the percentage of the stable features but does not solve the quantization problem. For

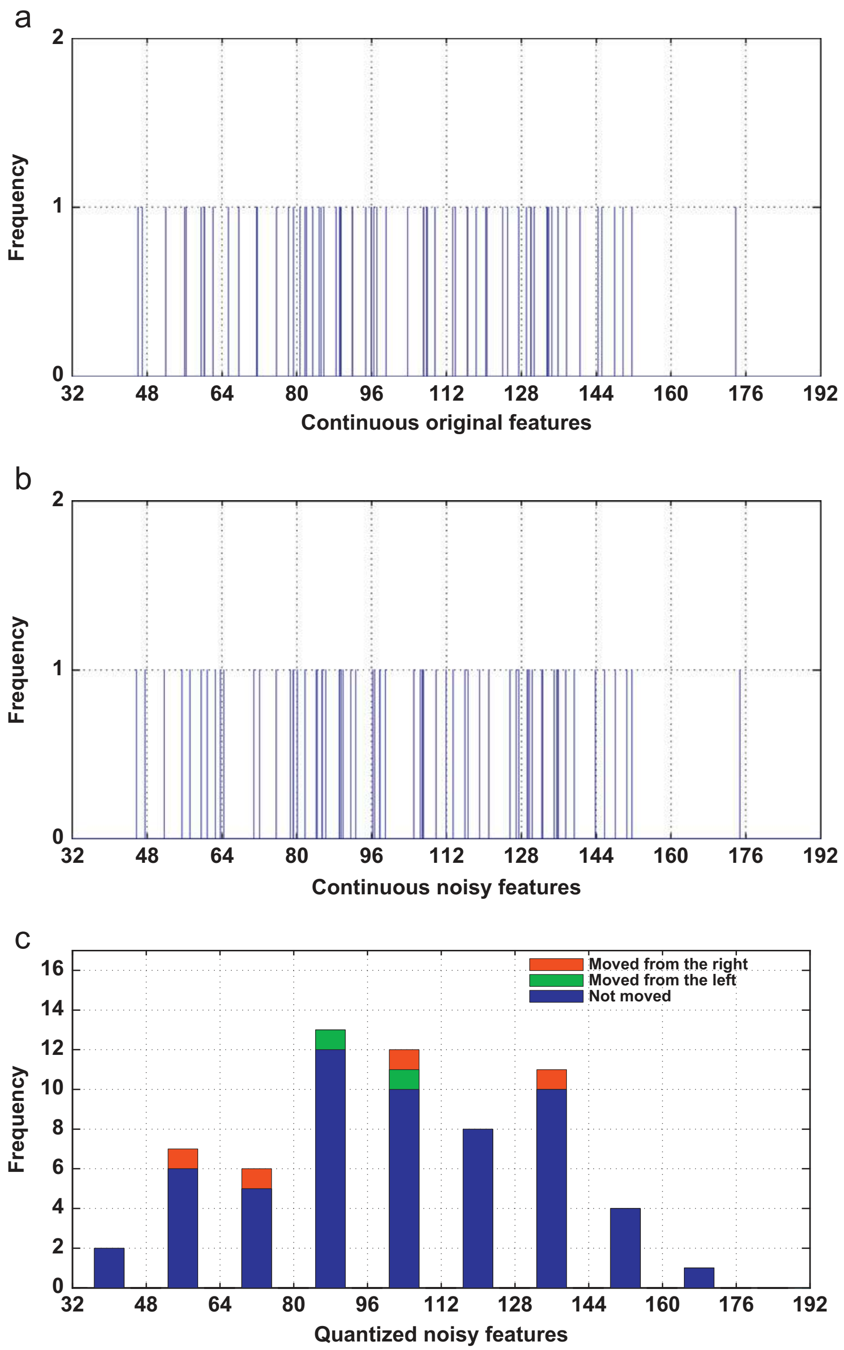


Fig. 15. Distributions of the continuous intermediate hash vectors V_m, V'_m and the binary intermediate hash vector $\overline{V}'_m$: (a) Distribution of the continuous intermediate hash vector V_m of the original image, (b) Distribution of the continuous intermediate hash vector V'_m of the Gaussian noisy image ($\sigma = 5$) and (c) Distribution of the binary intermediate hash vector $\overline{V}'_m$ of the Gaussian noisy image ($\sigma = 5$).

example, in case of a block size of 8 × 8 and $\sigma = 11$ (Table 3), we have $\approx 93.5\%$ of stable features when uniform quantization of step size $Q{=}16$ is applied. When using adaptive quantization, the percentage of stable features increases from $\approx 93.5\%$ to $\approx 96\%$. This kind of quantization is used when some measures like the bit error rate (BER), the Hamming distance and the peak of cross correlation (PCC) are used to compute distances and similarities between final perceptual hashes. Table 5 compares our approach with [32] and [6].

7. Conclusion

Robustness and security are the most important requirements for a perceptual hashing system. In this paper, we have addressed the theoretical aspects of the quantization

Table 4
Random selection of perceptual features and their AES encryption.

Random selection of 10 features	Number of stable features	Random sub-selection of five features	Compression & encryption of the original features	Compression & encryption of the noisy features
1	10	1	**44**	**44**
		2	**54**	**54**
		3	**09**	**09**
		4	***EA***	***EA***
		5	**8*D***	**8*D***
2	9	1	*F*8	32
		2	***BD***	***BD***
		3	*CA*	*FF*
		4	38	29
		5	**22**	**22**
3	9	1	***EA***	***EA***
		2	*A*6	4*F*
		3	***BE***	***BE***
		4	***F*6**	***F*6**
		5	48	19
4	8	1	*B*6	1*B*
		2	50	95
		3	**1*F***	**1*F***
		4	***99***	***99***
		5	*CD*	50
5	9	1	**02**	**02**
		2	***F*0**	***F*0**
		3	3*E*	96
		4	**36**	**36**
		5	97	85

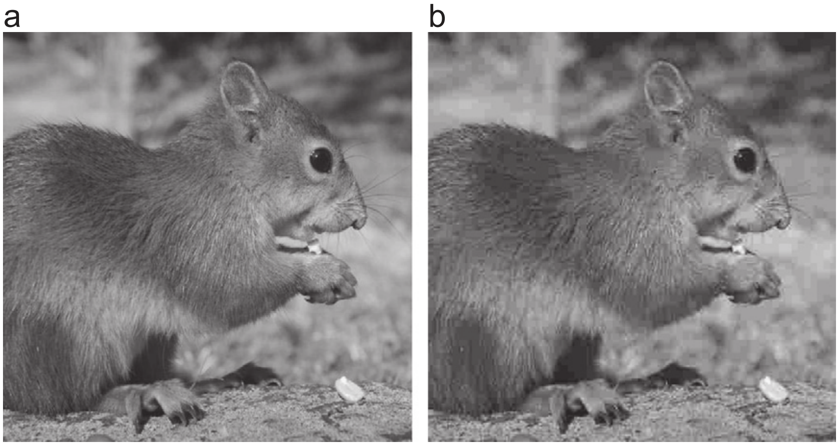


Fig. 16. JPEG compressed images of the squirrel image in Fig. 14(a): (a) JPEG compressed ($QF=70\%$) squirrel image $I_{QF=70}$ and (b) JPEG compressed ($QF=10\%$) squirrel image $I_{QF=10}$.

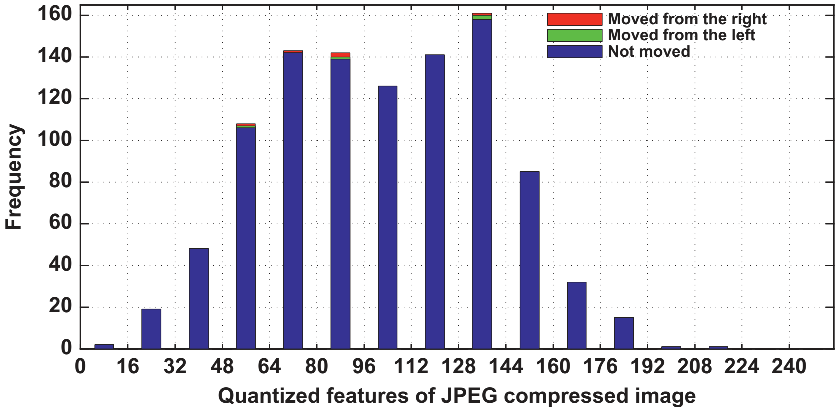


Fig. 17. Distributions of the binary intermediate hash vector of the JPEG compressed ($QF=70\%$) image.

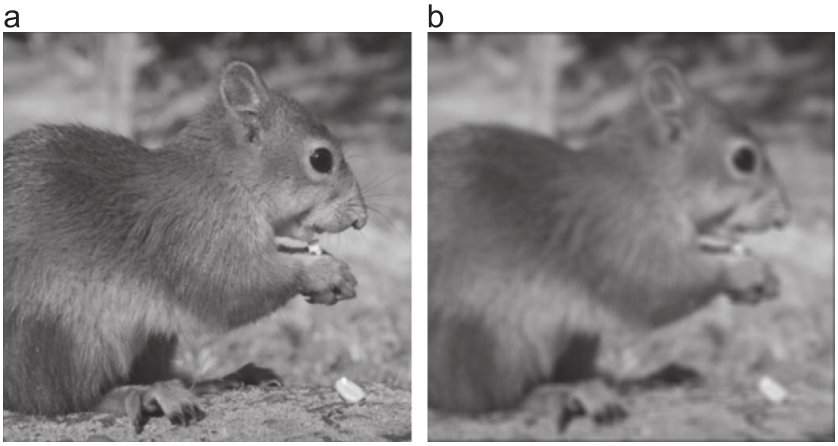


Fig. 18. Filtered images of the squirrel image in Fig. 14(a): (a) Filtered image $I_{[3,3]}$ with a low-pass filter of size [3,3] and (b) Filtered image $I_{[10,10]}$ with a low-pass filter of size [10,10].

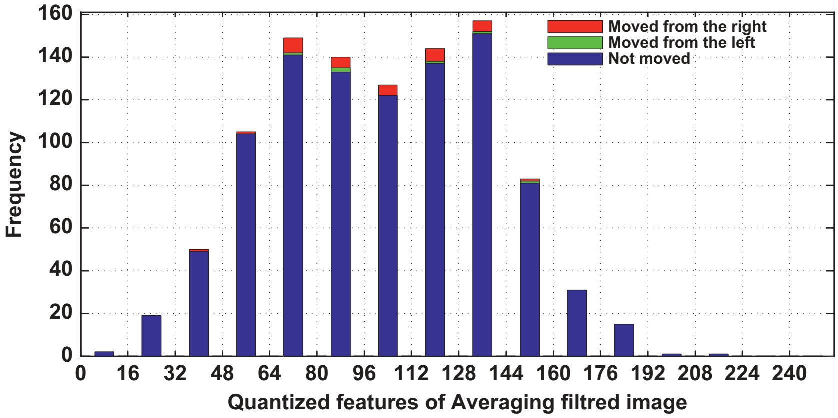


Fig. 19. Distributions of the binary intermediate hash vector of the filtered (low-pass filter of size [3,3]) image $I_{[3,3]}$.

Table 5
Comparison with other methods: [32] and [6].

Method	% of stable features	Problem of quantization is solved?	Secure perceptual hash?	Size of perceptual hash	Necessary additional information	
Adaptive quantization in [6]	≈96%	No	No	Depends of size of the extracted features	Threshold for Hamming distances	1 byte
Method in [32]	100%	Yes	Yes	160 bits	Additional vector to adjust the quantization of the binary intermediate hash	Same size of the extracted features
Our proposed approach	≈93.5%	No	Yes	160 bits	Threshold δ	1 byte

stage in image hashing systems. We presented a theoretical analysis describing the behavior of extracted features during the quantization stage. In the presented analysis, we simulated manipulations that the original image may undergo by Gaussian noise addition. Based on our proposed study, we proposed a new perceptual hashing method that takes the quantization stage into account. We tested the presented scheme by several experiments to demonstrate the effectiveness of the proposed theoretical model while giving a practical analysis for robust perceptual hashing. The presented scheme is applied on image hashing based on statistical invariance in the mean block features. The obtained results confirm our proposed theoretical analysis of the quantization problem. The same study can be extended for other types of features in block-based image hashing schemes like DCT domain features or DWT domain features.

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009

References

[1] A. Fawad, M.Y. Siyal, V.U. Abbas, A secure and robust hash-based scheme for image authentication, Signal Processing 90 (May) (2010) 1456–1470.

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

[3] R.L. Rivest, The MD5 message-digest algorithm, Technical Report RFC 1321, Internet Engineering Task Force (IETF), April 1992.

[4] NIST. 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, October 2008.

[5] A.J. Menezes, S.A. Vanstone, P.C.V. Oorschot, Handbook of Applied Cryptography, 1st ed. CRC Press, Inc., Boca Raton, FL, USA, 1996.

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

[7] M.K. Mihçak, R. Venkatesan, New iterative geometric methods for robust perceptual image hashing, in: Digital Rights Management Workshop, 2001, pp. 13–21.

[8] A. Kerckhoffs, La cryptographie militaire, Journal des sciences militaires 9 (1) (1883) 5–38.

[9] V. Monga, Perceptually Based Methods for Robust Image Hashing, Ph.D. Dissertation, University of Texas at Austin, 2005.

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

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

[12] F. Khelifi, J. Jiang, Perceptual image hashing based on virtual watermark detection, IEEE Transactions on Image Processing 19 (April) (2010) 981–994.

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

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

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

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

[17] J. Fridrich, M. Goljan, Robust hash functions for digital watermarking, in: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'00), IEEE Computer Society, Washington, DC, USA, March 27–29 2000, pp. 178–183.

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

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

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

[21] S.K. Bhattacharjee, M. Kutter, Compression tolerant image authentication, in: Proceedings of the IEEE International Conference on Image Processing, 1998, pp. 435–439.

[22] J. Fridrich, Visual hash for oblivious watermarking, in: SPIE Photonic West Electronic Imaging 2000, Security and Watermarking of Multimedia Contents, vol. 3971, SPIE, San Jose, California, January 2000, pp. 286–294.

[23] Y. Lei, Y. Wang, J. Huang, Robust image hash in Radon transform domain for authentication, Signal Processing: Image Communication 26 (July) (2011) 280–288.

[24] F. Lefebvre, B. Macq, J.D. Legat, Rash: radon soft hash algorithm, in: Proceedings of the European Signal Processing Conference (EUSIPCO'02), Toulouse, France, September 2002.

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

[26] X.C. Guo, D. Hatzinakos, Content based image hashing via wavelet and Radon transform, in: Proceedings of the Multimedia 8th Pacific Rim Conference on Advances in Multimedia Information Processing, PCM'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 755–764.

[27] F. Liu, L.-M. Cheng, Perceptual image hashing via wave atom transform, in: 10th International Workshop on Digital-forensics and Watermarking (IWDW11), Atlantic City, New Jersey, USA, Octorber 2011.

[28] S. Xiang, H.-J. Kim, J. Huang, Histogram-based image hashing scheme robust against geometric deformations, in: Proceedings of the 9th Workshop on Multimedia & Security, ACM, New York, NY, USA, 2007, pp. 121–128.

[29] Y. Ou, C. Sur, K.H. Rhee, An improved histogram-based image hashing scheme using *k*-means segmentation, in: The Fourth Joint Workshop on Information Security (JWIS 2009), Kaohsiung, Taiwan, August 2009.

[30] S. Xiang, H.-J. Kim, Histogram-based image hashing for searching content-preserving copies, Transactions on Data Hiding and Multimedia Security 6 (2011) 83–108.

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

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

[33] A. Hadmi, W. Puech, B. AitEssaid, A. Aitouahman, Statistical analysis of the quantization stage of robust perceptual image hashing, in: IEEE 3rd European Workshop on Visual Information Processing (EUVIP'11), Paris, France, July 2011, pp. 274–279.

[34] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

Please cite this article as: A. Hadmi, et al., A robust and secure perceptual hashing system based on a quantization step analysis, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2012.11.009