Robust Image Hashing Based on Random Gabor Filtering and Dithered Lattice Vector Quantization

Yuenan Li, Zheming Lu, *Senior Member, IEEE*, Ce Zhu, *Senior Member, IEEE*, and Xiamu Niu, *Member, IEEE*

***Abstract*—In this paper, we propose a robust-hash function based on random Gabor filtering and dithered lattice vector quantization (LVQ). In order to enhance the robustness against rotation manipulations, the conventional Gabor filter is adapted to be rotation invariant, and the rotation-invariant filter is randomized to facilitate secure feature extraction. Particularly, a novel dithered-LVQ-based quantization scheme is proposed for robust hashing. The dithered-LVQ-based quantization scheme is well suited for robust hashing with several desirable features, including better tradeoff between robustness and discrimination, higher randomness, and secrecy, which are validated by analytical and experimental results. The performance of the proposed hashing algorithm is evaluated over a test image database under various content-preserving manipulations. The proposed hashing algorithm shows superior robustness and discrimination performance compared with other state-of-the-art algorithms, particularly in the robustness against rotations (of large degrees).**

***Index Terms*—Dithered lattice vector quantization (LVQ), feature extraction, image authentication, robust hashing.**

I. Introduction

In cryptography, hash function serves as an effective tool for message authentication. Hash function (such as MD5 and SHA-1) defines a mapping from an arbitrary-length message to a short digest (i.e., hash string) [1]. In order to accomplish authentication, the avalanche effect is a desired property of cryptographic hash function, such that a single bit modification on the message can lead to significant changes in hash string. However, the excessive alertness of cryptographic hash function limits its applications to multimedia authentication. Different from text message authentication, multimedia authentication places more emphasis on the integrity and the authenticity of the perceptual content instead of the digital representation of media data. For example, an image should be authentic after Joint Photographers Expert Group (JPEG) compression as the compressed image can still faithfully convey the original perceptual content. Therefore, it is desired that the hash function for multimedia authentication should be able to distinguish content-preserving manipulations from malicious tampering. As a consequence, robust hashing [2] that is tailored to media data has been developed. Robust-hash function for media can be viewed as a mapping from the perceptual content of the media data to hash string; thus it is also referred to as perceptual hash. As its name implies, the robust-hash function can tolerate a certain degree of modifications, as long as the perceptual content of the media data is kept intact. Robust hashing and semifragile watermarking have composed the main methodologies for content authentication. Moreover, as a compact representation of perceptual content, robust hash can also facilitate other content-based applications such as identification [3], broadcast monitoring [4], and perceptual quality evaluation [5].

The framework of most robust-hashing algorithms can be decomposed into two constituents, namely, feature extraction and quantization [6]. Feature extraction is essential to the performance of robust hashing, where stable and discriminative features are desired. Quantization is implemented on feature vectors for data volume reduction, and it can also enhance the robustness of the hash function. For security considerations, most robust-hash functions employ key-dependent feature extraction and quantization. Here, we first provide a review of the state-of-the-art in robust hashing from the aspects of feature extraction, quantization, and security. Moreover, a brief survey of some emerging novel hashing paradigms is also included.

A. Feature Extraction

Most early robust-hash functions calculate the statistics in spatial and transform domains as features. For example, Venkatesan *et al.* proposed to use the variance of wavelet coefficients as features [2]. The most important contribution of this paper is that a paradigm of key-dependent feature extraction is proposed, and the idea of random tiling has been widely used in robust hashing. Fridrich *et al.* developed a random-projection-based image hashing algorithm, and each pixel block is projected on a randomly generated matrix [7]. If the random matrix and the pixel block are viewed as vectors in high-dimensional space, the result of projection corresponds to the angle between these two vectors. As a result, the hash string to some extent can reflect the distribution of image blocks in high-dimensional space. The first video hashing algorithm proposed in [8] computes the differences between the mean values of neighboring blocks in both spatial and temporal

Manuscript received December 11, 2010; revised May 03, 2011 and July 11, 2011; accepted August 25, 2011. Date of publication October 13, 2011; date of current version March 21, 2012. This work was supported in part by the Independent Innovation Foundation of Tianjin University under Grant 150-60302011 and Grant 150-60302038, by the National Science Foundation under Grant 60832010, and by Zhejiang Provincial Natural Science Foundation under Grant R1110006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hitoshi Kiya.

Y. Li is with the School of Electronic and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: ynli@tju.edu.cn).

Z. Lu is with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China (e-mail: zheminglu@zju.edu.cn).

C. Zhu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: eczhu@ntu.edu.sg).

X. Niu is with the School of Computer Science, Harbin Institute of Technology, Harbin 150080, China (e-mail: xiamu.niu@hit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2171698

directions. Its easy implementation and excellent performance make it suitable for real-time applications such as online video identifications. The intensity histogram is exploited as the feature for robust hashing in [9] and [10].

Another category of algorithms exploit the coarse representation of the input image as features. Based on the fact that the main structures of an image are stable in content-preserving manipulations, an iterative geometric hashing algorithm was proposed in [11], where the main structures detected by iterative filtering are binarized as hash string. Moreover, the edge map is extracted and compressed to form the hash for content authentication in [12]. Some researchers proposed to calculate the matrix invariance to enhance the robustness of the hash function, such as the singular-value-decomposition (SVD)-based [13] and the nonnegative-matrix-factorization (NMF)-based [14] hashing. It has been observed in [14] that the NMF-based hashing shows excellent robustness while its misclassification rate is lower than that of the SVD-based hashing. As pointed out in [14], the reason is that NMF has the ability of learning local image features. Vector quantization has been also employed as a feature extraction scheme in [15], where the relationships between the indexes of neighboring blocks are extracted as hash bits.

Similar to the case in digital watermarking, rotation also poses great challenges to robust hashing. Therefore, great efforts have been devoted to exploit rotation-resistant features. Inspired by the research work in computer vision, some researchers developed robust-hash functions using salient points. In [6], the salient points are first extracted using the end-stopped wavelet, and the coefficients are then randomly quantized. The well-known scale-invariant feature transform descriptor was employed in robust hashing in [16]. Alternatively, the invariance in the transform domain has been also utilized for hash computation. In [17], 2-D fast Fourier transform (FFT) is implemented on the autocorrelation of Radon coefficients, and the FFT coefficients are found to be rotation invariant. Similarly, the Fourier–Mellin transform was employed in [18] for geometric-invariant feature extraction. The radial-projection-based hashing algorithm was proposed in [19], where the variances of the pixels that lie in each radial line are computed as features.

B. Quantization

As previously mentioned, quantization is another component in robust hashing. However, compared with feature extraction, the research on quantization is quite rare. For simplicity, most algorithms generate binary hash strings by comparing feature data with a threshold [4], [7], [8], which can be viewed as a two-level quantizer. In order to increase the discrimination of robust hashing, some algorithms perform finer quantizations on feature data [6], [18], [20]. Recall that robust hashing is desired to be key dependent in authentication applications; thus, the quantizer should also exhibit a certain amount of randomness. The adaptive quantizer proposed in [20] is widely adopted in robust-hashing algorithms. The quantizer is designed on the foundation of nonuniform probabilistic quantization, while it partitions a random subregion in each interval. The feature data that lie in subregions are randomly quantized to one of its neighboring indexes. It should be emphasized that the requirement for quantizer design in robust hashing is different from that in data compression. In data compression, a quantizer is expected to yield the least distortion. Nevertheless, a quantizer with low distortion conflicts with the robustness requirement in robust hashing since similar features would be mapped to different indexes in fine quantization. In fact, a tradeoff between robustness and discrimination should be made in developing the quantizer for robust hashing. Monga *et al.* proposed a cost function for feature quantization that evaluates both of the robustness and discrimination criteria [21]. However, the problem of finding the optimum solution for the cost function has been proven to be NP complete in [21]. It has been pointed out in [22] that uniform quantizers should be a suboptimum choice for robust hashing. Moreover, the simulations in [22] demonstrate that the uniform quantizer can achieve a better overall performance than the nonuniform one.

C. Security

Security is an inevitable issue in authentication applications. The security of robust hashing has attracted increasing attentions in recent years, and current research works concentrate on evaluation metrics, theoretical analysis, attacking methods, and protocols. The first information-theoretic-based metric for security evaluation in robust hashing is proposed in [18], where the differential entropy is used to measure the amount of randomness of a given robust-hash function. A secure robust-hashing algorithm should be characterized by high differential entropy so that the estimation of the hash string without knowing the secret key would be computationally tough. The unicity distance was proposed in [23] to evaluate the security of feature extraction against chosen message attacks. It is defined as the minimum number of image-hash pairs required to break a given robust-hash function. Furthermore, the recent work of Li and Roy has proven that the information-theoretic security against forgery under chosen message attacks cannot be achieved in robust hashing [24]. Nevertheless, this fact will not hamper robust hashing from being applied in content authentication, as it is still possible to design a computationally secure hash function, whose key is still computationally infeasible to be estimated. Moreover, some protocols have been proposed to enhance the tampering detection ability of robust hashing. In [25], the features extracted from sensitive regions (e.g., the plate number of a car) are involved in the key generation. In this way, some small but semantically significant tampering on sensitive regions can lead to drastic changes in the output hash string.

D. New Hashing Paradigms

Some very recent works are devoted to the development of new paradigms for robust hashing. For instance, Jin *et al.* proposed a quantum-hashing (QH) system in [26] for multimedia identification. The hash value in the QH system is represented by a qubit, and the weights of the qubit bases correspond to the uncertainties of the binary hash bits being 0 and 1. It has been reported that the QH can provide a higher amount of robustness. In [27], a virtual-watermark-detection-based robust-hashing scheme was developed. A sequence of pseudorandomly generated numbers uniformly distributed in $[-1, 1]$ are used as the virtual watermark, and the image features are considered as the host signal. Each binary hash bit is generated by detecting the existence of the given virtual watermark from the features. Moreover, a hashing system that incorporates compressive sensing and Wyner–Ziv codec was introduced in [28]. Compressive sensing is employed to project the image subblocks onto a number of random bases, and the projections are then quantized by the Wyner–Ziv encoder to output the hash string. At the receiver side, the received image and the hash bits are passed to the Wyner–Ziv decoder for tamper identification, where the received image that might have been tampered or distorted serves as the side information of the decoder.

In this paper, we present a robust-hashing algorithm based on random Gabor filtering and dithered lattice vector quantization (LVQ). The Gabor filter is attractive for feature extraction due to its outstanding robustness and discrimination. However, the conventional Gabor filter is orientation sensitive, which will lead to poor robustness against rotation. On the other hand, the deterministic Gabor filtering cannot satisfy the security requirements in authentication applications. Thus, a rotation-invariant and random Gabor filter is designed in this paper, where the conventional Gabor filter is adapted to be orientation independent and its parameters keep varying during the filtering process under the control of a secret key. Moreover, we propose to quantize feature vectors via dithered LVQ in robust hashing. Because of its random nature, congruent cells, and source-independent quantization error, the dithered LVQ can lead to excellent robustness–discrimination tradeoff and security performance, which makes it well suited for feature quantization in robust hashing. Experimental results demonstrate that the proposed hash function shows excellent statistical performance. By the virtue of the rotation-invariant filter, the robustness of the proposed algorithm against rotation manipulations is much higher than the state-of-the-art ones. It is also shown by simulation that the proposed dithered-LVQ-based quantization scheme outperforms the widely used adaptive quantizer [20] and the uniform scalar quantizer (SQ). Analytical results reveal that, compared with other quantizers, the dithered LVQ can resist a higher amount of distortion, and it can therefore achieve higher robustness. To the best of our knowledge, this is the first paper that quantifies the robustness of quantizers for robust hashing. To sum up, the main contributions of this paper are twofold. First, a random and rotation-invariant Gabor filter is designed. Second, the dithered-LVQ-based random quantizer is proposed as a novel quantization paradigm for robust hashing.

The remainder parts of this paper are organized as follows: Section II introduces the design of the random Gabor filter and its application in feature extraction. The dithered-LVQ-based hashing quantization scheme is illustrated in detail and analyzed in Section III. In Section IV, the effectiveness of the proposed hashing algorithm is demonstrated by comparative experiments. Finally, we close with conclusions in Section V.

II. Feature Extraction Using Rotation-Invariant and Random Gabor Filter

A. *Extracting Rotation-Invariant Features*

The Gabor filter have been successfully applied in various applications, such as face recognition, texture analysis, retina identification, edge detections, etc. One reason is that the Gabor filter can effectively distinguish different patterns in visual contents. Moreover, some other properties of Gabor filter also make it well suited for robust hashing, such as the high robustness against distortions [29] and the similarities to the simple cortex cells in the human visual system [30]. Nevertheless, as it will be discussed later, the filter response is sensitive to the orientations of the filter and the input image. The orientation-sensitive property can benefit some recognition and identification applications, as it is possible to characterize visual contents along any direction. However, it will result in a poor robustness of the hash function against rotation. In this paper, the conventional Gabor filter is designed to be orientation independent, based on which a rotation-invariant and random feature extraction scheme is proposed.

A Gabor filter can be obtained by modulating a complex sinusoidal plane wave with the Gaussian window. The kernel of a 2-D Gabor filter is shown in (1), where f is the frequency of the sinusoidal plane wave, φ is the counterclockwise rotation angle of the Gaussian window and the sinusoidal plane wave, γ is the spatial width of the filter parallel with the plane wave, and η is the width perpendicular to the wave. The mesh of the real part of a Gabor filter and its projection on the xy-plane are shown in Fig. 1, where the meaning of each parameter is illustrated in the projection image, i.e.,

$$\psi(x,y;\varphi,f)=\frac{f^2}{\pi\gamma\eta}e^{-\left(\frac{f^2}{\gamma^2}\hat{x}^2+\frac{f^2}{\eta^2}\hat{y}^2\right)}e^{2\pi jf\hat{x}}$$
$$\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix}. \tag{1}$$

Given image $I(x,y)$ with its center located in the origin. If we rotate it around the origin counterclockwise by a degree of ϕ, then the rotated image $I_2(x,y)$ can be written as

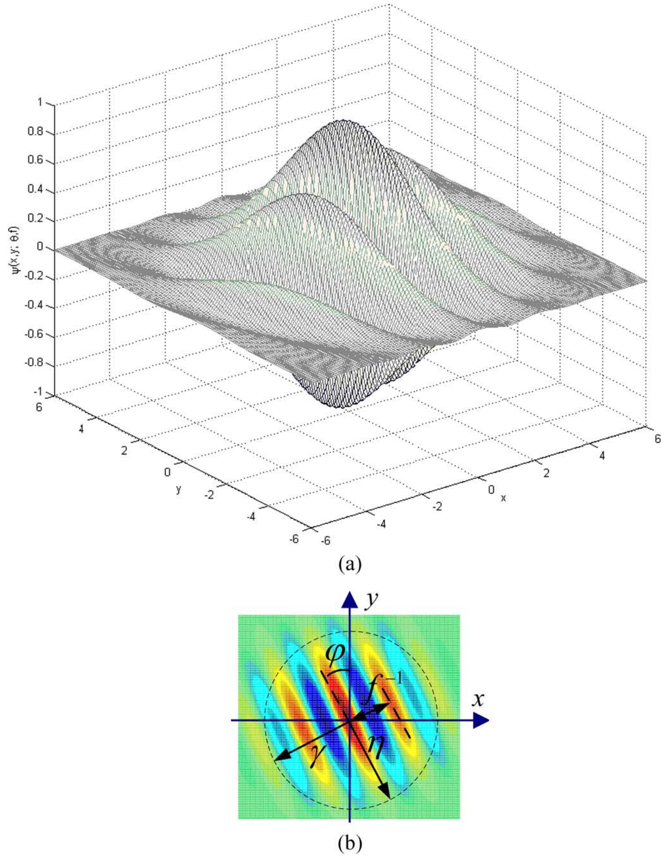
$$I_2(x,y)=I(\hat{x},\hat{y})$$
$$\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}\cos\phi & -\sin\phi\\ \sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix}. \tag{2}$$

The filter response of the rotated image is given by

$$\begin{aligned}r_2(x,y;\varphi,f)&=\psi(x,y;\varphi,f)*I_2(x,y)\\&=\iint_{-\infty}^{+\infty}\psi(x_\tau,y_\tau;\varphi,f)I_2(x-x_\tau,y-y_\tau)dx_\tau dy_\tau.\end{aligned} \tag{3}$$

Consider a Gabor filter with $\gamma=\eta$; if we rotate the coordinate systems (x,y) and (x_τ,y_τ) counterclockwise with the same angle ϕ, as shown in Fig. 2, the following result can be obtained by computing the integration over the rotated coordinate system (x'_τ,y'_τ) as

$$\begin{aligned}&r_2(x,y;\varphi,f)\\&\quad=\iint_{-\infty}^{+\infty}\psi\left(x'_\tau,y'_\tau;\varphi-\phi,f\right)I\left(x'-x'_\tau,y'-y'_\tau\right)dx'_\tau dy'_\tau\\&\quad=r(x',y';\varphi-\phi,f)\end{aligned} \tag{4}$$



(a)

(b)

Fig. 1. Example of a 2-D Gabor filter with $f = 2, \varphi = (\pi/6)$, and $\gamma = \eta = 1$: (a) Real part of the Gabor filter. (b) Projection of the filter on the xy-plane and illustrations of the parameters.

where $r(x', y'; \varphi - \phi, f)$ is the filter response of the original image. Obviously, the rotation of the input image results in a corresponding rotation of the filter response, as well as a shift of its orientation component. The Gaussian window $G(x, y) = (f^2/\pi\gamma\eta)e^{-((f^2/\gamma^2)\hat{x}^2+(f^2/\eta^2)\hat{y}^2)}$ is rotation invariant when $\gamma = \eta$. Hence, the orientation sensitive property of the Gabor filter is caused by the directional complex sinusoidal wave $s(x, y; \varphi, f) = e^{2\pi jf\hat{x}}$. The features for the hash calculation are expected to be rotation invariant; therefore, the sinusoidal wave should be isotropic in all directions. Similar to [31], we use a circularly symmetric complex sinusoidal wave instead, as $e^{2\pi jf\sqrt{x^2+y^2}}$. In this way, a rotation-invariant Gabor filter can be obtained by modulating the circularly symmetric complex sinusoidal wave with the Gaussian window, i.e.,

$$\omega(x, y; f) = \frac{f^2}{\pi\gamma^2}e^{-\frac{f^2}{\gamma^2}(x^2+y^2)}e^{2\pi jf\sqrt{x^2+y^2}}. \tag{5}$$

Now, we investigate the rotation property of the new filter, based on which the rotation-invariant feature extraction scheme is developed. Let $R(x, y; f)$ and $R_2(x, y; f)$ denote the responses of the original and rotated images under the new filter. Similar to (4), the response of the rotated image is also computed in the aforementioned rotated coordinate system (x'_τ, y'_τ). Since the new filter is rotation invariant, namely, $\omega(x_\tau, y_\tau; f) = \omega(x'_\tau, y'_\tau; f)$, it is straightforward to show that

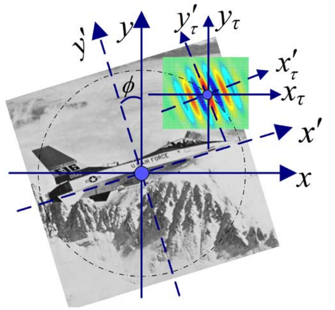


Fig. 2. Original and rotated coordinate systems.

$$\begin{aligned} R_2(x, y; f) &= \iint\limits_{-\infty}^{+\infty} \omega(x_\tau, y_\tau; f)I_2(x - x_\tau, y - y_\tau)dx_\tau dy_\tau \\ &= \iint\limits_{-\infty}^{+\infty} \omega\left(x'_\tau, y'_\tau; f\right) I\left(x' - x'_\tau, y' - y'_\tau\right) dx'_\tau dy'_\tau \\ &= R(x', y'; f). \end{aligned} \tag{6}$$

As shown in (6), the filter response of the rotated image is equal to the rotated response of the original image. If we represent the filter responses $R(x, y; f)$ and $R_2(x, y; f)$ in the polar coordinate system as $R(\rho, \theta; f)$ and $R_2(\rho, \theta; f)$, where ρ and θ denote the radial and angular coordinates of the polar coordinate system, then the rotation of the input image will lead to the shift of the angular coordinate θ of its filter response. Accordingly, (6) can be rewritten as

$$R_2(\rho, \theta; f) = R(\rho, \theta - \phi; f). \tag{7}$$

Consequently, the squared magnitude of the filter response is integrated along a circle centered at the origin with the radius of ρ as follows[1]:

$$P(\rho; f) = \int\limits_0^{2\pi} |R(\rho, \theta; f)|^2 \, d\theta. \tag{8}$$

Denote the integration result of $|R_2(\rho, \theta; f)|^2$ as $P_2(\rho; f)$; then, it is easy to identify that the integration result is rotation invariant, i.e.,

$$P_2(\rho; f) = P(\rho; f). \tag{9}$$

Up to now, we have obtained a stable feature that can be exploited for hash construction.

B. Randomization of the Rotation-Invariant Filter

As previously discussed, the feature for hash construction is expected to possess a certain amount of randomness. To facilitate secure feature extraction, a random filter is constructed

[1] It has been observed in our simulations that integrating the squared magnitude exhibits stronger robustness than integrating the magnitude itself.

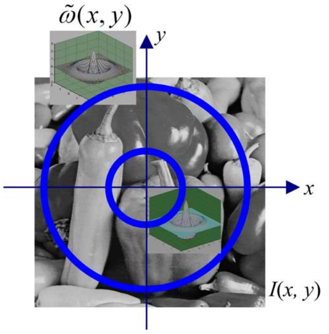


Fig. 3. Feature extraction using random filter.

based on the rotation-invariant filter. The random filter is designed to be key dependent, and its kernel keeps varying during the convolution process. In this paper, the random filter is designed by summing up Q rotation-invariant filters with random frequencies, i.e.,

$$\tilde{\omega}(x, y) = \sum_{k=1}^{Q} \tilde{\omega}_k(x, y; \tilde{f}_k). \tag{10}$$

Frequency $\tilde{f}_k$ in each filter $\tilde{\omega}_k(x, y; \tilde{f}_k)$ is randomly generated via a secret key. In (10), tildes are placed over variable and functions to emphasize their random nature. The random frequencies vary in the range of $[0, F)$. The whole frequency range is uniformly divided into Q intervals, and the kth random frequency is selected in the corresponding interval, as $\tilde{f}_k \in [((k-1)F/Q), (kF/Q))$, $k = 1, \cdots, Q$. It may be argued that using a single filter with its frequency randomly selected from the whole range $[0, F)$ can show a higher degree of randomness. However, a poor discrimination has been observed in this case. Since the feature computed under a single frequency can only reflect the information corresponding to a very narrow band of the spectrum, it may be inadequate in characterizing the input image. Moreover, from the viewpoint of energy distribution, most spatial energy of a natural image is concentrated in the low-frequency band of the spectrum; the low-frequency features are therefore prone to have much higher amplitudes than the features in other bands. Consequently, in the case where each feature is computed with a single random frequency, the relationships between the amplitudes of features would highly depend on the selection of random frequencies. As a result, even for content-distinct images, their hash strings could be very similar if their features are computed under the same set of random frequencies. This issue will be further discussed in Section IV-D with experimental results.

In the proposed algorithm, the parameters of the random filter keep varying during the calculation of the filter response. However, it is worth mentioning that the random feature extraction scheme should maintain the rotation-invariant property of the filter. Hence, the filter responses on each ring of the input image are computed via the same filter and a number of random filters are generated for different rings. As shown in Fig. 3, the features are extracted from those two rings using distinct filters whose frequencies are randomly selected.

To sum up, the feature extraction process of the proposed hash function can be described as follows:

Step 1) Normalize the input image into the standard size of 128 × 128, where the input image is first smoothed by a low-pass Gaussian filter of size 5 with a standard deviation of 3 before scaling. Let I_N denote the normalized image. Randomly generate a set of radii for feature extraction via a secret key.

Step 2) For a given radius ρ_i, the response is calculated for the ring given by $\Omega_i = \{\rho | \rho_i - 1 \leqslant \rho \leqslant \rho_i + 1\}$. Assign a random filter $\tilde{\omega}(x, y)$ for this ring as (10), where the frequencies are randomly selected. The filter responses $R(x, y)$ on this ring are calculated as

$$R(x, y)|_{(x,y)\in\Omega_i} = I_N(x, y) * \tilde{\omega}(x, y). \tag{11}$$

Step 3) Compute the rotation-invariant feature for ring Ω_i as in (12), where N_P denotes the number of the pixels in Ω_i. Equation (12) is equivalent to the integration in (8), while it is expressed in a discrete manner, i.e.,

$$F_i = \frac{1}{N_P} \sum_{(x,y)\in\Omega_i} |R(x, y)|^2 . \tag{12}$$

Step 4) Repeat steps 2) and 3) until the feature extraction for all the radii is finished.

As previously discussed, the filter responses located on the same ring are integrated to obtain rotation-invariant features. Similarly, the robust-hashing algorithm proposed in [18] exploits the rotation invariance in the Fourier–Mellin domain by integrating the transform magnitudes along a circle. However, our proposed feature extraction scheme differs from that in [18] in two aspects. First, the transforms in these two algorithms are implemented in different manners. The proposed algorithm adopts a random transform kernel (i.e., the rotation-invariant filter with varying frequencies), and the one in [18] uses a deterministic Fourier–Mellin kernel. Second, the frequency selection schemes are different. In the proposed algorithm, the filter responses computed with the frequencies randomly selected from each interval of the frequency range are involved in generating every individual feature. Thus, the features in the proposed paper can make a wider and more uniform coverage of the spectrum compared with those in [18], which results in the advantage of the proposed algorithm on discrimination as to be indicated by experimental results. In brief, the contribution of the proposed algorithm in feature extraction can be summarized as follows. First, a random filter is designed to extract robust and secure features for hash computation. Second, the impact of frequency selection on the discrimination of the hashing algorithm is studied, based on which an efficient random-frequency selection scheme is developed.

III. Feature Quantization Using Dithered LVQ

Here, we concentrate on the quantization process of the proposed robust-hash function. As previously discussed in Section I-B, the quantization scheme for robust hashing is expected to strike a good balance between robustness and discrimination, and it is thus believed that the uniform quantizer should be a better choice. Another problem to be considered in the quantizer design is the choice between vector quantization (VQ) and SQ. It has been demonstrated in [22] that quantizing feature data in the form of vectors can exhibit more superior performance than that of scalars. Based on these observations, uniform vector quantizers should be the most suitable quantization paradigm for robust hashing, whereas the only uniform quantizer in the family of VQ is the LVQ. Therefore, in this paper, we propose to quantize feature vectors using the dithered LVQ. The dithered LVQ inherits the uniform nature of the lattice quantizer, it can therefore lead to excellent robustness and discrimination performance. More importantly, the dithering process can also offer several advantages to robust hashing in terms of security. First, the randomly generated dither vector is added to the feature vector before lattice quantization; thus, different dither vectors can result in a series of possible quantization results. In this regard, an attacker is unlikely to succeed in estimating the hash string without knowing the secret key for dither vector generation. Second, the dithering process can also reduce the possibility of collision, thus making the finding of a collision pair even more computationally tough. Third, the quantization error in the dithered LVQ can be rendered as source independent, which can enhance the secrecy of the hash algorithm. Even in the case that the quantization error of the quantizer is available, the attacker is still not able to learn anything about of the feature data. Here, we first provide some preliminaries and notations for LVQ.

A. Preliminaries and Notations

The lattice is a set of regularly spaced vectors (i.e., lattice points) in the Euclidean space. The L-dimensional lattice $\Lambda \subset \mathbb{R}^L$ [32] consists of the integer linear combinations of basis vectors as

$$\Lambda = \left\{ \boldsymbol{\lambda} \,\middle|\, \boldsymbol{\lambda} = \sum_{i=1}^{L} n_i \boldsymbol{v}_i, n_i \in \mathbb{Z} \right\} \tag{13}$$

where $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_L\}$ are L basis vectors, all of which form the generator matrix of the lattice, as $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_L]^T$. Let $\boldsymbol{n}$ denote the array of integer coefficients $\boldsymbol{n} = [n_1, \cdots, n_L]^T$; then, any lattice point in Λ can be expressed as $\boldsymbol{\lambda} = \boldsymbol{V}^T \boldsymbol{n}$. These infinite lattice points form a structured codebook that can be used for VQ. LVQ is a mapping in $\mathbb{R}^L$, it maps an arbitrary input vector to a lattice point, i.e.,

$$\mathrm{LVQ}(\boldsymbol{x}) = \boldsymbol{V}^T \boldsymbol{n}. \tag{14}$$

In an optimum quantizer with minimum distortion, the input vector is mapped to its nearest lattice point (i.e., codeword). Therefore, each lattice point forms a Voronoi cell, and the cell of a given lattice point $\boldsymbol{\lambda}_0$ is defined as

$$\mathrm{VOR}(\boldsymbol{\lambda}_0) = \left\{ \boldsymbol{x} \,\middle|\, \|\boldsymbol{x} - \boldsymbol{\lambda}_0\| \le \left\| \boldsymbol{x} - \boldsymbol{V}^T \boldsymbol{n} \right\|, \forall \boldsymbol{n} \in \mathbb{Z}^L \right\}. \tag{15}$$

The Voronoi cells of lattice points are identical, and these cells generate a regular tiling of the space. Any Voronoi cell can be obtained by translating the cell of the origin $\mathrm{VOR}(\mathbf{0})$. According to the definition in [33], the notion of uniform quantizer refers to those with congruent cells. Therefore, LVQ is the uniform quantizer in multidimensional space.

B. Dithered-LVQ-Based Feature Quantization

In this paper, we adopt the D_4 lattice to construct the dithered lattice quantizer. The reason is that the process of nearest codeword searching in D_4 is quite easy and fast. The D_4 lattice is composed of the 4-D integer vectors with even component sum, and its generator matrix is

$$\boldsymbol{V} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

In what follows, the procedures of quantizing feature vectors will be described in detail.

1) Feature Vector Formation: Every four neighboring features obtained in Section II constitutes one vector for VQ. It should be noted that a robust-hash function should map an arbitrary input image to a fixed-length binary string. Accordingly, each feature vector should be quantized to a fixed number of binary bits. Therefore, the features are first normalized into the range of [0, 8). The normalized data represent the relationships between features, and they are more stable than the absolute amplitudes of features.

2) Dither Vector Generation: In the dithered LVQ, a random vector (i.e., dither vector) is added to the input vector. The final dithered vector is then quantized via LVQ. The study in [34] shows that the quantization error can be uniformly distributed in $\mathrm{VOR}(\mathbf{0})$ if and only if the dither vector is a Nyquist-V vector. Let $\boldsymbol{r}$ be a vector whose elements are independent and identically distributed and uniformly distributed in the range of $[-c, c)$, $(c > 0)$; then, $\boldsymbol{R} = \boldsymbol{V}^T \boldsymbol{r}$ is a Nyquist-V vector [34]. In this paper,$\boldsymbol{r}$ is randomly generated under the control of a secret key. Dither $\boldsymbol{R}$ is then added to the normalized feature vector $\boldsymbol{F}$ as $\boldsymbol{F}^* = \boldsymbol{F} + \boldsymbol{R}$, where $\boldsymbol{F}^*$ denotes the dithered vector.

3) Quantization of Dithered Vectors: The dithered vector $\boldsymbol{F}^*$ is quantized by finding its nearest lattice point. The nearest *integer* vector of $\boldsymbol{F}^*$ is first calculated as $N(\boldsymbol{F}^*)$, where $N(F_i^*) = [F_i^*]$ $(i = 1, 2, 3, 4)$ and $[\cdot]$ denotes the rounding operation. If $\sum N(F_i^*)$ is even, $N(\boldsymbol{F}^*)$ is the nearest lattice point of $\boldsymbol{F}^*$. Otherwise, the second nearest integer vector $S(\boldsymbol{F}^*)$ is obtained by modifying only one component of $N(\boldsymbol{F}^*)$. Let k denote the index of the component in $\boldsymbol{F}^*$ with the largest distance to the corresponding integer, namely, $k = \arg\max |[F_i^*] - F_i^*|$; then, $S(\boldsymbol{F}^*)$ can be obtained by modifying $N(F_k^*)$ as follows

$$S(F_k^*) = \begin{cases} [F_k^*] + 1 & \text{if } F_k^* > [F_k^*] \\ [F_k^*] - 1 & \text{otherwise} \end{cases}. \tag{17}$$

Notice that $|\sum S(\boldsymbol{F}^*) - \sum N(\boldsymbol{F}^*)| = 1$, $\sum S(\boldsymbol{F}^*)$ is even. As a result, $S(\boldsymbol{F}^*)$ should be the nearest lattice point of $\boldsymbol{F}^*$.

4) Binarization of Quantized Vectors: After lattice quantization, those dithered feature vectors are mapped to a set of lattice points (i.e., codewords). The next step is to convert these codewords to binary bits. In LVQ, labeling schemes are developed to assign an index for each lattice point. However, these labeling schemes are not suitable for robust hashing due to the following

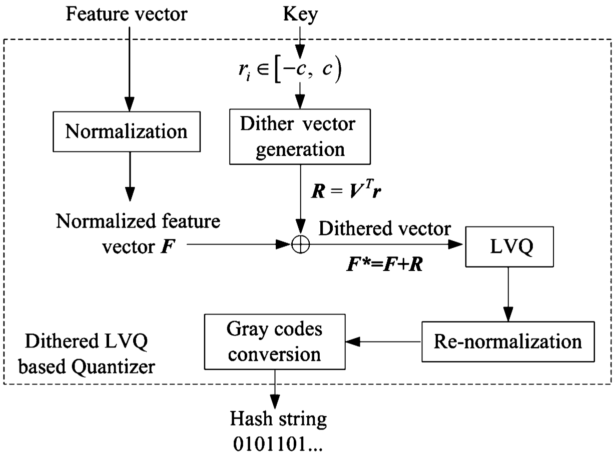


Fig. 4. Dithered-LVQ-based quantization scheme.

considerations. First, the indexes have variable lengths; thus, we cannot obtain a fixed-length hash string. Second, the distance between the indexes of neighboring lattice points may be quite large. As a result, the robustness of the hash function cannot be guaranteed. Third, some lattice labeling schemes are computationally intensive. In this paper, each component of the best matched codeword is independently converted into binary bits to form the final hash string. Since the dither vector is added to the normalized feature vector, thus, some components of its best matched codeword may exceed the range of [0, 8). Therefore, each component of the best matched codeword $\boldsymbol{\lambda}$ is normalized as $\lambda'_i = |\lambda_i| (\mathrm{mod} 8)$ $(i = 1, 2, 3, 4)$. To enhance the robustness of the hash function, each component of the normalized codeword is represented by its Gray code.

To sum up, the flowchart of the dithered-LVQ-based quantization scheme is shown in Fig. 4.

C. *Analytical Results for the Dithered-LVQ-Based Quantization Scheme*

1) Robustness Analysis Using Distortion Tolerance: In robust hashing, the feature data might be changed in the presence of distortion, while it is expected that the distorted feature could still be mapped to the same quantization index as the original one. In this sense, a robust quantizer should be the one that can tolerate a higher amount of distortion. In other words, the robustness of a quantizer in robust hashing depends on the maximal allowable amount of distortion associated with each quantization interval. Hence, here, we will give an analytical evaluation of the robustness of LVQ from the viewpoint of distortion tolerance. In addition, the comparisons on distortion tolerance among the LVQ, the adaptive quantizer, and the uniform SQ will be presented.

We start by estimating the distortion tolerance of LVQ. For the purpose of easy illustration and computation, we take the 2-D hexagonal lattice as the example, and its generator matrix can be expressed as

$$\boldsymbol{V} = \begin{bmatrix} \sqrt{3} & 1 \\ 0 & 2 \end{bmatrix}.$$

As the Voronoi cells are translationally identical, all the cells have exactly the same distortion tolerance. As a result, the basic cell VOR$(\mathbf{0})$ shown in Fig. 5(a) is selected to illustrate the computation of the distortion tolerance.

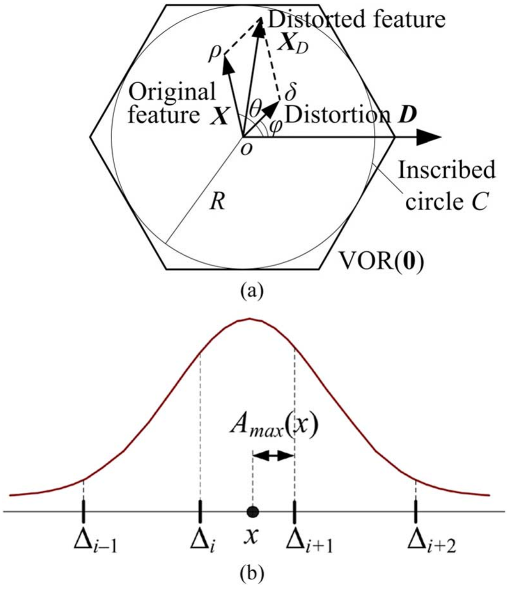


Fig. 5. Distortion tolerance in LVQ and the adaptive quantizer. (a) LVQ. (b) Adaptive quantizer.

Given the original vector $\boldsymbol{X} = (x_1, x_2) \in$ VOR$(\mathbf{0})$ and distortion $\boldsymbol{D} = (d_1, d_2)$, the hash bits (i.e., the quantization index of $\boldsymbol{X}$) will keep unchanged as long as the distorted feature $\boldsymbol{X}_D$ still lies in the same cell, as illustrated in Fig. 5(a), i.e.,

$$\boldsymbol{X}_D = \boldsymbol{X} + \boldsymbol{D} \in \mathrm{VOR}(\mathbf{0}). \tag{18}$$

Our goal is to estimate the upper bound of the distortion that can meet the aforementioned requirement. For computational convenience, cell VOR$(\mathbf{0})$ is approximated by its inscribed circle, as shown in Fig. 5(a). Accordingly, the original and distortion vectors are represented in the polar coordinate system as $\boldsymbol{X} = (\rho\cos\theta, \rho\sin\theta)$, $\boldsymbol{D} = (\delta\cos\varphi, \delta\sin\varphi)$. As a result, (18) can be rewritten as

$$\rho^2 + \delta^2 + 2\rho\delta\cos(\theta - \varphi) \leqslant R^2 \tag{19}$$

where R is the radius of the inscribed circle C. By solving (19), we have

$$\delta \leqslant -\rho\cos(\theta - \varphi) + \sqrt{R^2 - \rho^2\sin^2(\theta - \varphi)}. \tag{20}$$

Equation (20) gives the upper bound of the distortion magnitude below which $\boldsymbol{X}_D$ can be mapped to the same cell as $\boldsymbol{X}$. Since the upper bound depends on angle φ of the distortion and the original feature $\boldsymbol{X} = (\rho\cos\theta, \rho\sin\theta)$, we denote it by $\delta_{\max}(\varphi, \boldsymbol{X})$ (or equivalently $\delta_{\max}(\varphi, \rho, \theta)$), and we have $\delta_{\max}(\varphi, \boldsymbol{X}) = \delta_{\max}(\varphi, \rho, \theta) = -\rho\cos(\theta - \varphi) + \sqrt{R^2 - \rho^2\sin^2(\theta - \varphi)}$. Consequently, we estimate the distortion tolerance for each individual component of the feature vector so that the distortion tolerance of LVQ can be compared with those of the 1-D SQs. Taking the horizontal component for example, the maximal amount of allowable distortion is $D_{\max} = \delta_{\max}(\varphi, \boldsymbol{X})|\cos\varphi|$. In order to make a complete evaluation of distortion tolerance, we calculate the expectation of $D_{\max}$ over the domain of $\boldsymbol{X} \in C$ and $\varphi \in [0, 2\pi)$. Let $p(\boldsymbol{X})$ denote the probability of the feature vector, and the distortion tolerance of LVQ is given by[2]

$$\overline{D}_{\text{LVQ}} = \frac{\int_C \int_0^{2\pi} \delta_{\max}(\varphi, \boldsymbol{X}) |\cos\varphi| p(\varphi) p(\boldsymbol{X}) d\varphi d\boldsymbol{X}}{\int_C p(\boldsymbol{X}) d\boldsymbol{X}}. \quad (21)$$

Angle φ is uniformly distributed as $p(\varphi) = (1/2\pi)$. Let us take the independently and normally distributed feature for instance, where $x_1, x_2 \sim N(0,1)$ and $p(\boldsymbol{X}) = p(x_1)p(x_2) = (e^{-(\rho^2/2)}/2\pi)$; then, the $\overline{D}_{\text{LVQ}}$ in (21) can be written as

$$\overline{D}_{\text{LVQ}} = \frac{\int_0^R \int_0^{2\pi} \int_0^{2\pi} \delta_{\max}(\varphi, \rho, \theta) |\cos\varphi| e^{-\frac{\rho^2}{2}} \rho d\varphi d\theta d\rho}{4\pi^2 \int_0^R e^{-\frac{\rho^2}{2}} \rho d\rho}. \quad (22)$$

In the hexagonal lattice, the inscribed circle C has a normalized radius of $R = 1$. We compute (22) numerically, and the result shows that $\overline{D}_{\text{LVQ}} = 0.568$.

For comparison purposes, the distortion tolerance of the adaptive quantizer and the uniform SQ are also estimated and compared with that of the LVQ. It is obvious that the distortion tolerance of a quantizer depends on its average step size. Hence, the distortion tolerance is only comparable among the quantizers with the same average step size. In the following discussions, the adaptive quantizer and the uniform SQ under consideration are all in the range of [−8, 8], with eight levels. As a result, both of these two SQs have an average step size of 2 that is equal to the diameter of the inscribed circle of VOR($\mathbf{0}$) in LVQ. In this way, a fair comparison can be made.

In what follows, the distortion tolerance of the adaptive quantizer will be estimated. Here, we concentrate on the deterministic version of the adaptive quantizer, where no random region is assigned in each quantization interval. The quantization intervals $[\Delta_i, \Delta_{i+1}]$ are divided according to the distribution of the feature, i.e., $p(x)$, such that

$$\int_{\Delta_i}^{\Delta_{i+1}} p(x)dx = \int_{\Delta_{i+1}}^{\Delta_{i+2}} p(x)dx, \quad (i = 1, 2, \ldots, Q-1) \quad (23)$$

where Q is the level of the quantizer. As shown in Fig. 5(b), for a given feature data $x \in [\Delta_i, \Delta_{i+1}]$, the maximal allowable distortion is

$$A_{\max}(x) = \min\left(|x - \Delta_i|, |x - \Delta_{i+1}|\right). \quad (24)$$

Considering the eight-level adaptive quantizer operating in the range of [−8, 8], its distortion tolerance can be obtained by calculating the expectation of $A_{\max}(x)$, i.e.,

$$\overline{D}_{\text{AQ}} = \frac{\int_{-8}^{8} A_{\max}(x) p(x) dx}{\int_{-8}^{8} p(x) dx}. \quad (25)$$

As in LVQ, the value of $\overline{D}_{\text{AQ}}$ is numerically estimated for normally distributed features, and we have $\overline{D}_{\text{AQ}} = 0.187$. Likewise, the distortion tolerance of the uniform SQ is obtained, while the detailed calculation is not presented here for the sake of brevity. The result reveals that the distortion tolerance of the uniform SQ is $\overline{D}_{UQ} = 0.497$. It is evident that LVQ can tolerate a larger amount of distortion compared with the two SQs. In other words, LVQ is superior to both adaptive and uniform SQs in terms of robustness.

2) Randomness Analysis Using Entropy: Here, we evaluate the randomness of the dithered-LVQ-based quantization scheme using the entropy metric proposed in [18]. As described in Section III-B, the dithered vector $\boldsymbol{F}^* = \boldsymbol{F} + \boldsymbol{V}^T\boldsymbol{r}$ is quantized by LVQ. According to the generator matrix of D_4, each dimension of $\boldsymbol{F}^*$ can be written as

$$\begin{cases} F_1^* = F_1 + 2r_1 + r_2 + r_3 + r_4 \\ F_i^* = F_i + r_i \quad (i = 2, 3, 4). \end{cases} \quad (26)$$

The dithered vector $\boldsymbol{F}^*$ can be quantized to a series of possible codewords when r_i varies in $[-c, c)$, $c > 0$. For a given input vector $\boldsymbol{F}$, we investigate the possibility of the nearest codeword for its dithered vector, based on which the entropy of the random quantizer is calculated. The randomness of the quantization results depends on the value of c, and we take $c = (1/2)$ as the example to analyze the entropy of the dithered LVQ. As discussed in Section III-B, the nearest integer vector $N(\boldsymbol{F}^*)$ is first calculated for $\boldsymbol{F}^*$. According to (26), the component ranges of $N(\boldsymbol{F}^*)$ are as follows when $c = (1/2)$:

$$\begin{cases} N(F_1^*) \in \{[F_1], [F_1] \pm 1, [F_1] \pm 2, [F_1] \pm 3\} \\ N(F_i^*) \in \{[F_i], [F_i] \pm 1\} \quad (i = 2, 3, 4). \end{cases} \quad (27)$$

If $N(\boldsymbol{F}^*)$ is not a lattice point in D_4, the second nearest integer vector $S(\boldsymbol{F}^*)$ is calculated as the best matched codeword, as shown in (17). It is easy to verify that the components of $S(\boldsymbol{F}^*)$ are also within the range in (27). By combining those possible components in (27), we can obtain $7 \times 3^3 = 189$ vectors. Half of these vectors have even component sum; hence, there are 94 possible codewords for $\boldsymbol{F}^*$. These codewords have approximately equal probabilities as $p(\boldsymbol{\lambda}_k) = (1/94)$, $(k = 1, \cdots, 94)$. Thus, the average entropy per component of the codeword in the D_4-based dithered LVQ is

$$H_{\text{DLVQ}} = -\frac{1}{4} \sum_k p(\boldsymbol{\lambda}_k) \log_2 p(\boldsymbol{\lambda}_k) = \frac{\log_2 94}{4} = 1.639. \quad (28)$$

This entropy rate is also compared with that of the adaptive quantizer. It has been derived in [18] that the entropy of the adaptive quantizer is $H_{\text{AQ}} = r \log_2(e)$, where $r \leq (1/2)$ is the portion of the random region in each interval. The upper bound of H_{AQ} is 0.721, which is lower than that of the dithered LVQ. Therefore, the dithered LVQ can yield a higher amount of randomness than the adaptive quantizer.

In Section IV, the randomness of the dithered LVQ and the adaptive quantizer will be further demonstrated with experimental results, and the focus is placed on the amount of the perturbation introduced to the final hash value.

3) Analysis on the Selection of Lattice: The selection of a proper lattice is one of the primary concerns in the proposed feature quantization scheme, and two factors have to be considered, i.e., the category and the dimensionality of the lattice. In

[2] Although we take the horizontal component of the feature vector to calculate the distortion tolerance of LVQ, it should be noted that the result for the vertical component is exactly the same due to the symmetry of the integral. The integral in the denominator of (21) is used to normalize the probability of the vectors within C (i.e., $p(\boldsymbol{X})$ in the numerator).

this section, we analyze the influence of these two factors on the performance of the lattice quantizer in robust hashing.

The quantizer in the proposed hashing algorithm is constructed by D_4, which belongs to the family of checkerboard lattices. The primary reason is that the lattices form the checkerboard family, denoted as D_n, have fairly low encoding complexities. In what follows, we first analyze the encoding complexity of the D_n lattice and then compare it with another two commonly used lattices A_n and A_n^*. As discussed in Section III-B3, an arbitrary input vector can be mapped to its nearest D_4 lattice point by one rounding operation of the vector and, at most, one addition operation. Hence, its encoding complexity increases in proportion to the dimensionality of the lattice as $O(n)$. For the A_n lattice, we take the fast encoding algorithm introduced in [35] as the example to analyze its encoding complexity. A_n is an n-dimensional lattice whose lattice points are those integer vectors with zero component sum. In searching the nearest A_n lattice point of an input vector, the sorting of vector elements is the dominant part of the complexity. If we only take the sorting operation into account, the encoding complexity of A_n can be expressed as $O(n \log n)$. A_n^* is the dual lattice of A_n and can be written as the union of $(n+1)$ translations of A_n, i.e., $A_n^* = \bigcup_{i=0}^{n}(A_n + \boldsymbol{r}_i)$, where $\boldsymbol{r}_i$ is the ith translation vector [35]. Therefore, its encoding complexity is $(n+1)$ times that of A_n [35], namely, $O(n^2 \log n)$. Apparently, D_n has the lowest encoding complexity among these three common lattices. The low encoding complexity of D_n could benefit the applications of robust hashing on large-scale image databases.

The dimensionality of the checkerboard lattice is set to four in the proposed paper, and now, we investigate the relationship between the dimensionality of the lattice and the performance of the quantizer. The performance of an LVQ in robust hashing depends on the volume of its quantization cell. The volume of the quantization cell in LVQ can be calculated as $|\det \boldsymbol{V}|$ [32], where $\det$ denotes the determinant of a matrix and $\boldsymbol{V}$ is the generator matrix of the lattice. According to the definition of the D_n lattice, its generator matrix can be expressed as

$$\boldsymbol{V}_n = \begin{bmatrix} 2 & \mathbf{0}_{n-1}^T \\ \mathbf{1}_{n-1} & \boldsymbol{I}_{n-1} \end{bmatrix} \tag{29}$$

where $\mathbf{0}_{n-1} = [0, 0, \cdots, 0]^T$ is the $(n-1)$-dimensional vector with the elements of all zero, $\mathbf{1}_{n-1} = [1, 1, \cdots, 1]^T$ is the $(n-1)$-dimensional vector with the elements of all one, and $\boldsymbol{I}_{n-1}$ is the $(n-1) \times (n-1)$ identity matrix. It is straightforward that $|\det \boldsymbol{V}| = 2$ for any n. Accordingly, the quantization cells in the quantizers constructed by the D_n lattices with different dimensionalities have exactly the same volume. Therefore, it can be concluded that different selections of the lattice dimensionality will not affect the performance of the quantizer in robust hashing.

IV. Experimental Results and Discussions

A. Experimental Setup

A number of experiments and comparisons are carried out to evaluate the efficiency of the proposed algorithm. We start this section by describing the experimental setup, including the test base, comparative algorithms, and the associated parameter settings. The test database for robustness and overall statistical performance evaluation contains 500 gray-level images, and it is composed of the images in the Photography Image Database [36], standard benchmark images, scenery pictures captured by digital cameras, and those collected from the Internet. The test set for measuring the collision rate is even larger, and it contains 5×10^4 images since the collisions between hash strings could be only observed when the number of test images is large enough. To the best of our knowledge, this should be the largest scale image database for collision rate evaluation in robust hashing. A large portion of the images are collected from publicly available image databases, including ImageNet [37] and the database for object recognition [38]. The test images in this paper are of various sizes so that it can simulate the case in more practical applications with varying image size. The widths and the heights of the test images range from 96 to 4272. Some typical sizes are 96 × 96, 128 × 128, 256 × 256, 384 × 256, 400 × 300, 512 × 512, 1024 × 678, and 2848 × 4272. Four state-of-the-art hashing algorithms are simulated and compared with our proposed one, including the NMF-Hash in [14], the RASH in [19], the compressive-sensing-based hashing (CS-Hash) in [39], and the Radon-transform-based hashing (Radon-Hash) in [40]. The parameter settings in each algorithm are as follows. In the proposed algorithm, the random filter for each ring is the summation of ten rotation-invariant filters with random frequencies, and the frequency of the kth rotation-invariant filter is randomly selected from $[((k-1)/10), (k/10)), k = 1, \cdots, 10$. Parameter γ in each rotation-invariant filter shown in (5) is set to one. Forty rings are generated for hash computation, and each has a width of three pixels.[3] The generator matrix for constructing the dithered LVQ is the same as that shown in (16). In NMF-Hash, the number of randomly selected subimages is $p = 25$, the parameters for matrix factorization are $r_1 = 2$ and $r_2 = 2$, and the size of subimages is 100 × 100. In order to output the binary hash string, the features are finally quantized using an eight-level adaptive quantizer. In RASH, totally 180 projection lines are generated with the angle varying from 1° to 180°, and the variance of the pixels within each line are calculated. Forty lowest frequency discrete-cosine-transform coefficients of the variances are then quantized by a uniform quantizer to output the hash string. The parameter setting of CS-Hash is the same as described in [39], where the scrambled block Hadamard ensemble matrix is chosen as the measurement matrix in compressive sensing, and the components of the measurement vector are quantized by a 32-level nonuniform quantizer. In Radon-Hash, the matrix of the Radon transform coefficients is uniformly divided into 40 × 20 blocks, and the mean value of each block is calculated. The two-level Haar wavelet is adopted to decompose the mean value map for hash computation as in [40]. Then, the Fourier–Mellin-transform-based hashing algorithm (FM-Hash) that was developed in [18] to tackle rotation distortions is also simulated, and the

[3] The number of the selected rings depends on the application scenario. In content authentication, the rings should make a full coverage of the image, while in the applications with less rigorous requirements on false acceptance, a fewer number of rings could be selected to make a tradeoff between identification accuracy and computation complexity.

performance of our proposed paper and FM-Hash is compared in all the rotation related manipulations. In FM-Hash, each feature is calculated using the transform coefficients on five randomly selected circles in the Fourier–Mellin domain, and totally 360 equidistant coefficients are sampled on each circle. As suggested in [18], the features are quantized by the adaptive quantizer. The numbers of the hash bits in the proposed algorithm, NMF-Hash, RASH, CS-Hash, Radon-Hash, and FM-Hash are 120, 150, 200, 385, 200, and 200, respectively. All the experiments are carried out in MATLAB R2008b on a desktop computer with 2.8-GHz dual-core central processing unit and 2-Gb random access memory. The detailed simulation and the comparison results are presented in the following subsections.

B. Robustness Against Content-Preserving Manipulations

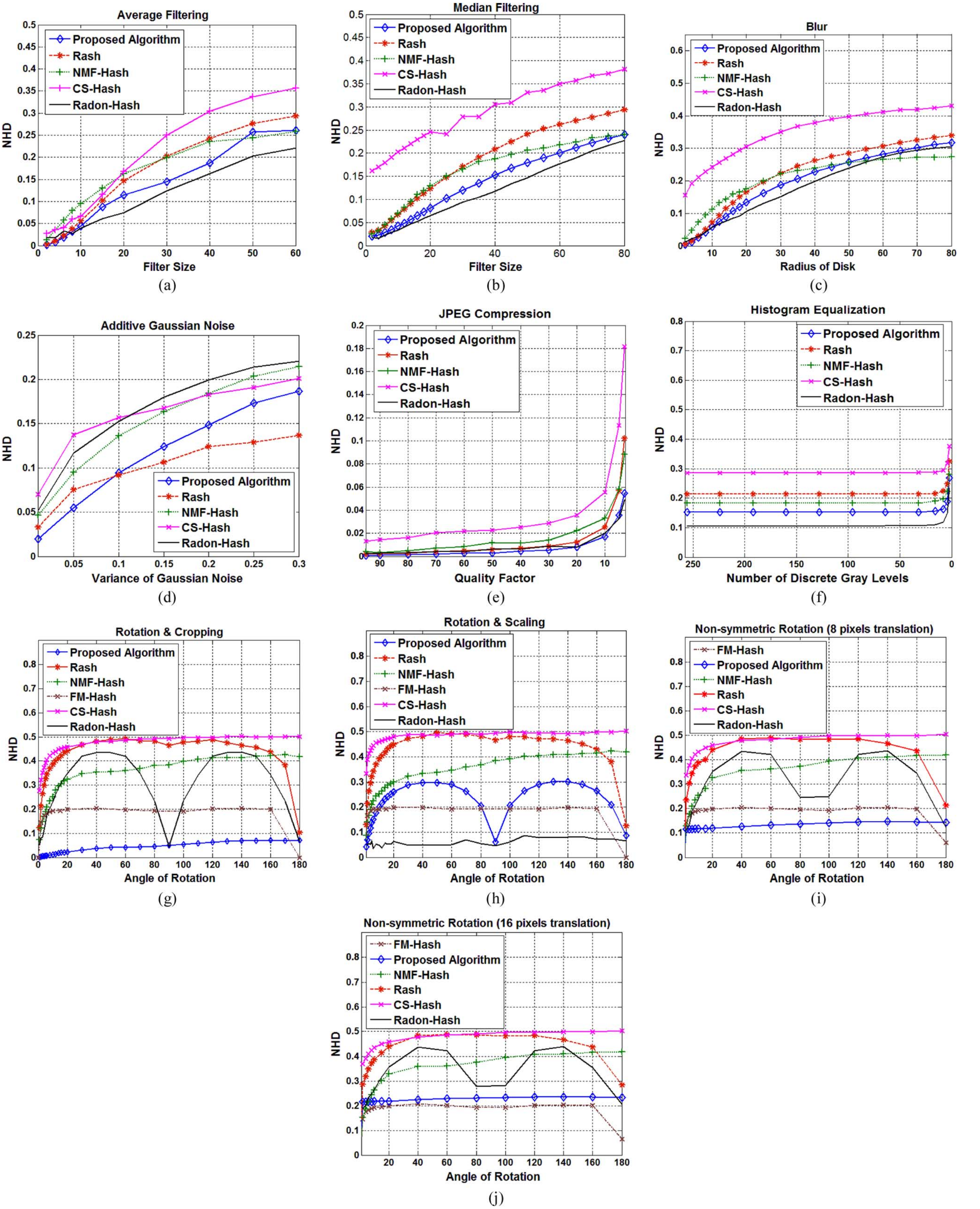
The robustness of the proposed hashing algorithm is evaluated using a series of content-preserving distortions, including average filtering, median filtering, blur, Gaussian noise addition, JPEG compression, histogram equalization, rotation+cropping, rotation+scaling, and nonsymmetric rotation. In rotation+cropping and rotation+scaling, the rotated images are cropped and scaled, respectively, to fit the original size. In nonsymmetric rotation, the input image is first horizontally translated by 8 and 16 pixels, respectively, and the rotation+cropping manipulation is then implemented on the translated image. The robustness of each hash function is evaluated by calculating the distance between the hash strings of the original image and its distorted version. The normalized hamming distance (NHD) is adopted as the distance metric for hash comparison. The NHD curves of these hash functions under each distortion are displayed in Fig. 6. Here, we place more emphasis on the simulations for rotation-related distortions by including more comparative algorithms. As shown in Fig. 6(g)–(j), the rotation-resistant FM-Hash is simulated as an extra comparative algorithm. Since those alignment preserving distortions such as JPEG compression are relatively easy to deal with and the other comparative algorithms can already represent most categories of state-of-the-art hashing algorithms, FM-Hash is not included in Fig. 6(a)–(f) for performance comparison. As shown in the figure, our proposed hash function exhibits satisfactory robustness even in the presence of large degree rotations. It is shown in Fig. 6(g) that the NHD of our proposed algorithm is 0.05 at the rotation angle of 80°, which is much lower than that of the other five algorithms. In rotation+scaling, Radon-Hash shows better robustness than other algorithms owing to the fact that the relationships between Radon transform coefficients are invariant to the scaling operation that follows the rotation. However, both FM-Hash and the proposed algorithm outperform Radon-Hash in terms of the robustness against rotation+cropping and nonsymmetric rotation manipulations, since Radon-Hash could not resist the cropping and translation operations.

C. Discrimination, Confusion, and Diffusion

This section focuses on the capability of the proposed hashing algorithm in discriminating between content distinct images, as well as its sensitivities to the changes in the image content and the secret key. We start this section by measuring the collision rate of the algorithm. The hash values of 5×10^4 images in the test database are computed and compared. The hash values of the test images are first calculated with distinct keys, and the hash distances are computed for each pair of images. No collision is observed in this case. The average NHD between hash values is 0.498, which is consistent with the expectation that nearly 50% of the bits are different in the hash strings of content-distinct images. It should be noted that, in key-dependent hashing, the hash value is determined by both the content of the image and the key for hash computation. Hence, different key selections will definitely decrease the collision rate. For this reason, the collision rate of the proposed hash function is also evaluated in the case where the hash values of all the test images are computed under a fixed key. Hash collisions are detected in this case with the rate of 3.2×10^{-3}, and the average NHD between hash values is 0.478.

The fragility of the hash function to malicious tampering is also investigated here. Content modifications are imposed on original images to alter their semantic information, and the original and tampered images are shown in Fig. 7. The original beach image shown in Fig. 7(a) is tampered by placing two sailboats on the beach. Almost 10% pixels are modified by object insertion, and the NHD between these two hash strings is 0.55. The tampering on the other two images is spatially slight but semantically significant, as shown in Fig. 7(d) and (f), where only a tiny fraction (less than 0.5%) of the pixels are modified but they could cause drastic changes on semantic information. For the two images displayed in Fig. 7(c) and (e), the hash distances between the original and tampered images are 0.46 and 0.33, respectively. The results reveal that the proposed hashing algorithm is also sensitive to spatially slight but semantically significant tampering, with some possible reasons as follows. First, the random filter in the proposed algorithm is designed to cover a wide range of frequencies, and thus, the filter responses corresponding to middle and high frequencies can be quite sensitive to malicious tampering. Second, the tampered region in the forgeries such as object insertion and removal is usually localized within a neighborhood of an image and covers several neighboring rings, which can consequently result in remarkable changes on filter responses. Moreover, the normalization of features can also improve the sensitivity of the algorithm. If the minimal or maximal feature is altered by tampering, all the normalized features will be accordingly changed since both the two values are involved in computing each normalized feature.

In addition, the confusion and diffusion capabilities of the proposed hash function are assessed with the test method proposed in [41]. The confusion of key-dependent robust hashing indicates the sensitivity of the output hash to the change in the secret key. In the simulation, the secret key for hash computation is increased by one at each time, and the distances between the hash values computed with the initial key and changing keys are then calculated. Fig. 8(a) shows the hash distance at each time during the variation of the secret key, from which it can be seen that the proposed algorithm shows satisfactory confusion capability. The diffusion of the algorithm is investigated by comparing the distance between the hash values of the original test image and its tampered versions with substituted perceptual units. The perceptual unit for a 512×512 image is defined



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j)

Fig. 6. Performance comparisons on the robustness against content-preserving manipulations.

as a 16 × 16 block in [41]. Both the localized and distributed substitutions mentioned in [41] are implemented. In localized substitution, the substituted perceptual units are localized within a specific neighborhood in the test image, while in distributed substitution, the position of each substituted perceptual unit is randomly selected. The hash distances between the original and

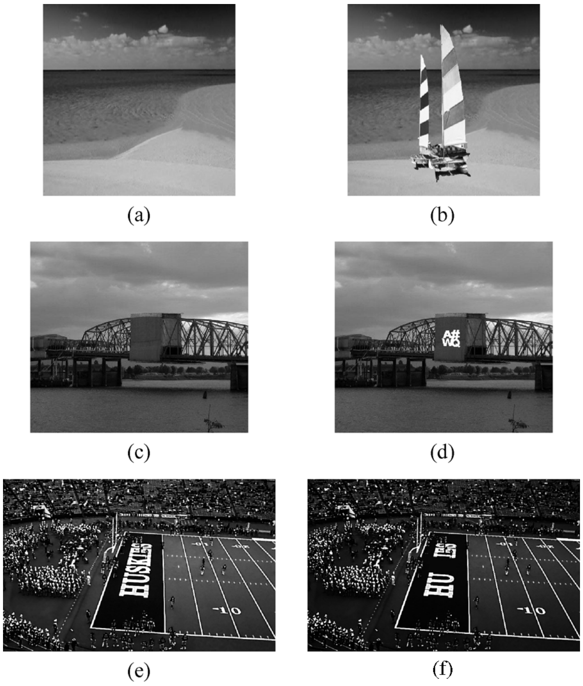


Fig. 7. Original and tampered images. (a), (c), (e) Original images. (b), (d), (f) Tampered images.

tampered images with increasing numbers of substituted blocks are calculated. Fig. 8(b) shows the plot of the hash distance versus the number of substituted blocks in both localized and distributed substitutions. Compared with the algorithms tested in [41], the proposed algorithm shows a stronger diffusion capability in localized substitution. However, the diffusion capability of the proposed algorithm in the distributed substitution is weaker than that in the localized substitution, with the major reason as follows. In distributed substitution, since the substituted blocks are separately distributed, each individual 16 × 16 block can only modify a tiny minority of the pixels on the rings for feature extraction. Apparently, the changes are diluted in distributed substitution. Therefore, with the same number of substituted blocks, the changes on features imposed by distributed substitution are with much smaller significance than those imposed by localized substitution.

D. Overall Performance Evaluation Using ROC Curve and EER

Here, we focus on the overall performance of the proposed hashing algorithm. The overall performance of each algorithm is first assessed using the receiver operating characteristic (ROC) curve that demonstrates the relationship between the probability of correct detection (P_D) and the false rejection rate (FRR) in hash comparison. The ROC curve can quantify the tradeoff of the hashing algorithm between robustness and discrimination. For comparison purpose, the ROC curves of all the comparative algorithms are plotted in the same figure, as shown in Fig. 9. In addition, the equal error rates (EER) of these hash functions are computed as a quantity criterion for performance comparison, as tabulated in Table I. As shown in the ROC curves and the EER values, our proposed algorithm shows remarkable superiority over the other five algorithms in rotation+cropping and nonsymmetric rotations due to its robustness against the rotations of large degrees. It has been revealed in Fig. 6(h) and (j) that compared with the proposed algorithm, FM-hash can provide a higher degree of robustness in rotation+scaling and nonsymmetric rotation, while the ROC curves demonstrate that the proposed algorithm is superior to FM-Hash in the overall performance. This fact implies that the proposed algorithm is more discriminative than FM-hash. As mentioned in Section II-B, the proposed algorithm can achieve a better discrimination since its random-frequency selection scheme can make a wider and more uniform coverage of the spectrum in each feature. More detailed discussions on the relationship between random-frequency selection and discrimination will be presented in the following subsection.

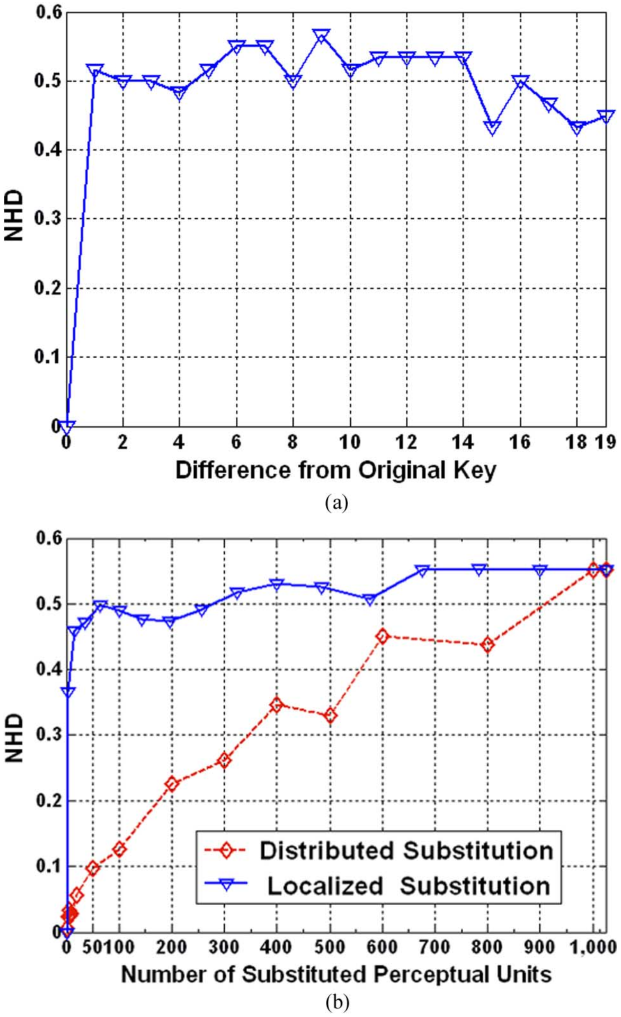
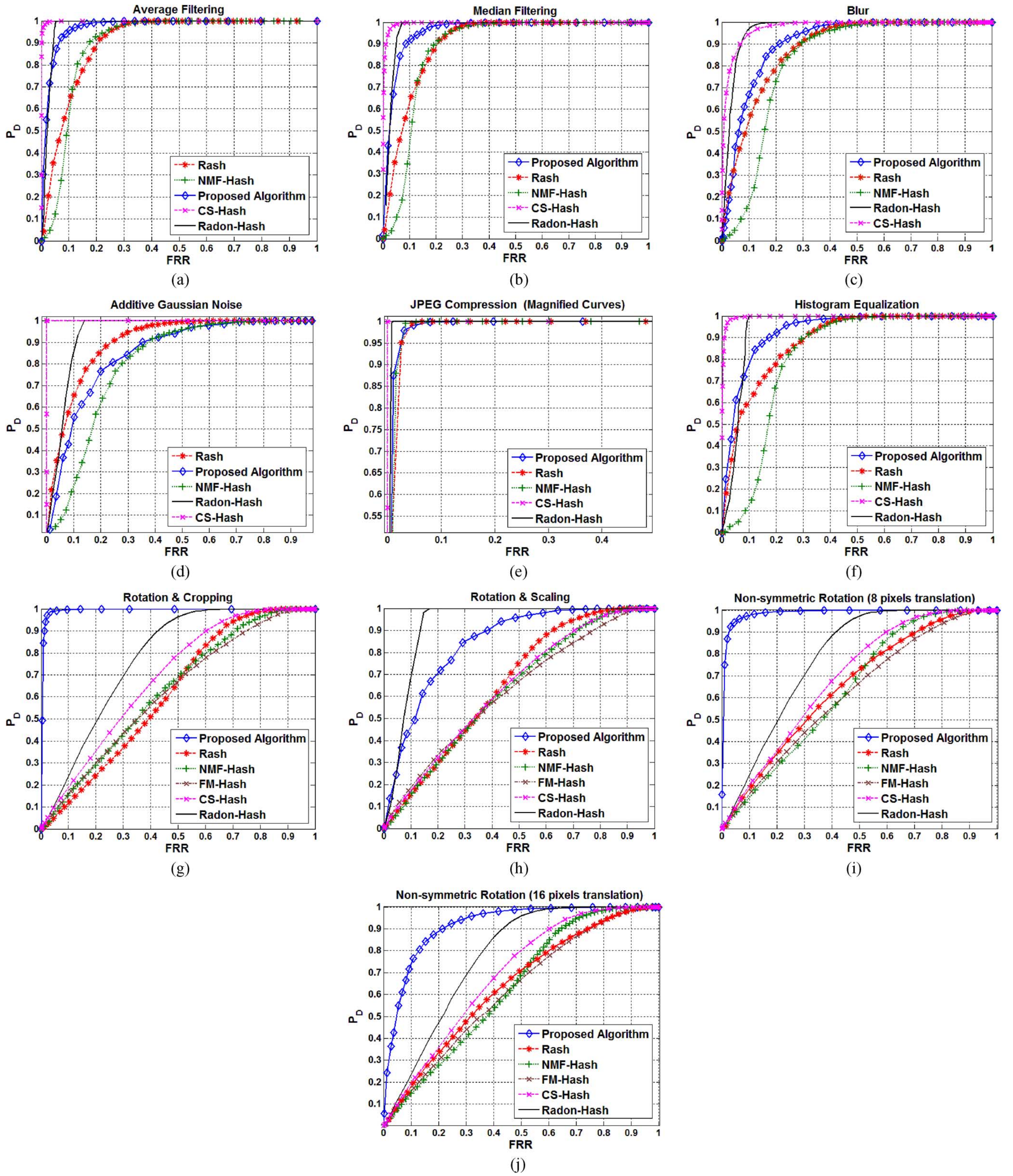


Fig. 8. Confusion and diffusion capabilities of the proposed algorithm. (a) Confusion capability. (b) Diffusion capability.

In previous simulations, each input image and its distorted versions are included in the set of reference images for hash comparison so that the FRR can be estimated. However, in the applications such as broadcast monitoring, the input image may not be present in the reference database, and the false acceptance rate (FAR) should be considered. Thus, simulations are performed to evaluate the FAR of the proposed algorithm in such case by taking broadcast monitoring as the example. Two



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j)

Fig. 9. ROC curves of different hashing algorithms under content-preserving manipulations.

image sets are involved in the simulation, namely, the sets of test images and reference images. The test set contains 1000 images selected from the Uncompressed Color Image Database [42], and the test images are used as the input of the broadcast monitoring system. The reference image set is the one used in discrimination assessment that is composed of 5×10^4 images. None of the images in the test set is included in the reference set; thus, these two image sets can be used to simulate the case where the input image is not present in the reference database. The NHD between the hash strings of the test and reference images is compared against a threshold to decide whether the test image is perceptually identical with any image in the reference

TABLE I
COMPARISONS ON EER

Manipulations	Proposed	NMF-Hash	RASH	CS-Hash	Radon-Hash
Aver. Filter.	0.074	0.134	0.091	0.016	0.047
Med. Filter.	0.076	0.134	0.108	0.031	0.066
Blur	0.125	0.197	0.143	0.077	0.078
Noise Add.	0.234	0.252	0.184	0.001	0.107
JPEG	0.003	0.002	0.009	0.001	0.001
Hist. Equal.	0.104	0.197	0.174	0.031	0.088
Rot.+Crop.	0.009	0.434	0.399	0.380	0.320
Rot.+Scal.	0.157	0.408	0.396	0.405	0.125
Non-symm. Rot. (8 pels transl.)	0.056	0.415	0.391	0.380	0.322
Non-symm. Rot. (16 pels transl.)	0.157	0.425	0.407	0.381	0.322

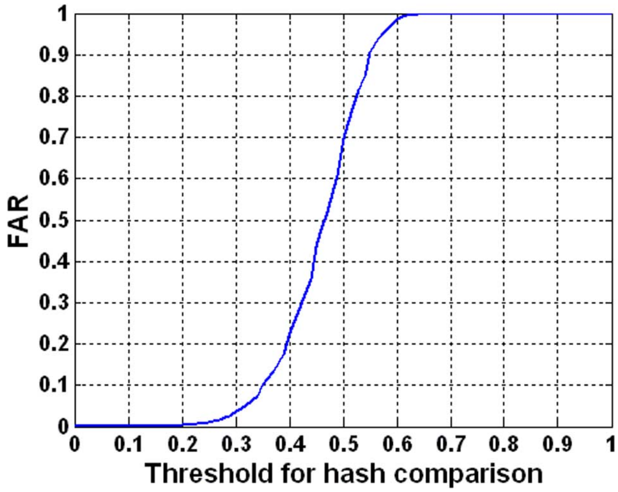


Fig. 10. Variation of FAR with hash comparison threshold.

database. The plot of FAR versus the threshold is displayed in Fig. 10.

E. Impact of Random-Frequency Selection on Discrimination

In the proposed random filter, ten random frequencies are selected from each interval of the frequency range, and the reason is that a wider coverage of the spectrum in each individual feature can lead to better discrimination of the hash function. The explanation of this phenomenon was given in Section II-B. Here, experiments are conducted to investigate the relationship between the discrimination and the number of random frequencies. The number of frequencies for computing each feature varies from 1 to 14 in the simulation, and we assess the discrimination of the algorithm under different parameter settings. The plot of the average hash distance between content-distinct images versus the number of random frequencies is presented in Fig. 11. As shown in the figure, the discrimination becomes better as the number of random frequencies increases, and a satisfactory discrimination can be achieved when ten frequencies are selected. Although selecting one frequency in the whole range can result in higher randomness, it would significantly degrade the discrimination of the hash function since the average hash distance between content-distinct images is only 0.21 in this case. Obviously, the features extracted using very few frequencies cannot effectively distinguish content-distinct images. Another experiment is carried out to demonstrate this fact. In the simulation, 40 features are extracted from two distinct images with one and ten random frequencies for each feature, respectively. As shown in Fig. 12(a), which corresponds to the case of single frequency, the feature curves of the two images have almost identical shapes. Accordingly, the quantization indexes of the feature data could be highly similar for these two images, which can explain the poor discrimination in this case. On the contrary, the curves of the features extracted from the two images are quite different when each feature is computed with ten random frequencies, as shown in Fig. 12(b).

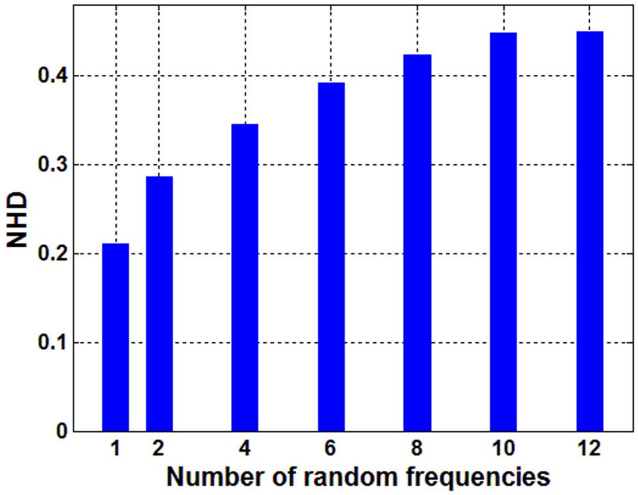


Fig. 11. Discrimination of the hashing algorithm versus the number of random frequencies.

F. Performance Evaluation of Quantization Schemes

1) Performance Comparisons Between LVQ and SQs: Here, the overall performance of the dithered-LVQ-based quantization scheme is investigated and compared with that of the adaptive quantizer and the uniform SQ. The feature data extracted by the random Gabor filter are quantized by three quantizers, respectively. The parameter setting for dithered LVQ is the same as that discussed in Section III, and the two SQs contain eight levels. We choose the average filtering as the distortion for performance evaluation. The NHD curves of the three quantization schemes are plotted in Fig. 13(a), from which we can see that the dithered LVQ outperforms the two SQs in terms of robustness. This observation is consistent with the analytical results of robustness presented in Section III-C.

The overall performance of quantization schemes is compared via ROC curves, as shown in Fig. 13(b). As revealed in the figure, the overall performance of the dithered LVQ outperforms the other two quantizers. The dithered LVQ and the uniform SQ can make a better balance between robustness and discrimination than the adaptive quantizer. Moreover, it is interesting to note that the multidimensional uniform quantizer (i.e., LVQ) outperforms the scalar one.

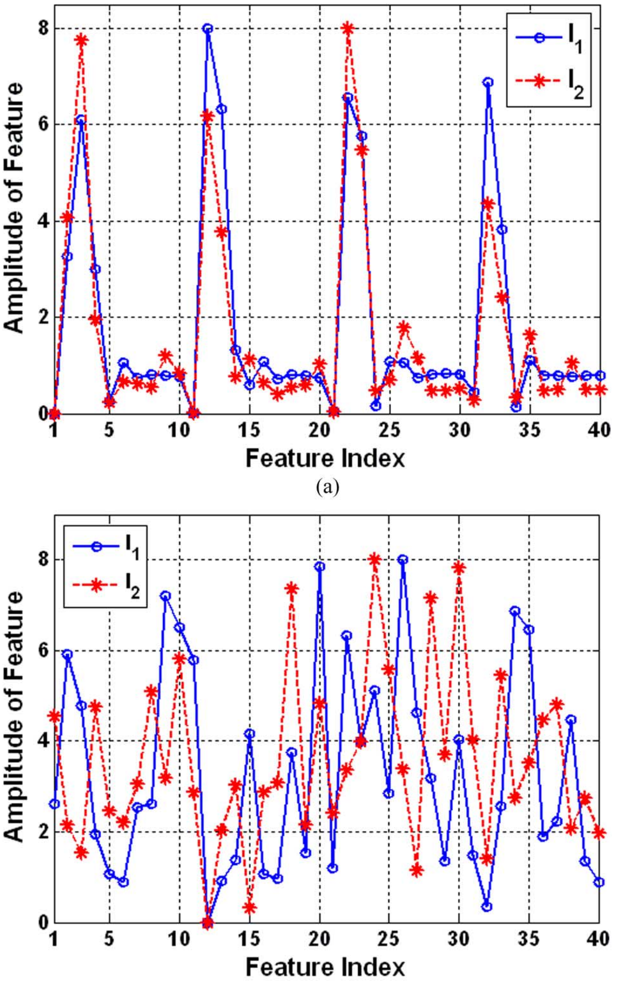


Fig. 12. Features extracted from two distinct images using different numbers of random frequencies: (a) Single random frequency. (b) Ten random frequencies.

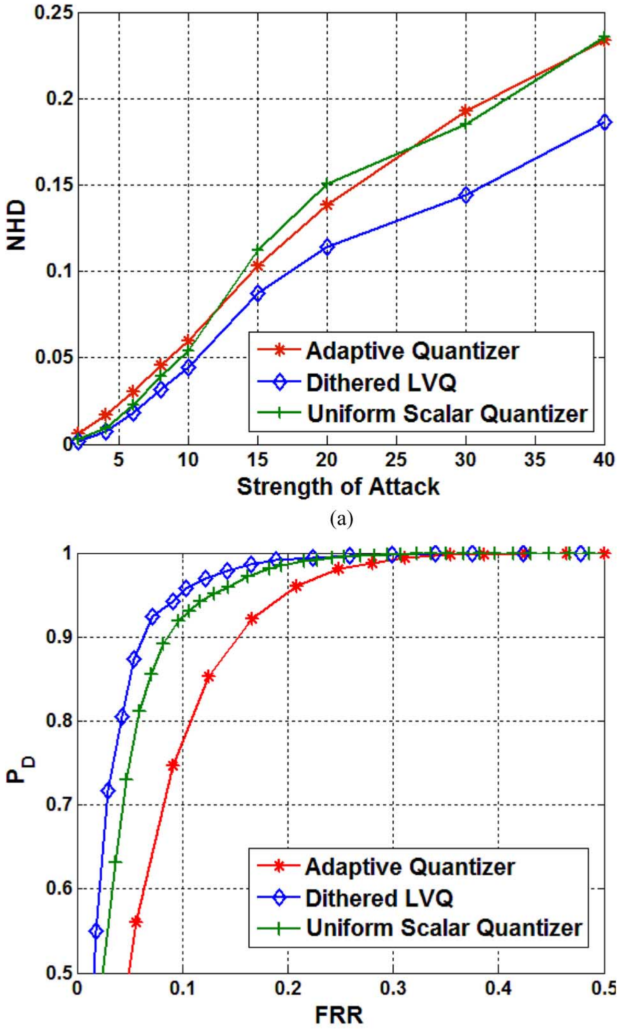


Fig. 13. Performance comparison between quantization schemes: (a) Robustness. (b) Overall performance.

2) Randomness Evaluation for Key-Dependent Quantizers: Both of the dithered LVQ and the adaptive quantizer are key-dependent random quantizers. The randomness of these two quantizers has been investigated in Section III with the entropy metric proposed in [18]. Here, the randomness of key-dependent quantizers is further evaluated by measuring the amount of the perturbation introduced to quantization results.

In the dithered LVQ, the feature vector could be quantized to a number of possible lattice points under different dither vectors. As mentioned in Section III, a random vector $\boldsymbol{r}$ whose components are independently and randomly drawn from the uniform distribution on $[-c, c)$, $(c > 0)$, is involved in generating the dither vector. Accordingly, the perturbation on quantization results imposed by the dithered LVQ is determined by the value of c. Simulations are performed to measure the amount of perturbation with varying c from 1 to 20 by a step size of 1. For each value of c, 100 dither vectors are randomly generated to quantize the same set of features, and the average NHD between the output hash values is calculated. The plot of average NHD versus c is shown in Fig. 14. We observe that the average NHD rapidly increases with c when $c < 7$. Once c exceeds 7, the average NHD maintains to be close to 0.5, which indicates that the same set of features could be mapped to statistically independent hash values under different dither vectors. It implies that a large value of c may benefit the randomness of the quantizer, and $c > 7$ could be the suitable choice.

For comparison purpose, the amount of the perturbation introduced by the adaptive quantizer is also measured. The amount of the perturbation on quantization indexes is determined by the portion of the random region in each quantization interval that is denoted by r. As mentioned in [18], $r \leqslant 0.5$; hence, the maximum amount of perturbation can be achieved when $r = 0.5$. The same set of features are quantized by the adaptive quantizer with $r = 0.5$ under 100 different keys, respectively. It has been observed that the average NHD between the output hash values is 0.15, which indicates that the dithered LVQ can introduce a higher amount of perturbation to quantization results than the adaptive quantizer.

3) Impact of Quantizers' Parameter Setting on Algorithm Performance: All the aforementioned quantizers contain several parameters, such as the average step size of the SQ, the lattice dimensionality, and parameter c of the dithered LVQ. Here,

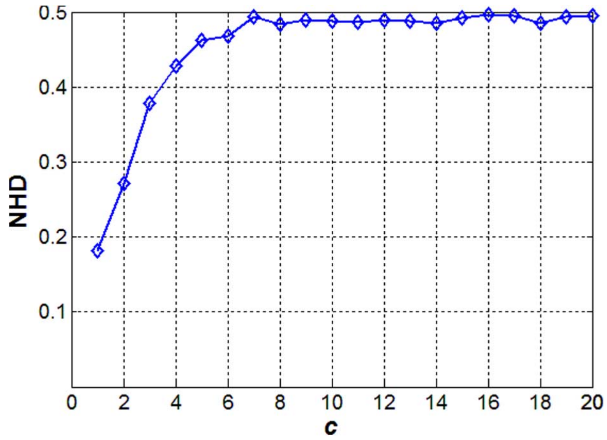


Fig. 14. Perturbation on the output hash imposed by dithered LVQ under different values of c.

simulations are carried out to demonstrate the impacts of these parameters on the performance of the hashing algorithm.

The performance of an SQ in robust hashing depends on its average step size. The increase in the average step size can lead to the improvement of its robustness and the degradation of its discrimination, and vice versa. We take the uniform quantizer in the range of $[-8, 8)$ as the example to show the relationship between the average step size and the quantization performance, and the number of quantization levels is set to 4, 8, 16, and 32, respectively. As before, the performance of the quantizer is evaluated by selecting the average filtering as the distortion. The NHD and ROC curves of the quantizers with different parameter settings are presented in Fig. 15. The NHD curves show that the robustness of the quantizer degrades as the number of quantization levels increases, while the overall performance of the quantizer is nearly unaffected since the ROC curves of the quantizers with different parameters almost coincide with each other. It can be inferred that the changes on robustness and discrimination are counteracted in the overall performance of the quantizer.

The analytical results in Section III-C reveal that the quantization performance of a D_n-based LVQ in robust hashing is independent of the dimensionality of the lattice. Simulations are conducted here to confirm this observation, where a series of D_n lattices with $n = 5, 8$ and 10 are used to build the quantizer, respectively. The performance of these lattice quantizers are compared with that of the D_4-based quantizer presented in Fig. 13. The experimental results confirm that, in spite of their differences on dimensionality, those D_n-based lattice quantizers exhibit exactly the same performance since their NHD and ROC curves are identical with those shown in Fig. 13.

As previously discussed, the randomness of the dithered LVQ is determined by the value of parameter c, and a large c value could introduce a sufficient amount of perturbation on the output hash. Here, our focus is placed on the impact of c on the overall performance of the hashing algorithm. We use the average filtering to produce the distortion and then plot the ROC curves under $c = 1, 4, 10$ and 20, respectively. For comparison purposes, all the ROC curves are plotted in the same figure. As in Fig. 16, although the curves are shown in a high magnification level, it is still difficult to distinguish among the curves. It can be seen that the overall performance is independent of c, with some further explanations as follows. The decision process of robust hashing in identifying the perceptual similarity between the query and reference images can be formulated as the following hypothesis testing problem, where H_0 and H_1 correspond to the cases that the query image is perceptually identical and distinct with the reference one. Let $\boldsymbol{Y}$ denote the dithered feature of the query image, $\boldsymbol{F}$ denote the feature of the reference image, $\boldsymbol{N}$ denote the noise introduced by content-preserving distortions, $\boldsymbol{G}$ denote the feature of the image that is perceptually distinct with the reference one, and $\boldsymbol{R}$ denote the dither vector. Then

$$\begin{aligned} H_0 &: \boldsymbol{Y} = \boldsymbol{F} + \boldsymbol{N} + \boldsymbol{R} \\ H_1 &: \boldsymbol{Y} = \boldsymbol{G} + \boldsymbol{R}. \end{aligned} \quad (30)$$

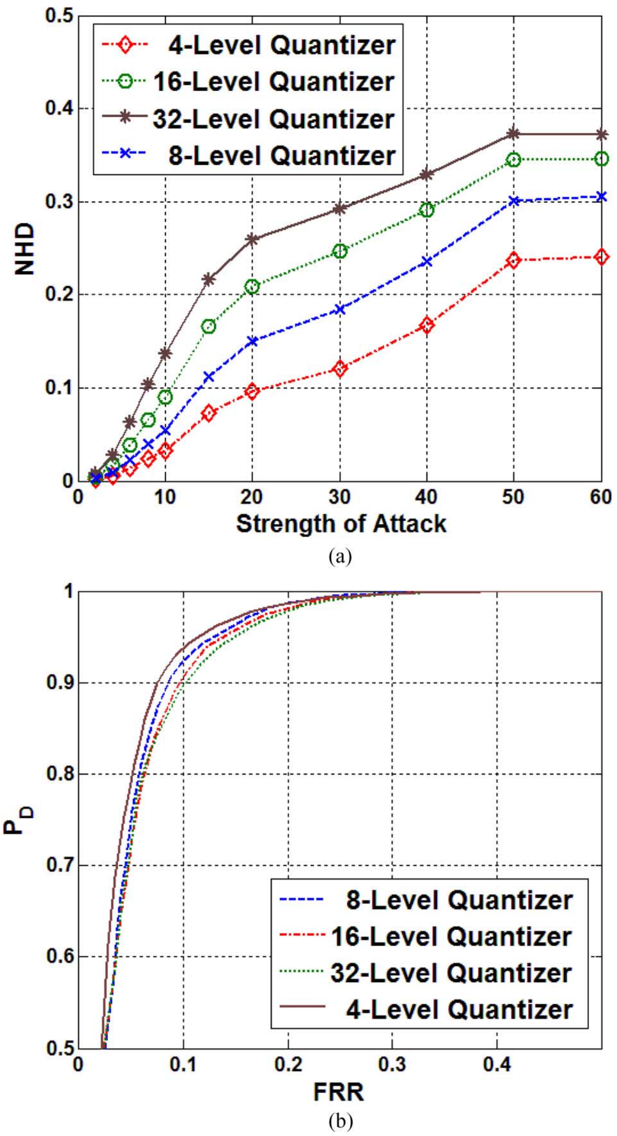


Fig. 15. Performance comparison between the SQs with different numbers of quantization levels. (a) Robustness. (b) Overall performance.

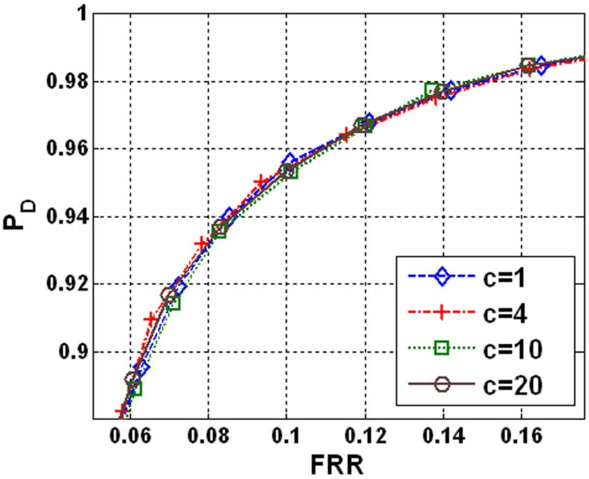


Fig. 16. ROC curves (magnified) under different values of c.

Since $\boldsymbol{R}$ is independent of $\boldsymbol{F}$, $\boldsymbol{N}$, and $\boldsymbol{G}$, the dithering process will not affect the correct detection probability P_D and the FRR of the hashing algorithm. As a result, the ROC curve is independent of $\boldsymbol{R}$ or c.

V. Conclusion

In this paper, we have presented a robust-hash function that employs the random Gabor filtering and the dithered LVQ. A random and rotation-invariant filter has been developed for feature extraction. It has been observed that the extracted features show remarkable robustness to rotation manipulations. Experimental results have demonstrated that the proposed paper also exhibits better statistical performance compared with some representative hashing algorithms. Moreover, the dithered LVQ that can produce satisfactory statistical and security performance has been employed as the quantizer for robust hashing. The robustness of different quantization schemes has been examined. As it has been validated by the simulation and analytical results, LVQ is superior to the adaptive quantizer and the uniform SQ in robust hashing.

References

[1] M. Stamp, *Information Security: Principles and Practice*, 2nd ed. Hoboken, NJ: Wiley, 2006.

[2] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. IEEE Conf. Image Process.*, Sep. 2000, vol. 3, pp. 664–666.

[3] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. Conf. Music Inf. Retrieval*, Oct. 2002, vol. 32, pp. 107–115.

[4] B. Coskun, B. Sankur, and N. Memon, "Spatio-temporal transform based video hashing," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1190–1208, Dec. 2006.

[5] P. J. O. Doets and R. L. Lagendijk, "Distortion estimation in compressed music using only audio fingerprints," *IEEE Trans. Speech Audio Lang. Process.*, vol. 16, no. 2, pp. 302–317, Feb. 2008.

[6] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3452–3465, Nov. 2006.

[7] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," in *Proc. IEEE Conf. Inf. Technol., Coding Comput.*, Mar. 2000, pp. 178–183.

[8] J. C. Oostveen, T. Kalker, and J. Haitsma, "Visual hashing of digital video: Applications and techniques," in *Proc. SPIE Appl. Dig. Image Process. XXIV*, Jul. 2001, vol. 4472, pp. 121–131.

[9] S. J. Xiang, H. J. Kim, and J. W. Huang, "Histogram-based image hashing scheme robust against geometric deformations," in *Proc. 9th Workshop Multimedia Security*, Dallas, TX, Sep. 2007, pp. 121–128.

[10] H. Kevin, S. Martin, and X. B. Zhou, "Histogram-based perceptual hashing for minimally changing video sequences," in *Proc. IEEE Conf. Autom. Production Cross Media Content Multi-Channel Distrib.*, Dec. 2006, pp. 236–241.

[11] M. K. Mihcak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *Proc. ACM Workshop Security Privacy Digit. Rights Manage.*, Philadephia, PA, Nov. 2001, vol. 2320, pp. 13–21.

[12] M. P. Queluz, "Towards robust, content based techniques for image authentication," in *Proc. IEEE Workshop Multimedia Signal Process.*, Dec. 1998, pp. 297–302.

[13] S. S. Kozat, R. Venkatesan, and M. K. Mihçak, "Robust perceptual image hashing via matrix invariants," in *Proc. IEEE Conf. Image Process.*, Oct. 2004, vol. 5, pp. 3443–3446.

[14] V. Monga and M. K. Mihcak, "Robust and secure image hashing via non-negative matrix factorizations," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 376–390, Sep. 2007.

[15] F. X. Yu and Z. M. Lu, "A DCT-VQ based multipurpose image hashing scheme for copyright protection and content authentication," *Int. J. Innov. Comput. Inf. Control*, vol. 5, no. 9, pp. 2703–2710, Sep. 2009.

[16] S. Roy and Q. Sun, "Robust hash for detecting and localizing image tampering," in *Proc. IEEE Conf. Image Process.*, Sep. 2007, vol. 6, pp. 117–120.

[17] S. S. Jin, J. Haitsma, T. Kalker, and C. D. Yoo, "A robust image fingerprinting system using the Radon transform," *Signal Process. Image Commun.*, vol. 19, no. 4, pp. 325–339, Apr. 2004.

[18] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230, Jun. 2006.

[19] C. D. Roover, C. D. Vleeschouwer, F. Lefèbvre, and B. Macq, "Robust video hashing based on radial projections of key frames," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 4020–4037, Oct. 2005.

[20] M. K. Mihcak and R. Venkatesan, "A perceptual audio hashing algorithm: A tool for robust audio identification and information hiding," in *Proc. Workshop Inf. Hiding*, Pittsburgh, PA, Apr. 2001, vol. 2137, pp. 51–65.

[21] V. Monga, A. Banerjee, and B. L. Evans, "A clustering based approach to perceptual image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 68–79, Mar. 2006.

[22] E. P. McCarthy, F. Balado, G. C. M. Silvestre, and N. J. Hurley, "A model for improving the performance of feature extraction based robust hashing," in *Proc. SPIE—Security, Steganography, and Watermarking of Multimedia Contents VII*, Jan. 2005, vol. 5681, pp. 59–67.

[23] Y. Mao and M. Wu, "Unicity distance of robust image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 462–467, Sep. 2007.

[24] Q. Li and S. Roy, "On the security of non-forgeable robust hash functions," in *Proc. IEEE Conf. Image Process.*, Oct. 2008, pp. 3124–3127.

[25] W. Li and B. Preneel, "Attacking some perceptual image hash algorithms," in *Proc. IEEE Conf. Multimedia Expo*, Jul. 2007, pp. 879–882.

[26] M. Jin and C. D. Yoo, "Quantum hashing for multimedia," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 4, pp. 982–994, Dec. 2009.

[27] F. Khelifi and J. M. Jiang, "Perceptual image hashing based on virtual watermark detection," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 981–994, Apr. 2010.

[28] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Hash-based identification of sparse image tampering," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2491–2504, Nov. 2009.

[29] J. K. Kamarainen, V. Kyrki, and H. Kälviäinen, "Noise tolerant object recognition using Gabor filtering," in *Proc. Conf. Digit. Signal Process.*, Jul. 2002, vol. 2, pp. 1349–1352.

[30] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, no. 7, pp. 1160–1169, Jul. 1985.

[31] R. Porter and N. Canagarajah, "Robust rotation-invariant texture classification: wavelet, Gabor filter and GMRF based schemes," *Proc. Inst. Elect. Eng.—Vis. Image Signal Process.*, vol. 144, no. 3, pp. 180–188, Jun. 1997.

[32] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. New York: Springer-Verlag, 1999.

[33] A. Gersho, "On the structure of vector quantizers," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 157–166, Mar. 1982.

[34] A. Kirac and P. P. Vaidyanathan, "Results on lattice vector quantization with dithering," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 12, pp. 811–826, Dec. 1996.

[35] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 227–232, Mar. 1982.

[36] J. Li, "Photography Image Database." [Online]. Available: http://www.stat.psu.edu/ jiali/index.download.html

[37] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "ImageNet: a large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

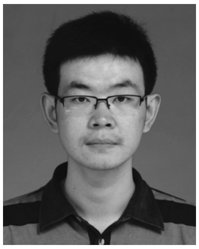
[38] Databse for Object and Concept Recognition [Online]. Available: http://www.cs.washington.edu/research/imagedatabase/groundtruth/

[39] L. W. Kang, C. S. Lu, and C. Y. Hsu, "Compressive sensing-based image hashing," in *Proc. IEEE Conf. Image Process.*, Nov. 2009, pp. 1285–1288.

[40] D. Wu, X. B. Zhou, and X. M. Niu, "A novel image hash algorithm resistant to print-scan," *Signal Process.*, vol. 89, no. 12, pp. 2415–2424, Dec. 2009.

[41] B. Coskun and N. Memon, "Confusion/diffusion capabilities of some robust hash functions," in *Proc. Conf. Inf. Sci. Syst.*, Mar. 2006, pp. 1188–1193.

[42] G. Schaefer and M. Stich, "UCID—An uncompressed colour image database," in *Proc. SPIE—Storage and Retrieval Methods and Applications for Multimedia*, Jan. 2004, pp. 472–480.



Yuenan Li received the B.S. and M.S. degrees in measuring technology and instruments and the Ph.D. degree in information and communication engineering from Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2010, respectively.

He is currently a Lecturer with the School of Electronic and Information Engineering, Tianjin University, Tianjin, China. His current research interests include multimedia signal processing, information security, and forensics.

Zheming Lu (M'03–SM'06) received the B.S. and M.S. degrees in electrical engineering, and the Ph.D. degree in measuring technology and instruments from Harbin Institute of Technology (HIT), Harbin, China, in 1995, 1997, and 2001, respectively.

In 1999, he became a Lecturer with HIT. In 2003, he was a Professor with the Department of Automatic Test and Control, HIT. He is currently a Professor with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China. He has published more than 240 papers and six monographs in Chinese and one monograph in Springer in the areas of multimedia signal processing and information hiding. His current research interests include multimedia signal processing, information security, and complex networks.

Ce Zhu (M'03–SM'04) received the B.S. degree from Sichuan University, Chengdu, China, in 1989 and the M.Eng. and Ph.D. degrees from Southeast University, Nanjing, China, in 1992 and 1994, respectively, all in electronic and information engineering.

He pursued postdoctoral research with the Chinese University of Hong Kong, the City University of Hong Kong, and the University of Melbourne, Melbourne, Australia. He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He has held visiting positions with Queen Mary, University of London, London, U.K., and Nagoya University, Melbourne, Japan, supported by Tan Chin Tuan Exchange Fellowship and the Scientist Exchange Program of the National University of Singapore/Nanyang Technological University–Japan Society for the Promotion of Science, respectively. His research interests include image/video coding, streaming and processing, 3-D video, joint source-channel coding, multimedia systems, and applications.

Dr. Zhu serves as an associate editor of the IEEE TRANSACTIONS ON BROADCASTING and the IEEE SIGNAL PROCESSING LETTERS and as an area editor of *Signal Processing: Image Communication*. He has served on technical/program committees, organizing committees and as track/session chairs for over 40 international conferences. He is a member of the Technical Committee on Multimedia Systems and Applications of the IEEE Circuits and Systems Society, and a voting member of Multimedia Communications Technical Committee of the IEEE Communications Society. He was a recipient of the 2010 Special Service Award from IEEE Broadcast Technology Society. His coauthored paper (with his student Y. Xu) won the Best Student Paper Award at the Seventh International Mobile Multimedia Communications Conference (MobiMedia 2011, Italy).

Xiamu Niu (M'03) received the B.S. and M.S. degrees in communication and electronic engineering, and the Ph.D. degree in measuring technology and instrument, in 1982, 1989, and 2000, respectively, all from Harbin Institute of Technology (HIT), China. From 2000 to 2002, he was an invited scientist and staff member in the Department of Security Technology for Graphics and Communication System, Fraunhofer Institute for Computer Graphics, Germany.

He is currently a Professor with the School of Computer Science, HIT. He is also the director of the Information Countermeasure Institute (ICT) at HIT. His current research interests include signal processing and information security.