Robust and Secure Image Hashing

Ashwin Swaminathan, *Student Member, IEEE*, Yinian Mao, *Student Member, IEEE*, and Min Wu, *Member, IEEE*

***Abstract*—Image hash functions find extensive applications in content authentication, database search, and watermarking. This paper develops a novel algorithm for generating an image hash based on Fourier transform features and controlled randomization. We formulate the robustness of image hashing as a hypothesis testing problem and evaluate the performance under various image processing operations. We show that the proposed hash function is resilient to content-preserving modifications, such as moderate geometric and filtering distortions. We introduce a general framework to study and evaluate the security of image hashing systems. Under this new framework, we model the hash values as random variables and quantify its uncertainty in terms of differential entropy. Using this security framework, we analyze the security of the proposed schemes and several existing representative methods for image hashing. We then examine the security versus robustness tradeoff and show that the proposed hashing methods can provide excellent security and robustness.**

***Index Terms*—Differential entropy, image authentication, image hashing, multimedia security.**

I. Introduction

In THE information era, the increasing availability of multimedia data in digital form has led to a tremendous growth of tools to manipulate digital multimedia. To ensure trustworthiness, multimedia authentication techniques have emerged to verify content integrity and prevent forgery [1], [2]. Traditionally data integrity issues are addressed by cryptographic hashes or message authentication functions, which are key dependent and sensitive to every bit of the input message. As a result, the message integrity can be validated when every bit of the message is unchanged [3]. While this sensitivity usually meets the need to authenticate text messages, the definition of authenticity for multimedia is not as straightforward. Multimedia data can allow for lossy representations with graceful degradation. The information carried by media data is mostly retained even when the multimedia has undergone moderate levels of filtering, geometric distortion, or noise corruption. Therefore, bit-by-bit verification is no longer a suitable way to authenticate multimedia data, and a media authentication tool that validates the content is more desired.

Manuscript received May 31, 2005; revised Feburary 20, 2006. This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N00014-05-10634 and in part by the U.S. National Science Foundation under CAREER Award CCR-0133704. Preliminary results of this work were presented in IEEE International Workshop on Multimedia Signal Processing, Siena, Italy, 2004 [14] and at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, 2005 [15]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Thomas Johansson.

The authors are with the Department of Electrical and Computer Engineering and the Institute of Advanced Computing Studies, University of Maryland, College Park, MD 20742 USA (e-mail: ashwins@eng.umd.edu; ymao@eng.umd.edu; minwu@eng.umd.edu).

Digital Object Identifier 10.1109/TIFS.2006.873601

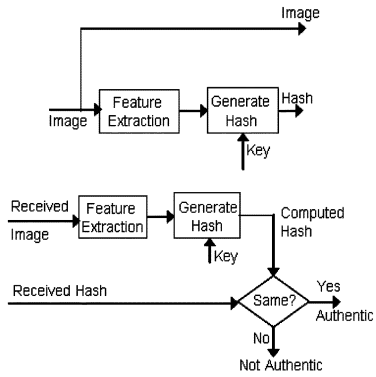


Fig. 1. Hash functions for image authentication.

A number of media-specific hash functions have been proposed for multimedia authentication [4]–[7]. A multimedia hash is a content-based digital signature of the media data. To generate a multimedia hash, a secret key is used to extract certain features from the data. These features are further processed to form the hash. The hash is transmitted along with the media either by appending or embedding it to the primary media data. At the receiver side, the authenticator uses the same key to generate the hash values, which are compared to the ones transmitted along with the data for verifying its authenticity. This process is illustrated in Fig. 1.

In addition to content authentication, multimedia hashes are used in content-based retrieval from databases [8]. To search for multimedia content, naïve methods such as sample-by-sample comparisons are computationally inefficient. Moreover, these methods compare the lowest level of content representation and do not offer robustness in such situations as geometric distortions. Robust image hash functions can be used to address this problem [4]. A hash is computed for every data entry in the database and stored with the original data in the form of a lookup table. To search for a given query in the database, its hash is computed and compared with the hashes in the lookup table. The data entry corresponding to the closest match, in terms of certain hash-domain distance that often accounts for content similarity, is then fetched. Since the hash has much smaller size with respect to the original media, matching the hash values is computationally more efficient.

Image hash functions have also been used in applications involving image and video watermarking. In nonoblivious image watermarking, the need for the original image in watermark extraction can be substituted by using hash as side information [1], [9], [10]. The hash functions have also been used as image-dependent keys for watermarking [11]. In video watermarking, it has been shown that adversaries can employ "collusion attacks" to devise simple statistical measures to estimate the watermark if they have access to multiple copies of similar frames

[12]. A solution to this problem is to use secure, content-dependent hash values as a key to generate the watermark [13].

There are two important design criteria for image hash functions, namely, robustness and security [4], [13]–[15]. By robustness, we mean that when the same key is used, perceptually similar images should produce similar hashes. Here, the similarity of hashes is measured in terms of some distance metric, such as the Euclidean or Hamming distance. In this work, we consider two images to be similar if one image can be obtained from the other through a set of content-preserving manipulations. This set of manipulations includes moderate levels of additive noise, JPEG compression, geometric distortions (such as the common rotation, scaling, and translation operations, or more generally, affine transformations), cropping, filtering operations (such as spatial averaging and median filtering), and watermark embedding.

The security of image hash functions is introduced by incorporating a secret key in generating the hash. Without the knowledge of the key, the hash values should not be easily forged or estimated. Additionally, some design criteria for generic data hash also apply to image hash functions, namely, the one-way and collision-free properties. A hash is one way if given a hash h and a hash function $g(\cdot)$, it is computationally expensive to find an image I such that $h = g(I)$. Collision-free property refers to the fact that given an image I and a hash function $g(\cdot)$, it is computationally hard to find a second image $\hat{I}$ such that $g(I) = g(\hat{I})$. Although some generic data hash functions, such as MD5, satisfy these criteria [3], they are highly dependent on every bit (or pixel) of the input data rather than on the content. Hence, most of the them are not suitable for the emerging multimedia applications and the need for building robust and secure image hash is paramount.

In this paper, we introduce a new method to construct robust and secure image hash functions. Our proposed method is based on the rotation invariance of the Fourier–Mellin transform and controlled randomization during image feature extraction. We show that the proposed scheme is robust to geometric distortions, filtering operations, and various content-preserving manipulations. We then present a new framework to study the security aspects of existing image hashing schemes. We propose to evaluate the security from an information theoretic perspective by measuring the amount of randomness in the hash vector using the differential entropy as a metric. We show that the suggested security evaluation framework is generic and can be used to analyze and compare the security of several classes of image hashing algorithms. We derive analytical expressions of security using an entropy-based metric for several representative image hashing schemes and demonstrate that the proposed hashing algorithm is more secure in terms of this metric. Finally, we use the proposed security metric to discuss the tradeoffs between robustness and security that are exhibited in most existing image hashing algorithms.

The rest of this paper is organized as follows. In Section II, we introduce the general framework for image hashing. We then present the proposed image hashing scheme and compare its performance with several existing schemes in Section III. We evaluate the security for a number image hashing schemes in Section IV. Finally, the discussions and concluding remarks are provided in Sections V and VI.

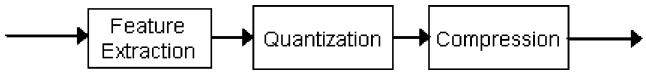


Fig. 2. Three-step framework for generating a hash.

II. General Framework and Prior Art

To achieve robustness and security in image hashing, most of the existing schemes follow a three-step framework to generate a hash. As shown in Fig. 2, these three steps include:

Step 1) generating a key-dependent feature vector from the image;
Step 2) quantizing the feature vector;
Step 3) compressing the quantized vector.

The most challenging part of this framework has been the feature extraction stage [4], [16], [17]. A typical approach is to extract image features that are invariant allowed content-preserving image processing operations [13], [18], [19], [22], [23]. These features are then used to generate the hash values. Some of the features that have been proposed in the literature include block-based histograms [24]–[26], image-edge information [27], relative magnitudes of the discrete cosine transform (DCT) coefficients [28], and the scale interaction model with the Mexican-Hat wavelets [29]. However, since these features are publicly known, using such features alone makes the scheme susceptible to forgery attacks [13], even when the final hash is obtained by encrypting these features [28], [29]. This is because the attacker may create a new image with different visual content, while still preserving the feature values. As the resulting hash will be the same, such hashing approaches may lead to misclassifications in database applications, and would also be vulnerable to counterfeiting attacks in authentication applications. Therefore, the security mechanism should be combined into the feature extraction stage.

By jointly considering security and robustness, Fridrich *et al.* propose to generate image hash by projecting an input image onto zero-mean random smooth patterns, generated using a secret key [13]. While the resulting hash is resilient to filtering operations, it does not perform very well for geometric distortions and is not collision free as shown in [30]. In [4], Venkatesan *et al.* use the principal values calculated from the wavelet transform of the image blocks to generate a feature vector invariant to general gray-scale operations. The resulting features are then randomly quantized and compressed to produce the final hash [5]. Recently, it has been shown that this scheme does not perform well for some manipulations, such as contrast changes, gamma correction, and object insertion [31]. An iterative key-dependent image hash based on repeated thresholding and spatial filtering was proposed in [16]. All of these algorithms [4], [13], [16] described above perform well under additive noise and common filtering operations, but not under desynchronization and geometric distortions. Considering these disadvantages, the radon soft hash algorithm (RASH), based on the properties of the radon transform, was proposed in [17] and [19]. Recently, other transform domain features have been employed for perceptual hashing. Features obtained from the singular value decomposition (SVD) of pseudorandomly chosen regions of the image [20] and Randlet

transform coefficients [21] have been shown to have good robustness properties especially for rotation and cropping attacks.

To enable fast comparison and searches, it is usually preferred that the final hash be a short sequence of bits rather than a set of real numbers. Therefore, the output of the feature extraction stage is usually quantized, converted to binary representation, and further compressed. Uniform, Lloyd–Max, or key-dependent randomized quantizers have been used for hash quantization [4], [5]; and the decoding stages of error-correcting codes have been used for compressing the quantized hash [4], [32], [33]. These methods reduce the length of the hash vector; yet preserve the Hamming distance. Some works also secure the compression stage by performing a key-dependent random selection from the quantized hash values [5].

Since the feature extraction stage is the most important stage in the general image hashing framework, we will investigate the feature extraction stage in greater detail in this paper. We design a randomized hashing scheme and examine its performance in terms of robustness and security.

III. Image Hashing Algorithms Based on Polar Fourier Transform

In this section, we present the proposed image hashing algorithm. Our proposed scheme is based on the Fourier–Mellin transform, which has been shown to be invariant to two-dimensional (2-D) affine transformations [34]–[36]. We incorporate key-dependent randomization into the Fourier–Mellin transform outputs to form secure and robust image hash.

A. Underlying Robustness Principle of the Proposed Algorithm

Consider an image $i(x,y)$ and its 2-D Fourier transform $I(f_x,f_y)$, where f_x and f_y are the normalized spatial frequencies in the range [0,1]. We denote a rotated, scaled and translated version of the $i(x,y)$ as $i'(x,y)$. We can relate them as

$$i'(x,y) = i\left(\sigma(x\cos\alpha + y\sin\alpha) - x_0,\ \sigma(-x\sin\alpha + y\cos\alpha) - y_0\right) \quad (1)$$

where the rotation, scaling, and translation (RST) parameters are α, σ, and (x_0,y_0), respectively. The magnitude of the 2-D Fourier transform of $i'(x,y)$ can be written as

$$|I'(f_x,f_y)| = |\sigma|^{-2}\left|I\left(\sigma^{-1}(f_x\cos\alpha + f_y\sin\alpha),\ \sigma^{-1}(-f_x\sin\alpha + f_y\cos\alpha)\right)\right|. \quad (2)$$

Consider now a polar coordinate representation in the Fourier transform domain, that is $f_x = \rho\cos\theta$ and $f_y = \rho\sin\theta$, where $\rho \in [0,1]$ is the normalized radius and $\theta \in [0,2\pi)$ is the angle parameter. The (2) can be written using polar coordinates as

$$|I'(\rho,\theta)| = |\sigma|^{-2}\left|I(\rho\sigma^{-1},\theta-\alpha)\right|. \quad (3)$$

In (3), we observe that the magnitude of the Fourier transform is independent of the translational parameters (x_0,y_0). Observing that a rotation in image domain leads to a rotation by the same amount in the Fourier transform domain, we integrate the transform magnitude $|I'(\rho,\theta)|$ along a circle centered at zero frequency with a fixed radius ρ to obtain

$$h(\rho) = \int_0^{2\pi} |I'(\rho,\theta)|\,d\theta \approx \int_0^{2\pi} |I(\rho,\theta-\alpha)|\,d\theta \approx \int_0^{2\pi} |I(\rho,\theta)|\,d\theta. \quad (4)$$

These properties of the Fourier transform enable us to construct robust features. In the next subsection, we present the detailed steps of the proposed algorithms.

B. Basic Steps of the Proposed Algorithms

The basic steps of the proposed algorithm include preprocessing, feature generation, and post processing.

1) Preprocessing: We first apply a low-pass filter on the input image and downsample it. We then perform histogram equalization on the downsampled image to get $i(x,y)$. We take a Fourier transform on the preprocessed image to obtain $I(f_x,f_y)$. The Fourier transform output is converted into polar coordinates to arrive at $I'(\rho,\theta)$ as in (3).

2) Feature Generation: We sum up $I'(\rho,\theta)$ along the θ-axis at K equidistant points in the range of $[0,2\pi)$ (i.e., for $\theta \in \{\pi/K, 3\pi/K, \ldots, (2K-1)\pi/K\}$) to obtain an image feature vector h_ρ. $K = 360$ is used in our implementation. Since the feature h_ρ is only dependent on the image content, we propose two randomization methods to obtain key-dependent features using h_ρ:

- **Scheme 1:**
  We obtain $|I'(\rho,\theta)|$ as in (3) and compute a weighted sum along the θ-axis to obtain the jth hash value

$$h_j = \sum_{i=0}^{K-1} \beta_{\rho_j,i}\left|I'\left(\rho_j, \frac{(2i+1)\pi}{K}\right)\right| \quad (5)$$

  where $\{\beta_{\rho_j,i}\}$ are key-dependent pseudorandom numbers that are normally distributed with mean m and variance σ^2.
- **Scheme 2:**
  We first use a secret key to generate random sets of radii $\{\Gamma_j\}$. We then take $|I'(\rho,\theta)|$ obtained in (3) and do a summation along the θ-axis for each radii in this set. A random linear combination of the resulting summations gives the jth hash value. This can be represented as

$$h_j = \sum_{\rho\in\Gamma_j} \beta_\rho \sum_{i=0}^{K-1}\left|I'\left(\rho, \frac{(2i+1)\pi}{K}\right)\right| \quad (6)$$

  where β_ρ are key-dependent pseudorandom numbers that are normally distributed with mean m and variance σ^2. This method is illustrated in Fig. 3.

3) Post Processing: We quantize the resulting statistics vector and apply gray coding to obtain the binary hash sequence [37]. This bit sequence is then passed through the decoding stage of an order-3 Reed–Muller decoder for compression [5]. This step may also be replaced with the Wyner–Ziv decoder

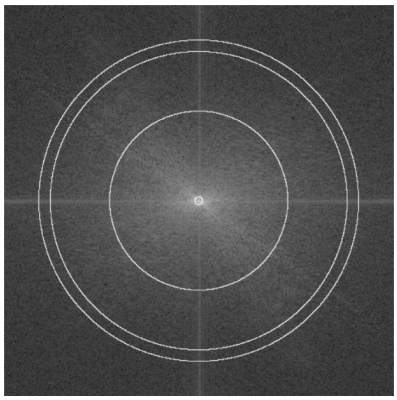


Fig. 3. 2-D Fourier transform of the Lena image. The jth hash value-h_j, is obtained by a random weighted summation along the circumference of chosen radii $\rho \in \Gamma_j$ in scheme-2. Some of the constant radii circles used in the summation are displayed in the figure. The magnitude of the Fourier transform is shown in the log scale and has been appropriately scaled for display purposes.

[32], [38]. Furthermore, we can enhance the security of the hash by making the quantization and compression stages key dependent. For example, randomized quantization algorithms may be used to quantize the hash [5]; for the compression stage, we can randomly select the hash values from the quantized hash vector [16] or randomly choose the order of the Reed–Muller decoder used for different subsections of the hash. These techniques would further enhance the security of the resultant hash vector. Finally, the compressed hash is randomly permuted according to a permutation table generated using the key.

C. Performance Study and Comparison

1) Performance Metrics and Experiment Setup: To measure the performance of image hashing, we choose the Hamming distance between the binary hashes, normalized it with respect to the length (L) of the hash as a performance metric. The normalized Hamming distance is defined as

$$d(h_1, h_2) = \frac{1}{L} \sum_{k=1}^{L} |h_1(k) - h_2(k)| \tag{7}$$

which is expected to be close to 0 for similar images and close to 0.5 for dissimilar ones. As more parts of a picture are changed, the manipulated image and the original image become more dissimilar. For an ideal hashing scheme, the normalized Hamming distance between the corresponding hashes should increase accordingly.

We test the proposed schemes on a database of around 157 200 images. In this database, there are 1200 original gray-scale images each of size 512 × 512. This includes around 50 classic benchmark images (such as Lena, Baboon, Pepper, etc.), and a variety of scenery and human activity photos taken by digital cameras. These camera photos were cropped, converted to gray scale, and downsampled to 512 × 512. For each original image in this set, we generate 130 similar versions by manipulating the original image according to a set of content-preserving operations listed in Table I. We measure the normalized Hamming distance between the hashes of the original image and the manipulated images. The results obtained for

TABLE I
SET OF CONTENT-PRESERVING MANIPULATIONS

Manipulation Operation	Parameters of the Operation	Number of Images
Additive Noise		
Gaussian distributed	Variance 0-0.2	10
Uniform distributed	Variance 0-0.5	10
Filtering Operations		
Spatial Averaging	Filter order 2-6	5
Median Filter	Filter order 2-11	10
Wiener Filter	Filter order 2-11	10
Sharpening	Filter order 3-11	5
Geometric Distortions		
Rotation	Degrees 1-20	20
Scaling	Percentage 0.5-1.5	10
Cropping	Percentage 1-30	10
Shearing	Percentage 1-10	10
Random deletion of lines	Percentage 1-20	10
Luminance Non-Linearities		
Gamma correction	$I^{\gamma}, \gamma \in$ [0.75-1.25]	10
JPEG compression	Quality Factor 10-99%	10
Total		130

TABLE II
HASH LENGTHS FOR VARIOUS HASHING SCHEMES

Hashing Method used	Hash Length
Mihçak's Algorithm B [16]	1000
Venkatesan's scheme [4]	805
Fridrich's scheme [13]	420
Proposed Scheme 1	420
Proposed Scheme 2	420

the proposed schemes are compared with three representative existing schemes by Fridrich [13], by Venkatesan *et al.* [4], and by Mihçak [16]. These three schemes are chosen because they adopt different ways to extract the robust image feature as well as different methods to randomize these features. We also consider the normalized Hamming distance between the hashes of dissimilar images, which indicates the discriminative capability of the hashing algorithm. We note that the computed hashes of all these schemes are short in length. For a 512 × 512 image, the hash lengths are on the order of a few hundred bits, as shown in Table II.

2) Experimental Results on Robustness of the Hash: To examine the robustness properties, we consider the performance of various hashing schemes to different content-preserving manipulations such as moderate RST, filtering, and image compression.[1] We show the comparison results in terms of normalized Hamming distance in Figs. 4–8. Our results indicate that the proposed schemes perform well under desynchronization distortions. The performance for rotation and shearing distortions, averaged over the 1200 images, are shown in Fig. 4. In the case of rotation distortions, we observe that the Hamming distance between the quantized feature vectors of the proposed schemes is smaller than those of the existing schemes, especially for a large rotation angle. This is expected since the summation along the θ-axis reduces the effects of rotation. We can also observe that scheme-2 gives better results than scheme-1, in terms of the normalized Hamming distance. This is attributed to the

[1]In all of the experiments, we use our implementation of the hashing methods [4], [13], [16] for the comparison study. Whenever possible, we verified the performance results with the ones reported in this paper. In all cases, the parameters of the hashing algorithms were chosen so as to maintain similar values for the security metric in order to facilitate a fair comparison. Refer to Section IV for details on the security metric.

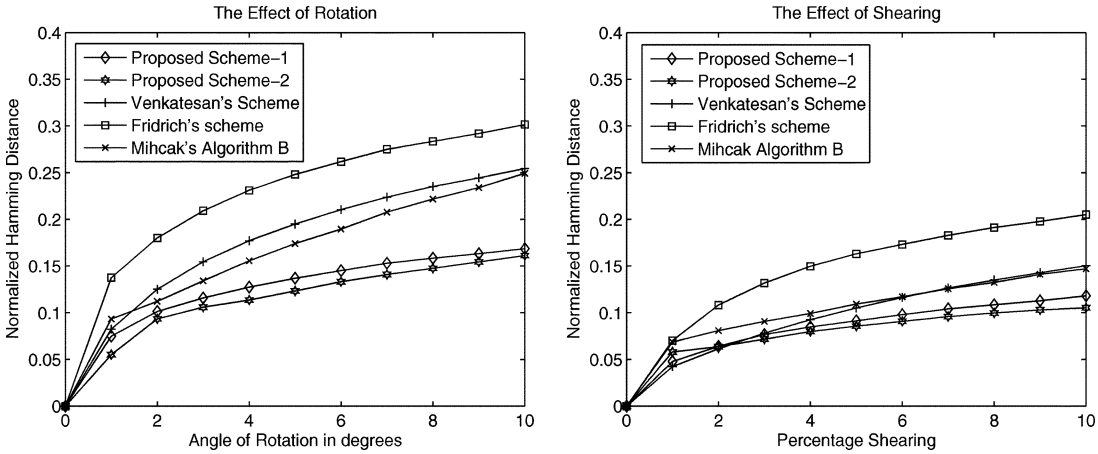


Fig. 4. Performance of various hashing schemes under desynchronization attacks. To generate a point on the curve, the input image was first rotated (or sheared) to give a larger image padded appropriately with zeros. This image was then cropped to exclude the zeros and resized to a predetermined canonical size. The hash of the resulting image was computed and the normalized Hamming distance from the hash of original image is shown in the Y-axis.

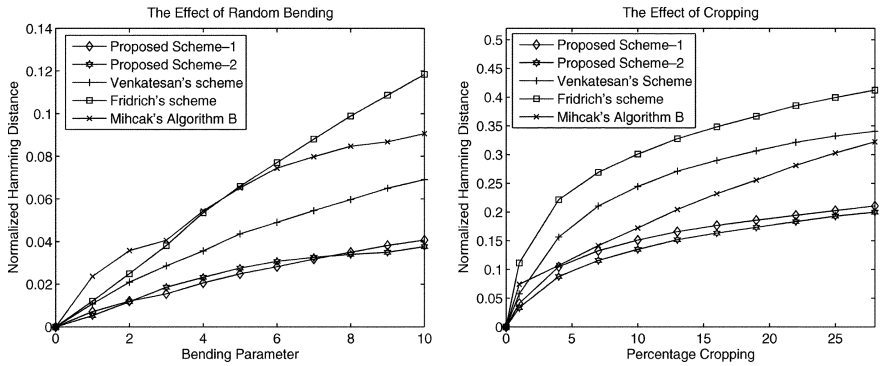


Fig. 5. Performance of various hashing schemes under (a) bending and (b) cropping. Cropped images were obtained by retaining the central portion of the image and removing the boundaries. The cropped image is resized to a predetermined canonical size before computing the hash.

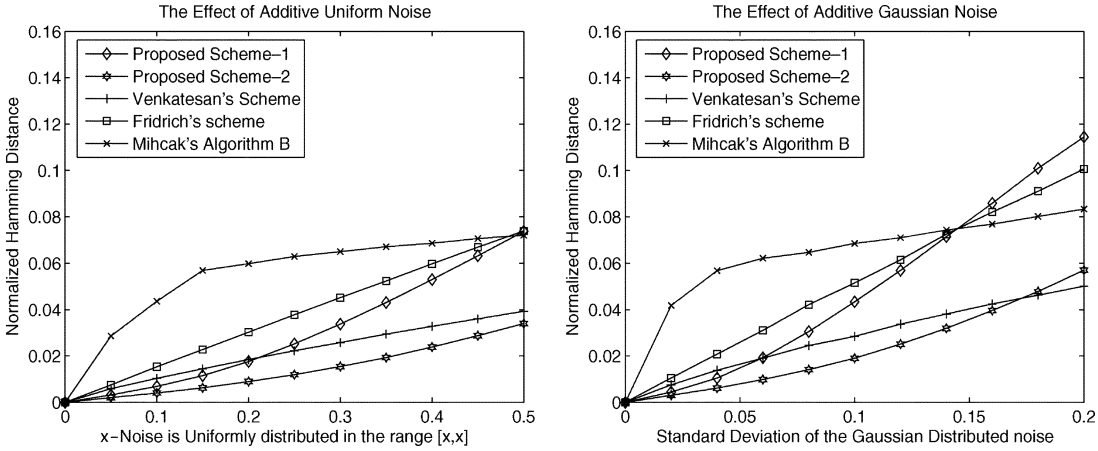


Fig. 6. Performance of various hashing schemes under additive noise. The noisy images were artificially generated by adding uniform/Gaussian distributed noise of different variances to the original image.

fact that performing a weighted sum along the θ-axis as in the proposed scheme-1 no longer preserves rotation invariance. The proposed algorithms also achieve comparable performance with most existing algorithms under shearing distortions. The performance results for random bending [39] and cropping are shown in Fig. 5(a) and (b), respectively. We observe that the proposed schemes perform very well for both of these distortions. This is because the magnitude of the low-frequency coefficients of the Fourier transform that contribute to the hash does not change much under moderate bending and cropping.

We show the performance of the hash algorithms under additive noise in Fig. 6. We observe from the figure that the pro-

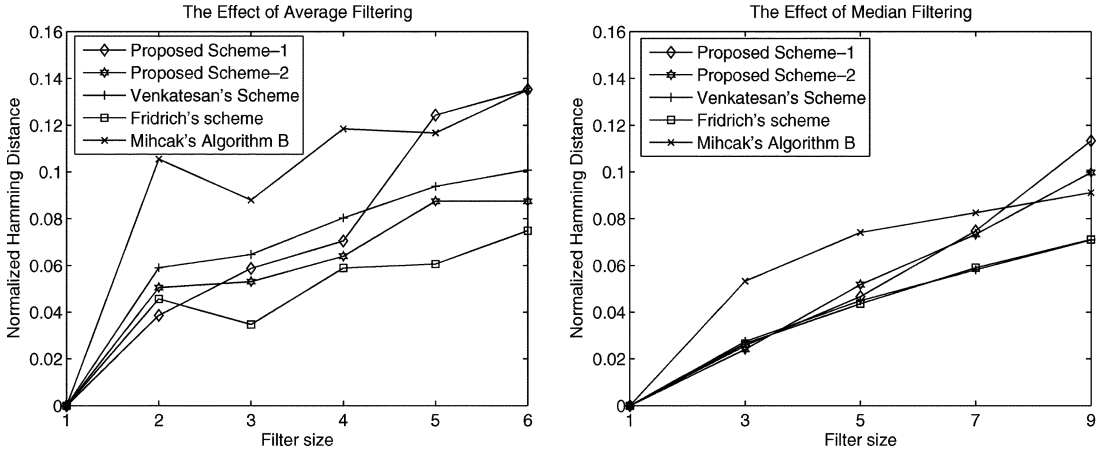


Fig. 7. Performance of various hashing schemes under filtering.

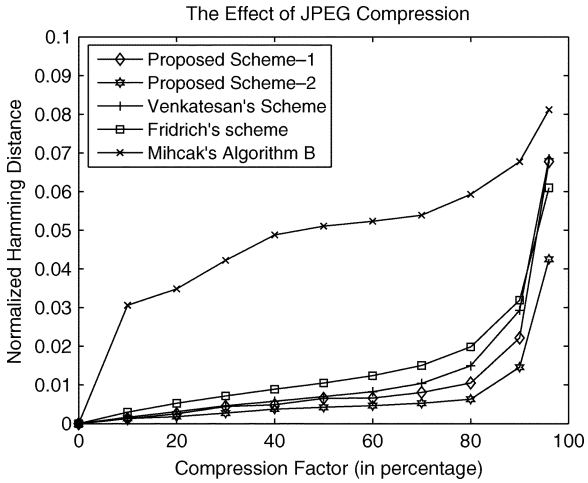


Fig. 8. Performance of various hashing schemes under JPEG compression.

posed scheme-2 does well compared to the proposed scheme-1 and other existing schemes. We further note that the normalized Hamming distance between the hashes of the noisy image and the original image is very small and on the order of 0.02. This performance is attributed to the low-pass filtering in the preprocessing step of the hash generation. The results for filtering and JPEG compression are shown in Figs. 7 and 8. We observe that the performance of the proposed schemes under these distortions is comparable to the existing schemes.

3) Discriminative Capability of Hash: Since image hash should be able to distinguish malicious manipulations from content-preserving ones, its performance in differentiating images with different contents is an important performance aspect. For images with different contents, an ideal hash algorithm should produce two statistically independent binary hash vectors, where half of the hash bits are expected to be the distinct and the other half are the same. This would result in a normalized Hamming distance of around 0.5. Our experiments with a set of 1200 different images indicate that the mean of normalized Hamming distance of the resulting 719 400 combinations was around 0.48. To further demonstrate the performance of the proposed scheme to inauthentic modifications, we consider the following cut-and-paste image editing as shown in Fig. 9, where a new image (c) is created by combining approximately equal parts from image (a) and (b). An ideal image hashing scheme should classify (c) as inauthentic. We perform this test on 500 images and list the normalized Hamming distance between the obtained hash vectors for different algorithms in Table III. We can see from the table that the proposed schemes find the image (c) to have large distances from (a) and (b) and, thus, correctly declare it inauthentic; on the other hand, the existing algorithms suggest a smaller distance and have lower reliability to distinguish (c) from (a) and (b).

4) Image Authentication as a Hypothesis Testing Problem: Generally speaking, the problem of image authentication can be considered as a hypothesis testing problem with the following two hypotheses

- H_0: image is not authentic;
- H_1: image is authentic.

Now, we examine the robustness and discriminative capabilities of various hashing schemes in terms of the receiver operating characteristics (ROC) [40], [41]. The ROC curve characterizes the receiver's performance by classifying the received signal into one of the hypothesis states. For each original image, we compute and store the hash values, which we denote as h_1. Given the received image, we find its hash value h_2 and declare it to be authentic if the normalized Hamming distance between the hashes satisfies $d(h_1, h_2) < \eta$ where η is a decision threshold. Based on ground truth, we record the number that are correctly classified as authentic to give us an estimate of the probability of correct detection (P_D). For a given η, we also record the number of processed versions of other images that are falsely classified as original image and obtain an estimate of the probability of false alarm (P_F). We repeat this process for different decision thresholds η, and arrive at the ROC. The ROC obtained from the experiments using 1200 different images is shown in Fig. 10. We can observe from the ROC curves that the proposed schemes attain a $P_D = 0.95$ when the P_F is 0.05, while the other schemes attain the same P_D when P_F is close to 0.15. Hence, the proposed scheme has a higher probability of correct detection for a given probability of false alarm and therefore achieves better performance. This further demonstrates the advantages of the proposed hashing schemes over the existing schemes.

(a) (b) (c)

Fig. 9 Example of inauthentic manipulations obtained by combining parts of multiple images. (a) and (b) are two original 512 × 512 images. Image (c) is obtained by combining parts of images (a) and (b).

TABLE III
PERFORMANCE OF THE ALGORITHM FOR DISSIMILAR IMAGES UNDER THE TYPE OF MANIPULATION SHOWN IN FIG. 10. HERE, d_{AB} DENOTES THE DISTANCE BETWEEN IMAGES (A) AND (B)

Hashing Method used	d_{AB}	d_{AC}	d_{BC}
Mihçak's Algorithm B [16]	0.50	0.20	0.28
Venkatesan's scheme [4]	0.37	0.15	0.31
Fridrich's scheme [13]	0.41	0.26	0.34
Proposed Scheme 1	0.49	0.28	0.37
Proposed Scheme 2	0.48	0.32	0.39

IV. SECURITY ANALYSIS

In addition to robustness, another important performance aspect of image hashing is security (i.e., the hash values should not be easily forged or estimated without the knowledge of the secret key. In this section, we introduce a framework to evaluate and compare the security of image hashing schemes. We propose to use differential entropy as a metric to study the security of randomized image features and derive analytical expressions of the proposed metric for some representative classes of image hashing algorithms. Further extensions of the proposed framework and other possible approaches to study security are described later in Section V-C.

A. Proposed Security Evaluation Framework

We propose to evaluate the security of image hashing schemes from an adversary viewpoint. The adversary knows the hashing algorithm $g(\cdot)$ and the image I, and tries to estimate the hash values without the knowledge of the secret key. The degree of success that can be attained by the adversary depends on the amount of randomness in the hash values. The higher the amount of randomness in the hash values, the tougher it would be to estimate or duplicate the hash without knowing the key. In the subsequent discussions, we shall focus on the security of the output of the feature extraction stage. Since the quantization and the compression stages are chained with the feature extraction stage, once the entropy of this stage is obtained, the entropy measure for the following stages can be obtained subsequently.

We start the discussion by reviewing the definition of differential entropy [42]. The differential entropy of a continuous random variable X is denoted by $\aleph(X)$ and given by

$$\aleph(X) = \int_{\Omega} f(x) \log_2 \left(\frac{1}{f(x)} \right) dx \tag{8}$$

where $f(x)$ is the probability density function of X, and Ω is the range of support of $f(x)$. In most image hashing schemes, the output of the feature extraction stage consists of two components—a deterministic part and a random part. The deterministic part is contributed by the image content, which we will consider to be known or can be well approximated from the test version of the image that the attacker can acquire. The random part is contributed by the pseudorandom numbers generated using the secret key. In our analysis, we model the output of the feature extraction stage as random variables and find the degree of uncertainty in terms of the differential entropy to arrive at the security metric [15]. In the following sections, we present the security analysis for our proposed scheme, and compare it with the results obtained for a number of representative prior works on image hashing [4], [13], [16].

B. Analytic Expressions of the Security Metric for the Proposed Schemes

In this part, we derive analytic expressions of the security metric for the proposed schemes. In the proposed scheme-1, the randomness in the hash is introduced by the variables $\{\beta_{\rho_k,i}\}$, which are key-dependent pseudorandom numbers, normally distributed with mean m and variance σ^2. The final hash can be considered as a weighted summation of these Gaussian distributed random variables as shown in (5), where the weights of the summation are determined by the image content and known to the users. Since the sum of Gaussian random variables is also Gaussian, the hash value h_k will be Gaussian distributed with mean and variance given by

$$E(h_k) = m \sum_{i=0}^{K-1} \left| I' \left(\rho_k, \frac{(2i+1)\pi}{K} \right) \right| \tag{9}$$

$$\mathrm{Var}(h_k) = \sigma^2 \sum_{i=0}^{K-1} \left| I' \left(\rho_k, \frac{(2i+1)\pi}{K} \right) \right|^2. \tag{10}$$

Therefore, the differential entropy of the feature extraction stage for the proposed scheme-1 can be written as

$$\aleph(h_k) = \frac{1}{2} \log_2 \left((2\pi e)\sigma^2 \sum_{i=0}^{K-1} \left| I' \left(\rho_k, \frac{(2i+1)\pi}{K} \right) \right|^2 \right). \tag{11}$$

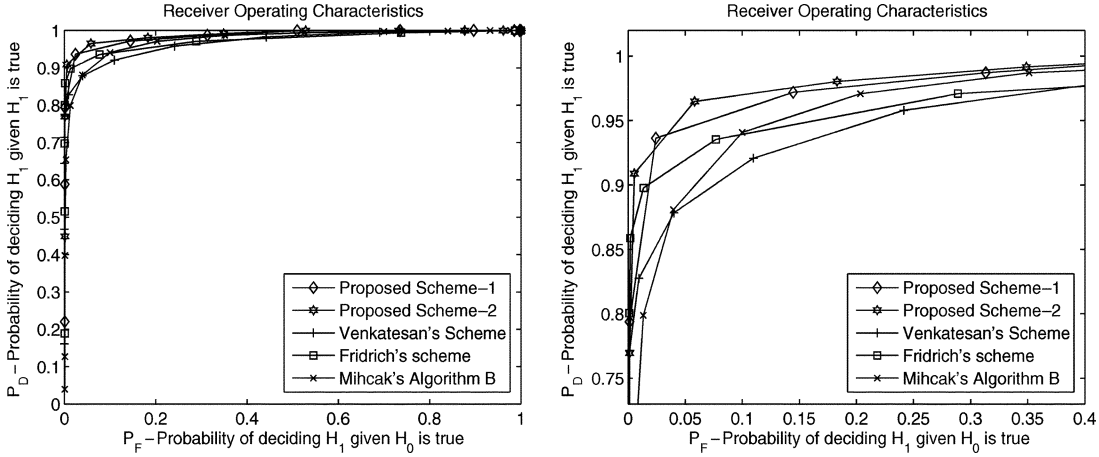


Fig. 10 Receiver operating characteristics of the hypothesis testing problem. The plots display the probability of correct decision (P_D) with respect to the probability of false alarm (P_F). A greater the value of P_D for the same P_F indicates more robustness. The original curve is shown on the left and the magnified version is shown on the right.

We observe that the differential entropy increases as the variance σ^2 becomes large and the scheme becomes more secure as expected. Additionally, we note that the differential entropy rises as the number of sample points K is increased. This is also expected since a higher value of K implies that we involve more random numbers for generating each hash value as shown in (5), and hence the hash would be more difficult to forge.

Next, we derive the security metric for the proposed scheme-2. In this scheme, we use the secret key to generate random sets of radii $\{\Gamma_k\}$, and the weights (β_ρ) for the summation in (6). To facilitate discussions, we define q_ρ as the summation of the polar Fourier transform coefficients at the radius ρ given by

$$q_\rho = \sum_{i=0}^{K-1} \left| I'\left(\rho, \frac{(2i+1)\pi}{K}\right)\right|. \tag{12}$$

The ρ values chosen for generating the hash are from $\Gamma_\rho = \{\rho_1, \rho_2, \ldots, \rho_N\}$. Let λ_{ik} be Bernoulli distributed random variables such that $P(\lambda_{ik}=0) = P(\lambda_{ik}=1) = 0.5$. We rewrite (6) in terms of q_ρ and λ_{ik} to obtain

$$h_k = \sum_{i=1}^{N} \lambda_{ik}\beta_{ik} q_{\rho_i}. \tag{13}$$

We observe that each hash value obtained is a weighted summation of N terms and each of these terms is a product of a Bernoulli and a Gaussian distributed random variable. Therefore, the hash value h_k is not Gaussian. To find the differential entropy of h_k, we first find the probability density function (pdf) of h_k using (13) and then use the pdf to find the entropy. To derive the pdf, we compute the characteristic function of h_k and apply its inverse Fourier transform [43]. It can be shown that the pdf $f_{h_k}(x)$ has a rather complicated form with 2^N terms and is given by

$$\begin{aligned} f_{h_k}(x) = {} & \frac{1}{2^N}\delta(x) + \frac{1}{2^N}\frac{1}{\sqrt{2\pi}}\sum_{i=1}^{N} e^{-\frac{(x-mq_{\rho_i})^2}{2\sigma^2 q_{\rho_i}^2}} \\ & + \frac{1}{2^N}\frac{1}{\sqrt{2\pi}}\sum_{i_1=1}^{N}\sum_{\substack{i_2=1\\ i_2\neq i_1}}^{N} e^{-\frac{\left(x-m\left(q_{\rho_{i_1}}+q_{\rho_{i_2}}\right)\right)^2}{2\sigma^2\left(q_{\rho_{i_1}}^2+q_{\rho_{i_2}}^2\right)}} \\ & + \frac{1}{2^N}\frac{1}{\sqrt{2\pi}}\sum_{\substack{i_1,i_2,i_3=1\\ i_1\neq i_2\neq i_3}}^{N} e^{-\frac{\left(x-m\left(q_{\rho_{i_1}}+q_{\rho_{i_2}}+q_{\rho_{i_3}}\right)\right)^2}{2\sigma^2\left(q_{\rho_{i_1}}^2+q_{\rho_{i_2}}^2+q_{\rho_{i_3}}^2\right)}} \\ & + \cdots + \frac{1}{2^N}\frac{1}{\sqrt{2\pi}} e^{-\frac{\left(x-m\sum_{i=1}^{N} q_{\rho_i}\right)^2}{2\sigma^2\left(\sum_{i=1}^{N} q_{\rho_i}^2\right)}} \end{aligned} \tag{14}$$

where $\delta(\cdot)$ denotes the dirac delta function. We observe that the pdf of h_k is a sum of many Gaussian pdfs and finding the exact expression for the differential entropy by integrating (8) would not be feasible. We instead find the lower and upper bounds of the differential entropy. Using the concavity property of the entropy, we arrive at a lower bound for the differential entropy

$$\begin{aligned} \aleph(h_k) \geq {} & \frac{1}{2^N}\sum_{i=1}^{N}\frac{1}{2}\log_2\left(2\pi e\sigma^2 q_{\rho_i}^2\right) \\ & + \frac{1}{2^N}\sum_{i_1=1}^{N}\sum_{\substack{i_2=1\\ i_1\neq i_2}}^{N}\frac{1}{2}\log_2\left(2\pi e\sigma^2\left(q_{\rho_{i_1}}^2+q_{\rho_{i_2}}^2\right)\right). \\ & + \frac{1}{2^{N+1}}\sum_{\substack{i_1,i_2,i_3=1\\ i_1\neq i_2\neq i_3}}^{N}\log_2\left(2\pi e\sigma^2\left(q_{\rho_{i_1}}^2+q_{\rho_{i_2}}^2+q_{\rho_{i_3}}^2\right)\right) \\ & + \cdots + \frac{1}{2^{N+1}}\log_2\left(2\pi e\sum_{i=1}^{N}\sigma^2 q_{\rho_i}^2\right). \end{aligned} \tag{15}$$

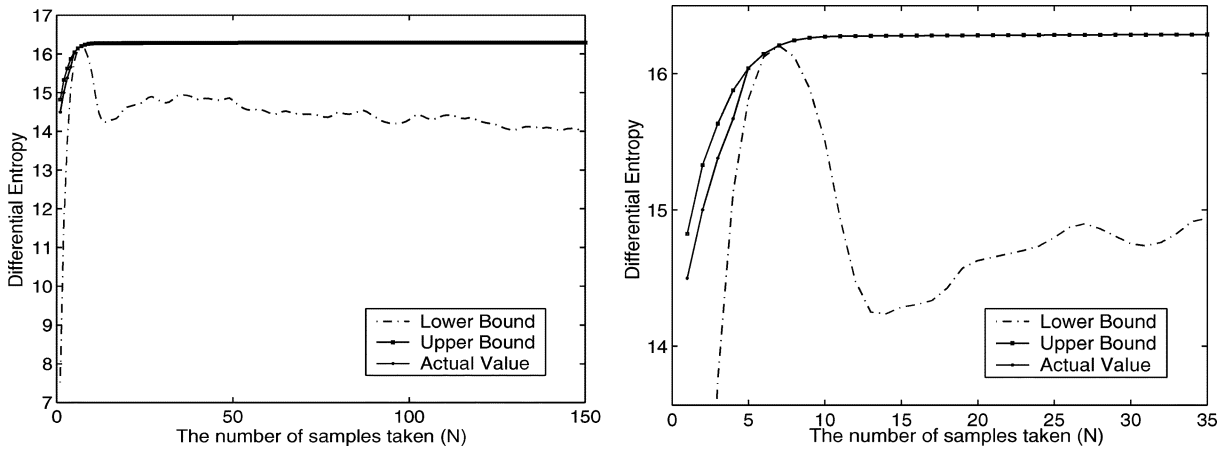


Fig. 11. Entropy of the hash values for the proposed scheme-2 plotted with respect to the number of sampling points N. The plots show the lower bound, the upper bound, and the actual value. The actual plot is shown on the left and the magnified version is shown on the right.

This lower bound can be simplified using the following energy compaction property of the Fourier transform. Without any loss of generality, we assume that the radii are ordered as $\rho_1 < \rho_2 < \rho_3 < \ldots < \rho_N$. Now, since q_{ρ_i} is the summation of the absolute values of the Fourier transform coefficients along the circumference of the circle of radius ρ_i, we have

$$q_{\rho_1} \geq q_{\rho_2} \geq \cdots \geq q_{\rho_N} \tag{16}$$

for most natural images. Using this inequality, (15) can be simplified to give a compact lower bound

$$\aleph(h_k) \geq \frac{2^N - 1}{2^{N+1}} \log_2 \left(2\pi e \sigma^2 q_N^2\right) + \frac{1}{2^N} \sum_{i=1}^{N} \binom{N}{i} \log_2(i). \tag{17}$$

Next, to derive the upper bound, we use the fact that the Gaussian distribution has the maximum differential entropy among all distributions with the same variance. Moreover, the differential entropy of a Gaussian distributed random variable depends only on its variance. Therefore, we obtain an upper bound on $\aleph(h_k)$ by finding variance of the hash values h_k, from the pdf in (14) to arrive at

$$\aleph(h_k) \leq \frac{1}{2} \log_2 \left((2\pi e) \left(\frac{\sigma^2}{2} + \frac{m^2}{4} \right) \sum_{j=1}^{N} q_{\rho_j}^2 \right). \tag{18}$$

In Fig. 11, we show the derived lower and upper bounds along with the actual value for a different number of sampling points (N). The true values were obtained by numerically computing the differential entropy from the pdf of the hash values. We observe that the upper bound plotted using (18) is very tight and is almost equal to the actual value. This is because the true pdf of the hash values is close to Gaussian with the same mean and variance as those used in the upper bound calculation.

C. Extending the Security Evaluation to Other Image Hashing Schemes

In this subsection, we show that the proposed security metric can be extended to study the security of various classes of image hashing schemes and is thus generally applicable. For our study, we consider two representative methods, namely, the scheme by Fridrich [13] and the hashing algorithm by Venkatesan *et al.* [4]. These schemes were chosen as they have very different approaches to introduce randomness in the feature extraction stage. For instance, the Fridrich's scheme [13] secures the hash by projecting the image onto random low-pass images; and the Venkatesan's scheme [4] introduces security by extracting image features from randomly chosen regions of the image.

1) Security of Fridrich's Scheme [13]: This scheme is based on the observation that any significant change made in the transform domain would be reflected as visible changes in the image domain. Key-dependent pseudorandom patterns $\{X^{(r)}\}$ of the same size of the image are initially generated. These patterns are then spatially averaged with a $m \times n$ low-pass filter $\{\alpha_{ij}\}$ to generate zero-mean smoothened random patterns $[Y^{(r)}]_{kl}$. The rth hash value h_r is obtained by projecting the input image on to $Y^{(r)}$, as given by

$$h_r = \sum_{k=1}^{H} \sum_{l=1}^{W} Y_{kl}^{(r)} I_{kl}. \tag{19}$$

To analyze the security of this scheme, we consider the hash values $\{h_r\}$ as random variables and find their distributions. Using this estimated pdf, we compute the differential entropy as

$$\aleph(h_r) = \frac{1}{2} \log_2 \left(2\pi e \frac{1}{12} \sum_{p=1}^{H} \sum_{q=1}^{W} I_{pq} I_{pq}^{(\alpha\alpha)} \right). \tag{20}$$

Here, $I^{(\alpha\alpha)}$ is the image obtained by filtering I twice with the filter $\{\alpha_{ij}\}$. The details of the analysis are presented in Appendix A.

Fig. 12 shows the plot of the differential entropy of the Fridrich's scheme for different orders of the averaging filter. We observe from the plot that the differential entropy decreases as the order of the filter is increased. This result is expected because on increasing the order of the averaging filter, the degree of uncertainty in the smoothened patterns $\{Y^{(r)}\}$ decreases,

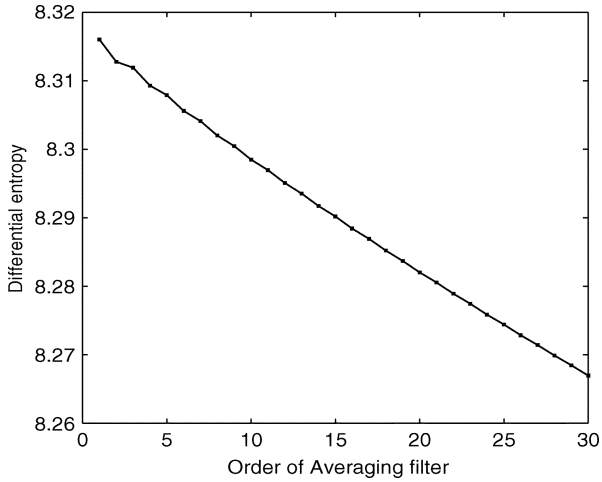


Fig. 12. Differential entropy of the hash for different orders of averaging filters in Fridrich's scheme [13].

as the original random images $\{X^{(r)}\}$ are low-pass filtered to a greater extent. Thus, the amount of randomness of the final hash values is reduced as a consequence.

2) Security of Venkatesan's Scheme [4]: In this scheme, the authors first perform a three-level DWT of the image and then a random tiling of each DWT sub-band of the image is generated. The mean (or variance) of the pixel values in the random rectangle is used to form the feature vectors [4]. These features are then randomly quantized and compressed to generate the hash.

There are two aspects of security in this scheme. To estimate the hash values, the adversary has to first find the locations and sizes of the random partitions and compute the image statistics in these partitions. Then, the adversary needs to arrange the estimated hash values in the correct order to obtain the hash vector. In our analysis, we consider these two aspects separately and obtain the differential entropy in each case.

We first show that the exact size and location of the random partitions is not required to estimate the hash. The attacker can instead make an intelligent guess of the image statistics by replacing the random partitions with uniformly spaced, equal-sized partitions. In [4], the width of the random partition is uniform in $[w_{\min}, w_{\max}]$, where $w_{\min}$ and $w_{\max}$ are the minimum and maximum widths of the random block. Therefore, a good estimate of the partition width would be its expected value $E_w = ((w_{\min} + w_{\max})/2)$. Similarly, the height is uniform in the range $[h_{\min}, h_{\max}]$ and its expected value is $E_h = ((h_{\min} + h_{\max})/2)$. The attacker can calculate the image statistics using uniform size partitions of the size $E_w \times E_h$ to obtain an estimate for the hash values. In Fig. 13(a), we plot the actual hash values, our estimates, and the corresponding difference (i.e., the estimation error). Here, the estimates are obtained by computing the statistics from the closest uniform-spaced partition. We note that the error has a much lower dynamic range than the actual value even though the location and size of the estimated partitions are not exactly the same as those used in hash generation. The amount of randomness in the hash values can be characterized by the degree of uncertainty in our estimation. Therefore, the differential entropy of the first aspect of security $h^{(1)}$ can be numerically obtained by first finding the pdf of the estimation error and then computing the entropy from the pdf. For the Lena image, $h^{(1)}$ can be numerically computed to be around 5.74. We also note that $h^{(1)}$ only characterizes one aspect of randomness in the hash values. Therefore, the actual differential entropy of the hash values $\aleph(h_k)$ would be greater than $h^{(1)}$.

The second aspect of the hash security that we consider here is the randomness associated with the order in which the individual hash values are concatenated together while creating the hash vector. Here, we compare the true hash vectors generated using the randomized block partitions and the ones estimated using uniform partitions and assume that both these hash vectors are obtained using a raster-scan order of the partitioning blocks. It is to be noted that any further permutation of the hash can be factored into the postprocessing stage which we shall not consider here as indicated before. A good uniform partition that emulates the randomized partition can be obtained as follows. We model the 2-D randomized partitioning as a combination of first partitioning the input image along the vertical direction into rows and then further partitioning each row into blocks. Let M denote the number of rows and N_i denote the number of partitions in the ith row. We can show that the expected value of M and N_i is

$$m_M = E(M) = \frac{2H}{h_{\min} + h_{\max}}$$
$$m_N = E(N_i) = E(N) = \frac{2W}{w_{\min} + w_{\max}} \quad \forall\, 1 \le i \le M. \tag{21}$$

The derivation is presented in Appendix B.

Since we use a uniform partition to approximate the randomized partition, there will be synchronization errors in each row of the estimated partition. Let us now denote the number of synchronization errors in the nth row by Y_n. The synchronization error is cumulative and can be written as

$$Y_n = \sum_{i=1}^{n} (N_i - m_N). \tag{22}$$

To obtain an upper bound of the amount of uncertainty in Y_n, we use the fact that of all random variables with the same variance, the Gaussian distribution has the maximum differential entropy. Further, the differential entropy of Gaussian distributed random variables is completely specified by the determinant of its correlation matrix. Therefore, we construct a $M \times M$ correlation matrix R_Y for the set of random variables $\{Y_1, Y_2, \ldots, Y_n\}$

$$R_Y(i,j) = E(Y_i Y_j) = \min(i,j)\sigma_N^2. \tag{23}$$

Here, σ_N^2 denotes the variance of N_i and can be computed from its pdf given in (38) of Appendix B. It can be shown that $|R_Y| = \sigma_N^{2M}$. Therefore, using the Gaussian upper bound, the differential entropy of the stage $(h^{(2)})$ considering the synchronization errors alone is given by

$$h^{(2)} \le \frac{1}{2}\log_2\left(2\pi e \sigma_N^2\right) + \frac{1}{2m_M}\log_2\left(1 + \frac{1}{12\sigma_N^2}\right). \tag{24}$$

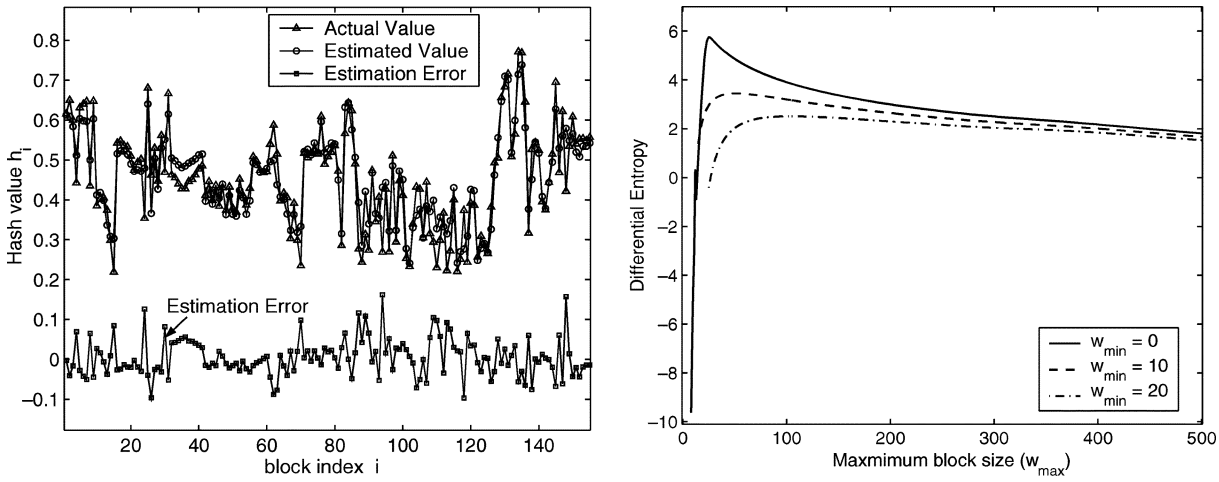


Fig. 13. Security analysis results for Venkatesan's scheme. (a) The plot of the actual and the estimated image statistics vector in the first stage of the hashing scheme along with their differences; for the Lena image with $w_{\min} = 10$, $w_{\max} = 40$, and $W = 512$. (b) The entropy obtained by modeling the synchronization errors plotted for different parameter values of $w_{\max}$ and $w_{\min}$ with $W = H = 512$, $w_{\min} = h_{\min}$, and $w_{\max} = h_{\max}$.

TABLE IV
COMPARISON OF DIFFERENTIAL ENTROPY OF VARIOUS HASHING SCHEMES SHOWN FOR THREE DIFFERENT IMAGES

Hashing Algorithm	Differential Entropy		
	Lena	Baboon	Peppers
Proposed Scheme−1	$8.2 - 15.6$	$13.58 - 16.18$	$8.76 - 15.46$
Proposed Scheme−2	16.28	16.39	16.18
Fridrich's scheme [13]	8.31	8.32	8.14
Venkatesan's scheme [4]	$5.74 - 11.48$	$5.96 - 11.70$	$5.65 - 11.39$
Mihçak's Algorithm B [16]	8	8	8

In Fig. 13(b), we show the plot of the upper bound as given by the right-hand side (RHS) of (24) for different values of $w_{\min}$ and $w_{\max}$. We observe that the upper bound heavily depends on the value of the variance σ_N^2. For very small $w_{\max}$, we have $\sigma_N^2 \to 0$ and, therefore, $h^{(2)} \to -\infty$, suggesting that the hashing algorithm becomes insecure for low σ_N^2. This result is expected because when $w_{\max} \approx w_{\min}$, the window widths and locations become approximately deterministic and the errors caused by synchronization are small.

Overall, when an attacker replaces the random partitions by uniformly spaced partitions to estimate the hash values, the two aspects of security will both contribute to the uncertainty of the hash algorithm. Thus, the final differential entropy can be approximated by $(h^{(1)} + h^{(2)})$.

The above analysis method can be generalized and extended to other hashing schemes alike. For example, analysis can be applied to the hashing scheme by Mihçak [16], which also introduces security by the choice of random regions in the image.

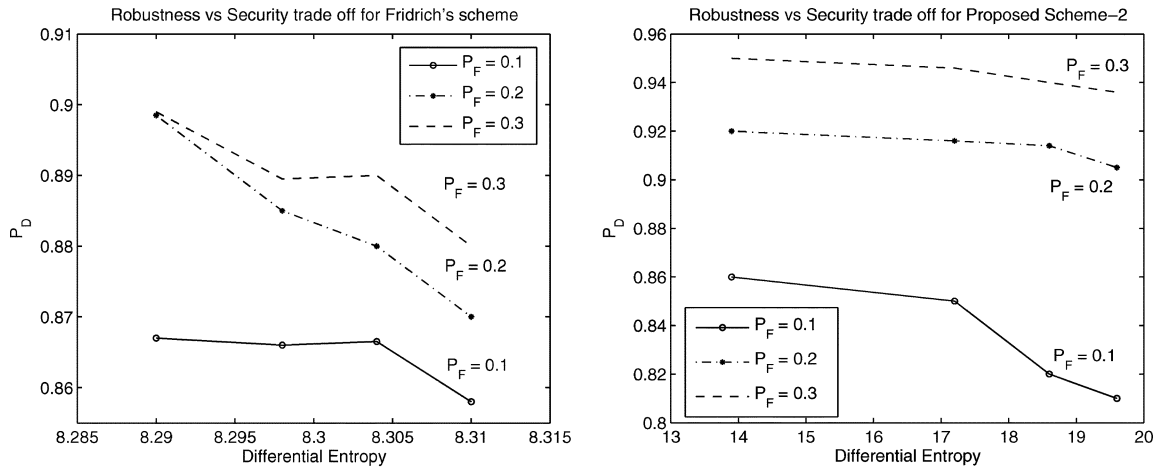
D. Comparison Results

In this subsection, we compare the security of image hashing schemes in terms of the differential entropy as a metric. We compute the differential entropy of the hash values on the Lena image for various schemes and present the results in Table IV.

The differential entropy of the proposed scheme-1 lies in the range 8.2–15.6. This is due to the fact that each hash value in the scheme-1 has a different amount of randomness based on the radius on which the summation in (5) is performed. If the corresponding Fourier transform coefficients have a higher magnitude, then the variance of the hash values would be larger. Thus, some of the hash values can be estimated easily, while it might be difficult to estimate some others. This can be considered as one of the disadvantages of the proposed scheme-1. The disadvantage is overcome in the proposed scheme-2 because the summation is done over randomly chosen subsets and, thus, all of the hash values would have a similar amount of randomness. We note that the differential entropy of the feature extraction stage of the proposed scheme-2 is higher than that of the scheme-1. This is expected because in the proposed scheme-2, the random weights are scaled by larger factors and, thus, the overall variance of the hash values would be higher.

Next, we observe that the differential entropy of the proposed scheme-2 is greater than that of Fridrich's scheme. This can be attributed to the low-pass filtering operations in Fridrich's scheme that reduces the variance of the random variables and, hence, its entropy. The differential entropy of Venkatesan's scheme is lower than those of the proposed schemes. This is because, even without the knowledge of the exact block partitions, the image statistics in Venkatesan's scheme can be estimated to be of reasonable accuracy. On the other hand, in the proposed schemes, the attackers need to guess the random variables in computing features (such as β_{ik}).

Notice that we only consider the security of the feature extraction stage in this work. It should be noted that while random permutation or other techniques alike can be applied to any scheme to bring further randomness, such postprocessing does not change the relative security results obtained in this work. This justifies our focus on evaluating the security of the feature extraction stage.



(a) (b)

Fig. 14. Robustness and security tradeoff for (a) Fridrich's scheme and (b) proposed scheme-2.

V. Discussions

A. Tradeoff Between Robustness and Security

In this section, we jointly consider the two main performance criteria for image hashing, namely, robustness and security. We observe a tradeoff between the two criteria for each hashing scheme and illustrate this phenomenon with some examples.

In Fig. 14(a), we show the tradeoff between robustness and security for the Fridrich's scheme [13]. The scheme was simulated for different orders of averaging filter; and the ROC and the differential entropy was obtained in each case. The ROC was sampled to obtain the probabilities of correct decisions P_D for three different probabilities of false alarm P_F, and plotted with respect to the differential entropy. We observe that as the robustness increases, the scheme becomes less secure and vice-versa. This trend is expected because on increasing the order of the averaging filters, the patterns $Y^{(r)}$ become more smooth, making the scheme more robust to content-preserving manipulations like the ones in Table I. However, the scheme becomes less secure because the smooth patterns $Y^{(r)}$ would be less random.

Similar behavior can also be observed for the proposed scheme-2. The performance of the scheme was studied for different parameter values; and the ROC and the differential entropy were obtained in each case. As shown in Fig. 14(b), we observe that for a fixed P_F, as we increase the variance of the random weights β_{ik}, the differential entropy increases and the robustness decreases. However, it is to be noted that proposed scheme exhibits a better tradeoff compared to Fridrich's scheme. This is evident by comparing the X-axis of Fig. 14(a) and (b). We observe that proposed scheme-2 is more secure than the Fridrich's scheme for the same amount of robustness. This demonstrates the advantages of the proposed scheme.

The robustness results in Fig. 9 and the differential entropy values in Table IV show that the proposed scheme-2 provides better tradeoff between robustness and security against guessing than the proposed scheme-1. This is attributed to the fact that the circular summation along the θ-axis in proposed scheme-2 can generate more robust features. In the mean time, we also remark that the circular summation is a double-edged sword and may reduce the resilience against collision and forgery attacks. It is possible for malicious attackers to perform meaningful changes by altering individual values of the Fourier transform coefficients while preserving the overall sum. In contrast, the proposed scheme-1 is more resilient to such collision attacks, as the weights of the summation are random and depend on a secret key unknown to adversaries. A possible improvement is to employ a weighted circular summation with gradually changing weights, where the varying trend of the weights is specified by a secret key. This hybrid scheme can combine the advantages of the two proposed schemes, improving the collision resistance compared to scheme-2 and also the robustness compared to scheme-1.

B. Extending the Security Analysis to Quantization Algorithms

We have shown that the differential entropy can be used as a metric to study the security of the feature extraction stage in image hashing. In this section, we extend the security analysis beyond the feature extraction stage and show that entropy can be used as a metric to study the degree of security of the quantization stage that follows feature extraction.

As an example, we consider the randomized quantization algorithm proposed in [5], which is an adaptive quantization algorithm that takes into account the distribution of the input data. The quantization bins $[\Delta_{i-1}, \Delta_i]$ are designed so that $\int_{\Delta_{i-1}}^{\Delta_i} p_X(x)dx = 1/Q$, where Q is the number of quantization levels and $p_X(\cdot)$ is the pdf of the input data X. The central points $\{C_i\}$ are defined so as to make $\int_{\Delta_{i-1}}^{C_i} p_X(x)dx = \int_{C_i}^{\Delta_i} p_X(x)dx = 1/(2Q)$, and the randomization interval $[A_i, B_i]$ are chosen such that $\int_{A_i}^{\Delta_i} p_X(x)dx = \int_{\Delta_i}^{B_i} p_X(x)dx = r/Q$, where $r \leq (1/2)$ is a randomization parameter. The overall quantization method can be expressed as

$$q(x)=\begin{cases} i-1 & \text{w.p. } 1, & \text{if } C_i \leq x \leq A_i \\ i-1 & \text{w.p. } \left(\dfrac{Q}{2r}\displaystyle\int_x^{B_i} p_X(t)dt\right), & \text{if } A_i \leq x \leq B_i \\ i & \text{w.p. } \left(\dfrac{Q}{2r}\displaystyle\int_{A_i}^{x} p_X(t)dt\right), & \text{if } A_i \leq x \leq B_i \\ i & \text{w.p. } 1, & \text{if } B_i \leq x \leq C_{i+1}. \end{cases} \tag{25}$$

We again use the conditional entropy $\aleph(h_k|I)$ as a security metric. Based on the detailed derivation in Appendix C, we can show that

$$H\left(q(X)|X\right) = r\log_2(e) \tag{26}$$

which quantifies the amount of randomness introduced by the randomized quantization. We note that the conditional entropy is directly proportional on the randomization parameter r, and is independent of the source distribution. Other quantization algorithms can be analyzed similarly using conditional entropy as a metric.

C. Further Discussions on Hash Security

In this paper, we have considered the conditional entropy of the hash values as a metric to study security. Our analysis is based on the premise that the adversary knows the image and the hashing algorithm being used and does not know the key used in generating the hash. Therefore, in our analysis, the adversary does not have access to the actual hash values and tries to estimate them based on his or her knowledge. Alternatively, we can evaluate the security of a hashing scheme by measuring the conditional entropy of the hashing key when the image, the hashing algorithm, and output hash values are known. This conditional entropy can be written as $\aleph(K|(I,h))$, where K denotes the key, I the image, and h is the corresponding hash value. In reality, if more information is available to the adversary, he or she may be able to come up with more sophisticated attacks to break the hashing algorithm. In such a case, the conditional entropy of the key will reduce with the increase in the number of observed image/hash pairs. Thus, $\aleph(K|(I_1,h_1),(I_2,h_2),\ldots(I_n,h_n))$ is a monotonically decreasing function with n. When n is large enough, it would be possible to uniquely identify the key K with very high probability. This is analogous to Shannon's discussion on the secrecy system and the definition of unicity distance [44]. Along these lines, we may define another notion of hashing security by requiring that the conditional entropy $\aleph(K|(I_1,h_1),(I_2,h_2),\ldots(I_n,h_n))$ is not negligible as long as the number of observed image/hash pairs n is upper bounded by a polynomial in key length. We note that for image hashing and other types of multimedia hashing, an adversary may not need to exactly recover the key in order to estimate a hash. The estimation type of attack introduced in [30] is clearly an example.

VI. Conclusion

Robustness and security are two important requirements for image hashing algorithms in applications involving authentication, watermarking, and image databases. In this paper, we have developed new image hashing schemes that have improved robustness and security features. We show that the proposed schemes is resilient to moderate filtering, and compression operations, and common geometric operations up to 10° of rotation and 20% of cropping. The proposed hashing scheme also has good discriminative capabilities and can identify malicious manipulations, such as a cut-and-paste type of editing, that do not preserve the content of the image. In addition to the study on robustness, we have introduced a general framework for analyzing the security in image hashing. We derive analytical expressions using differential entropy as a metric to study the security of the feature extraction stage for both the proposed schemes and several existing representative schemes. Our studies have shown that the proposed image hashing algorithm is highly secure in terms of this metric. The analysis can also be extended to incorporate other stages of the hashing operation, such as randomized quantization.

Overall, we developed a new image hashing algorithm. It is more robust compared to existing image hashing schemes and, at the same time, it is also secure against estimation and forgery attacks. Thus, it can provide a robust and secure representation of images for numerous applications.

Appendix A
Deriving the Security Metric for Fridrich's Scheme [13]

In Fridrich's scheme, key-dependent pseudorandom patterns $X^{(r)}(r = 1,2,\ldots N)$ of the same size of the input image are first generated. These pseudorandom patterns have uniform distributed pixel values. These patterns are then spatially averaged with a $m \times n$ low-pass filter $\{\alpha_{ij}\}$ to obtain zero-mean random images $[Y^{(r)}]_{kl}$

$$Y_{kl}^{(r)} = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} \alpha_{ij} X_{i+k,j+l}^{(r)}. \tag{27}$$

The input image I is projected on the N smooth patterns $\{Y^{(r)}\}$ to obtain the intermediate hash values h_r as given by

$$h_r = \sum_{k=1}^{H} \sum_{l=1}^{W} Y_{kl}^{(r)} I_{kl}. \tag{28}$$

These intermediate hash values are then quantized to generate the final hash. In our analysis, we model the intermediate hash values h_r as random variables and find its differential entropy to generate the security metric. The hash values h_r in (28) can be rewritten as

$$h_r = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} \alpha_{ij} V_{ij}^{(r)} \tag{29}$$

where the random variables $V_{ij}^{(r)}$ are defined as

$$V_{ij}^{(r)} = \sum_{k=1}^{H} \sum_{l=1}^{W} X_{i+k,j+l}^{(r)} I_{kl}. \tag{30}$$

We observe that $V_{ij}^{(r)}$ is a weighted sum of $W \times H$ uniformly distributed random variables $\{X_{ij}^{(r)}\}$ with the weights determined by the image pixel values (I_{kl}). According to the Central Limit

theorem, we approximate $V_{ij}^{(r)}$ to be Gaussian distributed, with mean $m_{ij}^{(r)}$ and variance $\sigma_{ij}^{2(r)}$ that can be shown to be

$$m_{ij}^{(r)} = E\left(V_{ij}^{(r)}\right) = \frac{1}{2}\left(\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}\right)$$
$$\sigma_{ij}^{2(r)} = \frac{1}{12}\left(\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}^2\right). \quad (31)$$

We also note that all $\{V_{ij}^{(r)}\}$ are identically distributed, but are not independent since the same random variables $\{X_{ij}^{(r)}\}$ are used to generate various $V_{ij}^{(r)}$. The dependence among the variables $\{V_{ij}^{(r)}\}$ can be expressed in terms of their correlation given by

$$E\left(V_{ij}^{(r)}V_{ab}^{(r)}\right) = \frac{1}{12}\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}I_{i+k-a,j+l-b} + \left(\frac{1}{2}\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}\right)^2. \quad (32)$$

Now, from (29), we see that h_r is a weighted sum of $m \times n$ Gaussian distributed random variables. So h_r is also Gaussian and its differential entropy is completely specified by its variance. The variance of h_r can be computed as

$$\begin{aligned}\sigma_{h_r}^2 &= E\left(h_r^2\right) - m_{h_r}^2 \\ &= E\left(\sum_{i=-\lfloor\frac{m}{2}\rfloor}^{\lfloor\frac{m}{2}\rfloor}\sum_{j=-\lfloor\frac{n}{2}\rfloor}^{\lfloor\frac{n}{2}\rfloor}\alpha_{ij}V_{ij}^{(r)}\right)^2 - \left(\frac{1}{2}\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}\right)^2 \\ &= \frac{1}{12}\sum_{p=1}^{H}\sum_{q=1}^{W} I_{pq}I_{pq}^{(\alpha\alpha)} \end{aligned} \quad (33)$$

where

$$I_{pq}^{(\alpha\alpha)} = \sum_{i,k=-\lfloor\frac{m}{2}\rfloor}^{\lfloor\frac{m}{2}\rfloor}\sum_{j,l=-\lfloor\frac{n}{2}\rfloor}^{\lfloor\frac{n}{2}\rfloor}\alpha_{ij}\alpha_{kl}I_{i+p-k,j+q-l}. \quad (34)$$

Note that $I^{(\alpha\alpha)}$ is the image obtained by filtering I the image twice with the filter $\{\alpha_{ij}\}$. Using the result in (33), we obtain the differential entropy of h_r as

$$\aleph(h_r) = \frac{1}{2}\log_2\left(2\pi e\frac{1}{12}\sum_{p=1}^{H}\sum_{q=1}^{W} I_{pq}I_{pq}^{(\alpha\alpha)}\right). \quad (35)$$

APPENDIX B
MODEL FOR BLOCK PARTITIONING IN VENKATESAN'S SCHEME [4]

As indicated in Section IV-C2, we approximate the 2-D block partitioning as a combination of two one-dimensional (1-D) problems, namely, partitioning along the horizontal direction and then along the vertical direction. To model the partition along the width of the image, we divide the space $(0, W)$ into several regions by successively generating random numbers $\{U_k\}$ as shown in Fig. 15, uniformly distributed in $[w_{\min}, w_{\max}]$, and $w_{\min}$ and $w_{\max}$ are the minimum and the maximum widths of the random blocks. The location of the nth partition is then given by a set of random variables T_n, where $T_n = \sum_{k=1}^{n} U_k$. Since T_n is the sum of n uniformly distributed random variables, we approximate T_n with a Gaussian distribution. Its mean m_{T_n} and variance $\sigma_{T_n}^2$ can be shown to be

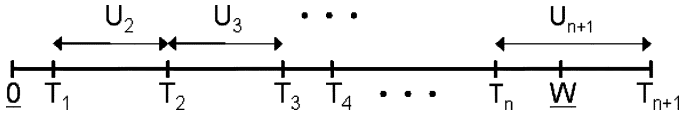


Fig. 15. Simplified model of the block partitioning algorithm in Venkatesan's scheme [4].

$$m_{T_n} = \frac{n}{2}(w_{\min} + w_{\max}) \qquad \sigma_{T_n}^2 = \frac{n}{12}(w_{\max} - w_{\min})^2. \quad (36)$$

Let N_i denote the number of partitions in the ith row. Using the distribution of T_n and noting that N_i is also the index for the last partition in the row, we can write the pdf of N_i as

$$\begin{aligned} Pr(N_i = n) &= Pr(T_n < W < T_{n+1}) \\ &= Pr(\max(\mathrm{W} - \mathrm{T_n}, \mathrm{w_{min}}) < \mathrm{U_{n+1}} < \mathrm{w_{max}}) \\ &= \int_{W-w_{\max}}^{W-w_{\min}} P(W - t < U_{n+1} < w_{\max}) f_{T_n}(t)dt \\ &\quad + \int_{W-w_{\min}}^{W} P(w_{\min} < U_{n+1} < w_{\max}) f_{T_n}(t)dt \end{aligned} \quad (37)$$

where $f_{T_n}(\cdot)$ is the pdf of T_n. Using the Gaussian assumption on T_n, the above expression can be simplified as

$$\begin{aligned} P(N_i = n) &= \frac{\sigma_n}{\sqrt{2\pi}(w_{\max} - w_{\min})} \\ &\quad \times \left(\exp\left(-\frac{(W - w_{\max} - m_{T_n})^2}{2\sigma_{T_n}^2}\right)\right. \\ &\quad \left. - \exp\left(-\frac{(W - w_{\min} - m_{T_n})^2}{2\sigma_{T_n}^2}\right)\right) \\ &\quad + \left(\frac{w_{\max} + m_{T_n} - W}{w_{\max} - w_{\min}}\right) \\ &\quad \times (F_{T_n}(W - w_{\min}) - F_{T_n}(W - w_{\max})) \\ &\quad + (F_{T_n}(W) - F_{T_n}(W - w_{\min})) \end{aligned} \quad (38)$$

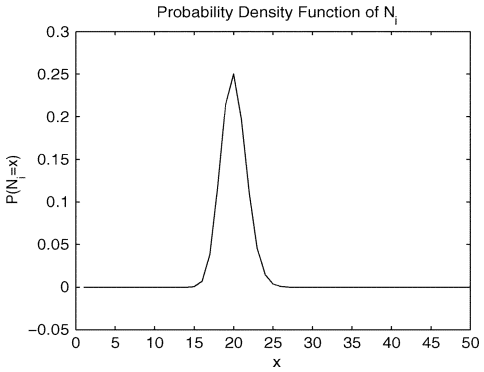


Fig. 16. Plot of the pdf of N_i—the number of blocks in ith row, where the parameters are $w_{\min} = 10$, $w_{\max} = 40$, and $W = 512$. Note that the random variable N_i has a very small variance and hence the mean would be a good estimate.

where $F_{T_n}(x)$ is the cumulative distribution function (cdf) of T_n and is given by

$$F_{T_n}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{x-m_{T_n}}{\sigma_{T_n}}} \exp\left(-\frac{z^2}{2}\right) dz. \tag{39}$$

The plot of the pdf of N_i is shown in Fig. 16. From this pdf, we can derive the expected value of N_i as $E(N_i) = (2W/(w_{\min} + w_{\max}))$.

Appendix C
Deriving the Security Metric for Randomized Quantization [5]

In this appendix, we provide the detailed derivations of the conditional entropy for the randomized quantization algorithm [5]. The conditional entropy $H(q(X)|X)$ can be written as

$$\begin{aligned} H\left(q(X)|X\right) &= \int_{x\in\Re} H\left(q(X)|X=x\right) p_X(x)dx \\ &= \sum_{i=1}^{Q} \int_{C_i}^{C_{i+1}} H\left(q(X)|X=x\right) p_X(x)dx \\ &= \sum_{i=1}^{Q} \int_{A_i}^{B_i} H\left(q(X)|X=x\right) p_X(x)dx \end{aligned} \tag{40}$$

where $p_X(\cdot)$ denotes the pdf of the input data X. The last step follows from (25) since the quantizer $q(X)$ is random only in the interval $A_i \le x \le B_i$. Now, we note that in this interval, $q(X)$ takes a value i with probability $p_i = (P_X(x) - P_X(A_i))(Q/2r)$, and a value $(i-1)$ with probability $(1-p_i)$. Therefore, (40) can be calculated and simplified as

$$\begin{aligned} H\left(q(X)|X\right) = &-\sum_{i=1}^{Q} \int_{A_i}^{B_i} \left(p_i \log_2(p_i) + (1-p_i)\right. \\ &\left. \times \log_2(1-p_i)\right) p_X(x)dx \\ = &\, r\log_2(e). \end{aligned} \tag{41}$$

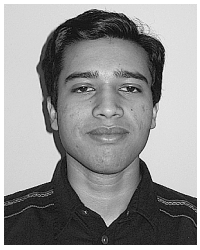
Acknowledgment

The authors would like to thank Prof. N. Memon and Dr. M. Kivanç Mihçak for their valuable comments and suggestions.

References

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Mateo, CA: Morgan Kaufmann, 2001.
[2] M. Wu and B. Liu, *Multimedia Data Hiding*. New York: Springer-Verlag, 2002.
[3] A. Menezes, V. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC, 1998.
[4] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, BC, Canada, Sep. 2000, vol. 3, pp. 664–666.
[5] M. K. Mihçak and R. Venkatesan, "A tool for robust audio information hiding: A perceptual audio hashing algorithm," in *Proc. 4th Int. Information Hiding Workshop*, Pittsburgh, PA, Apr. 2001.
[6] C. Kailasanathan, R. S. Naini, and P. Ogunbona, "Image authentication surviving acceptable modifications," in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal Image Processing*, Baltimore, MD, Jun. 2001.
[7] E. Martinen and G. W. Wornell, "Multimedia content authentication: fundamental limits," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, Sep. 2002, vol. 2, pp. 17–20.
[8] S. Lin, M. T. Ozsu, V. Oria, and R. Ng, "An extendible hash for multi-precision similarity querying of image databases," in *Proc. 27th Very Large Data Bases (VLDB) Conf.*, Roma, Italy, 2001.
[9] I. J. Cox and J.-P. M. G. Linnartz, "Public watermarks and resistance to tampering," in *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997, vol. 3, pp. 3–6.
[10] J. Cannons and P. Moulin, "Design and statistical analysis of a hash-aided image watermarking system," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1393–1408, Oct. 2004.
[11] M. Holliman, N. Memon, and M. M. Yeung, "On the need for image dependent keys for watermarking," in *Proc. Content Security and Data Hiding in Digital Media*, Newark, NJ, May 1999.
[12] K. Su, D. Kundur, and D. Hatzinakos, "Statistical invisibility for collusion-resistant digital video watermarking," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 43–51, Feb. 2004.
[13] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," in *Proc. IEEE Int. Conf. Information Technology: Coding Computing*, Mar. 2000, pp. 178–183.
[14] A. Swaminathan, Y. Mao, and M. Wu, "Image hashing resilient to geometric and filtering operations," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Siena, Italy, Sep. 2004.
[15] ——, "Security of feature extraction in image hashing," in *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing* , Philadelphia, PA, Mar. 2005.
[16] M. K. Mihçak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *Proc. ACM Workshop Security and Privacy in Digital Rights Management*, Philadelphia, PA, Nov. 2001.
[17] F. Lefbvre, B. Macq, and J.-D. Legat, "RASH: RAdon Soft Hash algorithm," in *Proc. EUSIPCO*, Toulouse, France, 2002.
[18] L. Xie, G. R. Arce, and R. F. Graveman, "Approximate image message authentication codes," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 242–252, Jun. 2001.
[19] F. Lefbvre, J. Czyz, and B. Macq, "A robust soft hash algorithm or digital image signature," in *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sep. 2003, vol. 2, pp. 495–498.
[20] S. S. Kozat, R. Venkatesan, and M. K. Mihçak, "Robust perceptual image hashing via matrix invariants," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, vol. 5, pp. 3443–3446.
[21] M. Malkin and R. Venkatesan, "The randlet transform: Applications to universal perceptual hashing and image authentication," in *Proc. Allerton Conf.*, Monticello, IL, 2004.
[22] J. Fridrich, "Visual hash for oblivious watermarking," in *Proc. IS&T/SPIE 12th Annu. Symp., Electronic Imaging, Security and Watermarking of Multimedia Content II*, San Jose, CA, Jan. 2000, vol. 3971.
[23] V. Monga and B. L. Evans, "Robust perceptual image hashing using feature points," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004.

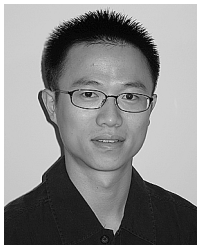
[24] M. Schneider and S.-F. Chang, "A robust content based digital signature for image authentication," in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Sep. 1996, vol. 3, pp. 227–230.
[25] A. M. Ferman, A. M. Tekalp, and R. Mehrotra, "Robust color histogram descriptors for video segment retrieval and identification," *IEEE Trans. Image Process.*, vol. 11, no. 5, pp. 497–508, May 2002.
[26] F. Jing, M. Li, H.-J. Zhang, and B. Zhang, "An efficient and effective region-based image retrieval framework," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 699–709, May 2004.
[27] M. P. Queluz, "Toward robust, content based techniques for image authentication," in *Proc. IEEE Signal Processing Society—Second Workshop on Multimedia Signal Processing*, Dec. 1998, pp. 297–302.
[28] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits Syst. Video Technology*, vol. 11, no. 2, pp. 153–168, Feb. 2001.
[29] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998, vol. 1, pp. 435–439.
[30] R. Radhakrishnan, Z. Xiong, and N. Memom, "On the security of the visual hash function," in *Proc. SPIE—Security and Watermarking of Multimedia Contents V*, San Jose, CA, 2003, vol. 5020.
[31] A. Meixner and A. Uhl, "Analysis of a wavelet based robust hash algorithm," in *Proc. SPIE-IS&T—Security, Steganography Watermarking of Multimedia Contents VI*, San Jose, CA, 2004, vol. 5306.
[32] M. Johnson and K. Ramachandran, "Dither-based secure image zhashing using distributed coding," in *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sep. 2003, vol. 2, pp. 751–754.
[33] R. E. Blahut, *Theory and Practice of Error-Correcting Codes*. Reading, MA: Addison-Wesley, 1994.
[34] J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," in *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997, vol. 1, pp. 536–539.
[35] C.-Y. Lin, M. Wu, J. A. Bloom, M. L. Miller, I. J. Cox, and Y.-M. Lui, "Rotation, scale, and translation resilient public watermarking for images," *IEEE Trans. Image Process.*, vol. 10, no. 5, pp. 767–782, May 2001.
[36] A. K. Jain, *Fundamentals of Digital Image Processing*, V ed. Upper Saddle River, NJ: Prentice-Hall, 2000.
[37] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1991.
[38] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.
[39] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," *Second International Workshop on Information Hiding (IHW)* Apr. 1998, Proc. Lecture Notes Comput. Sci. 1525. New York, Springer-Verlag, 3-540-65 386-4, pp. 219–239.
[40] C. W. Wu, "On the design of content-based multimedia authentication systems," *IEEE Trans. Multimedia*, vol. 4, no. 3, pp. 385–393, Sep. 2002.
[41] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, Feb. 1994.
[42] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, ser. Telecommun. New York: Wiley, 1991.
[43] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 2002.
[44] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Tech. J.*, vol. 28-4, pp. 656–715, 1949.



Ashwin Swaminathan (S'05) received the B.Tech degree in electrical engineering from the Indian Institute of Technology (IIT), Madras, India, in 2003, and is currently pursuing the Ph.D. degree in signal processing and communications at the Department of Electrical and Computer Engineering, University of Maryland, College Park.

His research interests include multimedia forensics, information security, and authentication.

Mr. Swaminathan's paper on multimedia security was selected as a winner of the Student Paper Contest at the 2005 IEEE International Conference on Acoustic, Speech, and Signal Processing.



Yinian Mao (S'04) received the B.E. degree in electrical engineering from Tsinghua University, Beijing, China, in 2001, and is currently pursuing the Ph.D. degree in signal processing and communications at the at the Department of Electrical and Computer Engineering, University of Maryland, College Park.

He was a Research Intern with Microsoft Research, Redmond, WA, in 2004. His research interests include information security and multimedia signal processing.

Mr. Mao is a co-author of a paper on multimedia security that was selected as a winner of the Student Paper Contest at the 2005 IEEE International Conference on Acoustic, Speech, and Signal Processing.

Min Wu (S'95–M'01) received the B.E. degree (Hons.) in electrical engineering and B.A. degree in economics (Hons.) from Tsinghua University, Beijing, China, in 1996 and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 2001.

Currently, she is an Assistant Professor in the Department of Electrical and Computer Engineering and the Institute of Advanced Computer Studies, University of Maryland at College Park. Previously, she was with NEC Research Institute and Panasonic Laboratories, Princeton. She coauthored *Multimedia Data Hiding* (Springer-Verlag, 2003) and *Multimedia Fingerprinting Forensics for Traitor Tracing* (EURASIP/Hindawi, 2005) and holds five U.S. patents. Her research interests include information security and forensics, multimedia signal processing, and multimedia communications. She served as a Guest Editor of a 2004 special issue in *EURASIP Journal on Applied Signal Processing*.

Dr. Wu received the National Scicnce Foundation CAREER Award in 2002, a University of Maryland George Corcoran Education Award in 2003, a Massachusetts Institute of Technology Technology Review's TR100 Young Innovator Award in 2004, and an Office of Naval Research (ONR) Young Investigator Award in 2005. She was a corecipient of the 2004 EURASIP Best Paper Award and the 2005 IEEE Signal Processing Society Best Paper Award. She is an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS. She was Publicity Chair of the 2003 IEEE International Conference on Multimedia and Expo.