International Journal of Innovative
Computing, Information and Control
Volume 8, Number 9, September 2012

 ISSN 1349-4198
pp. 6161–6172

PERCEPTUAL IMAGE HASHING BASED ON THE ERROR DIFFUSION HALFTONE MECHANISM

Chuan Qin[1,2], Chin-Chen Chang[2,3,*] and Pei-Ling Tsou[2]

[1]School of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
No. 516, Jungong Road, Shanghai 200093, P. R. China
qin@usst.edu.cn

[2]Department of Information Engineering and Computer Science
Feng Chia University
No. 100, Wenhwa Road, Taichung 40724, Taiwan
*Corresponding author: alan3c@gmail.com; pltsou02@gmail.com

[3]Department of Biomedical Imaging and Radiological Science
China Medical University
No. 91, Hsueh-Shih Road, Taichung 40402, Taiwan

Received May 2011; revised September 2011

Abstract. *This paper proposes an image hashing scheme using the halftone mechanism, which can be applied in the fields such as image authentication and retrieval. Image resizing and total variation based filtering are firstly used to pre-process the input image for regularization. Halftone transform based on the error diffusion mechanism is then performed to obtain one binary halftone image that can effectively represent the principle content of the original image. Two kinds of random pattern sequences with uniform and normal distributions are respectively utilized to extract the features of the halftone image. The generated binary feature string is finally scrambled to produce the resulting hash. The security of the scheme completely depends on the secret keys. Experiments are conducted to show that the present scheme has satisfactory robustness performance against perceptually content-preserving manipulations, and has very low anti-collision probability for the hashes of distinct images simultaneously.*
Keywords: Image hashing, Halftone, Error diffusion, Robustness, Anti-collision

1. **Introduction.** Image hash, also called image digest or image authentication code, is a fixed-length binary or real number sequence, which is generated by the image hashing algorithm. This hash sequence is a compact representation of the principle features of the image. In traditional cryptography, there are also a lot of hash functions, such as SHA-1 and MD5, which can inconvertibly map a variable length input message to a short, fixed length string. Since the cryptographic hash function is sensitive, any slight change of the input message will influence the result of the hash value significantly. However, for the scenario of image hashing, the traditional cryptographic hash functions will not be suitable. This is due to the fact that the images usually must undergo various processes and many of the processes are content-preserving, e.g., JPEG compression, filtering, and resizing, even though the digital representations of the images will be changed. The three desirable properties of image hashing are listed below:

(1) *Perceptual robustness*: The hash values of perceptually similar images should be identical or with small distances. That is to say, after the content-preserving manipulations, there should be a very high probability for the image to output the hash with small

distance to the original, even if the pixel values of the image might be altered. The image hash function should be robust to content-preserving manipulations.

(2) *Uniqueness*: The perceptually different images should produce significantly different hash values. That is to say, there should be a very low probability for two visually distinct images to output the hashes with small distances, which is also called anti-collision capability or discriminating capability.

(3) *Security*: The image hash generation procedure should be controlled by the secret key. There should be an extremely low probability that the attacker can estimate the correct hash value without the secret key. Also, without knowing a legal user's key, the attacker cannot forge an image and its corresponding hash to cheat the legal user.

Recently, the image hashing technique has been widely used in various applications, including image authentication, tamper detection, and content-based image retrieval (CBIR). Image watermarking is a kind of technique that can also be used for image authentication, but the watermark bits should be embedded into the cover image for future extraction and authentication, which might cause the degradation of image quality [1-5]. Different with image watermarking, the image hashing technique produces image hash bits, which are just sent to the receiver side together with the image. Therefore, the image hashing operation does not degrade the image quality, since no data are embedded into the image. In general, there are two main stages in most reported image hashing schemes, i.e., feature extraction and hash generation. First, salient features are extracted to represent the main content of the image compactly, and then the extracted features are quantized to form the final hash value. The flowchart of image hashing used for image authentication is illustrated in Figure 1.

In this paper, we propose a new robust and secure image hashing scheme based on the halftone mechanism, which can effectively resist common content-preserving manipulations with very low anti-collision probability simultaneously. Since general output devices, such as a printer, can only produce black and white points, halftone methods have been

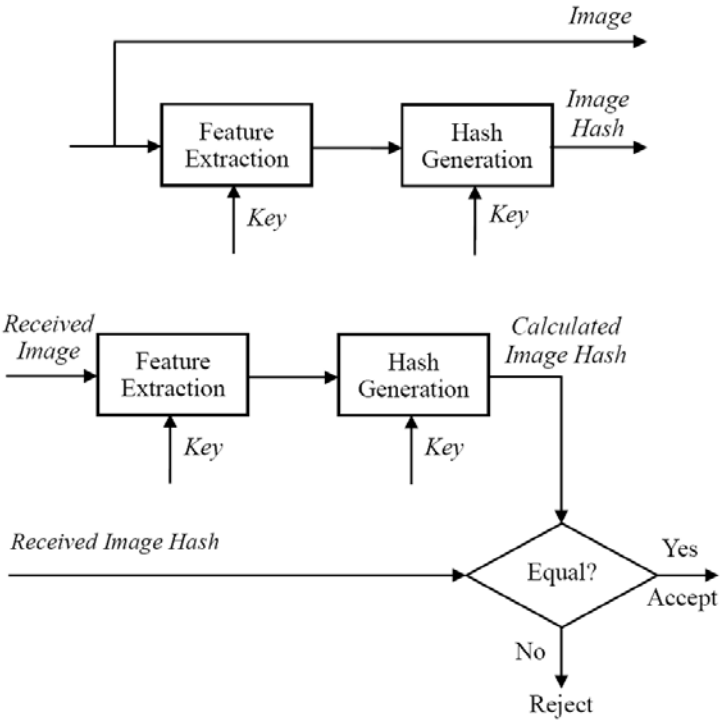


FIGURE 1. Image hashing for authentication

developed to solve the problem when these devices are used for outputting gray level images. The halftone methods produce binary images, which can reflect the visually salient features of corresponding gray level images. Among many reported halftone methods, such as ordered dithering, error diffusion, least squares, and dot diffusion [6,7], the error diffusion based method offers better visual quality and reasonable computational complexity. In the proposed scheme, first, we pre-process the input image for regularization and utilize the error diffusion based halftone method to extract the principle features. Then, two kinds of random pattern sequences generated by secret keys are used to quantize the extracted features and form the final hash securely.

The rest of the paper is organized as follows. Section 2 introduces the current related works. Section 3 describes the proposed robust and secure image hashing scheme based on the halftone mechanism. Experimental results and analysis are presented in Section 4, and Section 5 concludes this paper.

2. **Related Works.** Many image hashing schemes have been proposed in recent years. One category of current image hashing schemes utilizes classic image transforms, such as discrete cosine transform (DCT), discrete wavelet transform (DWT) and Fourier-Mellin transform, to extract the features. Fridrich et al. [8] proposed a DCT-based image hashing approach, in which they indicated that the image cannot maintain the invariance of its low frequency coefficients when significant changes occurred in the image. The DCT coefficient matrices are projected on N low-pass filtered random masks. By judging the signs of the inner product, the projecting results can be quantized into the final hash with a length of N bits. Because the projecting results of the DCT coefficients on smooth masks reflect the low-frequency features of the image, the generated hash is somewhat robust. In another work [9], the radial projections of the image pixels first build a RAdial Variance (RAV) vector, and then the 40 low-frequency DCT coefficients of the RAV vector are converted into the robust image hash called RASH. The RASH vector is more robust than the histogram-based feature vector, but its collision probability is not low enough.

Swaminathan et al. presented a robust image hashing method using Fourier-Mellin transform [10]. The sum of the magnitudes of Fourier coefficients with randomized weights is formed into the image feature. They formulated the robustness of the hashing method as one hypothesis testing problem and evaluated the performance under various image processing manipulations. The final hash generated from the extracted feature is resilient to some content-preserving modifications, such as moderate geometric and filtering distortions. But since the feature is only dependent on the magnitudes of the frequency coefficients, the attacker can forge a tampered image with the same hash by altering the phases of the frequency coefficients.

Another category of image hashing schemes takes advantage of the matrix decomposition or factorization to extract the image features. Kozat et al. suggested that the attacks on the image can be viewed as a sequence of linear operators, and they proposed an image hashing scheme based on matrix invariants using singular value decomposition (SVD) [11]. The scheme first constructs one secondary image from the input image by randomly extracting features that reflect the semi-global geometric characteristics, and then the final features of the spectral matrix invariants are extracted further from the secondary image by SVD to produce the hash value.

Monga et al. introduced the non-negative matrix factorization (NMF) into the image hashing problem [12]. Due to the non-negativity constraints, NMF is distinguished from traditional matrix dimensionality reduction techniques, such as SVD. In their work, first, NMF is applied to decompose the sub-blocks that are randomly chosen from the input image. Then, the secondary image is constructed using factorization factors, and NMF is

used again upon the secondary image to obtain the matrix approximation with dimensionality reduction via projection onto random vectors. The matrix entries are concatenated to form an NMF-NMF vector, which can generate the final image hash. In another study [13], Tang et al. indicated that the invariant relationship in the coefficient matrix of NMF can be used for producing robust hash. They quantized the coefficient matrix with the feature-bearing property to produce a binary matrix, and the resulting hash was obtained after scrambling. It has been shown that the NMF-based methods have more robust performance, but the complexity of the computations is relatively high.

There are some other representative image hashing methods. An image hashing paradigm using visually significant feature points was proposed in [14]. The feature points extracted by an iterative feature detector are invariant under insignificant distortions. Khelifi et al. presented a secure image hashing technique based on virtual watermark detection [15]. The idea relies on the fact that the watermark detector responds similarly to images that are perceptually close, using a non-embedded watermark.

3. **Proposed Image Hashing Scheme.** In the proposed scheme, the halftone method is utilized to extract robust principle features of the image, and then the features are quantized into a binary sequence to form the final hash. During the feature extraction and quantization procedure, the secret keys are introduced to enhance the security of the hashing scheme. The image hashing scheme is mainly composed of the following four stages: 1) image pre-processing, 2) halftone transform using error diffusion, 3) feature extraction using random patterns, and 4) generation of a binary sequence with secret keys as the final hash. The framework of the proposed scheme is shown in Figure 2.

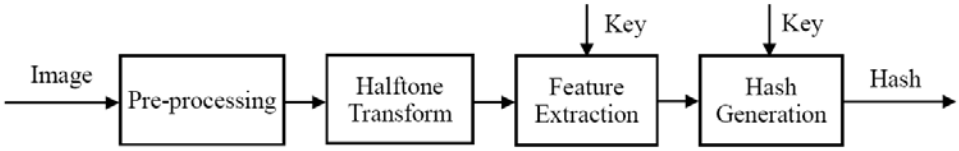


FIGURE 2. Framework of the proposed scheme

3.1. **Pre-processing.** Because the input images for hashing calculation could be color or gray-level images of different sizes, a series of pre-processing steps for normalization should be applied. First, we change all the input images into a standard size of $M \times M$ by bi-linear interpolation to ensure the generated hash has a fixed length. Then, for a color image, we only consider the luminance component Y of the YCrCb color space, since it contains the main structural and geometric information of the image. Finally, we utilize non-linear filtering based on total variation (TV) [16] to produce the pre-processed image. By denoting the Y component of the resized image as u_0, the non-linear filtering can be done by minimizing the following energy function:

$$\mathrm{E}_\lambda(u) = \int |\nabla u| dxdy + \frac{\lambda}{2} \int |u - u_0|^2 dxdy, \tag{1}$$

where λ is the Lagrange multiplier for this constrained variational problem. According to the Euler-Lagrange equations, we know that the minimization problem associated with Equation (1) can be transformed into a partial differential equation (PDE), which is easier to solve by the finite difference method:

$$\frac{\partial u}{\partial t} = \mathrm{div}\left[\frac{\nabla u}{|\nabla u|}\right] + \lambda(u_0 - u), \tag{2}$$

where u is the filtering result, t is the time index and $\mathrm{div}(\cdot)$ is the divergence operator. This kind of non-linear filtering can preserve the edges and remove noises simultaneously, which can effectively alleviate the influences of noise contamination and the loss of image details on the final hash.

After the pre-processing procedure mentioned above, the input image will be regularized and suitable to carry out feature extraction using halftone transform.

3.2. **Halftone transform.** In the error diffusion based halftone method, there exists one filter that is used to diffuse the feed-forward error to the neighboring pixels in the input gray image. The filter presented in [7] is shown in Figure 3. The symbol "-" in Figure 3 corresponds to the pixel location that has already been processed in the input image, and the symbol "$*$" denotes the pixel location that is currently being scanned. The pixel error information in the input gray image diffuses according to the proportion of the filter. Then, the binary output image can be obtained, which is visually equivalent to the original gray image, because the density of the black and white points can reflect the gray level of the original image. In the following, the halftone processing procedure using the error diffusion mechanism will be introduced.

–	$*$	7/16
3/16	5/16	1/16

FIGURE 3. Error diffusion filter

Denote the image after pre-processing as u and its pixel value $u(x, y) \in \{0, 1, \ldots, 255\}$. Scan the image u from left to right and from top to bottom. During the scanning, the current pixel will be converted to 0 or 1, and its neighboring pixels also will be updated by the feed-forward error diffusion. After all pixels in u are processed, the halftone image u_1 will be produced, and its pixel value $u_1(x, y)$ is either 0 or 1. The detailed scanning procedure is described as follows:

Step 1: The current halftone pixel is computed by comparing with the threshold $T = 128$.

$$u_1(x, y) = \begin{cases} 0, & \text{if } u(x, y) < T \\ 1, & \text{if } u(x, y) \geq T. \end{cases} \tag{3}$$

Step 2: Compute the feed-forward error e for diffusion.

$$e = \begin{cases} u(x, y) - 0, & \text{if } u_1(x, y) = 0 \\ u(x, y) - 255, & \text{if } u_1(x, y) = 1. \end{cases} \tag{4}$$

Step 3: Diffuse the feed-forward error to the neighboring pixels according to following equations, and continue the above steps to finish the processing of all pixels.

$$u(x, y + 1) = u(x, y + 1) + \frac{7}{16} \times e, \tag{5}$$

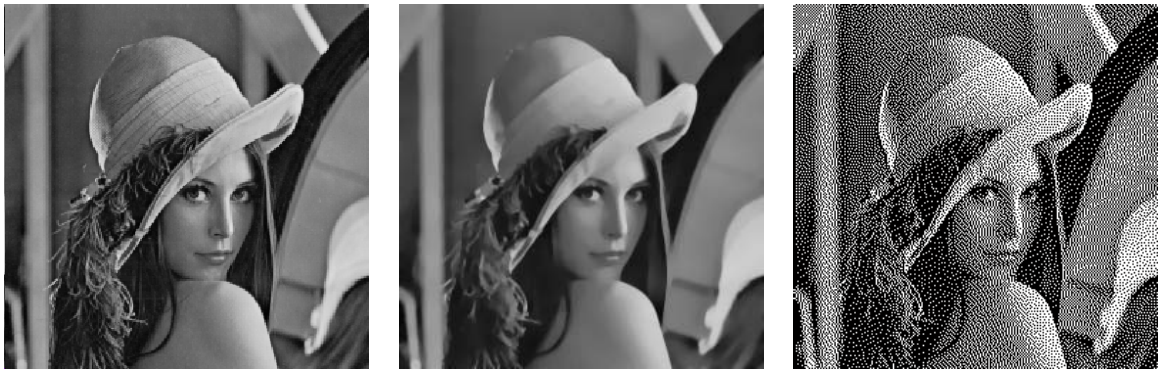
$$u(x + 1, y - 1) = u(x + 1, y - 1) + \frac{3}{16} \times e, \tag{6}$$

$$u(x + 1, y) = u(x + 1, y) + \frac{5}{16} \times e, \tag{7}$$

$$u(x + 1, y + 1) = u(x + 1, y + 1) + \frac{1}{16} \times e. \tag{8}$$

Figure 4 gives an example of halftone processing based on error diffusion. Figures 4(a) and 4(b) are the original Lena image and its pre-processed version after TV filtering,

respectively, and Figure 4(c) is the result of halftone transform. Note that Figure 4(c) only has two kinds of pixel values, i.e., 0 and 1, which has a similar visual effect with the original one. From the above, we find that the halftone version can effectively represent the main visual feature of the original image. So the halftone technique can be utilized to construct an image hashing scheme.



(a) (b) (c)

FIGURE 4. Result of halftone transform for Lena

3.3. **Feature extraction and hash generation.** In order to extract the features of the image version after halftone transform, we first divide the $M \times M$ halftone image u_1 into non-overlapping blocks with the size of $b \times b$. One binary hash bit will be generated from one image block, so it can easily be found that the final hash will have $L = [M/b]^2$ bits.

Denote all the image blocks as $B_1, B_2, \ldots, B_L$. We use secret key K_1 to randomly create L different patterns sized $b \times b$: $R_1, R_2, \ldots, R_L$, and each pattern has only two uniform distribution values, i.e., 0 and 255. The following equations are applied to extract the distribution feature of each halftone block B_i $(i = 1, 2, \ldots, L)$:

$$\begin{array}{lll} S_i(x,y) = 2, & \text{if } B_i(x,y) = 255 & \text{and } R_i(x,y) = 0 \\ S_i(x,y) = -2, & \text{if } B_i(x,y) = 0 & \text{and } R_i(x,y) = 255 \\ S_i(x,y) = 1, & \text{if } B_i(x,y) = 255 & \text{and } R_i(x,y) = 255 \\ S_i(x,y) = -1, & \text{if } B_i(x,y) = 0 & \text{and } R_i(x,y) = 0 \end{array}, \tag{9}$$

where S_i is the extracted intermediate feature of B_i, and $x, y \in \{1, 2, \ldots, b\}$. To further obtain the block feature, we generate another L patterns sized $b \times b$ with normal distribution by secret key K_2: $P_1, P_2, \ldots, P_L$, and all the elements in every pattern P_i $(i = 1, 2, \ldots, L)$ are the real numbers belonging to the interval $[0, 1]$. Using these normal distribution patterns, the intermediate feature S_i $(i = 1, 2, \ldots, L)$ can be quantized into a binary sequence:

$$h'(i) = \begin{cases} 1, & \text{if } \sum_{x=1}^{b} \sum_{y=1}^{b} S_i(x,y) \cdot P_i(x,y) \geq 0 \\ 0, & \text{if } \sum_{x=1}^{b} \sum_{y=1}^{b} S_i(x,y) \cdot P_i(x,y) < 0 \end{cases} \qquad i = 1, 2, \ldots, L. \tag{10}$$

Finally, to enhance the security of the hashing scheme, the binary sequence h' is randomly scrambled by the secret key K_3 to form the final hash $\mathbf{h}$.

4. **Experimental Results and Analysis.** Experiments were conducted to analyze the performances of the proposed hashing scheme with respect to perceptual robustness, anti-collision property, and key-dependent security. In the pre-processing procedure, all input images are resized to 512×512, and the image block size in the experiments is 32×32. Therefore, the hash length L of the proposed scheme is $[512/32]^2 = 256$ bits. We executed our codes of Matlab 6.5 on a computer with Intel Core2 2.4 GHz processor and 4 GB memory under the operating system of Windows 7. The whole hash calculating process of our scheme for each image took less than 0.0515 seconds averagely.

The normalized Hamming distance is utilized to measure the similarity between two hashes:

$$d\left(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}\right) = \frac{1}{L} \sum_{i=1}^{L} \left| h_i^{(1)} - h_i^{(2)} \right|, \tag{11}$$

where $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$ denote the two hash vectors, and $h_i^{(1)}$ and $h_i^{(2)}$ are the hash bits in the corresponding hash vectors respectively. If the normalized Hamming distance between two hashes is smaller than a predetermined threshold T, we can judge that the corresponding images of the two hashes are similar perceptually. Otherwise, the two images are visually distinct. In the evaluation of the performances of robustness and anti-collision, the same secret keys are used for all testing images.

4.1. **Perceptual robustness.** Like the robustness testing of image watermarking schemes [17], common content-preserving manipulations are also applied in the experiments to examine the robustness of our hashing scheme. The attacking manipulations include JPEG compression, Gaussian filtering, additive noise contamination, image scaling, and image rotation. The detailed parameters of these attacks are listed in Table 1.

TABLE 1. Image content-preserving manipulations

Names	Descriptions	Parameters
JPEG compression	Quality factor	$40, 42, \ldots, 100$
Gaussian filtering	Standard deviation	$0.1, 0.15, \ldots, 2$
Additive noise	Level	$0.002, 0.003, \ldots, 0.01$
Image scaling	Ratio	$0.5, 0.7, \ldots, 2.3$
Image rotation	Rotation angle	$1, 1.2, \ldots, 5$

In the experiments we compare our method mainly with two typical reported image hashing methods, i.e., Fridrich's method [8] and the RASH method [9]. However, because the hashes generated by the RASH method are decimal real numbers, we use Equation (12) to convert them to the corresponding binary numbers $c^{(b)}$ in order to compare the normalized Hamming distance conveniently.

$$c_i^{(b)} = \begin{cases} 1, & \text{if } c_i \geq c_{i+1} \\ 0, & \text{if } c_i < c_{i+1} \end{cases} \qquad i = 1, 2, \ldots, L_c, \tag{12}$$

where c_i denotes the ith decimal hash number in the original RASH method, L_c is its hash length, and $c_{Lc+1} = c_1$. The quantization procedure of Equation (12) can be understood to show that the robustness of the RASH method can guarantee the invariance of the sign relationships of the neighboring hash values under content-preserving manipulations [13]. In this way, the hashes of Fridrich's method, the RASH method, and the proposed method are all binary bit sequences with lengths of 256 bits, 128 bits, and 256 bits, respectively. So we can use the normalized Hamming distance to compare the performance impartially.

Experiments were carried out on a group of standard images sized 256×256 and 512×512, including the 20 images shown in Figure 5. In Figures 6(a)-6(e), the ordinate is the average value of the normalized Hamming distances between the hash pairs of the 20 original images in Figure 5 and their attacked versions. We can find that the average Hamming distances of present method against JPEG compression, Gaussian filtering, additive noises, scaling, and rotation are all smaller than they are for Fridrich's method or the RASH method. So it can be concluded that our image hashing method based on the halftone mechanism has better robustness against usual content-preserving operations than Fridrich's method and RASH method.

FIGURE 5. Standard images for robustness testing

4.2. **Anti-collision performance.** Anti-collision means that two images that are visually distinct have a very low probability of generating similar hashes. That is to say, the normalized Hamming distance between two visually distinct images should be larger than a pre-determined threshold T. If the normalized Hamming distance between two distinct images is smaller than T, the collision emerges.

In order to evaluate the anti-collision performance of the present method, we use the uncompressed color image database (UCID) [18] for testing in our experiment. There are 1,338 various color images in the database with sizes of 512×384. We first generate 1,338 hashes for the images, and then calculate the 894,453 normalized Hamming distances between the hash pairs of different images. The histogram of the normalized Hamming distances is shown in Figure 7. According to the parameter estimation method, we find

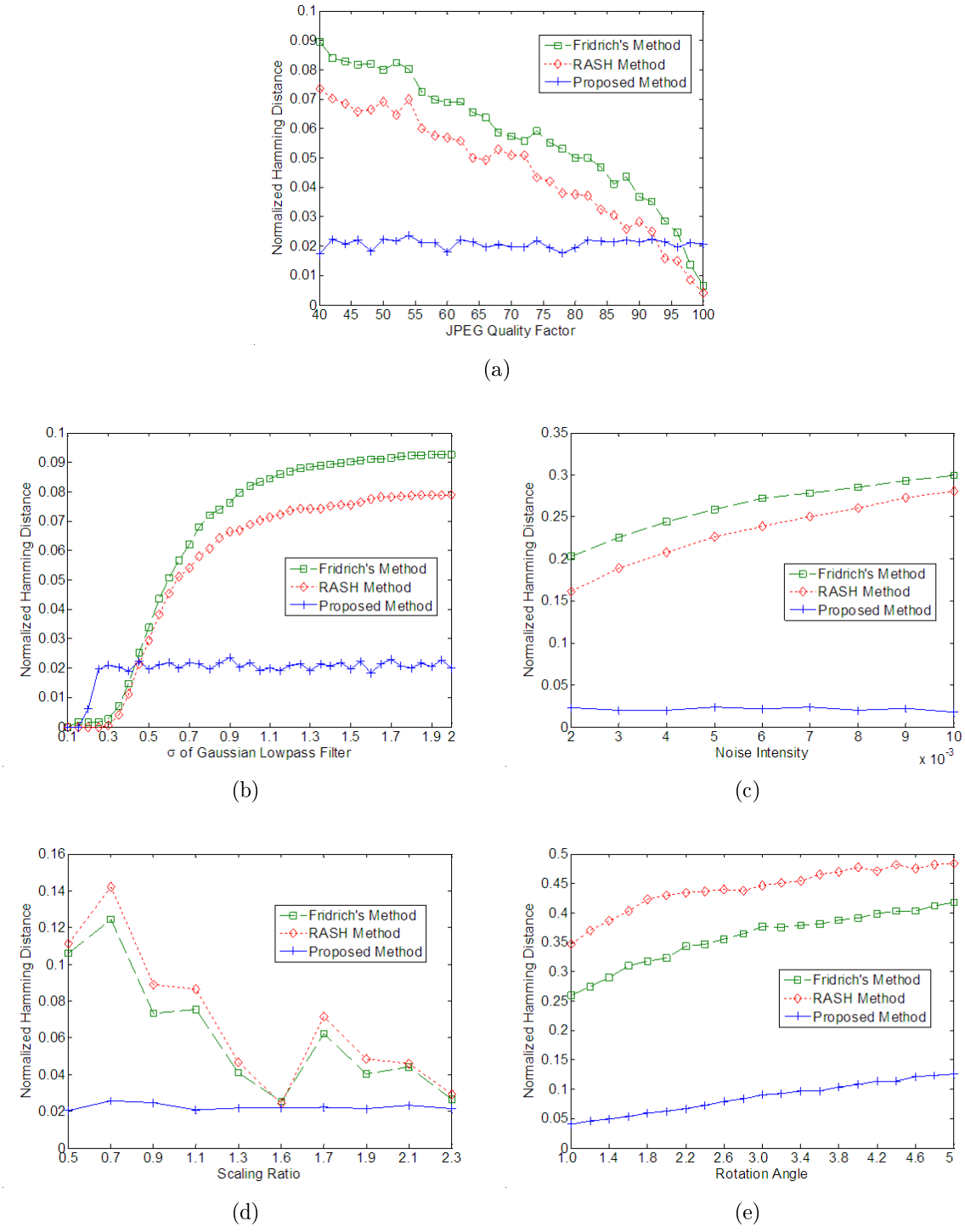


FIGURE 6. Results of robustness performance

that the distribution of the normalized Hamming distance approximates a normal distribution with its mean and standard deviation being $\mu = 0.49$ and $\sigma = 0.064$, respectively. So, once the threshold T is determined, the collision probability P_c for two distinct images is just the probability that the normalized Hamming distance is smaller than T:

$$\mathrm{P}_c(T) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{T}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]dx = \frac{1}{2}\mathrm{erfc}\left(-\frac{T-\mu}{\sqrt{2}\sigma}\right), \tag{13}$$

where erfc$(\cdot)$ is the complementary error function. Table 2 lists the collision probability of the proposed scheme for different threshold values of T. The second column in Table 2 shows that the normalized threshold T corresponds to $\lfloor L \times T \rfloor$ bits in the hashes with a length of L.

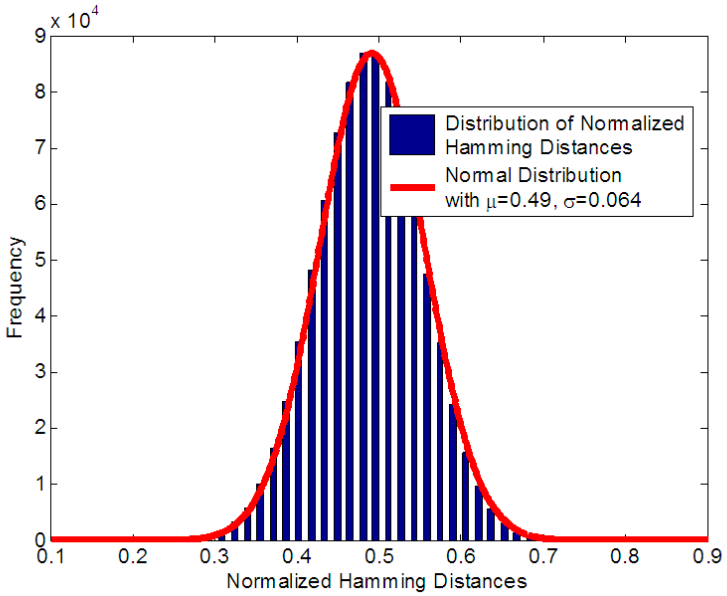


FIGURE 7. Distribution of normalized Hamming distances between hashing pairs of different images

Table 2 shows that, if the threshold is set smaller, the collision probability is also smaller. But on the other hand, the smaller threshold T will influence the robustness of performance, which possibly leads to higher probability of false judgment for content-preserving manipulations. Figure 6 shows that the normalized Hamming distances of our scheme against five common kinds of content-preserving manipulations are all below 0.1, except for the case of rotation with a larger angle. Therefore, we can set the normalized threshold $T = 0.1$ to achieve better robustness of performance and better anti-collision property simultaneously.

TABLE 2. Collision probabilities for different thresholds T

Threshold T	Number in $L = 256$ (bits)	Collision Probability
0.16	40	1.01×10^{-7}
0.14	35	1.78×10^{-8}
0.12	30	2.87×10^{-9}
0.10	25	4.18×10^{-10}
0.08	20	5.54×10^{-11}
0.06	15	6.67×10^{-12}
0.04	10	7.29×10^{-13}

4.3. **Key-dependent security.** As described in Section 2, the present scheme has three secret keys, i.e., K_1, K_2, and K_3, and they can be formed into one key string K. Different key strings will produce totally different hashes; see Figure 8. The abscissa of Figure 8 is the index of 1,000 wrong secret keys, and the ordinate is the average value of the normalized Hamming distances between the hash pairs of the images in Figure 5 obtained

by the correct and wrong keys. We can see that almost all of the normalized Hamming distances are in the vicinity of 0.5. So it is extremely difficult for the attacker to generate or estimate the same hash without knowing the correct key, K. Furthermore, the attacker cannot forge an image and its corresponding hash to trap the other user, because he or she does not have the user's secret key. The security of our image hashing scheme completely depends on the secret keys, which satisfies the security requirements in the sense of cryptography.

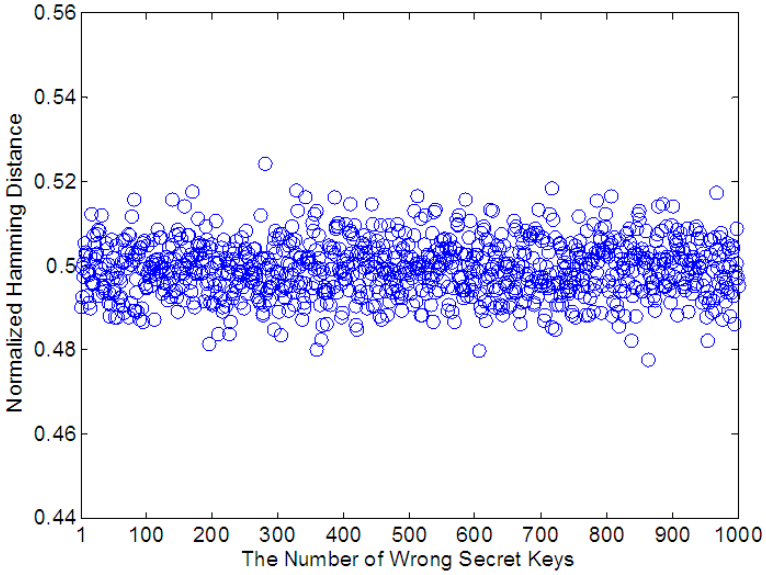


FIGURE 8. Normalized Hamming distances between hashing pairs using the correct and wrong secret keys

5. **Conclusions.** In this paper, we have proposed an efficient robust image hashing scheme that can be applicable to, for example, image authentication and image retrieval. There are four main stages of the hashing scheme, i.e., pre-processing, halftone transform, feature extraction, and hash generation. Some operations, such as resizing and TV filtering, are applied to achieve regularization in the pre-processing stage. Then, the halftone transform based on error diffusion is utilized to represent the principle content of the regularized image. Two kinds of random pattern sequences with the uniform and normal distributions are used to extract features of the halftone image. Finally, the intermediate binary feature string is scrambled to produce the resulting hash securely. The computational complexity of the proposed scheme is low. The obtained hash is robust against common content-preserving manipulations, such as JPEG compression, Gaussian filtering, additive noise contamination, image scaling, and image rotation. The collision probability between the hashes of visually distinct images is very low. Compared with other reported methods, our scheme has better robustness of performance and better anti-collision property, which can be achieved by setting the appropriate threshold. Also, the security of the present scheme is dependent on the secret keys, i.e., the attacker cannot forge an image and its corresponding hash to cheat the legal user.

Now, we only consider the luminance component of the image for hash calculation, the chrominance information isn't incorporated in the current scheme. Future work will take the color-related features of the image into consideration and create a hashing scheme that can discriminate between images with distinct colors. In addition, robustness against more

content-preserving manipulations, such as larger angle image rotation and the print-scan operation, deserves in-depth investigations.

Acknowledgment. This work is supported by the Shanghai Specialized Research Foundation for Excellent Young Teacher in University (slg09005), and the OECE Innovation Foundation of USST (GDCX-Y-103). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

REFERENCES

[1] I. J. Cox, J. Kilian, F. T. Leighton and T. Shamoon, Secure spread spectrum watermarking for multimedia, *IEEE Trans. on Image Processing*, vol.6, no.12, pp.1673-1687, 1997.

[2] C.-C. Chang, C.-C. Lin and Y.-S. Hu, An SVD oriented watermark embedding scheme with high qualities for the restored images, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.609-620, 2007.

[3] W.-L. Tai and C.-C. Chang, Data hiding based on VQ compressed images using Hamming codes and declustering, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.2043-2052, 2009.

[4] C.-C. Lin, Y.-H. Chen and C.-C. Chang, LSB-based high-capacity data embedding scheme for digital images, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4283-4289, 2009.

[5] L. Li, D. Zhang and L. Ruan, A new similarity degree of digital watermarks, *ICIC Express Letters*, vol.5, no.3, pp.727-731, 2011.

[6] D. L. Lau and G. R. Arce, *Modern Digital Halftoning*, Marcel Dekker, New York, 2001.

[7] X. Wang, Z. Xu and P. Niu, A feature-based digital watermarking scheme for halftone image, *International Journal of Electronics and Communications*, vol.64, no.10, pp.924-933, 2010.

[8] J. Fridrich and M. Goljan, Robust hash function for digital watermarking, *Proc. of International Conf. on Information Technology: Coding and Computing*, Las Vegas, NV, USA, pp.173-178, 2000.

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

[10] A. Swaminathan, Y. Mao and M. Wu, Robust and secure image hashing, *IEEE Trans. on Information Forensics and Security*, vol.1, no.2, pp.215-230, 2006.

[11] S. S. Kozat, R. Venkatesan and M. K. Mihcak, Robust perceptual image hashing via matrix invariants, *Proc. of International Conf. on Image Processing*, Singapore, pp.3443-3446, 2004.

[12] V. Monga and M. K. Mhcak, Robust and secure image hashing via non-negative matrix factorizations, *IEEE Trans. on Information Forensics and Security*, vol.2, no.3, pp.376-390, 2007.

[13] Z. Tang, S. Wang, X. Zhang, W. Wei and S. Su, Robust image hashing for tamper detection using non-negative matrix factorization, *Journal of Ubiquitous Convergence Technology*, vol.2, no.1, pp.18-26, 2008.

[14] V. Monga and B. L. Evans, Perceptual image hashing via feature points: Performance evaluation and tradeoffs, *IEEE Trans. on Image Processing*, vol.15, no.11, pp.3452-3465, 2006.

[15] F. Khelifi and J. Jiang, Perceptual image hashing based on virtual watermark detection, *IEEE Trans. on Image Processing*, vol.19, no.4, pp.981-994, 2010.

[16] T. Chan and J. Shen, Nontexture inpainting by curvature-driven diffusions, *Journal of Visual Communication and Image Representation*, vol.12, no.4, pp.436-449, 2001.

[17] F. A. P. Petitcolas, Watermarking schemes evaluation, *IEEE Signal Processing Magazine*, vol.17, no.5, pp.58-64, 2000.

[18] G. Schaefer and M. Stich, UCID – An uncompressed color image database, *Proc. of SPIE, Storage and Retrieval Methods and Applications for Multimedia*, San Jose, USA, pp.472-480, 2004.