Fundamenta Informaticae 106 (2011) 75–91
DOI 10.3233/FI-2011-377
IOS Press

Structural Feature-Based Image Hashing and Similarity Metric for Tampering Detection

Zhenjun Tang*, Shuozhong Wang†, Xinpeng Zhang, Weimin Wei
School of Communication and Information Engineering
Shanghai University, Shanghai 200072, China
tangzj230@163.com; shuowang@shu.edu.cn, xzhang@shu.edu.cn, wwm@shu.edu.cn

Abstract. Structural image features are exploited to construct perceptual image hashes in this work. The image is first preprocessed and divided into overlapped blocks. Correlation between each image block and a reference pattern is calculated. The intermediate hash is obtained from the correlation coefficients. These coefficients are finally mapped to the interval $[0, 100]$, and scrambled to generate the hash sequence. A key component of the hashing method is a specially defined similarity metric to measure the "distance" between hashes. This similarity metric is sensitive to visually unacceptable alterations in small regions of the image, enabling the detection of small area tampering in the image. The hash is robust against content-preserving processing such as JPEG compression, moderate noise contamination, watermark embedding, re-scaling, brightness and contrast adjustment, and low-pass filtering. It has very low collision probability. Experiments are conducted to show performance of the proposed method.

Keywords: image hashing, tampering detection, content-based image retrieval, structural feature, similarity metric, watermarking

*This work was supported by the Natural Science Foundation of China (60773079, 60872116, and 60832010), the High-Tech Research and Development Program of China (2007AA01Z477), the Innovative Research Foundation of Shanghai University for Ph.D. Programs (shucx080148), and the Scientific Research Foundation of Guangxi Normal University for Doctors. The authors would like to thank the anonymous referees for their valuable comments and suggestions.
Also works: Department of Computer Science, Guangxi Normal University, Guilin 541004, China
†Address for correspondence: School of Communication and Information Engineering, Shanghai University, 149 Yanchang Road, Shanghai 200072, China.

1. Introduction

Since a digital image is easy to process, copy and transfer, security of its contents is an important issue that attracts much research interest. Fragile watermarking [1] is a useful technique for content integrity verification, which involves data embedding and inevitably introduces distortion to the host image. An alternative approach is to extract an authentication code from the image that represents the contents, called an image hash. If the image is maliciously altered, the hash value should change. Unlike cryptographic hashes such as SHA-1 and MD5 that are extremely sensitive to slight changes, even one bit, of the message, an image hash must be robust against content-preserving modifications. Perceptual robustness is important because images often undergo various digital processing such as contrast enhancement, de-noising, and JPEG compression. These are generally considered as normal manipulations, therefore should not cause significant changes in the hash value. Simultaneously satisfying the requirements of perceptual robustness and tampering detection capability is a challenging task in developing a high performance hashing algorithm.

Image hashing can be used in image authentication, tampering detection, digital watermarking, image indexing, and content-based image retrieval (CBIR). Since no additional data are inserted into the image, the image itself is intact. In general, an image hash should have the following properties:

- Perceptual robustness: The hash function should map visually identical images to the same hash even if their digital representations are not exactly the same. Visually similar images without significant differences may have hashes with a small distance.
- Uniqueness, or anti-collision capability: Probability of two different images having an identical hash value, or very close hash values, should tend to zero.
- Sensitivity to visual distinction: Perceptually important changes to an image should lead to a completely different hash even though the change is limited to a small region. This feature is essential for the image hash to be useful in image authentication and digital forensics.
- Key-dependence: The hash must be generated under the control of a secret key or several keys. It should be virtually impossible to estimate the hash without a correct key. The last two requirements may collectively be considered as security of image hashing.

In this paper, we propose a new method that focuses on the image hash's capability of detecting local area tampering. In Section 2, the related works in image hashing will be given. We describe the hashing method in Section 3, and present experimental results in Section 4. In Section 5, performance comparisons between the proposed method and three other methods are made. Section 6 concludes the paper.

2. Related works

Earlier methods of image hashing include the wavelet coefficient statistics-based scheme, DCT-based approach, relation based technique, Radon transform method, etc. In [2], Venkatesan et al. introduce a me-

thod using randomized signal processing strategies for a non-reversible compression of images into random binary strings. It is robust against compression and small geometric distortion. The method is analogous to message authentication codes (MAC) to minimize collision probability. However, this method is not robust enough against contrast adjustment, gamma correction, and cannot detect malicious object insertion [3]. Fridrich and Goljan [4] observe that the magnitude of a low-frequency DCT coefficient cannot be changed easily without causing visible changes to the image. They divide an image into non-overlapping blocks, extract N bits from each block, and concatenate them to form the hash. For each block, N DC-free random smooth patterns are generated from a secret key, and the DCT basis vectors are replaced with these smooth patterns (with DCT coefficients equivalent to projections onto the patterns). Absolute values of projection smaller than a threshold are denoted by 0 and the others by 1. Their hash is robust against a set of common processing operations, but not sensitive enough to small-area tampering. Lin and Chang [5] present a technique for image authentication, which can distinguish JPEG compression from malicious attacks. They find that relations between DCT coefficients at the same position in separate blocks are preserved after JPEG compression. The method can also handle distortions introduced by various acceptable manipulations such as rounding, image filtering, image enhancement, and scaling-rescaling. However, this method is still vulnerable to some perceptually insignificant modifications that introduce distortion with statistical properties different from compression induced blurring.

In [6], Lefebvre et al. use the Radon transform to obtain image characteristics invariant against rotation and scaling. It is also robust against basic image processing operations and StirMark attacks. Another method [7] is basically a one-way function for images, which also uses the Radon transform together with principal component analysis (PCA) to extract characteristics robust against geometric transformation including rotation and scaling, and normal image manipulations such as compression, filtering, and blurring. However, it is not applicable to texture images.

In recent years, more works on image hashing have been reported. In [8], Kozat et al. view images and attacks as a sequence of linear operators, and propose to calculate hashes using transforms based on matrix invariants. They first construct a secondary image from the input image by pseudo-randomly extracting features that capture semi-global geometric characteristics. From the secondary image they extract the final features to be used as a hash value. The authors use spectral matrix invariants as embodied by singular value decomposition (SVD). The SVD-based hashing scheme improves robustness against geometric transformations at the expense of increasing collision possibility. In another work [9], a RAdial Variance (RAV) vector is first extracted using the radial projections of image pixels. The low-frequency DCT coefficients of the RAV vector are then quantized to form the image hash. This scheme is resilient to image rotation and re-scaling, but its collision risk is not low enough. Swaminathan et al. [10] propose to generate an image hash based on Fourier transform features and controlled randomization. They formulate robustness of image hashing as a hypothesis testing problem and evaluate the performance under various image processing operations. The hash function is resilient to several content-preserving modifications such as moderate geometric transformations and filtering. Since the hash is only dependent on magnitudes of Fourier coefficients but independent of phases, the attacker can easily create a similar image corresponding to a totally different hash or a synthesized image corresponding to the same hash [11]. In another study [12], Mao and Wu investigate the unicity distances of the hashing methods [2, 10] and conclude that the key reused several dozen times is no longer reliable. In [13], Li and Roy prove that information-theoretical security discussed in [10] is impossible if the attacker has unbounded computation power and is able to observe pairs of random images and their hashes.

Monga and Mihcak [14] first propose to derive the image hash using non-negative matrix factorization (NMF). They apply NMF to some sub-images, use factorization factors to construct a secondary image, and obtain a low-rank matrix approximation of the secondary image using NMF again. The matrix entries are concatenated to form an NMF-NMF vector. To get a short vector, they calculate the inner product between the NMF-NMF vector and a set of weight vectors which have i.i.d. Gaussian components of zero mean and unit variance. The NMF-NMF-SQ hashing has been shown to have good performances in robustness attacks, but its anti-collision capability is also inadequate.

3. Structural feature-based image hashing

As shown in Figure 1, the proposed hashing method consists of three steps. In the first step, we pre-process the input image to produce a normalized image for feature extraction. In the second step, the structural features are extracted to represent the normalized image. The structural features then undergo a sequence of operations to give a short and secure image hash. As an integrated part of the hashing scheme, a similarity metric is defined and taken into account in the hash generation.

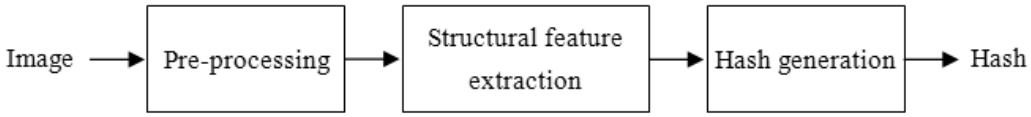


Figure 1. Three steps of image hashing.

3.1. Pre-processing

In the preprocessing, a set of manipulations including image resizing, color space conversion, and Gaussian low-pass filtering is applied to the input image. Image re-sizing changes the input image into a standard size $M \times M$ using bi-linear interpolation. This ensures the final hash to have the same length, and makes the hash scaling-resistant. For a color image, we only consider the luminance component $\mathbf{Y}$ of the YCrCb representation in the present work since the luminance plane contains the structural information in the image. Color features will be considered in the future to take into account color-related tampering. The resized $\mathbf{Y}$ plane is passed through a Gaussian low-pass filter to produce a pre-processed image denoted $\mathbf{U}$. The purpose of low-pass filtering is to reduce high frequency components and alleviate influences of minor image modifications, e.g., noise contamination and filtering, on the final hash value. An example of pre-processing is shown in Figure 2.

3.2. Structural feature extraction

Visually identical images have almost the same structural information, while different images contain different structures. In general, the human visual system (HVS) can distinguish various images by extracting their structural information. If an image is tampered, its original structure is destroyed by the inserted object. In Figure 3, (a) is an 80×80 image block, (b) is its low-pass filtered version, (c) is a

Figure 2. Image pre-processing.

different image block, and (d) is a tampered version of (a), obtained by inserting the central part of (c) into (a). In this case, (a) and (b) are visually alike and have similar structural information, while (a) and (c) have different structures. When a part of (c) is inserted into (a), it destroys the original structure of (a) to produce a different image (d).

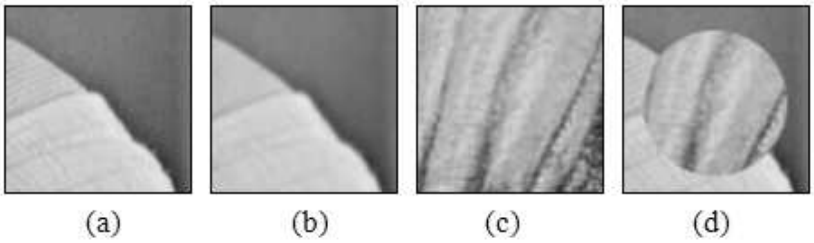


Figure 3. Structures of image blocks.

To capture the structure of a given image, a reference pattern, which may either be an image or a pseudo-random data array, is needed. We generate such a pattern using a similar method as described in [4], i.e., convolving a random array with a Gaussian mask iteratively. Since pixels in a small region are highly correlated, neighboring elements in the reference pattern should also be correlated. If the elements within a close neighborhood varied violently like a random noise pattern, the reference pattern would not effectively capture the image structure. In another extreme case, if all elements had a constant value, the extracted structure would have limited distinguishing capability. Thus, a relatively smooth pattern with some fluctuation is desired. That is the reason why Gaussian low-pass filter is used in the pattern generation.

Let $\mathbf{I}$ be an image, and $\mathbf{R}$ be the reference pattern. Following [15, 16], we use the correlation coefficient between $\mathbf{I}$ and $\mathbf{R}$ to measure the structure of $\mathbf{I}$:

$$\rho = \frac{\sigma_{\mathbf{I},\mathbf{R}}}{\sigma_{\mathbf{I}}\sigma_{\mathbf{R}}} \tag{1}$$

where $\sigma_{\mathbf{I}}$ and $\sigma_{\mathbf{R}}$ are standard deviations of $\mathbf{I}$ and $\mathbf{R}$, respectively, and $\sigma_{\mathbf{I},\mathbf{R}}$ is the covariance:

$$\sigma_{\mathbf{I},\mathbf{R}} = \frac{1}{Q-1}\sum_{j=1}^{Q}[I(j)-\mu_{\mathbf{I}}][R(j)-\mu_{\mathbf{R}}] \tag{2}$$

Q is the total pixel number in $\mathbf{I}$, $I(j)$ and $R(j)$ are the jth pixel value of $\mathbf{I}$ and $\mathbf{R}$, and μ_I and μ_R their means. If both standard deviations are zero, set $\rho = 1$. If only one of them is zero, let $\rho = 0$.

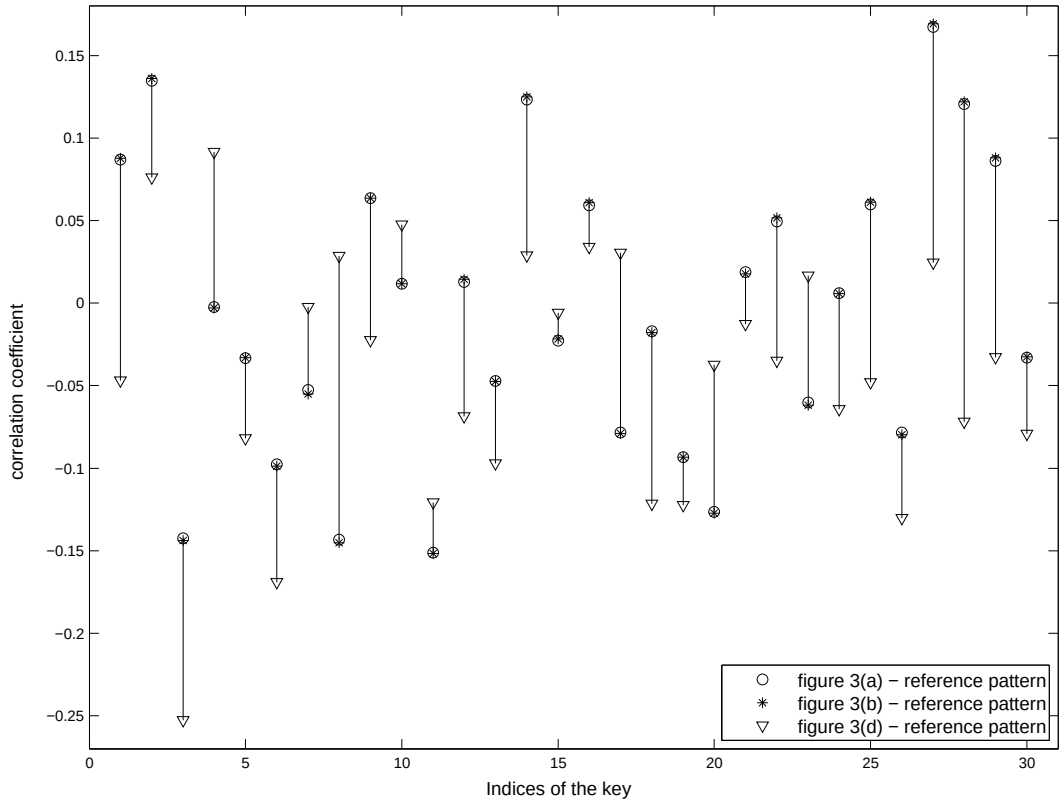


Figure 4. Correlation between 30 reference patterns and the images in Figure 3.

Figure 4 shows correlation coefficients between 30 different reference patterns and the image blocks in Figure 3. We observe that correlation between Figure 3(a) and the 30 reference patterns is very close to that between 3(b) and the same reference patterns. Correlation between the tampered version 3(d) and the same reference patterns is significantly different. This indicates that the structural feature defined in this section can be used to distinguish malicious tampering from normal manipulations.

To make the image hash sensitive to local modifications, we divide the $M \times M$ preprocessed image $\mathbf{U}$ into blocks. To do so, $\mathbf{U}$ is first divided into non-overlapping blocks of size $t \times t$. For convenience, we let M be an integer multiple of t. Expand the $t \times t$ blocks in each of the four directions by $t/2$ to make the blocks overlap. The expanded blocks are sized $2t \times 2t$. Blocks located on the rim of the

image are only expanded towards the inside of the image, and then resized to $2t \times 2t$ with bi-linear interpolation. Thus the total number of blocks is $N = (M/t)^2$. Figure 5 shows the blocking scheme, where (a) is an image divided into non-overlapping blocks, and the center part of (b) is an expanded block. To extract structural features of each block, we generate a reference pattern sized $2t \times 2t$ using a key, and calculate N correlation coefficients between each of the N blocks and the reference pattern. The correlation coefficients $\rho_1, \rho_2, \rho_3, \ldots, \rho_N$ are indexed from top to bottom and from left to right.

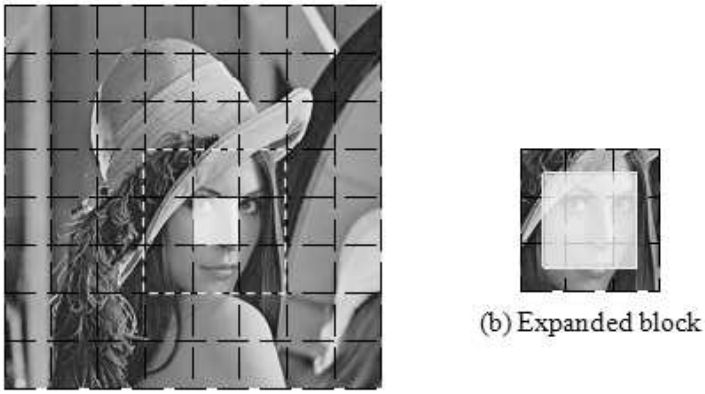


Figure 5. Overlapping blocks.

3.3. Hash generation

First, the correlation coefficient ρ_i is mapped to $c_i \in [0, 1]$:

$$c_i = \tfrac{1}{2}(\rho_i + 1) \quad i = 1, 2, \ldots, N \tag{3}$$

To make the final hash short, we reduce the number of coefficients to $L = \lceil N/2 \rceil$, where $\lceil \bullet \rceil$ means upward rounding:

$$q_i = c_{2i-1} c_{2i} \quad i = 1, 2, \ldots, L \tag{4}$$

If N is an odd number, set $c_{2L} = 1$. The sequence $q_1, q_2, \ldots, q_L$ is converted to a sequence of integers:

$$h_i = [100 q_i + 0.5] \quad i = 1, 2, \ldots, L \tag{5}$$

where $[\bullet]$ is a rounding operation. Since $q_i \in [0, 1]$, the range of h_i is $[0, 100]$. Therefore each hash element h_i can be represented by a 7-bit binary number. To enhance security, the concatenated binary string is scrambled based on a secret key. Thus the hash is $7L$ bits long. For example, if $\mathbf{U}$ sized 512×512 is divided into 64×64 overlapping blocks, $M = 512$ and $t = 32$, $L = (M/t)^2/2 = 128$ so that the hash length is $7 \times 128 = 896$ bits.

3.4. Similarity metric

As an integrated part of the hashing algorithm, we define a new similarity metric that fully explores both perceptual robustness and anti-tampering sensitivity intrinsic in the obtained image hash. Let $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ be the hashes of image $\mathbf{I}^{(1)}$ and $\mathbf{I}^{(2)}$, respectively. To measure similarity between $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$, the scrambled binary string are first decrypted to retrieve the hash elements $h_i^{(1)}$ and $h_i^{(2)}$. Then, calculate the following ratio derived from them:

$$r_i = \frac{\min[h_i^{(1)}, h_i^{(2)}]}{\max[h_i^{(1)}, h_i^{(2)}] + \varepsilon} \quad i = 1, 2, \ldots, L \tag{6}$$

where ε is a small constant to avoid singularity when $\max[h_i^{(1)}, h_i^{(2)}] = 0$. Clearly, the more similar the two image blocks, the smaller the difference between $h_i^{(1)}$ and $h_i^{(2)}$, thus the more the r_i value is close to 1. If the two input images are identical, all r_i are equal to 1. Therefore, we define the following similarity metric between $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$:

$$S(\mathbf{H}^{(1)}, \mathbf{H}^{(2)}) = \frac{\prod_{r_i \in \Gamma_{\text{small}}} r_i}{\prod_{r_i \in \Gamma_{\text{large}}} r_i + \varepsilon} \tag{7}$$

where Γ_{small} and Γ_{large} consist of m smallest r_i and m largest r_i, respectively.

The similarity metric defined here is a ratio between the least and the most similar pairs of blocks. $S \in [0, 1]$ because $r_i \in [0, 1]$. Similar images lead to a large value of S. If $\mathbf{I}^{(2)}$ is identical to $\mathbf{I}^{(1)}$, $S = 1$. If $\mathbf{I}^{(2)}$ is a tampered version of $\mathbf{I}^{(1)}$, there is at least one small r_i, resulting in a small S.

4. Experimental results

In the experiments, a 64×64 reference pattern is generated by 10 times iterative low-pass filtering using a 3×3 Gaussian mask with a unit standard deviation. The input image is resized to 512×512, low-pass filtered using a 3×3 Gaussian low-pass mask with a unit standard deviation, and divided into overlapping blocks of size 64×64, i.e., $M = 512$ and $t = 32$. Thus $L = 128$. The number of smallest and largest terms used to calculate the similarity metric in (7) is $m = 3$.

4.1. Perceptual robustness

To check robustness of the proposed method against content-preserving manipulations, we use StirMark 4.0 [17] to perform attacks on five test images Airplane, Baboon, House, Peppers, and Lena, all sized 512×512. Content-preserving manipulations used in the experiment include JPEG compression, watermark embedding, additive noise contamination, and image rescaling. In addition, brightness adjustment and contrast adjustment with Photoshop, and gamma correction and 3×3 Gaussian filtering using MATLAB are also tested. Parameters used in these operations are listed in Table 1. Calculate the hash similarity S between the original images and their attacked versions, and then obtain the results as shown in Figure 6. Indices of the abscissa indicate various attacks on the image as listed in Table 1. The ordinate is the similarity S between the original and attacked images. It is observed that all the results are greater than 0.7. The results show that the proposed method is robust against content preserving processing. In this case, we can safely set a threshold $T = 0.6$. If for a pair of images S is greater than T, they are

considered as visually identical. Otherwise, the two images are different, or one is a tampered version of the other.

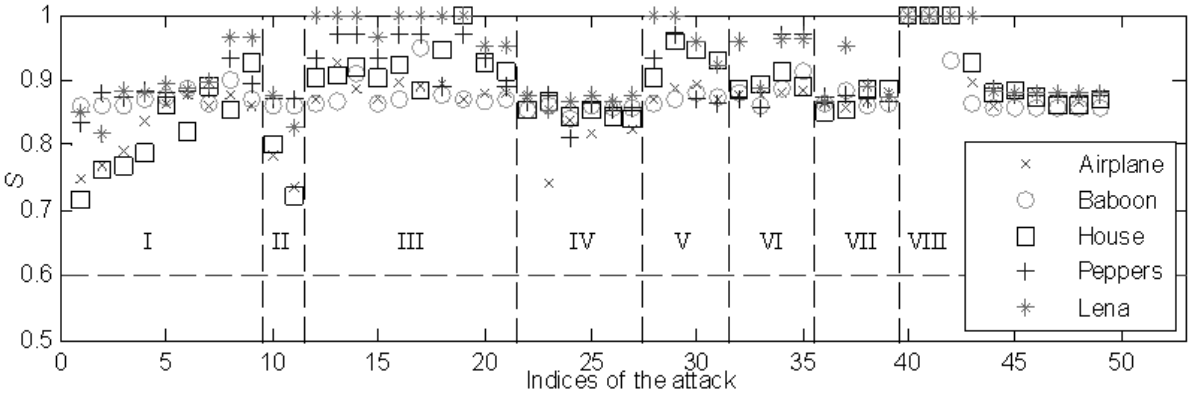


Figure 6. Robustness performance. Indices in the abscissa indicate different attacks as listed in Table 1.

Table 1. Manipulations corresponding to indices in Figure 6

Indices	Code	Manipulation	Description	Parameter value
1˜9	I	JPEG compression	Quality factor	$20, 30, \ldots, 100$
10˜11	II	Additive noise	Level	$1, 2$
12˜21	III	Watermark embedding	Strengthen	$10, 20, \ldots, 100$
22˜27	IV	Scaling	Ratio	$0.5, 0.75, 0.9, 1.1, 1.5, 2.0$
28˜31	V	Brightness adjustment	Photoshop's scale	$10, 20, -10, -20$
32˜35	VI	Contrast adjustment	Photoshop's scale	$10, 20, -10, -20$
36˜39	VII	Gamma correction	γ	$0.75, 0.9, 1.1, 1.25$
40˜49	VIII	3×3 Gaussian filtering	Standard deviation	$0.1, 0.2, \ldots, 1.0$

4.2. Collision probability

Collision happens if similarity S between two different images is greater than the threshold T. To determine the collision probability, we collect 2000 test color images, including 12 standard images of sizes 256×256 or 512×512, 100 images captured with digital cameras with sizes ranging from 1280×960 to 2592×1944, 200 images downloaded from the Internet with sizes ranging from 415×260 to 853×530, and 1688 images from the image database of Washington University [18] with sizes ranging from 480×722 to 883×589. Generate hashes of these images, and calculate S between each pair of hashes. Thus, $1,999,000$ results are obtained. On a desktop computer with a 2.8GHz Pentium D CPU and 512 MB RAM, it spent about 157 minutes in hash extraction and 183 seconds in similarity calculation.

Chi-square tests [19] are performed on all $1,999,000$ results to identify the most probable distribution satisfied by S among Poisson, Rayleigh, Weibull, lognormal, normal, and Gamma distributions. Since $S \in [0,1]$, we first quantize the interval with a step size 0.005 and obtain a set of discrete points, i.e., $0, 0.005, 0.01, \ldots, 1$. Parameters of these distributions are obtained from the maximum likelihood estimation, and χ^2 calculated:

$$\chi^2 = \sum_{i=0}^{K} \frac{(n_i - np_i)^2}{np_i} \tag{8}$$

where n is the number of trials, n_i occurring frequency of the similarity S being i ($i = 0, 0.005, 0.01, \ldots, 1$), K the total number of discrete points, and p_i the probability at i obtained by using the probability density function (PDF). The results are given in Table 2. Since χ^2 of Gamma distribution is the smallest, we may consider that S obeys the Gamma distribution with $a = 33.85$ and $b = 0.007$.

Table 2. Results of Chi-square test for S

Distribution type	Estimated parameters	χ^2
Poisson	$\lambda = 0.238$	6.26×10^{26}
Rayleigh	$\beta = 0.171$	2.6×10^{6}
Weibull	$\beta = 0.256, \eta = 6.23$	4.5×10^{15}
Lognormal	$\mu = -1.45, \sigma = 0.174$	3.96×10^{8}
Normal	$\mu = 0.252, \sigma = 0.041$	1.66×10^{4}
Gamma	$a = 33.85, b = 0.007$	9.65×10^{3}

Figure 7 compares the data set obtained in the above and the ideal Gamma distribution. Given a threshold T, the collision probability can be calculated:

$$P(S > T) = 1 - \int_0^T \frac{1}{b^a \Gamma(a)} x^{(a-1)} e^{\frac{-x}{b}} \, \mathrm{d}x \tag{9}$$

where $\Gamma(a)$ is the Gamma function:

$$\Gamma(a) = \int_0^{+\infty} t^{a-1} e^{-t} \, \mathrm{d}t \tag{10}$$

As T increases, the calculated collision probability will decrease, while the probability of false judgment for content-preserving modification will rise. In the experiments, we have chosen a $T = 0.6$ to give a satisfactory balance between probabilities of collision and robustness. In this case, the collision probability is 4.68×10^{-11}.

4.3. Tampering detection

In this section, we present experimental results to show capability of the proposed hashing method in detecting object insertion and deletion. Pairs of original images and their tampered versions are tested. Several examples are presented in Table 3. The third and the fourth columns are the original and tampered

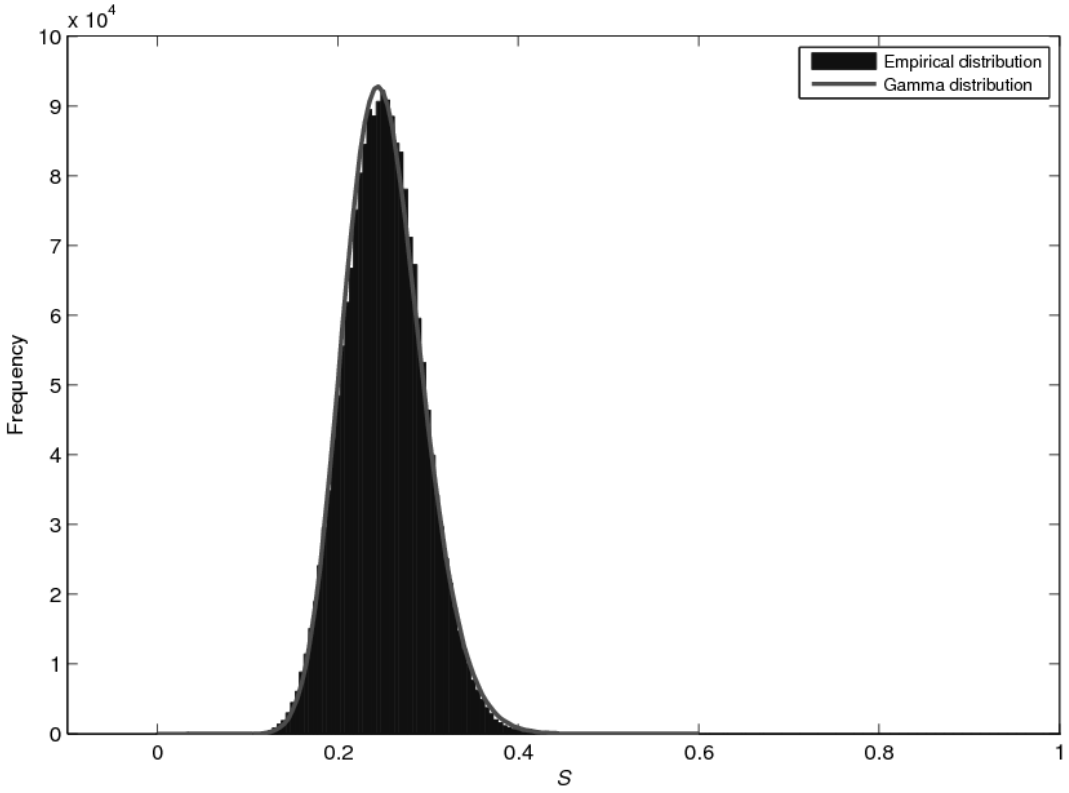


Figure 7. The Gamma distribution and experimental data set.

versions, respectively, and the last column gives the image size. The detected results are listed in Table 4, in which all S values are below the threshold $T = 0.6$.

5. Performance comparison

In this section, we compare the proposed method with three previous techniques, Fridirch's method [4], RASH method [9], and NMF-NMF-SQ scheme [14]. For these, we use the same images for robustness validation, collision analysis, and tampering detection. Only the luminance component $\mathbf{Y}$ of color images is considered. The distance metrics used in the respective papers are adopted in the comparison.

5.1. Fridrich's method

To compare with [4], the images are also resized to 512×512 as in our experiments, and then divided into 64×64 non-overlapping blocks. For each block, $N = 8$ bits are extracted, thus the hash length is 512 bits. Calculate Hamming distances between hashes of the original and modified images to produce results shown in Figure 8. The ordinate is the Hamming distance d. With a threshold $T = 40$, the

Table 3. Original images and their corresponding tampered images

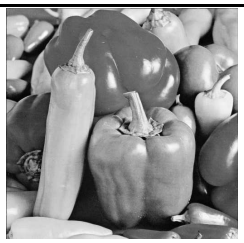
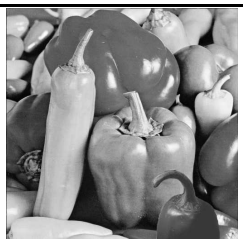
No.	Type	Original image	Tampered image	Size
1	Insert			512×512
2	Insert			600×399
3	Insert			1280×1024
4	Delete			570×395
5	Delete			543×407
6	Delete			500×340

Table 4. S values between the original and tampered images listed in Table 3

Image pair No.	1	2	3	4	5	6
S	0.398	0.480	0.444	0.480	0.496	0.487

method described in [4] is robust against the same set of content-preserving manipulations as tested in our experiments.

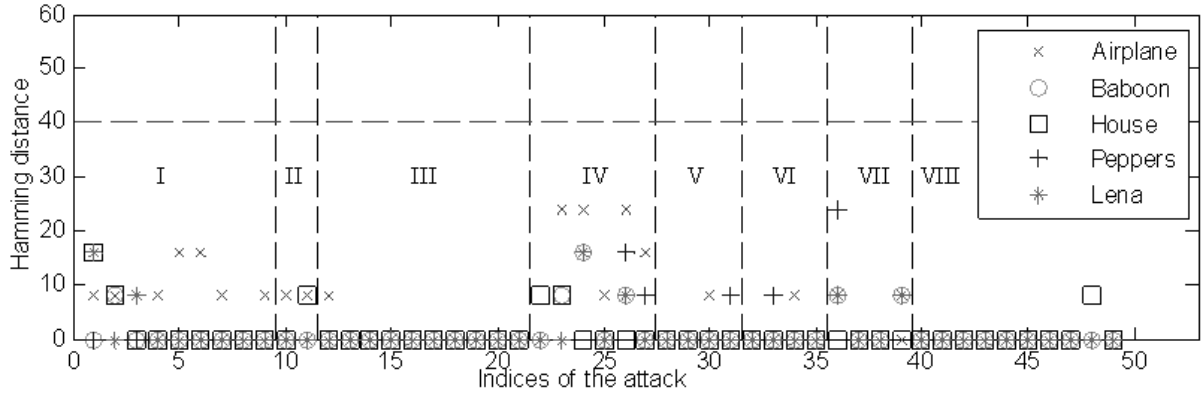


Figure 8. Robustness of the method in [4]. Indices of the abscissa indicate different attacks as listed in Table 1.

To check the anti-collision performance, generate hashes of $2,000$ images used in Subsection 4.2, and compute the Hamming distance d between each pairs of them. Based on a Chi-square test, it is found that d obeys a normal distribution with the mean $\mu = 255.3$ and standard deviation $\sigma = 32.4$. With a threshold $T = 40$, the collision probability is 1.52×10^{-11}. This is in the same order of magnitude with our method.

Apply the method in [4] to each pair of the original and tampered images used in Subsection 4.3, and calculate the Hamming distances. The results are given in Table 5. Of the six Hamming distances, only one reaches $T = 40$, and the others are all below 40. This means that the hash is not sensitive to local changes in the image. In other words, content changes in a small area do not have enough influence on the final hash value.

Table 5. Hamming distances between the original and tampered images presented in Table 3

Image pair No.	1	2	3	4	5	6
Hamming distance	24	16	32	16	40	16

5.2. RASH method

The RASH method [9] exploits peak of cross correlation (PCC) to measure similarity between two hashes. The authors take 0.87 as the threshold. If PCC > 0.87, the corresponding images are considered perceptually similar. Figure 9 shows the PCC values between hashes of the original and modified images. We observe that all PCC values are greater than the given threshold 0.87, showing the RASH method's robustness against content-preserving modifications. Collision probability reported in [9] is 5.86×10^{-6} that is significantly worse than the proposed method.

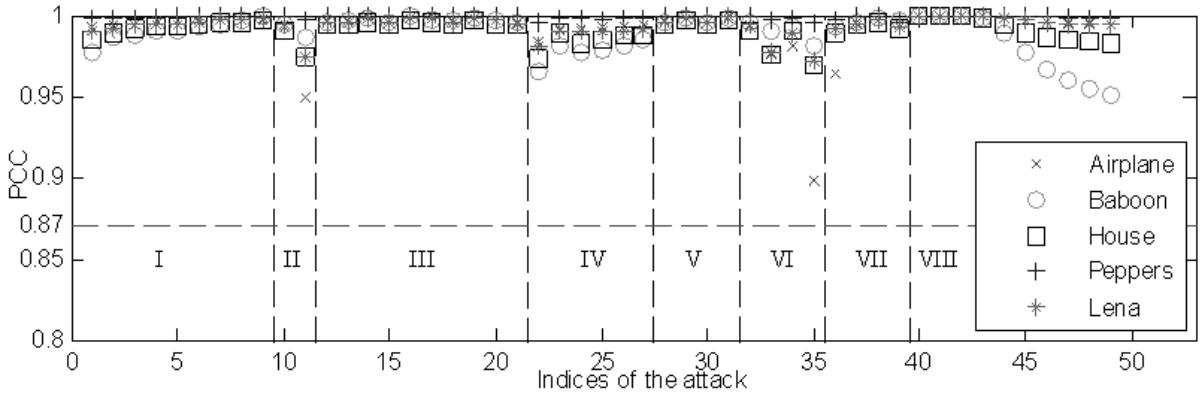


Figure 9. Robustness of RASH method. Indices of the abscissa indicate different attacks as listed in Table 1.

PCC values between the hashes of the original and tampered images listed in Table 3 are given in Table 6. All the results are greater than the threshold value 0.87. It means that the RASH method cannot detect small-area tampering. This is because the radial variance (RAV)

Table 6. PCC values between the hashes of the original and tampered images listed in Table 3

Image pair No.	1	2	3	4	5	6
PCC value	0.991	0.978	0.982	0.968	0.928	0.989

extracted from the image is a global-based feature, which is unable to capture local changes in the image.

5.3. NMF-NMF-SQ hash scheme

To compare with [14], all images are normalized to a fixed-size 512×512 before NMF-NMF-SQ hash extraction. The parameters used are: number of sub-images $p = 80$, length and width of sub-images $m = 64$, rank of the first NMF $r_1 = 2$, rank of the second NMF $r_2 = 1$, and the hash length $M = 64$.

Calculate the L2 norm between hashes of the original and modified images. The results are shown in Figure 10. The ordinate denotes the L2 norm τ. All the τ values are below 2800 except a few cases

of strong watermarking attacks, brightness adjustments, and gamma corrections. The τ value increases approximately linearly with the increasing watermark strength.

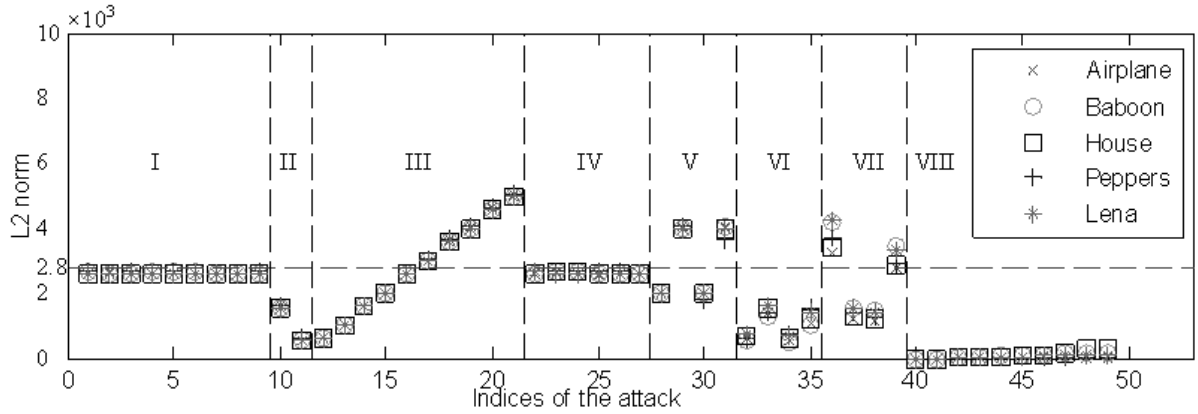


Figure 10. Robustness of NMF-NMF-SQ hash. Indices of the abscissa indicate different attacks as listed in Table 1.

To check its anti-collision characteristics, generate NMF-NMF-SQ hashes of the same $2,000$ images as used in Section 4, and compute the L2 norm between each pair of two hashes. A Chi-square test shows that the L2 norm satisfies a Gamma distribution with $a = 9.93$ and $b = 1513$. With a threshold $T = 2800$, collision probability is 2.76×10^{-5}, even greater than that of RASH.

Compute the L2 norm between hashes of the original and tampered images given in Table 3, and list the results in Table 7. None of these L2 norms are above the threshold value 2800, showing that the NMF-NMF-SQ hash is also insensitive to small-area tampering. An advantage of RASH and NMF-NMF-SQ hashes is their capability of resisting rotation attacks.

In the above comparisons, the proposed method has shown similar perceptual robustness with [4], and also with RASH and NMF-NMF-SQ methods except for anti-rotation performance. The proposed method has the best anti-collision performance. As to the capability of detecting small area tampering, the proposed technique is the best among all the tested methods. The average length of the proposed hash is 896 bits, and those of [4] and [9] are 512 bits and 320 bits, respectively. The NMF-NMF-SQ hash contains M decimal digits. Taking Baboon as an example, $M = 64$ with the decimal entries ranging from -10369 to 7391, thus each entry needs at least 15 bits for storage. Therefore in a binary form, the hash has 960 bits, longer than the other three methods.

Table 7. L2 norms between the hashes of the original and tampered images listed in Table 3

Image pair No.	1	2	3	4	5	6
L2 norm	2203	1542	1997	1437	521	1162

6. Conclusions

Structural information can effectively represent the principal visual contents of an image. In this paper, we have developed a perceptual image hashing method by extracting the structural features of overlapping blocks, which makes the hash sensitive to improper modifications of small image areas whereas maintains perceptual robustness to normal image processing. The structural feature is obtained by calculating the correlation coefficient between image blocks and a reference pattern. Since the reference pattern is controlled by a secret key, the extracted feature is very difficult to guess without knowledge of the key. This can avoid security loopholes existing in some early techniques such as discussed in [11]. To further improve security of the final hash, we can always scramble the binary sequence with a separate key. Therefore, attempts to forge an image hash are virtually impossible to succeed.

Experimental results show that the proposed hash is robust against a set of content-preserving manipulations including JPEG coding, noise contamination, watermarking, rescaling, adjustments of brightness and contrast, and low-pass filtering. Probability of collision between hashes of different images is very low. A central part of the method is the proposed similarity metric. It is designed specifically to reveal small area alterations and, when used in conjunction with the block-based features, provides the capability of exposing local tampering. This makes the proposed method suitable for in tampering detection applications.

Further research on robust image hashing is in order. This includes establishing a new image hashing framework, and consideration of more types of manipulation such as rotation and color changes.

References

[1] Zhang X., and Wang S.:Fragile watermarking with error-free restoration capability,*IEEE Transactions on Multimedia*, 2008, 10, (8), pp.1490–1499.

[2] Venkatesan R., Koon S.-M., Jakubowski M. H., and Moulin P.:Robust image hashing, *Proc. of IEEE International Conference on Image Processing*, Vancouver, BC, Canada, September 2000, pp.664–666.

[3] Meixner A., and Uhl A.:Analysis of a wavelet-based robust hash algorithm, *Proc. SPIE-IS&T - Security, Steganography, and Watermarking of Multimedia Contents VI*, San Jose, CA, January 2004, pp.772–783.

[4] Fridrich J., and Goljan M.:Robust hash functions for digital watermarking, *Proc. of IEEE International Conference on Information Technology: Coding and Computing*, Las Vergas, USA, March 2000, pp.178–183.

[5] Lin C. Y., and Chang S. F.:A robust image authentication system distinguishing JPEG compression from malicious manipulation, *IEEE Transactions on Circuits System and Video Technology*, 2001, 11, (2), pp.153–168.

[6] Lefebvre F., Macq B., and Legat J.-D.:RASH: Radon soft hash algorithm, *Proc. of European Signal Processing Conference*, Toulouse, France, September 2002, pp.299–302.

[7] Lefebvre F., Czyz J., and Macq B.:A robust soft hash algorithm for digital image signature, *Proc. of IEEE International Conference on Image Processing*, September 2003, pp.495–498.

[8] Kozat S. S., Mihcak K., and Venkatesan R.:Robust perceptual image hashing via matrix invariants, *Proc. of IEEE International Conference on Image Processing*, Singapore, October 2004, pp.3443–3446.

[9] Roover C. D., Vleeschouwer C. D., Lefebvre F., and Macq B.:Robust video hashing based on radial projections of key frames, *IEEE Transactions on Signal Processing*, 2005, 53, (10), pp.4020–4036.

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

[11] Wang S., and Zhang X.:Attacks on perceptual image hashing, *Proc. of the 2nd International Conference on Ubiquitous Information Technologies and Applications*, Bali, Indonesia, Dec. 2007, pp.199–203.

[12] Mao Y. and Wu M.:Unicity distance of robust image hashing, *IEEE Transactions on Information Forensics and Security*, 2007, 2, (3), pp.462–467.

[13] Li Q., and Roy S.:On the security of non-forgeable robust hash functions, *Proc. of IEEE International Conference on Image Processing*, San Diego, California, USA, October 2008, pp.3124–3127.

[14] Monga V., and Mihcak M. K.:Robust and secure image hashing via non-negative matrix factorizations, *IEEE Transactions on Information Forensics and Security*, 2007, 2, (3), pp.376–390.

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

[16] Tang Z., Wang S., Zhang X., and Wei W.:Perceptual similarity metric resilient to rotation for application in robust image hashing, *Proc. of the 3rd International Conference on Multimedia and Ubiquitous Engineering*, Qingdao, China, June 4-6, 2009, pp.183–188.

[17] Petitcolas F. A. P.:Watermarking schemes evaluation, *IEEE Signal Processing Magazine*, 2000, 17, (5), pp.58–64.

[18] Ground Truth Database.[online]. Available: http://www.cs.washington.edu/research/imagedatabase/groundtruth/, accessed May 2008.

[19] Lehmann E. L., and Romano J. P.:*Testing Statistical Hypotheses*. (New York, USA, Springer, 2005, 3rd Ed), pp.590–599.