Perceptual image hashing using center-symmetric local binary patterns

Article *in* Multimedia Tools and Applications · March 2015

DOI: 10.1007/s11042-015-2496-6

3 authors, including:

Reza Davarzani
Islamic Azad University, Shahrood Branch, Shahrood, Iran
8 PUBLICATIONS **251** CITATIONS
SEE PROFILE

Saeed Mozaffari
Semnan University
77 PUBLICATIONS **1,208** CITATIONS
SEE PROFILE

Multimed Tools Appl
DOI 10.1007/s11042-015-2496-6

Perceptual image hashing using center-symmetric local binary patterns

Reza Davarzani · Saeed Mozaffari · Khashayar Yaghmaie

Received: 3 August 2014 / Revised: 24 December 2014 / Accepted: 4 February 2015

Abstract Perceptual image hashing finds increasing attention in several multimedia security applications such as image identification/authentication, tamper detection, and watermarking. Robust feature extraction is the main challenge in hashing schemes. Local binary pattern (LBP) is a new feature which is due to its simplicity, discriminative power, computational efficiency, and robustness to illumination changes has been used in various image applications. In this paper, we propose a robust image hashing scheme using center-symmetric local binary patterns (CSLBP). In the proposed image hashing, CSLBP features are extracted from each non-overlapping block within the original gray-scale image. For each block, the final hash code is obtained by inner product of its CSLBP feature vector and a pseudorandom weight vector. Furthermore, singular value decomposition (SVD) is combined with CSLBP to introduce a more robust hashing method called SVD-CSLBP. Performances of the proposed hashing schemes are evaluated with two groups of popular applications in perceptual image hashing schemes: image identification and image authentication. Experimental results show that the proposed methods are robust to a wide range of distortions and attacks such as additive noise, blurring, brightness changes and JPEG compression. Moreover, the proposed methods have this capability to localize the tampering area, which is not possible in all hashing schemes.

Keywords Center-symmetric local binary patterns · Perceptual image hashing · Singular value decomposition (SVD) · Tamper detection

1 Introduction

In recent years, we have witnessed the development of multimedia information in many aspects of our daily lives. Advantages of digital multimedia have led to fast progress on media acquisition tools, powerful hardware, sophisticated editing software and network technologies that provide various media sharing and streaming services. Multimedia finds its application in various areas including, scientific research, art, entertainment, engineering, medicine, business

R. Davarzani · S. Mozaffari (✉) · K. Yaghmaie
Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran
e-mail: mozaffari@semnan.ac.ir

Published online: 03 March 2015

and etc. However, digital multimedia suffers from illegal access and unauthorized distributions. Professional forgers with advanced technology can alter multimedia data without any trail on forged information. Therefore, it is necessary to create new tools and techniques to discover the authenticity and integrity of digital media [14].

In recent decade, several methods have been extensively studied for intellectual property protection of digital images and image forgeries detection. These methods can be categorized into three groups: image watermarking-based schemes [5], digital image forensic-based schemes [2], and perceptual image hashing- based schemes [33, 39]. In *watermarking techniques*, the watermark information and ownership identification are embedded imperceptibly into the digital image. With the assumption that tampering will alter a watermark, an image can be authenticated by verifying the extracted watermark. However, embedding process would inevitably cause some degradation in the quality of image content which is not acceptable in some applications. Digital *image forensics techniques* try to detect potential tampering in digital images without any watermark or hash attachment. The core of these techniques is appropriate feature extraction which depends on the type of image forgery [9]. Such forensic techniques typically focus on similarity/inconsistency in the intrinsic features and signal statistics to detect potential processing operations. However, lack of any side information about the original data makes them computationally intensive.

For security/authentication of digital images, *perceptual image hashing* was introduced based on traditional cryptosystems [21]. Traditional cryptographic hash functions have been used for data integrity and retrieval [30]. However, it should be noted that the purpose of a cryptographic hash function and a perceptual image hash function is totally different. Hash functions in traditional cryptosystems are very sensitive that changing even one bit in the input considerably changes the output. However, digital images should undergo content-preserving manipulations such as compression, enhancement, cropping, and scaling. An image hash function should tolerate such permissible changes and produce similar hash values for images with the same visual appearance [40]. On the other hand, perceptual hashing system should be sensitive to content-changing distortions and reject malicious manipulations and attacks. Perceptual image hash functions generally consist of two steps, as illustrated in Fig. 1 [21]. The first step extracts a feature vector from the image which depends on the image content or its characteristics. In the second step, this feature vector is compressed and quantized into a binary or real number sequence to form the final hash value. Since the image hash also serves as a secure tag, a secret key is incorporated into either feature extraction or hash generation to guarantee that the hash value is hardly obtainable by unauthorized adversaries without the secret key.

In this paper, we investigate the use of local binary patterns (LBP) for perceptual image hashing. In feature extraction step, we propose to use both sign and magnitude information of local differences between neighboring pixels. So, the algorithm combines gradient-based and LBP-based descriptors for feature extraction. Since LBP features are sensitive to noise,

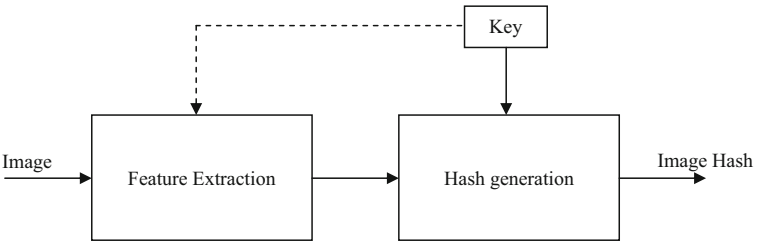


Fig. 1 Two main stages of the general image hashing scheme

singular value decomposition (SVD) is performed as a pre-processing step for hash generation. To increase security, two secret keys are incorporated in feature extraction and hash generation steps. Several experiments show that the proposed algorithm can reach a good balance between robustness and discrimination and outperforms some well-known algorithms. High robustness against luminance changes, applicability in image tampering detection, acceptable hash length and running time are advantages of the proposed hashing method.

This paper is organized as follows. In Section 2 the basic concepts of perceptual image hashing are presented. Sections 3 and 4 review the background literature of perceptual image hashing methods and center-symmetric local binary patterns, respectively. Section 5 presents the proposed new image hashing schemes. Experimental results are given in Section 6. Finally, conclusion is drawn in Section 7.

2 Basic concepts of perceptual image hashing

Assume that the input image is shown as I and its perceptually similar and distinct forms are defined as I_{ident} and I_{diff}, respectively. Let P denotes probability and H_k indicates an image hashing function depending on a secret key k. Assume that H_k function produces a binary hash string of length l from the input image. The desirable properties of a perceptual image hash are summarized as follows [21]:

1). Perceptual Robustness
Perceptually identical images should have similar hashes.

$$P(H_k(I)\approx H_k(I_{ident}))\geq 1-\varepsilon,\ \ 0\leq\varepsilon < 1 \tag{1}$$

2). Uniqueness
Perceptually distinct images should have unique signatures.

$$P\left(H_k(I)\neq H_k\left(I_{diff}\right)\right)\geq 1-\theta,\ \ 0\leq\theta < 1 \tag{2}$$

3). Unpredictability
Equal distribution of hash values.

$$P(H_k(I)=h_I)\approx\frac{1}{2^l},\ \ \forall h_I\in\{0,1\}^l \tag{3}$$

where h_I is the l-bit binary hash value for image I.

4). Compactness
The size of the hash signature should be much smaller than the original image I.

$$\mathrm{Size}(H_k(I))\ll\mathrm{Size}(I) \tag{4}$$

5). Pair-wise independence for perceptually different images I and I_{diff}

$$P\left(H_k(I)=h_I|H_k\left(I_{diff}\right)=h_{I_{diff}}\right)\approx P(H_k(I_{ident})=h_I),\ \ \forall h_I,h_{I_{diff}}\in\{0,1\}^l \tag{5}$$

3 Related works

Feature extraction is the key step in the image hashing which differentiates perceptual hashing methods. The aim of feature extraction is to provide a compact and robust representation of the

image content. The extracted features are quantized into a binary or real numbers sequence to form the final hash value. Proposed hash generation algorithms can be classified into three categories based on their feature extraction method: transform based schemes, matrix factorization based schemes, and local feature pattern based schemes.

3.1 Transform based schemes

Classical transformations like DCT and DFT have been used frequently for features extraction. Various properties of DCT can be used to create perceptual image hash functions. For example, low-frequency DCT coefficients of an image are mostly stable under image manipulations. Fridrich and Goljan proposed a robust hashing algorithm based on the stability of low-frequency DCT coefficients [10]. For two different 8×8 blocks of an image, DCT coefficients at the same position represent invariant relationships before and after JPEG compression. In [3], low-frequency DCT coefficients were used for an effective image authentication. Although such coefficients are robust to JPEG compression, they are vulnerable to several other perceptually insignificant modifications such as blurring.

Fourier-Mellin transform (FMT) is scale and rotation invariant. Magnitudes of the Fourier transform coefficients which were randomly weighted and summed are utilized for perceptual image hashing [36]. This method is robust to various content-preserving manipulations such as geometric distortions, filtering operations, and etc. They also proposed a new framework to evaluate the security issues of image hashing schemes based on the differential entropy of hash values. Another method based on the FMT and compressive sensing was proposed in [35]. In this algorithm FMT is used to provide robustness against geometric attacks and the property of dimension reduction is exploited for hash design.

In recent developments, Radon transform has been used for image hashing [8, 18]. Radon transform is used to divide the image into radial projections and build a RAdial Variance (RAV) vector of image pixels. Then, the first 40 DCT coefficients of the RAV vectors are converted into the image hash called RASH. This method is scaling and rotation invariant but its discriminative capability needs to be improved. In [19], moment features are extracted from Radon transform coefficients of input image. Then the discrete Fourier transform (DFT) is applied on the moment features to achieve rotation-invariant property. Finally, the perceptual image hash is constructed by normalizing and quantizing magnitude of the significant DFT coefficients. This method has good detection performance, perceptual robustness and hash size.

Venkatesan et al. proposed a perceptual image hashing technique based on the quantized statistics of randomized rectangles in the discrete wavelet domain (DWT) [38]. In their method, averages or variances of the random blocks in wavelet image are computed and then quantized using a randomized rounding to form a secure binary hash. This method is robust against a limited range of geometric attacks but is sensitive to contrast adjustment and gamma correction. To use both advantages of the frequency localization property of DWT and the shift/rotation invariant property of the Radon transform, they are combined together to generate a robust perceptual image hash [11].

3.2 Matrix factorization based schemes

In the second type of image hashing schemes, the advantages of matrix factorization or decomposition are used to extract the image features. In this category, singular value

decomposition (SVD) and non negative matrix factorization (NMF) are popular. Kozat et al. proposed a two steps hashing algorithm based on SVD [17]. The method is robust against some small variations in rotation and scaling. In another work Non-negative Matrix Factorization (NMF) is used as a dimensionality reduction technique for image hashing [24]. Although, this method is can tolerate a large class of perceptually insignificant attacks, it is vulnerable to brightness changes and large geometric transformations.

A lexicographical image hashing system has been proposed in [37]. The method consists of two parts: dictionary construction and maintenance, and hash generation. The dictionary is constructed based on a large collection of feature vectors called words taken from various image blocks. Under the framework, discrete cosine transform (DCT) and non-negative matrix factorization (NMF) were used to implement the hashing scheme. Fast Johnson-Lindenstrauss transform (FJLT) is used for image hashing which can be categorized in dimension reduction group [41].

3.3 Local feature pattern based schemes

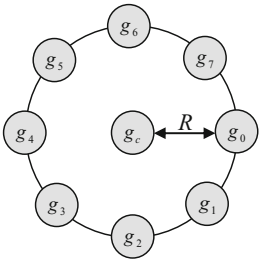
In the third group of perceptual image hashing methods, local feature patterns are used to represent the content of each image. They include low-level features such as edges, corners, blobs, salient regions, key points and so on. Generally, the main advantage of using local feature patterns lies in their robustness against geometric distortions, including rotation, scaling, shearing and etc. But local features are sensitive to classical attacks such as noise addition, blurring, and compression which make them unpractical for image hashing. Improving the strength of local feature patterns against the above attacks, based hashing schemes, is an open issue for future investigations. Local feature extraction is time consuming task and their average running time are higher than the other methods. [31] and [42] use robust local SIFT features for image hashing.

4 Center-symmetric local binary patterns (CSLBP)

Among the feature descriptors, Local Binary Patterns (LBP) is one of the most famous and powerful ones. The idea of LBP is originally proposed by Ojala et al. [26, 27] for texture classification. Because of its low computational complexity and invariance to monotonic gray-scale changes, LBP has been used for a wide range of applications including : texture classification [6, 27, 43], face analysis [1, 20, 32], detecting moving objects [15], copy-move image forgery [7], image region descriptors [13, 16], and so on.

In the original LBP, signed gray level differences of each pixel with its neighboring pixels are described as a binary form. However, the LBP operator produces rather long histograms and it is therefore difficult to use in the context of a region descriptor. Furthermore, the original LBP feature is not robust on flat images. To address the problems, center-symmetric local binary patterns, CSLBP, as a modified version of LBP was proposed in [16].

Let $I(x,y)$ be a gray level image and g_c indicates the gray level of an arbitrary pixel positioned at (x_c,y_c), i.e., $g_c=I(x_c,y_c)$ .Gray values of P equally spaced circular neighborhood pixels on a circle of radius $R(R>0)$ around g_c are shown by $g_p, p = 0, 1, \ldots, P-1$, (See Fig. 2). The CSLBP form shown by $CS\_LBP_{P,R,T}(x_c,y_c)$ is obtained as follows:



$$CSLBP_{8,R,T}(g_c) = 2^0 \times s(g_0 - g_4) + 2^1 \times s(g_1 - g_5) + 2^2 \times s(g_2 - g_6) + 2^3 \times s(g_3 - g_7)$$

Fig. 2 Circularly symmetric neighborhoods and CSLBP feature for radius R and P=8 neighborhood pixels

$$\begin{aligned} g_p &= I\left(x_p, y_p\right),\ p = 0, \ldots, P-1 \\ x_p &= x_c + R\ \cos(2\pi p/P) \\ y_p &= y_c - R\ \sin(2\pi p/P) \\ CSLBP_{P,R,T}(x_c, y_c) &= \sum_{p=0}^{P/2-1} s\left(g_p - g_{p+(P/2)}\right) 2^p, \\ s(x) &= \begin{cases} 1, & x > T \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{6}$$

where, T is a small value used to threshold the gray-level difference to increase the robustness of the CSLBP feature on flat image regions. CSLBP captures better gradient information than the basic LBP, because instead of comparing the gray-level of each pixel with the center pixel, gray-level differences between center-symmetric pairs of opposite pixels in a neighborhood are compared.

5 Proposed algorithm for perceptual image hashing

In this section, we propose image hashes generation based on block feature extraction using center-symmetric local binary patterns (CSLBP). Furthermore, since noise sensitivity is the fundamental weakness of LBP features, SVD-CSLBP-based hashing method is introduced. As shown in Fig. 3, the proposed technique consists of three steps: pre-processing, feature extraction, and hash generation. To guarantee the security requirements, two secret keys, *K1* and *K2*, are also incorporated in feature extraction and hash generation steps. The following sections describe our motivation and details of hashing algorithms.

5.1 Motivation

Feature extraction is one of the most important parts in different image hashing methods. Robust features make image hashing more resilient against different types of content-

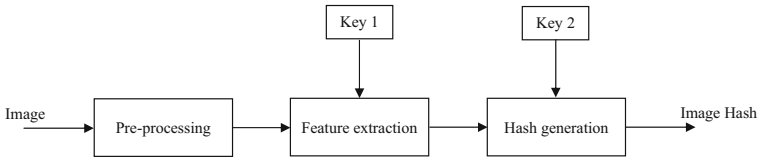


Fig. 3 Block diagram of the proposed hashing algorithms

preserving distortions. LBP and SVD are used for hash generation because of their desirable properties:

1) One of the important factors in perceptual image hashing is hashing algorithm's speed. LBP has low computational complexity that makes it a good candidate for this purpose.
2) In perceptual image hashing, a unique signature should be extracted from the input image based on its appearance. So, a highly discriminative feature is essential here. LBP can be regarded as a unifying approach to traditionally divergent statistical and structural models of texture image description [28]. From this point of view, LBP is a good option for content-based feature extraction.
3) The proposed CSLBP-based hashing method belongs to the group of transform based hashing schemes. Because LBP operator, transforms the input image into a gray-scale invariant image [27]. Assume T as a local neighborhood of a monochrome texture image $I(x,y)$ which is characterized by the joint distribution of gray values of $P(P>1)$ pixels:

$$T = t(g_c, g_0, g_1 \ldots, g_{P-1}) \tag{7}$$

In the original LBP, without loss of information, the center pixel value is subtracted from the gray values of local neighborhood:

$$T = t(g_c, g_0 - g_c, g_1 - g_c \ldots, g_{P-1} - g_c) \tag{8}$$

To achieve invariance with respect to shifts in gray scale, assume that differences $(g_p - g_c)$ are independent of g_c which allows us to have an approximation of joint distribution by factorize $t(g_c)$:

$$T \approx t(g_c)\ t(g_0 - g_c, g_1 - g_c \ldots, g_{P-1} - g_c) \tag{9}$$

Since the first factor $t(g_c)$ describes the overall luminance of the image $I(x,y)$, it contains no related information to local texture patterns and can be disregarded for texture analysis. Instead, much of the information about the textural characteristics is located in the joint distribution of local differences.

$$T \approx t\left(g_0 - g_c, g_1 - g_c \ldots, g_{P-1} - g_c\right) \tag{10}$$

Finally, since the sign of local differences $(g_p - g_c)$ are not affected by luminance changes, to achieve gray scale invariant features, only the signs of the differences are considered.

$$T \approx t(\mathrm{sign}(g_0 - g_c), \mathrm{sign}(g_1 - g_c) \ldots, \mathrm{sign}(g_{P-1} - g_c)) \tag{11}$$

4) Singular value decomposition (SVD) can be seen as a method for data reduction that exposes the substructure of the original data more clearly and orders it from the most to the least variation. It approximates a high dimensional data with a low dimensional data with minimum error. Accordingly, SVD is a powerful tool for data analysis, dimensionality reduction and data compression. In digital image processing, SVD compactly capture the essence of the semi-global features and geometric invariant properties of an image. The singular values are unique for a matrix and significant components of SVD (the largest singular values and their corresponding eigenvectors) contain most energy of each

image block which can be used as a steady representation of image blocks. This property is used for noise reduction and perceptual robustness improvement in our paper.

5.2 Pre-processing

In this work, RGB images are first converted to grayscale images using standard color space conversion. Since real images may have different size, to ensure that the final generated hash has a fixed length, all input images are rescaled into a standard resolution of $M \times N$ by bi-linear interpolation. Then, the resized input image is divided into non-overlapping blocks of $B \times B$ pixels and feature extraction is applied to each block. Size of blocks makes a trade-off between hash length, discriminative capability and perceptual robustness. A large block size means few features which will inevitably reduces discriminative capability. When the size of block is decreased, discrimination can be improved but perceptual robustness is easily affected by minor modification. In addition, a smaller block size will increase the hash size. In experiments, we find that B=32 is an acceptable moderate size for 256×256 images. In this scheme, before feature extraction, we first filter each block with an edge-preserving adaptive low-pass filter. The adaptive filter is more selective than a comparable linear filter, because of preserving edges and other high-frequency parts of an image. For this purpose, we use a pixel-wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. Our experiments have shown that this filtering has considerably improvements on robustness of image hash. The consecutive operations in the pre-processing step are drawn in Fig. 4.

5.3 Feature extraction

Feature extraction is represented in two aspects: CSLBP and SVD-CSLBP-based hashing algorithms. In CSLBP-based image hashing, features are directly extracted to represent the main content of the image compactly. While in the second method, SVD is applied to each sub-image as a pre-processing step to produce more robust features. Then, in hash generation step, the extracted features are converted into a real number sequence to form the final hash value. Details of these feature extraction methods are presented and compared in the following.

5.3.1 CSLBP-based hashing algorithm

Given a central pixel g_c and its P equally spaced circular neighborhood pixels $g_p, p = 0, 1, \ldots, P-1$, we can simply calculate gray-level differences between center-symmetric pairs of opposite pixels in a neighborhood as Eq. 12:

$$d_{p,q} = g_p - g_q,\ q = p + (P/2), p = 0, 1, \ldots, (P/2-1) \tag{12}$$

$d_{p,q}$ can be further decomposed into two components:

$$\begin{aligned} d_{p,q} &= s_{p,q} \times m_{p,q} \\ s_{p,q} &= \mathrm{sign}\left(g_p - g_q\right),\ m_{p,q} = \left|g_p - g_q\right| \end{aligned} \tag{13}$$

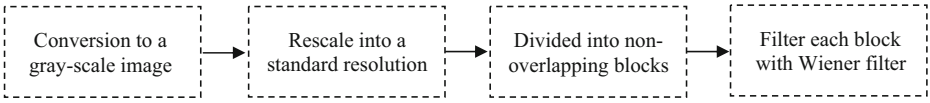


Fig. 4 Pre-processing steps in our algorithm

Where $s_{p,q}$ and $m_{p,q}$ are the sign and the magnitude of $d_{p,q}$, respectively. The original CSLBP operator disregards the magnitude information of the difference between the center-symmetric pairs of pixels and uses only the sign information (see Eq. 6). In other words, for a given $M \times N$ image, after identifying the CSLBP pattern of each pixel (i,j), the normalized histogram of CSLBP codes is computed over the image and it is used as a feature vector, Eq. (14):

$$H(b) = \left(\frac{1}{M \times N}\right) \sum_{i=1}^{M} \sum_{j=1}^{N} f\left(CSLBP_{P,R,T}(i,j), b\right), \quad b \in [0, B], \tag{14}$$

$$f(x,y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases}$$

where B is the maximal CSLBP pattern value.

In a gradient-based descriptor, image gradients are used to obtain magnitude and orientation for each image pixel. Since, definition of magnitude in CSLBP operator, $m_{p,q}$, is similar to image gradient, it contains useful information for image description [12]. In our proposed CSLBP-based image hashing, the motivation for the proposed feature descriptor is an efficient combination of CSLBP-based descriptor and gradient-based descriptor using sign and magnitude information of local differences. We extract two features for each pixel of the image by using the CSLBP operator. The sign feature is the same as the original CSLBP code defined by Eq. 6 and the magnitude feature vector corresponds to gradient information which is achieved by the magnitude of local differences.

The CSLBP operator has three parameters: radius R, number of neighboring pixels P, and threshold on the gray level difference T. In our experiments best results are achieved by R=1, P=8, and T=0.01. Using 8 neighboring samples, for each pixel (i,j), the sign feature (CSLBP code) can get 16 decimal numbers from 0 to 15, and the magnitude feature vector, MV, is defined by Eq. 15:

$$CSLBP_{8,1,0.01}(i,j) = \sum_{p=0}^{3} s\left(g_p - g_{p+4}\right) 2^p \tag{15}$$

$$MV(i,j) = \left[m_{0,4}, m_{1,5}, m_{2,6}, m_{3,7}\right]$$

Where $m_{p,q}$, $q=p+4$, p=0,1,2,3 is defined by Eqs. 12 and 13. Four histograms are built considering four components of magnitude vector. Each pixel of the image with a given CSLBP code is assigned to a bin in the histogram according to its magnitude. In other words, the magnitude feature, MV, is used as an adaptive weight in histogram calculation of CSLBP codes, Eq. 16.

$$H_p(b) = \sum_{i=1}^{B} \sum_{j=1}^{B} m_{p,q}(i,j) \times f\left(CSLBP(i,j), b\right), \tag{16}$$

$$b \in [0, 15], \; p = 0, 1, 2, 3 \text{ and } q = p + 4.$$

Finally, the obtained histograms are joined together to create the feature vector of an image block, FV, which is used for hash generation, Eq. 17. Each FV has 64 elements and to enhance the security of the scheme, all the elements in each FV are randomly scrambled according to a secret key $K1$.

$$FV = [H_0, H_1, H_2, H_3] \tag{17}$$

5.3.2 SVD-CSLBP-based hashing algorithm

LBP is noise sensitivity because noise affects the sign of local differences between neighboring pixels. To increase LBP robustness against noise, preprocessing steps are needed. In order to enhance the perceptual robustness of CSLBP-based hashing method, SVD transform is applied on each image block as an additional preprocessing step in our algorithm. In SVD-CSLBP, first SVD transform is performed on each block to extract the most fundamental features from it to reduce noise effect. Then, LBP hashing codes are obtained from the secondary block.

The block diagram of SVD-CSLBP is shown in Fig. 5. After preprocessing and image blocking, we reconstruct a secondary sub-image for each block derived from input image by SVD transform. From the secondary image block (which does not perceptually resemble the input), we further extract the CSLBP features which can be used as a hash value. Formation of the secondary image not only introduces further robustness against noise attacks, but also enhances the security properties.

SVD is based on a theorem from linear algebra which says that a rectangular matrix A can be broken down into the product of three matrices - an orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V:

$$A_{mn} = U_{mm} S_{mn} V_{nn}^{T} \tag{18}$$

where $U^T U=I$, $V^T V=I$; the columns of U and V are orthonormal eigenvectors of AA^T and $A^T A$, respectively, and S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order.

In SVD-CSLBP method, first, each block of the input image is further divided into four 16×16 non-overlapping (or possibly with overlap) sub-blocks. We find the SVD of each image sub-block.

$$SB_b^{(i)} = U_b^{(i)} S_b^{(i)} V_b^{T(i)}, \quad 1 \le b \le N,\ i = 1,2,3,4 \tag{19}$$

where $SB_b^{(i)}$ is the ith sub-block in bth input image block and N is the total number of blocks. Next, for each block, we collect the first singular vectors of U and V (i.e., the singular vectors that correspond to the largest singular value) of four sub-blocks, Eq.20:

$$\Gamma_b = \left[\overrightarrow{u}_b^{(1)}, \overrightarrow{u}_b^{(2)}, \overrightarrow{u}_b^{(3)}, \overrightarrow{u}_b^{(4)}, \overrightarrow{v}_b^{(1)}, \overrightarrow{v}_b^{(2)}, \overrightarrow{v}_b^{(3)}, \overrightarrow{v}_b^{(4)} \right], \quad 1 \le b \le N \tag{20}$$

where $\overrightarrow{u}_b^{(i)}$ and $\overrightarrow{v}_b^{(i)}$, $i = 1,2,3,4$ with size 16×1 are the first singular vectors in $U_b^{(i)}$ and $V_b^{T(i)}$, respectively. Then, we form a secondary image block, S_b, with size 16×8 by using a pseudo-random repetition of $\overrightarrow{u}_b^{(i)}$ and $\overrightarrow{v}_b^{(i)}$, $i = 1,2,3,4$ vectors, such that the elements of Γ_b form the columns of S_b in a pseudo-random order. Construction a secondary block for 8th block of a sample image is illustrated in Fig. 6.

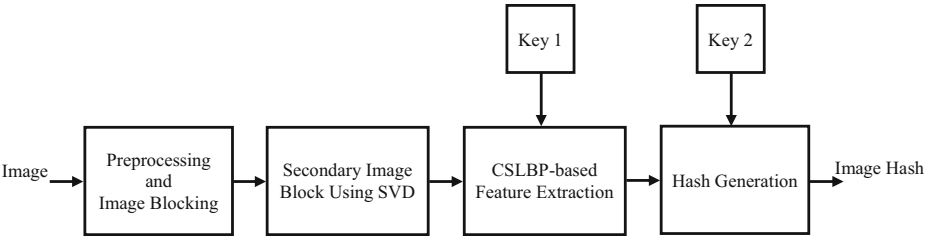


Fig. 5 Block diagram of SVD-CSLBP hashing algorithm

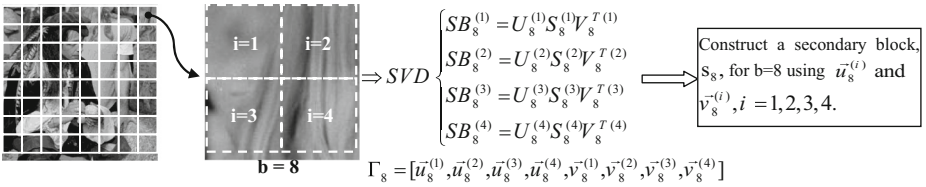


Fig. 6 Construction a secondary image block for b=8 using SVD transform

5.4 Hash generation

In the previous step, a feature vector was constructed for each non-overlapping block of input image. We generate pseudorandom weights $\omega=\{\alpha_i\}, i=1,\ldots,64$ from the normal distribution $N(u,\sigma^2)$ using a secret key, $K2$. ω is a random vector with 64 dimensions, with the same size of FV. Let $H=\{h_b\}, b=1,\ldots,N$ be the hash vector of input image where N is the total number of non-overlapping image blocks; we define FV_b as feature vector of bth block and its corresponding h_b component by Eq.21:

$$h_b = \langle FV_b, \omega \rangle \tag{21}$$

Where $\langle U, V \rangle$ indicates the inner product of two vectors U and V.

6 Experiments

In this section, the proposed algorithms are evaluated in two perceptual image hashing applications: image identification and image authentication [42]. Furthermore, some experiments are also included to analyze the performances of the proposed hashing schemes with respect to forged region detection and key-dependent security. Finally, the experiments are finished with a precise discussion on the desirable properties of perceptual image hashing.

6.1 Evaluation of image identification

One usage mode of perceptual image hashing is searching large databases for desired image content, (identification/recognition). Using perceptual hash functions for such applications has two advantages: first, only the hash values and the corresponding Meta data (e.g., file name) are needed to be stored in the database. This reduces the size of the database significantly. Second, if the image has been modified in a perceptually insignificant way, it still can be found in the database. Figure 7 illustrates this usage mode for perceptual image hashing.

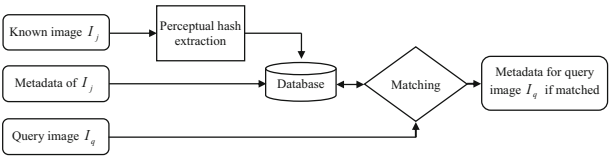


Fig. 7 Image identification framework

6.1.1 Experimental setup: database, dissimilarity measure

In this experiment, we collect 1000 color images from two databases: The Corel database [4] and Columbia photographic images and photorealistic computer graphics dataset [25]. The content of those images includes beach, buildings, flowers, human, animals, and so on. The images are in different size: 256×384, 384×256 and 722×480. All the color images are resized to 256×256 and converted as gray-scale images in the experiments. To validate perceptual robustness, many commonly content-preserving manipulations are considered. These distortions may be happened due to the noise in transmission channel, lossy compression, and geometric manipulations. For each original image, we generate 68 distorted versions by manipulating the original image according to a set of legitimate operations which are defined in Table 1. The manipulations consisted of additive noise, blurring, brightness changes, geometric attacks and JPEG compression.

Many similarity measurements such as Hamming distance, Euclidean distance and so on can be used to compare image hashing codes. Since hash values are approximately linearly changed, we exploit correlation coefficient to measure hash similarity. Given $H^{(1)}=(h_1^{(1)},h_2^{(1)},\ldots,h_L^{(1)})$ and $H^{(2)}=(h_1^{(2)},h_2^{(2)},\ldots,h_L^{(2)})$ as two image hash vectors with size L. The correlation coefficient is defined by Eq. 22:

$$S=\frac{\left\langle \left(H^{(1)}-\mu^{(1)}\right).\left(H^{(2)}-\mu^{(2)}\right)'\right\rangle}{\sqrt{\left\langle \left(H^{(1)}-\mu^{(1)}\right).\left(H^{(1)}-\mu^{(1)}\right)'\right\rangle}\times\sqrt{\left\langle \left(H^{(2)}-\mu^{(2)}\right).\left(H^{(2)}-\mu^{(2)}\right)'\right\rangle}}$$
$$\mu^{(1)}=\frac{1}{L}\sum_{i=1}^{L}h_i^{(1)},\ \mu^{(2)}=\frac{1}{L}\sum_{i=1}^{L}h_i^{(2)} \tag{22}$$

where $\langle X.\ Y\rangle$ denotes inner product between two vectors X and Y.

6.1.2 Performance of the proposed methods

To realize perceptual robustness and discriminative capability of the proposed hashing methods, an investigation on average and distribution of hash distances between similar and distinct image pairs are provided. There are 1000 original images, and each one has been

Table 1 Content-preserving manipulations with some details in parameters description and setting

Manipulation		Parameter description	Parameter setting	Copies
Additive noise	Gaussian noise	Mean (m), Variance (v)	m=0, $v\in(0.0005\sim0.005)$	10
	Speckle noise	Noise variance (N_v)	$N_v\in(0.001\sim0.01)$	10
Blurring	Gaussian blurring	Standard deviation (σ), window size (F_s)	F_s=3,$\sigma\in(0.5\sim5)$	10
	Motion blurring	Linear motion by (len) pixel, Angle (θ°)	$len\in(1,2,3)$, $\theta\in(0^\circ,45^\circ,90^\circ)$	9
Luminance changes	Gamma correction	Gamma (γ)	$\gamma\in(0.7\sim1.3)$	10
	Histogram equalization	Discrete gray levels (n)	$n\in(8,16,32,64)$	4
Geometric attacks	Small rotation	Rotation angle (α°)	$\alpha^\circ=1^\circ,2^\circ,3^\circ,4^\circ,5^\circ$	5
	Scaling	Scaling factor (s)	$s\in(0.5\sim1.5)$	5
JPEG Compression		Quality factor (Q)	$Q\in(10\sim50)$	5

affected by several types of content-preserving operations: additive noise, blurring, luminance changes, geometric attacks and JPEG compression (see Table 1). Each original image and its distorted versions are counted as a pair of similar image. All images (original and distorted versions) are represented by their hashing codes. Then, hash distance between each original image and its perceptually similar versions are computed for each type of distortions. We also prepare a set of $\binom{1000}{2} = 499500$ pairs of different images, to obtain the average and frequency distribution of hash distances in distinct image pairs. The distribution of hash distances for two proposed methods CSLBP- and SVD-CSLBP-based hashing, are shown in Fig. 8. Generally, it can be seen that the distance of similar image pairs are smaller than distances between distinct image pairs. The concentration of hash distances in distinct images is close to 1, while in similar images it is typically less than 0.4. Therefore, the similar versions can be easily separated from the distinct images by a suitable threshold. One main interest to LBP is mainly due to its gray-scale invariant and high robustness to local variations [27]. So, generating robust hashing codes against luminance changes is a great benefit of using LBP features in perceptual image hashing. Generally, the extracted hash can tolerate JPEG compression, blurring and luminance changes very well, since the distribution of hash distances between the original images and these distorted versions are all less than 0.3 in CSLBP and 0.2 in SVD-CSLBP. Notice that the results show SVD-CSLBP achieves better robustness than CSLBP-based hashing in almost all cases. It is in accordance with the usefulness of SVD as a method to steady representation of image and also noise reduction.

In an alternative representation of the results, Fig. 9 shows the average distances between the original images and distorted versions for different types of content-preserving manipulations which are listed in Table 1. The horizontal axis depicts the parameters of distortions. Notice that the new results are compatible with the distribution of hash distances in Fig. 8. From the results, it can be seen that correlation distance of our

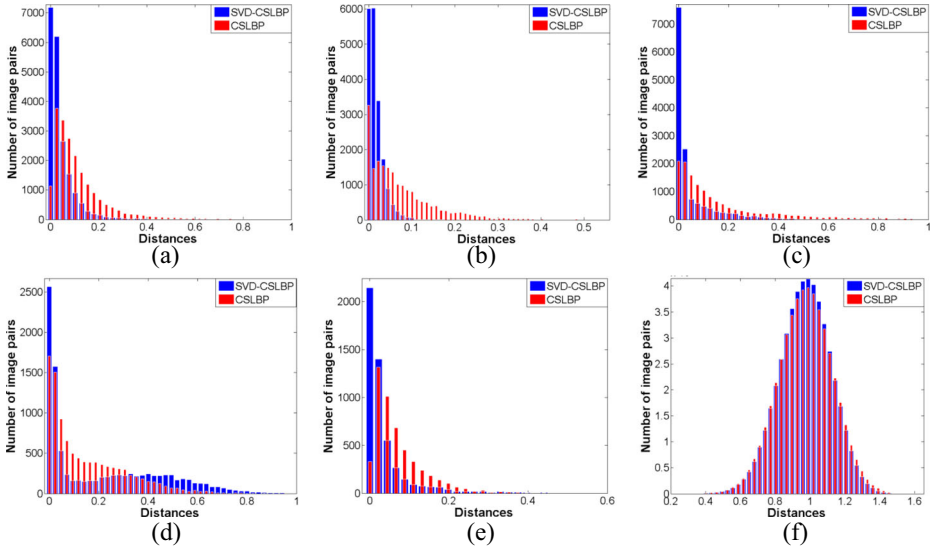


Fig. 8 Distribution of hash distances in CSLBP- and SVD-CSLBP-based hashing methods. (**a**)-(**e**) distribution of hash distances between original images and perceptual similar images in different content-preserving attacks: (**a**) additive noise, (**b**) blurring, (**c**) luminance changes, (**d**) geometric Attacks, (**e**) JPEG compression. (**f**) Distribution of hash distances between distinct images

schemes against nine kinds of content-preserving distortions are all below 0.2, except for the case of rotation and histogram equalization. Similar to distribution of hash distances, SVD-CSLBP achieves smaller average of hash distances than CSLBP-based hashing. The SVD-CSLBP combines the properties of SVD and LBP to improve the robustness of hashing against processing operations, but at the cost of sensitivity to rotation attacks. Generally, since rotation changes the location of image blocks, both of the proposed approaches are sensitive to rotation, (more details in Section 6.1.4). It is also worth noting that the average of hash distance between distinct images is close to 1, (see Fig. 8f). Therefore, the average of distance in distinct images is much larger by a noticeable margin in each parameter of content-preserving distortions. So, the proposed methods are very robust to legal image operations.

6.1.3 Performance comparisons

In image identification, the performance comparison with other methods is conducted in two aspects: identification accuracy and receiver operating characteristics (ROC) analysis. We compare our schemes with three reported hashing methods: ASCH [42],

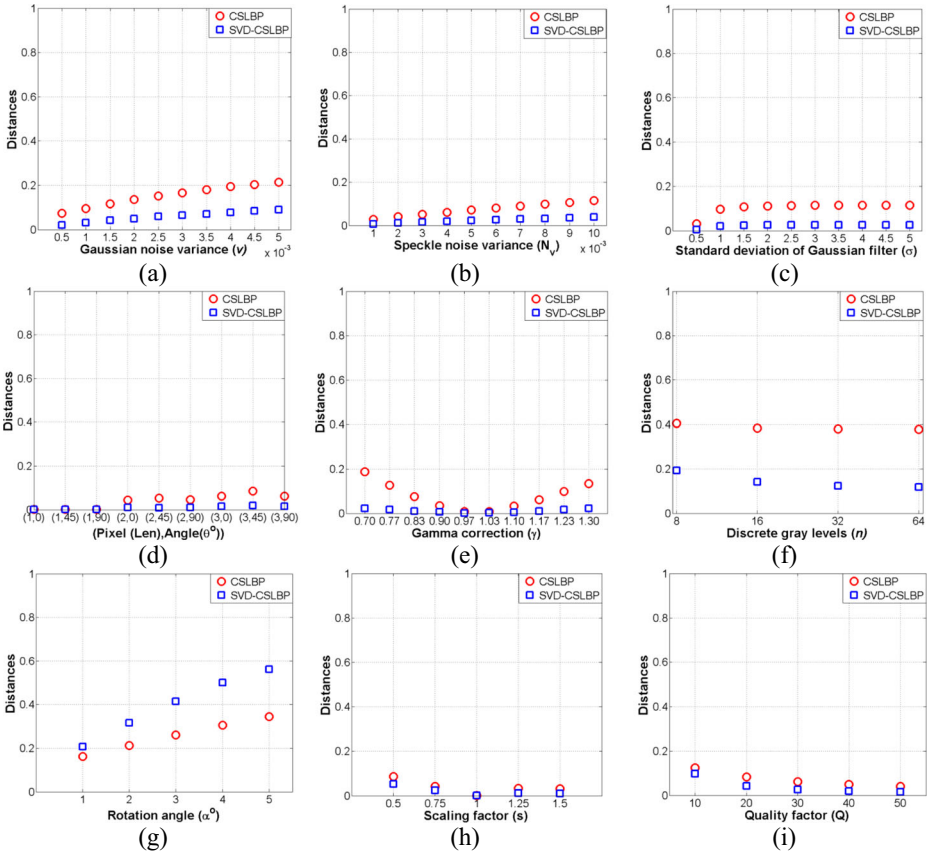


Fig. 9 Average distances between original images and their distorted versions in different content-preserving operations. (**a**) Gaussian noise, (**b**) speckle noise, (**c**) Gaussian blurring, (**d**) motion blurring, (**e**) gamma correction, (**f**) histogram equalization, (**g**) rotation, (**h**) scaling and (**i**) JPEG compression

RSCH [42] and DWT [38]. The selected methods for comparing belong to two different categories. The methods ASCH and RSCH are feature points-based hashing schemes while DWT is a transform-based image hashing. The setting parameters for DWT are: the number of random rectangles 150, wavelet decomposition to level "3" and wavelet basis "db4". The parameters were experimentally optimized for DWT algorithm. The default parameters of ASCH and RSCH are used based on the original paper. Similarity metrics in DWT and ASCH/RSCH are L2 norm and Euclidean distance, respectively.

(1) Identification accuracy

As shown in Fig. 7, an identification system compares the perceptual hash of an unidentified image to each of the hash values in the database. Assuming that the original dataset contains N images, $O=\{I_m\}_{m=1}^{N}$. The corresponding hash values in identification system are defined as $H(O)=\{H(I_m)\}_{m=1}^{N}$, where $H(I_m)=\{h_{1m},h_{2m},\ldots,h_{Lm}\}$ is a hash vector with length L for image I_m. Given a query image I_Q, we first generate its perceptual hash, $H(I_Q)$, then to find the best matching in the database, we utilize the nearest neighborhood classifier with the correlation distance as the dissimilarity measure between two image hash vectors. The identification accuracy is the percentage of the distorted images that are correctly classified. The aim of this experiment is to evaluate perceptual robustness of hashing methods in the presence of content-preserving distortions. Table 2 gives the results of identification accuracy for five methods in different distortions.

(2) Receiver Operating Characteristics Analysis

The perceptual robustness of various hashing schemes in image identification can be considered as a hypothesis testing problem. For each hash value stored in the database, when comparing a pair of hash values, two hypotheses H_0 and H_1 are defined as:

H_0: The pair of hash values corresponds to similar images.
H_1: The pair of hash values corresponds to different images.

The ROC curve characterizes the identification's performance by classifying the identified image into one of the hypothetical states. The ROC curves can be obtained by the probability of true identification P_T and probability of false alarm P_F as:

$$\begin{aligned} P_T &= \text{Probability}\{D(H(I_Q),H(I)) < \varepsilon | \text{H0 is true}\} \\ P_F &= \text{Probability}\{D(H(I_Q),H(I)) < \varepsilon | \text{H1 is true}\} \end{aligned} \tag{23}$$

Table 2 Percentages of identification accuracy under different attacks

Manipulation		RSCH	ASCH	DWT	CSLBP	SVD-CSLBP
Additive Noise	Gaussian noise	74.90	89.70	100	97.88	99.87
	Speckle noise	85.63	93.90	100	99.93	100
Blurring	Gaussian blurring	76.80	91.50	100	99.88	100
	Motion blurring	92.88	98.03	100	100	100
Luminance changes	Gamma correction	87.00	95.53	79.71	99.89	99.92
	Histogram equalization	11.42	28.80	28.65	74.20	99.15
Geometric Attacks	Rotation	90.91	96.87	100	94.60	80.02
	Scaling	78.12	91.92	100	99.94	100
JPEG Compression		86.40	94.46	100	99.98	99.72

where $D(H(I_Q),H(I))$ indicates the distance between hash values of query image and original image. Based on all the distances between manipulated images and original images, by choosing different values for threshold parameter, ε, different combinations of P_T and P_F can be derived. They correspond to the receiver operating characteristic (ROC) curves, as shown in Fig. 10.

6.1.4 Discussion on image identification results

Refer to content-preserving operations in Table 1, the identification accuracy of different hashing approaches are listed in Table 2. It is apparent from the table that the proposed hashing methods based on the LBP features can tolerate almost all kinds of content-preserving distortions. The results show that DWT, CSLBP and SVD-CSLBP hashing methods are superior to the feature points-based schemes (RSCH, ASCH) under additive noise, blurring and JPEG compression. The main reason is that detection of local feature points is sensitive to processing operations which degrade image quality. DWT and our hashing methods generate hashes by extracting robust features from predefined image patches, where the locations and

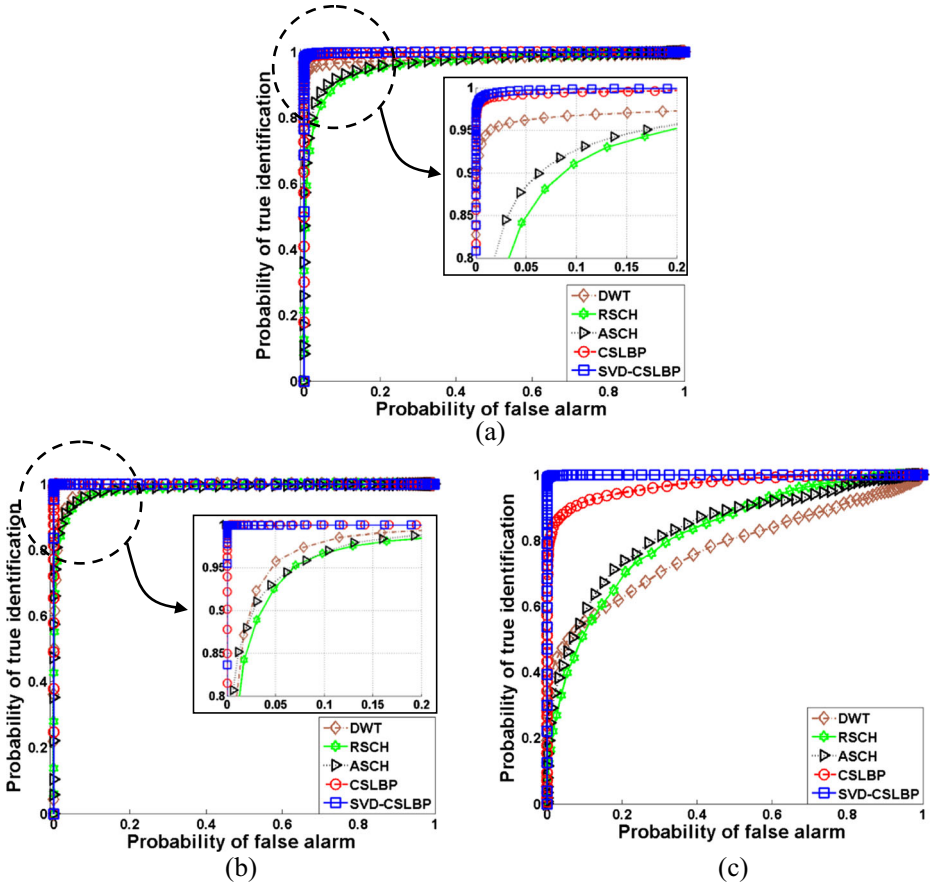


Fig. 10 Comparison of ROC curves between five hashing schemes in terms of various image processing operations. (**a**) Overall ROC, (**b**) ROC under Gamma correction and (**c**) ROC under histogram equalization

sizes of image patches are fixed and invulnerable to additive noise and blurring and etc. Based on the results, both DWT and our methods have almost the same identification accuracy up to 100 % in the mentioned distortions. However, it is also worth noting that the hash length of DWT is near 3 times of our methods.

Extraction of image features from the entire image only reflects the global information, and also there is no capability for local tampered region detection. A way of dealing with these problems is to divide the image into blocks and calculate features for each block separately. However, the intensity of pixels in blocks may be changed under some geometric attacks. One example of blocks in original image and in the rotated image is shown in Fig. 11. Obviously, the corresponding blocks in two figures cover different regions of the image and finally result in different block features. DWT and the proposed methods use block-based feature extraction; therefore, they are not designed to be robust against large-range rotation angles. Nonetheless, both DWT and our methods demonstrate acceptable robustness against small-angle rotation. Since the rotated image is generated by interpolation of pixels, the detected keypoints are not exactly the same as the original image. From the table, it can be seen that the local feature points-based hashing schemes (ASCH and RSCH) could not reach the 100 % identification accuracy in rotation attack. However, it should be noted, since detection of local feature points is invariant to geometric attacks, feature points-based hashing schemes have the advantage of robustness against large-range rotation angles.

For the scaling attacks, since we first resize all tested images to a default size (256×256), the image quality is degraded due to losing some details of the image during the down-sampling or up-sampling process. The effect of scaling is analogues to image blurring, so their results are too close together.

Finally, our proposed hashing algorithms show better accuracy than the other methods in luminance changes (Gamma correction and histogram equalization). Especially, a considerable improvement is achieved under histogram equalization. To explain these results in our methods, note that LBP is a robust image descriptor based on the joint distribution of signed gray level differences which are not affected by luminance changes. Hence, the underlying reason for good results in our hashing methods is due to employing gray-scale invariant LBP features. RSCH and ASCH demonstrate better performance than DWT in image Gamma correction. RSCH and ASCH algorithms are based on the SIFT key points and SIFT descriptors. SIFT descriptors are constructed from gradient histogram of image pixels within the neighborhood of detected keypoints. Since the essence of image gradient is relative difference of pixel values, variation of image luminance would have less effect on the gradient distribution. However, the two feature points-based schemes are still very vulnerable to histogram equalization. Although SIFT descriptors are invariant to illumination changes, severe attacks in brightness can significantly disturb SIFT key point detection process. In DWT hashing, brightness changes would inevitably introduce significant distortion in

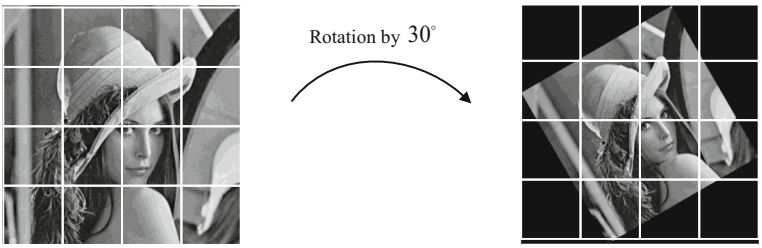


Fig. 11 The effect of rotation on image blocking

quantized statistics (averages or variances) of rectangles in wavelet image. Therefore, identification performance of DWT would degrade in the Gamma correction. The results in Table 2 support our statements.

Since the major improvements of the proposed hashing schemes lie in the robustness against luminance changes, receiver operating characteristic (ROC) curve is also used to better demonstration of the relationship between the probability of true detection (TPR) and the false positive rate (FFR) under these manipulations. The ROC curve can quantify the tradeoff of the hashing algorithm between robustness and discrimination. For comparison purpose, the ROC curves of all the comparative algorithms are plotted in the same figure, as shown in Fig. 10. The results are obtained using the same set of images in Table 1. As shown in the ROC curves, our proposed algorithm shows remarkable superiority over the other three algorithms in Gamma correction (Fig. 10b) and histogram equalization (Fig. 10c) due to use of gray-scale invariant LBP features. Furthermore, it can be seen from Fig. 10a that our methods obtain the best overall performance under all manipulations listed in Table 1.

6.2 Evaluation of image authentication

Image authentication experiment is designed to measure the sensitivity of our method to distinguish malicious attacks from content-preserving distortions. In image authentication, the hash of an original image, H_{org}, is available which is called the reference hash. The hash of a test image, H_{test}, is extracted using the same perceptual image hashing algorithm. Then, these two hashes are compared together. Now, the test image is declared to be authentic if $d(H_{test}, H_{org})<T$, where $d(.)$ is a distance measure and T is a predefined threshold. The problem of image authentication is considered as a hypothesis testing problem with two hypotheses: (H_0: *Image is authentic*) and (H_1: *Image is not authentic*). Each test image is classified into one of the hypothetical states. We use the receiver operating characteristics (ROC) curve to examine the discriminative capabilities of various hashing schemes for image authentication. True positive rate (TPR) and false positive rate (FPR) are two axes of ROC curve, which are defined by Eq. 24 and Eq.25, respectively.

$$TPR(T) = \frac{\text{Number of true images detected as authentic images}}{\text{Total number of authentic images}} \tag{24}$$

$$FPR(T) = \frac{\text{Number of forged images detected as authentic images}}{\text{Total number of forged images}} \tag{25}$$

Based on a given threshold (T), TPR gives us an estimate of the probability of true detection and FPR shows the percentage of images that are falsely classified as original image.

To implement authentication experiment, we construct three databases: an *original images database, a similar images database* and a *forged images database*. The original images database is exactly the same as the database used in identification experiment. 1000 similar images are obtained by apply a combination of various content-preserving operations on each original image. It consists of geometric and processing attacks as follows: Gamma correction (γ=1.1), JPEG compression (Q=70), Gaussian low-pass filtering (σ=0.5,F_s=3) and scaling (s=1.5). Our forged image database is constructed by splicing image forgery. Image splicing is a simple form of photomontage technology where is defined by simple combining image fragments from two or more different images without further post-processing. An example of original image and its similar and forged images are shown in Fig. 12. Forged image database

Fig. 12 An example in image authentication database. (**a**) original image, (**b**) perceptual similar image and (**c**) tampered image

contains 1000 tampered images which are created by splicing image forgery. In each forged image, the pasted area is 10 % of the host image.

Figure 13 compares the ROC curves of the proposed methods with previous efforts presented in [38], [23] and [42]. According to Eqs. 24 and 25, it is clear that TPR and FPR indicate robustness and discrimination, respectively. It can be observed from Fig. 13 that the proposed methods have shown stronger ability than the other four methods to distinguish content-preserving distortions from malicious attack. For example, with the same probability of false detection FPR=0.2, CSLBP and SVD-CSLBP achieve higher probability of correct detection (TPR=0.77 and TPR=0.97, respectively) than other hashing methods. In the same FPR, the TPR for DWT [38], RSCH [42], ASCH [42] and FP [23] hashing methods are 0.24, 0.46, 0.58 and 0.022 respectively.

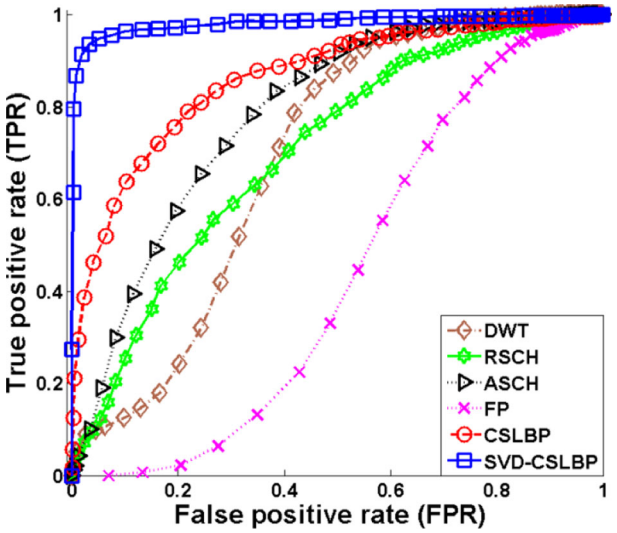


Fig. 13 ROC curve comparisons between our hashing and other methods in image authentication test

6.3 Tampering localization

A forger can easily modify local regions of an image and alter its original content by photo-editing software. In digital image forensics, many techniques have been proposed to address forged region detection problem [2, 34]. Block-based matching is one of the main methods for forged region localization. In this method, first, the image is divided into overlapping or non-overlapping blocks. Then, the perceptual hash function generates a hash value for each block. During forensic analysis, hashing codes are extracted from the corresponding blocks of the suspect image and a block-wise comparison reveals potential tampered regions. For tampering localization, size of blocks controls the trade-off between hash length and detection performance. Larger block size gives a smaller hash length but can introduce higher false detection than a smaller block size. An illustration of tampering localization functionality of the proposed methods is provided by the following experiment. We use a database of 100 image pairs, which includes the original image and tampered copy. Tampered images are generated by splicing technique where some regions of the original image are replaced with foreign blocks. We assess the accuracy of tampering localization by ROC analysis. In the results, the receiver operating curve is a plot of the probability of true positive rate versus the false positive rate as a function of system threshold . The two probabilities are defined as follows:

$$TPR(T) = \frac{\text{Number of tampered blocks detected as tampered}}{\text{Total number of tampered blocks}} \tag{26}$$

$$FPR(T) = \frac{\text{Number of genuine blocks detected as tampered}}{\text{Total number of genuine blocks}} \tag{27}$$

Where T is the variable threshold parameter.

The result of ROC analysis is presented in Fig. 14, in which the true positive rates and false positive rates are averaged for 100 image pairs. The figure compares the sensitivity of forged region detection in our methods and that in [31]. Roy et al. use quantized edge direction histogram features to localize tempered blocks [31]. We can see from Fig. 14 that the proposed methods have higher accuracy than [31]. Because our methods for hash generation combine LBP-based descriptors and gradient-based descriptors. It is worth noting that the DWT [38] and FP [23] are not applicable for tampering detection. In this sense, the proposed hashing scheme is more generally applicable.Some examples of forged region detection, using CSLBP-based hashing method, are also illustrated in Fig. 15. From left to right, columns 1 to 3 are original images, tampered images and detection results, respectively. As it can be seen, the algorithm locates the forged regions accurately even in the images with small tampered regions. It should be noted that in Fig. 15, three different kinds of malicious attacks are examined. In other words, from top to bottom, rows 1 to 3, show splicing image forgery, copy-move forgery and destroying image content, respectively.

6.4 Key-dependent security

Since an image hash serves as a secure tag of the image, in the proposed schemes two secret keys are used to prevent unauthorized user from hash generation. The first key is incorporated in the feature extraction step which is utilized to permute the elements of LBP-based feature

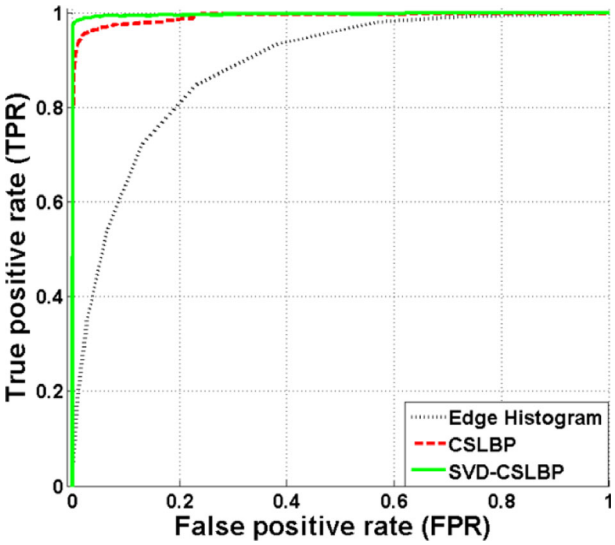


Fig. 14 The ROC curve for tampering localization

vector. Next, in hash generation step, the second key is employed to generate pseudorandom weights from the normal distribution $N(u,\sigma^2)$. In the key-dependent security, it is desirable that image hashes extracted from the same image with different secret keys be totally different. In our experiment, 10 test images are used to generate image hashes. Since the construction of CSLBP is similar to SVD-CSLBP, we take CSLBP as an example to prove the key-dependence security of our method. First, each image is coded by CSLBP-based hashing using two secret keys, *K1* and *K2*. Then, for each image, 1000 hashes are generated by 1000 wrong secret keys. In hash generation, all parameters are kept unchanged, except the keys. The axis of Fig. 16 are the index of 1000 wrong secret keys and correlation distances between the hash pairs of the 10 images obtained by the correct and wrong keys. Almost all of the correlation distances are in the vicinity of 1 and the minimum and maximum of the distances are 0.5 and 1.5, respectively. On the other hand, the average of correlation distances between distinct images is 1 and most similar images have average distances less than 0.2, (See Fig. 9). So, it would be extremely difficult for the attacker to generate or estimate the same hash value without knowing the correct keys. The key-dependent security of our hashing algorithm is completely confirmed by the empirical results.

6.5 Hash length and CPU running time

Hash length in the methods based on the block feature extraction is proportional to the number of image blocks. In the local feature point-based schemes, the number of key points determines the hash length. According to the parameters used in the experiment, the hash length of different methods is listed in Table 3. The hash length of our methods is 64 decimal digits. Among the compared methods, SCH and DWT have minimum and maximum hash length, respectively. Table 3 also compares the average time of producing each image hash in different methods. For implementation, we used a Sony Vaio laptop, Intel Core 2 Duo Processor P8800 (2.66 GHz), memory 4 GB and software of MATLAB 7.7.0. The processing time of hash generation is computed through the average time on 100 images of size 256×256. Since

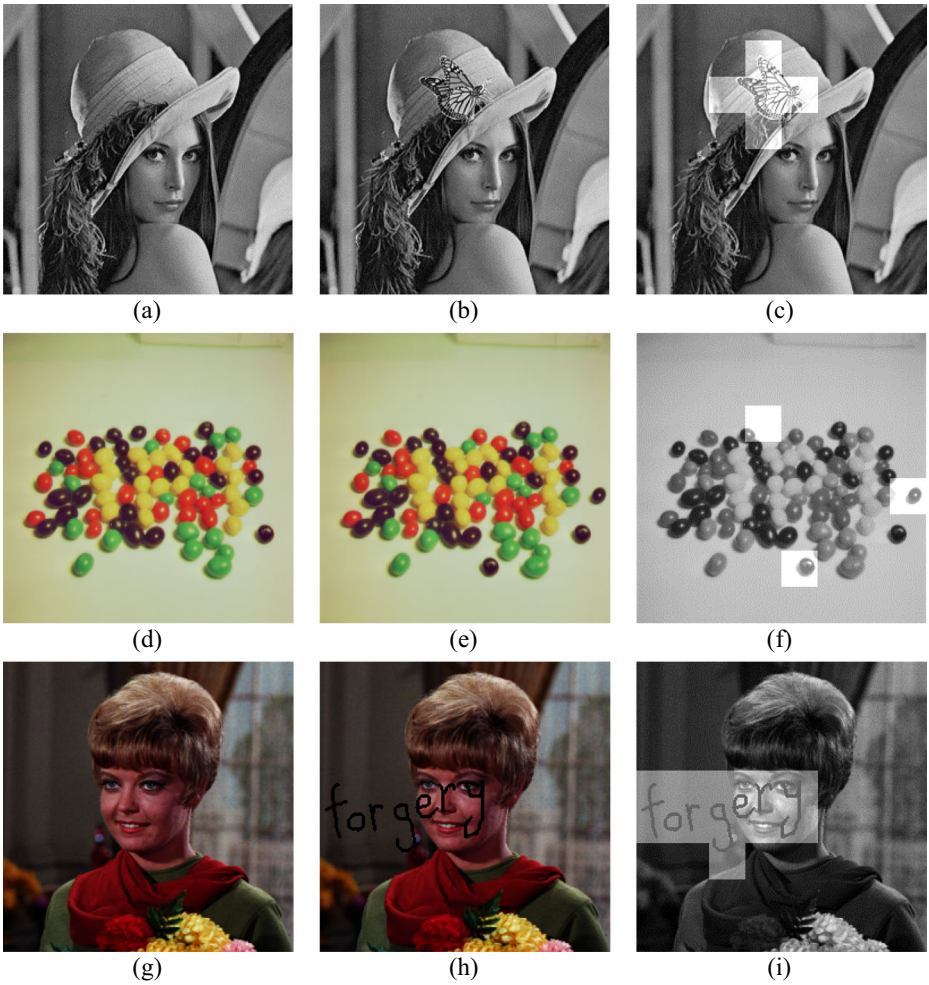


Fig. 15 Visualized forged detection results using CSLBP-based hashing method. From the left, columns 1 to 3 are original images, tampered images and detection results, respectively

computation time of the algorithms in [42] includes local key point detection, the average time in ASCH and RSCH is higher than the other methods. Fast hash extraction in our method can be attributed to the low computational complexity of LBP features. DWT has the lowest computation time.

6.6 Discussion on the desirable properties of perceptual image hashing

1) *Perceptual robustness* of the hash is known as the resilience against non intentional or perceptually insignificant modifications to the image. Experimental results in image identification demonstrated that our proposed schemes are robust to a wide range of distortions and attacks such as additive noise, blurring, brightness changes and JPEG compression.

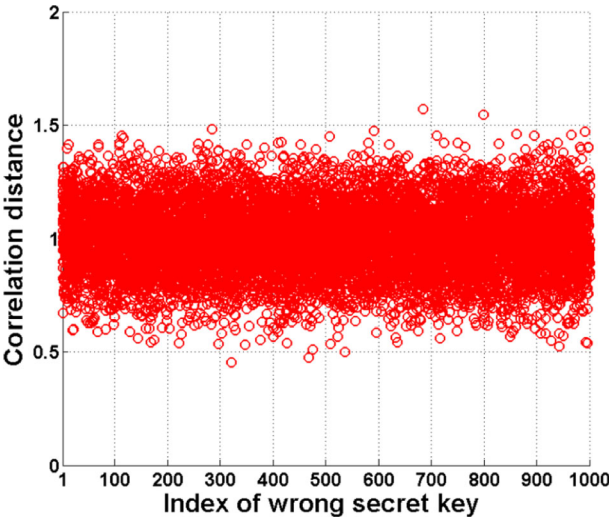


Fig. 16 Correlation distances between hashing pairs by the correct and wrong keys for ten images

2) *Uniqueness* refers to the fact that two visually distinct images have a very low probability to generate similar hashes. This property is also referred as: "*Anti-collision* [29, 37]" or "*fragility to visually distinct images* [22, 42]". In a perceptual image hashing system, collision happens if hash distance between two distinct images is less than a predefined threshold. To find the collision probability between different images, a set of 499500 pair of distinct images is used. Distribution of hash distances between each pair of different images is shown in Fig. 17. Using chi-square test, the distribution curve visually can be approximated with normal distribution where its mean and standard deviation are μ=0.97 and σ=0.15, respectively. Collision probability is the probability that the hash distance (*HD*) is smaller than *T* and can be calculated as follows:

$$\Pr(HD \le T) = \frac{1}{\sqrt{2\pi}\sigma}\int_0^T \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx = \frac{1}{2}\,erfc\left(-\frac{T-\mu}{\sqrt{2}\sigma}\right) \tag{28}$$

where *erfc*(·) is the complementary error function. The collision probabilities of the proposed method for different threshold values (*T*) are listed in Table 4. As Table shows, when *T* decreases, the collision probability is also decreases. However, too small thresholds impair the perceptual robustness of the hashing system in judgment of content-preserving operations.

Table 3 Time and hash length comparisons among different algorithms

Method	Hash length	Average time (sec)
DWT	150 decimal digits	0.05
SCH	20 decimal digits	3.55
CSLBP	64 decimal digits	0.1
SVD-CSLBP	64 decimal digits	0.53

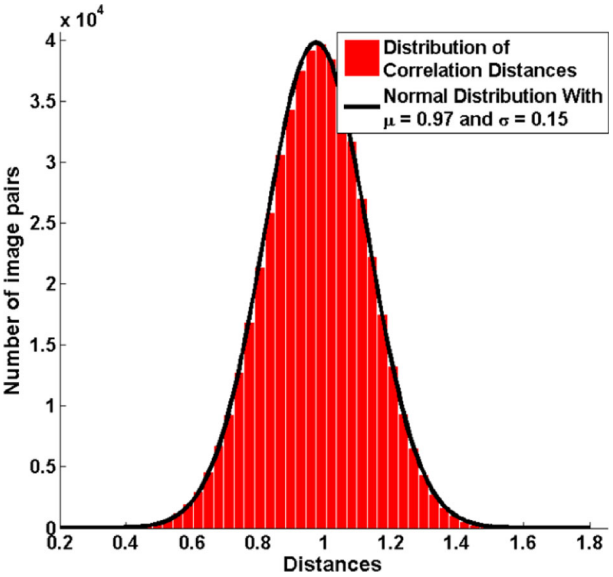


Fig. 17 Distribution of correlation distances between hashing pairs of different images

3) *Unpredictability* is a security measure of hashing systems which is directly related to the key-dependent randomization. Without knowing the secret keys, it is too difficult for an adversary to predict the hash values. Therefore, hash values must have high entropy over the key space. Since the construction of CSLBP is similar to SVD-CSLBP, we quantify the differential entropy of the CSLBP hash vector as a measure for *Unpredictability*, as proposed in [36].

From Section 5.4, let $X=\{x_i\}, i=1,2,\ldots,N$ be the CSLBP hash vector that is given by:

$$X = \{\langle FV_1, \omega\rangle, \ldots, \langle FV_N, \omega\rangle\}. \tag{29}$$

where $x_i=\langle FV_i,\omega\rangle, i=1,\ldots,N$ is a hash vector component of X. The differential entropy of a continuous random variable X with a probability density function (pdf) f is given by:

$$H(X) = \int_{\Omega} f(x)\log\frac{1}{f(x)}dx \tag{30}$$

where Ω defines the support area of $f(x)$. Since obtaining analytical model for the pdf of CSLBP hash vector component is not plausible, the probability space is

Table 4 Collision probabilities for different thresholds T

Threshold T	Collision probability
0.40	7.2348×10^{-5}
0.35	1.7877×10^{-5}
0.30	3.9724×10^{-6}
0.25	7.9333×10^{-7}
0.20	1.4233×10^{-7}
0.15	2.2929×10^{-8}

realized by the hash vector of a fixed image when the secret key is varied. In our experiment, the hash vectors of Lena image are created for 5000 different secret keys. The histogram of a typical component from the CSLBP hash vector is shown in Fig. 18a. From this figure it can be inferred that the density of the hash component is close to a Gaussian distribution. Similarly, based on the histograms of other components, we can see that the CSLBP hash vector approximately follows a random variable Gaussian distribution. Figure 18b shows the covariance matrix of the CSLBP hash vector that is approximately a diagonal matrix. It means that the proposed hash vector consists of statistically independent Gaussian components. It is well known from information theory, for a given variance, a random vector with Gaussian distribution has the maximum differential entropy. Therefore, from a security point of view, we can argue that the proposed CSLBP hashing is highly secure and unpredictable. The differential entropy of the CSLBP hash vector X is given by:

$$H(X) = \frac{1}{2}\log(2\pi e)^N |\mathrm{Cov}| \text{ bits}, \tag{31}$$

where $|\mathrm{Cov}|$ denotes the determinant of the covariance matrix of X, and N refers to length of the CSLBP hash vector.

7 Conclusion

In this paper we proposed a new perceptual image hashing method based on local binary patterns (LBP). In this algorithm a simple and efficient version of LBP feature, called center-symmetric LBP (CSLBP), was used for feature extraction. The original CSLBP descriptor uses only the sign information of local differences. In this paper, however,

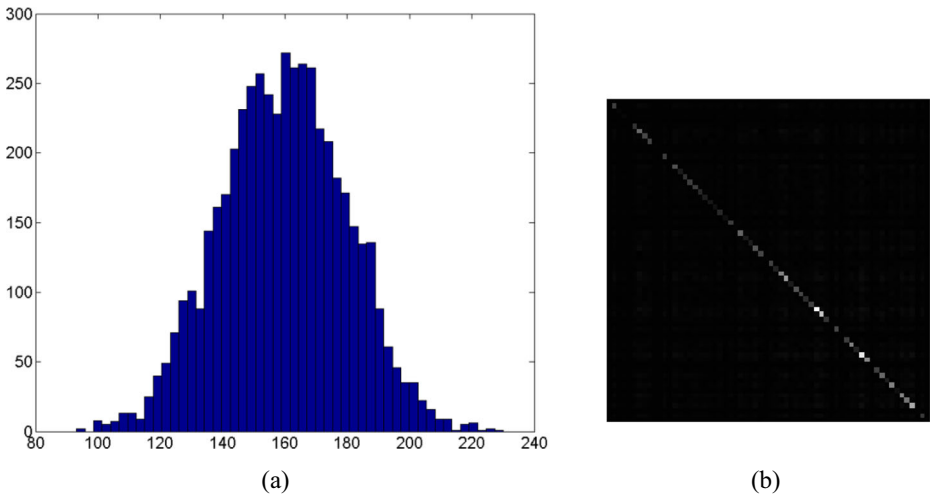


Fig. 18 (**a**) The histogram of a typical CSLBP hash vector component realized for image Lena from 5000 different secret keys. (**b**) The covariance matrix of the CSLBP hash vector for image Lena from 5000 different secret keys

both sign and magnitude information are utilized for image hashing to make benefit of gradient-based and LBP-based descriptor simultaneously. Since sensitivity to noise is a fundamental weakness of LBP features, SVD-CSLBP-based hashing method is introduced to tackle this problem.

To increase security of our hashing method, two secret keys were used in feature extraction and hash generation steps. To evaluate our proposed methods, comprehensive experiments were conducted for image identification and image authentication. The results demonstrated that our proposed schemes could tolerate almost all kinds of content-preserving distortions. The key strength of this study is high robustness against luminance changes (Gamma correction and histogram equalization). Finally, applicability in image tampering detection, acceptable hash length and running time can be also counted as the advantage of our proposed hashing method.

Future work will be focused on two subjects. First, to make the proposed method robust against geometrical distortions, Radon transform or Fourier-Mellin transform can be applied on the image. Presenting the image in Log-Polar domain is another solution for this purpose. Second, LBP descriptor that is robust to illumination variation and compression process can be used as preprocessing step in conjunction with well-established image hashing methods.

Acknowledgment The authors are grateful for the anonymous reviewers' insightful comments and valuable suggestions sincerely. We would like appreciate Dr. Xudong Lv, Dr. Vishal Monga and Dr. Divyanshu Vats for letting us to use their codes for comparing the results.

References

1. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Transa Pattern Anal Mach Intell 28:2037–2041
2. Birajdar GK, Mankar VH (2013) Digital image forgery detection using passive techniques: a survey. Digit Investig 10:226–245
3. Ching-Yung L, Shih-Fu C (2001) A robust image authentication method distinguishing JPEG compression from malicious manipulation. IEEE Trans Circuits Syst Video Technol 11:153–168
4. Corel (2001) test set.[Online]. http://wang.ist.psu.edu/~jwang/test1.tar. Accessed 10 Feb 2013
5. Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2008) Digital Watermarking and Steganography, Morgan Kaufmann Publishers Inc
6. Davarzani R, Mozaffari S, Yaghmaie K (2015) Scale- and rotation-invariant texture description with improved local binary pattern features. Signal Processing,111: 274-293
7. Davarzani R, Yaghmaie K, Mozaffari S, Tapak M (2013) Copy-move forgery detection using multiresolution local binary patterns. Forensic Sci Int 231:61–72
8. De Roover C, De Vleeschouwer C, Lefebvre F, Macq B (2005) Robust image hashing based on radial variance of pixels, in: Image Processing, 2005. ICIP 2005. IEEE International Conference on, pp. III-77-80
9. Farid H (2009) A survey of image forgery detection. IEEE Signal Process Mag 2:16–25
10. Fridrich J, Goljan M (2000) Robust hash functions for digital watermarking, in: Information Technology: Coding and Computing, 2000. Proceedings. International Conference on, pp. 178–183
11. Guo X, Hatzinakos D (2007) Content Based Image Hashing Via Wavelet and Radon Transform, in: H.S. Ip, O. Au, H. Leung, M.-T. Sun, W.-Y. Ma, S.-M. Hu (Eds.) Advances in Multimedia Information Processing – PCM 2007, Springer Berlin Heidelberg, pp. 755–764
12. Guo Z, Zhang D (2010) A completed modeling of local binary pattern operator for texture classification. IEEE Trans Image Proc 19:1657–1663
13. Guoying Z, Ahonen T, Matas J, Pietikainen M (2012) Rotation-invariant image and video description with local binary pattern features. IEEE Trans Image Proc 21:1465–1477
14. Haouzia A, Noumeir R (2008) Methods for image authentication: a survey. Multimed Tools Appl 39:1–46

15. Heikkila M, Pietikainen M (2006) A texture-based method for modeling the background and detecting moving objects. IEEE Trans Pattern Anal Mach Intell 28:657–662
16. Heikkilä M, Pietikäinen M, Schmid C (2009) Description of interest regions with local binary patterns. Pattern Recogn 42:425–436
17. Kozat SS, Venkatesan R, Mihcak MK (2004) Robust perceptual image hashing via matrix invariants, in: Image Processing, 2004. ICIP '04. 2004 International Conference on, pp. 3443–3446 Vol. 3445
18. Lefbvre F, Macq B, Legat J D, (2002) RASH: RAdon Soft Hash algorithm
19. Lei Y, Wang Y, Huang J (2011) Robust image hash in Radon transform domain for authentication. Signal Processing Image Commun 26:280–288
20. Liu W, Wang Y, Li S (2011) LBP feature extraction for facial expression recognition. J Inf Compu Sci 8:412–421
21. Monga V (2005) Perceptually based methods for robust image hashing (Ph.D. thesis), in: Electrical and Computer Engineering, Electrical and Computer Engineering, The University of Texas at Austin, Austin (Texas), pp. 120
22. Monga V, Mihcak MK (2007) Robust and secure image hashing via Non-negative matrix factorizations. IEEE Trans Inf Forensics and Secur 2:376–390
23. Monga V, Vats D, Evans B.L (2005) Image Authentication Under Geometric Attacks Via Structure Matching, in: Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, pp. 229–232
24. Monga V, x, x00E, M.K. ak, Robust and Secure Image Hashing via Non-Negative Matrix Factorizations, Information Forensics and Security, IEEE Transactions on, 2 (2007) 376–390.
25. Ng T T, Chang S F, Hsu Y F, Pepeljugoski M (2005) Columbia Photographic Images and Photorealistic Computer Graphics Dataset, in: ADVENT Technical Report, #203-2004-3, Columbia University
26. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29:51–59
27. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24:971–987
28. Pietikäinen M, Hadid A, Zhao G, Ahonen T (2011) Computer Vision Using Local Binary Patterns, in: Computer Vision Using Local Binary Patterns, Springer London, pp. E1-E2
29. Qin C, Chang C-C, Tsou P-L (2013) Robust image hashing using non-uniform sampling in discrete fourier domain. Digital Signal Process 23:578–585
30. Rivest R (1992) The MD5 Message-Digest Algorithm, RFC Editor
31. Roy S, Sun Q (2007) Robust Hash for Detecting and Localizing Image Tampering, in: Image Processing, 2007. ICIP 2007. IEEE International Conference on, pp. VI - 117-VI - 120.
32. Shan C, Gong S, McOwan PW (2009) Facial expression recognition based on local binary patterns: a comprehensive study. Image Vis Comput 27:803–816
33. Shuo-zhong W, Xin-peng Z (2007) Recent development of perceptual image hashing. J of Shanghai Univ 11: 323–331
34. Stamm MC, Wu M, Liu KJR (2013) Information forensics: an overview of the first decade. Access, IEEE 1: 167–200
35. Sun R, Zeng W (2014) Secure and robust image hashing via compressive sensing. Multimed Tools Appl 70: 1651–1665
36. Swaminathan A, Mao Y, Wu M (2006) Robust and secure image hashing. IEEE Trans Inf Forensics Secur 1: 215–230
37. Tang Z, Wang S, Zhang X, Wei W, Zhao Y (2011) Lexicographical framework for image hashing with implementation based on DCT and NMF. Multimed Tools Appl 52:325–345
38. Venkatesan R, Koon SM, Jakubowski MH, Moulin P (2000) Robust image hashing, in: Image Processing, 2000. Proceedings. 2000 International Conference on, pp. 664–666 vol.663.
39. Li W (2012) Perceptual Multimedia Hashing (Ph.D. thesis), in: Department of Electrical Engineering (ESAT), Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Heverlee (Belgium) pp. 208
40. Wu M, Mao Y, Swaminathan A (2007) A Signal Processing and Randomization Perspective of Robust and Secure Image Hashing, in: Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop on, pp. 166–170
41. Lv X, Wang ZJ (2008) Fast Johnson-Lindenstrauss Transform for robust and secure image hashing, in: Multimedia Signal Processing, 2008 I.E. 10th Workshop on, pp. 725–729
42. Lv X, Wang ZJ (2012) Perceptual image hashing based on shape contexts and local feature points. IEEE Trans Inf Forensics Secur 7:1081–1093
43. Li Z, Liu G, Yang Y, You J (2012) Scale- and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift. IEEE Trans Image Proc 21:2130–2140

Reza Davarzani received the B.Sc.Eng. degree in Electronic Engineering from Shahrood University of Technology, Shahrood, Iran, in 2006, the M.S. degree in Electrical Engineering from Semnan University, Semnan, Iran, in 2009. Now, he is pursuing toward Ph.D. degree at Semnan University. His research interests include image processing, digital image forensics, image watermarking and computer vision.

Saeed Mozaffari received his B.Sc., M.S and Ph.D. degrees in Electronic Engineering from Amirkabir University of Technology, Tehran, Iran. Since 2006, he is a faculty member in Electrical and Computer Department of Semnan University. His research interests include digital image processing, computer vision, and Pattern Recognition.

Khashayar Yaghmaie was born in 1957 in Semnan, Iran. He received his M.Sc. degree in Telecommunication from Tehran University, Iran, in 1985 and his Ph.D. in speech processing from university of Surrey in 1993, U.K. He works currently as an assistant professor and the head of Research department in Semnan University. His interests include voice and speech coding and image processing.