Robust Image Hashing with Tensor Decomposition

Zhenjun Tang, *Member, IEEE*, Lv Chen, Xianquan Zhang, and Shichao Zhang, *Senior Member, IEEE*

Abstract—This paper presents a new image hashing that is designed with tensor decomposition (TD), referred to as TD hashing, where image hash generation is viewed as deriving a compact representation from a tensor. Specifically, a stable three-order tensor is first constructed from the normalized image, so as to enhance the robustness of our TD hashing. A popular TD algorithm, called Tucker decomposition, is then exploited to decompose the three-order tensor into a core tensor and three orthogonal factor matrices. As the factor matrices can reflect intrinsic structure of original tensor, hash construction with the factor matrices makes a desirable discrimination of the TD hashing. To examine these claims, there are 14,551 images selected for our experiments. A receiver operating characteristics (ROC) graph is used to conduct theoretical analysis and the ROC comparisons illustrate that the TD hashing outperforms some state-of-the-art algorithms in classification performance between the robustness and discrimination.

Index Terms—Image hashing, tensor construction, tensor decomposition, Tucker decomposition

1 Introduction

Many image hashing algorithms have been proposed for retrieving similar images from large-scale image database. However, most of these algorithms do not reach satisfied performance due to their limitation in classification between robustness and discrimination. Therefore, this paper studies a new image hashing that is designed with tensor decomposition (TD), referred to TD hashing, where image hash generation is viewed as deriving a compact representation from a tensor. Tensor is a higher-order generalization of matrix, and has been widely used in multimedia. For example, [1] and [2] factorize a relational tensor between images and their meta-data to learn a new representation of images, and [3] decomposes context tensors to make image recommendation. Tensor methods are efficient tools for data analysis [4] and TD has been successfully applied in many applications, such as data mining, graph analysis, signal processing and computer vision, but its use in image hashing is rarely discussed. In this paper, we exploit TD to design a novel image hashing, referred to TD hashing, so as to reach a good classification performance between robustness and discrimination. Our main contributions are summarized as follows.

(1) Hash generation is viewed as deriving a compact representation from a tensor, and a stable three-order tensor is constructed for representing input image. Since influences of digital operation on the normalized image are mitigated, the three-order tensor constructed from the normalized image is stable and thus provides good robustness of our TD hashing.

(2) A popular TD algorithm called Tucker decomposition is exploited to decompose the three-order tensor into a core tensor and three orthogonal factor matrices, and the factor matrices are used to construct image hash. As factor matrices can reflect intrinsic structure of original tensor, desirable discrimination of our TD hashing is thus achieved.

Experiments with 14551 images (i.e., 13213 images for robustness validation and 1338 images for discrimination test) are conducted to validate performance of our TD image hashing. Receiver operating characteristics (ROC) curve comparisons illustrate that the TD image hashing is superior to some state-of-the-art algorithms in classification performance between robustness and discrimination.

The remainder of this paper is organized as follows. Section 2 reviews the related work and Section 3 describes the TD image hashing. Section 4 discusses experimental results of our TD image hashing and Section 5 presents performance comparisons with some state-of-the-art algorithms. Conclusions are finally given in Section 6.

2 Related Work

Image hashing [5], [6] is a technique for deriving a content-based compact representation called hash from input image. Generally, it should satisfy two basic properties [7], [8]: robustness and discrimination. Robustness means that visually identical images should have the same or very similar hashes no matter their digital representations are the same or not. In other words, image hashing should be robust against content-preserving operations, such as image compression and geometric transforms. Discrimination requires that

• *The authors are with the Guangxi Key Lab of Multi-Source Information Mining & Security, Department of Computer Science, Guangxi Normal University, Guilin 541004, China. E-mail: {tangzj230, zxq6622}@163.com, 403642208@qq.com, zhangsc@gxnu.edu.cn.*

Manuscript received 30 Sept. 2017; revised 7 Apr. 2018; accepted 10 May 2018. Date of publication 17 May 2018; date of current version 4 Feb. 2019. (Coressponding author: Shichao Zhang).
Recommended for acceptance by H. Lee.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2018.2837745

different images should have different hashes. This means that the distance between hashes of different images should be large enough. Certainly, image hashing could have other properties for specific applications. For example, it must be key-dependent and sensitive to visual content changes when it is applied to image authentication [9], [10].

An early work of image hashing is proposed by Schneider and Chang [11]. As image hashing plays an important role in many applications [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], such as image retrieval, image authentication, approximate nearest neighbor search, digital watermarking, image indexing, image quality assessment and multimedia forensics, it attracts much attention from multimedia community in the past several years. In terms of the used techniques, the existing image hashing algorithms can be roughly divided into four research directions as follows.

(1) *Image hashing based on histogram*. For example, Schneider and Chang [11] extracted histogram of block pixels to construct image hash. This method can resist JPEG compression, but is fragile to large-angle rotation. Xiang et al. [25] used invariance of histogram to form image hash. This scheme is robust to geometrical transforms including large-angle rotation. In another study, Choi and Park [26] presented a hierarchical strategy for improving robustness of the histogram-based hashing [25]. Tang et al. [27] proposed to extract multiple histograms from different rings of input image. In another work, Tang et al. [28] used histogram of color vector angles to generate image hash. Both hashing algorithms [25], [26] can resist image rotation with large angles, but their discriminations are not good enough. Recently, Vadlamudi et al. [29] divided input image into non-overlapping blocks, distributed block-based histogram bins into large containers, and calculated the ratio of pixel count between two neighboring containers. The local histogram based hashing is resilient to content-preserving operations, but its discrimination should be improved.

(2) *Image hashing based on orthogonal transform*. For example, Venkatesan et al [7] introduced discrete wavelet transform (DWT) to image hashing and used statistics of DWT coefficients to make image hash. This hashing is robust to JPEG compression and small-angle rotation, but is fragile to gamma correction and contrast adjustment. Monga and Evans [30] applied the end-stopped wavelet transform to detecting visually significant points for hash construction. This method has good robustness against JPEG compression and moderate rotation. Qin et al. [31] jointly used discrete Fourier transform (DFT) and non-uniform sampling to construct hash. This scheme can resist JPEG compression and noise contamination, but is not robust enough to large angle rotation. Fridrich and Goljan [32] found that the magnitude of a low-frequent discrete cosine transform (DCT) coefficient cannot be changed easily without causing visible changes to the image. Based on this observation, they proposed to calculate hash by projecting input image on some DC-free random smooth patterns. This method can withstand some commonly-used operations, except image rotation. In another work, Tang et al. [33] used dominant DCT coefficients to generate image hash. The method is robust against many digital operations, and shows good performance in image copy detection.

(3) *Image hashing based on projection transform*. The commonly-used techniques are radon transform (RT) and fan-beam transform. The early use of RT in image hashing is given by Lefebvre et al. [34]. They exploited medium points of RT projections to form image hash. This method has good performance against geometric transform, but its discrimination should be improved. Wu et al. [35] extracted the mean matrix of RT coefficients, and compressed it by DWT and DFT. This hashing can resist many robustness attacks, including print-scan attack. In another study, Lei et al. [36] extracted moment features in RT domain and exploited significant DFT coefficients of the moments to calculate hash. The scheme has good rotation robustness. To reduce computational cost, Tang et al. [37] exploited a variation of RT called fan-beam transform to generate hash. The fan-beam transform based hashing has better performances than the conventional RT based hashing [35] in classification and running speed.

(4) *Image hashing based on dimensionality reduction*. For example, Kozat et al. [38] exploited singular value decomposition (SVD) twice to calculate hash. The SVD-based hashing reaches good rotation robustness, but its discrimination is hurt. Motivated by [38], Monga and Mihcak [39] presented a similar image hashing by replacing SVD with non-negative matrix factorization (NMF). The NMF-based hashing shows better classification performance than the SVD-based hashing, but its secret key is not enough secure [40]. Tang et al. [41] proposed a lexicographical framework for hash generation and gave an implementation with DCT and NMF. Hash generation of the DCT-NMF hashing is dependent on dictionary and secret keys. Since the used dictionary cannot be exactly duplicated, hash security is thus achieved. Ghouti [42] calculated image hash by exploiting quaternion SVD (QSVD). The QSVD hashing has better classification performance than the SVD-based hashing, but its performance should be improved. Recently, Davarzani et al. [43] presented a robust image hashing based on SVD and center-symmetric local binary patterns (CSLBP). The SVD-CSLBP hashing can resist JPEG compression, blurring and brightness change, but its discrimination is not good enough. Tang et al. [44] proposed to calculate image hash with multidimensional scaling (MDS). The MDS hashing outperforms some popular hashing algorithms [27], [39] in classification performance.

Besides the above algorithms, other hashing techniques are also reported in recent years. For example, Li et al. [45] exploited random Gabor filtering (GF) and dithered lattice vector quantization (LVQ) to calculate hash. The GF-LVQ hashing is resilient to image rotation, but its discrimination is not desirable enough. Lv and Wang [46] presented a hashing algorithm based on scale invariant feature transform (SIFT) and Harris detector. Zhao et al. [47] used Zernike

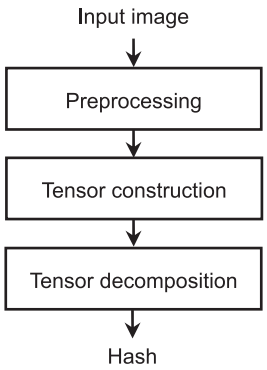

Fig. 1. Block diagram of our TD image hashing.

moments to calculate image hash. Tang et al. [48] designed a novel hashing by ring partition and invariant vector distance. Yan et al. [49] exploited adaptive and local feature extraction techniques to develop a multi-scale image hashing for tampering detection. Qin et al. [50] used block truncation coding to generate image hash. Huang et al. [51] presented a novel hashing method by random walk on zigzag blocking. The random walk based hashing has good security, but its classification performance should be improved. In [52], Qin et al. proposed to extract image hash by using dual-cross pattern encoding and salient structure detection. This hashing is secure, but its classification is also not desirable yet.

From the above review, it is found that most image hashing algorithms do not reach good classification performance with respect to robustness and discrimination. Therefore, more efforts are needed to design efficient hashing algorithms. Aiming at this issue, we exploit TD to develop a new image hashing, so as to reach good classification between robustness and discrimination.

3 Our TD Image Hashing

In this study, we view hash generation as deriving a compact representation from a tensor and develop a robust image hashing with three steps, i.e., preprocessing, tensor construction and tensor decomposition, as illustrated in Fig. 1. In the first step, input image is converted to a normalized image by a set of operations. And then, a three-order tensor is constructed from the normalized image. Finally, tensor decomposition is applied to the three-order tensor for generating a compact hash. These steps are detailed as follows.

3.1 Preprocessing

To mitigate the influences of content-preserving operations on the input image, a set of digital operations are applied, including interpolation, color space conversion and low-pass filtering. Firstly, bi-linear interpolation is used to resize the input image to a standard size $M \times M$, which makes our hashing robust against image rescaling. For RGB color image, the resized image is then converted into CIE $\mathrm{L^*a^*b^*}$ color space and the $\mathbf{L}^*$ component is used to represent the resized image. The reason why we select the $\mathbf{L}^*$ component is that CIE $\mathrm{L^*a^*b^*}$ color space is perceptually uniform and its $\mathbf{L}^*$ component closely matches human perception of lightness. Let L^* be lightness of a color pixel, a^* and b^* be its chromaticity coordinates, respectively. Thus, they can be determined by the following Eqs. (1), (2), (3) [53].

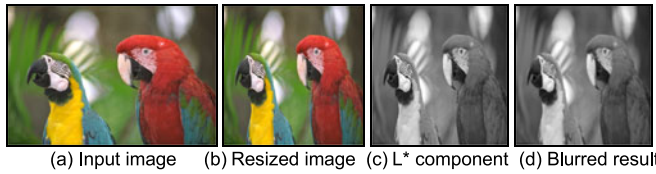

Fig. 2. An example of our preprocessing.

$$L^* = 116\, f(Y/Y_{\mathrm{w}}) - 16 \tag{1}$$

$$a^* = 500[f(X/X_{\mathrm{w}}) - f(Y/Y_{\mathrm{w}})] \tag{2}$$

$$b^* = 200[f(Y/Y_{\mathrm{w}}) - f(Z/Z_{\mathrm{w}})], \tag{3}$$

where X, Y and Z are the CIE XYZ tristimulus values, $X_{\mathrm{W}} = 0.950456$, $Y_{\mathrm{W}} = 1.0$ and $Z_{\mathrm{W}} = 1.088754$ are the CIE XYZ tristimulus values of the reference white point, and $f(t)$ is determined by Eq. (4) as follows.

$$f(t) = \begin{cases} t^{1/3}, & \text{If } t > 0.008856 \\ 7.787t + 16/116, & \text{Otherwise} \end{cases} \tag{4}$$

Moreover, the CIE XYZ tristimulus values X, Y and Z are calculated by Eq. (5) as follows [54].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix}, \tag{5}$$

in which R_1, G_1 and B_1 are the red, green and blue components of the color pixel. Finally, a Gaussian low-pass filtering is exploited to blur the $\mathbf{L}^*$ component. This is to alleviate influences of minor change on input image, such as noise and filtering. Generally, the low-pass filtering can be achieved by a convolution mask, whose element can be determined by

$$F(i,j) = \frac{F^{(1)}(i,j)}{\sum_i \sum_j F^{(1)}(i,j)}, \tag{6}$$

where $F^{(1)}(i,j)$ is defined as follows.

$$F^{(1)}(i,j) = e^{\frac{-(i^2+j^2)}{2\sigma^2}}, \tag{7}$$

where σ is the standard deviation of all elements in the mask. Fig. 2 illustrates an example of our preprocessing. Fig. 2a is the input image, Fig. 2b is the resized image, Fig. 2c is the $\mathbf{L}^*$ component and Fig. 2d is the blurred result.

3.2 Tensor Construction

In general, tensor is a multi-dimensional array. To derive a compact hash with tensor decomposition, we construct a three-order tensor from the normalized image. To do so, the $\mathbf{L}^*$ component is divided into non-overlapping blocks with a small size $S \times S$. For simplicity, let M be the integral multiple of S. Thus, there are both $D = M/S$ blocks along the x-axis and the y-axis directions, respectively. To make an initial compression, mean value of each block is calculated and then a feature matrix $\mathbf{U}$ is obtained as follows.

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,D} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,D} \\ \cdots & \cdots & \cdots & \cdots \\ u_{D,1} & u_{D,2} & \cdots & u_{D,D} \end{bmatrix}, \tag{8}$$

where $u_{i,j}$ is the mean value of the block in the ith position and the jth position along the y-axis and the x-axis directions $(1 \leq i \leq D, 1 \leq j \leq D)$, respectively. Besides initial compression, this operation also makes our algorithm robust against rotation with small angle, which can be understood as follows. Rotation with small angle will change pixel positions, but the block mean will not be significantly altered since pixels in a local small region have similar values. Clearly, a big block size is helpful to robustness against large-angle rotation. However, a bigger block size means fewer features in $\mathbf{U}$, which will inevitably hurt discrimination. In experiments, we select 2×2 as block size, which can reach a good balance between robustness and discrimination.

Next, the feature matrix $\mathbf{U}$ is further divided into non-overlapped blocks sized $Q \times Q$. For simplicity, let D be the integral multiple of Q. Thus, there are $n = (D/Q)^2$ blocks in total. To make a secure hash, we randomly select another n blocks sized $Q \times Q$ under the control of a secret key k_1. During selection, overlapping region between random blocks can exist. But the same block is not selected again. As a result, there are $L = 2n$ different blocks selected. To enhance security of our algorithm, we use another secret key k_2 to randomly stack these L blocks. Finally, a three-order tensor $\mathbf{X}$ sized $Q \times Q \times L$ is obtained.

Note that our security is ensured by random block selection and random block stack. Clearly, there are many possibilities of selecting n blocks. Without knowledge of the secret key, it is difficult to correctly guess the random block pattern. In addition, there are $L!$ permutations of random block stack. For example, if $L = 32$, there are 2.63×10^{35} possible permutations. As L increases to 64, the number of permutations of random block stack will reach about 1.27×10^{89}. A huge number of block permutations means a secure tensor. In practice, it is almost impossible to simultaneously and correctly guess random block pattern and random block stack. Therefore, our image hash derived from the tensor is secure. Key dependence of our TD hashing will be verified in Section 4.3.

3.3 Tensor Decomposition

Tensor decomposition (TD) [55], [56] is an efficient technique for data mining, graph analysis, signal processing, computer vision, and so on. In this study, we exploit TD to derive image hash. Here, the well-known algorithm called Tucker decomposition is selected to achieve TD. The classical Tucker decomposition is introduced by Tucker [57], and has been successfully applied in many real applications, such as face image recognition [58], image quality assessment [59], noise reduction [60] and data analysis [61].

For a three-order tensor $\mathbf{X} \in \mathbb{R}^{Q \times Q \times L}$, Tucker decomposition will decompose it into a core tensor $\mathbf{G} \in \mathbb{R}^{I \times J \times K}$ and three orthogonal factor matrices, i.e., $\mathbf{A} \in \mathbb{R}^{Q \times I}$, $\mathbf{B} \in \mathbb{R}^{Q \times J}$ and $\mathbf{C} \in \mathbb{R}^{L \times K}$. Mathematically, Tucker decomposition can be formulated as follows.

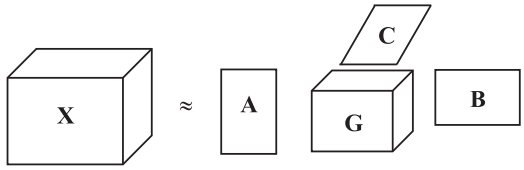

Fig. 3. Schematic diagram of Tucker decomposition.

$$\mathbf{X} \approx [\![\mathbf{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} g_{i,j,k} \left(\mathbf{a}_i \circ \mathbf{b}_j \circ \mathbf{c}_k\right) \tag{9}$$

where $\mathbf{a}_i, \mathbf{b}_j$ and $\mathbf{c}_k$ are the column vectors of $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, $g_{i,j,k}$ is the element of $\mathbf{G}$, the symbol '$\circ$' represents outer product between two vectors, and the symbol '$[\![\ ]\!]$' is a concise representation of Tucker decomposition given in [55]. Elementwise, the Eq. (9) can be rewritten as:

$$x_{w,h,r} \approx \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} g_{i,j,k} a_{w,i} b_{h,j} c_{r,k}, \tag{10}$$

in which $x_{w,h,r}, a_{w,i}, b_{h,j}$ and $c_{r,k}$ are the elements of $\mathbf{X}, \mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively. Calculation of Tucker decomposition is equivalent to solving an optimization problem [62] as follows.

$$[\![\mathbf{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \arg \min_{g_{i,j,k}, \mathbf{a}_i \mathbf{b}_j \mathbf{c}_k} \left\| \mathbf{X} - \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} g_{i,j,k} \left(\mathbf{a}_i \circ \mathbf{b}_j \circ \mathbf{c}_k\right) \right\|_2 \tag{11}$$

where $||\ ||_2$ is the Frobenius norm. In general, this optimization problem can be solved by the well-known method called alternating least squares (ALS) algorithm [61], [62]. Details of ALS algorithm can be referred to [55], [61], [62]. Fig. 3 presents a schematic diagram of Tucker decomposition.

Since the orthogonal factor matrices of Tucker decomposition can reflect intrinsic structure of the original tensor, we exploit them to construct image hash. For the factor matrix $\mathbf{A}$, we calculate the mean of each row and then obtain a feature vector as follows.

$$\mathbf{p}^{(\mathrm{A})} = \left[p_1^{(\mathrm{A})},\ p_2^{(\mathrm{A})},\ \ldots,\ p_Q^{(\mathrm{A})}\right]^{\mathrm{T}}, \tag{12}$$

in which $p_i^{(\mathrm{A})}$ is the mean of the ith row of $\mathbf{A}\,(1 \leq i \leq Q)$. Next, $\mathbf{p}^{(\mathrm{A})}$ is converted to a binary sequence as below.

$$h_i^{(\mathrm{A})} = \begin{cases} 0, & \text{if } p_i^{(\mathrm{A})} < m^{(\mathrm{A})} \\ 1, & \text{Otherwise,} \end{cases} \tag{13}$$

where $m^{(\mathrm{A})}$ is the mean of all elements of $\mathbf{p}^{(\mathrm{A})}$. Similarly, the feature vector of $\mathbf{B}$ is extracted as follows.

$$\mathbf{p}^{(\mathrm{B})} = \left[p_1^{(\mathrm{B})}, p_2^{(\mathrm{B})}, \ldots, p_Q^{(\mathrm{B})}\right]^{\mathrm{T}}, \tag{14}$$

in which $p_i^{(\mathrm{B})}$ is the mean of the ith row of $\mathbf{B}\,(1 \leq i \leq Q)$. Furthermore, $\mathbf{p}^{(\mathrm{B})}$ is quantized by the below rule.

$$h_i^{(\mathrm{B})} = \begin{cases} 0, & \text{if } p_i^{(\mathrm{B})} < m^{(\mathrm{B})} \\ 1, & \text{Otherwise,} \end{cases} \tag{15}$$

where $m^{(\mathrm{B})}$ is the mean of all elements of $\mathbf{p}^{(\mathrm{B})}$. Likewise, the feature vector of $\mathbf{C}$ is computed by the following equation.

(a) Kodak image dataset

(b) INRIA Copydays dataset

Fig. 4. Typical images of the used image datasets.

$$\mathbf{p}^{(\mathrm{C})} = \left[p_1^{(\mathrm{C})},\ p_2^{(\mathrm{C})},\ \ldots,\ p_L^{(\mathrm{C})}\right]^{\mathrm{T}}, \tag{16}$$

in which $p_i^{(\mathrm{C})}$ is the mean of the ith row of $\mathbf{C}\,(1 \leq i \leq L)$. And then, $\mathbf{p}^{(\mathrm{C})}$ is mapped to a bit sequence.

$$h_i^{(\mathrm{C})} = \begin{cases} 0, & \text{if } p_i^{(\mathrm{C})} < m^{(\mathrm{C})} \\ 1, & \text{Otherwise,} \end{cases} \tag{17}$$

where $m^{(\mathrm{C})}$ is the mean of all elements of $\mathbf{p}^{(\mathrm{C})}$. Finally, our image hash $\mathbf{h}$ is available by concatenating these binary elements as follows.

$$\mathbf{h} = \left[h_1^{(\mathrm{A})}, h_2^{(\mathrm{A})}, \ldots, h_Q^{(\mathrm{A})}, h_1^{(\mathrm{B})}, h_2^{(\mathrm{B})}, \ldots, h_Q^{(\mathrm{B})}, h_1^{(\mathrm{C})}, h_2^{(\mathrm{C})}, \ldots, h_L^{(\mathrm{C})}\right]. \tag{18}$$

Therefore, our hash length is $N = 2Q + L$ bits.

3.4 Hash Similarity Evaluation

As our image hash is a binary representation, we select the well-known distance metric called Hamming distance to measure similarity of two image hashes. Let $\mathbf{h_1}$ and $\mathbf{h_2}$ be two image hashes. Thus, their Hamming distance is defined as:

$$d_{\mathrm{H}}(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{N} |h_1(i) - h_2(i)|, \tag{19}$$

where $h_1(i)$ and $h_2(i)$ are the ith elements of $\mathbf{h_1}$ and $\mathbf{h_2}$, respectively. In general, a smaller Hamming distance means more similar images of the corresponding hashes. If the Hamming distance is not bigger than a given threshold, the images of the input hashes are judged as visually identical images. Otherwise, they are classified as distinct images.

TABLE 1
Digital Operations and Their Parameter Settings

Operation	Parameter	Parameter value	Number
Brightness adjustment	Photoshop's scale	$\pm 10, \pm 20$	4
Contrast adjustment	Photoshop's scale	$\pm 10, \pm 20$	4
Gamma correction	γ	$0.7, 0.9, 1.1, 1.2$	4
3×3 Gaussian low-pass filtering	Standard deviation	$0.3, 0.4, \ldots, 1.0$	8
Speckle noise	Variance	$0.001, 0.002, \ldots, 0.01$	10
Salt and pepper noise	Density	$0.001, 0.002, \ldots, 0.01$	10
JPEG compression	Quality factor	$30, 40, \ldots, 100$	8
Watermark embedding	Strength	$10, 20, \ldots, 100$	10
Image scaling	Ratio	$0.5, 0.75, 0.9, 1.1, 1.5, 2.0$	6
Rotation, cropping and rescaling	Agree in degree	$0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2$	8
Total			72

4 Experimental Results

In the experiments, the selected parameter values of our TD hashing are as follows. The input image is resized to a standard size 256×256; a 3×3 Gaussian low-pass filter with zero mean and a unit standard deviation is taken; the $\mathbf{L}^*$ component is divided into 2×2 non-overlapping blocks and thus the size of the matrix $\mathbf{U}$ is 128×128; the matrix $\mathbf{U}$ is further divided into non-overlapped blocks of size 32×32 and the number of random blocks is 16. The selected values of the parameters I, J and K of Tucker decomposition are all 1. In other words, our used parameters are: $M = 256$, $D = 128$, $S = 2$, $Q = 32$, $n = 16$, $I = 1$, $J = 1$ and $K = 1$. Consequently, the number of blocks used for tensor construct is $L = 2n = 32$. Therefore, the length of our hash is $N = 2Q + L = 96$ bits. In the following sections, the experiments of robustness, discrimination, key dependence and effect of the parameters of Tucker decomposition on hash performances are discussed.

4.1 Robustness

A database with 13213 images is constructed to evaluate robustness of our image hashing. To do so, two popular open image datasets called Kodak lossless true color image suite [63] and INRIA Copydays dataset [64] are both selected. The Kodak image dataset is distributed by the Eastman Kodak company for free use. It consists of 24 true color images whose sizes are either 768×512 or 512×768. The INRIA Copydays dataset includes 157 color images, whose sizes range from 1200×1600 to 3008×2000. Consequently, there are $24 + 157 = 181$ original color images. These color images contain various contents, such as buildings, sports, human beings, animal and landscape. Fig. 4 illustrates some typical images of the used image datasets. To generate visually similar images of these color images, ten commonly-used content-preserving operations provided by StirMark [65] and Photoshop, MATLAB are exploited to conduct robustness attacks. For each content-preserving operation, different parameter values are selected. Table 1 presents the used

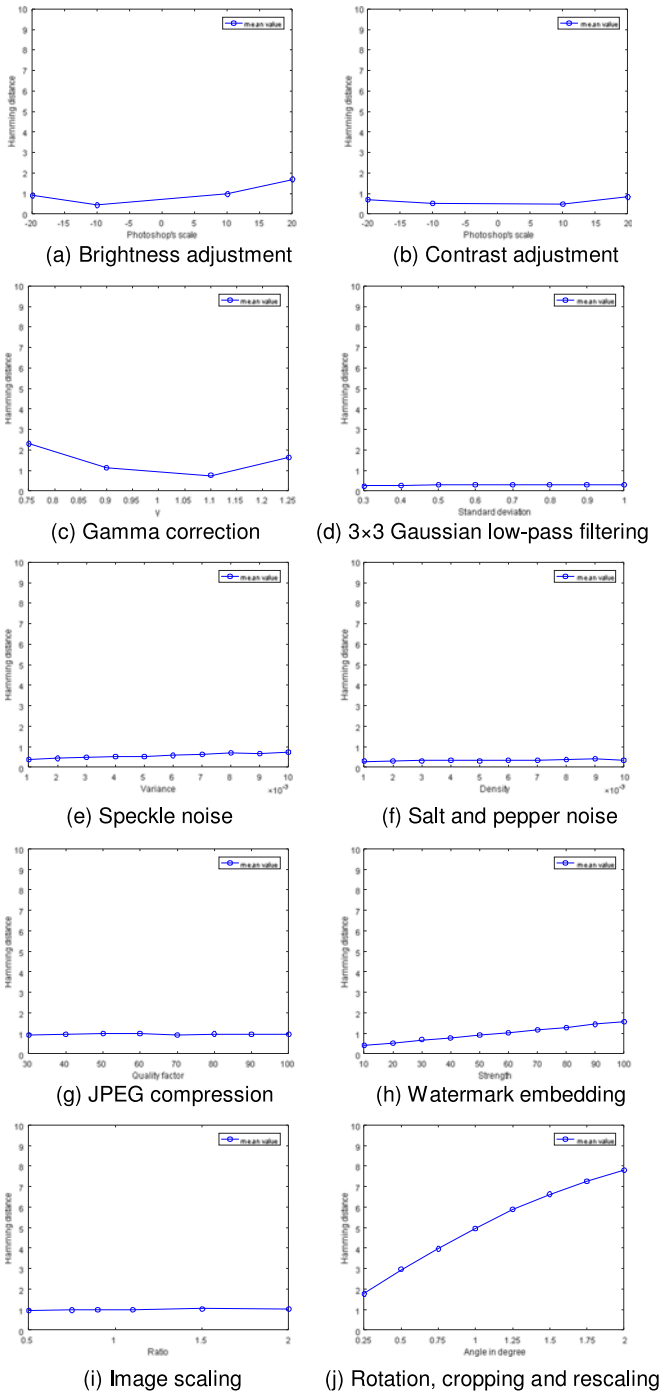

Fig. 5. Robustness test based on the Kodak image dataset and INRIA Copydays dataset.

content-preserving operations and their detailed parameter settings. Clearly, there are 72 manipulations in total. This means that every original image has 72 visually similar versions and the total pairs of visually similar images is $181 \times 72 = 13032$. Therefore, the number of the used images in the robustness test is $181 + 13032 = 13213$.

Image hashes of each pair of visually similar images are extracted and their similarity is evaluated with Hamming distance. Fig. 5 presents the mean Hamming distances of image hashes under each parameter of different content-preserving operations, where the y-axis is the Hamming distance and the x-axis is the parameter value of digital operation. It is

TABLE 2
Statistics of Hamming Distances Based on Kodak Image Dataset and INRIA Copydays Dataset

Operation	Max	Min	Mean	Standard deviation
Brightness adjustment	8	0	1.00	0.55
Contrast adjustment	7	0	0.64	0.18
Gamma correction	14	0	1.45	0.52
3×3 Gaussian low-pass filtering	4	0	0.29	0.01
Speckle noise	8	0	0.56	0.13
Salt and pepper noise	4	0	0.34	0.03
JPEG compression	8	0	0.95	0.02
Watermark embedding	13	0	0.98	0.40
Image scaling	8	0	0.99	0.01
Rotation, cropping and rescaling	26	0	5.15	1.03

observed that the mean Hamming distances are all smaller than 5, except some distances of rotation, cropping and rescaling. The mean distances of rotation, cropping and rescaling are bigger than those of other operations. This is because the operations of rotation, cropping and rescaling is a combinational attack, which will introduce more distortions on images than a single operation. Nevertheless, the mean distances of the combinational attack under different parameter settings are much smaller than 10.

To view the detailed robustness performance of different operations, Table 2 illustrates their statistics of Hamming distances. It can be seen that all means are smaller than 6 and their standard deviations are small. Moreover, the minimum values of all digital operations are 0, and the maximum values of all operations are smaller than 15, except the combinational attack of rotation, cropping and rescaling. For the combinational attack, its maximum Hamming distance is 26. Since the mean distances of all operations are small, we can select 10 as the threshold to resist most digital operations. In this case, 98.64 percent similar images are correctly classified as visually identical images. If no rotated images are included in the application, the percentage of correct detection will reach 99.92 percent.

4.2 Discrimination

To test discriminative capability, another open image database called UCID [66] is taken. The UCID contains 1338 different true color images and the image sizes are either 512×384 or 384×512. Fig. 6 shows some typical images of UCID. In

Fig. 6. Some typical images of UCID.

TABLE 3
The Estimated Parameters and χ^2 Results of the Test Distributions

Distribution	Estimated parameter	χ^2
Normal	$\mu = 42.85, \sigma = 14.87$	9890.92
Lognormal	$\mu = 3.69, \sigma = 0.40$	1.86×10^{14}
Rayleigh	$\beta = 32.07$	1.83×10^{5}
Poisson	$\lambda = 42.85$	7.88×10^{12}
Weibull	$a = 47.89, b = 3.17$	1928.10
Gamma	$a = 7.24, b = 5.92$	81792.22

the experiment, image hashes of these 1338 color images are firstly extracted. For each image, the Hamming distances between its hash and the hashes of other 1337 images are then calculated. Consequently, $C_{1338}^2 = 1338 \times 1337/2 = 894453$ Hamming distances are generated. It is observed that the minimum and the maximum distances are 1 and 94, respectively. Moreover, the mean and standard deviation of these distances are 42.84 and 14.88, respectively.

To make theoretical analysis of our discrimination, the well-known hypothesis testing method called chi-square test [67] is exploited. To do so, the maximum likelihood estimation is firstly taken to estimate parameters of the evaluated distributions, i.e., normal distribution, lognormal distribution, Rayleigh distribution, Poisson distribution, Weibull distribution, and Gamma distribution. The statistics χ^2 is then calculated by Eq. (20) as follows.

$$\chi^2 = \sum_{i=0}^{N} \frac{(n_i - n_{\text{total}} P_i)^2}{n_{\text{total}} P_i}, \tag{20}$$

where n_i is the frequency of Hamming distance equaling i, n_{total} is the number of trials, N is the hash length, and P_i is the probability at i determined by the probability density function of the evaluated distribution. Table 3 presents the estimated parameters and χ^2 results of the test distributions. Clearly, the χ^2 value of Weibull distribution is the smallest one. Consequently, it can be considered that our Hamming distances follow a Weibull distribution with $a = 47.89$ and $b = 3.17$. Fig. 7 is the comparison between the empirical distribution and the theoretical Weibull distribution. When the threshold T is given, the corresponding collision probability of our hashing can be calculated by Eq. (21) as follows.

$$\Pr(d_{\text{H}} \le T) = \int_0^T ba^{-b}t^{b-1}e^{-\left(\frac{t}{a}\right)^b} \mathrm{d}t. \tag{21}$$

TABLE 4
Collision Probabilities under Different Thresholds

Threshold	Collision probability
1	4.72×10^{-6}
2	4.25×10^{-5}
3	1.53×10^{-4}
4	3.82×10^{-4}
6	1.40×10^{-3}
8	3.40×10^{-3}
10	7.00×10^{-3}

Table 4 presents our collision probabilities under different thresholds. Obviously, a small T leads to a low collision probability (a good discrimination). At the meanwhile, a small T will also inevitably reduce robustness performance. In practice, we can select a proper threshold in terms of the specific application to keep a tradeoff between robustness and discrimination.

4.3 Key Dependence

To validate our key dependence, color images in the Kodak image dataset are also taken as test images. For each image, different secret keys are used to generate hashes and then Hamming distances of these different hashes are calculated. The results show that all Hamming distances are big enough. For space limitation, a typical example is presented here, where the last image in the third row of Fig. 4a is the test image. Firstly, a group of keys (k_1, k_2) is used to extract image hash of the test image. Secondly, another 100 different groups of keys are exploited to generate hashes of the same image. In the experiment, only the secret keys are changed and other parameters are kept unchanged. Fig. 8 presents the Hamming distances between the hashes

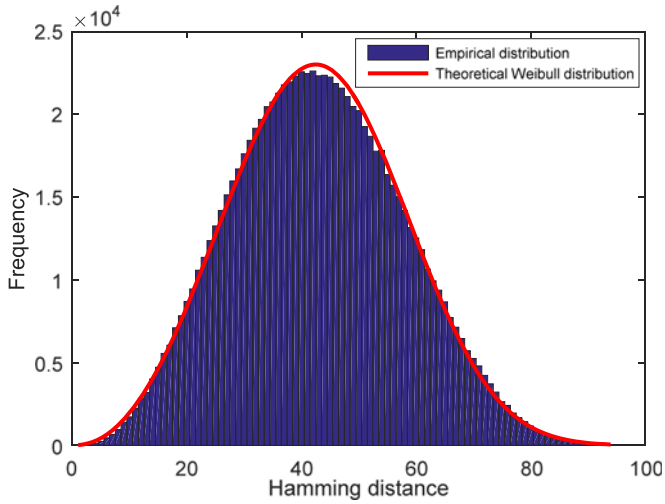

Fig. 7. Comparison between empirical distribution and theoretical Weibull distribution.

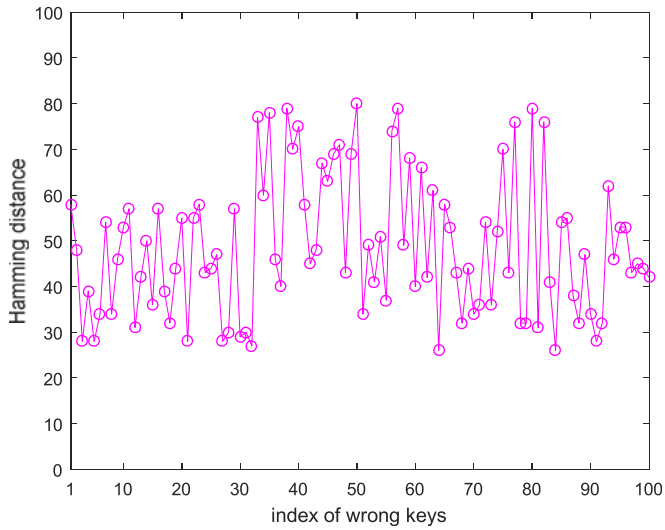

Fig. 8. Hamming distances between hashes of an image controlled with different keys.

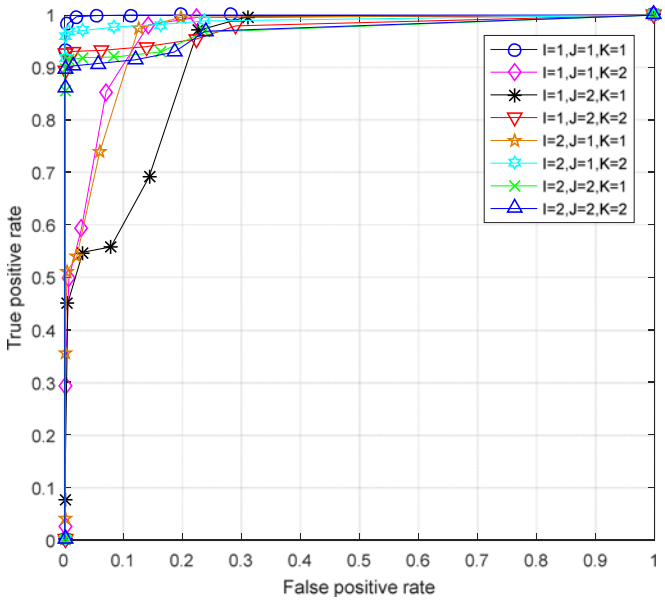

Fig. 9. ROC curve comparisons among different parameters settings.

controlled by different groups of keys, where the x-axis is the index of wrong keys and the y-axis is the Hamming distance. It can be seen that all Hamming distances are much bigger than 20. This illustrates that our hashing is key-dependent.

4.4 Effect of Key Parameters on Hash Performances

In this section, we discuss effect of the parameters of Tucker decomposition, i.e., I, J and K, on hash performances. Generally, the parameter values of I, J and K should be smaller than the tensor order in practice. Since a three-order tensor is constructed in this study, the possible parameter values of I, J and K are 1 or 2. Consequently, there are 8 combinations of these parameters as follows: (1) $I = 1, J = 1$ and $K = 1$; (2) $I = 1, J = 1$ and $K = 2$; (3) $I = 1, J = 2$ and $K = 1$; (4) $I = 1, J = 2$ and $K = 2$; (5) $I = 2, J = 1$ and $K = 1$; (6) $I = 2, J = 1$ and $K = 2$; (7) $I = 2, J = 2$ and $K = 1$; (8) $I = 2, J = 2$ and $K = 2$.

To validate our classification performance between robustness and discrimination under these parameter settings, the open image datasets used in Sections 4.1 and 4.2 are also taken. Here, the well-known tool called receiver operating characteristics (ROC) graph [68] is used to theoretically analyze experimental results. Generally, the x-axis of the ROC graph is false positive rate (FPR) and the y-axis is the true positive rate (TPR). Let P_{FPR} and P_{TPR} be FPR and TPR, respectively. Thus, their definitions are as follows.

$$P_{\text{FPR}}(d_{\text{H}} \leq T) = \frac{N_{\text{false}}}{N_{\text{different}}} \tag{22}$$

$$P_{\text{TPR}}(d_{\text{H}} \leq T) = \frac{N_{\text{true}}}{N_{\text{same}}}, \tag{23}$$

where N_{false} is the pairs of different images falsely classified as similar images, $N_{\text{different}}$ is the total pairs of different images, N_{true} is the pairs of visually similar images correctly detected as the same images, and N_{same} is the total number of visually similar images. It is clear that P_{FPR} and P_{TPR} represent discrimination and robustness, respectively. By varying the threshold T, a set of points $(P_{\text{FPR}}, P_{\text{TPR}})$ for drawing ROC curve is then available. Note that, as a good discrimination means a small P_{FPR} and a good robustness implies a big P_{TPR}, the ROC curve near the top-left corner is better than that far away from it.

TABLE 5
AUCs of Different Parameter Combinations

I, J, K	AUC
1, 1, 1	0.9993
1, 1, 2	0.9655
1, 2, 1	0.9213
1, 2, 2	0.9767
2, 1, 1	0.9636
2, 1, 2	0.9907
2, 2, 1	0.9709
2, 2, 2	0.9693

Fig. 9 presents ROC curve comparisons among these parameter settings. It is observed that the ROC curve of $I = 1, J = 1$ and $K = 1$ are closer to the top-left corner than those of other combinations of parameter values. To conduct quantitative analysis, the area under the ROC curve (AUC) [68] is taken as a measure for comparing classification performance. Note that the range of AUC is [0, 1]. A bigger AUC means a better classification performance. Table 5 lists AUCs of different parameter combinations. Obviously, the AUC of $I = 1, J = 1$ and $K = 1$ is the biggest one. This means that the classification performance of $I = 1, J = 1$ and $K = 1$ is better than those of other parameter combinations. In addition, computational time comparisons of extracting a hash are also conducted. Our TD hashing is implemented with MATLAB 2016a, running on a personal computer with 3.40 GHz Intel Core i7-6700 CPU and 8.0 GB RAM. The operating system installed in this computer is Windows 7 professional (64-bit version). The total consumed time of extracting hashes of 1338 different images in discrimination test are calculated to find the average time of a hash. It is found that the average time of these parameter combinations is about 0.17 seconds.

5 Performance Comparisons

To show advantage of our TD hashing, we compared it with some state-of-the-art algorithms, including GF-LVQ hashing [45], SVD-CSLBP hashing [43], random walk based hashing [51], local histogram based hashing [29] and MDS hashing [44]. The reason why we select them as compared algorithms is that they are recently reported in famous journals or conference. In the comparisons, those images used in Sections 4.1 and 4.2 are also selected to evaluate robustness and discrimination of the assessed algorithms. To make fair comparisons, all images are resized to 256×256 during hash generation, and the values of other parameters of the compared algorithms are the same with those reported in their original papers. Meanwhile, hash similarity metrics presented in their original papers are also used, i.e., correlation coefficient for SVD-CSLBP hashing and MDS hashing, and normalized Hamming distance for GF-LVQ hashing, random walk based hashing and local

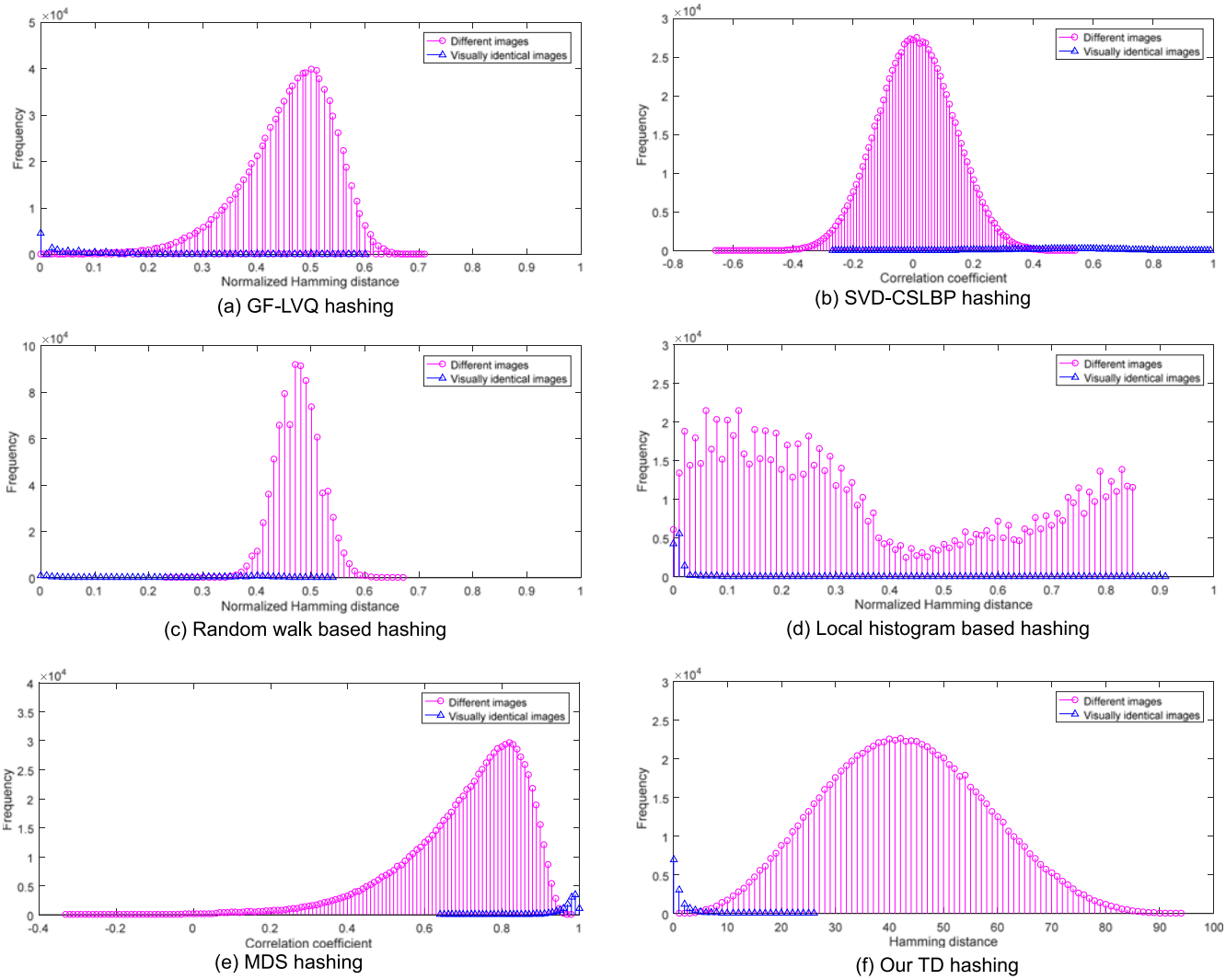

Fig. 10. Visual classification comparisons among different algorithms.

histogram based hashing. The results of our TD hashing with $I = 1, J = 1$ and $K = 1$ are taken for comparisons. Therefore, hash lengths of SVD-CSLBP hashing and MDS hashing are 64 floats and 180 integers (equaling 720 bits), and those of GF-LVQ hashing, random walk based hashing and local histogram based hashing are 120, 144 and 448 bits, respectively.

Fig. 10 illustrates visual classifications of different hashing algorithms, where the similarity results of visually identical images and different images are both drawn in the same figure for easy comparisons. The results of GF-LVQ hashing, SVD-CSLBP hashing, random walk based hashing, local histogram based hashing, MDS hashing and our TD hashing are shown in Figs. 10a, 10b, 10c, 10d, 10e and 10f, respectively. It can be seen that the overlapping regions between the distribution results of visually identical images and different images always exist for every hashing algorithm. However, the number of identical images and different images in the overlapping interval of our TD hashing is smaller than those in the overlapping intervals of other assessed hashing algorithms. Note that the results in the overlapping interval will be inevitably misclassified. A small image number in the overlapping interval means a better classification. Therefore, it can be intuitively concluded that the classification performance of our TD hashing outperforms those of the compared algorithms.

Moreover, ROC graph is exploited again to make theoretical analysis. Fig. 11 presents the ROC curve comparisons among these assessed hashing algorithms. It is observed that the ROC curve of our TD hashing is above those of other compared algorithms, and is much closer to the top-left corner than others. To make quantitative analysis, AUCs of the assessed algorithms are also calculated. The results of GF-LVQ hashing, SVD-CSLBP hashing, random walk based hashing, local histogram based hashing, MDS hashing and our TD hashing are 0.9929, 0.9601, 0.9417, 0.9572, 0.9909 and 0.9993, respectively. Clearly, AUC of our TD hashing is bigger than the AUCs of other compared algorithms. This means that our TD hashing is superior to the compared algorithms in classification between robustness and discrimination.

Computational time comparison for generating a hash is also evaluated. To do so, the total consumed time of generating hashes in discrimination test is recorded to calculate the average time of extracting a hash. All hashing algorithms are coded with MATLAB 2016a, running on the personal computer with the same configuration as the above mentioned. It is found that the average time of GF-LVQ hashing,

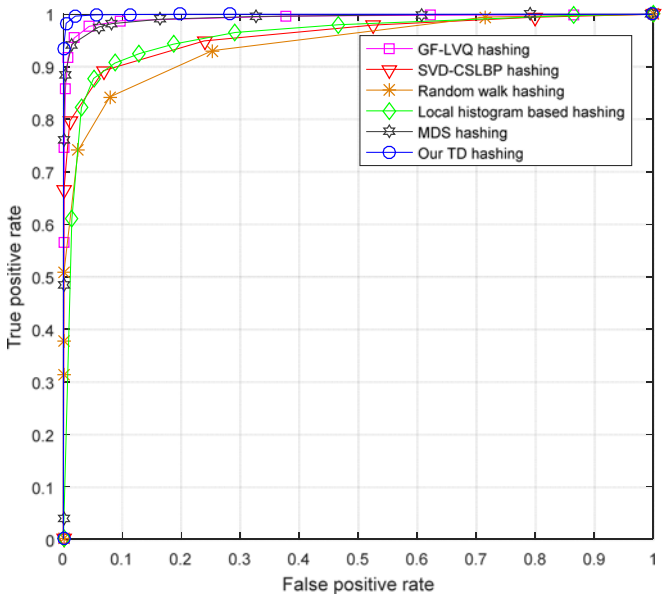

Fig. 11. ROC curve comparisons among different hashing algorithms.

SVD-CSLBP hashing, random walk based hashing, local histogram based hashing, MDS hashing and our TD hashing are 0.32, 0.10, 0.03, 0.02, 0.22 and 0.17 seconds, respectively. Our TD hashing is faster than GF-LVQ hashing and MDS hashing, but slower than SVD-CSLBP hashing, random walk based hashing and local histogram based hashing. Summary of performance comparisons among different hashing algorithms is presented in Table 6. It can be seen that our TD hashing is better than the compared algorithms in classification performance in terms of AUC. As to computational time, our TD hashing has moderate performance. Our hash length is 96 bits, which is much shorter than GF-LVQ hashing, random walk based hashing, local histogram based hashing and MDS hashing. For SVD-CSLBP hashing, its hash length is 64 floats. According to the IEEE standard for floating-point arithmetic [69], 32 bits at least are required for storing a float. Therefore, in binary form, the hash length of SVD-CSLBP hashing is $64 \times 32 = 2048$ bits, which is much longer than our length.

6 Conclusions

We have proposed an efficient image hashing based on tensor decomposition, so as to reach good robustness and desirable discrimination. The main contributions are the three-order tensor construction and the novel use of Tucker decomposition. Our robustness is contributed by the three-order tensor constructed from the normalized image, and the discrimination is ensured by Tucker decomposition since the factor matrices can reflect intrinsic structure of original tensor. Experiments with three open image datasets have been conducted, and the results have shown that our TD hashing is robust against many content-preserving operations, reaches good discrimination and is key-dependent. Comparisons with some state-of-the-art algorithms have been also carried out, and the results have illustrated that the performances of our TD hashing are better than those of the compared algorithms in classification and hash length. In future work, we will apply TD to video hashing, investigate the use of deep learning techniques in image hashing, and develop image hashing with visual attention model.

TABLE 6
Summary of Performance Comparisons among the Assessed Hashing Algorithms

Algorithm \ Performance	AUC	Time (s)	Hash length
GF-LVQ hashing	0.9929	0.32	120 bits
SVD-CSLBP hashing	0.9601	0.10	64 floats
Random walk based hashing	0.9417	0.03	144 bits
Local histogram based hashing	0.9572	0.02	448 bits
MDS hashing	0.9909	0.22	720 bits
Our TD hashing	0.9993	0.17	96 bits

Acknowledgments

The authors would like to thank the anonymous referees and the editor for their valuable comments and suggestions, which can substantially improve this paper. The authors would like to thank Dr. Y. Li for providing his source code [45]. This work is partially supported by the National Natural Science Foundation of China (61562007, 61672177, 61762017), the National Key R & D Program of China (2016YFB1000905), the Guangxi "Bagui Scholar" Teams for Innovation and Research, the Guangxi Natural Science Foundation (2017GXNSFAA198222, 2015GXNSFDA139040), the Project of Guangxi Science and Technology (GuiKeAD17195062), Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, and the Project of the Guangxi Key Lab of Multi-source Information Mining & Security (16-A-02-02).

References

[1] G. Qi, X. Hua, and H. Zhang, "Learning semantic distance from community-tagged media collection," in *Proc. 17th ACM Int. Conf. Multimedia*, 2009, pp. 243–252.
[2] G. Qi, C. Aggarwal, Q. Tian, J. Heng, and T. Huang, "Exploring context and content links in social media: A latent space method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 850–862, May 2012.
[3] J. Wang, Z. Zhao, J. Zhou, H. Wang, B. Cui, and G. Qi, "Recommending Flickr groups with social topic model," *Inf. Retrieval*, vol. 15, no. 3–4, pp 278–295, 2012.
[4] X. Liu, S. Ji, W. Glänzel, and B. D. Moor, "Multiview partitioning via tensor methods," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 1056–1069, May 2013.
[5] C. Qin, M. Sun, and C. C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," *Signal Process.*, vol. 142, pp. 194–205, 2018.
[6] L. Liu and L. Shao, "Sequential compact code learning for unsupervised image hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2526–2536, Dec. 2016.
[7] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. IEEE Int. Conf. Image Process.*, 2000, pp. 664–666.
[8] Z. Tang, X. Zhang, and S. Zhang, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 711–724, Mar. 2014.
[9] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A secure and robust hash based scheme for image authentication," *Signal Process.*, vol. 90, no. 5, pp. 1456–1470, 2010.
[10] Z. Tang, S. Wang, X. Zhang, W. Wei, and S. Su, "Robust image hashing for tamper detection using non-negative matrix factorization," *J. Ubiquitous Convergence Technol.*, vol. 2, no. 1, pp. 18–26, 2008.

[11] M. Schneider and S. F. Chang, "A robust content based digital signature for image authentication," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 227–230, 1996.
[12] A. Gionis, P. Indyky, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. 25th VLDB Conf.*, 1999, pp. 518–529.
[13] J. Yagnik, D. Strelow, D. A. Ross, and R. Lin, "The power of comparative reasoning," in *Proc. 13th IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2431–2438.
[14] K. Li, G. Qi, J. Ye, and K. A. Hua, "Linear subspace ranking hashing for cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1825–1838, Sep. 2017.
[15] X. Wang, T. Zhang, G. Qi, J. Tang, and J. Wang, "Supervised quantization for similarity search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2018–2016.
[16] T. Zhang, G. Qi, J. Tang, and J. Wang, "Sparse composite quantization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4548–4556.
[17] J. Song, Y. Yang, X. Li, Z. Huang, and Y. Yang, "Robust hashing with local models for approximate similarity search," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1225–1236, Jul. 2014.
[18] X. Zhu, L. Zhang and Z. Huang, "A sparse embedding and least variance encoding approach to hashing," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3737–3750, Sep. 2014.
[19] C. S. Lu, C. Y. Hsu, S. W. Sun, and P. C. Chang, "Robust mesh-based hashing for copy detection and tracing of images," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2004, pp. 731–734.
[20] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph PCA hashing for similarity search," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2033–2044, Sep. 2017.
[21] X. Lu, X. Zheng, and X. Li, "Latent semantic minimal hashing for image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 355–368, Jan. 2017.
[22] C. Qin, C. C. Chang, and P. Y. Chen, "Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism," *Signal Process.*, vol. 92, no. 4, pp. 1137–1150, 2012.
[23] X. Lv and Z. J. Wang, "Reduced-reference image quality assessment based on perceptual image hashing," in *Proc. IEEE Int. Conf. Image Process.*, 2009, pp. 4361–4364.
[24] W. Lu and M. Wu, "Multimedia forensic hash based on visual words," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 989–992.
[25] S. Xiang, H. J. Kim, and J. Huang, "Histogram-based image hashing scheme robust against geometric deformations," in *Proc. ACM Multimedia Security Workshop*, 2007, pp. 121–128.
[26] Y. S. Choi and J. H. Park, "Image hash generation method using hierarchical histogram," *Multimedia Tool Appl.*, vol. 61, no. 1, pp. 181–94, 2012.
[27] Z. Tang, L. Huang, Y. Dai, and F. Yang, "Robust image hashing based on multiple histograms," *Int. J. Digital Content Technol. Appl.*, vol. 6, no. 23, pp. 39–47, 2012.
[28] Z. Tang, Y. Dai, X. Zhang, and S. Zhang, "Perceptual image hashing with histogram of color vector angles," *Lecture Notes Comput. Sci.*, vol. 7669, pp. 237–246, 2012.
[29] L. N. Vadlamudi, R. P. V. Vaddella, and V. Devara, "Robust hash generation technique for content-based image authentication using histogram," *Multimedia Tools Appl.*, vol. 75, no. 11, pp. 6585–6604, 2016.
[30] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: performance evaluation and trade-offs," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3453–3466, Nov. 2006.
[31] C. Qin, C. C. Chang, and P. L. Tsou, "Robust image hashing using non-uniform sampling in discrete Fourier domain," *Digital Signal Process.*, vol. 23, no. 2, pp. 578–85, 2013.
[32] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," in *Proc. IEEE Int. Conf. Inf. Technol.: Coding Comput.*, 2000, pp. 178–183.
[33] Z. Tang, F. Yang, L. Huang, and X. Zhang, "Robust image hashing with dominant DCT coefficients," *Optik-Int. J. Light Electron Optics*, vol. 125, no. 18, pp. 5102–5107, 2014.
[34] F. Lefebvre, B. Macq, and J.-D. Legat, "RASH: Radon soft hash algorithm," in *Proc. Eur. Signal Process. Conf.*, 2002, pp. 299–302.
[35] D. Wu, X. Zhou, and X. Niu, "A novel image hash algorithm resistant to print–scan," *Signal Process.*, vol. 89, no. 12, pp. 2415–2424, 2009.
[36] Y. Lei, Y. Wang, and J. Huang, "Robust image hash in radon transform domain for authentication," *Signal Process.: Image Commun.*, vol. 26, no. 6, pp. 280–288, 2011.
[37] Z. Tang, L. Huang, F. Yang, and X. Zhang, "Robust image hashing based on fan-beam transform," *ICIC Express Letters*, vol. 8, no. 8, pp. 2365–2372, 2014.
[38] S. S. Kozat, K. Mihcak, and R. Venkatesan, "Robust perceptual image hashing via matrix invariants," in *Proc. IEEE Int. Conf. Image Process.*, 2004, pp. 3443–3446.
[39] V. Monga and M. K. Mihcak, "Robust and secure image hashing via non-negative matrix factorizations," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 376–390, Sep. 2007.
[40] F. Khelifi and J. Jiang, "Analysis of the security of perceptual image hashing based on non-negative matrix factorization," *IEEE Signal Process. Letters*, vol. 17, no. 1, pp. 43–46, Jan. 2010.
[41] Z. Tang, S. Wang, X. Zhang, W. Wei, and Y. Zhao, "Lexicographical framework for image hashing with implementation based on DCT and NMF," *Multimedia Tool Appl.*, vol. 52, no. 2–3, pp. 325–45, 2011.
[42] L. Ghouti, "Robust perceptual color image hashing using quaternion singular value decomposition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 3794–3798.
[43] R. Davarzani, S. Mozaffari and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4639–4667, 2016.
[44] Z. Tang, Z. Huang, X. Zhang, and H. Lao, "Robust image hashing with multidimensional scaling," *Signal Process.*, vol. 137, pp. 240–250, 2017.
[45] Y. Li, Z. Lu, C. Zhu, and X. Niu, "Robust image hashing based on random Gabor filtering and dithered lattice vector quantization," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1963–1980, Apr. 2012.
[46] X. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1081–93, Jun. 2012.
[47] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using Zernike moments and local features," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 55–63, Jan. 2013.
[48] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 200–214, Jan. 2016.
[49] C. Yan, C. Pun, and X. Yuan, "Multi-scale image hashing using adaptive local feature extraction for robust tampering detection," *Signal Process.*, vol. 121, pp. 1–16, 2016.
[50] C. Qin, X. Chen, D. Ye, J. Wang, and X. Sun, "A novel image hashing scheme with perceptual robustness using block truncation coding," *Inf. Sci.*, vol. 361, pp. 84–99, 2016.
[51] X. Huang, X. Liu, G. Wang, and M. Su, "A robust image hashing with enhanced randomness by using random walk on zigzag blocking," in *Proc. IEEE TrustCom/BigDataSE/ISPA*, 2016, pp. 14–18.
[52] C. Qin, X. Chen, X. Luo, X. Zhang, and X. Sun, "Perceptual image hashing via dual-cross pattern encoding and salient structure detection," *Inf. Sci.*, vol. 423, pp. 284–302, 2018.
[53] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (3rd edition), Englewood Cliffs, NJ, USA: Prentice-Hall, 2007.
[54] W. Burger and M. J. Burge, *Principles of Digital Image Processing: Core Algorithm*, London, UK: Springer, 2009.
[55] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[56] X. Cao, X. Wei, Y. Han, and D. Lin, "Robust face clustering via tensor decomposition," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2546–2557, Nov. 2015.
[57] L. R. Tucker, Implications of factor analysis of three-way matrices for measurement of change. In C. W. Harris (Ed.), *Problems in measuring change*, Madison, Wisconsin, USA: University of Wisconsin Press, pp. 122–137, 1963.
[58] L. Zaorálek, M. Prílepok, and V. Snášel, "Recognition of face images with noise based on tucker decomposition," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2015, pp. 2649–2653.
[59] C. Cheng and H. Wang, "Quality assessment for color images with Tucker decomposition," in *Proc. 19th IEEE Int. Conf. Image Process.*, 2012, pp. 1489–1492.
[60] A. Karami, M. Yazdi, and A. Z. Asli, "Noise reduction of hyperspectral images using kernel non-negative tucker decomposition," *IEEE J. Selected Topics Signal Process.*, vol. 5, no. 3, pp. 487–493, Jun. 2011.
[61] L. Li and D. Boulware, "High-order tensor decomposition for large-scale data analysis," in *Proc. IEEE Int. Congress Big Data*, 2015, pp. 665–668.
[62] P. M. Kroonenberg and J. Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, pp. 69–97, 1980.

[63] R. Franzen, Kodak lossless true color image suite. [Online]. Available: http://r0k.us/graphics/kodak/, Accessed April 15, 2017.
[64] H. Jégou, INRIA Copydays dataset. [Online]. Available: http://lear.inrialpes.fr/∼jegou/data.php, Accessed May 28, 2016.
[65] F. A. P. Petitcolas, "Watermarking schemes evaluation," *IEEE Signal Process. Mag.*, vol. 17, no. 5, pp. 58–64, Sep. 2000.
[66] G. Schaefer and M. Stich, "UCID - An uncompressed colour image database," in *Proc. SPIE Storage Retrieval Methods Appl. Multimedia*, 2004, pp. 472–480.
[67] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses* (3rd Ed), New York, NY, USA, Springer, 2005, pp. 590–599.
[68] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Letters*, vol. 27, no. 8, pp. 861–874, 2006.
[69] IEEE, "IEEE Standard for Floating-Point Arithmetic," *IEEE Std 754-2008*, pp. 1–70, Aug. 2008.

Zhenjun Tang (M'15) received the BS and MEng degrees from Guangxi Normal University, Guilin, China, in 2003 and 2006, respectively, and the PhD degree from Shanghai University, Shanghai, China, in 2010. He is now a professor with the Department of Computer Science, Guangxi Normal University. His research interests include image processing and multimedia security. He has contributed more than 50 international journal papers. He is a reviewer of more than 20 SCI journals, such as IEEE/IET journals, Elsevier journals, and Springer journals. He is a member of the IEEE.

Lv Chen received the BS degree from Guangxi Normal University, Guilin, China, in 2015. He is working towards the MEng degree in computer science at Guangxi Normal University.

Xianquan Zhang received the MEng degree from Chongqing University, Chongqing, China. He is currently a professor with the Department of Computer Science, Guangxi Normal University. His research interests include image processing and computer graphics. He has contributed more than 100 papers.

Shichao Zhang (M'04-SM'04) received the PhD degree in computer science from Deakin University, Geelong, VIC, Australia. He is currently a China 1000-Plan Distinguished Professor with the Department of Computer Science, Guangxi Normal University, Guilin, China. He has authored more than 60 international journal papers and 70 international conference papers. His current research interests include data quality and pattern discovery. He is a member of the Association for Computing Machinery. As a chief investigator, he has won four Australian Large ARC Grants, three China 863 Programs, two China 973 Programs, and five NSFs of China Grants. He served/serves as an associate editor of the *IEEE Transactions on Knowledge and Data Engineering*, *Knowledge and Information Systems*, and the *IEEE Intelligent Informatics Bulletin*. He served as a PC chair or conference chair for six international conferences. He is a senior member of the IEEE