EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Robust image hashing with compressed sensing and ordinal measures

Zhenjun Tang[1*], Hanyun Zhang[1], Shenglian Lu[1], Heng Yao[2] and Xianquan Zhang[1]

* Correspondence: tangzj230@163.com; zjtang@gxnu.edu.cn
[1]Guangxi Key Lab of Multi-Source Information Mining & Security, and Department of Computer Science, Guangxi Normal University, Guilin 541004, China
Full list of author information is available at the end of the article

Abstract

Image hashing is an efficient technology for processing digital images and has been successfully used in image copy detection, image retrieval, image authentication, image quality assessment, and so on. In this paper, we design a new image hashing with compressed sensing (CS) and ordinal measures. This hashing method uses a visual attention model called Itti model and Canny operator to construct an image representation, and exploits CS to extract compact features from the representation. Finally, the CS-based compact features are quantized via ordinal measures. L2 norm is used to judge similarity of hashes produced by the proposed hashing method. Experiments about robustness validation, discrimination test, block size discussion, selection of visual attention model, selection of quantization scheme, and effectiveness of the use of ordinal measures are conducted to verify performances of the proposed hashing method. Comparisons with some state-of-the-art algorithms are also carried out. The results illustrate that the proposed hashing method outperforms some compared algorithms in classification according to ROC (receiver operating characteristic) graph.

Keywords: Image hashing, Visual attention model, Saliency map, Compressed sensing, Ordinal measures

1 Introduction

In the Internet era, many people publish their daily photos on the web via social platform, such as Twitter, Facebook, and Instagram. Some of them would like to copy the photos of their friends and re-distribute them on the web. Consequently, there are many copies of some images in cyberspace. Therefore, detecting image copy is an important task of the community of image processing research. In the past years, many researchers try to solve the problem of image copy detection by an efficient technology called image hashing [1, 2]. This technology can not only quickly find similar copies of a given image, but also effectively distinguish different images.

In general, image hashing maps a digital image to a short sequence of numbers called image hash in a one-way manner. As image hash can represent its original image in practice and its storage cost is low, image hashing can achieve efficient processing in many image applications [3–7], such as image copy detection, image forensics, image authentication, image quality assessment, and image retrieval. Generally speaking, image hashing should meet two basic properties [8–10]. One property is robustness, which requires that image hashing should produce the same or similar hashes from those images with the same visual contents regardless of their digital bit-representations. Since some people may process image copy via editing tool (e.g., ACD-See and PhotoShop) before republishing them, this property can ensure high correct detection of image copies. The other property is discrimination, which is also called anti-collision capability in some hashing papers. This property demands that hashing algorithm must extract discriminative features from input image, and thus it can significantly reduce the number of images falsely returned. In other words, discriminative hashes should be produced from different images.

The concept of image hashing is firstly proposed at the end of the 20th century [11], but it has attracted much attention of multimedia community in the past decade. The early techniques of hashing algorithms include discrete wavelet transform (DWT) [11], Radon transform [12], singular value decomposition (SVD) [13], discrete Fourier transform (DFT) [14], feature point [15], discrete cosine transform (DCT) [16], and so on. In recent years, some other techniques are also exploited to build hashing algorithms for different application purposes. For example, Li et al. [17] jointly used Gabor filtering and vector quantization to construct hash for resisting image rotation. To improve discrimination, Ghouti [18] proposed to calculate hash of color image via quaternion SVD. Similarly, Tang et al. [19] selected color vector angle (CVA) as the feature of color image and conduct feature compression by DWT. In another study, Li et al. [20] derived hash from color image by quaternion polar cosine transform. To improve rotation robustness, Tang et al. [21] extracted perceptual statistical features from image rings invariant to rotation and compressed them by using vector distance. Huang et al. [22] incorporated random walk into zigzag blocking for enhancing hash security. Tang et al. [23] proposed a novel hashing scheme by using CVA and Canny operator. To build a hashing with good robustness, Qin et al. [24] computed perceptual features based on block truncation coding and center-symmetrical local binary pattern. In another work, Yan et al. [25] proposed a novel hashing algorithm for tampering localization by combining quaternion Fourier-Mellin moments and quaternion Fourier transform. Zhang et al. [26] improved the image hashing based on non-negative matrix factorization [2] by converting a rectangular image to a circular image using interpolation mapping. In another study, Zhang et al. [27] exploited non-subsampled contourlet transform and salient region detection to design hashing method for authentication. Tang et al. [28] constructed a feature matrix invariant to rotation by log-polar transform and DFT, and learned hash from the matrix by multidimensional scaling. Recently, Qin et al. [29] utilized hybrid features based on CVA, Canny operator, and SVD to construct hash of color image. Tang et al. [30] combined a visual attention model with DFT's phase spectrum and ring partition to design a hashing algorithm resilient to rotation. Li et al. [31] exploited neural network to build a new hashing algorithm for learning robust hash. The above-mentioned hashing algorithms have shown competitive performances in their applications. But their classification between robustness and discrimination do not reach the expected performance yet.

In this paper, we develop a new hashing method based on compressed sensing and ordinal measures. Compared with the current hashing algorithms, our work has two significant contributions.

(1) We exploit compressed sensing (CS) to extract compact features from the image representation constructed by visual attention model and Canny operator. The use of visual attention model can make the constructed representation indicating visual attention of human eyes, and thus improves perceptual robustness of the extracted features. The Canny operator can efficiently find image edges, which are discriminative features for human visual system (HVS). Therefore, compressed sensing applied to the image representation can derive a compact sequence of robust and discriminative features.

(2) We propose to quantize CS-based compact features via ordinal measures. As the ordinal measure is an efficient technique for feature compression, the use of ordinal measures can derive a short hash from the CS-based compact features.

Various experiments are done with open image databases to validate performances of the proposed method. The results demonstrate that the proposed method reaches good classification performance and is superior to some current hashing algorithms in terms of robustness and discrimination. The structure of the remainder of this paper is as follows. Section 2 introduces the proposed method. Section 3 presents experimental results and discussion, and Section 4 conducts comparison with some current hashing algorithms. Section 5 makes conclusions of this paper.

2 Proposed method

The proposed method consists of five steps, as shown in Fig. 1. Input image is firstly interpolated to a normalized size $Q \times Q$ by bicubic interpolation. This operation can reach two functions. The first one is that our hashing method can resist image resizing. The second one is that hashes of input images with different sizes have the same hash length. The second step includes two operations. The first operation is the saliency map extraction from the resized image and the second operation is the edge detection. Next, the results of saliency map extraction and edge detection are combined to produce a weighted image representation. And then, compressed sensing is exploited to extract compact features from the image representation. Finally, the compact features are quantized by using ordinal measure. Details of saliency map extraction, edge detection, weighted representation computation, compressed sensing, and ordinal measures are introduced in the below sections.

2.1 Saliency map extraction

To improve perceptual robustness, we incorporate saliency map into hash generation. In this paper, saliency map is extracted via a famous visual attention model proposed by Itti et al. [32]. The Itti model can effectively extract the saliency map of the focus

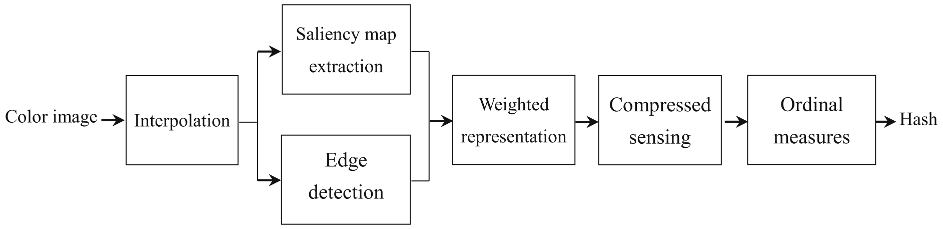


Fig. 1 Block diagram of the proposed method

area of human eye and has been widely applied to many fields, such as image classification [33], feature detection [34], and image search [35]. Generally, the Itti model is decomposed of four steps. The first step is the extraction of the saliency map of colors by conducting the operations of Gaussian pyramids, center-surround operations, and across-scale combinations. The second step is the extraction of the saliency map of intensity by similar procedure of the saliency map extraction of colors. Similarly, the third step is the extraction of saliency map of orientations by similar procedure of the saliency map extraction of colors. Lastly, the final saliency map is generated by using the above three maps as follows:

$$\mathbf{S} = \frac{1}{3}(\mathbf{S}_1 + \mathbf{S}_2 + \mathbf{S}_3) \tag{1}$$

where $\mathbf{S}_1$, $\mathbf{S}_2$, and $\mathbf{S}_3$ are the saliency maps of colors, intensity, and orientations, respectively. More details of the classical algorithm of the Itti model can be referred to its original paper [32]. Figure 2 presents an example of the results of detecting saliency map by the Itti model, where (a) is an input image, (b) is the color map $\mathbf{S}_1$, (c) is the intensity map $\mathbf{S}_2$, (d) is the orientation map $\mathbf{S}_3$, and (e) is the final map $\mathbf{S}$. Here, the Itti model is chosen to conduct saliency map extraction due to the following reason. Compared with other visual attention models, such as SR model [36] and PFT model [37], the Itti model can provide our hashing method a better classification performance. Experiment will prove this in Section 3.4.

2.2 Edge detection

Image edge is a useful visual feature and has been successfully used in many applications, such as image matching, image denoising and image retrieval. In general, different images have different image edges. HVS can discriminate different images according to their edges. Based on these considerations, we select image edge as discriminative feature for hash generation. To do so, the well-known algorithm called Canny operator [38] is exploited to conduct edge detection. Generally speaking, Canny operator consists of five phases as follows: (1) a smooth image is generated for alleviating noise effect on detection result by a Gaussian filter. (2) Intensity gradients of the smooth image are then extracted by a first-order difference operator. (3) Non-maximum suppression is exploited to reduce spurious response to edge detection. (4) Potential edges are determined by using double thresholds. (5) Final edges are extracted by suppressing those edges which are weak and not connected to strong edges. Details of the classical algorithm of Canny operator can be found in [38].

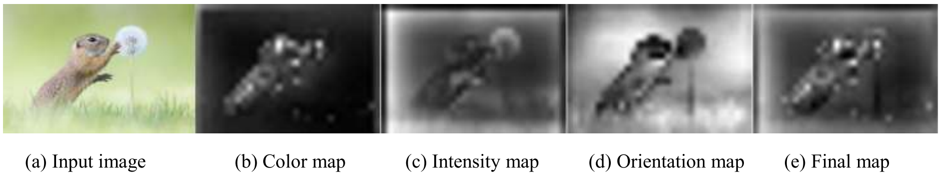

(a) Input image (b) Color map (c) Intensity map (d) Orientation map (e) Final map

Fig. 2 An example of the results of detecting saliency map by Itti model

As the input of Canny operator is a grayscale image, we select luminance component of color image for representation. To do so, the resized color image in RGB color space is mapped to the YCbCr color space by the below formula.

$$\begin{bmatrix} Y \\ C_{\mathrm{b}} \\ C_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{2}$$

where *R*, *G*, and *B* are the red, green, and blue components of a color pixel, *Y* is its luminance component, C_{b} and C_{r} are its blue-difference and red-difference chromas, respectively. Let **D** be the detection result of Canny operator. Thus, its element *D*(*i*,*j*) in the *i*th row and *j*th column is determined by the below rule.

$$D(i,j) = \begin{cases} 1, & \text{If } J(i,j) \text{ is an edge point,} \\ 0, & \text{Otherwise.} \end{cases} \tag{3}$$

in which *J*(*i*,*j*) is the pixel of in the *i*th row and *j*th column of the resized color image. Figure 3 demonstrates an example of Canny operator, where (a) is the luminance component of Fig. 2a, (b) is the result of edge detection by Canny operator.

2.3 Weighted representation computation

To generate perceptual edges of color image, visual saliency map is incorporated into the detected result of Canny operator. Specifically, the detected edges and the detected saliency map are combined to produce a weighted representation of color image. Let **I** be the weighted representation, where *I*(*i*, *j*) is its element in the *i*th row and *j*th column (1 ≤ *i* ≤ *Q*, 1 ≤ *j* ≤ *Q*). Thus, it can be determined by the following formula.

$$I(i,j) = D(i,j) \times S(i,j) \tag{4}$$

where *S*(*i*, *j*) is the element of the detected saliency map **S** in the *i*th row and *j*th column.

2.4 Compressed sensing

As the dimensions of the weighted representation are the same with the resized color image, compressed sensing is exploited to extract compact features from the weighted representation. Compressed sensing (CS) [39] also called compressive sensing [40] is a new and effective way of signal processing. CS theory breaks through the limitations of sampling with Nyquist theorem and can directly achieve compression during the

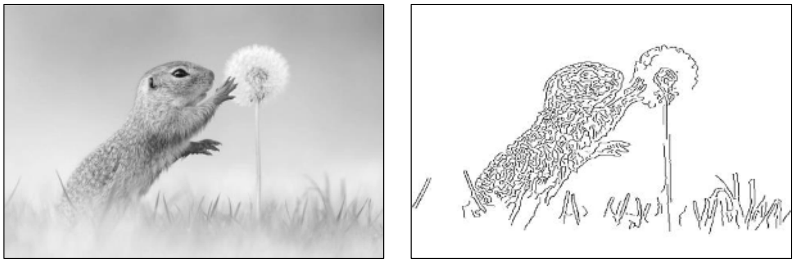

(a) Luminance component of Fig. 2 (a) (b) Image edge of Fig. 3 (a) (image negative for display)

Fig. 3 An example of Canny operator

sampling process. CS theory has illustrated that if a signal is sparse in an orthogonal space, it can be sampled at a low frequency and it can be also reconstructed from the sampled data by solving an optimization problem. In the past years, CS has attracted much attention and has been successfully used in many applications [40, 41], such as image processing, image steganography, video processing, pattern recognition, and communication system. Let $\mathbf{x} \in \mathbb{R}^{N \times 1}$ be a real-value signal. Assume that $\mathbf{x}$ can be sparsely denoted with the sparse basis set $\mathbf{\Psi} \in \mathbb{R}^{N \times P}$ by the following formula.

$$\mathbf{x} = \mathbf{\Psi}\boldsymbol{\alpha} \tag{5}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{P \times 1}$ is K-sparse and $K<<N$. Thus, CS can obtain a measurement vector $\mathbf{y} \in \mathbb{R}^{M \times 1}$ ($M<<N$) by the below formula.

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\boldsymbol{\alpha} = \boldsymbol{\theta}\boldsymbol{\alpha} \tag{6}$$

in which $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is the sensing matrix (measurement matrix) and $\boldsymbol{\theta}$ is the perceptual matrix (the product of $\mathbf{\Phi}\mathbf{\Psi}$). As the number of the elements in $\mathbf{y}$ is much smaller than the number of the elements in $\mathbf{x}$, $\mathbf{y}$ is generally viewed as the compression of $\mathbf{x}$. More details of CS can be referred to [39, 40]. In this study, the wavelet transform is selected as the sparse basis set and the measurement vector is exploited to construct compact feature.

To extract local discriminative features, the weighted representation $\mathbf{I}$ is divided into non-overlapping blocks sized $b \times b$. For simplicity, let Q be the integral multiple of b. Therefore, there are $L=(Q/b)^2$ blocks in total. Suppose that $\mathbf{x}_i$ is the ith block of the weighted representation numbered from top to bottom and left to right ($1< i \leq L$). Here CS is applied to the block $\mathbf{x}_i$ and its measurement vector $\mathbf{y}_i$ is then generated. To indicate element fluctuation of the measurement vector $\mathbf{y}_i$, the variance is chosen as the block feature which can be calculated by the following formula.

$$v_i = \frac{1}{M-1}\sum\nolimits_{j=1}^{M}[y_i(j)-m_i]^2 \tag{7}$$

where $y_i(j)$ is the jth element of $\mathbf{y}_i$, and m_i is the mean of $\mathbf{y}_i$, which is determined by the below formula.

$$m_i = \frac{1}{M}\sum\nolimits_{j=1}^{M} y_i(j) \tag{8}$$

After the calculation of vector variance, a small vector $\mathbf{v}$ is available as follows.

$$\mathbf{v} = [v_1, \ v_2, \ldots, v_L]^{\mathrm{T}} \tag{9}$$

Clearly, the vector $\mathbf{v}$ consists of L floating-point numbers.

2.5 Ordinal measures

According to the IEEE standard [42], 32 bits are needed to store a floating-point number. This means that the storage cost of the vector $\mathbf{v}$ is $32L$ bits. To reduce the cost of hash storage and further improve classification performance between robustness and discrimination, the vector $\mathbf{v}$ is represented by using the well-known ordinal measures [43]. The ordinal measures are robust and compact features and have been widely used in many applications, such as video signature [44], iris recognition [45], and face recognition [46]. In general, the ordinal measures of the elements of a data sequence can be generated by sorting these elements in ascending order and taking their positions in the sorted sequence for representation. Table 1 demonstrates an example of ordinal measures, where the second row is an original data sequence with 10 elements, the third row is the sorted version the original sequence in ascending order, and the final row is the ordinal measures of the elements of the original sequence. Clearly, the first element of the original sequence is 2, which locates at the 2nd position of the sorted sequence. Therefore, its ordinal measure is 2. Similarly, the second element of the original sequence is 8, which located at the 6th position of the sorted sequence. Therefore, its ordinal measure is 6.

Here, the ordinal measures of the elements of the vector **v** are selected as our hash elements. More specifically, our hash **h** is represented by

$$\mathbf{h} = [h_1, h_2, \ldots, h_L] \tag{10}$$

where the ith element h_i of **h** is the position of v_i of **v** in the sorted sequence in ascending order ($1< i \leq L$). Clearly, the length of our hash is L integers. Since the fixed-length encoding is used to store hash elements, $\lceil \log_2 L \rceil$ bits are needed for a hash element, where $\lceil \cdot \rceil$ is the upward rounding operation. Therefore, the length of our hash is $L\lceil \log_2 L \rceil$ bits in binary form. Section 3.6 will validate effectiveness of the use of ordinal measures. To achieve easy understanding of the proposed method, a visual example of our hash generation is presented in Fig. 4.

3 Results and discussion

In the experiments, the parameter settings of our method are as follows. Input image is interpolated to a fixed size 512×512 and the block size is 64×64. In other words, Q=512 and b=64. Consequently, $L=(Q/b)^2=(512/64)^2=64$. Therefore, our hash consists of 64 integers. In binary form, our hash length is $L\lceil \log_2 L \rceil = 64\lceil \log_2 64 \rceil = 384$ bits. To judge similarity of the hashes of two images, the metric called $L2$ norm is taken. Let $\mathbf{h_1} = [h_1(1),\ h_1(2), \ldots, h_1(L)]$ and $\mathbf{h_2} = [h_2(1),\ h_2(2), \ldots, h_2(L)]$ be two hashes of images. Thus, their $L2$ norm can be determined by the below formula.

$$d(\mathbf{h}_1, \mathbf{h}_2) = \sqrt{\sum\nolimits_{j=1}^{M} [h_1(j) - h_2(j)]^2} \tag{11}$$

where $h_1(j)$ and $h_2(j)$ are the j-th elements of $\mathbf{h}_1$ and $\mathbf{h}_2$, respectively. Generally, the $L2$ norm of the hashes of two similar images (e.g., one is a copy of the other image) is expected to be small. If the $L2$ norm is bigger than a given threshold T, the corresponding images are judged as different images. The used platform for implementing our method is MATLAB 2016a. The configurations of the used computer are as follows. The CPU is an Intel Core i7-7700 processor with 3.60 GHz and size of the memory is 8 GB. Section 3.1 and Section 3.2 validate the performances of robustness and discrimination, respectively. Section 3.3, Section 3.4, Section 3.5 and Section 3.6 present block size

Table 1 An example of ordinal measures

Position	1	2	3	4	5	6	7	8	9	10
Original sequence	2	8	6	4	9	12	3	10	14	1
Sorted sequence	1	2	3	4	6	8	9	10	12	14
Ordinal measures	2	6	5	4	7	9	3	8	10	1

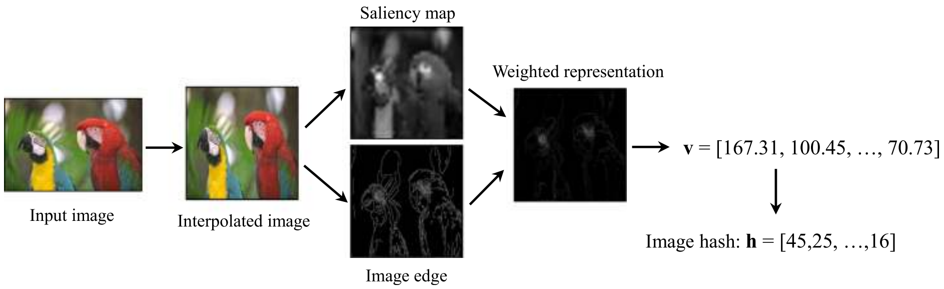


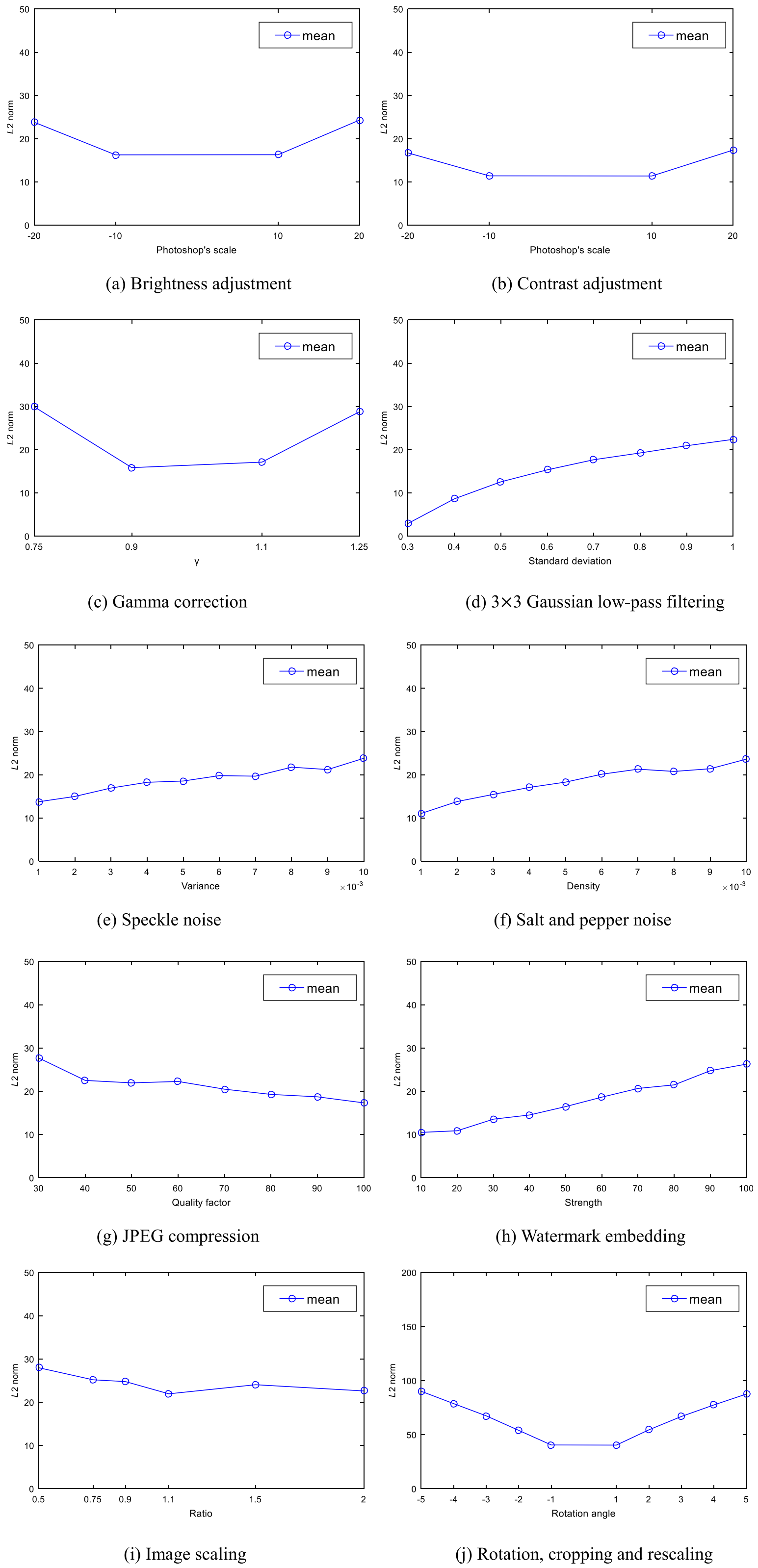
Fig. 4 A visual example of our hash generation

discussion, selection of visual attention model, selection of quantization scheme, and effectiveness of the use of ordinal measures, respectively.

3.1 Robustness

To measure robustness performance, the Kodak image database [47] is selected as the test dataset. This database consists of 24 color images. The sizes of these images can be divided into two kinds. One kind is 768×512 and the other size is 512×768. In this experiment, three tools, i.e., Photoshop, MATLAB and StirMark [48], are taken to produce similar images of the 24 color images. Specifically, the used operations of Photoshop are the adjustments of contrast and brightness (four parameters per operation). The used operations of MATLAB include gamma correction (4 parameters), 3×3 Gaussian low-pass filtering (8 parameters), salt and pepper noise (10 parameters), and speckle noise (ten parameters). The provided operations of StirMark are JPEG compression (eight parameters), watermark embedding (ten parameters), image scaling (six parameters), and combinational operation of rotation, cropping, and rescaling (10 parameters). In summary, ten digital operations are used and they contribute 74 manipulations in total. Consequently, every original image has 74 similar versions. Therefore, there are 24×74=1776 pairs of similar images in the robustness test and the number of the used images reaches 1776 + 24 = 1800.

Figure 5 demonstrates robustness performances of our method under different operations based on the Kodak database, where the x-axis represents the parameter values of the used operation, and the y-axis represents the mean value of the $L2$ norms of the hashes between each original image and its similar image produced by the used operation with corresponding parameter. From Fig. 5, it can be seen that the maximum means of the used operations with all parameters are smaller than 40, except those of the combinational operation of rotation, cropping and rescaling. Table 2 presents the detailed statistical results of different operations. It is easy to find that the mean $L2$ norms of all operations are less than 25, except that of the combinational attack of rotation, cropping and rescaling. The mean $L2$ norm of the combinational operation is about 66. It is much bigger than those of other operations. This is because, compared with single operation, combinational operation brings much distortion to similar images. Moreover, the maximum $L2$ norm of the combinational operation is 145.70 and those of other operations are less than 65. Therefore, when the threshold is set as T = 80, correct detection rate of similar images is 96.11%. If there is no similar image



(a) Brightness adjustment

(b) Contrast adjustment

(c) Gamma correction

(d) 3×3 Gaussian low-pass filtering

(e) Speckle noise

(f) Salt and pepper noise

(g) JPEG compression

(h) Watermark embedding

(i) Image scaling

(j) Rotation, cropping and rescaling

Fig. 5 Robustness results on Kodak database

Table 2 Detailed statistical results of different operations

Operation	Maximum	Minimum	Mean	Standard deviation
Brightness adjustment	64.76	4.24	20.18	10.53
Contrast adjustment	36.74	3.74	14.23	6.12
Gamma correction	62.08	7.62	22.92	10.07
3 × 3 Gaussian low-pass filtering	36.47	0	14.96	7.90
Speckle noise	57.97	6.16	18.88	9.21
Salt and pepper noise	47.14	3.46	18.30	8.35
JPEG compression	59.32	7.35	21.28	9.19
Watermark embedding	60.66	3.74	17.77	9.81
Image scaling	59.11	9.80	24.42	9.62
Rotation, cropping, and rescaling	145.70	21.82	65.74	24.06

produced by the combinational operation, the correct detection rate can reach 100%. Similarly, when the threshold increases to $T = 100$, correct detection rate of similar images is 98.93%. If the threshold is set as $T = 150$, our method can correctly recognize all similar images.

3.2 Discrimination

An open image dataset called UCID [49] is taken to test discriminative capability of our method. The UCID consists of 1338 color images. The sizes of these color images can be also divided into two kinds. One kind is 512 × 384 and the other kind is 384 × 512. The hashes of these 1338 images are firstly extracted by using our method. For each image, the $L2$ norms between its hash and the hashes of other 1337 images are then computed. Consequently, the number of the valid $L2$ norms reaches $C_{1338}^{2} = 1338 \times (1338 - 1)/2 = 894453$. Figure 6 presents the distribution of these $L2$ norms, where the abscissa is the $L2$ norm and the ordinate is the frequency of the corresponding $L2$ norm. Statistics of these $L2$ norms are also calculated. The results are as follows: the minimum $L2$ norm is 38.37, the maximum $L2$ norm is 284.03, the mean is 200.20, and

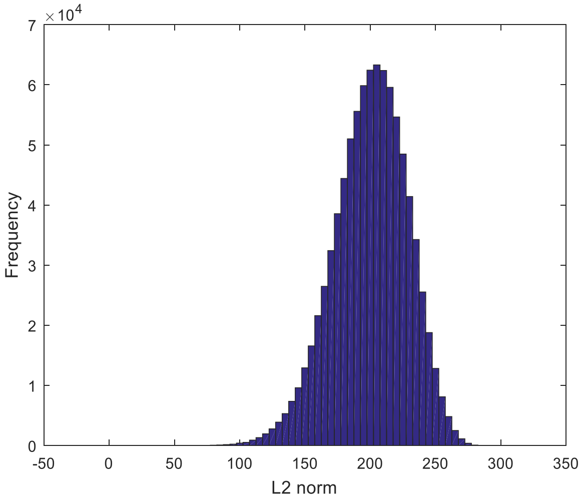


Fig. 6 Distribution of *L*2 norms based on UCID

the standard deviation is 28.00. From Fig. 6, it can be observed that most $L2$ norms are bigger than 100. This means that we can select the threshold around 100 according to the practical performances. Note that both performances of robustness and discrimination are closely related to the selected threshold. In general, a low threshold will improve discrimination but decrease robustness, and vice versa. Table 3 presents our robustness and discrimination performances under different thresholds, where the correct detection rate (R_1) represents the robustness performance, the false recognition rate (R_2) denotes the discrimination performance, and the total error rate ($(1 - R_1) + R_2$) indicates the whole performance of our method. Clearly, the smaller the total error rate, the better the whole performance. From Table 3, it is found that the threshold 100 can be selected as a recommended value since it reaches the smallest total error rate.

3.3 Block size discussion

To view effect of block size, the experiments of our method with different settings of block size are discussed in this section. The selected block sizes include 16×16, 32×32, 64×64, 128×128, and 256×256. In the experiments, only the block size is different and other parameters are all the same. The datasets used for the experiments of robustness and discrimination are the same databases mentioned in Sections 3.1 and 3.2.

To make theoretical analysis of the experimental results, the receiver operating characteristic (ROC) graph [50] is exploited. Here, false positive rate (P_1) is selected as the abscissa of the ROC graph and true positive rate (P_2) is taken as the ordinate of the ROC graph. More specifically, the values of P_1 and P_2 can be calculated by the following equations.

$$P_1(d \le T) = \frac{N_{1,1}}{N_{1,2}} \tag{12}$$

$$P_2(d \le T) = \frac{N_{2,1}}{N_{2,2}} \tag{13}$$

in which $N_{1,1}$ is the number of different images falsely judged as similar images, $N_{1,2}$ is the number of all different images, $N_{2,1}$ is the number of similar images correctly detected as similar images, and $N_{2,2}$ is the number of all similar images. Clearly, P_1 and P_2 correspond to discrimination and robustness. A low P_1 means good discrimination, while a high P_2 implies good robustness. Note that a curve in the ROC graph consists of a set of points (P_1, P_2), which can be obtained by using a set of thresholds. As the

Table 3 Our performances under different thresholds

Threshold	Correct detection rate (R_1)	False recognition rate (R_2)	Total error rate = (1 - R_1) + R_2
80	96.110%	0.006%	3.896%
90	97.750%	0.027%	2.277%
100	98.930%	0.081%	0.012%
110	99.380%	0.210%	0.830%
120	99.660%	0.500%	0.840%
130	99.890%	1.130%	1.300%
140	99.940%	2.340%	2.360%
150	100%	4.540%	4.540%

curve near the top-left corner of the ROC graph has a low P_1 and a high P_2, this can be used to intuitively judge whether the evaluated hashing reaches a good performance or not. To conduct quantitative analysis, the area under ROC curve (AUC) is often calculated, whose value ranges from 0 to 1. The bigger the AUC, the better the hashing performance.

The ROC curves of different block sizes are illustrated in Fig. 7. To show details, the curves near the top-left part are zoomed in and placed in the right-bottom of Fig. 7. From the results, it can be found that the curves of 32 × 32 and 64 × 64 are much nearer the top-left corner than those of other block sizes. As to the AUC, the values of 16 × 16, 32 × 32, 64 × 64, 128 × 128, and 256 × 256 are 0.99978, 0.99991, 0.99993, 0.99918, and 0.89944, respectively. Since the AUC of 64 × 64 is bigger than those of other block sizes, our method with block size 64 × 64 is better than our method with other block sizes in terms of ROC graph. Computational costs of different block sizes are also tested. To do so, the total consumed time of extracting hashes of 1338 images in UCID is calculated. It is found that the block sizes 16 × 16, 32 × 32, 64 × 64, 128 × 128, and 256 × 256 need 1397.586, 659.103, 389.702, 303.891, and 277.832 s, respectively. Our method with 64 × 64 runs faster than our method with 16 × 16 or 32 × 32, but it is slower than our method with 128 × 128 or 256 × 256. Similarly, the length of our method with 64 × 64 is 64 integers. It is shorter than that of our method with 16 × 16 or 32 × 32, but it is longer than that of our method with 128 × 128 or 256 × 256. Table 4 lists summary of performance comparison among different block sizes.

3.4 Selection of visual attention model

To make robust hash, visual attention model is exploited to extract saliency map in the second step of our method. To validate effectiveness of our selection, Itti model is compared with other two visual attention models, i.e., SR model [36] and PFT model [37]. The selected models are both reported in the famous conference about computer vision and widely used in many applications of image processing. The SR model calculates spectral residual (SR) with the log spectrum of an image and transforms the SR to

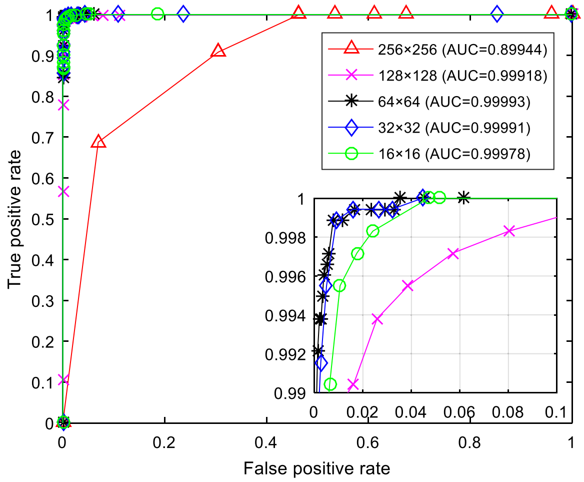


Fig. 7 ROC curves of different block sizes

Table 4 Our performances under different block sizes

Block size	AUC	Hash length (integer)	Total run time (s)
16 × 16	0.99978	1024	1397.586
32 × 32	0.99991	256	659.103
64 × 64	0.99993	64	389.702
128 × 128	0.99918	16	303.891
256 × 256	0.89944	4	277.832

spatial domain for detecting saliency map. The PFT model exploits phase spectrum of Fourier transform (PFT) to find saliency map of an image. More details of the SR model and the PFT model can be found in [36, 37], respectively.

Figure 8 demonstrates ROC curve comparisons among different visual attention models, where the curves near the top-left corner is enlarged and presented in the right-bottom part of the figure. It can be seen that the curve of the Itti model is much nearer the top-left corner than those of the SR model and the PFT model. As to AUC, the values of the SR model, the PFT model and the Itti model are 0.99978, 0.98075, and 0.99993, respectively. The AUC of the Itti model is bigger than those of other models. This means that our method with the Itti model is better than our method with the SR model and the PFT model in terms of ROC graph. Computational time of extracting hashes of 1338 images is also compared. The time of the SR model, the PFT model and the Itti model is 270.451, 293.565, and 389.702 s, respectively. Our method with the Itti model runs slower than our method with the SR model and our method with the PFT model. The hash lengths of our method with different models are all 64 integers since their block numbers are the same. Table 5 lists performance comparisons among different visual attention models.

3.5 Selection of quantization scheme

To reduce storage cost of the extracted vector, ordinal measures are exploited to conduct quantization in the fifth step of our method. To illustrate effectiveness of our

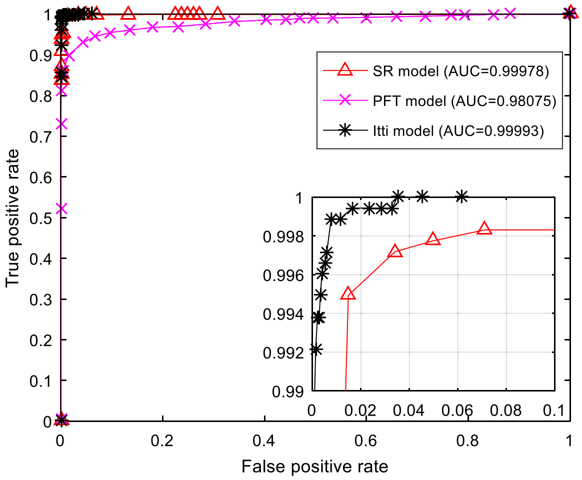


Fig. 8 ROC curves of different visual attention models

Table 5 Performance comparisons among different visual attention models

Model	AUC	Hash length (integer)	Total run time (s)
SR model	0.99978	64	270.451
PFT model	0.98075	64	293.565
Itti model	0.99993	64	389.702

selection, the performances of our method with ordinal measures are compared with the performances of our method with other quantization scheme. Here, the selected schemes are the well-known methods called median quantization and mean quantization. For the scheme of median quantization, the elements of the vector **v** are also sorted in ascending order and the element in the median position of the sorted sequence is taken as the threshold to binarize the elements of **v** (i.e., the element bigger than the threshold is represented by 1. Otherwise, it is denoted by 0). For the scheme of mean quantization, the mean value of all elements of the vector **v** is first calculated and then taken as the threshold to binarize the elements of **v**. Since the hashes of median quantization and mean quantization both consist of bits, the Hamming distance is used to calculate similarity instead of $L2$ norm.

Figure 9 illustrates the ROC curves of different quantization schemes, where the details of the curves near the top-left part are enlarged in the bottom-right of the figure. It can be seen that the curve of ordinal measures is nearer to the top-left corner than the curves of median quantization and mean quantization. As to AUC, the values of median quantization, mean quantization, and ordinal measures are 0.99973, 0.99962, and 0.99993, respectively. The AUC of ordinal measures is bigger than those of other quantization schemes. This means that our method with ordinal measures is better than our method with median quantization and mean quantization in terms of ROC graph. As to computational cost, the total time of median quantization, mean quantization and ordinal measures is 392.027, 391.069 and 389.702 s for hash generation of 1338 images, respectively. Our method with ordinal measures is slightly better than our method with median quantization and mean quantization in computational complexity. In addition, the hash lengths of median quantization, mean quantization,

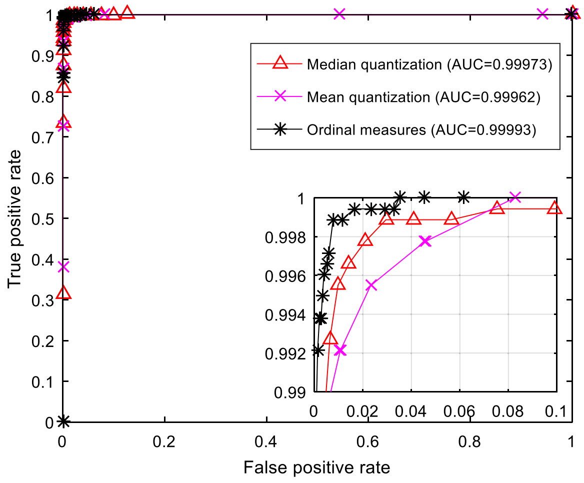


Fig. 9 ROC curves of different quantization schemes

and ordinal measures are 64, 64, and 384 bits, respectively. Table 6 summarizes performance comparison among different quantization schemes.

3.6 Effectiveness of the use of ordinal measures

To show advantage of the use of ordinal measures, the ROC curve of our hashing without ordinal measures is also calculated. Note that our hashing without ordinal measures is obtained by removing the step of ordinal measures in the proposed method. Figure 10 is the ROC curve comparison between our hashing with ordinal measures and our hashing without ordinal measures. It can be seen that the ROC curve of our hashing with ordinal measures is much nearer the top-left corner than the curve of our hashing without ordinal measures. As to AUC, the values of our hashing with ordinal measures and our hashing without ordinal measures are 0.99993 and 0.99959, respectively. The AUC of our hashing with ordinal measures is bigger than that of our hashing without ordinal measures. It means that our hashing with ordinal measures is better than our hashing without ordinal measures in terms of ROC graph. This validates effectiveness of the use of ordinal measures in our proposed method. In addition, the hash length of our hashing without ordinal measures is L floating numbers, equaling to $32L$ bits in binary form according to the IEEE standard [39]. For our hashing with ordinal measures, its hash length is $L\lceil \log_2 L \rceil$. It is clear that $L\lceil \log_2 L \rceil < 32L$ when $L < 2^{32} = 4.295 \times 10^9$. Note that L is the block number, which is a small value in practice. For example, L=64 in the experiments. Therefore, the hash lengths of our hashing without ordinal measures and our hashing with ordinal measures are 2048 and 384 bits, respectively. Obviously, our hashing with ordinal measures is better than our hashing without ordinal measures in the performance of hash length. In summary, the use of ordinal measures can not only make a short hash, but also improves classification performance between robustness and discrimination in terms of AUC.

4 Performance comparisons

In this section, our hashing method is compared with some state-of-the-art algorithms. The selected hashing algorithms include random-walk hashing [22], CVA-Canny hashing [23], and hybrid features-based hashing [29]. The main procedures of the compared algorithms are as follows:

(1) **Random-walk hashing:** This hashing consists of three steps. It firstly divides input image into small rectangles under the control of a secret key. Secondly, it exploits random-walk algorithm to generate several zigzag blocks by combining these rectangles. This operation is also controlled by a secret key. If some rectangles still exist after the second step, they are split by random-walk algorithm again. Finally, expectation of luminance of every zigzag block is used to form image hash.

Table 6 Performance comparison among different quantization schemes

Quantization scheme	AUC	Hash length (bit)	Total run time (s)
Median quantization	0.99973	64	392.027
Mean quantization	0.99962	64	391.069
Ordinal measures	0.99993	384	389.702

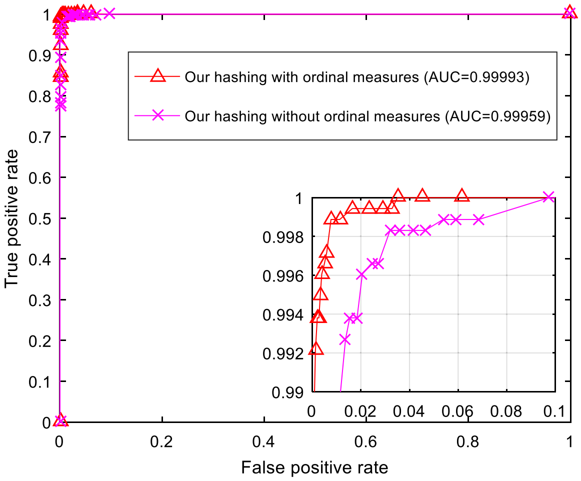


Fig. 10 ROC curve comparison between our hashing with ordinal measures and our hashing without ordinal measures

(2) **CVA-Canny hashing:** This hashing firstly creates a normalized image by interpolation and a Gaussian low-pass filter. Secondly, it calculates CVAs of all pixels and extracts image edge via Canny operator. Finally, it divides CVA matrix into concentric circles, extracts variances of CVA of those edge pixels on the concentric circles and quantizes them to produce a compact hash.

(3) **Hybrid features-based hashing:** This hashing also includes three steps: preprocessing, hybrid feature extraction, and hash generation. In the pre-processing, image normalization, Gaussian low-pass filter and SVD are jointly exploited to improve robustness. In the second step, the hybrid features, i.e., the circle-based structural features and the block-based structural features, are extracted by using CVAs and Canny operator. Finally, the hybrid features are quantized and scrambled to make a short hash.

From the above reviews, it can be found that our hashing is significantly different from the compared algorithms, especially the used techniques of saliency map extraction, CS and ordinal measures. In the experiments, those images used in Sections 3.1 and 3.2 are both selected to test robustness performances and discriminative capabilities of the compared hashing algorithms, where all images are converted to a standard size 512 × 512 before hash generation. As to our hashing method, the experimental results under the settings of block size 64 × 64, Itti model and ordinal measures are taken for performance comparisons.

Figure 11 presents ROC curve comparison between our hashing method and the compared hashing algorithms. To view details of the ROC curves around the top-left corner, a zoomed-in view of the ROC curves is placed in the right-bottom part of Fig. 11. Clearly, the ROC curve of our hashing is much nearer the top-left corner than those of the compared hashing algorithms. It can be intuitively concluded that our hashing method is better than the compared hashing algorithms in classification performance of robustness and discrimination. Moreover, the AUCs of the assessed algorithms are also computed and the values of random-walk hashing, CVA-Canny hashing, hybrid features-based hashing and our hashing are 0.96650, 0.99297, 0.99469, and 0.99993,

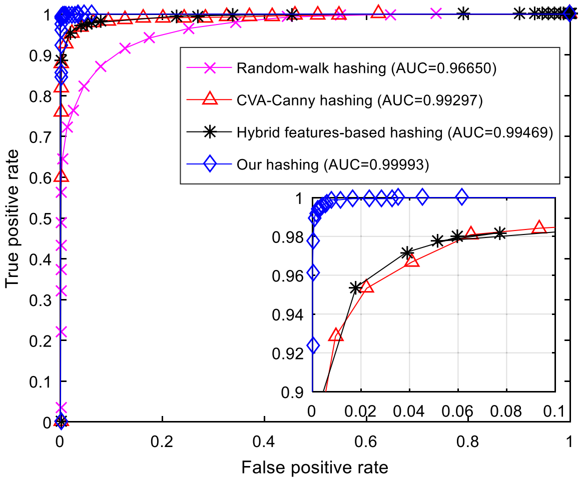


Fig. 11 ROC curve comparisons among different algorithms

respectively. The AUC of our hashing method is bigger than those of the compared hashing algorithms. This validates that our hashing method is superior to the compared hashing algorithms in the performances of classification between robustness and discrimination.

Computational time of the assessed hashing algorithms is also compared. In the experiments, the average time of calculating a hash is chosen. To do this, the assessed hashing algorithms are all exploited to calculate hashes of the 1338 images in UCID and then the total consumed time is used to compute the average time. It is found that the average time of random-walk hashing, CVA-Canny hashing, hybrid features-based hashing, and our hashing is 0.0377, 0.0843, 32.3029 and 0.2913 s, respectively. Our hashing is slower than random-walk hashing and CVA-Canny hashing. However, our hashing is faster than the hybrid features-based hashing. The hybrid features-based hashing has a low speed due to the high computational cost of SVD. Hash storages are also compared. The lengths of the hashes generated by random-walk hashing, CVA-Canny hashing and hybrid features-based hashing are 144, 400, and 3328 bits, respectively. The length of our hash is 384 bits. It is longer than the length of random-walk hashing, but it is shorter than those of CVA-Canny hashing and hybrid features-based hashing. Performance summary of different algorithms is demonstrated in Table 7. From this table, it can be easily found that our hashing is better than the compared algorithms in classification between robustness and discrimination according to AUC. Our hashing has moderate performance in computational time. It is better than hybrid features-based hashing, but it is not better than other compared algorithms. As to hash length, our hashing is better than all compared algorithms, except random-walk hashing.

Table 7 Performance summary of different algorithms

Performance	Random-walk hashing	CVA-Canny hashing	Hybrid features-based hashing	Our hashing
AUC	0.96650	0.99297	0.99469	0.99993
Average time (s)	0.0377	0.0843	32.3029	0.2913
Hash length (bit)	144	400	3328	384

5 Conclusions

In this paper, we have proposed a new image hashing with CS and ordinal measures. The CS is exploited to find compact features from the weighted image representation, which is determined by jointly using the Itti model and Canny operator. Since the Itti model can effectively detect saliency map indicating visual attention of human eyes, perceptual robustness of the image features extracted from the weighted representation is improved. As the ordinal measures can efficiently achieve feature compression, the use of ordinal measures can derive a short hash from the CS-based compact features. Experiments of robustness and discrimination have been done and discussions about block size selection, selection of visual attention model, selection of quantization scheme, and effectiveness of the use of ordinal measures have been also made. Comparisons with some state-of-the-art algorithms have illustrated that our hashing method outperforms the compared algorithms in classification between robustness and discrimination according to ROC graph. As to the performances of computational time and hash length, our hashing is also superior to some compared algorithms.

Abbreviations

CS: Compressed sensing; ROC: Receiver operating characteristic; DWT: Discrete wavelet transform; SVD: Singular value decomposition; DFT: Discrete Fourier transform; DCT: Discrete cosine transform; CVA: Color vector angle; HVS: Human visual system; AUC: Area under ROC curve; SR: Spectral residual; PFT: Phase spectrum of Fourier transform; UCID: Uncompressed colour image database

Acknowledgements

The authors would like to thank the anonymous referees for their helpful comments and suggestions, and Miss M. Sun [29] for sharing their code for comparison.

Authors' contributions

ZT did the main work. HZ implemented the proposed method and carried out the experiments. SL participated in data analysis and discussion. HY and XZ offered the suggestions. All authors read and approved the final manuscript.

Authors' information

Zhenjun Tang received the B.S. and M.Eng. degrees from Guangxi Normal University, Guilin, P.R. China, in 2003 and 2006, respectively, and the PhD degree from Shanghai University, Shanghai, P.R. China, in 2010. He is now a professor with the School of Computer Science and Information Technology, Guangxi Normal University. His research interests include image processing and multimedia security. He has contributed more than 60 papers in international journals, such as *IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Information Forensics and Security*, *The Computer Journal, Computers & Security, Signal Processing*, *Digital Signal Processing, IET Image Processing, Multimedia Tools and Applications*, and *Fundamenta Informaticae*. He is a member of IEEE and a senior member of China Computer Federation (CCF) and a reviewer of more than 30 SCI journals, such as ACM journals, IEEE journals, IET journals, Elsevier journals, Springer journals, and Taylor & Francis journals.

Hanyun Zhang received the B. S. degree in computer science and technology from Guangxi Normal University in 2017. Currently, she is pursuing the M. Eng. degree in the School of Computer Science and Information Technology, Guangxi Normal University. Her research interests include image processing and multimedia security.

Shenglian Lu received his PhD degree in mechanical engineering from Shanghai Jiao Tong University in 2008. He works in the School of Computer Science and Information Technology Guangxi Normal University as an associate professor since 2016, before that he worked in National Engineering Research Center for Information Technology in Agriculture of China for 8 years. His research interests include image processing, machine learning, deep learning and its application in agriculture.

Heng Yao received the B.S. degree from Hefei University of Technology, China, in 2004, the M.S. degree from Shanghai Normal University, China, in 2008, and the PhD degree from Shanghai University, China, in 2012. Since 2012, he has been with School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China, where he is currently an Associate Professor. His research interests include digital forensics, data hiding, image processing, and pattern recognition. He has contributed more than 30 international journal papers.

Xianquan Zhang received the M. Eng. degree from Chongqing University, Chongqing, P.R. China. He is a Professor with the School of Computer Science and Information Technology, Guangxi Normal University. His research interests include image processing and information hiding. He has contributed more than 100 papers.

Funding

This work is partially supported by the National Natural Science Foundation of China (61962008, 61762017, 61762013, 61702332), Guangxi "Bagui Scholar" Team for Innovation and Research, the Guangxi Talent Highland Project of Big Data Intelligence and Application, the Guangxi Natural Science Foundation (2017GXNSFAA198222), the Project of Guangxi Science and Technology (GuiKeAD17195062), Guangxi Collaborative Innovation Center of Multi-source

Information Integration and Intelligent Processing, and the Innovation Project of Guangxi Graduate Education (YCSW2020109).

Availability of data and materials
The image datasets used to support the findings of this study can be downloaded from the public websites whose hyperlinks are provided in the article.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests

Author details
[1]Guangxi Key Lab of Multi-Source Information Mining & Security, and Department of Computer Science, Guangxi Normal University, Guilin 541004, China. [2]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China.

Received: 8 March 2020 Accepted: 28 April 2020
Published online: 08 June 2020

References
1. C.S. Lu, C.Y. Hsu, S.W. Sun, P.C. Chang, in *Proceedings of IEEE International Conference on Multimedia & Expo*. Robust mesh-based hashing for copy detection and tracing of images, vol 1 (ICME 2004, 2004), pp. 731–734
2. Z. Tang, X.Q. Zhang, S. Zhang, Robust perceptual image hashing based on ring partition and NMF. IEEE Trans. Knowl. Data Eng. **26**(3), 711–724 (2014)
3. W. Lu, M. Wu, in *Proceedings of IEEE International Conference on Image Processing*. Multimedia forensic hash based on visual words (ICIP 2010, 2010), pp. 989–992
4. J. Ouyang, X. Wen, J. Liu, J. Chen, Robust hashing based on quaternion Zernike moments for image authentication. ACM Trans. Multimed. Comput. Commun. Appl. **76**(2), 2609–2626 (2017)
5. Z. Tang, Z. Huang, H. Yao, X.Q. Zhang, L. Chen, C. Yu, Perceptual image hashing with weighted DWT features for reduced-reference image quality assessment. Comput. J. **61**(11), 1695–1709 (2018)
6. R.K. Karsh, R.H. Laskar, Aditi, Robust image hashing through DWT-SVD and spectral residual method. EURASIP J. Image Video Processing **31** (2017)
7. J. Song, L. Gao, L. Liu, X. Zhu, N. Sebe, Quantization-based hashing: a general framework for scalable image and video retrieval. Pattern Recogn. **75**, 175–187 (2018)
8. R. Venkatesan, S.-M. Koon, M.H. Jakubowski, P. Moulin, in *Proceedings of the IEEE International Conference on Image Processing*. Robust image hashing (ICIP 2000, 2000), pp. 664–666
9. C. Qin, X. Chen, X. Luo, X.P. Zhang, X. Sun, Perceptual image hashing via dual-cross pattern encoding and salient structure detection. Inf. Sci. **423**, 284–302 (2018)
10. Z. Tang, L. Chen, X.Q. Zhang, S. Zhang, Robust image hashing with tensor decomposition. IEEE Trans. Knowl. Data Eng. **31**(3), 549–560 (2019)
11. M. Schneider and S. F. Chang, A robust content based digital signature for image authentication, In: *Proceedings of IEEE International Conference on Image Processing* (ICIP 1996), vol. 3, pp.227–230,1996
12. F. Lefebvre, B. Macq and J. D. Legat, RASH: Radon soft hash algorithm, In: *Proc. of European Signal Processing Conference*, pp. 299–302, 2002
13. S. S. Kozat, M. K. Mihcak and R. Venkatesan, Robust perceptual image hashing via matrix invariants, In: *Proc. of IEEE International Conference on Image Processing* (ICIP 2004), pp.3443–3446, 2004
14. A. Swaminathan, Y. Mao, M. Wu, Robust and secure image hashing. IEEE Trans. Inform. Forensics Security **2**(1), 215–230 (2006)
15. V. Monga, B.L. Evans, Perceptual image hashing via feature points: performance evaluation and trade-offs. IEEE Trans. Image Process. **11**(15), 3453–3466 (2006)
16. Z. Tang, S. Wang, X.P. Zhang, W. Wei, Y. Zhao, Lexicographical framework for image hashing with implementation based on DCT and NMF. Multimed. Tools Appl. **52**(2-3), 325–345 (2011)
17. Y. Li, Z. Lu, C. Zhu, X. Niu, Robust image hashing based on random Gabor filtering and dithered lattice vector quantization. IEEE Trans. Image Process. **21**(4), 1963–1980 (2012)
18. L. Ghouti, Robust perceptual color image hashing using quaternion singular value decomposition, In: *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing* (ICASSP 2014), Oct. 27-30, pp.3794–3798, 2014
19. Z. Tang, Y. Dai, X.Q. Zhang, L. Huang, F. Yang, Robust image hashing via colour vector angles and discrete wavelet transform. IET Image Process. **8**(3), 142–149 (2014)
20. Y. Li, P. Wang, Y. Su, Robust image hashing based on selective quaternion invariance. IEEE Signal Proc. Lett. **22**, 2396–2400 (2015)
21. Z. Tang, X.Q. Zhang, X. Li, S. Zhang, Robust image hashing with ring partition and invariant vector distance. IEEE Trans. Inform. Forensics Security **11**(1), 200–214 (2016)
22. 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. of IEEE Trustcom/BigDataSE/ISPA*, pp.23–26, 2016
23. Z. Tang, L. Huang, X.Q. Zhang, H. Lao, Robust image hashing based on color vector angle and Canny operator. AEÜ Int. J. Electron. Comm. **70**(6), 833–841 (2016)

24. C. Qin, X. Chen, D. Ye, J. Wang, X. Sun, A novel image hashing scheme with perceptual robustness using block truncation coding. Inf. Sci. **361**, 84–99 (2016)
25. C. Yan, C.M. Pun, X. Yuan, Quaternion-based image hashing for adaptive tampering localization. IEEE Trans. Inform. Forensics Sec. **11**(12), 2664–2677 (2016)
26. Q. Zhang, Q. Dou, Z. Yang, Y. Yan, Perceptual hashing of color images using interpolation mapping and non-negative matrix factorization. J. Inf. Hiding Multim. Signal Proc. **8**(3), 525–535 (2017)
27. Q. Zhang, Z. Yang, Q. Dou, Y. Yan, Robust hashing for color image authentication using non-subsampled contourlet transform features and salient features. J. Inf. Hiding Multim. Signal Proc. **8**(5), 1029–1042 (2017)
28. Z. Tang, Z. Huang, X.Q. Zhang, H. Lao, Robust image hashing with multidimensional scaling. Signal Process. **137**, 240–250 (2017)
29. C. Qin, M. Sun, C.C. Chang, Perceptual hashing for color images based on hybrid extraction of structural features. Signal Process. **142**, 194–205 (2018)
30. Z. Tang, Y. Yu, H. Zhang, M. Yu, C. Yu, X.Q. Zhang, Robust image hashing via visual attention model and ring partition. Math. Biosci. Eng. **16**(5), 6103–6120 (2019)
31. Y. Li, D. Wang, L. Tang, Robust and secure image fingerprinting learned by neural network. IEEE Trans. Circuits Syst. Video Technol. **30**(2), 362–375 (2020)
32. L. Itti, C. Koch, E. Niebur, A model of saliency based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
33. F. Moosmann, E. Nowak, F. Jurie, Randomized clustering forests for image classification. IEEE Trans. Pattern Anal. Mach. Intell. **30**(9), 1632–1646 (2008)
34. J. van de Weijer, T. Gevers, A.D. Bagdanov, Boosting color saliency in image feature detection. IEEE Trans. Pattern Anal. Mach. Intell. **28**(1), 150–156 (2006)
35. J. Huang, X. Yang, X. Fang, W. Lin, R. Zhang, Integrating visual saliency and consistency for re-ranking image search results. IEEE Trans. Multimedia **13**(4), 653–661 (2011)
36. X. Hou and L. Zhang, Saliency Detection: A spectral residual approach, In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2007), pp.1-8, 2007
37. C. Guo, Q. Ma and L. Zhang, Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform, In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2008), pp.1-8, 2008
38. J. Canny, A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**(8), 679–698 (1986)
39. D.L. Donohn, Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
40. M. Rani, S.B. Dhok, R.B. Deshmukh, A systematic review of compressive sensing: Concepts, Implementations and Applications. IEEE Access **6**, 4875–4894 (2018)
41. J. Pan, W. Li, C. Yang, L. Yan, Image steganography based on subsampling and compressive sensing. Multimed. Tools Appl. **74**, 9191–9205 (2015)
42. IEEE Std754–2008, IEEE Standard for Floating-Point Arithmetic, pp.1–70, 2008
43. D. N. Bhat and S. K. Nayar, Ordinal Measures for visual correspondence, In: *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition* (ICCVPR 1996), pp.351-357, 1996
44. X. Hua, X. Chen and H. Zhang, Robust video signature based on ordinal measure, In: *Proc. of the IEEE International Conference on Image Processing* (ICIP 2004), pp.685-688, 2004
45. Z. Sun, T. Tan, Ordinal measures for iris recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(12), 2211–2226 (2009)
46. Z. Chai, Z. Sun, H. Méndez-Vázquez, R. He, T. Tan, Gabor ordinal measures for face recognition. IEEE Trans. Inform. Forensics Security **9**(1), 14–26 (2014)
47. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/, Accessed 15 Apr 2017
48. F.A.P. Petitcolas, Watermarking schemes evaluation. IEEE Signal Process. Mag. **17**(5), 58–64 (2000)
49. G. Schaefer and M. Stich, UCID -An Uncompressed Colour Image Database, In: *Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia*, pp. 472-480, 2004
50. T. Fawcett, An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.