Perceptual Image Hashing with Histogram of Color Vector Angles

Zhenjun Tang[1], Yumin Dai[1], Xianquan Zhang[1], and Shichao Zhang[1,2,*]

[1] Department of Computer Science, Guangxi Normal University, Guilin 541004, P.R. China
[2] Faculty of Information Technology, University of Technology, Sydney, NSW 2007, Australia
zhangsc@gxnu.edu.cn

Abstract. Image hashing is an emerging technology for the need of, such as image authentication, digital watermarking, image copy detection and image indexing in multimedia processing, which derives a content-based compact representation, called image hash, from an input image. In this paper we study a robust image hashing algorithm with histogram of color vector angles. Specifically, the input image is first converted to a normalized image by interpolation and low-pass filtering. Color vector angles are then calculated. Thirdly, the histogram is extracted for those angles in the inscribed circle of the normalized image. Finally, the histogram is compressed to form a compact hash. We conduct experiments for evaluating the proposed hashing, and show that the proposed hashing is robust against normal digital operations, such as JPEG compression, watermarking embedding, scaling, rotation, brightness adjustment, contrast adjustment, gamma correction, and Gaussian low-pass filtering. Receiver operating characteristics (ROC) curve comparisons indicate that our hashing performs much better than three representative methods in classification between perceptual robustness and discriminative capability.

Keywords: Perceptual hashing, image hashing, image authentication, color vector angle, color histogram.

1 Introduction

As today's digital images are easy to copy, edit and redistribute, content security of the digital images has become an actual and challenging issue. An efficient way for content protection is digital watermarking [1], which embeds a signal, called watermark, into the image and achieves authentication by verifying integrity of the extracted watermark. But modification during the embedding procedure will inevitably degrade visual quality of the image. This weakness makes digital watermarking unsuitable for those applications with highly demanding quality, such as medical images. Another alternative technology for data authentication is cryptographic hash functions, e.g., MD5 and SHA-1, which extract a short string called authentication code from the input message. Conventional cryptographic hash functions do not

* Corresponding author.

R. Huang et al. (Eds.): AMT 2012, LNCS 7669, pp. 237–246, 2012.

modify the input data and then preserve its digital representation. However, these algorithms are sensitive to bit-level change. One bit difference will lead to a completely different string. This property makes it applicable to text or data file verification, but unsuitable for digital images. In practice, digital images will undergo normal digital operations, which change the digital representation of images, but keep their visual appearances unchanged. Therefore, image authentication code should be a visual content-based string. In this work, we study the issue of image hashing, an emerging multimedia technology which can totally preserve image quality and achieve image authentication.

Image hashing is a technology for deriving a content-based compact representation, called image hash, from an input image, and has been widely used in many applications, such as image retrieval, image authentication, digital watermarking, image copy detection, image indexing, and multimedia forensics. In general, image hash function must satisfy two basic properties as follows. (1) Perceptual robustness: Visually identical images have the same or very similar hashes no matter what their digital representations are. In other words, image hashing should be robust against normal digital operations, such as image compression and geometric transforms. (2) Discriminative capability: Different images have different image hashes. It means that hash distance between different images should be large enough.

Many researchers have devoted themselves to developing image hashing algorithms in last decade. For example, Venkatesan et al. [2] used wavelet coefficient statistics to construct image hashes. Lin and Chang [3] designed image authentication system using robust hashing. Lefebvre et al. [4] pioneered the use of Radon transform (RT) to construct robust hash. Swaminathan et al. [5] used discrete Fourier transform coefficients to produce image hashes. Kozat et al. [6] proposed to calculate hashes using singular value decompositions (SVDs). Monga and Mihcak [7] are the first of applying non-negative matrix factorization (NMF) to derive image hashing. This NMF-based hashing is resilient to geometric attacks. Ou et al. [8] applied the RT to the input image, randomly selected 40 projections to perform 1-D discrete cosine transform (DCT), and took the first AC coefficient of each projection to construct hash. Tang et al. [9] used structural features to design image hashing and proposed a similarity metric for tamper detection.

Although many image hashing algorithms have been successfully developed for real applications, there are still some limitations. For example, most of the existing hashing algorithms [4, 7, 9] consider only the gray images. For color images, they use luminance components in YCbCr color space for representation. As hue and saturation information are discarded, discriminations existing in color images are not fully exploited. On the other hand, many hashing algorithms [2, 3] are sensitive to rotation. Some algorithms [4, 6, 8] are resilient to rotation, but their discriminative capabilities are not desirable. In this work, we propose a robust image hashing algorithm based on histogram of color vector angles. Our algorithm can overcome the above mentioned weaknesses due to the following strategies. (1) The color vector angle is calculated by fully exploiting all components of the color pixels, and therefore makes our hashing discriminative. (2) Since histogram is a global invariant statistics, our hashing is resistant to image rotation. Our experiments demonstrate that the proposed algorithm is

robust against normal digital operations including image rotation with arbitrary angle, and has good discriminative capability.

The rest of this paper is organized as follows. Section 2 describes the proposed image hashing. Section 3 and Section 4 present the experimental results and performance comparisons, respectively. Conclusions are finally drawn in Section 5.

2 Proposed Image Hashing

As shown in Fig. 1, our image hashing is generated in four phases. The input image is preprocessed to produce a normalized image in first phase. Color vector angles of the normalized image are then extracted. The third phase is a procedure of computing the histogram of those color vector angles in a circle region. In last phase, DCT is exploited to compress histogram and significant coefficients are selected as image hash. The proposed hashing is illustrated as follows.

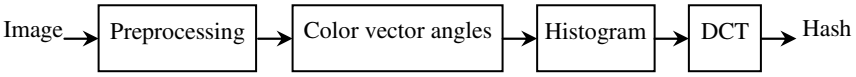

Fig. 1. Block diagram of our image hashing

2.1 Preprocessing Phase

In this phase, the input image is converted to a normalized size $M \times M$ by bi-cubic interpolation. This operation is to make our hash resistant to those images with different resolutions. After resizing, the image is then blurred by a Gaussian low-pass filter, which can be achieved by a convolution mask. Let $T_{\text{Gaussian}}(i, j)$ be the element in the i-th row and the j-th column of the convolution mask. It can be obtained with Formula (1) as follows.

$$T_{\text{Gaussian}}(i,j) = \frac{T^{(1)}(i,j)}{\sum_i \sum_j T^{(1)}(i,j)} \tag{1}$$

where $T^{(1)}(i, j)$ is defined as

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

in which σ is the standard deviation of all elements in the convolution mask. The filtering manipulation is useful to alleviating influences of minor modifications on the final hash, such as noise contamination.

2.2 Color Vector Angle Extraction

A color image can be depicted by its hue, saturation and luminance, where the hue represents color appearance, the saturation describes the amount of white contained in the color, and the luminance, also called intensity, is an indicator of brightness. Clearly, image features only extracted from luminance component are ineffective in measuring color change. To overcome this weakness, we exploit color vector angles to generate robust hash. Color vector angle is insensitive to intensity variations, but sensitive to the differences of hue and saturation when changing visual appearance of an image [10]. It has been successfully used in edge detection [10] and image retrieval [11]. Comparing with the Euclidean distance in RGB color space, color vector angle is more effective in evaluating perceptual differences between two colors. To illustrate this, we take Fig. 2 (adopted from [11]) as an example, where the color pair (C1, C2) is more similar than (C3, C4). The Euclidean distance between C1 and C2 is the same with that between C3 and C4, but the vector angle of (C1, C2) is much smaller than that of (C3, C4). The advantage of color vector angle is attributed to its sensitiveness to hue differences. This means that color vector angle can capture perceptual color difference. Let $P_1=[R_1, G_1, B_1]^T$ and $P_2=[R_2, G_2, B_2]^T$ be two colors in RGB color space, where R_1 and R_2, G_1 and G_2, B_1 and B_2, are their red, green, and blue components, respectively. The angle θ between P_1 and P_2 can be calculated with Formula (3) as follows.

$$\theta = \arcsin\left(1-\frac{(P_1^T P_2)^2}{P_1^T P_1 P_2^T P_2}\right)^{1/2} \tag{3}$$

To reduce computation, we use the sine value of θ for representation in our hashing.

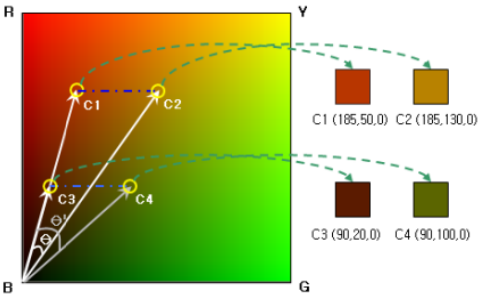

Fig. 2. Comparisons between the Euclidean distances and color vector angles of two color pairs

As the calculation of color vector angle requires two colors, we set a reference color $P_{ref}=[R_{ref}, G_{ref}, B_{ref}]^T$, where R_{ref}, G_{ref} and B_{ref} are the means of red, green and blue components of all pixels in the normalized image. With the reference color P_{ref}, we can calculate $\sin\theta_{i,j}$ for the pixel $P_{i,j}$ in the i-th row and j-th column ($1\leq i, j \leq M$). And then, we can obtain the matrix **A** of color vector angles of the input image as follows.

$$\mathbf{A} = \begin{bmatrix} \sin\theta_{1,1} & \sin\theta_{1,2} & \ldots & \sin\theta_{1,M} \\ \sin\theta_{2,1} & \sin\theta_{2,2} & \ldots & \sin\theta_{2,M} \\ \ldots & \ldots & \ldots & \ldots \\ \sin\theta_{M,1} & \sin\theta_{M,2} & \ldots & \sin\theta_{M,M} \end{bmatrix} \tag{4}$$

2.3 Histogram Calculation

Since rotation manipulation often takes image center as origin of coordinates, those pixels in the inscribed circle of an image are kept unchanged. Therefore, we calculate histogram of those color vector angles in the inscribed circle of **A** and use it to represent the input image. To do so, we find those color vector angles for histogram computation, whose coordinates (x, y) satisfy Formula (5) as follows.

$$(x - x_c)^2 + (y - y_c)^2 \le r^2 \tag{5}$$

where (x_c, y_c) is the coordinates of image center and $r=M/2$ is the radius of the inscribed circle. If M is an even number, $x_c=y_c=M/2+0.5$. Otherwise, $x_c=y_c=M/2$. As the result of $\sin\theta$ is a real number in the interval [0, 1], we quantize the interval with a step size 0.005 and then obtain $N=201$ discrete values, i.e., 0.000, 0.005, 0.010, …, 1.000. Let $f(t)$ be the value of the $(t+1)$-th bin of the histogram, where $t=0,1,\ldots,N-1$. Consequently, the total number of the histogram bins is N.

2.4 Compression Phase

To make the hash as short as possible, we exploit one dimensional DCT to compress the histogram of color vector angels. This is done as follows.

$$C(u) = a(u)\sum_{t=0}^{N-1} f(t)\cos\left[\frac{(2t+1)u\pi}{2N}\right] \quad u = 0,1,\ldots,N-1 \tag{6}$$

where $a(u)$ is determined by

$$a(u) = \begin{cases} \sqrt{1/N}, & u = 0 \\ \sqrt{2/N}, & u = 1,2,\ldots,N-1 \end{cases} \tag{7}$$

Then, we use the first n AC coefficients to form the hash **h**:

$$h(i)=C(i) \tag{8}$$

where $h(i)$ is the i-th element of **h**, and $i=1, 2, \ldots, n$.

2.5 Hash Similarity

To measure the similarity between a pair of image hashes $\mathbf{h}_1$ and $\mathbf{h}_2$, we take L_2 norm as similarity metric, which is defined as follows.

$$d(\mathbf{h}_1, \mathbf{h}_2) = \sqrt{\sum_{i=1}^{n} [h_1(i) - h_2(i)]^2} \tag{9}$$

where $h_1(i)$ and $h_2(i)$ are the i-th elements of $\mathbf{h}_1$ and $\mathbf{h}_2$, respectively. The more similar the images of the input hashes, the smaller the L_2 norm. If the L_2 norm is smaller than a pre-defined threshold T_d, the two images are considered as visually identical images. Otherwise, they are different images.

3 Experimental Results

We conduct experiments for illustrating the performances of our hashing. Comparisons with the existing algorithms will be given in Section 4. In the following experiments, all images are resized to 512×512, and blurred by a 3×3 Gaussian low-pass mask with a unit standard deviation. Total number of the used AC coefficients is 30. In other words, M=512, σ=1 and n=30.

3.1 Perceptual Robustness

Five standard color images sized 512×512 are taken as test images and attacked by different digital operations, which are achieved by Photoshop, MATLAB and StirMark 4.0 [12]. Detailed operations and their parameter values are summarized in Table 1. As rotation will expand the sizes of the processed images, only the 361×361 central parts of the original and the processed images are taken for hash generation. After the digital operations, each image has 60 attacked versions.

Table 1. The used operations and their parameter values

Tool	Operation	Description	Parameter value
Photoshop	Brightness adjustment	Photoshop's scale	10, 20, −10, −20
Photoshop	Contrast adjustment	Photoshop's scale	10, 20, −10, −20
MATLAB	Gamma correction	γ	0.75, 0.9, 1.1, 1.25
MATLAB	3×3 Gaussian low-pass filtering	Standard deviation	0.3, 0.4, ..., 1.0
StirMark	JPEG compression	Quality factor	30, 40, ..., 100
StirMark	Watermark embedding	Strength	10, 20, ..., 100
StirMark	Scaling	Ratio	0.5, 0.75, 0.9, 1.1, 1.5, 2.0
StirMark	Rotation	Angle in degree	1,2,5,10,15,30,45,90, −1,−2,−5,−10,−15, −30, −45, −90

Extract image hashes of the original and the attacked images, and calculate their similarities by the Formula (9). The maximum, minimum and mean L_2 norms of each operation and its standard deviation are presented in Table 2. It is observed that most L_2 norms of the test operations are all smaller than 14000 except a few results. This means that we can choose T_d=14000 as a threshold to resist most of the above operations. In this case, 2.67% attacked images are falsely judged as different images. If T_d reaches 16000, there are only 1.33% attacked images considered as different images.

Table 2. Maximum, minimum, and mean L_2 norms of different operations and their standard deviations

Operation	Max.	Min.	Mean	Standard deviation
Brightness adjustment	12869	2510	6176	3023.2
Contrast adjustment	16498	3927	8805	3914.1
Gamma correction	13102	2855	6622	3313.6
3×3 Gaussian low-pass filtering	880	2	397	268.5
JPEG compression	11948	3094	6459	2534.2
Watermark embedding	14421	751	5772	3449.3
Scaling	10237	3219	5771	2418.3
Rotation	19708	1802	6939	3933.0

3.2 Discriminative Capability

To construct a color image database for discrimination test, we downloaded 67 images from the picture channel of the SOHU.com, captured 33 images by digital

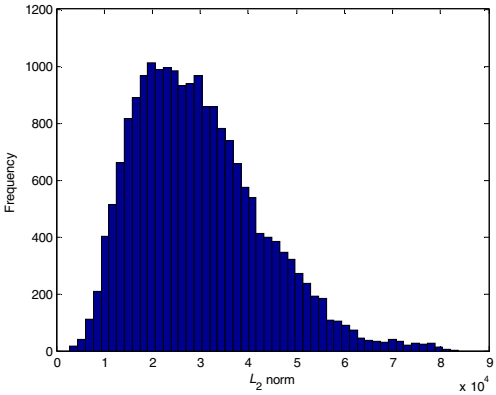

Fig. 3. L_2 norm distribution between hashes of different images

cameras, and took 100 images from the Ground Truth Database [13]. These image sizes range from 256×256 to 2048×1536. Calculate image hashes of the 200 color images and compute L_2 norm between each pair of hashes. Thus, 19900 results are available. Fig. 3 is the distribution of the results, where the x-axis is the L_2 norm and the y-axis represents the frequency of L_2 norm. It is observed that the maximum, minimum, and mean distances are 83665, 2883, and 29583, respectively, and the standard deviation is 13118.4. When T_d=14000, 9.37% different images are falsely judged as similar images. If T_d=16000, there are 14.37% different images considered as visually identical images.

4 Performance Comparisons

In this section, we compare our algorithm with some notable hashing methods, i.e., the SVD-SVD hashing [6], the NMF-NMF-SQ hashing [7], and the RT-DCT hashing [8]. To make fair comparisons, the same images used in Section 3 are also adopted to validate their perceptual robustness and discriminative capability. Since the input images of these algorithms [6, 7, 8] are gray images, we convert the RGB color images into YCbCr color space and take their luminance components for hash generation. The parameters used in the SVD-SVD hashing are as follows: the first number of overlapping rectangles is 100, rectangle size is 64×64, the second number of overlapping rectangles is 20 and the rectangle size is 40×40. For the NMF-NMF-SQ hashing, the used parameter values are: sub-image number is 80, height and width of sub-images are 64, rank of the first NMF is 2 and rank of the second NMF is 1. The SVD-SVD hashing and the NMF-NMF-SQ hashing both use L_2 norm as distance, while the RT-DCT hashing takes normalized Hamming distance as metric.

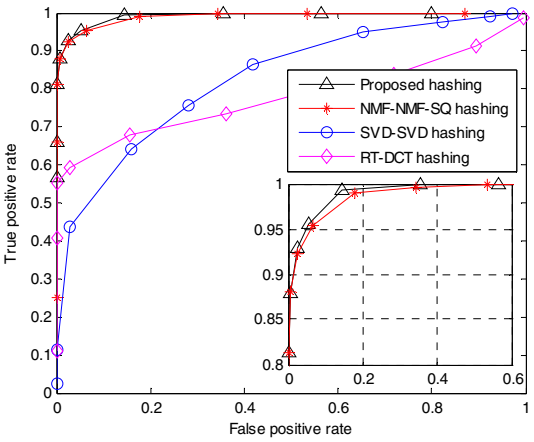

Fig. 4. ROC curve comparisons among different hashing methods

Exploit the assessed methods to generate image hashes and calculate their hash distances between each pair of images. ROC graph [14] is then exploited for comparing classification performances. We choose thresholds for each hashing, calculate the true positive rate (TPR) and false positive rate (FPR), and arrive at the ROC curves. Fig. 4 is the ROC graph comparisons among our hashing and the above three methods. We observe that ROC performances of our hashing are slightly better than those of the NMF-NMF-SQ hashing, and both the algorithms are much better than other hashing methods. For example, when FPR is near 0, TPRs of our hashing and the NMF-NMF-SQ hashing are both 0.82 and those of the SVD-SVD hashing and RT-DCT hashing are about 0.11 and 0.55, respectively. Similarity, when TPR reaches 1.0, optimal FPR of our hashing is about 0.36, and those of the NMF-NMF-SQ hashing, the SVD-SVD hashing and the RT-DCT hashing is approximately 0.53, 0.95 and 0.98.

5 Conclusions

In this paper, we have proposed a robust image hashing algorithm based on the histogram of color vector angles. As color vector angle is effective in measuring hue and saturation differences, our hashing has a good discriminative capability. Since pixels in the inscribed circle are not changed by rotation, the histograms of their color vector angles are almost the same before and after rotation, which makes our hashing resistant to image rotation with arbitrary angle. Experiments have demonstrated that our hashing is robust against normal digital operations, such as JPEG compression, watermarking embedding, scaling, rotation, brightness adjustment, contrast adjustment, gamma correction and Gaussian low-pass filtering. ROC curve comparisons between our hashing and three notable algorithms are carried out, and the results have shown that our hashing has better performances than the compared algorithms.

Acknowledgements. This work is supported in part by the Australian Research Council (ARC) under large grant DP0985456; the China “1000-Plan” National Distinguished Professorship; the China 863 Program under grant 2012AA011005; the China 973 Program under grant 2013CB329404; the Natural Science Foundation of China under grants 61165009, 60963008, 61170131; the Guangxi Natural Science Foundation under grants 2012GXNSFGA060004, 2012GXNSFBA053166, 2011GXNSFD018026, 0832104; the Guangxi “Bagui” Teams for Innovation and Research; the Project of the Education Administration of Guangxi under grant 200911MS55; and the Scientific Research and Technological Development Program of Guangxi under grant 10123005–8.

References

1. Qin, C., Chang, C.C., Chen, P.Y.: Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism. Signal Processing 92, 1137–1150 (2012)
2. Venkatesan, R., Koon, S.-M., Jakubowski, M.H., Moulin, P.: Robust image hashing. In: 7th IEEE International Conference on Image Processing, pp. 664–666. IEEE Press, New York (2000)

3. Lin, C.Y., Chang, S.F.: A robust image authentication system distinguishing JPEG compression from malicious manipulation. IEEE Transactions on Circuits System and Video Technology 11, 153–168 (2001)
4. Lefebvre, F., Macq, B., Legat, J.-D.: RASH: Radon soft hash algorithm. In: 11th European Signal Processing Conference, pp. 299–302 (2002)
5. Swaminathan, A., Mao, Y., Wu, M.: Robust and secure image hashing. IEEE Transactions on Information Forensics and Security 1, 215–230 (2006)
6. Kozat, S.S., Venkatesan, R., Mihcak, M.K.: Robust perceptual image hashing via matrix invariants. In: 11th IEEE International Conference on Image Processing, pp. 3443–3446. IEEE Press, New York (2004)
7. Monga, V., Mihcak, M.K.: Robust and secure image hashing via non-negative matrix factorizations. IEEE Transactions on Information Forensics and Security 2, 376–390 (2007)
8. Ou, Y., Rhee, K.H.: A key-dependent secure image hashing scheme by using Radon transform. In: IEEE International Symposium on Intelligent Signal Processing and Communication Systems, pp. 595–598. IEEE Press, New York (2009)
9. Tang, Z., Wang, S., Zhang, X., Wei, W.: Structural feature-based image hashing and similarity metric for tampering detection. Fundamenta Informaticae 106, 75–91 (2011)
10. Dony, R.D., Wesolkowski, S.: Edge detection on color images using RGB vector angles. In: IEEE Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 687–692. IEEE Press, New York (1999)
11. Kim, N.W., Kim, T.Y., Choi, J.S.: Edge-Based Spatial Descriptor for Content-Based Image Retrieval. In: Leow, W.-K., Lew, M., Chua, T.-S., Ma, W.-Y., Chaisorn, L., Bakker, E.M. (eds.) CIVR 2005. LNCS, vol. 3568, pp. 454–464. Springer, Heidelberg (2005)
12. Petitcolas, F.A.P.: Watermarking schemes evaluation. IEEE Signal Processing Magazine 17, 58–64 (2000)
13. Ground Truth Database, `http://www.cs.washington.edu/research/imagedatabase/groundtruth/`
14. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874 (2006)