Perceptual Image Hashing Using Block Truncation Coding and Local Binary Pattern

Xueqin Chen, Chuan Qin and Ping Ji
University of Shanghai for Science and Technology, Shanghai, China
E-mail: cxqmm0818@163.com, qin@usst.edu.cn, chuckping@163.com Tel: +86-21-55272562

***Abstract*— In this paper, we propose a novel image hashing scheme based on block truncation coding (BTC) and local binary pattern (LBP), which can be applied in image authentication and retrieval. In the proposed scheme, the pre-processing is first conducted on input image by bilinear interpolation, Gaussian low pass filtering, and singular value decomposition (SVD) to construct a secondary image for regularization. Then, BTC is applied on the secondary image to obtain the high/low quantized levels and the corresponding binary map that can reflect the contents of image. The concatenated image feature sequence is generated with the assist of the center-symmetrical local binary pattern (CSLBP). Finally, data dimensionality reduction is exploited on the image feature sequence to produce the part of hash. Combined with high/low quantized-levels, the final hash can be obtained. Experimental results show that the proposed scheme has the satisfactory performances of robustness, anti-collision, and security.**

I. INTRODUCTION

With the widespread application of image processing tools, image contents can be easily tampered. Therefore, verifying image authenticity becomes a principal issue in many actual aspects. As a result, image hashing as a new technology in multimedia security emerged and attracted many researchers' attention in the past decades [1-2].

Image hashing maps an input image to a short string for representing image contents, which has been widely applied in many fields, such as copy detection, image authentication and retrieval, tamper detection, and digital forensics. Classical cryptographic hash functions, such as MD5 and SHA-1, can compress input data into a short string, but they are so sensitive to slight changes that not suitable for digital images. In general, an image hashing scheme should satisfy three requirements: (1) perceptual robustness: visually similar images should have the same hash; (2) anti-collision capability: visually different images should have distinct hashes; (3) security: image hashing can't be estimated without the knowledge of secret key.

Recently, many image hashing schemes have been reported. Observing that the dominant discrete cosine transform (DCT) coefficient can represent the image feature, Tang *et al.* exploited dominant DCT coefficient to construct robust hash against digital operation [1]. The length of hash is too short. Thus, its anti-collision capability is pretty good but robustness is a little bad. As for discrete wavelet transform (DWT), Venkatesen *et al*. firstly proposed to extract statistics of wavelet coefficients through DWT to generate hash [2]. The method was resilient to JPEG compression, but it was sensitive to contrast adjustment. Qin *et al.* introduced another scheme where a non-uniform sampling was performed to extract image features after applying discrete Fourier transform (DFT) [3]. It was resistant to small angle rotations. Kozat *et al.* observed matrix variants and proposed a method which was reapplied to use SVD to calculate hashes [4]. This hashing method was robust to rotation at the cost of increasing misclassification. Choi and Park used hierarchical histogram which meant moderate specific histogram bins to generate image hash strings [5]. Its robustness resisting rotation was quite good, but this algorithm was sensitive to scaling. Davarzani *et al*. applied SVD and CSLBP into hashing [6], and produced robust feature with sacrificing the length of hashes. Non-negative matrix factorization (NMF) was widely widespread in image authentication area, Monga and Mihcak used NMF twice through extracting relevant coefficients to generate hashes [7]. Another classical method [8] proposed by Fridrich and Goljan was adopted in digital watermarking. Although its image feature was selected from front left DCT coefficients, the robustness and anti-collision capability should be improved much better.

The above reported methods still have some problems. For example, classification performances of many algorithms [2, 4, 7, 8] are not good enough. In this work, we propose an image hashing scheme using BTC and LBP, which can resist content-based operations effectively and achieves low anti-collision capability and better classification performances.

II. PROPOSED IMAGE HASHING SCHEME

Our image hashing scheme mainly includes three stages, i.e., pre-processing, feature extraction, and hash generation. The flowchart of our scheme is given in Fig. 1.

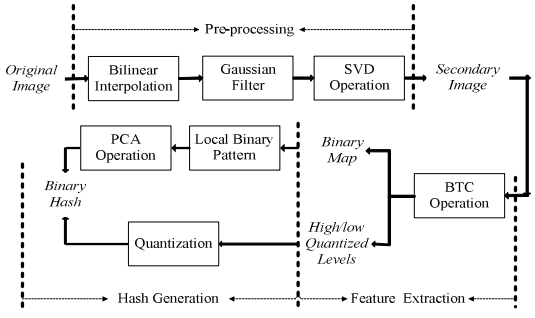


Fig. 1 Flowchart of the proposed scheme

A. Pre-processing

In order to obtain fixed length of hash, input image is firstly modified to **I** sized $M\times M$ by bilinear interpolation. Gaussian low-pass filtering is then applied on **I** to improve robustness of the scheme. The filtered image is divided into a series of non-overlapping blocks sized $k_1\times k_1$, and the SVD operation is conducted on each block to enhance attack resistance, see (1).

$$\mathbf{B}_i = \mathbf{S}_i\mathbf{V}_i\mathbf{D}'_i, \quad 1\le i\le N_1, \tag{1}$$

where $\mathbf{B}_i$ is denoted as the i-th block in the image and N_1 is the number of blocks. We denote that $\mathbf{V}_i^{\mathrm{P}}$ is the first ten singular vectors of $\mathbf{V}_i$, and $\mathbf{D}_i^{\mathrm{P}'}$ is the first ten singular values of orthogonal matrix $\mathbf{D}_i$'. Next, we select $\mathbf{S}_i$ sized $k_1\times k_1$, $\mathbf{V}_i^{\mathrm{P}}$ sized $k_1\times 10$ and $\mathbf{D}_i^{\mathrm{P}'}$ sized $10\times k_1$ to construct a secondary block $\mathbf{G}_i$, see (2).

$$\mathbf{G}_i = \mathbf{S}_i\mathbf{V}_i^{\mathrm{P}}\mathbf{D}_i^{\mathrm{P}}{}', \quad 1\le i\le N_1, \tag{2}$$

After all blocks $\mathbf{G}_i$ are collected, the secondary image $\mathbf{I}_1$ can be acquired.

B. Perceptual feature extraction

In this stage, the principle image features are extracted from $\mathbf{I}_1$ with the assist of BTC and LBP.

1) Feature of reconstructed levels

Let $\mathbf{I}_1$ be divided into a series of blocks sized $k_2\times k_2$. Through BTC algorithm [9], a binary map, whose element is zero or one corresponding to the pixel smaller or not smaller than $\overline{x}$ (i.e., mean value in each $k_2\times k_2$ block), is generated, and high/low reconstructed quantized-level (i.e., b and a) for each block can be calculated. Thus, for the whole image, the binary map sized $M\times M$ can be obtained and two quantized-level matrices (i.e., **H** and **L**) both sized $M/k_2\times M/k_2$ can also be obtained, see (3-4).

$$\mathbf{H} = \begin{bmatrix} b_{(1,1)} & b_{(1,2)} & \cdots & b_{(1,M/k_2)} \\ \cdots & \cdots & \cdots & \cdots \\ b_{(M/k_2,1)} & b_{(M/k_2,2)} & \cdots & b_{(M/k_2,M/k_2)} \end{bmatrix}, \tag{3}$$

$$\mathbf{L} = \begin{bmatrix} a_{(1,1)} & a_{(1,2)} & \cdots & a_{(1,M/k_2)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{(M/k_2,1)} & a_{(M/k_2,2)} & \cdots & a_{(M/k_2,M/k_2)} \end{bmatrix}, \tag{4}$$

where $b_{(i,j)}$ and $a_{(i,j)}$ are the high and low reconstructed quantized-levels of corresponding image block.

2) Feature of binary map

The CSLBP algorithm [10] is conducted on the binary map sized $M\times M$. Each pixel in the binary map is compared with its center-symmetrical pixels. It can be seen in (5-6), for a given pixel p_c, its value of CSLBP can be calculated.

$$\mathrm{CSLBP}_{(R,T_1,N_2)}(p_c) = \sum_{i=0}^{N_2/2-1} \mathrm{s}(p_i - p_{i+N_2/2})\times 2^i, \tag{5}$$

$$\mathrm{s}(x) = \begin{cases} 1, & x > T_1, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where T_1 is regarded as a threshold value to increase the robustness of CSLBP features, and N_2 is the number of pixels in circular neighborhood around the center pixel p_c and determined by the radius R of the circle. After conducting the above operation on each pixel of the binary map, a feature matrix $\mathbf{I}_2$ with the same size as the binary map is generated from CSLBP operator.

C. Hash generation

As mentioned above, we obtain three feature matrices, two reconstructed level matrices **H** and **L**, and a binary feature matrix $\mathbf{I}_2$, respectively.

First, **H** and **L** are both divided into blocks sized $k_3\times k_3$. For each block in both **H** and **L**, their mean values are collected to form two sets, i.e., $\mathbf{H}_{\mathrm{mean}}$ and $\mathbf{L}_{\mathrm{mean}}$, both including $M^2/k_2^2/k_3^2$ elements. Then, with a parameter ψ, each element in $\mathbf{H}_{\mathrm{mean}}$ and $\mathbf{L}_{\mathrm{mean}}$ is quantized by (7-8).

$$\mathbf{H}_1 = \Phi(\lfloor \mathbf{H}_{\mathrm{mean}}/\psi \rfloor), \tag{7}$$

$$\mathbf{L}_1 = \Phi(\lfloor \mathbf{L}_{\mathrm{mean}}/\psi \rfloor), \tag{8}$$

where the function $\Phi(\cdot)$ can convert decimal numbers in the input matrix into a one-dimensional binary sequence through quantization. Thus, each element in $\mathbf{H}_{\mathrm{mean}}$ and $\mathbf{L}_{\mathrm{mean}}$ is converted into $8-\lfloor\log_2\psi\rfloor$ bits, and there are totally $(8-\lfloor\log_2\psi\rfloor)\times M^2/k_2^2/k_3^2$ bits in both $\mathbf{H}_1$ and $\mathbf{L}_1$.

Then, we reshape $\mathbf{I}_2$ sized $M\times M$ into a new matrix $\mathbf{I}_3$ sized $M/8\times 8M$. However, data volume of $\mathbf{I}_3$ is so huge that it should be compressed into a new tiny feature matrix. Thus, the technique of principal component analysis (PCA) is applied on $\mathbf{I}_3$. The covariance matrix **C** sized $8M\times 8M$ of $\mathbf{I}_3$ consists of each covariance between each two samples in the total $8M$ samples. According to (9), the diagonal matrix **W** including the eigenvalues of **C** and the eigenvectors matrix **U** of **C** can be generated with matrix decomposition.

$$\mathbf{C} = \mathbf{U}\times\mathbf{W}\times\mathbf{U}^{-1}. \tag{9}$$

We take out the N_3 eigenvectors in **U** corresponding to the first N_3 largest eigenvalues in **W** as the eigenvector matrix **F** sized $8M\times N_3$. Finally, the new data set $\mathbf{I}_4$ sized $(M/8)\times N_3$ can be obtained by (10).

$$\mathbf{I}_4 = \mathbf{I}_3\times\mathbf{F}. \tag{10}$$

Here we call $\mathbf{I}_4$ as the principal components containing main feature of $\mathbf{I}_3$. Then, $\mathbf{I}_4$ is divided into non-overlapping blocks sized $k_4\times k_4$. The average value $a(i)$ of each block is compared with the mean value t of matrix $\mathbf{I}_4$ to obtain the feature sequence of binary map, i.e., $\mathbf{H}_2$, which has $(M/8)/k_4\times(N_3/k_4)$ bits, see (11).

$$\mathbf{H}_2(i)=\begin{cases}1, & a(i)\le t,\\ 0, & a(i)>t,\end{cases} \quad i=1,2,...,(M/8)/k_4\times(N_3/k_4). \tag{11}$$

As mentioned above, we obtain three feature sequences, i.e., $\mathbf{H}_1$ and $\mathbf{L}_1$ for high and low quantized levels, and $\mathbf{H}_2$ for binary map, respectively. We concatenate $\mathbf{H}_2$ with $\mathbf{H}_1$ and $\mathbf{L}_1$ to produce a binary sequence $\mathbf{Z}$, see (12).

$$\mathbf{Z}=[\mathbf{H}_1,\mathbf{H}_2,\mathbf{L}_1]. \tag{12}$$

The length of $\mathbf{Z}$ is the total length of $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{L}_1$. At last, a security key is utilized to generate the final hash sequence through scrambling the order of the elements in $\mathbf{Z}$. The length of the final image hash is $2\times(8-\lfloor\log_2\psi\rfloor)\times M^2/k_2^2/k_3^2+(M/8)/k_4\times(N_3/k_4)$ bits.

III. Experimental Results and Analysis

Experiments were conducted to test the performances of the proposed scheme with respect to robustness, uniqueness, and key-dependent security. The normalized Hamming distance was adopted to measure the similarity between two hashes:

$$\mathrm{Dist}(\mathbf{Z}_1,\mathbf{Z}_2)=\frac{1}{L}\sum_{i=1}^{L}|\mathbf{Z}_1(i)-\mathbf{Z}_2(i)|, \tag{13}$$

where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are two hash sequences, L is the hash length, and $\mathbf{Z}_1(i)$ and $\mathbf{Z}_2(i)$ denotes the i-th bit in $\mathbf{Z}_1$ and $\mathbf{Z}_2$, respectively.

In the experiments, the parameters of M, k_1, k_2, N_1, T_1, N_2, N_3, k_3, k_4, and ψ in our scheme were set to 256, 16, 4, 256, 0, 8, 16, 8, 4, and 10, respectively. Consequently, the length of $\mathbf{H}_1$, $\mathbf{L}_1$ and $\mathbf{H}_2$ are 320 bits, 320 bits and 32 bits, and the final hash has 672 bits.

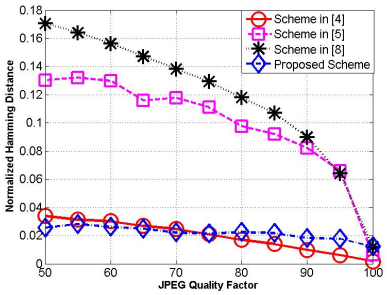
A. Perceptual robustness

Common content-preserving manipulations, i.e., JPEG compression, Gaussian filtering, average filtering and rotation, are tested on twenty standard images. We compared our method with three classical image hashing methods [4, 5, 8], see Fig. 2 (a-d). The ordinate is the average value of the normalized Hamming distances between the two hash pairs of the original image and its attacked version. It can be found that, in general, the average Hamming distance of our scheme against these content-preserving operations are smaller than those of the schemes [4, 5, 8]. In other words, the robustness of our scheme is superior to [4, 5, 8].

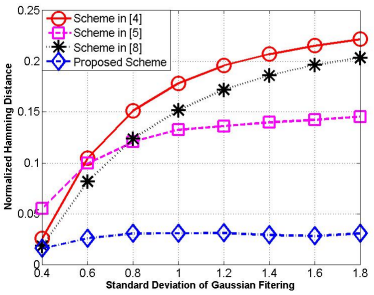
B. Uniqueness

We used the uncompressed color image database (UCID) [11] to evaluate the anti-collision performance. The UCID database contains 1338 various color images. First, we generated 1338 hashes and calculated 894453 normalized Hamming distances between hash pairs of different images. The histogram of the normalized Hamming distance is shown in Fig. 3. By normal distribution estimation, we find that the distribution of normalized Hamming distance obeys a normalized distribution with its mean μ = 0.46 and standard variation σ = 0.046. In our experiments, the threshold T_2 was used to justify the similarity of two images. The two images are visually similar when the normalized Hamming distance is smaller than T_2 and vice versa. Once T_2 is determined, the collision probability can be calculated by (14):

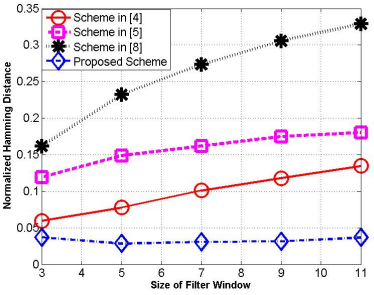
$$\begin{aligned}\mathrm{G}_c(T_2)&=\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{T_2}\exp[-\frac{(x-\mu)^2}{2\sigma^2}]\mathrm{d}x\\ &=\frac{1}{2}\mathrm{erfc}(-\frac{(T_2-\mu)^2}{2\sigma^2}).\end{aligned} \tag{14}$$



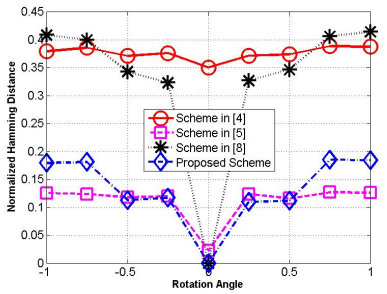
(a) JPEG Compression



(b) Gaussian Filtering



(c) Average Filtering



(d) Rotation

Fig. 2 Results of robustness performances.

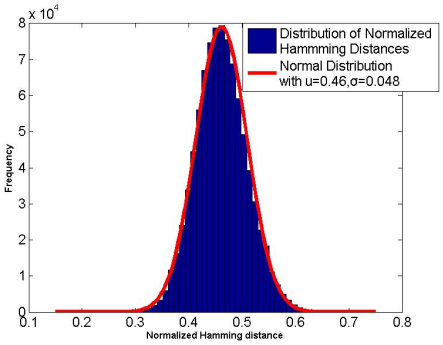


Fig. 3 Distribution of normalized Hamming distance between 894453 hash pairs of 1338 different images

TABLE I
COLLISION PROBABILITY FOR DIFFERENT THRESHOLDS T_2

Threshold T_2	Collision probability
0.26	1.5412×10^{-5}
0.24	2.2805×10^{-6}
0.22	2.2855×10^{-7}
0.20	3.0222×10^{-8}
0.18	2.7022×10^{-9}
0.16	2.0399×10^{-10}
0.14	1.2995×10^{-11}
0.12	6.9822×10^{-13}
0.10	3.1630×10^{-14}

TABLE I lists the collision probabilities with the different thresholds T_2. We can observe that, the smaller T_2 is set, the smaller the collision probability, and, obviously, the worse robustness is. It can be found from Fig. 2 that, the normalized Hamming distance against four common content-preserving operations are almost below 0.2. Thus, in order to achieve the satisfactory robustness and uniqueness simultaneously, we set the threshold T_2 to 0.2.

C. *Key-dependent security*

As Section II described, the proposed method has one secret key. Different secret keys produce distinct hashes. As shown in Fig. 4, the abscissa is the index of 1000 wrong keys and the ordinate is the average value of the normalized Hamming distance between the hash pairs of images obtained by the correct and wrong secret keys. We can observe that, the ordinates are in the vicinity of 0.5, thus, it is extremely difficult for attacker to produce or estimate the same hash without the correct key.

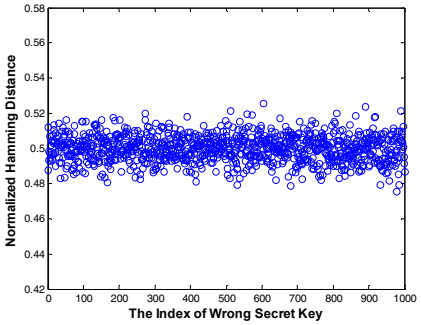


Fig. 4 Normalized Hamming distance between hash pairs using the correct and wrong secret keys.

IV. CONCLUSIONS

In this work, a novel image hashing scheme based on BTC and CSLBP is proposed, which can be applied in image retrieval and authentication. To make final hash more robust, Gaussian low-pass filtering and SVD are applied to construct the secondary image. BTC is conducted on secondary image to obtain the high/low quantized levels and corresponding binary map. Image feature of binary map is obtained with the assist of CSLBP and PCA operations. The final hashing can be produced based on three main feature matrices. Experimental results show the proposed scheme has better robustness than the reported schemes. Additionally, the satisfactory performances for uniqueness and security of our scheme can also be achieved.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (61303203), the Natural Science Foundation of Shanghai, China (13ZR1428400), and the Innovation Program of Shanghai Municipal Education Commission (14YZ087). Corresponding author: Chuan Qin, Email: qin@usst.edu.cn.

REFERENCES

[1] Z. J. Tang, F. Yang, L. Y. Huang, and X. Q. Zhang, "Robust image hashing with dominant DCT coefficients," *Optik - International Journal for Light and Electron Optics*, 2014, vol. 125, pp. 5102-5107.
[2] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," *Proceeding of IEEE International Conference on Image Processing*, 2000, pp. 664-666.
[3] C. Qin, C. C. Chang, and P. L. Tsou, "Robust image hashing using non-uniform sampling in discrete Fourier domain," *Digital Signal Processing*, 2013, vol. 23, no.2, pp. 578-585.
[4] S. S. Kozat, R. Venkatesan, and M. K. Mihcak, "Robust perceptual image hashing via matrix invariants," *Proceeding of IEEE International Conference on Image Processing*, 2004, vol. 5, pp. 3443-3446.
[5] Y. S. Choi and J. H. Park, "Image hash generation method using hierarchical histogram," *Multimedia Tools Application*, 2011, vol. 61, no. 1, pp. 181-194.
[6] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools Application*, 2015, pp. 1-29.
[7] V. Monga and M. K. Mihcak, "Robust and secure image hashing via non-negative matrix factorizations," *IEEE Transactions on Information Forensics and Security*, 2007, vol.2, no.3, pp. 376-390.
[8] J. Fridrich and M. Goljan, "Robust hash function for digital watermarking," *Proceedings of International Conference on Information Technology: Coding and Computing,* 2000, pp. 173-178.
[9] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications*, 1977, vol. 27, no. 9, pp. 1335-1342.
[10] M. Heikkila, M. Pietikainen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, 2009, vol. 42, no. 3, pp. 425-436.
[11] G. Schaefer and M. Stich, "UCID – An uncompressed color image database," *Proceedings of SPIE in Storage and Retrieval Methods and Applications for Multimedia*, 2004, vol. 5307, pp. 472-480.