Compact Fingerprints for Image Authentication Based on PCA of DCT Coefficients *

Sheng Tang[1,2], Jin-Tao Li[1], and Yong-Dong Zhang[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2]Graduate School of the Chinese Academy of Sciences, Beijing, 100039, China

Abstract

We propose a novel image fingerprinting method for authentication in the DCT Domain, which is based on Principal Components Analysis (PCA) of the block DCT coefficients, and can be directly extended and applied to MPEG video authentication without DCT transforms. We also present an algorithm for location of tampered regions based on the method. Experiments on large-scale database show that the proposed method is fairly discriminative, robust against compression, and sensitive to malicious modifications. Additionally, since we take the quantized eigenvector matrix (10×10) as fingerprint, its length is the same 100 bytes for any image of arbitrary size.

Keywords: Image authentication; Fingerprinting; PCA; DCT coefficients.

1 Introduction

With the rapid growth of multimedia applications, protection of intellectual property is becoming more prominent. Although watermarking allows for verification of originality of the contents, it is not well suited for authentication [1, 4] due to its security and modification of contents. With respect to this, identifying content intrinsically based on its own features rather than some embedded identifiers can be an alternative solution. Fingerprints are perceptual features or short summaries of a multimedia object [1]. They can be used for identifying contents just as human fingerprints are used for identification. It is an emerging research area that is receiving increased attention.

Many applications of fingerprinting were detailed in [1–3]. A typical example is multimedia authentication, the key issue of which is to protect the content itself instead of the particular representation of the content without access to the original signals [3–5]. This renders traditional cryptographic schemes using bit-sensitive hash functions not applicable [1, 3, 8], for multimedia signals can be represented equivalently in different forms, and undergo various manipulations during distribution that may carry the same perceptual information. Therefore, fingerprints should be both discriminative and robust [1].

Researchers have paid great efforts on fingerprinting. Up to now, many image fingerprinting methods have been proposed [1–8]. While many existing methods perform well against compression, some of them can not capture distinguishing features of images, hence they ignore the discriminabilities of fingerprints between different images and are not secure because it is easy to modify images without changing their fingerprints such as block-histogram-based method in [4], and block-mean-based method in [6], and moment features in [5], or they are inadequate to detect some modifications inside the images such as salient-feature-point-based method in [8]. Additionally, some fingerprints are not short [4, 5], and the fingerprint lengthes depend on image sizes [4–7]. Recently, a compact and robust fingerprinting method based on radon transform has been proposed in [1]. But it is not intended for authentication, and does not address how to locate tampered regions. Additionally, it is not based on the DCT domain, hence can not directly extended to compressed video streams. As is pointed out in [3], the major challenge in content-based authentication is to define a computable feature vector that can capture the major content characteristics from a human perspective. For multimedia fingerprinting, extracting features that allow direct access to the relevant distinguishing information is crucial [1].

To achieve the above goals, we present a novel fingerprinting scheme in the DCT domain. Our strategy is to use the DC and low-frequency terms of block DCT coefficients as the distinguishing features of images, and apply PCA to the features, finally, take the quantized eigenvector matrix as fingerprint. Based on the new scheme, we also present an algorithm for locating tampered regions. The rest of the paper is organized as follows. Section 2 and 3 describes fingerprint generation and matching. Section 4 addresses how to locate tampering. Section 5 reports experimental results. Section 6 summarizes our contributions.

*This work was supported by National Nature Science Foundation of China (60473002) and Beijing Science and Technology Planning Program of China (Z0004024040231).

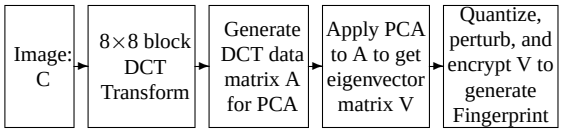

Figure 1: Procedure for generating fingerprint

2 Fingerprint generation

The procedure for generating fingerprint is diagramed in Fig.1. First, we transform the image C into 8×8 block-DCT domain. Then, we prepare data matrix A for PCA: divide the DC and 7 low-frequency AC terms (as shown in Fig.10.10 in [9]) by the corresponding values of the quantization matrix used in JPEG, and place the 8 quantized coefficients, and the row and column indices of each block into the N×10 matrix A in row or column order, where N is the total number of blocks. This can be represented as:

$$A = \begin{pmatrix} D_{11}\,,\, D_{12}\,,\cdots,\, D_{18}\,,\, I_1\,,\, J_1 \\ D_{21}\,,\, D_{22}\,,\cdots,\, D_{28}\,,\, I_2\,,\, J_2 \\ \cdots\cdots\cdots\cdots\cdots \\ D_{N1}, D_{N2}, \cdots, D_{N8}, I_N, J_N \end{pmatrix} \quad (1)$$

where D_{ij} denotes the j^{th} DCT coefficients of the i^{th} block of the image C, I_i and J_i are the row and column indices of the i^{th} key block for consideration of its spatial location. Since PCA ignores the order of the blocks, we take the I_i and J_i into consideration for more secure fingerprints.

Before PCA, for the purpose of achieving high speed, instead of using the covariance matrix of A adopted by conventional PCA algorithm, we adopt the standardized matrix B of A [10], i.e., standardize A by dividing each column by its standard deviation. The function *princomp()* in the Matlab Statistics Toolbox describes the PCA algorithm adopted by us.

Then, we apply PCA to the standardized matrix B, and use the resultant 10×10 eigenvector matrix V, i.e., the first output of the function *princomp()*, as fingerprint. To get more compact fingerprint, we quantize each element α of the eigenvector matrix V to an one-byte integer α_q according to:

$$\alpha_q = \lfloor 127(1+\alpha) + 0.5 \rfloor. \quad (2)$$

For image authentication, a secure fingerprint is required. So we use a private key to to perturb all the elements of the quantized fingerprint matrix in a random order, and encrypt the fingerprint for secure communication. The adoption of perturbation is to ensure that only the right key can generate the authentication fingerprint, i.e., the fingerprint function depends on the private key [1]. Therefore, different fingerprints generated with different keys do not match according to the fingerprint matching method in section 3 due to their disordered eigenvectors.

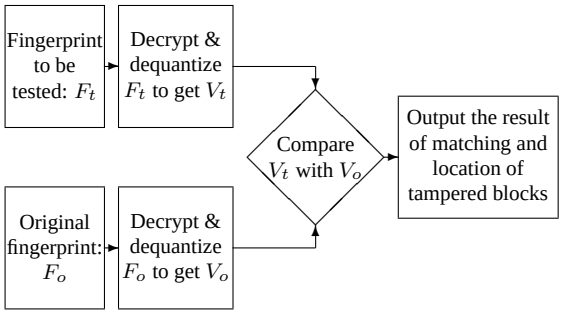

Figure 2: Procedure for fingerprint matching

3 Fingerprint matching

The procedure for fingerprint matching is diagramed in Fig.3. Two images are declared authentic if the similarity S between their fingerprints is above a certain threshold T, which can be determined by experiments can be determined or on user's demands according to various applications. The main idea of fingerprint matching is that if two images are considered authentic, corresponding eigenvectors from the two fingerprints should be high correlative. Thus, S can be calculated by computing correlation between corresponding pairs of eigenvectors, that is, the cosine of the angle between them. Let $V_o = (\alpha_{o1}, \alpha_{o2}, \ldots, \alpha_{o10})$ and $V_t = (\alpha_{t1}, \alpha_{t2}, \ldots, \alpha_{t10})$ be the eigenvector matrices dequantized from the fingerprints of the original image C_o and image C_t to be tested respectively. Since eigenvector matrix is orthogonal, S can be calculated by computing the arithmetic mean of correlations of all the pairs as:

$$S = \frac{1}{10}(\sum_{i=1}^{10} |\alpha'_{oi}\alpha_{ti}|) \quad (3)$$

where the "$'$" denotes the transpose operation.

4 Locating tampered regions

For image authentication, locating tampering is useful [3, 9]. Based on the novel fingerprint method, we present an algorithm for locating tampered regions if the calculated S between the images C_o and C_t is below the threshold T. The main idea is to employ Hotelling's T^2 statistic (HTS). By comparing the HTS values of two corresponding block DCT coefficients of the image C_o and C_t, we can easily determine which block is most possibly tampered. The algorithm is described as follows.

First, after PCA, according to [10], HTS value of the image C_t can be calculated from the $N \times 10$ standardized matrix B_t , 10×10 eigenvector matrix V_t, and 10×10 diagonal eigenvalue matrix λ_t as:

$$\mathrm{HTS_t} = |\tfrac{1}{\sqrt{\lambda_t}}(B_t V_t)'|' \quad (4)$$

Table 1: Means and Std of the measured S between 10000 images and corresponding JPEG images

JPEG Compression	Mean	Std
JPEG(Q=10%)	0.8997	0.0908
JPEG(Q=20%)	0.9590	0.0524
JPEG(Q=30%)	0.9712	0.0445
JPEG(Q=40%)	0.9727	0.0466
JPEG(Q=50%)	0.9641	0.0528
JPEG(Q=60%)	0.9855	0.0295
JPEG(Q=70%)	0.9873	0.0303
JPEG(Q=80%)	0.9944	0.0164
JPEG(Q=90%)	0.9944	0.0172

where the operator "||" returns a row vector of the Euclidian length of each column.

Then, replace V_t with V_o in (4) to estimate $\mathrm{HTS_o}$ of the original image C_o:

$$\mathrm{HTS_o} = |\tfrac{1}{\sqrt{\lambda_t}}(B_t V_o)'|'. \quad (5)$$

Finally, determine which block is the region most possibly tampered by computing the difference between the $\mathrm{HTS_t}$ with $\mathrm{HTS_o}$ according to:

$$(i,j) = argmax\{(\mathrm{HTS_t} - \mathrm{HTS_o})^2\} \quad (6)$$

where the returned (i,j) denotes the indices (or location) of the required block.

5 Experimental results

In evaluating our proposed method, we tested it on the well-known images "lena" (512×512) and "F14" (732×500), and 10000 test images randomly selected from the Corel Gallery database including many kinds of images (256×384 or 384×256). All the colour images are transformed into 8 bits/pixel gray level images. Before experiments, we first extracted fingerprints from all the 10000 images, and it took only about 260 seconds on the PC of Pentium IV 2.4G although we implemented the method with Matlab C++ Math Library, which shows the method is efficient.

5.1 Robustness Test

To test robustness of the method, we compressed the 10000 images to various JPEG images with different quality levels Q ranging from 10% to 90%, and calculated S between images and their corresponding JPEG images. The means and standard deviations (Std) of the measured S were shown in Table.1. It shows that our method is fairly robust against compression.

Other experiments show that our method can also resist additive noise attack to some extent. However, our method is not robust against geometric manipulations such as cropping and scaling and rotation.

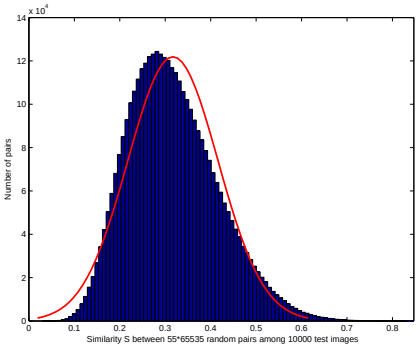

Figure 3: Histogram of the measured S between 3603875 pairs of images randomly selected from the 10000 images. The red line represents the ideal random i.i.d. case N(0.3170, 0.0993).

But they can be clearly specified by users [7].In our following experiments, we set the mean S=0.9590 of JPEG(Q=20%) as the threshold T for authentication.

5.2 Discriminability Test

We randomly selected 3603875 pairs of fingerprints from the 10000 images, and calculated S between each pair. As shown in Fig.3, all the measured S were within 0.0295 and 0.8470. The mean μ and Std σ were 0.3170 and 0.0993. As the histogram closely approaches the ideal random i.i.d. case N(μ, σ), we can conclude that the proposed method is pair-wise independent, and can calculate the false alarm rate $\mathrm{P_{FA}}$ (the probability that declare different images as authentic) according to:

$$\mathrm{P_{FA}} = \int_T^\infty \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx = \frac{1}{2} erfc(\frac{T-\mu}{\sqrt{2}\sigma}).$$

Substituting μ=0.3170, σ=0.0993, T=0.9590, we got very low $\mathrm{P_{FA}} = \mathrm{erfc}(4.5716)/2 = 5.0563 \times 10^{-11}$. It shows that our method is fairly discriminative.

5.3 Authentication test

We made small modifications of "F14" as shown in Fig.4(b). The measured S between the original and tampered images was 0.8484 ($< T$), so we successfully detected that the image was tampered, and located the tampering as shown in Fig.4(c). It shows our method is sensitive to malicious modifications.

5.4 Key dependence test

We perturbed the fingerprint matrix of the image "lena" in random orders generated by different seeds (keys) ranging from 0 to 65535. As shown in Fig.5, all the measured S were within 0.0599 and 0.5233. The mean and Std were 0.2539 and 0.0577, which shows the method is key dependent.

(a) (b) (c)

Figure 4: Authentication test: (a)original image (732×500); (b) tampered image (732×500) with copying char "E" on the empennage to front of the plane; (c) block-based HTS difference map (91×62) between (a) and (b), the highlight intensity is proportional to the possibility of being tampered.

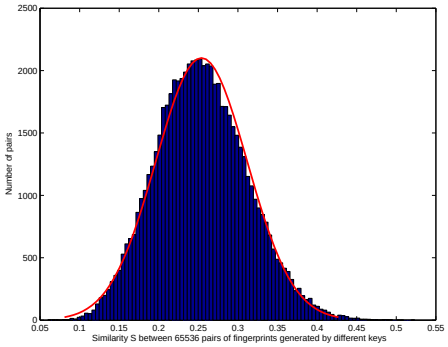

Figure 5: Histogram of the measured S between 65536 pairs of fingerprints randomly perturbed by random orders with different keys. The red line represents the ideal random i.i.d. case N(0.2539, 0.0577).

Compared with the current methods such as that in [7], the signature length of our method is very short (only 100 bytes long) and independent on image sizes. Since how to extract short, robust and invariant information is one of the two major challenges in developing authentication watermarks [2], it is of great importance to embed the short fingerprint into the middle-frequency terms of block DCT coefficients to avoid additional communication channel for authentication. Although the fingerprint length of the radon-transform-based method in [1] is very short, but it do not address the problem of how to locate tampered regions, and can not directly extended and applied to MPEG video for real-time processing.

6 Conclusion

In this paper, we present a compact image fingerprinting method for authentication based on PCA of block DCT coefficients. Experiments show that the proposed method is efficient, discriminative, robust against compression, key dependent, sensitive to malicious modifications and can locate tampered regions. It is convenient to extend our method to verify MPEG video streams without DCT transforms.

References

[1] J.S.Seoa, J.Haitsmab, T.Kalkerb, C.D.Yoo, "A robust image fingerprinting system using the Radon transform," *Signal Processing: Image Communication*, 19(4):325-339, April 2004.

[2] C.-Y. Lin and S.-F. Chang, "Robust digital signature for multimedia authentication", *Circuits and Systems Magazine, IEEE*, 3(4):23-26, 2003.

[3] B.B.Zhu, M.D.Swanson, A.H.Tewfik, "When seeing isn't believing [multimedia authentication technologies]", *Signal Processing Magazine, IEEE*, 21(2):40-49, March 2004.

[4] M. Schneider, S.-F. Chang, "A robust content based digital signature for image authentication", *In: Proceedings of IEEE ICIP 96, Lausanne, Switzerland*, Vol.3:227-230, October 1996.

[5] M.P. Queluz, "Authentication of digital images and video: generic models and a new contribution", *Signal Processing: Image Communication*, 16(5):461-475, Jan. 2001.

[6] Lou DC, Liu JL. "Fault resilient and compression tolerant digital signature for image authentication", *IEEE Transactions on Consumer Electronics*, 46(1):31-39, 2000.

[7] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation", *IEEE Trans. Circuits Syst. Video Technol.*, 11(2): 153-168, 2001.

[8] S. Bhattacharjee, M. Kutter, "Compression tolerant image authentication", *In: Proceedings of the IEEE ICIP 1998, Chicago, IL*, October 1998.

[9] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, NewYork: Morgan Kaufmann, 2001.

[10] J. Edward Jackson, *A User's Guide to Principal Components*, John Wiley & Sons, Inc., pp. 1-25, 1991.