

A NOVEL METHOD FOR DETECTING CROPPED AND RECOMPRESSED IMAGE BLOCK

WeiQi Luo[†], Zhenhua Qu[†], Jiwu Huang^{†*}, Guoping Qiu[‡]

[†]Guangdong Key Lab. of Information Security Technology
Sun Yat-Sen University, Guangdong, China, 510275

[‡]School of Computer Science, University of Nottingham, NG 8, 1BB, UK

ABSTRACT

One of the most common practices in image tampering involves cropping a patch from a source and pasting it onto a target. In this paper, we present a novel method for the detection of such tampering operations in JPEG images. The lossy JPEG compression introduces inherent blocking artifacts into the image and our method exploits such artifacts to serve as a ‘watermark’ for the detection of image tampering. We develop the blocking artifact characteristics matrix (BACM) and show that, for the original JPEG images, the BACM exhibits regular symmetrical shape; for images that are cropped from another JPEG image and re-saved as JPEG images, the regular symmetrical property of the BACM is destroyed. We fully exploit this property of the BACM and derive representation features from the BACM to train a support vector machine (SVM) classifier for recognizing whether an image is an original JPEG image or it has been cropped from another JPEG image and re-saved as a JPEG image. We present experiment results to show the efficacy of our method.

Index Terms— Digital Forensic, Blocking Artifacts, Block Artifact Characteristics Matrix

1. INTRODUCTION

With the advancement in image processing, tampering digital images without leaving obvious traces has become easier and easier. The problem of digital image counterfeiting is potentially very serious. It will get worse as counterfeiting techniques get more and more sophisticated. Authentication of digital images presents many challenges.

Digital watermarking has been proposed as a means to authenticate the contents of digital images. The watermark-based methods, however, must insert the watermark when creating the digital images, which would limit their applications. Furthermore, the security of watermark in terms of resisting hostile attacks needs to be strengthened.

Recently, several researchers have started to develop techniques for detecting various forms of digital image forgery

without using watermark. Fridrich et al presented methods for camera identification based on detecting the pattern noise of the sensor in the digital cameras [1], and the identification of double JPEG compressed images [2]. Swaminathan et al proposed some methods for non-intrusive component forensics [3, 4]. Farid and Popescu developed several statistical methods for detecting forgeries based on color filter interpolation [5], and re-sampling[6]. Ng and Chang proposed a model of image spicing for detecting photomontage [7], physics-based models for distinguishing computer graphics from nature photographs [8] and so on.

JPEG is a commonly used compression standard and has been widely used in the Internet and other applications. Detection of forgery in JPEG images can play an important role in countering image forgery. In [2], Lukas and Fridrich presented a method for the estimation of primary quantization matrix from a double compressed JPEG image. In [9], Popescu proposed a method for detecting and quantifying double compressed images. However the methods proposed in [2] and [9] can not determine whether a given JPEG image had been cropped and recompressed which always occurs in a composite or region-duplication image. So far, we are not aware of any research work that addresses this issue.

The organization of the paper is as follows. Section 2 we describes the tampering process we attempt to address. Section 3 presents the blocking artifact characteristics matrix (BACM), and 14 representation features derived from the BACM. Experimental results are shown in section 4 and concluding remarks are presented in section 5.

2. MODEL OF TAMPERING IN JPEG IMAGES

One of the most common types of digital image tampering is compositing in which two or more images are spliced together to create a composite image, as illustrated in Fig.1.

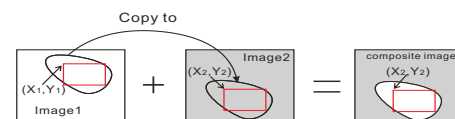


Fig. 1. Composite image

This work was supported by NSFC (60325208, 90604008, 60633030), 973 Program (2006CB303104), NSF of Guangdong (04205407).

*Contact Author (isshjw@mail.sysu.edu.cn)

With reference to Fig. 1, we assume that the first image $Image_1$ and the composite image are JPEG images with the quantization factors QF_1 and QF_2 respectively. (x_1, y_1) , (x_2, y_2) are the coordinates in the tampered region before/after tampering respectively, where $x_1 - x_2 \equiv m_1(mod 8)$, $y_1 - y_2 \equiv m_2(mod 8)$. In our method, we assume that m_1 and m_2 are not 0 or 4 at the same time. This is a very reasonable assumption: assuming that m_1, m_2 are uniform distribution in $[0, 7]$, then we have $p((m_1 = 0 \& m_2 = 0), (m_1 = 4 \& m_2 = 4)) = \frac{2}{8^2} = 3.125\%$

The tampered region in the composite image had been cropped and JPEG-recompressed, as shown in Fig.2.

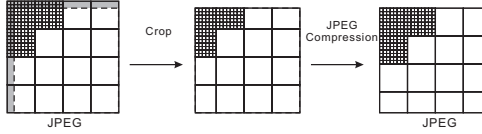


Fig. 2. Cropped and recompressed

The solution for detecting the composite region is then converted to that of identifying whether it contains a cropped and recompressed block in the image.

In this paper, we focus on the problem: given a JPEG image block, identify whether it has been cropped from another JPEG image and recompressed by JPEG compression.

3. PROPOSED METHOD

Our proposed method first analyses the process in JPEG compression, and then derives the blocking artifact characteristics matrix (BACM) to measure the symmetrical property of the blocking artifacts introduced by JPEG encoder, and finally, we train a SVM classifier using feature vectors derived from the BACMs before/after cropped-recompressed operations.

3.1. Detection of Blocking Effects

In the JPEG encoder, the image is first divided into small 8×8 non-overlapping blocks. Each block is DCT-transformed, quantized and then entropy encoded to yields a data stream.

One of simple and effective ideas for detection of JPEG block artifacts have been proposed in [10]. In [10], it assumes that if there is no compression the pixel differences across blocks should be similar to those within blocks.

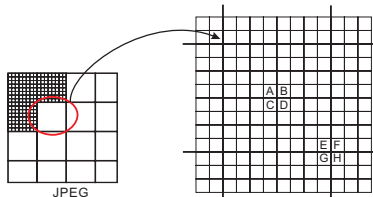


Fig. 3. Blocking artifacts detection

If the image is JPEG-compressed, the differences across blocks should be different due to block artifacts. As shown in Fig. 3, assume the block grid is known. We then calculate the differences within a block and spanning across a block boundary. For each block, we compute

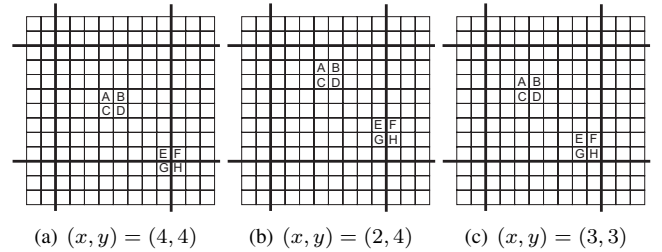
$$Z'_{(x,y)} = |A + D - B - C|, Z''_{(x,y)} = |E + H - F - G| \quad (1)$$

where $A \sim H$ are the values of the pixels in the position, and the (x, y) is the coordinate of A in each block. The coordinates of A to H in each block change according to the coordinate of A , as shown in Fig.4 (a) (b) and (c). For example, the coordinate of E : $P(E) = P(A) + (4, 4)$.

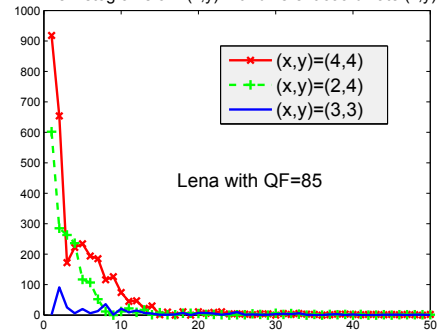
In Fig.4, we firstly compute the histograms H_I , H_{II} of $Z'_{(x,y)}$ and $Z''_{(x,y)}$ with $(x, y) = (4, 4)$, $(2, 4)$ and $(3, 3)$ respectively. Then the energy K of the difference between H_I and H_{II} with the value n is calculated as follows.

$$K_{(x,y)}(n) = |H_I(n) - H_{II}(n)| \quad (2)$$

Where $n \in [0, 255 \times 2]$, $H_I(n)$, $H_{II}(n)$ are the total number in the Z' and Z'' respectively with the lever n . Fig.4(d) shows the $K_{(x,y)}$ with different coordinates of A . We can observe that the differences are larger across a JPEG block boundary e.g. Fig.4(a) and (b). The biggest differences always occur when $P(A) = (4, 4)$, and when $(x = 4)$ or $(y = 4)$ the difference is also large. When the coordinates of A to D and E to H are all inside a block respectively, then the difference is small.



The histograms of $K_{(x,y)}$ with different coordinate (x,y)



(d) Three histograms of $K_{(x,y)}$

Fig. 4. Comparing the $K_{(x,y)}$ with the different coordinates of A in the Lena Image with QF 85

3.2. Symmetry from Blocking Artifacts

We first divide an image into non-overlapping 8×8 blocks. For each block, we compute $Z'_{(x,y)}$ and $Z''_{(x,y)}$ using Eq. (1). Then we compute the difference between the histograms via Eq.(2) and get the average of $K_{(x,y)}$, denoted as $M(x,y) = \frac{\sum K_{(x,y)}(n)}{255 \times 2 + 1}$, for each $1 \leq x, y \leq 8$. Lastly normalize the average matrix $M(x,y)$. The matrix $M(x,y)$ is called the blocking artifact characteristics matrix (BACM).

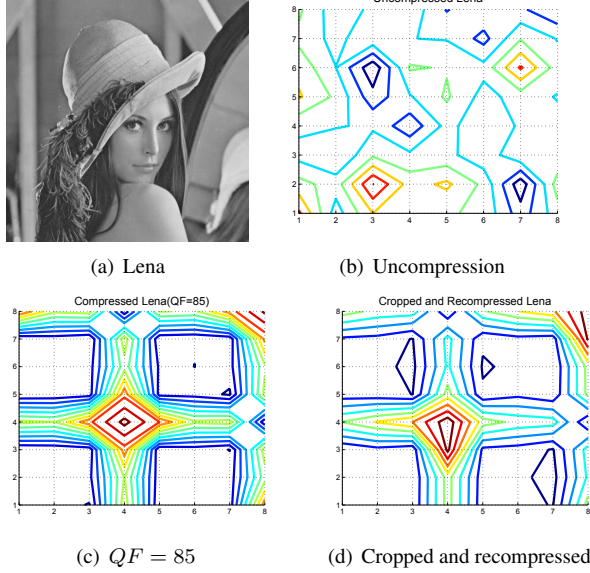


Fig. 5. Comparing the contour of $M(x,y)$ in uncompressed Lena (b), original JPEG image with quality factor 85 (c), original JPEG image with $QF_1 = 50$ had been cropped 2 rows and 3 columns and recompressed with $QF_2 = 85$ (d).

Fig. 5 shows the contour of $M(x,y)$ for uncompressed, JPEG compressed, and cropped and JPEG recompressed Lena images. From the figures above, we observe that the values of $M(x,y)$ in uncompressed image are random. While in the original JPEG image the values of $M(x,y)$ become regular. The max value occurs in $M(4,4)$ and the values in the 4th row and column are bigger. There are four flat regions around the center $M(4,4)$ as shown in (c). Fig5.(d) shows the contour of $M(x,y)$ in the cropped and recompressed image, the symmetry of the values of $M(x,y)$ obviously descends comparing with that in Fig.5(c).

Due to the nature of the JPEG image, we can get blocking artifacts, an inherent ‘semi-fragile watermark’ from the block-based compression. The ‘watermark’ measured using the symmetry of the BACM $M(x,y)$ would change after being cropped and recompressed. So the key idea of our method is to identify the differences between the BACMs as shown in (c) and (d) of Fig.5.

3.3. Feature Vector from BACM

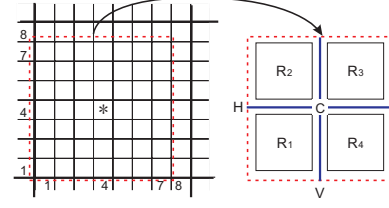


Fig. 6. the Symmetry of BACM

As shown in Fig 6, we first crop 7×7 block from the matrix $M(x,y)$, and then divide the block into 7 non-overlapping parts: region R_1, R_2, R_3, R_4 , the horizontal direction H , vertical direction V and the center point $M(4,4)$.

$$R_1 : \{M(1,1), M(1,2), M(1,3), M(2,1), M(2,2), M(2,3), M(3,1), M(3,2), M(3,3)\}$$

$$R_2 : \{M(1,5), M(1,6), M(1,7), M(2,5), M(2,6), M(2,7), M(3,5), M(3,6), M(3,7)\}$$

$$R_3 : \{M(5,5), M(5,6), M(5,7), M(6,5), M(6,6), M(6,7), M(7,5), M(7,6), M(7,7)\}$$

$$R_4 : \{M(5,1), M(5,2), M(5,3), M(6,1), M(6,2), M(6,3), M(7,1), M(7,2), M(7,3)\}$$

$$H : \{M(1,4), M(2,4), M(3,4), M(5,4), M(6,4), M(7,4)\}$$

$$V : \{M(4,1), M(4,2), M(4,3), M(4,5), M(4,6), M(4,7)\}$$

$$C : \{M(4,4)\}$$

We construct the following 14 features:

- 1) The first 2 features describe the symmetry of H and V around the center point C ;
- 2) Then the symmetry of the four flat region R_1, R_2, R_3 and R_4 around H, V and C are recorded ($C_4^2 = 6$ features);
- 3) Lastly the percentage of the center point C occupying the region R_1, R_2, R_3, R_4, V and H are recorded respectively (6 features).

Note that the symmetry feature is the sum of energy of differences between the values in the matrix. For example, the symmetry of H around C is: $|M(1,4) - M(7,4)| + |M(2,4) - M(6,4)| + |M(3,4) - M(5,4)|$, other features in 1) and 2) are derived similarly. For the features in 3), e.g. the percentage of the center point C occupying the region R_1 is defined as $\frac{M(4,4)}{S}$, where $S = \sum r, r \in R_1$.

4. EXPERIMENTAL RESULTS AND DISCUSSION

In our experiments, we first collect 1128 uncompressed images. The images are taken using Panasonic Lumix DMC-FZ30 with TIFF format including outdoor and indoor scenes with different camera setting. The maximal and minimal resolutions of the camera are 3264×2448 and 1280×960 . We have the 3 sizes (1600×1200 , 1280×960 and 640×480)

from cropping the original images for our experiments. The method for creating experimental data is as following:

1) Original JPEG images. For each TIFF image, we convert it to JPEG image with quality factor QF_2 .

2) Tampered JPEG images. For each TIFF image, we simulate the process as shown in Fig.2. Firstly convert it to JPEG image with a random quality factor QF_1 . Then crop the image randomly and resave it with QF_2 .

For a given quality factor QF_2 , we obtain 2256 JPEG images, and then divide them into two categories, one for training(including 500 original and 500 tampered JPEG images respectively) and the other for testing . We calculate the BACM $M(x, y)$ and obtain a feature vector for each JPEG image. The feature vectors are fed to SVM[11] and to train a classifier from the training data set to distinguish the original from the tampered images. Fig. 7 and Table 1 show the experimental results using the classifier on the testing data set for different quality factors and sizes.

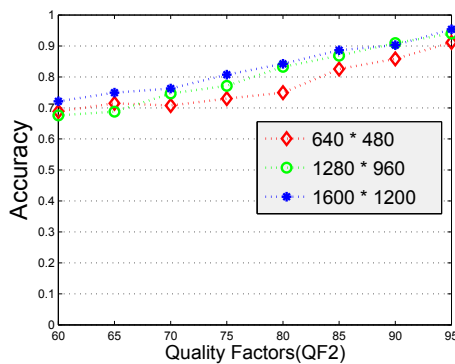


Fig. 7. $QF_1 \in [50, 95]$, the curves show the accuracy increase with increasing quality factor QF_2 from 60 to 95 with step 5.

Table 1. Detection accuracy(%). In this experiment, $QF_1 \in [50, 59], [60, 69], [70, 79], [80, 89]$, QF_2 is from 60 to 95 with step 5, the block are of 1280×960 pixels.

QF1 \ QF2	60	65	70	75	80	85	90	95
50-59	84.6	89.3	94.7	97.8	99.0	98.9	97.8	96.3
60-69	72.4	81.9	89.6	91.7	97.6	99.2	97.9	97.1
70-79	68.0	67.8	80.9	81.8	90.2	96.5	97.8	96.9
80-89	66.7	63.9	64.8	64.4	72.2	80.4	92.2	95.9

From the experimental results above, we can see that when the original JPEG image with lower quality factor had been cropped and recompressed with higher quality factor, our proposed method work well, which implies that the symmetry of blocking artifacts is obvious in low quality JPEG images and changes slightly after tampering with high quality factor.

5. CONCLUDING REMARKS

When creating a digital forgery, it is sometimes necessary to copy a part of an image and then move it to the same image

or another image in order to conceal or create an important object in the scene. These types of manipulations will lead to inconsistent blocking artifacts in the tampered region, which can therefore be used as evidence of tampering. Our contribution in this paper is the introduction of the blocking artifact characteristics matrix (BACM) which exhibits a symmetrical shape for the original JPEG images and that this symmetrical property will be altered by cropping and recompression operations. We have presented a method that exploits this property of the BACM for effectively detecting cropping and recompression operations in JPEG images.

6. REFERENCES

- [1] J. Lukas, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," *Proc. of SPIE*, vol. 6072, pp. 60720Y, 2006.
- [2] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed jpeg images," *Proc. of DFRWS*, Cleveland, Oh, USA, August 2003.
- [3] A. Swaminathan, M. Wu, and K.J.R. Liu, "Component forensics of digital cameras: A non-intrusive approach," *Proc. of Conf. on Information Sciences and Systems*, Princeton, NJ, March 2006.
- [4] A. Swaminathan, M. Wu, and K.J.R. Liu, "Non-intrusive forensic analysis of visual sensors using output images," *Proc. of IEEE ICASSP*, Toulouse France, vol. 5, PP.V401-404, May 2006.
- [5] A.C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, pp. 3948-3959, 2005.
- [6] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Trans. on Signal Processing*, vol. 53, no. 2, pp. 758-767, 2005.
- [7] T.T. Ng and S.F. Chang, "Blind detection of digital photomontage using higher order statistics," ADVENT Technical Report, Columbia University, June 2004.
- [8] T.T. Ng, S.F. Chang, J. Hsu, L. Xie, and M.P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," *Proc. of ACM Multimedia*, Singapore, vol. 5, pp. 239-248, November 2005.
- [9] A.C. Popescu, *Statistical Tools for Digital Image Forensics*, Ph.D. thesis, Department of Computer Science, Dartmouth College, Hanover, NH, 2005.
- [10] Z. Fan and R.L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. on Image Processing*, vol. 12, no. 2, pp. 230-235, February 2003.
- [11] C.W. Hsu, C.C. Chang, and C.J. Lin, "A practical guide to support vector classification," <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.