Journal of Energy and Power Engineering 10 (2016) 358-363
doi: 10.17265/1934-8975/2016.06.004

# Blur Invariant Image Forgery Detection Method Using Local Phase Quantization

Beste Ustubioglu, Elif Baykal, Gul Muzaffer and Guzin Ulutas
*Department of Computer Engineering, Karadeniz Technical University, Trabzon 61080, Turkey*

Received: April 07, 2016 / Accepted: April 20, 2016 / Published: June 30, 2016.

**Abstract:** With the rapid development of powerful image, editing software makes the forgery of the digital image easy. Researchers proposed methods to cope with image authentication in recent years. We proposed a passive image authentication technique to determine the copy move forgery that copied a part of an image and pasted it on the other region in the same image. First, the method divides the image into overlapping blocks. It uses LPQ (local phase quantization) to label each block. The column average value of labeled blocks constitutes the feature vector for the block. Similarity among the feature vectors gives a clue about the forgery. Local phase quantization has not been used to detect copy move forgery in the literature before. Experimental results show that, the method has higher accuracy ratios and lower false negative values under blurring operation at high levels compared to other methods. Our method can also detect multiple copy move forgery.

**Key words:** Copy move forgery, LPQ, blur invariant.

## 1. Introduction

Digital images can be used in a wide variety of applications including medical imaging, journalism, criminal and forensic investigations. Users want to edit images to improve quality by linear and non-linear image editing tools (for example, Photoshop, 3D Max, GIMP). Easy to use these editing tools can be used to tamper images. Thus, the process of approving the authenticity and integrity of digital images is extremely challenging problem. When an image is used as evidence in a courtroom or is used to make critical decisions in medicine, authenticity of it must be ensured. Therefore, researchers propose techniques to examine the originality of images in digital forensics. Techniques reported in the literature to authenticate images can be grouped into active and passive methods.

Active methods such as digital watermarking or digital signatures try to detect the presence of the watermark or signature to authenticate it. Active methods are not practical, because they require additional information to be transmitted and they also need key management procedures. On the other hand, passive methods do not need any prior information and use statistics of the images to authenticate images. The advantages of the passive methods make them popular to researchers in recent years.

There are several image forgery techniques in the literature and copy move forgery is the most common technique among these techniques. In the copy move forgery technique, a part of an image is copied and pasted into another region in the same image to hide some of the objects or replicate a particular object in the image. But detecting the same regions is very difficult, because some post processing operations such as JPEG (joint photographic experts group) compression, noise adding or Gaussian blurring can be applied on the forged image to hide the clues about forgery. Thus, forgery detection method must be robust to these post processing operations. Original image and example of copy move forgery image are given in Figs. 1a and 1b, respectively.

**Corresponding author:** Beste Ustubioglu, Ph.D., research field: data security.

(a) (b)

**Fig. 1 (a) Original image and (b) forged image.**

Fridrich [1] is the first attempt in the literature to detect the copy move forgery operations. His method divides the image into overlapping square blocks. DCT (discrete cosine transform) is used to extract feature vectors from the blocks. Their method sorts the quantized DCT coefficients lexicographically to relocate similar blocks closer and then checks whether the neighboring vectors are similar. However, their method is sensitive to noise. In 2004, Popescu and Farid [2] used PCA (principal component analysis) to extract feature vectors from the blocks. Their work decreased the dimension of feature vector utilizing the characteristic of PCA. The method is more robust to additive noise. Li et al. [3] employed SVD (singular value decomposition) to decompose the low frequency sub band of the image. Bayram et al. suggested using FMT (Fourier-Mellin transform) to create the feature vectors [4]. Rotation invariant feature of the FMT ensures the rotation invariance. However, their method is successful only for slight rotations. Bravo and Nandi [5] used log polar coordinates to represent image blocks. Their work used sum of angle values to achieve rotation invariance. In 2011, Huang et al. improved the performance of the Fridrich's method by reducing the dimension of feature vector [6]. Their method divides the image into overlapping blocks and extract feature vectors from the blocks using DCT as in Ref. [1]. However, their method truncates the feature vectors to represent each block with most significant frequency coefficients. Their method also quantizes the feature vectors to make the method more robust to compression attacks.

Using the LPQ (local phase quantization) as block feature has not been investigated before in this field as can be seen in the literature. First the proposed method divides image into overlapping blocks and then LPQ is used to label each block. The column average value of labeled blocks constitutes the feature vector for the block. The feature vectors are lexicographically sorted to make the similar vectors closer. Similarity among blocks gives a clue about forgery. The proposed method can detect forgery operation with acceptable accuracy ratios. It also detects forged areas even if the forged image is post processed with Gaussian blurring at high levels. The method also gives better results compared to other works as can be seen in the results.

The rest of the paper is organized as follows. In Section 2, local phase quantization is explained. The details of the proposed method are given in Section 3. Tests to demonstrate the effectiveness of the method are explained in Section 4. Conclusions are also drawn in Section 5.

## 2. Local Phase Quantization

The LPQ is a blur insensitive texture classification method which is proposed by Ojansivu and Heikkila in 2008 [1]. It utilizes the local phase information of the image. This information is extracted by using the 2-D STFT (short-term Fourier transform) computed locally in a window for every image position. The lower frequency resolution reveals the higher spatial resolution. The low frequency phase angles are indicated to be invariant to centrally symmetric blur. The phases of the four low frequency coefficients are decorrelated and quantized in an eight-dimensional space uniformly. A histogram of the result is achieved and it is used as a feature in texture classification.

More detailed explain is that, LPQ extracts local information using an STFT computed over a rectangular window at each pixel position $x$ of the image $f(x)$ defined as follows:

$$F(u, x) = \sum_{y \in N_x} f(x\text{-}y) e^{-j2\pi u^T y} \quad (1)$$

where, $x \in \{x_1, x_2, \ldots x_N\}$ compose of simply 1-D

convolution for the rows and then columns. The local Fourier coefficients $F(u, x)$ are computed at four angles [0, π/2, π, 3π/2]. In 2-D frequencies, the angles were indicated as $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, and $u_4 = [a, -a]^T$ where $a = 1/m$ ($m$ is window size). We set $m$ value to 9.

For each pixel position, the results are represented as follows:

$$F_x^c = [\{F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)\}] \quad (2)$$

$$F_x = [Re\{F(x), Im\{F(x)\}]^T \quad (3)$$

where, $Re\{\cdot\}$ and $Im\{\cdot\}$ are real parts and imaginary parts of the complex number.

Then, $G_x$ (the DFT (discrete Fourier transforms) of the blurred image) is computed for pixel and the resulting vectors are quantized by using a simple scalar quantizer.

$$q_j = \begin{cases} 1, & \text{if } g_j \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

This $g_j$ is the $j$th component of the vector $G(x) = [Re\{F(x), Im\{F(x)\}]$.

It results into a 2 bit code and through 4 coefficients, so an 8 bit codeword is generated between 0 and 255 as:

$$f_{LPQ}(x) = \sum_{j=1}^{8} q_j 2^{j-1} \quad (5)$$

Finally, a histogram of these integer values from all image positions is computed and used as a 256-dimensional feature vector. The diagram of the computing LPQ representation scheme is shown in Fig. 2.

## 3. The Proposed Algorithm Based on LPQ

In this section, we give the details of the proposed method. The proposed method consists of two stages: (i) feature extraction and (ii) similarity matching and marking.

Feature extraction: The method divides the image into overlapping blocks and extract features via LPQ from these blocks. Feature vectors are placed into a matrix. This matrix is lexicographically sorted to make the similar vectors closer. The feature extraction phas can be given in the form of steps as below.

Step 1: The suspicious image is a gray image I of the size $M \times N$ is divided into overlapping fixed-size $b \times b$ blocks. We used $b$ to be 18 in this work.

Step 2: Each block denoted by $B^i$, $i$ = 1…$(N - 17)(M - 17)$ is obtained LPQ values for this each pixel with $m$ = 9 as given in Section 2. LPQ values are quantized using a predefined value $qt$. Therefore, each block will be calculated using $B^i = [LPQ(B^i)/qt]$. A new $B^i$, size of 10 × 10, and then the column average value of $B^i$ constitute the feature vector $F^i$ size of 1 × 10 for the block. Feature vectors of each block $F^i$ constitute a matrix denoted by A of size $[(N - 17)(M - 17), 10]$. The matrix is lexicographically sorted to make the similar vectors. Lexicographically sort

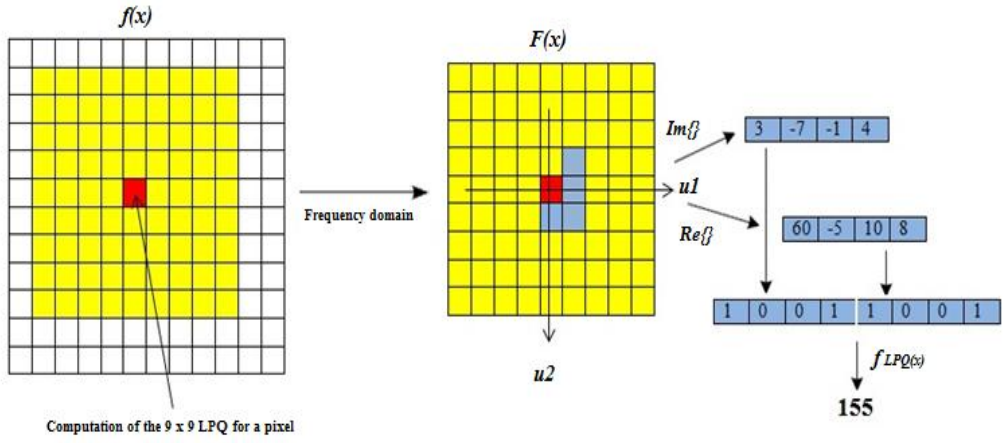



**Fig. 2 A summary of LPQ method.**

operation before the matching procedure speeds up the search because a vector will be compared to only predefined amount of neighboring feature vectors instead of all feature vectors.

Similarity matching and marking: The matrix is searched for similar blocks. If any two vectors are similar, the algorithm will calculate the corresponding shift vector and record the vector into a list. Matching algorithm for a feature vector denoted by $F_i$ can be given in the form of steps as the following. Following algorithm will be applied to all feature vectors in the matrix A.

Step 1: Similarity between the vectors is determined by Euclidean distance. The distance is compared with a predetermined threshold to judge the similarity. For a threshold value of $t_s$, the method judges the similarity between two vectors as given in Eq. (6).

$$F^i = (F_1^i, F_2^i \dots F_{10}^i)$$
$$\sqrt{\sum_{k=1}^{10}\left(F_k^i - F_k^j\right)^2} \le t_s \quad (6)$$

Vector $F_i$ will be compared to $t_n$ feature vectors.

Step 2: Similarity between the neighboring vectors is required to decide a possible forged region but it is not sufficient. The distance among similar blocks must be greater than a predefined threshold value $t_d$ to prevent smooth regions look like forgery. Assume that upper left coordinate of the vectors $F_i$ and $F_j$ be $(x_i, y_i)$ and $(x_j, y_j)$, respectively. The following criterion is used to test the distance between blocks is appropriate or not. The method necessitates that the distance between the two block must be at least threshold $t_d$.

$$\forall\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \ge t_d \quad (7)$$

Step 3: If any two vectors satisfy the rules given in Eqs. (6) and (7), the shift vector between them will be calculated and saved to help the judgment about the forgery at the last step. Shift vectors between them are calculated from each suspicious block pairs. A pair of integers designates shift vector between two blocks. Shift vector $(x_i - x_j,\ y_i - y_j)$ connects the upper left points of these blocks. At last, if the number of suspicious block pairs that have the same shift vectors exceeds a predetermined threshold value, these blocks are marked as forged.

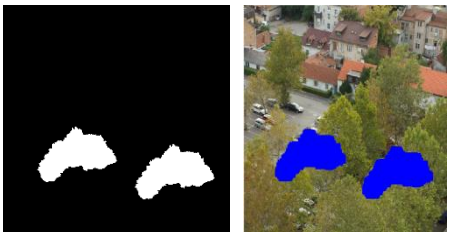
## 4. Experimental Results

This section gives the detailed analysis to show the effectiveness of the method. The forged images were created by an open source image editing software, GIMP, using images of size 512 × 512 pixels and 1,024 × 768 pixels from Google image search and Comofod database [8, 9]. In our experiments, we set all the parameters as $t_s = 1.5,\ t_d = 32,\ t_n = 100,$ $qt = 10$. The success of the detection method is measured with a metric called accuracy $p$ in $[0-1]$ range. An accuracy of 1 corresponds to detection of all copied and pasted regions. False negative, $f$ is also used during experiments to have a measure of the regions detected as forged whereas they are not. The success of the detection algorithm improves as the value of $f$ approaches to zero. Let copied and pasted regions in a fake image be $D_1$ and $D_2$, respectively, whereas copied and pasted regions detected by the algorithm be $R_1$ and $R_2$, respectively. The accuracy ratio of the algorithm is calculated by using Eq. (8).

$$p = \frac{|D_1 \cap R_1| + |D_2 \cap R_2|}{|D_1| + |D_2|},\ f = \frac{|D_1 \cup R_1| + |D_2 \cup R_2|}{|D_1| + |D_2|} - p \quad (8)$$

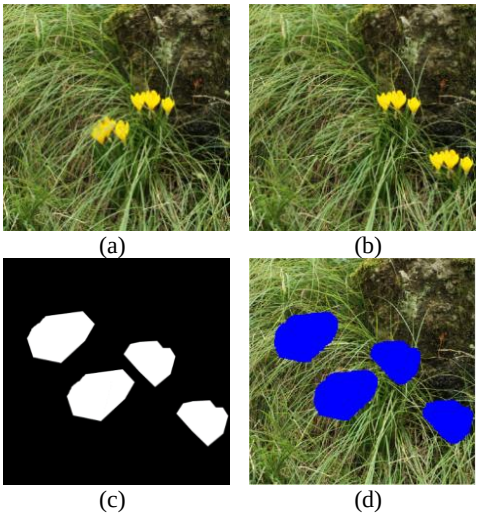
The first experiment gives an idea about the capability of the method when the simple attack is applied on the forged image. Figs. 1a and 1b show original image and forged image, respectively. Mask image is given in Fig. 3a. Visual result of the method shown in Fig. 3b designates that, the method can detect the forged regions even if the forged regions have non-regular shape. The method gives approximately 0.98 accuracy ratio.

Multiple copy move forgery is also realized as the second experiment to show the effectiveness of the method when the forged image has more than one region. Multiple regions of the original image given in Fig. 4a are used to create the forgery. Forged image



(a) (b)

**Fig. 3 (a) Mask of the forgery operation and (b) visual result of the detection algorithm.**



(a) (b)

(c) (d)

**Fig. 4 (a) Original image, (b) forged image, (c) mask of the forgery operation and (d) visual result of the detection algorithm.**

given in Fig. 4b is created using the mask image given in Fig. 4c. The result of the detection algorithm shown in Fig. 4d demonstrates copied and forged regions. Accuracy ratio and false negative values are approximately 0.98 and 0.04, respectively even if the original image has multiple forged regions.

The proposed method is blur invariant. Figs. 5a-5c show that, the visual result of the method when the image is blurred with $w = 5$ $\sigma = 5$, $w = 7$ $\sigma = 7$ and $w = 9$ $\sigma = 9$, respectively. Results show that, the method can detect forgery approximately 0.92 accuracy ratio even if it has been blurred at high levels such as $w = 9$ $\sigma = 9$. False negative values for three results are also smaller than 0.08.

Furthermore, we again select about 30 images with the size of 512 × 512. Then randomly copy a square region and paste it to a non-overlapping position. The forged images are then distorted by Gaussian blurring

(a) $w = 5$ $\sigma = 5$ (b) $w = 7$ $\sigma = 7$ (c) $w = 9$ $\sigma = 9$

**Fig. 5 Visual result of the detection algorithm for different blurring levels.**

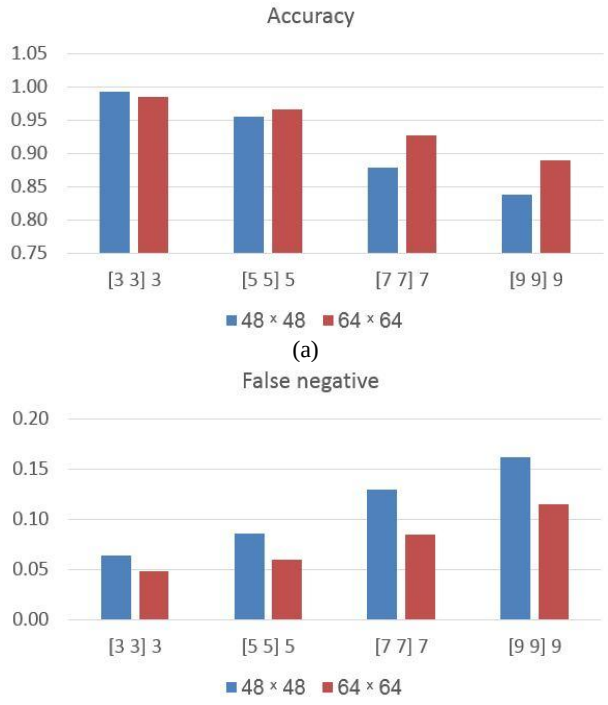



**Fig. 6 The average (a) accuracy and (b) false negative performance of the proposed method.**

operations. In our test, the sizes of the square regions are of 32 × 32 and 48 × 48. The average accuracy, false negative performance with Gaussian blurring over 30 images is shown in Figs. 6a and 6b.

The last experiment is realized to show the difference of the method from the similar works [4, 6] when the Gaussian blurring is used. In our test, the sizes of the square regions are of 48 × 48. Figs. 7a and 7b give the accuracy ratios and false negative values of the methods, respectively. Gaussian Blurring filter with $w = 5$ and $\sigma = 0.5$, 1, 1.5, 2, 2.5, 3 parameters are used to blur the forged images. The method gives higher accuracy ratios (over the 0.96) compared to other works as can be seen in Fig. 7a. The result

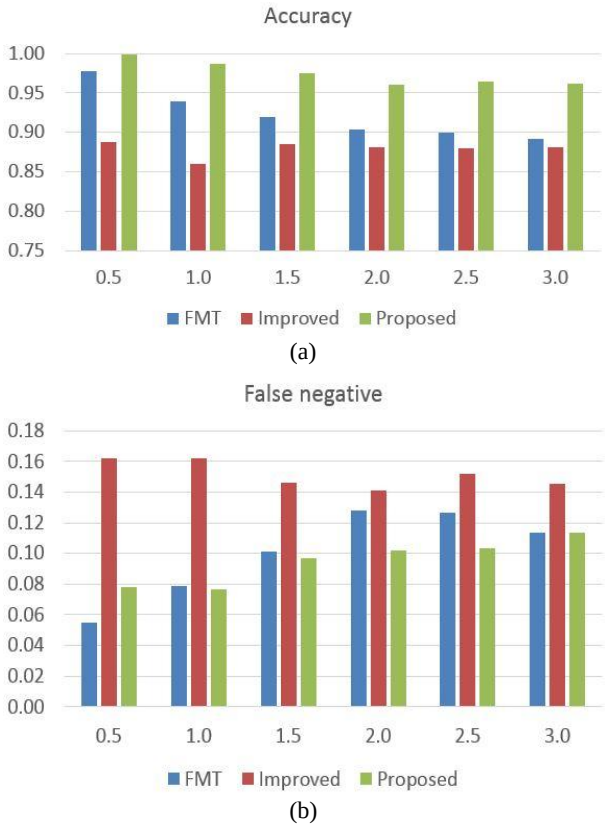



**Fig. 7 Comparison of the method with FMT and Improved according to (a) accuracy and (b) false negative.**

shows that, the method gives better results from the other works for various standard deviations ($\sigma$). False negative values of the method are also lower than (smaller than 0.1) the other works in the literature as can be seen in Fig. 7b. Results indicate that, the method detects corrupted regions successfully.

## 5. Conclusions

We have proposed a new blur invariant method that is the first one in the literature to detect copy move forgery with LPQ. Compared with the other methods in the literature, the method detects the copied and pasted regions with higher accuracy and lower false negative even if the image has undergone blurring operations at high levels. The method can also detect multiple copy move forgery.

## References

[1] Fridrich, J. 2003. "Detection of Copy-Move Forgery in Digital Images." In *Proceedings of the Digital Forensic Research Workshop*, 19-23.

[2] Popescu, A. C., and Farid, H. 2004. *Exposing Digital Forgeries by Detecting Duplicated Image Regions.* Technical report TR-515.

[3] Li, G., Wu, Q., Tu, D., and Sun, S. 2007. "A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries Based on DWT and SVD." In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 1750-3.

[4] Bayram, S., Sencar, H., and Memon, N. 2009. "An Efficient and Robust Method for Detecting Copy-Move Forgery." In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1053-6.

[5] Bravo, S. S., and Nandi, A. K. 2009. "Passive Method for Detecting Duplicated Regions Affected by Reflection, Rotation and Scaling." In *Proceedings of the European Signal Processing Conference*, 824-8.

[6] Huang, Y. 2011. "Improved DCT-Based Detection of Copy-Move Forgery in Images." *Forensic Science International* 206 (1-3): 178-84.

[7] Ojansivu, J. H. V. 2008. "Blur Insensitive Texture Classification Using Local Phase Quantization." *Image and Signal Processing* 5099 (July): 236-43.

[8] Google Image Search. Accessed January 10, 2015. http://images.google.com/.

[9] CoMoFoD database. Accessed January 15, 2015. http://www.vcl.fer.hr/comofod.