

COPY-MOVE DETECTION ALGORITHM BASED ON LOCAL DERIVATIVE PATTERNS

N.I. Evdokimova¹, A.V. Kuznetsov^{1,2}

¹Samara National Research University, Samara, Russia

²Image Processing Systems Institute - Branch of the Federal Scientific Research Centre "Crystallography and Photonics" of Russian Academy of Sciences, Samara, Russia

Abstract. Copy-move forgery is one of most frequently used types of image forgery. During copy-move process a fragment of the image is copied and pasted to another location of the same image to hide some important area of the image. The purpose of copy-move forgery detection algorithm is to identify duplicated areas in the image. This method is based on the calculating features in a sliding or an overlapping window. In this paper, the copy-move detection algorithm using local derivative pattern is proposed. Experimental results show high detection accuracy when the proposed solution is used. Distinctive characteristics of the local derivative pattern based features are robustness to distortions of the duplicated area and low computational complexity.

Keywords: copy-move, forgery, local derivative pattern, local binary pattern, hash value

Citation: Evdokimova NI, Kuznetsov AV. Copy-move detection algorithm based on local derivative pattern. CEUR Workshop Proceedings, 2016; 1638: 304-312. DOI: 10.18287/1613-0073-2016-1638-304-312

1 Introduction

At present, digital images and videos are the most common instruments of information transfer. For example, they are used to confirm an event in mass media. Wide software availability has led to the fact that the process of changing images becomes very simple. In the last 10-15 years, scientists developed algorithms of artificial forgery detection in digital images to provide data security [1-3]. These algorithms make possible to find image area modifications and define forgery characteristics made by criminals.

Embedding copy-move fragments is one of most frequently used methods of image forgery because its easy use. The process of embedding consists of three consecutive stages: copying of an image fragment, distortion of this fragment and pasting it in another area of the same image that has to be hidden. Despite the large number of existing solutions, many of them are not computationally efficient. That is the task of

development of copy-move detection algorithms with high accuracy and low computational complexity is challenging.

This paper is organized as follows. The first section provides basic information about local binary pattern and local derivative pattern. The second section is dedicated to the copy-move forgery detection algorithm description. The third section contains the experimental results carried out using the set of images with copy-move forgery.

2 Local Patterns

2.1 Local Binary Pattern

The idea of local binary pattern (LBP) belongs to Wang [4]. In the original LBP variant, a 3×3 window is used. Eight neighborhood point values around a pixel are compared with central point value. Depending on the comparison result, the central pixel is labeled as 8-bit number. The process of LBP code calculation is presented in Fig. 1.

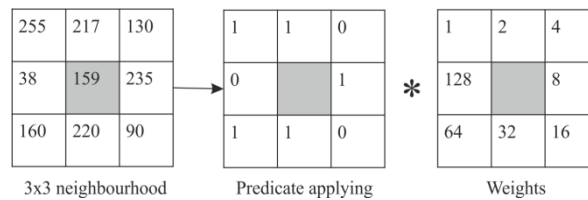


Fig. 1. Base LBP algorithm

The image can be formally represented as an intensity function of the pixel $f(x, y)$. LBP-code is computed for every pixel using pixel values from circular neighborhood $f_i = f(x_i, y_i)$, $i = \overline{1, P}$, where P is a number of equidistant neighbors of the central pixel located on a circle of radius R (Fig. 2) as follows:

$$LBP(N, R) = \sum_{i=0}^{N-1} I_1(f_i - f_0) \cdot 2^i, \quad (1)$$

where I_1 is a predicate defined as follows:

$$I_1(m) = \begin{cases} 0, & m < 0 \\ 1, & m \geq 0 \end{cases}. \quad (2)$$

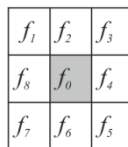


Fig. 2. Example of 8-neighborhood of f_0 with $R = 1$

Also it should be noted that the LBP method is very effective because of its low computational complexity. The main disadvantage of this type of pattern is its sensitivity to noise distortions [5].

2.2 Local Derivative Pattern

In fact, LBP code is a result of applying the first-order derivative of the neighborhood of the center pixel as shown in (1). For a more informative description of the image the operator LDP was proposed in [6]. This operator allows encoding the information obtained as a result of high-order derivatives calculation. n-order derivative along various directions that are used in this method are calculated using binary encoding function.

The first-order derivatives along $0^\circ, 45^\circ, 90^\circ$ and 135° directions are denoted as $f'_\alpha(x_0, y_0)$, where $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$. The first-order derivatives at (x_0, y_0) position can be written as follows:

$$\begin{aligned} f'_0(x_0, y_0) &= f_0 - f_4, & f'_{45}(x_0, y_0) &= f_0 - f_3, \\ f'_{90}(x_0, y_0) &= f_0 - f_2, & f'_{135}(x_0, y_0) &= f_0 - f_1. \end{aligned} \quad (3)$$

The second-order directional LDP along the α direction at (x_0, y_0) is defined as follows:

$$\begin{aligned} LDP_\alpha^2(x_0, y_0) &= I_2(f'_\alpha(x_0, y_0), f'_\alpha(x_1, y_1)), \\ &I_2(f'_\alpha(x_0, y_0), f'_\alpha(x_2, y_2)), \\ &\dots \\ &I_2(f'_\alpha(x_0, y_0), f'_\alpha(x_8, y_8)), \end{aligned} \quad (4)$$

where I_2 is a predicate defined as follows:

$$I_2(m, n) = \begin{cases} 0, & m \cdot n \geq 0 \\ 1, & m \cdot n < 0 \end{cases}. \quad (5)$$

Finally, the second-order LDP is defined as the concatenation of the four 8-bit directional LDPs:

$$LDP^2(x, y) = \{LDP_\alpha^2(x, y) \mid \alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ\}. \quad (6)$$

As it can be seen from the equations above, LDP operator assigns a code to each pixel of the image by comparing two derivatives along the same direction for two adjacent pixels. Then obtained codes are concatenated into a single sequence of 32 bits. Equations derivatives along α direction are performed using 16 primitive patterns [4] (see Fig. 3). These patterns characterize distinctive spatial relationships in the local area.

Unlike LBP, the second-order LDP allows to encode the change of derivative value along a fixed direction. This change represents a characteristic of a local image area.

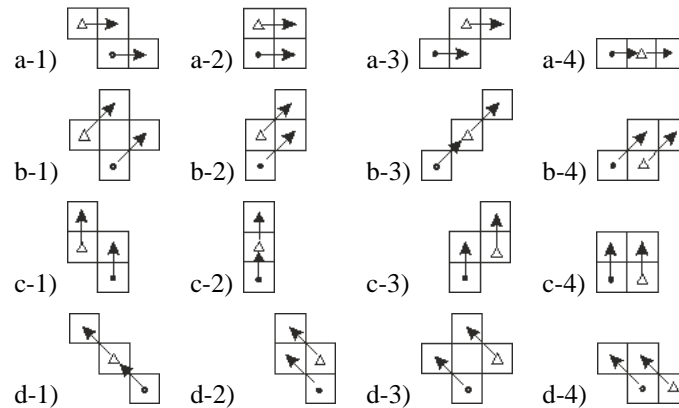


Fig. 3. Primitive patterns used in LDP

Fig. 4 illustrates the types of local transitions in primitive patterns used in LDP. Transitions represent a binary value. Each of the 16 primitive patterns in Fig. 3 belongs to one of the two groups: a 3-point pattern or a 4-point pattern. A 3-point pattern is assigned to the value '0' according to (9), if it increases or decreases monotonically (Fig. 4c.). In other case, a pattern is assigned to the value '1' (Fig. 4a.). Similarly, 4-point pattern is assigned to the value '1' if there is a change of the derivative sign (Fig. 4b.), and '0' if it increases or decreases monotonically (Fig. 4d.). Thus, the binary code of the image area is formed using the local transitions based on the primitive patterns.

As an example, let us consider some area of the image shown in Fig. 5. To calculate the second-order LDP along $\alpha = 0^\circ$ direction at the f_0 position, the four patterns in Fig. 3a are applied to the image by superposition of $*$ and Δ to the f_0 in series. When applying the pattern a-1 (Fig. 3a.) by superposition of $*$ to f_0 , both directional derivatives increase monotonically as shown in Fig. 4(d-1). Thus, the primitive pattern corresponds to the value of '0'. Similarly, applying the patterns a-2, a-3 and a-4 (Fig. 3a.) by superposition of $*$ to f_0 corresponds to the cases shown in Fig. 4(b-1), Fig. 4(d-1) and Fig. 4(a-2), correspondingly. As a result, the following three bits of directional LDP are calculated as '101'. We then repeat the above procedure with the same four patterns but now using superposition of Δ to f_0 . According to this we get a bit sequence '0100' for the last four bits. As a result, the code '01010100' is formed for directional LDP along $\alpha = 0^\circ$. In the same way, the codes of the primitive patterns shown in Fig. 3b-3d are formed along $\alpha = 45^\circ, 90^\circ, 135^\circ$ directions, correspondingly. Finally, we get the following 32-bit code:

$$LDP^2(f_0) = 0101010000 \ 1011111101 \ 0000110001 \ 10,$$

that is generated by concatenation of the four 8-bit LDP along $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ directions in accordance with (6).

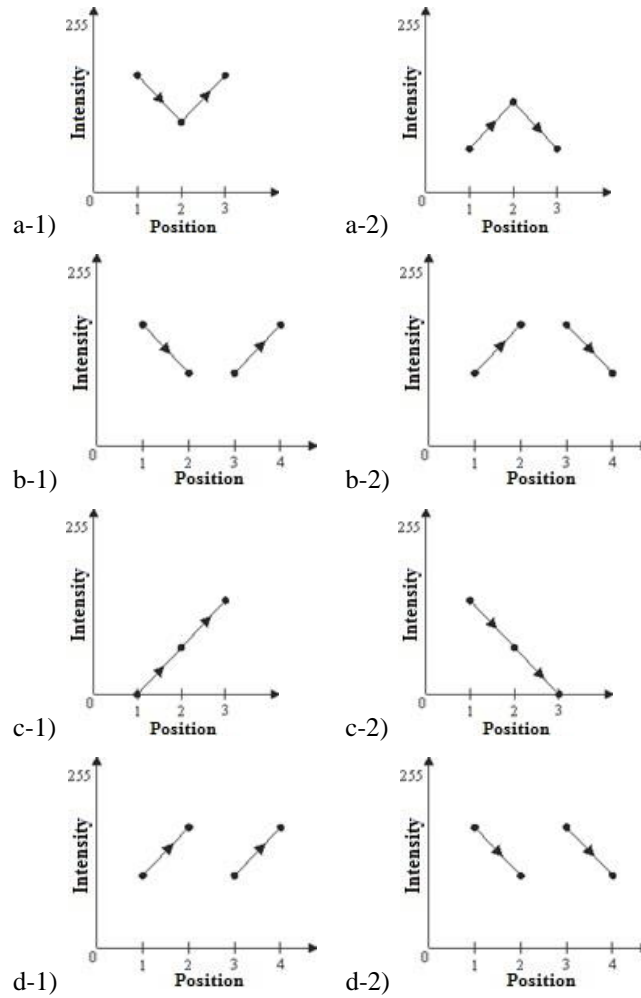


Fig. 4. Types of local transitions used for encoding primitive patterns of LDP

122	125	123	125	121
126	127	129	121	125
122	123	124	128	122
123	122	123	122	129
121	122	123	122	121

Fig. 5. Sample image fragment with $f_0 = 124$

2.3 n-order Local Derivative Pattern

Calculation of the n-order derivatives along directions $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ at the f_0 position provides n-order directional LDP that are defined as follows:

$$\begin{aligned}
 LDP_\alpha^n(x_0, y_0) = & I_2(f_\alpha^{n-1}(x_0, y_0), f_\alpha^{n-1}(x_1, y_1)), \\
 & I_2(f_\alpha^{n-1}(x_0, y_0), f_\alpha^{n-1}(x_2, y_2)), \\
 & \dots \\
 & I_2(f_\alpha^{n-1}(x_0, y_0), f_\alpha^{n-1}(x_8, y_8)),
 \end{aligned} \tag{7}$$

where $f_\alpha^{n-1}(x_0, y_0)$ is the (n-1)-order derivative along the α direction at (x_0, y_0) . It represents the change of (n-1)-order gradient and provides additional information about the local neighborhood. Finally, the n-order LDP is determined by the following expression:

$$LDP^n(x, y) = \{LDP_\alpha^n(x, y) \mid \alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ\}. \tag{8}$$

Thus, n-order LDP is also based on derivatives along the four directions. High-order local derivative patterns allow to describe local neighborhood in a more detailed way than first-order local derivative pattern that is used in LBP. However, they become more sensitive to noise with derivative order increase. In the paper [6], the authors proposed that the LDP sensitivity to noise increases when the order $n \geq 4$. Given this constraint, we will use only second- and third-order LDP.

The advantages of LDP to LBP can be summarized as follows:

- LBP is not able to provide a detailed description of the image features. n-order LDP allows forming a more detailed description of the image local area using (n-1)-order directional derivatives.
- LBP describes the relationship between the central pixel and its neighbors. LDP allows encoding various spatial relationships in the local area of the image and therefore represents more information about spatial features.

3 Copy-move Detection Algorithm

Most of existing solutions [1,7] use features extraction in overlapping blocks of an image and then nearest features search is applied. This approach definitely leads to missing duplicates increase due to overlapping blocks analysis.

In this paper we use the algorithm proposed in the papers of Kuznetsov AV, Myasnikov VV and Glumov NI [8-10]. The algorithm consists of the following main steps:

1. Calculation of the values of local pattern codes (LBP or LDP) using the sliding the sliding window of the size 5×5 . The values are put in the code matrix.
2. Hash values calculation [9] based on the code matrix data using the sliding window.
3. Putting hash values into the hash value matrix.
4. Construction hash values histogram.
5. Analysis of the hash value histogram. Hash values frequency exceeding '1' are searched in the histogram.
6. Output binary image generation which shows the places of potential duplicates.

The hash function of an image fragment is defined as a transform that puts each value of the code matrix in correspondence with a non-negative integer value (hash value) [8-10]. The histogram of hash values stores hash values frequency appearance in the image. The frequency value characterizes potential copy-move forgery regions.

4 Experiments

The experiments were carried out on a desktop PC with Intel Core i5-3470 processor and 8 GB RAM using MATLAB R2014a software.

Ten gray-scale digital images with dimensions 512×512 were chosen as the objects of experiments. The authors developed a copy-move generation procedure which enables to add distortions and control their parameters (linear contrast enhancement and Gaussian noise).

We used F1_Score value as a quality measure. It combines detection accuracy value and false positive and false negative errors to estimate the quality of the proposed solution. F1_Score is defined as follows:

$$F1 = \frac{2 \cdot tp}{2 \cdot tp + fp + fn}. \quad (9)$$

The results of experiments that were carried out on the set of generated images are showed in Table 1 and Table 2.

Table 1. The research of forgery detection algorithms for embedding dimensions 128×128

	LBP	LDP-2	LDP-3
Linear Contrast	0.87	0.93	0.96
Gaussian Noise	0.84	0.91	0.95

Table 2. The research of forgery detection algorithms for embedding dimensions 64×64

	LBP	LDP-2	LDP-3
Linear Contrast	0.82	0.91	0.96
Gaussian Noise	0.80	0.89	0.94

5 Conclusion

During the described research the algorithm that based on LDP was developed. It allows to detect distorted copy-move forgeries in digital images. Due to the specifics of local patterns, the algorithm has low computational complexity so it can be used to analyze large images (e.g. remote sensing data). Research of copy-move forgery detection with geometric distortions and compression is not presented in this study and will be done further. We also plan to compare the quality of the proposed copy-move forgery detection algorithm with existing solutions based on the features calculation in future works.

Acknowledgements

This work was supported by Samara National Research University State contract №2014/198 (project code 2298) and the Russian Foundation for Basic Research (RFBR) grant № 16-37-00056 мол_a.

References

1. Christlein V, Riess C, Jordan J, Angelopoulou E. An evaluation of popular copy-move forgery detection approaches. *IEEE Trans. Inf. Forensic Secur.*, 2012; 7(6): 1841-1854.
2. Popescu A, Farid H. Exposing digital forgeries by detecting duplicated image regions. URL: <http://www.ists.dartmouth.edu/library/102.pdf>
3. Fridrich J, Soukal D, Lukas J. Detection of copy-move forgery in digital images. URL: <http://www.ws.binghamton.edu/fridrich/Research/copymove.pdf>
4. Wang L, He D-C. Texture classification using texture spectrum. *Pattern Recognition*, 1990; 23(8): 905-910.
5. Ren J, Jiang X, Yuan J. Noise-Resistant Local Binary Pattern With an Embedded Error-Correction Mechanism. *IEEE Trans. Im. Proc.*, 2013; 22(10): 4049-4060.
6. Zhang B, Gao Y, Zhao S, Liu J. Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE Trans. Im. Proc.*, 2010; 19(2): 533-544.

7. Kuznetsov AV, Myasnikov VV. A copy-move detection algorithm based on binary gradient contours. *Computer Optics*, 2016; 40(2): 284-293. DOI: 10.18287/2412-6179-2016-40-2-284-293.
8. Glumov N, Kuznetsov A, Myasnikov V. The algorithm for copy-move detection on digital images. *Computer Optics*, 2013; 37(3): 360-367.
9. Kuznetsov A, Myasnikov V. A fast plain copy-move detection algorithm based on structural pattern and 2D Rabin-Karp rolling hash. In: Campilho A, Kamel M. (eds.) *ICIAR 2014*, Springer, Heidelberg. Part I. LNCS, 8814: 461-468.
10. Kuznetsov AV, Myasnikov VV. Efficient linear local features based copy-move detection algorithm. *Computer Optics*, 2013; 37(4): 489-495.