

AN EFFICIENT AND ROBUST METHOD FOR DETECTING COPY-MOVE FORGERY

Sevinc Bayram

Husrev Taha Sencar

Nasir Memon

Polytechnic Institute of NYU
ECE Dept.
Brooklyn, NY

TOBB University of
Economics & Technology
Comp. Engineering Dept.
Ankara, TURKEY

Polytechnic Institute of NYU
CIS Dept.
Brooklyn, NY

ABSTRACT

Copy-move forgery is a specific type of image tampering, where a part of the image is copied and pasted on another part of the same image. In this paper, we propose a new approach for detecting copy-move forgery in digital images, which is considerably more robust to lossy compression, scaling and rotation type of manipulations. Also, to improve the computational complexity in detecting the duplicated image regions, we propose to use the notion of counting bloom filters as an alternative to lexicographic sorting, which is a common component of most of the proposed copy-move forgery detection schemes. Our experimental results show that the proposed features can detect duplicated region in the images very accurately, even when the copied region was undergone severe image manipulations. In addition, it is observed that use of counting bloom filters offers a considerable improvement in time efficiency at the expense of a slight reduction in the robustness.

Index Terms— Digital Forensics, tamper detection, copy-move forgery, duplicated region detection

1. INTRODUCTION

Powerful digital media editing tools made it possible to produce good quality forgeries for almost anyone. One of the specific type of forgeries, which is the main interest of this paper, is copy-move forgery, that can be done very easily by using the tools such as Cloning in Photoshop. This type forgery usually aims to cover an unwanted scene in the image, by copying another scene from the same image, generally a textured region, and pasting it onto the unwanted region. In Fig. 1, an example of copy-move forgery can be seen, where the leaves of the trees are duplicated to remove a person from the image. Therefore, the goal of copy-move forgery detection techniques is detecting the duplicated image regions. However, these regions might not be the exact duplicates, since the tamperer could use retouching tools, add noise, or compress the resulting image. Furthermore, in real life copy-move forgeries, it is very likely for the copied and moved image part to



Fig. 1. Left is the original, right is the tampered image.

be subjected to slight rotation, scaling, or blurring for better blending purposes. Hence, a copy-move forgery detection technique should be robust to such operations as well.

One direct approach to this problem is proposed by Fridrich et al. [1], which essentially performs an exhaustive search by comparing the image to every cyclic-shifted versions of itself. Since this approach requires $(MN)^2$ steps for an image sized $M \times N$, it is not practical. The same authors also proposed to use the autocorrelation properties of the image to detect the duplicated regions. Although this method is more efficient as compared to exhaustive search, it requires the duplicated region to be impractically large to perform reliably.

Another approach for detecting copy-move forgeries is the block-matching procedure, which first divides the image into overlapping blocks. The aim of this approach is to detect connected image blocks that were duplicated, instead of detecting the whole duplicated region. Since the copied region would consist of many overlapping blocks and moving the region means moving all the blocks by the same amount, the distance between each duplicated block pair would be the same. Therefore, the decision of forgery can be made only if there are more than a certain number of duplicated image blocks within the same distance and these blocks are connected to each other.

One of the challenges here is to find the robust representations for the image blocks, so that the duplicated blocks can be identified under modifications. Several authors proposed to use different features for this purpose. Fridrich et al. extracted DCT coefficients, which are known to be robust

to certain modifications that include (low-pass) filtering and compression [1]. Later, in [2], Popescu and Farid proposed to apply principal component analysis to obtain compact representations for the blocks. These representations were also robust to additive gaussian noise as well as medium level JPEG compression. In a similar manner, Li et al. extracted features by applying singular value decomposition to low frequency wavelet transform bands [3].

Another challenge is detecting the duplicated block pairs, which would expected to have same/similar features, in a reasonable time. Since brute-force search would be computationally very expensive, in [1], [2], and [3], the authors proposed to lexicographically sort the feature vectors, so that blocks with similar features would follow each other.

In this paper, we propose to extract features from the image blocks by using Fourier-Mellin Transform (FMT). These features would not be only robust to lossy JPEG compression, blurring, or noise addition, but also known to be scaling and translation invariant. In our experiments, we first used lexicographic sorting method and we compare the robustness of our features with the ones utilized in [1], and [2]. Furthermore, we attempt to reduce the detection time by using counting bloom filters, instead of lexicographic sorting.

The rest of the paper is organized as follows. In Section 2, we give a brief summary of the proposed features. In Section 3, we explain the different detection schemes, and in Section 4, we describe the decision process. The experimental results and our discussions are given in Sections 5 and 6, respectively.

2. ROTATION SCALE AND TRANSLATION INVARIANT FEATURES

Unlike the previous works in the area, we assume that the tamperer would scale, rotate and/or blur the image part before pasting it over to reduce the visual artifacts. Therefore, the utilized features should be insensitive to those operations as well. It should be noted that, the tamperer can only afford to slightly rotate, scale or blur the duplicated image region. Aside from visual intactness considerations, since the type of traces introduced by these modifications depend on the strength of the modification, various other tamper detection techniques could be effectively used for their detection.

In this paper, we rely on the properties of Fourier-Mellin Transform [4], which includes translation, scaling, and rotation invariancy. These properties of FMT were previously exploited in the context of watermarking to combat against desynchronization attacks due to geometric transformations [5]. To achieve these properties, we first divide the image into $b \times b$ overlapping blocks. Consider a block $i(x, y)$ and its rotated, scaled, and translated version $i'(x, y)$ where $i'(x, y) = i(\sigma(x \cos \alpha + y \sin \alpha) - x_0, \sigma(-x \sin \alpha + y \cos \alpha) - y_0)$ and (x_0, y_0) , σ and α indicates translation, scaling and rotation parameters respectively. The following procedure is applied

to the blocks:

- * Take the fourier transform of the block. This will ensure that the transform is translation invariant.

$$|I'(f_x, f_y)| = |\sigma|^{-2} |I(\sigma^{-1}(f_x \cos \alpha, f_y \sin \alpha), \sigma^{-1}(-f_x \sin \alpha + f_y \cos \alpha))| \quad (1)$$

- * Re-sample the resulting magnitude values into log-polar coordinates.

$$|I'(\rho, \theta)| = |\sigma|^{-2} |I(\rho - \log \sigma, \theta - \alpha)| \quad (2)$$

- * Project the log values onto 1-D

$$g(\theta) = \sum_j \log(|I(\rho_j, \theta)|) \quad (3)$$

Here we only compute $g(\theta)$ for $\theta \in [0^\circ, 2^\circ, \dots, 180^\circ]$

- * Add two halves of $g(\theta)$ together.

$$g_1(\theta') = g(\theta') + g(\theta' + 90^\circ) \quad (4)$$

- * Quantize the values of $g_1(\theta')$ and obtain 45 features.

Essentially, these features are invariant of translation and scaling but not rotation. To obtain rotational invariancy one should consider every cyclic shift of the feature vector but since the number of blocks is very large this cannot be realized. Therefore, our feature vector is expected to be only scale and translation invariant; however, the experimental results showed that these features are invariant to rotation to small degrees as well.

3. DETECTION SCHEMES

For detecting duplicated regions, the blocks that yield the same and/or similar feature vectors have to be determined. In this section, we describe two methods for this purpose.

3.1. Lexicographic Sorting

After obtaining a feature vector for each block as described in Section 2, a matrix A is constructed by inserting the feature vectors into the matrix in a way that the rows of A would correspond to the blocks and columns would indicate the feature vectors. For an image of size $M \times N$, matrix A would have $(M - b + 1) \times (N - b + 1)$ rows and F columns, where F is the number of features. Note that if the two blocks in the image are very similar, their feature vectors and the corresponding rows in matrix A would be similar as well. The detection can be done by lexicographically sorting the rows of A matrix, so that the features of the duplicated block pairs will come successively. This step would require $MN \log_2(MN)$ steps. For example, for an image of size 1000x1000, it would take 10^6 steps which would be computationally very expensive.

3.2. Counting Bloom Filters

To improve the efficiency of detection step, we propose to use counting bloom filters as followed, which essentially compares the hashes of features as opposed to features themselves. This is realized as following.

- Form an array K with k elements which are all zero initially.
- Hash the feature vector f_i of each block such that each hash value will indicate an index number in the array K .
- If the feature vectors of two blocks are identical they would give the same hash value yielding same index value, increment the value of the corresponding element in K . That is,

$$h = \text{hash}(f_i) \quad (5)$$

$$K(h) = K(h) + 1 \quad (6)$$

We assume any element of array K that is higher than 2 indicates duplicated block pairs.

One can imagine this scheme would require the duplicated blocks to be exactly same, and the resulting image to be saved without any compression. Although we chose our features to overcome this problem, we don't expect these scheme to be as robust as lexicographic sorting, due to the fact that lexicographic sorting scheme requires the duplicated blocks to have similar feature vectors only. On the other hand, these approach would reduce the computational time significantly, since the hashing and forming the array K will be executed at the same step as feature extraction. So the only computation time added by this scheme will be due to finding the elements which has value more than 2, at a complexity of $O(\text{length}(MN))$.

4. FORGERY DECISION

Finding the duplicated blocks is not enough for deciding the forgery, since most of the natural images would have many similar blocks. There should be more than a number of connected blocks within the same distance to make such a decision. We can calculate the distance between the two blocks that are detected to be the duplicated pairs, as described in Section 3, a_i and a_j , whose starting positions are (x_i, y_i) and (x_j, y_j) respectively, as follows:

$$d_x(i, j) = |x_i - x_j|, d_y(i, j) = |y_i - y_j| \quad (7)$$

Note that in the lexicographically sorting scheme, a_i and a_j would correspond to the blocks which were coming successively in matrix A , and in the bloom filter scheme, a_i and a_j would indicate the blocks whose feature vectors yielded to the same hash value. To measure how many blocks are detected

as duplicates within the same distance, a distance vector D is constructed. The values of D are set to zero initially. When a distance between two blocks are calculated, the corresponding index value of D is incremented by one :

$$D(d_x, d_y) = D(d_x, d_y) + 1 \quad (8)$$

Any value of $D(d_x, d_y)$, which is more than the threshold TH indicates the blocks that are copied and moved along the same distance. If these blocks are connected to each other, then a decision of forgery can be made.

5. EXPERIMENTAL RESULTS

In our experiments, we tampered several images by copying and pasting one image block over another, in the same image. We also downloaded several images from the Internet. In the first set of our experiments, we have extracted FMT features and by using lexicographic sorting we analyzed the performance of the proposed scheme. We also implemented the methods, described in [1], and [2] and compared the robustness of these three methods. We used $b = 16$ as a block size and we slide the blocks one pixel each time. We assumed that the smallest size of duplication would be at least 32×32 . In this case, there would be $(32 - 16 + 1) \times (32 - 16 + 1) = 289$ connected duplicated blocks. Considering the modifications, we chose our preset threshold TH to be 150.

Fig. 2 displays the results for the images, where no modification is applied on the copied and moved image region and the images were saved under high quality JPEG compression. The methods based on DCT coefficients, and Eigenvalues that are obtained by principal component analysis, gave similar results for these images as well.

Table 1. Performance Results

Manipulation Type	FMT	DCT	Eigenvalues
JPEG	20	40	50
Rotation	10°	5°	0°
Scaling	10%	10%	0%

In the second set of our experiments, we have copied and moved a small block of size (32×32) in Lena image. We obtained a set of images by saving this image with various JPEG quality levels. We have also obtained images by rotating and scaling the copied block before pasting. One can see the comparison of performance results for the three methods in Table 1. The values in this table indicate upper limits where the methods can successfully detect the forgeries under the modifications. Here, we see that our features are very robust to JPEG compression and forgeries can be detected even if the forged image is saved at JPEG quality factor 20. The method can also detect rotations of up to 10° while DCT coefficients are successful up to 5° and Eigenvalues are not sensitive to

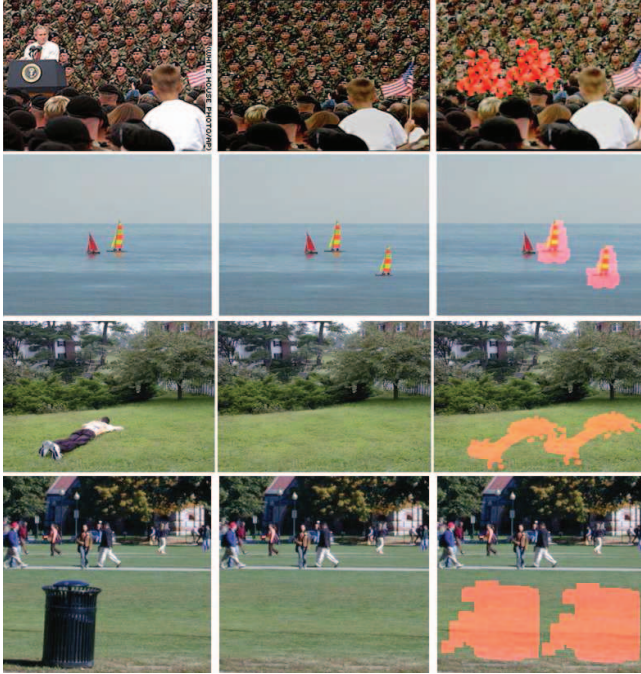


Fig. 2. Each row represent different images and for each row the first column is the original image, second column is the forged image and the last column is the detection result with our algorithm. Third and fourth images are taken from [2].

rotation at all. We can also see that FMT and DCT features are insensitive to scaling up to 10%-scaling while Eigenvalues can not detect scaling at all. (see, Fig. 3).

In the third and last set of our experiments, we have used counting bloom filters in the detection step. So far we have tested this scheme only on JPEG compressed images. Fig. 4 displays the results for the scheme when the forged image is compressed under JPEG quality factors of 90, 80, 70 and 60. One can see that, with this form, the detection is robust to quality levels of 70 or more. For quality factor 60, since the number of connected blocks did not reach the threshold value we set, the forgery decision is not made by the system.

We have also compared the computation times for the two detection schemes. For a 200x200 Lena image, while lexicographic sorting step takes 25 seconds, the scheme based on counting bloom filters take only 2 seconds. Hence, to utilize the computational advantage, the quality of the image has to be first determined prior to testing. If the image is not heavily compressed counting bloom filters offers a computational advantage; otherwise, lexicographic sorting should be preferred.

6. CONCLUSION

In this paper, we studied the problem of copy-move forgery detection. To detect the forgeries under the modifications, we proposed to use FMT features which are invariant to scaling



Fig. 3. Shown are the detection results for tampered Lena images. First image has no operation, second saved with JPEG quality of 20, the copied area in the third image scaled 5% and it is rotated 5° in the forth image.



Fig. 4. Shown are original image, tampered image and detection results for JPEG compression with 90,80,70 and 60 respectively.

and translation. Our experimental results show that we can detect copy-move forgery very accurately even if the forged image is rotated, scaled or highly compressed. We compared the robustness of our method with the previously proposed schemes which use DCT coefficients and Eigenvalues as features, and we showed that our method is more robust to various types of processing. We also presented a new detection scheme that make use of counting bloom filters. We have seen that while this detection scheme improves the efficiency, the robustness of the method is reduced. Our experiments for all schemes is going on for additive gaussian noise and blurring type of operation.

7. REFERENCES

- [1] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," *Proc. Digital Forensic Research Workshop, Cleveland, OH*, August 2003.
- [2] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Technical Report, TR2004-515, Dartmouth College, Computer Science*, 2004.
- [3] Guohui Li, Qiong Wu, Dan Tu, and ShaoJie Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd," *ICME*, 2007.
- [4] Yunlong Sheng and Henri H. Arsenault, "Experiments on pattern recognition using invariant fourier-mellin descriptors," *J. Opt. Soc. Am. A*, vol. 3, no. 6, pp. 771, 1986.
- [5] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Processing*, vol. 10, pp. 767–782, 2001.