

A SURVEY OF COPY-MOVE FORGERY DETECTION TECHNIQUES

Sevinc Bayram

Polytechnic Institute of NYU
ECE Dept.
Brooklyn, NY

Husrev Taha Sencar

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

Nasir Memon

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 generally to conceal unwanted portions of the image. Hence, the goal in detection of copy-move forgeries is to detect image areas that are same or extremely similar. In this paper, we review several methods proposed to achieve this goal. These methods in general use block-matching procedures, which first divide the image into overlapping blocks and extract features from each block, assuming similar blocks will yield similar features. Later, a matching step takes place where the aim is to find the duplicated blocks based on their feature vectors. A forgery detection decision is made only if similar features are detected within the same distance of features associated to connected blocks. We examine several different block-based features proposed for this purpose in relation to their time complexity and robustness to common processing scaling up/down, compression, and rotation. We will also include comparative results for better evaluation.

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

1. INTRODUCTION

Maliciously manipulate, and tamper digital images without leaving any obvious clues became very easy with the widely available, easy to use and extremely powerful digital image processing tools such as Photoshop and Freehand. As a result, there is a rapid increase of the digitally manipulated forgeries in mainstream media and on the Internet. This trend indicates serious vulnerabilities and decreases the credibility of the digital images. Therefore, developing techniques to verify the integrity and the authenticity of the digital images became very important, especially considering the images presented as evidence in a court of law, as news items, as a part of a medical record, or as a financial document. In this sense, image tamper detection is one of the primary goals in image forensics.

Recently, many authors studied the problem of detecting image forgeries, assuming that even if the tampered images

do not reveal any visual artifacts or anomalies, the underlying statistics of these images would be different than the original ones. Based on this, Bayram et al. described features that are sensitive to various common image processing operations and built classifiers to detect such operations [1]. Similarly, Popescu et al. proposed a method to detect correlations in the image which occurs due to scaling or rotation [2]. Based on the assumption that tampering an image would introduce differences in lighting of the objects in the image, Johnson et al. described a technique for estimating the direction of an illuminating light source [3]. Later on, in [4] Swaminathan et al. used inconsistencies in color filter array interpolation to detect tampered parts of an image and in [5], Sutcu et al. proposed a method for estimating sharpness/blurriness value of an image assuming that the average sharpness/blurriness value of the tampered region would be different compared to the non-tampered parts of the image.

In this paper, we are considering a specific type of image forgery where a part of the image is copied and pasted on another part of the same image mostly to cover an important object. An example for this type of forgery can be seen in Fig. 1, where a group of soldiers are duplicated to cover President George W. Bush. This process can be done without any modifications on the duplicated regions. As a result, the tampered region would exhibit the same characteristics as the rest of the image which makes it hard to identify using the tools that are designed to detect the anomalies in the image. Hence, to detect copy-move forgeries, we need techniques that can detect the image regions which occurs more than once in the image. However, finding the very same region might not be enough in all cases, since the tamperer could use retouching tools, add noise, or compress the resulting image. Furthermore, for better blending purposes the copied area might be slightly rotated, scaled, or blurred without disturbing the image statistics, or revealing the forgery. Therefore, a good copy-move forgery technique should detect the duplicated image regions, without getting affected by the slight modifications and/or operations such as noise addition, and compression.

To accomplish this task several copy-move forgery detection techniques have been proposed. In this paper, we will



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

give an overview of these techniques. The outline of the paper is as follows. In Section 2, we will review the copy-move forgery techniques. In Section 3, we will present the comparative results and in section 4, we will conclude.

2. COPY-MOVE FORGERY DETECTION TECHNIQUES

The goal in copy-move forgery detection is detecting duplicated image regions, even if they are slightly different from each other. One direct solution to this problem would be an exhaustive search, which involves comparison of the image to every cyclic-shifted version of itself. However, this approach would be computationally very expensive and would take $(MN)^2$ steps for an image of size $M \times N$. Also, this type of search might not work in the case where the copied area has undergone some modifications. A second and more efficient approach, which was proposed by Fridrich et al. in [6], is the use of autocorrelation properties. Nevertheless, this approach is shown to be effective only when the duplicated regions were a large portion of the image.

Another approach, which is the main interest of this paper, is block-matching procedure. In this approach, the image is segmented into overlapping blocks first. The task here is to detect connected image blocks that are copied and moved, instead of detecting the whole duplicated region. Note that the copied region would consist of many overlapping blocks and since each block would be moved with the same amount of shift, the distance between each duplicated block pair would be the same, as well. Therefore, the forgery decision can be made only if there are more than a certain number of similar image blocks within the same distance and these blocks are connected to each other so that they form two regions of the same shape.

One of the important points 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 to represent the image blocks. These features and their robustness properties will be discussed in the next section. Another important issue is to be able to find the block pairs that have same/similar representations. Since brute-force search would be computationally very expensive, different matching techniques are proposed

which will be described and evaluated in Section 2.2. Finally in Section 2.3, the decision process that involves counting the number of block pairs in the same distance, will be explained in detail.

2.1. Features

The first step of block-matching procedure is dividing the image into overlapping blocks. This step is followed by a feature extraction process, where the biggest challenge is to determine the features, that would yield to the same or very similar values for duplicated blocks, even under modifications. In [6], the authors proposed to use quantized Discrete Cosine Transform (DCT) coefficients for this purpose. The advantage of DCT is that the signal energy would be concentrated on the first few coefficients, while most other coefficients are negligibly small. Therefore, the changes in high frequencies, which would occur due to the operations such as noise addition, compression, and retouching should not affect these first coefficients. In their paper, the authors showed that their technique was robust to the retouching operations, however they did not perform any other robustness tests.

Later on, Popescu et al. proposed to perform Principal Component Analysis (PCA) to derive an alternative representation of the blocks [7]. For this purpose, the coefficients in each block were vectorized and inserted in a matrix and the corresponding covariance matrix was computed. A new linear basis was obtained by finding the eigenvectors of the covariance matrix. To reduce dimensionality, the projection of each block onto these basis vectors with higher eigenvalues was obtained and used as new representations. These new representations were known to be robust to compression and noise addition [8], however re-sampling in the image (scaling, rotation) would affect the eigenvalues. In their paper, the method was shown to be robust to compression up to JPEG quality level 50 and to noise addition with SNR 36dB and 29dB.

In a more recent paper, instead of extracting the feature vectors directly from the image blocks, Li et al. proposed to decompose the image into four sub-bands using discrete wavelet transform (DWT) first [9]. They divided the low-frequency sub-band into overlapping blocks to reduce the number of blocks, and speed up the process, based on the fact that most of the energy would be concentrated at this sub-band. They applied singular value decomposition (SVD) on these blocks. Since SVD and PCAs are directly related to each other, one can expect that this method would perform similarly in similar situations as in [7]. However, they showed in their paper that the method was robust to compression only up to JPEG quality level 70.

Alternative to the features which were being used generally in image compression techniques, in [10], Luo et al. proposed to use features based on the color information of the blocks. The first set of features included the average of red, blue and green color components. In the second set the

blocks are divided into two parts, in 4 directions; the ratio of the one part's total intensity value over the block's total intensity value are calculated. Experimental results showed that this method was very robust to JPEG compression, up to the quality level of 30. The method was also robust against Gaussian blurring (5x5 window, $\sigma = 1$) and additive noise with SNR 24dB.

And recently, we proposed to apply Fourier Mellin Transform (FMT) on the image blocks [11]. For this purpose, we first obtained the fourier transform representation of each block, re-sampled the resulting magnitude values into log-polar coordinates. We obtained a vector representation by projecting log-polar values onto 1-D and used these representations as our features. These features were previously used in the context of watermarking by Wu et al. [12] and shown to be rotation, scale and translation invariant (RST). In our experiments, as well, we showed them to be robust to compression up to JPEG quality 20, rotation with 10° and scaling by 10%.

2.2. Matching the Duplicated Blocks

After finding robust representations for each block, the blocks that have same/similar representations have to be determined in a reasonable time. For this purpose, the papers which was mentioned in Section 2.1, proposed to use a common matching step, which includes lexicographically sorting. In this step, the new representation of each block is vectorized and inserted into a matrix A , where the rows of the matrix correspond to the blocks and columns of the matrix correspond to the features. In [6],[7], [10],[11], this matrix would consist of $(M - b + 1) \times (N - b + 1)$ rows and F columns, where F is the number of features. However, since in [9], the blocks were obtained by dividing the low-frequency sub-band of DWT of the image, there would be only $(M/4 - b + 1) \times (N/4 - b + 1)$ rows. If two blocks in the image are similar, their feature vectors therefore corresponding rows in matrix A would be similar as well and if the rows of A matrix is sorted lexicographically these feature vectors would come successively. To be more clear, we can say that the corresponding blocks whose feature vectors come successively in the matrix A would be the candidates of block duplicates.

The biggest challenge in this step is the computation time. For lexicographically sorting this time depends on the number of blocks, e.g. number of rows in the matrix A , mostly. While the number of features is also a factor for the speed, since it is generally a small number, one doesn't need to consider it. Lexicographically sorting requires $R \log_2(R)$ steps for a matrix which has R rows ($R = (M - b + 1) \times (N - b + 1)$ in [6],[7], [10],[11] and $R = (M/2 - b + 1) \times (N/2 - b + 1)$ in [9]). Therefore, the method described in [9] would require 4 times fewer step than the other methods.

In our previous paper, we described another method for matching the blocks which have the same feature vectors [11].

This method was based on counting bloom filters which essentially compare the hashes of features instead of features themselves. To realize counting bloom filters, we first formed an array K with k elements that are all initialized to zero. We hashed the feature vector of each block so that each hash value would indicate an index number in the array K . We incremented the value of the corresponding element in K by one after each hashing operation. Since the identical feature vectors would give the same hash values, corresponding index numbers would be the same and the corresponding element would be more than one in that case. As a result, the matching blocks can be detected checking which blocks set the elements of array K higher than 2. If we examine the computational complexity of this method, we can see that, since the hashing operation will be executed in the same step as feature extraction, time added by this scheme would be only due to finding the elements which have values more than 2. Obviously this step depends on the size of array K which can simply be chosen close to the value $M \times N$.

2.3. Forgery Decision

Matching similar blocks is not enough since most of the natural images would have many similar blocks. The forgery decision can be made only if there are more than a certain number of blocks that are connected to each other within a same distance. The distance between the two blocks that have the similar feature vectors, a_i and a_j , whose starting positions are (x_i, y_i) and (x_j, y_j) respectively, can be calculated as follows:

$$dx(i, j) = |x_i - x_j|; dy(i, j) = |y_i - y_j| \quad (1)$$

Note that for the lexicographically sorting approach a_i and a_j indicate the blocks which are coming successively in matrix A and for the bloom filter approach a_i and a_j indicate the blocks whose feature vectors yield to the same hash value. Following this, a distance vector D is constructed and the value of D is incremented by one every time same distance between two rows is calculated:

$$D(dx, dy) = D(dx, dy) + 1 \quad (2)$$

The values of D are initialized to zero at the beginning. If there are many blocks which gives the similar feature values in the same distance, at least one of the values of $D(dx, dy)$ should be more than a threshold value. The forgery decision can be made if these blocks are connected to each other.

3. PERFORMANCE EVALUATION

In order to evaluate and compare the performance of the copy-move forgery detection methods, we picked three methods where DCT features[6], PCA features[7], and FMT features[11] are used and we implemented these methods. As block size,

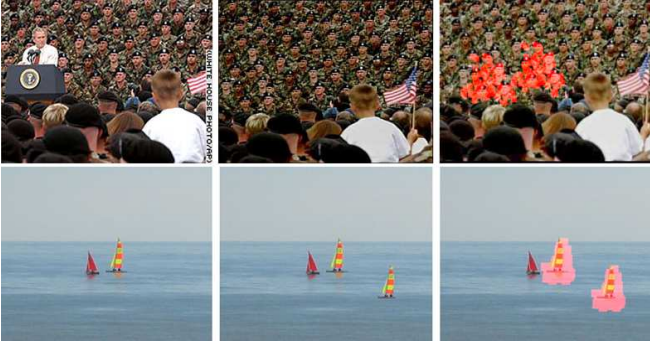


Fig. 2. Each row represent different images and for each row the first column is the original image, second column is the forged image and the last column is the detection result with our algorithm.

we used $b = 16$. We assumed that the duplicated region would be at least 32×32 . In this case, the image should contain $(32 - 15 + 1) \times (32 - 15 + 1) = 289$ duplicated blocks. Considering the blocks which would be affected by the image processing operations we chose the threshold value as 150. While comparing these methods, we used lexicographically searching in our matching step.

When the copied region is moved without applying any modifications, all of the three methods were able to detect the forgery very accurately. In figure 2, we presented the results with FMT features on two different images downloaded from the Internet.

To see how these methods perform under the modifications, we designed another set of experiments, where we used Lena image to illustrate copy-move forgery. First, we copied and moved a small block (size 32×32) in Lena and obtained several images by saving this image with various JPEG quality levels. In another scenario, we performed rotation and scaling operations before pasting the region we copied. The robustness of the three methods against these operations can be seen in Table 1. In this table, the numbers show, up to what value the methods are robust. From the table we can see that our method outperforms the other methods for these modification types. As expected, PCA values are changed when there is re-sampling but they are robust until JPEG 50. The performance of our method on sample images can be seen in Figure 3.

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

In the next step, we used bloom filters to detect the blocks which yields to the same features. We used the same Lena images as the previous experiment and we only considered



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 fourth image..



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

FMT features. In this case, the computational time is reduced from 25 seconds to 2 comparing with lexicographically sorting. However, the robustness of the scheme is reduced as well. So far, we only tested bloom filters on JPEG compressed images. As seen in the example in Figure 4, the scheme was robust against quality level 70 at most. For quality level 60, since the number of connected blocks did not reach the threshold value we set, the forgery decision is not made by the system.

On the other hand, these results show us that we can improve the efficiency of copy-move forgery techniques by using counting bloom filters. Especially when the image quality is high, the user should use bloom filters, otherwise to be able to catch the modifications it is better to use lexicographically sorting.

4. CONCLUSION

The copy-move forgery detection is one of the emerging problems in the field of digital image forensics. Many techniques have been proposed to address this problem. One of the biggest issue these techniques had to deal with was, being able to detect the duplicated image regions without getting affected by the common image processing operations, e.g. compression, noise addition, rotation. The other challenge was computational time, which becomes important considering the large databases, these techniques would be used on. In this paper, we reviewed several methods, where the main idea in common was to detect connected block duplicates, instead of detecting the whole duplicated regions. We explained the common

steps of these methods in detail, and discussed the robustness properties and time complexity of each method. Furthermore, for better evaluation the methods we performed several different experiments and presented comparative results. While the performance of the methods were satisfactory enough, the computational time was still an issue to be improved. Ultimately, faster detection techniques which meet the robustness criteria have to be designed.

5. REFERENCES

- [1] Bayram S., Avcibas I., Sankur, and B. Memon N., "Image manipulation detection," *Journal of Electronic Imaging – October - December 2006 – Volume 15, Issue 4, 041102 (17 pages)*, vol. 15(4), 2006.
- [2] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53(2), pp. 758–767, 2005.
- [3] M.K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," *Proc. ACM Multimedia and Security Workshop, New York*, pp. 1–9, 2005.
- [4] M. Wu A. Swaminathan and K. J. Ray Liu, "Image tampering identification using blind deconvolution," *Proc. IEEE ICIP*, 2006.
- [5] Sencar H. T. Memon N. Sutcu Y., Coskun B., "Tamper detection based on regularity of wavelet transform coefficients," *Proc. ICIP, International Conference on Image Processing*, 2007.
- [6] J. Fridrich, D. Soukal, and J. Luk, "Detection of copy-move forgery in digital images," *Proc. Digital Forensic Research Workshop, Cleveland, OH*, August 2003.
- [7] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Technical Report, TR2004-515, Dartmouth College, Computer Science*, 2004.
- [8] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, 1991.
- [9] Qiong Tu Dan Sun Shaojie Li, Guohui Wu, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd," *ICME*, 2007.
- [10] Weiqi Luo, Jiwu Huang, and Guoping Qiu, "Robust detection of region-duplication forgery in digital image," in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, Washington, DC, USA, 2006, pp. 746–749, IEEE Computer Society.
- [11] Sevinc Bayram, Taha Sencar, and Nasir Memon, "An efficient and robust method for detecting copy-move forgery," *submitted to ICASSP 2009*, 2009.
- [12] J. A. Bloom I. J. Cox M. L. Miller C. Y. Lin, M. Wu and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Processing*, vol. 10, pp. 767–782, 2001.