A Non-intrusive Method for Copy-Move Forgery Detection

Najah Muhammad[1], Muhammad Hussain[2], Ghulam Muhamad[2], and George Bebis[2,3]

[1] CCIS, Prince Norah Bint Abdul Rahman University
[2] CCIS, King Saud University, Saudi Arabia
[3] CSE, University of Nevada, Reno, USA
nagahsubaie@yahoo.com, mhussain@ksu.edu.sa, bebis@cse.unr.edu

Abstract. The issue of verifying the authenticity and integrity of digital images is becoming increasingly important. Copy-move forgery is one type of image tempering that is commonly used for manipulating digital content; in this case, some part of an image is copied and pasted on another region of the image. Using a non-intrusive approach to solve this problem is becoming attractive because it does not need any embedded information, but it is still far from being satisfactory. In this paper, an efficient non-intrusive method for copy-move forgery detection is presented. The method is based on image segmentation and a new denoising algorithm. First, the image is segmented using a multi-scale segmentation algorithm. Then, using the noise pattern of each segment, a separate noise image is created. The noise images are used to estimate the overall noise of the image which is further used to re-estimate the noise pattern of different segments. The image segments with similar noise histograms are detected as tampered. A comparison with a state-of-the art non-intrusive algorithm shows that the proposed method performs better.

1 Introduction

Due to recent advances in imaging technologies, it has become very easy to preserve any event in the form a digital image, and this digital pictorial information is being used widely for multiple purposes. On the other hand, due to the development of sophisticated editing software, even a novice person can tamper with the digital contents with an ease. Authenticity of images cannot be taken for granted. The issue of verifying the authenticity and integrity of digital contents is increasingly becoming important. This motivated the need of techniques which can be used to validate the authenticity of digital content.

The existing techniques for forgery detection can be classified into two main categories: intrusive and non-intrusive. Intrusive techniques required that some sort of digital signature is embedded in the image at the time of its creation. Therefore, their scope is limited because not all digital devices have the capability of embedding a digital signature at the time of capturing an image. On the other hand, non-intrusive approaches do not require embedding any information. Though non-intrusive approach is attractive, and some work has been done in this direction, research on this approach is still in its infancy, and more efforts are required for proposing stable solutions.

G. Bebis et al. (Eds.): ISVC 2011, Part II, LNCS 6939, pp. 516–525, 2011.

Copy-move forgery is one type of tempering that is commonly used for manipulating digital content; in this case, a part of an image is copied and pasted on another region of the image. In this paper, the focus is on detecting copy-move forgery. The task of tamper detection becomes more difficult with copy-move forgery because the copied region will have the same characteristics of the image such as noise component, color palette, dynamic range etc. This indicates that detection methods that search for tampered image regions using inconsistencies in statistical measures will fail.

There are a number of methods that provide solutions for copy-move forgery detection. Each of these methods provides a solution under a set of conditions or assumptions; the method will fail if its assumptions are not realized [3, 9, 13, 14].

In this paper, we present preliminary results on a new algorithm for copy-move forgery detection. Our solution is based on the idea that copied and pasted regions must have the same noise pattern. The proposed solution depends on image segmentation and noise estimation for each segment. The noise patterns of the image segments are then compared for identifying forgery. Image segments with similar noise patterns are detected as tampered. The proposed method outperforms state-of-the art methods.

The rest of the paper is organized as follows. The next section discusses published work related to copy-move forgery detection. In Section 3, the proposed method is explained. Section 4 contains the experimental results for the proposed method. In Section 5, we discuss our results and Section 6 concludes the paper.

2 Related Work

This section gives an overview of non-intrusive methods dealing with copy-move forgery. The most commonly used non-intrusive approach for copy-move forgery detection is based on block matching. In block based methods, an image is partitioned into equal sized blocks, and tempering is detected using feature similarities between image blocks. The features of each block are extracted to form a feature vector. The feature vectors are then sorted so that similar vectors are grouped together and neighboring information is analyzed; a similarity threshold is set based on experiments. Similar feature vectors indicate that their corresponding image blocks are copies of each other.

In [3], a detection method based on matching the quantized lexicographically sorted discrete cosine transform (DCT) coefficients of overlapping image blocks has been proposed. Experimental results show reliable decisions when the retouching operations are applied. However, the authors don't show robustness tests.

Another method which is invariant to the presence of blur degradation, contrast changes and additive Gaussian noise is presented in [11]. Features of the image blocks are represented by a blur using moment invariants. The experimental results show that the algorithm performs well with the blurring filter and a lossy JPEG compression quality down to 70. However, like other similar methods, this algorithm may falsely label unmatched areas as matched. This problem arises in case of uniform regions such as sky. Another disadvantage of this algorithm is its computational time. The average running time of the algorithm with block size of 20, a similarity threshold 0.97 and image size of 640×480 RGB image, using a processor of 2.1 GHz and 512 MB RAM is 40 minutes.

In [6], Singular Value Decomposition is used to obtain singular values feature vectors for block representation. The feature vectors are sorted using lexicographical sort. The experimental results show that the algorithm is comprehensive. It has been shown that the algorithm performs well even in images with uniform areas such as sky and ocean. The running time with one color channel of 256×256 images running on a 1.8 GHz processor and 256MB RAM when block size is 20, is approximately 120 seconds.

In [4], SIFT features has been used. The experimental results show that about 38 matches can be reached if the threshold for Euclidean distance between the matched descriptor vectors is set to 0.45.

The algorithm proposed in [16] is based on pixel matching to detect tampering. The approach uses the Discrete Wavelet Transform (DWT) to get reduced data representation. Also, phase correlation is used to compute the spatial offset between the copied and pasted regions in the image.

The work in [1] uses a feature representation that is invariant not only to noise addition or blurring, but also to several geometric transformations such as scaling and rotation that may be applied to the copied region before pasting. These properties are achieved using Fourier-Mellin Transform (FMT) feature representation. The counting bloom filter [1] is used instead of lexicographic sorting to improve time complexity. Their experimental results showed that the proposed representation is more robust to JPEG compression. Furthermore, it can deal with rotations up to10°.

The method proposed in [8] divides each image into overlapping blocks of equal size, and represents each block with nine features, which are normalized to integers in the range [0, 255]. Then a counting sort [5] is used to sort the feature vectors. Experimental results show that about 98% of detection rates can be achieved with different sets of 50 images with/without modifications such as compression and Gaussian noise. This method can detect a copy-move forgery with rotation.

The most important issues in the block matching approach are: the block feature representation and the sorting algorithm. A robust feature extraction method must be employed that is insensitive to different types of post-processing and involves the lowest complexity. In addition, the sorting algorithm must have the lower run time complexity.

3 Proposed Method

Copy-move tampering is done by copying a region of the image and pasting it on another place in the same image. Blurring may be applied to hide borders and to integrate the pasted region with the image background. When a region is copied and pasted to another place, it will keep some of its underling features that can be used to indentify tampering. The feature used here is the noise pattern.

Specifically, we studied the noise present in an image and found that an original image has different noise patterns associated with regions related to different objects. However, a tampered image, where one of its regions is replicated, will have almost the same noise pattern for both the copied and pasted parts. The general framework of our algorithm is as follows:

Step-1: Segment the input image.
Step-2: Using the segmented image, estimate image noise.
Step-3: Analyze noise pattern of each segment.
Step-4: The image is tampered if the noise patterns of at least two segments are similar

In the following subsections, we elaborate on each step.

3.1 Image Segmentation

The input image ***I*** of size $m \times n$ is segmented into j segments S_1, S_2 ,..., S_j. in such a way that each object is fully contained in a single segment, and the segment is almost homogeneous. For this purpose, we used the algorithm presented in [18]. This algorithm works on multiple scales of the image in parallel, without iteration, to capture both coarse and fine level details. This segmentation algorithm works simultaneously across the graph scales, with an inter-scale constraint to ensure communication and consistency between the segmentations at each scale.

3.2 Image Noise Estimation

Though many different denoising algorithms exist in the literature, which can be used for noise estimation, each algorithm focuses on a particular type of noise. So these methods do not serve our purpose. We present a new noise estimation method, which can be useful for other applications as well. Our idea of denoising and noise estimation is based on the following well-known fact [19]:

"If the image $g(m, n)$ is formed by averaging the noisy versions $f_i(m,n)$ of an image $f(m,n)$, then $g(m,n)$ approaches $f(m,n)$ when the number of noisy versions $f_i(n, m)$ is sufficiently large".

We use segments of an image to generate its noisy versions. The detail of the noise estimation algorithm is given below.

Step-1: For each segment S_i, compute the average gray level g_i as follows:

$$g_i = \frac{\sum_{x=1}^{l_i} S_i(x)}{l_i} \tag{1}$$

where l_i is the size of S_i i.e. the total number of pixels in S_i and $S_i(x)$ is the intensity of the pixel x. Each segment is represented as a vector.

Step-2: Subtract the average gray level g_i from each pixel value of S_i, the result is the noise segment S_i' because each segment S_i is homogeneous.

Step-3: For each noise segment S_i', create a noise image $\boldsymbol{I}_i$ of size $m \times n$ (see Sub-section 3.2.1)

Step-4: For each noise $\boldsymbol{I}_i$, subtract $\boldsymbol{I}_i$ from $\boldsymbol{I}$ to get the denoised image $\boldsymbol{I}_i^d$ i.e.

$$\boldsymbol{I}_i^d = \boldsymbol{I} - \boldsymbol{I}_i \tag{2}$$

Step-5: Find the average of the denoised images:

$$\boldsymbol{I}^d = \frac{1}{j} \sum \boldsymbol{I}_i^d \tag{3}$$

Step-6: Estimate the noise by subtracting the average denoised image $\boldsymbol{I}^d$ from the given image:

$$\boldsymbol{NL} = \boldsymbol{I} - \boldsymbol{I}^d \tag{4}$$

NL is noise estimation of the image and is used to analyze the noise pattern corresponding to each segment.

3.2.1 Creating a Full Noise Image from Noise Segment

In this section, we give the details of our method for creating a full noise image from each segment. The main steps of the algorithm are as follows:

Step-1: For each segment S_i', create a list L_i of the pixel values in $\mathrm{S_i'}$.
Step-2: Create an image $\boldsymbol{I}_i$ of size $m\times n$ containing random numbers where each random number *r* is an integer between 0 and l_i-1, where l_i is the number of pixels in $\mathrm{S_i'}$.
Step-3: Replace each random number *r* in $\boldsymbol{I}_\mathrm{i}$ with the value from list L_i at index *r*.

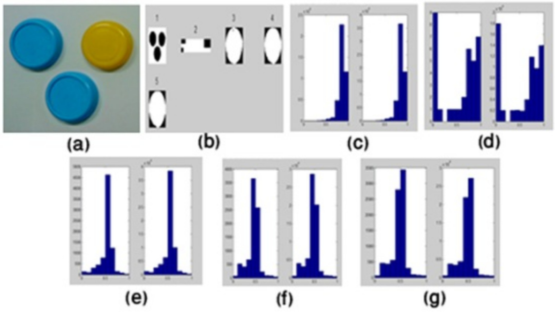


Fig. 1. (a) The input image, (b) its segmentation into five segments; (c, d, e, f, g) are the pairs of histograms corresponding to the noise segments $\mathrm{S_i}'$, i=1, 2, 3, 4, 5 and the respective noise images I_i, i=1, 2 ,3, 4, 5

This method creates a full noise image from a given segment having same statistical properties as of the segment. As noise is a random signal and I_i is created using a random process from the given segment, then the noise in $\boldsymbol{I}_i$ and the segment will have the same statistical properties. This assertion is further validated by the histograms of the segments and those of the corresponding noise images as shown in Fig. 1. The figure shows that the histogram of a segment and that of its corresponding noise image are similar.

3.3 Analyzing Noising Pattern for Each Image Segment

For each segment S_i obtained in the image segmentation step, the corresponding segment NS_i is extracted from the estimated noise image $\boldsymbol{NL}$. To analyze the noise pattern of NS_i, we compute its histogram h_i with *l* bins and calculate the probabilities

$$p_i(x) = \frac{h_i(x)}{l} \tag{5}$$

from the histogram $h_\mathrm{i.}$

We found that the histograms of the copied and pasted segments are almost the same.

3.4 Detect Tampered Regions

The estimated noise patterns corresponding to the copied and the pasted regions have the similar histograms. In view of this, detecting the tampered regions is equivalent to checking the similarity of the corresponding histograms. Many different methods can be used to test the similarity of histograms. We employed the following simple statistical measures:

Moments of first and second order

$$m_i = \sum_{x=0}^{L} x^j p_i(x), \qquad j = 1,2 \tag{6}$$

Central moments of order up to 4

$$\mu_i = \sum_{x=0}^{L} (x - m_1)^j p_i(x), \qquad j = 1,2,3,4 \tag{7}$$

Absolute central moments of order up to 4

$$\hat{\mu}_i = \sum_{x=0}^{L} | x - m_1 |^j \; p_i(x), \qquad j = 1,2,3,4 \tag{8}$$

After performing several tests, we found that the similarity of the histograms can be detected using central moment of the second order. As such, we selected it to be the measure of the histogram similarity.

4 Results and Comparison

We tested the performance of our proposed method on a number of forged images, and found encouraging results. In this section, we present test results on 4 images, which are shown in Figures 2~5. For each image, we choose the minimum number of segments that can segment the image objects correctly.

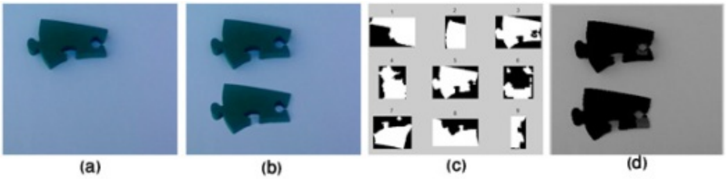


Fig. 2. (a) Original image, (b) Tampered image, (c) Segments of the image in (b) white region corresponds to a segment, (d) detected tampered region

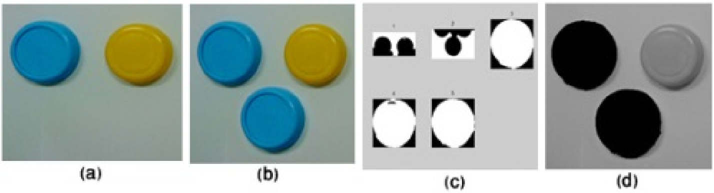


Fig. 3. (a) Original image, (b) Tampered image, (c) Segments of the image in (b) white region corresponds to a segment, (d) detected tampered regions

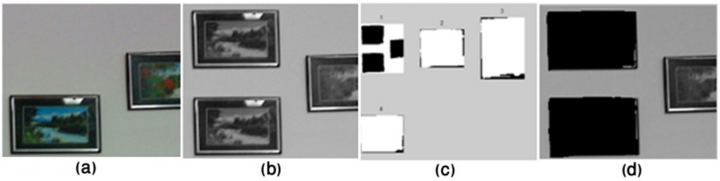


Fig. 4. (a) Original image, (b) tampered image, (c) segments of the image in (b) white region corresponds to a segment, (d) detected tampered regions

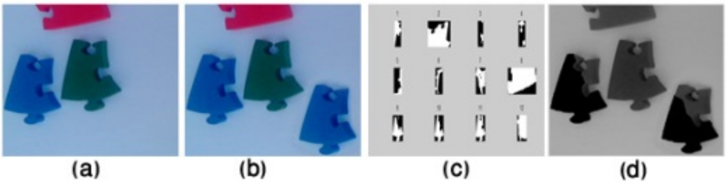
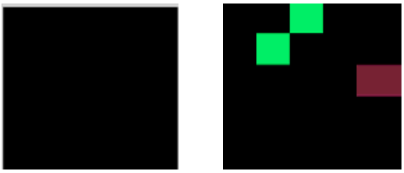


Fig. 5. (a) Original image, (b) Tampered image, (c) Segments of the image in (b) white region corresponds to a segment, (d) Detected tampered regions

Fig. 2(b) is the tampered image where the upper object has been copied and pasted in the lower part. Fig. 2(c) shows the segmentation of the tampered image, here each white region corresponds to a segment. Fig. 2(d) shows the result of the detection process. The central moments for each segment are shown in the second row of Table 1; it is obvious that copied and pasted segments 3 and 5 have similar central moments and are detected correctly.

Fig. 3(b) is the tampered image where the blue cap in the top row has been copied and pasted in the lower row. Fig. 3(c) shows the segmentation of the tampered image and Fig. 3(d) shows the result of the detection process. The central moments for all segments are shown in the fourth row of Table 1; one can see that copied and pasted segments 3 and 5 have been detected correctly.

Fig. 4(b) is the tampered image where the picture in the lower part has been copied and pasted in the upper part. Fig. 4(c) shows the segmentation of the tampered image and Fig. 4(d) display the result of the detection process. The central moments for all segments are shown in the sixth of Table 1and it is clear that tampered segments 2 and 4 are detected correctly.



(a) B=40, T=1

(b) B=20, T=0.5

Fig. 6. The detection result for the tempered image shown in Fig.4 using blocks size B and a similarity threshold T. In (a) we used the same parameters values which have been used in [12]. In (b) we changed the parameter values.

Table 1. Central Moments

Seg.#	1	2	3	4	5	6	7	8	9	10	11
Image 1 (Figure 2)											
	251.4	132.66	55.44	2.8	55.44	103.0	13.2	130.6	183.0		
Image 2 (Figure 3)											
	324.16	87.04	5.04	5.44	5.04						
Image 3 (Figure 4)											
	8.960	2.40	8.24	2.40							
Image 4 (Figure 5)											
	29.44	136.00	24.96	1.360	7.360	95.44	15.04	69.04	100.96	100.96	62.64

Fig. 5(b) is the tampered image where the blue object has been copied and pasted on the right side of green object. Fig. 5(c) shows the segmentation of the tampered image and Fig. 5(d) shows the result of the detection process. The central moments for each segment are shown in the eighth row of Table 1 and it is obvious that tampered segments 9 and 10 are detected correctly.

We also compared our method with a recent non-intrusive forgery detection method presented by Babak et al. [12]; this method partitions an image into equal size rectangular blocks and uses Discrete Wavelet Transform (DWT) to estimate the image noise for detecting image tampering. The noise feature used by them is MAD, median absolute deviation, which is employed to measure the noise inconsistency between blocks. If there is no noise inconsistency across all the blocks, then it is original, otherwise it is tempered. We applied their algorithm using our test images; the results show that our algorithm can produce more precise and clear results. For example, Fig. 6 shows the detection results of the tampered image depicted in Fig. 4; in this figure, the regions with homogenous noise level are shown in black while other regions are assigned random colors. Fig. 6(a) implies that the noise level is consistent over the entire image and there is no tampering but this is a false result. For Fig. 6(b), the green (and similarly pink) regions represent the places where the noise level is not consistent; the green region partially detects the tampered region whereas pink region is a false detection.

5 Discussion

The proposed algorithm represents a promising non-intrusive algorithm for copy-move forgery detection, which is based on the analysis of noise pattern. Other methods that use image noise for forgery detection are proposed in [2, 7, 10, and 17]; some of them require training a classifier with hundreds of images from several cameras. These algorithms can detect tampering in images captured by the same camera used to capture the training images. Because of that, these algorithms require previous knowledge about the camera used, which is not always available. However, our proposed algorithm finds the replicated regions in an image without any previous knowledge about the camera used to capture the image.

The proposed algorithm is affected by the segmentation of the image. It can provide better results with segmentation algorithm that can segment an image into complete objects more accurately. As it can be observed from Fig. 5, the segmentation algorithm divides a single object into several parts. The unequal parts in both the copied and pasted objects in the image will have different statistical features and hence, will result in false detection.

6 Conclusion

We have studied a challenging problem in digital image forgery detection. In this paper, we presented the initial finding of our study. We proposed a new algorithm that can effectively detect tampering in an image without requiring any knowledge about the camera used to capture the image.

So far, we have tested our algorithm on images where the background is simple. We will explore it further for images with complicated background and texture. This will require a more robust and reliable segmentation algorithm. Second, besides using histograms, we will investigate using more robust features for representing noise patterns and being able to differentiate between the tampered and un-tampered segments.

Acknowledgment. This work is supported by the National Plan for Science and Technology (NPST), King Saud University, Riyadh, Saudi Arabia under the project 10-INF1140-02. We are thankful to B. Mahdian and co-workers, the authors of the work in [12], for providing their code to compare their results with ours.

References

1. Bayram, S., Sencar, H.T., Memon, N.: An Efficient and Robust Method for Detecting Copy-Move Forgery. In: Proc. IEEE ICASSP, pp. 1053–1056 (2009)
2. Chen, M., et al.: Determining Image Origin and Integrity Using Sensor Noise. IEEE Transactions on Information Forensics and Security, 74–90 (2008)
3. Fridrich, J., Soukal, D., Lukas, J.: Detection of Copy Move Forgery in Digital Images. In: Digital Forensic Research Workshop, Cleveland, OH (2003)
4. Huang, H., et al.: Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm. In: Pacific-Asia Workshop on Computational Intell. Industrial App., pp. 272–276 (2008)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn., Section 8.2: Counting sort, pp. 168–170. MIT Press and McGraw-Hill (2001)
6. Kang, X., Wei, S.: Identifying Tampered regions using singular value decomposition in Digital image forensics, pp. 926–930. IEEE Computer Society, USA (2008)

7. Li, Y., Li, C.-T.: Decomposed Photo Response Non-Uniformity for Digital Forensic Analysis. In: Sorell, M. (ed.) e-Forensics 2009. LNICST, vol. 8, pp. 166–172. Springer, Heidelberg (2009)
8. Lin, H.-J., Wang, C.-W., Kao, Y.-T.: Fast Copy-Move Forgery Detection. WSEAS Trans. Signal Process, 188–197 (2009)
9. Lin, H.J., Wang, C.W.,, Y.: Fast Copy-Move Forgery Detection. In: World Scientific and Engineering Academy and Society (WSEAS), pp. 188–197 (2009)
10. Lukáš, J., et al.: Detecting Digital Image Forgeries Using Sensor Pattern Noise. In: Proc. of SPIE (2006)
11. Mahdian, B., Saic, S.: Detection of Copy Move Forgery Using a Method Based on Blur Moment Invariants. Forensic Science International 171, 180–189 (2007)
12. Mahdian, B., Saic, S.: Using Noise Inconsistencies for Blind Image Forensics. Image and Vision Computing, 1497–1503 (2009)
13. Popescu, A.C., Farid, H.: Exposing Digital Forgeries by Detecting Duplicated Image Regions. Dept. Comput. Sci., Dartmouth College,Tech.Rep. TR2004-515 (2004)
14. Sutcu, Y., et al.: Tamper Detection Based on Regularity of Wavelet Transform Coefficients. In: Proc. IEEE ICIP, pp. 397–400 (2007)
15. Wang, J., et al.: Detection of Image Region Duplication Forgery Using Model with Circle Block. In: International Conf. Multimedia Inform. Network. and Security, pp. 25–29 (2009)
16. Zhang, J., Feng, Z., Su, Y.: A New Approach for Detecting Copy-Move Forgery in Digital Images. In: IEEE Singapore Int. Conf. Comm. Sys., China, pp. 362–366 (2008)
17. Zhang, P., Kong, X.: Detecting Image Tampering Using Feature Fusion. In: International Conference on Availability, Reliability and Security, pp. 335–340 (2009)
18. Cour, T., et al.: Spectral Segmentation with Multiscale Graph Decomposition. In: IEEE International Conference on Computer Vision and Pattern Recognition, CVPR (2005)
19. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. (2002)