An Efficient Method for Copy-Move Forgery Detection

HWEI-JEN LIN[1], CHUN-WEI WANG[1] and YANG-TA KAO[2]
[1]Department of Information Engineering and Computer Science
Tamkang University
[2]Department of Network Information Technology
Chihlee Institute of Technology
151 Ying-Chuan Road, Tamsui
Taipei, Taiwan
[1]086204@mail.tku.edu.tw, http://pria.cs.tku.edu.tw
[2]ydkao@mail.chihlee.edu.tw, http://int.chihlee.edu.tw

Abstract: - This paper proposes a method for detecting copy-move forgery over images tampered by copying some regions and pasting them onto other regions. To detect those forgeries, we divide the given image into overlapping blocks, extract feature for each blocks, and sort the extracted feature vectors by radix sort. The difference (shift vector) of the coordinates of every pair of adjacent vectors in the sorting list is computed. The accumulated number of each of the shift vectors is evaluated. Finally, the medium filtering and connected component analysis are performed to obtain the detected result. Compared with other methods, employing the radix sort makes the detection much more efficient without degradation of detection quality.

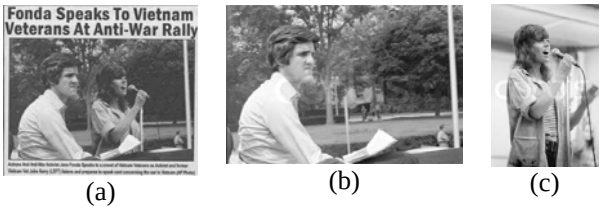
Key-Words: - forgery detection, copy-move, radix sort, connected component analysis, medium filtering.

1 Introduction

With the development of Internet and image processing techniques, images can be easily acquired through internet and tampered using some commonly available software, such as Photoshop and Photoimpact. As shown in Fig. 1, the forged image "*Fonda Speak To Veitnam Veterans At Anti-War Rally*" in Fig. 1(a) was synthesized using the images shown in Fig.s 1(b) and 1(c). For protecting the copyright and preventing forgery with a bad intention, methods for forgery detection have become more and more important.

Recently, many methods for detecting forged images have been proposed. Popescu [8] detected forgeries with linear interpolation, scaling or rotation based on the relationship between each pixel and its neighbors. Nillius et al. [2] and Johnson et al. [7] used light source consistency to detect forged images. Li et al. [10] detected tampered watermarked images with the embedded information and then recover the images.

Defects of cameras such as chromatic aberration and sensor pattern noise, and the color filter arrays the cameras use for interpolating colors can be used to detect forgeries [4][9][12]. Copy-move images are easily made by copying certain regions and pasting them on some other regions. The existing methods [1][5][6][11] for detecting such kind of forgeries are all time- consuming. In this paper, we shall propose a simple and fast method to detect copy-move forgeries. Compared with other methods, our algorithm is more efficient without degradation of detection rates. The rest of this paper is organized as follows. Related work is discussed in Section 2. In section 3, the proposed method is described in details. In Sections 4 and 5, we show some experimental results and make a conclusion for this paper.



(a) (b) (c)

Fig. 1 (a). A synthesized image "*Fonda Speak To Veitnam Veterans At Anti-War Rally*", (b)&(c). original images.

2 Related Work

In most methods of copy-move forgery detection, the detected image is divided into overlapping blocks, which are then represented as vectors, which are then lexicographically sorted for later detection. Suppose a detected image of size $N \times N$ is divided into $(N-b+1)^2$ overlapping blocks of size $b \times b$, which are represented as vectors of b^2 dimension, and sorted in a lexicographical order. Vectors corresponding to blocks of similar content would be close to each other in the list, so that identical regions could be easily detected. The image given in Figure 2(a) was tampered by copy-move forgery, as shown in Figure 2(b), in which block B_1, B_2, and block B_3 are copies of blocks A_1, A_2, and block A_3, respectively, and thus $V_{A1}=V_{B1}$, $V_{A2}=V_{B2}$, and $V_{A3}=V_{B3}$, where V_X denotes the vector corresponding to block X. As shown in Figure 2(c), identical vectors are adjacent in the sorted list, from which the copy-move regions could be easily detected. In the previously mentioned methods, the vectors were sorted by the lexicographical sort, which took $O(k \lg k)$ time to sort on each entry in the vectors, where $k = (N-b+1)^2$. The time complexity of lexicographical sorting on theses vectors is $O(b^2 k \lg k)$ when the vectors are of b^2. Farid et al. [6] reduced the time complexity to $O(32 \times k \lg k)$ by using PCA (principle component analysis).

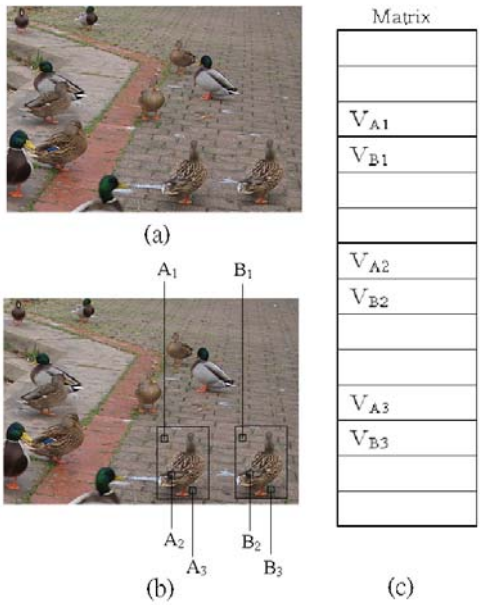


Fig. 2 (a). An original image, (b). Three pairs of identical blocks are marked by squares, (c). feature vectors corresponding to the divided blocks are sorting in a list.

The time complexity of the method proposed by G. Li et al. [1] was reduced to $O(8 \times k \lg k)$ by using SVD. W. Luo et al. [11] defined a feature vector of 7-dimension to represent blocks so as the time complexity is reduced to $O(7 \times k \lg k)$. In this paper, we shall propose a further efficient method for copy-move forgery detection.

3 The Propose Method

For resistance against various modifications and improving the sorting time, we represent each block B by a 9-dimensional feature vector $v_B = (x_1, x_2, \ldots, x_9)$, which is defined as follows. Firstly, the block B is divided into four equal-sized sub-blocks, S_1, S_2, S_3, and S_4, as shown in Fig. 3 and let *Ave*(.) denote the average intensity function. Then as described in (1), f_1 denotes the average intensity of the block B, the entries f_2, f_3, f_4, and f_5 denote the ratios of the average intensities of the blocks S_1, S_2, S_3, and S_4 to f_1, respectively, and f_6, f_7, f_8, and f_9 stand for the differences of the average intensities of the blocks S_1, S_2, S_3, and S_4 from f_1, respectively. Finally, entries f_i's are normalized to integers x_i's ranging from 0 to 255, as described in (2). Although these 9 entities contain duplicated information, they together possess higher capability for modification resistance, such as JPEG compression and Gaussian noise.

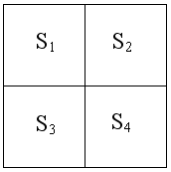


Fig. 3 A block is divided into four equal-sized sub-blocks.

$$f_i = \begin{cases} f_i = Ave(S) & \text{if } i = 1, \\ Ave(S_{i-1})/(4\,Ave(S) + \varepsilon_1) & \text{if } 2 \le i \le 5, \\ f_i = Ave(S_{i-5}) - Ave(S) & \text{if } 6 \le i \le 9. \end{cases} \quad (1)$$

$$x_i = \begin{cases} \lfloor f_i \rfloor & \text{if } i = 1, \\ \lfloor 255 \times f_i \rfloor & \text{if } 2 \le i \le 5, \\ \left\lfloor 255 \times \dfrac{f_i - m_2}{m_1 - m_2 + \varepsilon_2} \right\rfloor & \text{if } 6 \le i \le 9, \end{cases} \quad (2)$$

where $m_1 = \max_{6 \le i \le 9} \{f_i\}$ and $m_2 = \min_{6 \le i \le 9} \{f_i\}$.

Unlike the matrix constructed by Farid et al. [6], which stores floating numbers (the PCA coefficients), the feature vectors we extract store integers. As a result, we may use the efficient radix sort algorithm to perform lexicographical sorting over those vectors. If the given image of size $N \times N$ is divided into overlapping blocks

of size $b \times b$, then there are totally k blocks, where $k = (N-b+1)^2$. Let $v_1, v_2, \ldots, v_k$ be the feature vectors corresponding to these k blocks. To perform radix sort on these vectors of 9 dimensions, we regard each of them as a 9-digit number with each digit ranging from 0 to 255. The sorting algorithm is given in the following, where the input array A stores these vectors; that is, $A[i] = v_i, 1 \le i \le k$, and $d = 9$.

RADIX-SORT(A,d)
for $i \leftarrow 1$ **to** d
do use a stable sort to sort array A on digit i

Since each digit is in the range 0 to 255, which is not large, counting sort is chosen as the stable sort used in our radix sort. Each pass over k numbers then takes time O(256+k). There are 9 passes, so the total time for sorting the feature vectors is O(9(256+k)) = O(9k) since 256 << k.

The position of the top-left corner point of each block B is recorded in $P(v_B)$ for later use. From the sorted list of the feature vectors, we detect the copy-move regions by examining the accumulated number of each shift vector. A shift vector is defined as the difference of two adjacent feature vectors in the sorted list as shown in (3).

$$u(i) = P(v_{i+1}) - P(v_i) \qquad (3)$$

For the number of a shift vector greater than a given threshold T_1, the top-left points of all the corresponding blocks are marked. For example, if the accumulated number of a shift vector u_0 is greater than T_1, then for each i, the top-left points of the respective blocks corresponding to v_i and v_{i+1} are marked if $u(i) = u_0$. Fig. 4(a) shows the result of marked points for Fig. 2(b). To reduce the false alarms, we delete the shift vectors with a small accumulated amount (less than a threshold T_2). Finally, the medium filtering is performed to remove noises and the connected component analysis is applied to obtain the final detected result as given in Fig. 4(b)

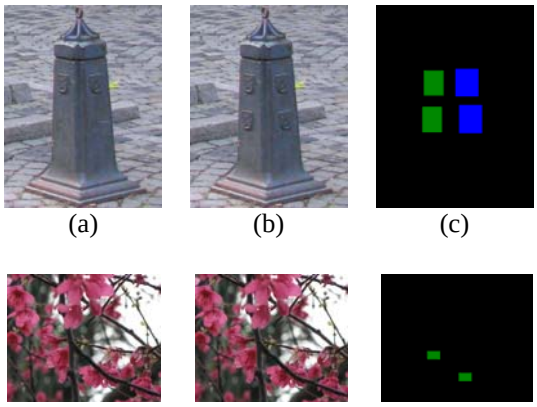
(a) (b)

Fig. 4 (a). Corner points of some blocks are marked according to the accumulated numbers of shift vectors for the tampered image given in Fig. 2(b), (b). final detected result.

4 The Experimental Results

The proposed method was implemented on a computer of CPU 3.0GHz with memory 1GB. The test images were cropped from 50 natural images. We tested over 50 tampered images, 150 compressed tampered images, and 150 tampered images with Gaussian noise. For detecting on color images, only the green channel is used since the human eyes are most sensitive to the green color. For parameter setting, we set b = 16, T_1 = 100, and T_2 = 10.

More detected results over tampered images are shown in Fig. 5. Fig. 6 shows the detected results over compressed tampered images with various quality factors. Fig. 7 shows the detected results over some images with Gaussian noise at various SNRs (signal to noise ratios). Detection rates for some datasets of copy-move images under various modifications are shown in Table 1.



(a) (b) (c)

(d) (e) (f)

Fig. 5 Detected results over tampered images: (a) & (d). original images, (b) & (e). tampered images, (c) & (f). detected results.

(a) (b) (c)

Fig. 6 Detected results over compressed versions of the image given in Fig. (2a), with various quality factors (QFs): (a). QF = 90, (b). QF = 70, (c). QF = 50.

(a) (b) (c)

Fig. 7 Detected results for the image given in Fig. (2a) with Gaussian noise at various SNRs: (a). SNR=10db, (b). SNR=20db, (c). SNR=35db.

Data sets of Copy-move images	No. of images	Detection rate (%)
without midification	50	98
JPEG compression QF = 100	50	98
JPEG compression QF = 90	50	98
JPEG compression QF = 80	50	96
Gaussian noise SNR = 10	50	98
Gaussian noise SNR = 20	50	98
Gaussian noise SNR = 35	50	94

Table 1 Detection rates for datasets of copy-move images.

4 Conclusion

In this paper, we propose an efficient method for copy-move forgery detection. Using of radix sort dramatically improves the time complexity and the adopted features enhance the capability of resisting of various attacks such as JPEG compression and Gaussian noise. Both efficiency and high detection rates have been demonstrated in our experimental results. However, a few small copied regions were not successfully detected. In the future, we would like to extend our work to video images.

References:

[1] 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," *in Proceedings of IEEE International Conference on Multimedia and Expo*, Beijing China, 2-5 July 2007, pp. 1750-1753.

[2] Peter Nillius and Jan-Olof Eklundh, "Automatic estimation of the projected light source direction," *in Proceedings of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2001, pp. 1076- 1083.

[3] Wei Lu, Fu-Lai Chung, and Hongtao Lu, "Blind fake image detection scheme using SVD," *IEICE Transaction on Communications*, Vol. E89-B, No. 5, May 2006, pp. 1726-1728.

[4] J. Lukas, J. Fridich, and M. Goljan, "Detecting digital image forgeries Using sensor pattern noise," *in Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, Vol. 6072, January 2006, pp. 362-372.

[5] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," *in Proceedings of Digital Forensic Research Workshop*, August 2003.

[6] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Technical Report, TR2004-515, Dartmouth College*, Computer Science 2004.

[7] M.K. Johnson and H. Farid, "exposing digital forgeries by detecting inconsistencies in lighting," *in Proceedings of ACM Multimedia and Security Workshop*, New York, 2005, pp. 1-9.

[8] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, Vol. 53, 2005, pp. 758-767.

[9] A.C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, Vol. 53, 2005, pp. 3948–3959.

[10] Kwang-Fu Li, Tung-Shou Chen, and Seng-Cheng Wu, "Image tamper detection and Recovery System Based on Discrete Wavelet Transformation," *Intelligent Conference Communications, Computers and Signal Processing*, Vol. 1, pp. 26-28, 2001.

[11] Wwiqi Luo, Jiwu Huang, and Guiping Qiu, "Robust detection of region- duplication forgery in digital image," *in Proceedings of the 18th International Conference on Pattern Recognition*, Vol. 4, 2006, pp. 746-749.

[12] N. Khanna, A.K. Mikkilineni, G.T.C. Chiu, J.P. Allebach, and E.J. Delp, "Scanner identification using sensor pattern noise," *in Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, Vol. 6505, No. 1, 2007, pp. 65051K.