Robust Detection of Region-Duplication Forgery in Digital Image

Weiqi Luo, Jiwu Huang
GuangDong Key Lab. of Information Security
Sun Yat-Sen University,Guangzhou,China,510275
isshjw@mail.sysu.edu.cn

Guoping Qiu
School of Computer Science
University of Nottingham, NG8 1BB, UK
qiu@cs.nott.ac.uk

Abstract

Region duplication forgery, in which a part of a digital image is copied and then pasted to another portion of the same image in order to conceal an important object in the scene, is one of the common image forgery techniques. In this paper, we describe an efficient and robust algorithm for detecting and localizing this type of malicious tampering. We present experimental results which show that our method is robust and can successfully detect this type of tampering for images that have been subjected to various forms of post region duplication image processing, including blurring, noise contamination, severe lossy compression, and a mixture of these processing operations.

1. Introduction

Rapid advancement in imaging technology has made it remarkably easy to manipulate digital image contents. With the proliferation of digital cameras and computers, as well as software for image editing, the problem of digital image forgery is potentially very serious. Digital image counterfeiting has already appeared in many disturbing forms. For example, in the 2004 America presidential election, a widely circulated photograph showing the Democratic candidate and a famous Hollywood actress shared a demonstration podium in the 1970s was a fake in fact [1]. A popular British newspaper was forced to apologize for publishing photographs showing British soldiers abusing an Iraqi prisoner, which were proved to be fakes [2]. These examples are just a tip of the iceberg and it will get worse.

Recently, several researchers have started to develop techniques for detecting various forms of image forgery. Farid and Popescu developed several statistical methods for detecting forgeries based on region duplication, color filter interpolation, and re-sampling [3 - 6]. Fridrich presented a method for detecting copy-move type of forgery [7]. Ng and Chang proposed models of image spicing for detecting photomontage [8]. They have also recently developed physics-based model for distinguishing Computer Graphics from natural photographs [9].

A common form of digital image tampering is object removal, where regions of unwanted objects inside an image are replaced by pixels from other part of the image. Several researchers have developed methods for detecting this form of forgery. In [7], the authors analyzed the DCT coefficients for each block, while method [4] employs principal component analysis to capture the image blocks' features.

In this paper, we propose an efficient and robust algorithm for detecting and locating duplicated regions within an image thus exposing possible tampering of the image. Compared with the methods in [4] and [7], our algorithm has lower computational complexity and is more robust against stronger attacks and various types of after-copying manipulations, such as lossy compressing, noise contamination, blurring and a combination of these operations.

2. Model of Region-Duplication Forgery

Due to the nature of Region-Duplication, there will be at least two similar regions in the tampered image. However, in most natural images (except for images with large smooth regions), it is unusual to have two large similar regions in an image. Our analysis on over 100 natural images shows that, inside a single image, it is unlikely to have two very similar regions that are larger than 0.85% of the image size. The task of finding region duplication forgery is that of finding two large similar regions.See the Fig. 1.

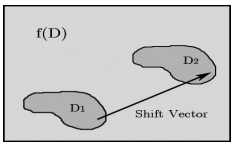

Figure 1. Region-Duplication Forgery

Given an image $f(x, y)$, the tampered image $f'(x, y)$, must subject to: $\exists$ regions D_1 and D_2 are subsets of D and a shift vector $d = (dx, dy)$, (we assume that $|D_1| = |D_2| >$

$|D|*0.85\%$ and $|d| > L$), $f'(x,y) = f(x,y)$ if $(x,y) \notin D_2$ and $f'(x,y) = f(x-dx, y-dy)$ if $(x,y) \in D_2$, where D_1 is the source and D_2 is the target region, $D_2 = D_1 + d$.

It would be easy to detect above forgery via exact match. However, to make the tampered image harder to detect, the attacker may perform various processing on $f'(x,y)$. Then the tampered image becomes $f''(x,y) = \Theta(f'(x,y))$, where Θ is the processing operator,such as JPEG compression and adding noise. The post processing attack makes the task of detecting forgery significantly harder. In the next section, we present an efficient method for detecting region duplication forgery which is also robust against various forms of post region duplication processing.

3. Proposed Algorithm

Our algorithm first divides an image into small overlapped blocks and it then compares the similarity of these blocks and finally identifies possible duplicated regions. Detail procedure is as follows:

1) Extracting block Characteristics Vector. The input image is split into overlapping blocks of $b \times b$ pixels. Assuming that the image is an M x N color image, there are S=(M-b+1)x(N-b+1) blocks. For each block B_i (i=1,2 ... S), seven characteristics features c_j (j =1,2 ... 7) are computed.

i)c_1, c_2, c_3 are the average of red, green, and blue components respectively.

ii)In the Y channel(Y=0.299R+0.587G+0.114B), we divide the block into 2 equal parts in 4 directions as shown in Fig. 2 and compute c_4,c_5,c_6,c_7 according to

c_i=sum(part(1))/sum(part(1)+part(2)) i = 4,5,6,7

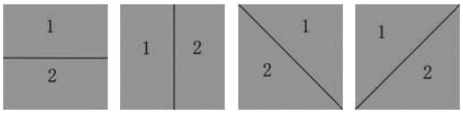

Figure 2. Four Directions

These characteristics features will not change significantly after some common processing operations. Taking additive white Gaussian noise(AWGN) operation for example, we assume that the noise for each pixel ε is an i.i.d. with mean 0 and variance v. The noisy block is $B'_i = B_i + \varepsilon$ then $c'_1 = c_1 + \varepsilon'$ (where E(ε') =0, D (ε') =v/ (b^2)), when, b$\geq$16, we have $c'_1 \approx c_1$. Similarly, we have $c'_2 \approx c_2$ and $c'_3 \approx c_3$. c'_4=(sum(part(1))+ε_1)/(sum(part(1)+part(2))+ε_2), where E(ε_1)=0, E(ε_2)=0, D(ε_1) $= b^2v/2$, D(ε_2) $= b^2v$. Usually sum(part(1))$\gg \varepsilon_1$,sum(part(1)+part(2))$\gg \varepsilon_2$ when v is small (SNR $\geq$18db), therefore $c'_4 \approx c_4$. and c_5, c_6 and c_7 have similar properties. For JPEG compression and Gaussian blurring, these operations will discard some high frequency components but change the low frequency components slightly. So $c_j (j = 1 \ldots 7)$ are robust to these operations. For each block B_i, a block characteristics vector V(i) =(c_1(i),c_2(i),c_3(i),c_4(i),c_5(i),c_6(i),c_7(i)) is computed and saved in an array A.

2) Searching similar block pairs. The array A is lexicographically sorted. For every pair B_i and B_j, we compute their similarity using their respective characteristics feature vector V(i) and V(j) in A as follows: Let Diff(k) = $|c_k$(i)- c_k(j)|, if the following conditions are satisfied (where $P(k)$'s, t_1 and t_2 are preset thresholds):

(i) Diff(k)<P(k)

(ii) Diff(1)+Diff(2)+Diff(3)<t_1, and

(iii) Diff(4)+Diff(5)+Diff(6)+Diff(7)<t_2 and if the shift vector between B_i, and B_j is greater than a preset threshold L then we record the pair as similar blocks and compute and store their shift vector $d' = (dx', dy')$ where $dx' = x_i - x_j$, $dy' = y_i - y_j$ where (x_i, y_i) and (x_j, y_j) are the upper left corner coordinates of block B_i and B_j.

3) Finding correct matching block pairs. Not all similar block pairs in 2) are equally likely to come from two duplicated regions. Therefore, we need to identify those similar blocks that come from two duplicated regions and remove those that are not. Assuming that the source and the target regions are larger than the block size, then all corresponding blocks in the source and the target regions will have the same shift vectors. Assuming that block B_1(i) is a block in the source region and B_2(i) is its corresponding block in the target region, then we have Shift(B_1(i), B_2(i))=Shift(B_1(j), B_2(j)), as illustrated in Fig. 3.

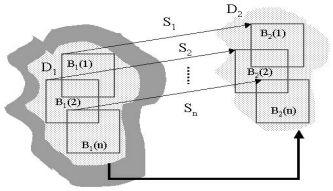

Figure 3. The Same Shift Vector

Based on this observation, we use the shift vectors of the similar block pairs to compile a histogram, H(d′) = H(dx′, dy′), which records the frequency of occurrence of the shift vector between two similar blocks. We then identify the shift vector having the highest frequency of occurrence as the main shift vector, d = arg(max(H(d′)). We regard a similar block pair as incorrect matching pair when its shift vector is much different from d. Let d=(dx,dy), we discard all pairs whose shift vector d′=(dx′,dy′) satisfy $|dx'-dx| > 2$ or $|dy'-dy| > 2$. All remaining similar blocks are then put in a binary image the same size as the original image with the areas covered by the blocks set to a white value and the rest set to a black value. We then perform the opening operation to eliminate small islands and then fill holes for all connected components on binary image [10]. This is a reasonable approach because if two blocks are similar and not

caused by a region duplication, then, it is unlikely there will be many of such blocks having the same shift vector.

4) Determining forgery. After finding two large connected regions, our algorithm determines whether a region duplication tampering has occurred based on following rules.Let R_1 and R_2 are two regions obtained from above procedure, if min($|R_1|$, $|R_2|$)> αM*N*0.85%, and $||R_1| - |R_2||/max(|R_1|, |R_2|) < Tr$, where α and Tr are preset thresholds, then extract the boundary of R_1, R_2, and locate them as duplication regions, set $tag = 1$ (meaning forgery detected). Otherwise the image is authentic, $tag = 0$. Because the image may be distorted by some processing operations, R_1, R_2 may be smaller than D_1, D_2. We use the parameter $\alpha \in (0.45, 1]$ to take into account these changes in all our experiments. We fix $\alpha = 0.5$. Also the two regions R_1, R_2 may not be the same size, we use Tr to take into account this possibility and fix $Tr = 12.5\%$.

4. Experimental Results

In our experiments, all images are of 300x400 pixels. The parameters are set as follow: $P(1)$=$P(2)$=$P(3)$ =1.80, $P(4)$=$P(5)$=$P(6)$=$P(7)$=0.0125, t_1=2.80, t_2=0.02, $b = 16$, $L = 50$.

If $tag = 0$ but the input image is tampered (fail to detect forgery), or $tag = 1$ but the image is authentic (wrong detection), we set J=1. Otherwise J = 0.

If the input image is a forgery and $tag = 1$, we define the accuracy r, and the false negative w:

$$r = \frac{|R_1 \cap D_1| + |R_2 \cap D_2|}{|D_1| + |D_2|}, w = \frac{|R_1 \cup D_1| + |R_2 \cup D_2|}{|D_1| + |D_2|} - r$$

The first example is shown in Fig.4. The tampered image is detected with $r = 0.9888, w = 0.1266$.

Figure 4. Left: Original image, Middle: Tampered image, Right: Detected Region

The tampered image is then distorted by lossy JPEG compression and additive white Gaussian noise. Fig. 5 show the detection results comparing with that of [4] (using default parameters and process green channel to yield duplication maps).

We can see that our algorithm achieves better accuracy, and is more robust to attacks. Even with the JPEG quality factor decreasing to as low as 30, our algorithm can still achieve $r = 0.8474$ and $w = 0.2170$. For the noisy image with SNR=16 db, our algorithm still manage to achieve

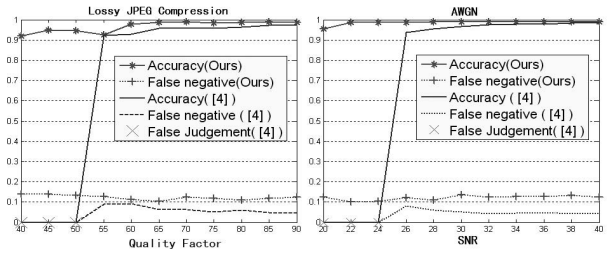

Figure 5. Robust to JPEG(Left),AWGN(Right)

$r = 0.8479$ and $w = 0.1956$. The algorithm in [4] failed when the JPEG quality factor was smaller than 50 or SNR was lower than 24 db. Our algorithm is also robust against Gaussian blurring (5x5 window,$\delta = 1$) and a combination of Gaussian blurring, AWGN with SNR 24 db, and JPEG lossy compression at quality factor of 60. The detection results are: Gaussian blurring: $(r, w) = (0.9920, 0.1291)$. Mixed operations: $(r, w) = (0.9176, 0.1751)$. Similar detection results are obtained in another example (downloaded from [4], see Fig.6). The tampered image without processing operations is detected with $r = 0.9953, w = 0.0878$. Fig. 7 show the detection results for various post region duplication compression and noise contamination comparing with [4]. For Gaussian blurring and Mixed operations, our algorithm achieves following results: Gaussian Blurring: $(r, w) = (0.9931, 0.0905)$. Mixed operations: $(r, w) = (0.9631, 0.0966)$.

Figure 6. Example 2

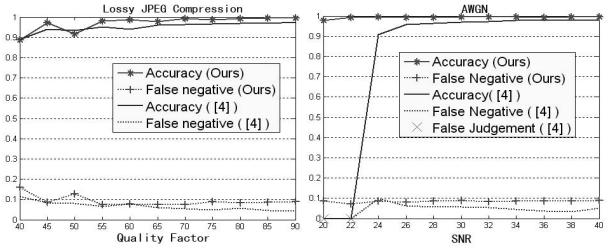

Figure 7. Robust to JPEG,AWGN

In order to test the efficiency and robustness of our algorithm further, we collected 100 images. For each image, a random square region was copied then pasted onto a non-overlapping position. The tampered images are then distorted by different processing operations. In our test, the square regions' sizes are of 32x32, 48x48,64x64 and 80x80. The average r,w and J over 100 images are shown in Fig. 8, and the tables 1 - 5.

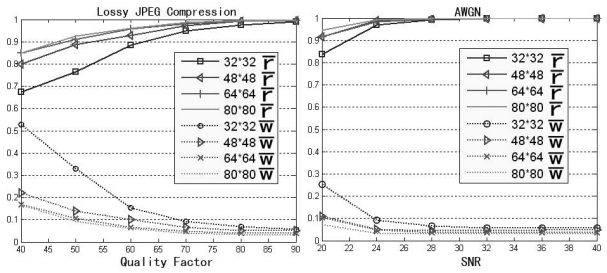

Figure 8. Robust to JPEG,AWGN

Table 1. Without any processing

	32*32	48*48	64*64	80*80
$\bar{r}$	1.0000	1.0000	1.0000	1.0000
$\bar{w}$	0.0679	0.0454	0.0375	0.0328
$\bar{J}$	0.06	0.01	0	0

Table 2. Gaussian Blurring

	32*32	48*48	64*64	80*80
$\bar{r}$	0.9478	0.9790	0.9846	0.9870
$\bar{w}$	0.1261	0.0722	0.0513	0.0433
$\bar{J}$	0.05	0.01	0	0

Table 3. False Judgment for JPEG

$\bar{J}$	40	50	60	70	80	90
32*32	0.30	0.12	0.10	0.05	0.05	0.05
48*48	0.08	0.02	0.02	0.01	0.01	0.01
64*64	0.04	0	0	0	0	0
80*80	0.01	0	0	0	0	0

Table 4. False Judgment for Noisy images

$\bar{J}$	20db	24db	28db	32db	36db	40db
32*32	0.20	0.06	0.07	0.09	0.10	0.08
48*48	0.05	0.02	0.01	0.01	0.01	0.01
64*64	0.01	0	0	0	0	0
80*80	0.01	0	0	0	0	0

Table 5. Mixture Operations

	32*32	48*48	64*64	80*80
$\bar{r}$	0.7275	0.8635	0.8849	0.9027
$\bar{w}$	0.4669	0.1612	0.1348	0.1131
$\bar{J}$	0.08	0.01	0	0

All the tables and figures above show that the bigger the block sizes copied and the higher the qualities, the better are the detection results. In general, if the copied block is bigger than 1.9% of the tampered image, the detection results are good. Inevitably there are a few images wrongly judged, especially for the images with similar regions or with large smooth regions or the image had been distorted badly. We believe that human interpretation is necessary in these special cases. We may change the parameters according to different images, or specify a suspicious region to match or eliminate some authentic regions before applying the algorithm. In our experience, such human intervention can always find the tampered regions.

5. Concluding Remarks

Region-duplication forgery is an effective technique to remove an object in digital image. In this paper, we have proposed a novel algorithm to detect tampered images automatically and effectively. Compared with [4] and [7], our algorithm has lower computational complexity and is more robust against various post region duplication image processing operations. Advances in computer vision and computer graphics have made image manipulation more and more sophisticated (see for example [11], [12]), How to detect forgeries using these advanced computer graphics technologies remains an open question.

Acknowledgement: We thank the authors of [4] for providing the code of their algorithm, and the support received from NSFC(60325208,90604008),NSF of Guangdong(04205407).

References

[1] K. Light. Fonda, Kerry and Photo Fakery. *The Washington Post, Saturday*, Feb. 28, 2004, Page A21.

[2] Voice of the Mirror. Sorry.. we were hoaxed: Iraqi PoW abuse pictures handed to us WERE fake. *Daily Mirror Newspaper*, 15 May 2004

[3] H. Farid. A picture tells a thousand lies. *New Scientist*, 6 Sep. 2003

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

[5] A C. Popescu and H. Farid. Exposing Digital Forgeries in Color Filter Array Interpolated Images. *IEEE Trans. on signal processing*, 53(10): 3948-3959,Oct. 2005.

[6] A.C. Popescu and H. Farid. Exposing Digital Forgeries by Detecting Traces of Resampling. *IEEE Trans. on Signal Processing*, 53(2):758-767,Feb. 2005

[7] J.Fridrich, D. Soukal, and J. Lukas. Detection of Copy-Move Forgery in digital Images. *Proc. of Digital Forensic Research Workshop*, Aug. 2003.

[8] T-T Ng, S-F Chang. A Model for Image Splicing. *ICIP*, 1169-1172 Vol.2, Oct. 2004

[9] T-T Ng, et al. Physics-Motivated Features for Distinguishing Photographic Images and Computer Graphics. *In ACM Multimedia*, 239-248, Nov. 2005

[10] R. C. Gonzalez and Richard E. woods. *Digital Image Processing*. Pearson Education, 2002.

[11] A. Criminisi, P. P'erez and K. Toyama. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Trans. on image processing*, 13(9): 1200-1212, Sept. 2004.

[12] L-Y Wei. Texture Synthesis by Fixed Neighborhood Searching. Ph.D. dissertation, Stanford University, Nov. 2001.