Tampering Detection

Sri Raghu Malireddi
Electrical Engineering
IIT Gandhinagar

November 25, 2015

Abstract

The word 'tampering' actually refers to the interference caused by using something which will cause unauthorized alterations in the original. In this paper, we will actually discuss about the digital forgeries in images and the procedure we adapted to identify the forgery in images. The method we implement will be different from the traditional water marking methods and will actually work on the basis of correlations between the pixels of the image.

1 Introduction

With the advent of many new technologies and sophisticated software, digital forgeries have become very easy and the forged images look very much identical to the original images. Besides having very minimal advantages of this technique, this has become a very serious problem in the case of Court of Law, validity of the Media's breaking news, and fake gossips about the celebrities etc. So, the field of digital image forensics had emerged to solve this kind of issues. For this particular problem of tampering detection, many sophisticated methods have been developed. One of the most popular methods is embedding watermarks or signatures into the images.

Though the watermarking method has advantages, it has its own disadvantages. We have to embed the watermark exactly at the time of capturing the image from digital camera. This can be done only when we have the access to the hardware that captured the image. But in real life and inside the Court of Law, you can have access to only the images, not (always) to the hardware used to capture those images. So, we thought why not we deal with digital tampering without involving any kind of watermark embedding techniques? Any kind of resampling in an image will introduce some correlations into the image. For understanding the above statement, we need to know how a camera will actually capture the color image. A digital color camera is equipped with a single CCD or CMOS sensor. They capture the color images using this single sensor with array of color filters, in which, each element will capture the intensity level of the either of the red, green or blue channels. The array that is used to capture image is called Color Filter Array (CFA). The red and blue pixels are sampled using the rectilinear lattices and green pixels are sampled using the quincunx lattice. The sensors will capture only 25% of red and blue light and 50% of

green light. The input array can be split into individual channels by using the following equations.

$$\begin{aligned}\tilde{R} &= S(x,y) \quad if\, S(x,y) = r_{x,y} \\ &= 0 \quad otherwise\end{aligned} \tag{1}$$

$$\begin{aligned}\tilde{G} &= S(x,y) \quad if\, S(x,y) = g_{x,y} \\ &= 0 \quad otherwise\end{aligned} \tag{2}$$

$$\begin{aligned}\tilde{B} &= S(x,y) \quad if\, S(x,y) = b_{x,y} \\ &= 0 \quad otherwise\end{aligned} \tag{3}$$

where $\tilde{R}, \tilde{G}, \tilde{B}$ are the red, blue and green channels reconstructed from the channel S received from the sensor. Overall, we have only $1/3^{rd}$ of the final image. The rest $2/3^{rd}$ rd of the image is designed using some interpolation techniques such as bilinear interpolation, cubic interpolation etc. These interpolations will introduce specific correlations into the image. The introduced correlations can be understood by the following equations.

$$R(x,y) = \sum_{u,v=-N}^{N} h_r(u,v)\tilde{R}(x-u, y-v) \tag{4}$$

$$G(x,y) = \sum_{u,v=-N}^{N} h_g(u,v)\tilde{G}(x-u, y-v) \tag{5}$$

$$B(x,y) = \sum_{u,v=-N}^{N} h_b(u,v)\tilde{B}(x-u, y-v) \tag{6}$$

where R, G and B are the reconstructed red, green and blue channels and h is a linear filter of size $(2N+1 \times 2N+1)$, and N is the neighborhood of the pixel that is taken into consideration. In bilinear interpolation, the filter h will be in the form $h = (0.5, 1, 0.5)$, in 1D. The 2D form of filter h can be obtained by doing the outer product of h. Here, h is called the binomial filter. Binomial filters are used for the compact rapid approximation of discrete gaussian. Some of its applications include multiscale imaging and volume representations. The realization of binomial filter can be done with the rows of Pascal triangle. The binomial filter is a low pass filter with no ripples in the stop band.

$$h_N[u] = (0.5 + 0.5 \times cos(u))^N \tag{7}$$

where h_N is a binomial filter with neighborhood N. This introduces correlations between the pixels in the final image we obtained. These correlations are altered while doing any kind digital forgery. If these correlations are known as specific kind, then it will be straight forward for us to determine which samples are correlated to their neighbors and which samples are not and hence finding the tampered regions. But, in real-life scenarios, we don't know those correlations. So, we employ the Expectation Maximization (EM) algorithm to simultaneously estimate the correlations and correlated samples.

2 Algorithm

The algorithm that we are implementing in this paper is taken from the following references [6] [7]. Here is a short glance of the algorithm steps:

1. Pre-process the image
2. Apply the EM algorithm
3. Testing on images

2.1 Pre-process the image

The image we take will be the final image obtained after applying some kind of CFA interpolation as per the hardware specifications of the camera. The algorithm what we will be implementing is valid for the bilinear CFA interpolated images. So, we need to pre-process the images to the raw image format and apply the bilinear interpolation on our own to get the desired image for testing. This process can be achieved by applying the inverse mathematical operations using the equations 1-6.

2.2 Apply the EM algorithm

The EM algorithm is a two step process. In the E-step, we will estimate the probability of each sample belonging to each model. In the M-step, we will estimate the specific form of correlations between samples. We will start with a random α. There are many versions of EM algorithm. We have devised our own method of EM algorithm, using the paper [7] as our inspiration. The key equations in our EM algorithm will be as follows.

$$R(x,y) = f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) \tag{8}$$

where $R(x,y)$ is the residual of the pixel value of the image f at position (x,y), α is a 3×3 matrix with $\alpha_{2,2} = 0$. After calculating the residual, we will update the new pixel value of the image as stated in the following equation.

$$f(x,y) = \sum_{u,v=-N}^{N} f(x+u, y+v) \tag{9}$$

Once the above equation is applied over all the pixels in the image, we will have the updated image which we will be using in the next iteration.

$$W(x,y) = Pr(\alpha | R(x,y)) = \frac{e^{-R^2(x,y)/\sigma}}{e^{-R^2(x,y)/\sigma} + 1/\delta} \tag{10}$$

where $W(x,y)$ in the posterior probability, and the element in the numerator is the conditional probability of the pixel at location (x,y).

$$E(\alpha) = \sum_{x,y} W(x,y) R^2(x,y) \tag{11}$$

where $E(\alpha)$ is actually the error for the present value of α. Our concern is to reduce this error upon each iteration. So, if we substitute the equation 8 into equation 11 and do the partial differentiation of E with respect to α then we will be left with the following equation.

$$\begin{aligned}\sum_{u,v=-N}^{N} \alpha_{u,v} \Big(\sum_{x,y} W(x,y) f(x+s, y+t) f(x+u, y+v) \Big) \\ = \sum_{x,y} W(x,y) f(x+s, y+t) f(x,y)\end{aligned} \tag{12}$$

The equation 12 is actually the α update rule that we implement in our algorithm. We will store the new estimate of α into a variable named α_{new}. The loop will iterate and perform all these steps until it meets the following stopping criteria.

$$|\alpha - \alpha_{new}| < \epsilon \tag{13}$$

where $\epsilon = 0.0075$ in our algorithm. If the stopping criteria is satisfied, we will apply the Fourier transform on the conditional probability map we obtained in our last iteration and then apply the high pass filter on the result we just obtained to remove the undesired low frequency noise and show the results by applying the transform mentioned in the below mentioned equation.

$$P_G = \left(\frac{P_H}{max(|P_H|)} \right)^4 \times max(|P_H|) \tag{14}$$

where P_G is the final image we observe, P_H is the image of the probability map after applying the Fourier transform and high pass filter operations. There are no clear explanations in the paper for why we are applying this transformation. The filter we applied is the gaussian high pass filter of size 5×5.

2.3 Testing on images

The algorithm we designed can be implemented on the images by first applying the pre-processing part and then applying the EM algorithm. The algorithm that we implemented is what all that was discussed in the paper. Since the field of digital image forensics is really vast, some things in this paper are taken as granted (or) known to the people who are already working in this field. Since, I am still a novice, I could just implement the understanding I gained from paper and some assumptions. The results that are obtained while testing on images will be discussed in the results session.

3 Results

The dataset we are using for this testing purpose is the standard image forensics dataset taken from [3]. Assuming the pre-procesing part as just conversion of RGB image to gray-scale, we implemented the EM algorithm on Figure 1 (b). The dataset has a tampered region showing a bird behind the horn of the cattle in the picture. We implemented the steps we stated for visualizing

the probability map and the Figure 1 $\{d, e\}$ show the results testing on the tampered and non-tampered regions of the image. We can observe a smooth variation of frequencies in the Tampered region when compared to the original region. Likewise we can observe the results for two other datasets in Figure 2 and Figure 3. Results in Figure 2 (d) are not visible because of the reduction in the size of the image. The light blue blocks in part (c) of each of the figures are the selected regions of the tampered and non-tampered regions of the image and which specific block refers to which portion is not mentioned because it can be considered obvious by observing the $\{a, b\}$ portions in each figure.

4 Conclusion

The challenges I faced while implementing the paper [7] are as follows. The paper discusses only about the simple linear/cubic interpolations. It does not discuss about the images which are tampered using some highly advanced non linear methods. We didn't get the original dataset on which the authors actually tested their algorithms. So we need to use the standard dataset from the work [3] and [5]. The visualization of the probability map is not clear and the usage of high pass filter which they used is not clearly specified. This might be because, these things they considered it as granted for any skilled person in the field. We referred to the following paper for implementing a specific high pass filter and bilinear filter [[1], [4], [2]]. To rectify some of the problems, we have modified the above algorithm which is discussed in the EM algorithm subsection. The modified algorithm can give decent results even on compressed JPEG format, where the paper talks about testing the images only on uncompressed TIFF format. Periodic patterns or spikes can only be observed when the image is resampled. In case of warping the portion of image using advanced forging techniques, we may not observe those spikes. The modified algorithm can find the non-correlations in the pixels even during the implementation of some advanced forging techniques in less number of iterations than the proposed algorithm in the paper. The initial choice of α and the termination criteria variable ϵ's precision do also matter for the accuracy of the results. Overall, as per the criteria of the paper the validity of the code's correctness hadn't been tested due to unavailability of the results. The code has been bit modified to work on compressed images and produce decent results. Further research in this field of testing the compressed images can be done and improved.

References

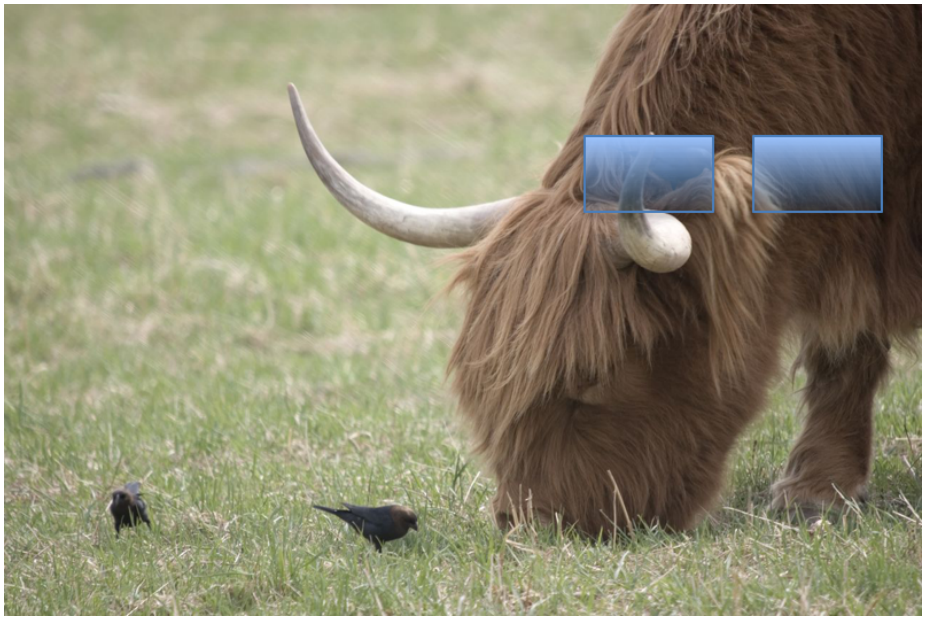
[1] Matthew Aubury and Wayne Luk. Binomial filters. *Journal of VLSI signal processing systems for signal, image and video technology*, 12(1):35–50, 1996.

[2] Peter J Burt. Fast filter transform for image processing. *Computer graphics and image processing*, 16(1):20–51, 1981.

[3] Vincent Christlein, Christian Riess, Johannes Jordan, Corinna Riess, and Elli Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *Information Forensics and Security, IEEE Transactions on*, 7(6):1841–1854, 2012.

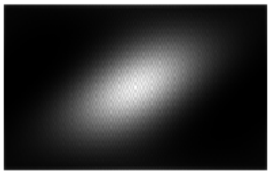
(a) Original

(b) Tampered



(c) Testing Locations

(d) Original



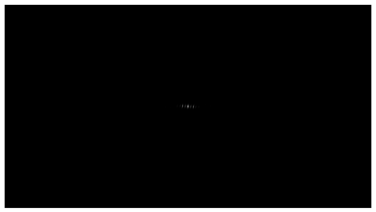
(e) Tampered

Figure 1: Results 1

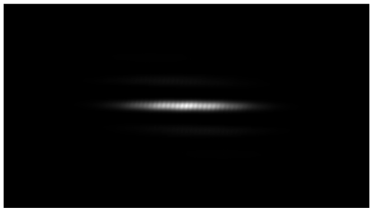
(a) Original

(b) Tampered

(c) Testing Locations



(d) Original

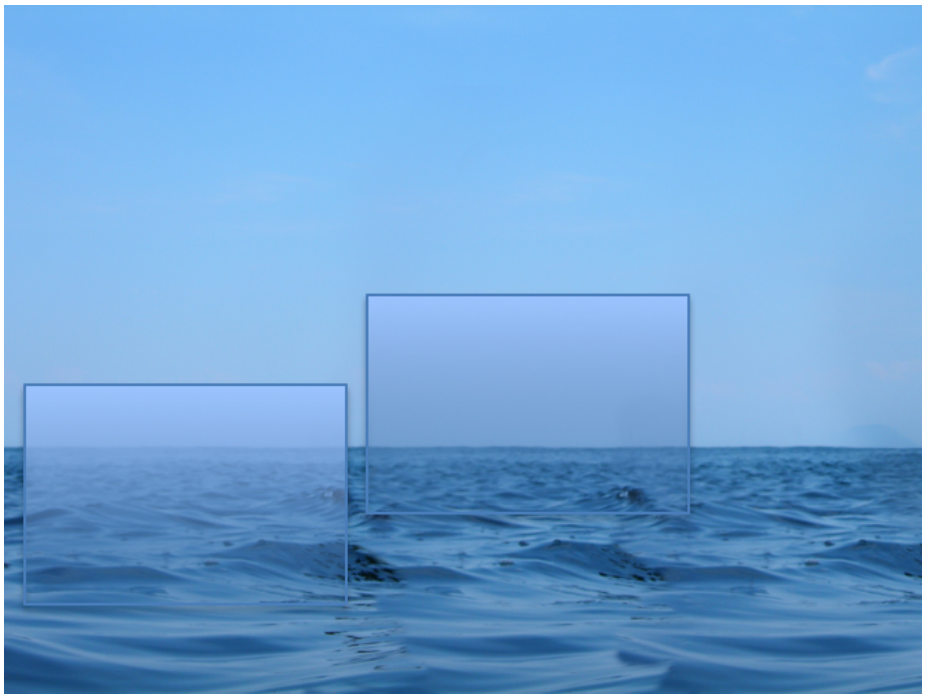


(e) Tampered

Figure 2: Results 2

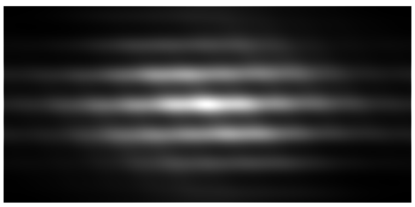
(a) Original

(b) Tampered



(c) Testing Locations

(d) Original



(e) Tampered

Figure 3: Results 3

[4] Konstantinos G Derpanis. Overview of binomial filters, 2005.

[5] Matthias Kirchner and Thomas Gloe. On resampling detection in recompressed images. In *Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on*, pages 21–25. IEEE, 2009.

[6] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *Signal Processing, IEEE Transactions on*, 53(2):758–767, 2005.

[7] Alin C Popescu and Hany Farid. Exposing digital forgeries in color filter array interpolated images. *Signal Processing, IEEE Transactions on*, 53(10):3948–3959, 2005.