Analysis of Copy-Move Forgery Detection

Computer Security Final Project

Duquesne University, Fall 2015

Brady Sheehan

Digital Image forensics has been a growing field for the past two decades. Photographs can often serve as evidence for proof that an event occurred. As a result, it can be advantageous to modify or doctor photos so that they will portray a different scene than they originally did. This leads to work in the area of image forgery. For this project, an analysis was performed on a copy-move forgery detection (CMFD) algorithm and its effectiveness on images with rotation and scaling in the copied region was tested. The paper is broken into four sections. First, a brief background on image forgery techniques is given. Then, a description of an algorithm for CMFD and its performance on rotated and scaled image forgeries is discussed. Finally, there is a brief description of ideas for future work in this area.

1. Background

The field of image manipulation has divided the doctoring of images into three broad categories: 1) Tampering - adjusting brightness or removing imperfections, 2) Splicing - two or more images will be combined to form a single image, and 3) Cloning or Copy-Move - a region of an image is copied and placed over a different region of the image. For this project, an algorithm that falls into category three will be discussed. In the event that someone manipulated an image with cloning, it can be highly difficult to detect this manipulation with the human eye. Much literature has been published in the area of copy-move forgery detection (CMFD). The goal of this project is to implement the algorithm proposed in [2] and test its effectiveness on images that have rotation and scaling in the copied-moved region.

When a copy-move forgery image is analyzed, there will always be some amount of correlation between the original region and the copied region. This correlation can be exploited and used for detection. As described by Fridrich et al, an algorithm for CMFD will need to satisfy three basic requirements. The first being that the algorithm must be able to handle approximate matches of small regions. This problem lends itself to a natural brute-force exhaustive search solution. However, a decent algorithm should run in a reasonable amount of time and produce few false positives. It should also be understood that the forged region will likely be a complete copied block rather than a collection of small

blocks or individual pixels and a successful algorithm should take this into consideration.

Many algorithms for CMFD have been designed based on a block matching solution. These algorithms are variants on the basic exhaustive search solution. However, since blocks cannot be compared pixel by pixel due to partial matches, there is a necessary component to extracting information from the blocks and comparing that information instead.

The model for CMFD described in this paper is based on using the discrete cosine transform (DCT) for representing and comparing blocks in the image. The DCT expresses a function or signal in terms of the sum of sinusoids with different frequencies and amplitudes. It is similar to the discrete Fourier Transform (DFT) except that the DCT uses only the real valued portion of the DFT. This transform is often used for image compression, which is the method used in JPEG compression. To apply the DCT to an image, the one-dimensional DCT is applied first to each row of an image and then to each column of the resulting image. Through this transformation, the image can be broken into parts (or spectral sub-bands) of differing importance. Put simply, the resulting image is a simpler version of the original – any information that was not essential to describing the image is removed. The result of this transform is a sparse set of coefficients that represent the image.

2. Model

The algorithm studied for this project is a variation on the algorithm described in [2]. The algorithm Fridrich's described can be broken into four steps. The first step is to compute the DCT of each $16x16$ sliding window in the image. For each of these blocks, the coefficients of the DCT are quantized with an extended JPEG quantization matrix described in [2]. After all of the blocks are quantized, they are inserted as rows into a matrix and then lexicographically sorted. Next, adjacent rows that match in the matrix are identified and for each match a shift vector is calculated. The shift vector is calculated as $s_i = [x_1 - x_2, y_1 - y_2]$ where $x_i$and y_i are the coordinates of the blocks in the image that match. For each s_i, a count is kept of the number of times it has been seen. For all s_i that have a count greater than some user specified threshold, the blocks corresponding to that shift vector are colored in the original image as possible forged regions.

It is important to point out that the algorithm has two parameters that influence its effectiveness. The first input is a quality factor that weights the quantization matrix. The higher the quality factor, the more likely dissimilar blocks will be marked as matches. The lower the quality factor, the less likely it is to have false matches and a higher quality match is likely to be found. The other important parameter of

the algorithm is the threshold mentioned before. This threshold determines the number of shift vectors that must be the same for a match to be marked as a forged region. A higher shift vector enforces a restriction that more blocks must be copied together for that region to marked as a forged region.

After implementing the algorithm described above in MATLAB, many false positives were identified regardless of the parameters chosen. After many tests, it was noticeable that a majority of the false positives were from shift vectors [1,0], [0,1], and [1,1]. Therefore, the algorithm was adjusted to not consider any of these three shift vectors as forged regions. The results of this slight variation are shown in section three and the effectiveness of this algorithm on regions with rotation and scaling is explained.

3. Results

The algorithm was run with many test inputs and five are shown in this paper. The algorithm is shown to work well on images with copy-move regions that do not include scale or rotation. In Figure 1[1], the algorithm is run on a crime scene photo with five objects shown on pavement. Despite the similarity of the pavement, when using a threshold of 10 and a quality factor of 0.5, the algorithm was correctly able to identify that the fifth object had been covered. In the output image, the copied region is highlighted and shown as the purple region being copied and moved to the green region (or vice-versa). Another example is given in Figure 2. In this figure, the middle image shows the result of the American flag on the right side of the image being copied and pasted above the telephone pole. With quality factor 0.35 and threshold 27, the algorithm was able to correctly identify the flag in purple as the copied-moved region. In some tests, the algorithm would be able to recognize the forged region but also incorrectly identify regions that were flat like a grassy field or the sky. These regions would get highlighted by the algorithm almost randomly and appear as noise in the output. An example where this happened is given in Figure 3.

Figure 1. Left: Original image; Middle: Input image; Right: Output image with threshold 10 and quality factor 0.5.

Figure 2. Left: Original image; Middle: Input image; Right: Output image with threshold 27 and quality factor 0.35.

In Figure 3, the person was copied with a rectangular region to the left and the shadow of the person was copied with a rectangular region below the shadow. Using a threshold of 32 and a quality factor of 0.745, the algorithm was able to identify in blue and green the two copy-moved regions. However, due to the false positives that appear as noise in the output, this example shows the need for human interpretation of the data from a CMFD algorithm.

In tests that included even a slight rotation in the copied region, the algorithm does not detect any of that region as a possible forged region. This result is demonstrated in Figure 4. Figure 4 shows the same crime scene photo as before except instead of a region being copied and directly moved overtop of the number 5, the copied region was rotated first by 180 degrees and then placed over the number 5.

Figure 3. Left: Original image; Middle: Input image; Top Right: Output image with threshold 32 and quality factor 0.745; Bottom Right: Test image showing only the regions that were highlighted.

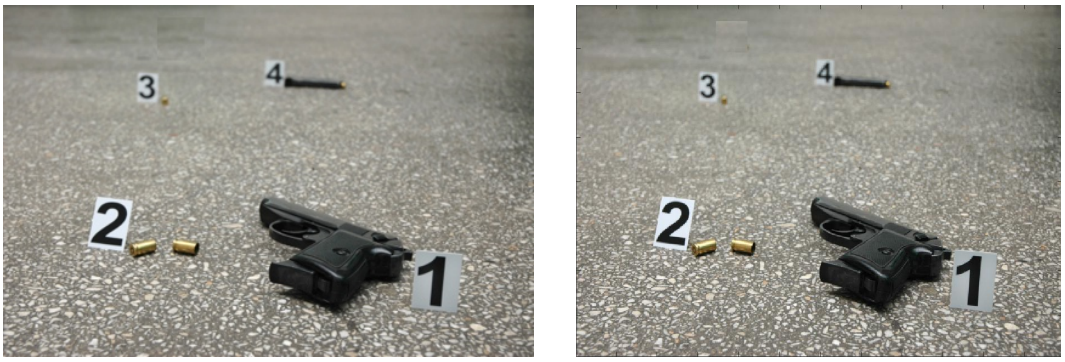

Figure 4. Left: Input image with number 5 copied using a rotation of 180 degrees; Right: Output image showing no highlighting, quality factor 0.5 and threshold 10.

The results were the same for any variation of rotation greater than 0 degrees and zero shift vectors were identified.

The algorithm also was unable to identify images where the copied region was scaled in any way. Figure 5 demonstrates the algorithms inability to identify scaled regions. In this figure, the same crime scene photo is used. The region boxed in red in the image on the left was copied, scaled to 175% its original size, and then moved over numbers 3, 4 and 5. With this input, zero shift vectors were identified. Similarly, for scaling of 125%, 150%, and 200% the algorithm identified zero shift vectors.

Figure 5. Left: Original image with box around the region that will be copied; Middle: Input image where the highlighted region from left was scaled by 150% and used to cover numbers 3, 4, and 5. Right: Output image showing no highlighting, quality factor 0.75 and threshold 10.

4. Future Work

This algorithm does work for copy-move forgeries that do not have rotation or scaling. However, finding images that have been manipulated to include rotated and scaled regions is common and an effective algorithm for CMFD would need to identify these types of forgeries as well. In the future, it would be worthwhile to investigate other descriptors that are invariant to rotation and scaling in

place of the DCT. Two methods to consider would be replacing the DCT with the Riesz Transform or the Fourier-Mellin Transform, both of which have been shown to be invariant to rotation and scaling.

References

1. Digital Image Forensics by Hany Farid.
2. J.Fridrich, D. Soukal, and J. Lukas. Detection of Copy-Move Forgery in digital Images. *Proc. Of Digital Forensic Research Workshop*, Aug. 2003.
3. B.L.Shivakumar and S.Santhosh Baboo, "Detecting Copy- Move Forgery in Digital Images: A Survey and Analysis of Current Methods", GJCST, Vol. 10, 2010.
4. Cao Y, Gao T, Fan L, Yang Q. A robust detection algorithm for copy-move forgery in digital images. *Forensic Science International*, 2012. 214 (1–3):33–43.
5. A.C. Popescu and H. Farid. Exposing digital forgeries by detecting duplicated image regions. *Technical Report TR2004-515*, Dartmouth College, Aug. 2004.
6. S. Bayram, H. T. Sencar, and N. Memon, A Survey of Copy-Move Forgery Detection Techniques, *IEEE Western New York Image Processing Workshop*, September 2008, pp. 538-542
7. S. Bayram, H.T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in Proc. IEEE ICASSP, 2009.
8. Crime scene photo is courtesy of https://mexicoinstitute.wordpress.com/tag/crime/.