A Direct Comparison Method for Detecting Copy Move Image Forgeries

Gavin Lynch

joint work with Professor Frank Shih
INTRODUCTION

- Image forgery is becoming increasing prevalent, especially now that image editing software is so accessible.

- Image forgery has been around nearly since the invention of the photograph.
Overview

- Image Forgery Detection
  - Active Authentication
  - Passive Authentication

- Copy Move Forgery Detection
  - The Naïve Approach
  - The Indirect Approach
  - The Proposed Direct Approach

- Experimental Results
OVERVIEW

- Image Forgery Detection
  - Active Authentication
  - Passive Authentication

- Copy Move Forgery Detection
  - The Naïve Approach
  - The Indirect Approach
  - The Proposed Direct Approach

- Experimental Results
**Image Forgery – Active Authentication**

- Active authentication implies one has prior knowledge about the image.

- **Example:**
  - Watermarking – an image is embedded with an invisible watermark which is corrupted if the image is changed.
  - The example below embeds the NJIT logo into the least significant bit of each pixel in the Lena image.

![Original Image](image1)

![Watermark](image2)

![Watermarked Image](image3)
Image Forgery – Passive Authentication

- In passive, or blind, authentication one does not have any prior knowledge regarding the image.

- There are several types of image forgery which require passive authentication techniques:
  - Retouching
  - Splicing
  - Copy Move
IMAGE FORGERY – COPY MOVE

- A copy move forgery copies one part of the image and places it over another part of the same image
  - Hides an important feature and/or
  - Adds an import feature

- The forgery below is from a 2004 George Bush political ad

![Original Image](image1.jpg) ![Forged Image](image2.jpg)
OVERVIEW

- Image Forgery Detection
  - Active Authentication
  - Passive Authentication

- Copy Move Forgery Detection
  - The Naïve Approach
  - The Indirect Approach
  - The Proposed Direct Approach

- Experimental Results
COPY MOVE FORGERY DETECTION

How can this type of forgery be detected?

The copied region may not be an exact copy of the original region:

- The image could be blurred.
- Noise can be added to the image.
- Saved under lossy compression.
- The copied region could be rotated.
- The copied region could be made lighter/darker than the original.
**The Naïve Approach**

- Naïve approach:
  1. Divide the image into small overlapping blocks.
  2. Compare all the blocks with one another.
  3. If two blocks match, then the two blocks represent potential forgeries.

- This approach is too slow!
THE NAÏVE APPROACH

- Why the naïve approach is not good:
  - Suppose the image is $n \times n$ pixels with block size $b \times b$ pixels and the blocks are compared pixel by pixel, then there are:
    $$\frac{(n-b+1)(n-b)}{2} b^2$$
    comparisons.
  - If $n = 256$ and $b = 16$, then there are 7.4 million comparisons.
  - If $n = 1024$ and $b = 64$, then there are 1.9 billion comparisons.
**The Indirect Approach**

- A popular method by Fridrich *et. al.* (2003) is the sliding block method
  1. Divide the image into small overlapping blocks.
  2. Extract a few descriptive features from each block.
  3. Sort the blocks lexicographically based on these features.
  4. After sorting, blocks that are near each other are considered matching.
  5. A decision is made as to whether the image contains a forgery based on the matching blocks.
THE INDIRECT APPROACH

- The sliding block method is only as good as its feature extraction in step 2.
  - Fridrich *et. al.* (2003) extracted coefficients based on the discrete cosine transform (DCT) of the block.
  - Popescu and Farid (2004) used a principal component analysis (PCA) approach to extract features.
  - Shih and Yuan (2010) extracted the mean and standard deviation (statistical) of the intensity of the pixels in the block.

- This method does not allow direct block comparison even though it could be a more accurate method!
THE PROPOSED DIRECT APPROACH

The Expanding Block Algorithm:
1. Divide the image into small overlapping blocks.
2. Extract a dominant feature from each block (i.e. the mean intensity of the pixels in the blocks)
3. Place the blocks into buckets based on the dominant feature.
4. Compare each block only with other blocks in the same bucket using an expanding block comparison.
THE PROPOSED DIRECT APPROACH

Expanding Block Comparisons:
1. Blocks are compared to one another on a small portion of the block (i.e. the 2 x 2 pixel upper-left region of the block).

2. If a block does not match any other block in the bucket, it is removed.

3. The comparison region is expanded and steps 1 and 2 are repeated until:
   a) The comparison region is larger than the block size
   b) There are no more blocks in the bucket.
Advantages of the proposed algorithm:

- **Block Bucketing**: Reduces the number of overall block comparisons.

- **Expanding Block Comparisons**: Removes blocks from the bucket based on a computationally quick decision.

- **Direct Block Comparison**: Allows the blocks to be compared directly – pixel by pixel.


**Block Comparisons**

- Blocks comparison is based on statistical hypothesis testing
  - Consider comparing block X with block Y

  ![X versus Y](image)

  - Block X has pixels \( p_x = [p_{x,1}, \ldots, p_{x,b^2}] \).
  - Block Y has pixels \( p_y = [p_{y,1}, \ldots, p_{y,b^2}] \).

  - Assume every pixel can be represented by a mean \( \mu \) and a random error \( \varepsilon \) which comes from noise, blurring, lossy compression, etc.
    - \( p_x = \mu_x + \varepsilon_x \) and \( p_y = \mu_y + \varepsilon_y \).
**Block Comparisons**

- Under this model, the null and alternative hypotheses for determining whether block X is different than block Y is
  - $H_0: \mu_x = \mu_y$
  - $H_1: \mu_x \neq \mu_y$

- Assume the error terms are independent Gaussians with mean 0 and variance $\sigma^2$.

- The test statistic is:
  $$ T = \frac{(p_x - p_y)^T (p_x - p_y)}{2\sigma^2} $$

- If $T$ is large conclude $H_1$ is true; otherwise conclude $H_0$ is true.
**Block Comparisons**

- $T$ is adapting to the variance:
  - If the variance is small, then small differences between $p_x$ and $p_y$ will be magnified.
  - If the variance is large, then small differences between $p_x$ and $p_y$ will be negligible.

$$T = \frac{(p_x - p_y)^T (p_x - p_y)}{2\sigma^2}.$$
OVERVIEW

- Image Forgery Detection
  - Active Authentication
  - Passive Authentication

- Copy Move Forgery Detection
  - The Naïve Approach
  - The Indirect Approach
  - The Proposed Direct Approach

- Experimental Results
EXPERIMENTAL RESULTS

- An experiment was carried out to compare the proposed expanding block algorithm (labeled EB) with the sliding block method using the DCT, PCA, and statistical feature extract methods.

- The algorithms were tested with 100 non-forged 256 x 256 pixel images.

<table>
<thead>
<tr>
<th></th>
<th>Number of False Positives</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT</td>
<td>1</td>
</tr>
<tr>
<td>Statistical</td>
<td>2</td>
</tr>
<tr>
<td>PCA</td>
<td>0</td>
</tr>
<tr>
<td>EB</td>
<td>3</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS

For each image, 10 forged images were created by copying a random region of size between 24 x 24 and 64 x 64 and placing onto a random non-overlapping region.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of True Positives</th>
<th>% of Forged Region Identified</th>
<th>Avg. Number of Pixels Incorrectly Identified as Forged</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT</td>
<td>1000</td>
<td>100%</td>
<td>633.21</td>
</tr>
<tr>
<td>Statistical</td>
<td>1000</td>
<td>100%</td>
<td>784.4</td>
</tr>
<tr>
<td>PCA</td>
<td>1000</td>
<td>100%</td>
<td>305.81</td>
</tr>
<tr>
<td>EB</td>
<td>1000</td>
<td>100%</td>
<td>85.12</td>
</tr>
</tbody>
</table>

Average time in seconds (code written in Matlab):

<table>
<thead>
<tr>
<th>Seconds</th>
<th>DCT</th>
<th>PCA</th>
<th>Statistical</th>
<th>EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seconds</td>
<td>2.23</td>
<td>22.41</td>
<td>4.98</td>
<td>10.32</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS

(a) The original image, (b) the forged image, (c) DCT copy move detection, (d) statistical copy move detection, (e) PCA copy move detection, (f) expanding block copy move detection.
**EXPERIMENTAL RESULTS - BLURRING**

- The effect of Gaussian blurring was investigated
EXPERIMENTAL RESULTS - BLURRING

(a) The original image, (b) the forged image with $10 \times 10$ Gaussian blurring, (c) DCT copy-move detection, (d) statistical copy-move detection, (e) PCA copy-move detection, (f) expanding block copy-move detection.
**Experimental Results – Non-Square Shapes**

### % of True Positives

<table>
<thead>
<tr>
<th>Shape</th>
<th>DCT</th>
<th>Statistical</th>
<th>PCA</th>
<th>EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangle</td>
<td>0.968</td>
<td>0.964</td>
<td>0.967</td>
<td>0.995</td>
</tr>
<tr>
<td>Circle</td>
<td>0.701</td>
<td>0.701</td>
<td>0.703</td>
<td>0.696</td>
</tr>
<tr>
<td>Square w/o Corners</td>
<td>0.878</td>
<td>0.877</td>
<td>0.878</td>
<td>1</td>
</tr>
<tr>
<td>Donut</td>
<td>0.721</td>
<td>0.718</td>
<td>0.711</td>
<td>0.972</td>
</tr>
</tbody>
</table>

### Pixels Incorrectly Identified

<table>
<thead>
<tr>
<th>Shape</th>
<th>DCT</th>
<th>Statistical</th>
<th>PCA</th>
<th>EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangle</td>
<td>826.52</td>
<td>920.25</td>
<td>423.82</td>
<td>81.215</td>
</tr>
<tr>
<td>Circle</td>
<td>680.38</td>
<td>680.38</td>
<td>807.59</td>
<td>305.75</td>
</tr>
<tr>
<td>Square w/o Corners</td>
<td>704.61</td>
<td>808.23</td>
<td>380.74</td>
<td>85.139</td>
</tr>
<tr>
<td>Donut</td>
<td>668.4</td>
<td>763.07</td>
<td>310.29</td>
<td>112.75</td>
</tr>
</tbody>
</table>

### % Correctly Identified

<table>
<thead>
<tr>
<th>Shape</th>
<th>DCT</th>
<th>Statistical</th>
<th>PCA</th>
<th>EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangle</td>
<td>0.968</td>
<td>0.96386</td>
<td>0.967</td>
<td>0.99497</td>
</tr>
<tr>
<td>Circle</td>
<td>0.66415</td>
<td>0.66415</td>
<td>0.65954</td>
<td>0.65707</td>
</tr>
<tr>
<td>Square w/o Corners</td>
<td>0.878</td>
<td>0.87688</td>
<td>0.878</td>
<td>1</td>
</tr>
<tr>
<td>Donut</td>
<td>0.69643</td>
<td>0.68373</td>
<td>0.6825</td>
<td>0.89731</td>
</tr>
</tbody>
</table>
Questions?
REFERENCES

