A Survey on Copy-Move Image Forgery Detection

Miss Sonali Navale
Department of E&TC, G. E. S. R. H. Sapat College of Engineering, Nashik

Abstract—Due to rapid advances and availabilities of powerful image processing software’s, it is easy to manipulate and modify digital images. So it is very difficult for a viewer to judge the authenticity of a given image. For digital photographs to be used as evidence in law issues or to be circulated in mass media, it is necessary to check the authenticity of the image. So In this paper, describes various methods for an Image forgery detection.

Keywords—Image tampering, geometric transformation, image forensics, authenticity.

I. INTRODUCTION

Images have acquired the reputation of being inarguable evidence. However, with the development of imaging technology and the accessibility of powerful affordable image editing tools like Photoshop, the evidence of tampering on digital images is extremely difficult to uncover. As a result, today digital images are losing authenticity and taking their authenticity for granted is becoming increasingly difficult in legal cases, in electronic media, in medical profession, and in financial institutions. Image splicing and copy-move are the most common techniques used for creating digital image forgeries. In image splicing, forgery is done by copying a part from one image and pasting to another one. On the other hand, in the copy-move, the copied part is pasted elsewhere in the same image to either add or hide objects. Usually, some processing is done on the copied part either before (e.g. scaling and rotation) or after (e.g. blurring and adding noise) pasting to make the editing less obvious and to eliminate irregularities that could show the image as tampered. The availability of powerful digital image processing programs, such as Photoshop, makes it relatively easy to create digital forgeries from one or multiple images. In a Copy-Move forgery, a part of the image itself is copied and pasted into another part of the same image. This is usually performed with the intention to make an object “disappear” from the image by covering it with a segment copied from another part of the image.

Figure 1. Example image of a typical copy-move forgery. Left: the original image. Right: the tampered image.

Textured areas, such as grass, foliage, gravel, or fabric with irregular patterns, are ideal for this purpose because the copied areas will likely blend with the background and the human eye cannot easily
discern any Suspicious artifacts. Because the copied parts come from the same image, its noise component, and most other important properties will be compatible with the rest of the image and thus will not be detectable using techniques that look for incompatibilities in statistical measures in different parts of the image, and to make the forgery even harder to detect, one can use the available photo editing software’s and image processing tools.

II. COPY-MOVE FORGERY DETECTION METHODS

This paper gives survey on various Copy Move Forgery Detection Methods are shown in table 1.

<table>
<thead>
<tr>
<th>Sr.n o</th>
<th>Title of Paper and Author</th>
<th>Publication</th>
<th>Method</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>“Detection of copy-move forgery in digital images” by A. J. Fridrich, B. D. Soukal, and A. J. Lukáš</td>
<td>Proceeding Digit. Forensic Res. Workshop, 2003</td>
<td>Used to detect when the copied area is saved in a lossy format i.e. JPEG</td>
<td>Not perform for the smooth region</td>
</tr>
<tr>
<td>3.</td>
<td>“Robust detection of region duplication forgery in digital image” by W. Luo, J. Huang, and G. Qiu</td>
<td>Proceeding 18th International Conference Pattern Recognition (ICPR), volume 4, pp. 746–749, 2006</td>
<td>An efficient and robust algorithm for detecting and localizing this type of malicious tampering.</td>
<td>Advanced computer graphics cannot be detect</td>
</tr>
<tr>
<td>6.</td>
<td>“An efficient and robust method for detecting</td>
<td>IEEE Transaction Information Forensics</td>
<td>Based on zernike moments of small</td>
<td>Limitation to the detectors based on</td>
</tr>
<tr>
<td>No.</td>
<td>Description</td>
<td>Journal/Conference Details</td>
<td>Features/Invariance</td>
<td>Remarks</td>
</tr>
<tr>
<td>-----</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>“On rotation invariance in copy-move forgery detection” by V. Christlein, C. Riess, and E. Angelopoulou</td>
<td>Proceeding IEEE Workshop International Information Forensics Security. (WIFS), pp.,1–6 December 2010.</td>
<td>Uses block wise feature vectors find similar feature vectors and select pairs that share highly similar shift vector</td>
<td>Involving a rigorous evaluation on the impact of noise and scale invariance SATS</td>
</tr>
<tr>
<td>8</td>
<td>“Image copy-move forgery detection based on SURF” by X. Bo, W. Junwen, L. Guangjie, and D. Yuweii</td>
<td>Proceeding International Conference Multimedia Information Network Security (MINES), pp. 889–892 , November 2010.</td>
<td>Based on SURF also quite robust to additive noise and blurring</td>
<td>Not investigate the automatic locate the tampered region and its boundry</td>
</tr>
</tbody>
</table>

### III. TYPICAL WORK FLOW FOR COPY MOVE FORGERY DETECTION

![Common processing pipeline for the detection of copy-move forgeries. The feature extraction differs for keypoint-based features (top) and block-based features (bottom).](image)

@IJRTER-2016, All Rights Reserved 254
Although a large number of CMFD methods have been proposed, most techniques follow a common pipeline, as shown in Fig. 2. Given an original image, there exist two processing alternatives. CMFD methods are either keypoint-based methods or block-based methods. In both cases, preprocessing of the images is possible. For instance, most methods operate on grayscale images, and as such require that the color channels be first merged. For feature extraction, block based methods subdivide the image in rectangular regions. For every such region, a feature vector is computed. Similar feature vectors are subsequently matched. By contrast, keypoint-based methods compute their features only on imageresions with high entropy, without any image subdivision. Similar features within an image are afterwards matched. A forgery shall be reported if regions of such matches cluster into larger areas. Both, keypoint and block-based methods include further filtering for removing spurious matches. An optional postprocessing step of the detected regions may also be performed, in order to group matches that jointly follow a transformation pattern.

a) **Matching**: High similarity between two feature descriptors is interpreted as a cue for a duplicated region. For block-based methods, most authors propose the use of lexicographic sorting in identifying similar feature vectors. In lexicographic sorting a matrix of feature vectors is built so that every feature vector becomes a row in the matrix. This matrix is then row-wise sorted. Thus, the most similar features appear in consecutive rows. Other authors use the Best-Bin-First search method derived from the kd-tree algorithm to get approximate nearest neighbors. In particular, keypoint-based methods often use this approach. Matching with a kd-tree yields a relatively efficient nearest neighbor search. Typically, the Euclidean distance is used as a similarity measure. In prior work, it has been shown that the use of kd-tree matching leads, in general, to better results than lexicographic sorting, but the memory requirements are significantly higher. Note, however, that the dimensions of some feature sets are ordered by importance. For these features, the performance gain over lexicographic sorting is minimal. In our setup we matched feature vectors using the approximate nearest neighbor method. It uses multiple randomized kd-trees for a fast neighbor search.

b) **Filtering**: Filtering schemes have been proposed in order to reduce the probability of false matches. For instance, a common noise suppression measure involves the removal of matches between spatially close regions. Neighboring pixelsoften have similar intensities, which can lead to false forgery detection. Different distance criteria were also proposed in order to filter out weak matches. For example, several authors proposed the Euclidean distance between matched feature vectors.

c) **Postprocessing**: The goal of this last step is to only preserve matches that exhibit a common behavior. Consider a set of matches that belongs to a copied region. These matches are expected to be spatially close to each other in both the source and the target blocks (or keypoints). Furthermore, matches that originate from the same copy-move action should exhibit similar amounts of translation, scaling and rotation. The most widely used postprocessing variant handles outlier by imposing a minimum number of similar shift vectors between matches. A shift vector contains the translation (in image coordinates) between two matched feature vectors. Consider, for example, a number of blocks which are simple copies, without rotation or scaling. Then, the histogram of shift vectors exhibits a peak at the translation parameters of the copy operation. Consider a pair of matched feature vectors as forged if: a) they are sufficiently similar, i.e. their Euclidean distance is below a threshold, and b) the neighborhood around their spatial locations contains similar features. An area threshold can also be applied, so that the detected region has at least a minimum number of points. To handle rotation and scaling, use RANSAC. For a certain number of iterations, a random subset of the matches is selected, and
the transformations of the matches are computed. The transformation which is satisfied by most matches (i.e., which yields most inliers) is chosen.

3.1. Block based Algorithm
We investigated 13 block-based features, which we considered representative of the entire field. They can be grouped in four categories: moment-based, dimensionality reduction-based, intensity-based, and frequency domain-based features.

**Moment-based**: We evaluated 3 distinct approaches within this class of 24 blur-invariant moments as features (BLUR), Hu moments (HU), Zernike moments (ZERNIKE).

**Dimensionality reduction-based**: The feature matching space was reduced via principal component analysis (PCA), the Kernel-PCA (KPCA) variant of PCA, also computed the singular values of a reduced-rank approximation (SVD). A fourth and Singular Value Decomposition did not yield reliable results in setup and was, thus, excluded from the evaluation.

**Intensity-based**: The first three features are the average red, green and blue components. Additionally, used directional information of blocks (LUO) while Bravo-Solorio consider the entropy of a block as a discriminating feature (BRAVO). Lin (LIN) computed the average grayscale intensities of a block and its sub-blocks, mean intensities of circles with different radii around the block center (CIRCLE).

**Frequency-based**: The use of 256 coefficients of the discrete cosine transform as features (DCT). The coefficients of a discrete wavelet transform (DWT) using Haar-Wavelets were recommended the use of the Fourier-Mellin Transform (FMT) for generating feature vectors.

3.1.1. Keypoint based Algorithm
Unlike block-based algorithms, keypoint-based methods rely on the identification and selection of high-entropy image regions (i.e., the “keypoints”). A feature vector is then extracted per keypoint. Consequently, fewer feature vectors are estimated, resulting in reduced computational complexity of feature matching and post-processing. The lower number of feature vectors dictates that post-processing thresholds are also to be lower than that of block-based methods. A drawback of keypoint methods is that copied regions are often only sparsely covered by matched keypoints. If the copied regions exhibit little structure, it may happen that the region is completely missed. We examined two different versions of keypoint-based feature vectors. One uses the SIFT features while the other uses the SURF features. They are denoted as SIFT and SURF, respectively. The feature extraction is implemented in standard libraries. However, particular differences of keypoint-based algorithms lie in the postprocessing of thematched features.
Figure 3. The image beachwood (upper left) is forged with a green patch (bottom left) to conceal a building (upper right). A ground truth map (bottom right) is generated where copy-moved pixels are white, unaltered pixels are black and boundary pixels are gray.

IV. CONCLUSION

A keypoint-based method, e.g. based on SIFT features, can be very efficiently executed. Its main advantage is the remarkably low computational load, combined with good performance. Keypoint-based methods, however, are sensitive to low-contrast regions and repetitive image content. Here, block-based methods can clearly improve the detection results. In a number of experiments, five block-based feature sets stood out, namely DCT, DWT, KPCA, PCA and Zernike. Among these we recommend the use of ZERNIKE, mainly due to its relatively small memory footprint.

REFERENCES