ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Detecting Multiple Copies of Copy-Move Forgery Based on SURF

K.Kiruthika, S.Devi Mahalakshmi, K.Vijayalakshmi

Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi, TamilNadu, India

Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi, TamilNadu, India

Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi, TamilNadu, India

Abstract — An Extensive growth in software technologies results in tampering of images. A major problem that occurs in the real world is to determine whether an image is authentic or forged. Copy-Move Forgery Detection is a special type of forgery detection approach and widely used under digital image forensics. In copy-move forgery, a specific area is copied and then pasted into any other region of the image. The main objective of this paper is to detect the multiple copies of the same region and different regions. In this paper, keypoint-based method are used. In keypoint-based method, SURF (Speeded Up Robust Features) method is used for feature extraction. The g2NN strategy is done for identifying the matched points. Then the Agglomerative Hierarchical Clustering is done on the matched points so that false detection rate can be reduced

Keywords — Copy-Move Forgery, SURF, HAC, image forensics, g2NN strategy.

I. INTRODUCTION

The goal of blind image forensics is to determine the authenticity and origin of digital images without the support of an embedded security scheme [1]. Within this field, copy-move forgery detection (CMFD) is probably the most actively investigated subtopic [2]. A copy-move forgery denotes an image where part of its content has been copied and pasted within the same image. Typical motivations are either to hide an element in the image [3], or to emphasize particular objects. Copy-move Forgery Detection methods are either keypoint-based methods or block-based methods. Keypoint-based methods compute their features only on image regions with high entropy, without any image subdivision for feature extraction. Similar features within an image are afterwards matched. There are two types of keypoint-based methods such as Scale Invariant Feature Transform (SIFT) [8], [18] and Speeded Up Robust Features (SURF) [11]. Block-based methods subdivide the image into rectangular regions that is tile the image into overlapping blocks for feature extraction. For every such region, a feature vector is computed. Similar feature vectors are subsequently matched. There are 13 block-based features and it can be grouped into four categories: Moment-based (Blur, Hu, Zernike [12]), Dimensionality reduction-based (PCA [5], SVD, KPCA [6]), Intensity-based (Luo, Bravo, Lin [7], Circle), Frequency-based (DCT [9], DWT [6], FMT [4]). The main goal of this paper is to detect the multiple copies of the same region and detect the multiple copies of different region.

(a) (b)

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

The paper is organized as follows. The Section II describes the proposed work. Section III discuss about the employed error metrics. Section IV focus on results and observations. Section V discuss about performance

analysis. Section VI contains a conclusion and future work.

II. PROPOSED METHOD

Copy-Move Forgery Detection has five steps. The block diagram for copy-move forgery detection is shown in Figure 2.

A. Pre-Processing

Here the image is converted from RGB to Gray representation.

B. Feature Extraction

The features can be extracted by using SURF (Speeded Up Robust Features) method. SURF is the robust local feature detector. It is based in sums of 2D Haar Wavelet responses and makes an efficient use of integral images. SURF features can be extracted using the following steps:

- Integral Image
- Keypoint Detection
- Orientation Assignment
- Feature Descriptor Generation

1) Integral Image:

Integral Image increases the computation speed as well as the performance, its value is calculated from an upright rectangular area, the sum of all pixel intensities is calculated by the formula,

$$\sum = A + D - (C + B) \quad \textbf{(1)}$$

Which is in the rectangular area whose vertices are A, B, C and D. It allows for fast computation of box type convolution filters. Suppose an input image I and a point (x, y) is given. The integral image $I_{\sum}$ is calculated by the sum of the values between the point and the origin.

$$I_{\sum}(x, y) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(x, y) \quad \textbf{(2)}$$

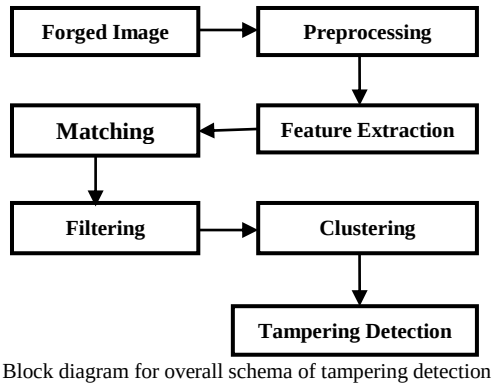


Fig. 2. Block diagram for overall schema of tampering detection

2) Keypoint Detection:

This step requires scale space generation for the extraction of keypoints. To detect the blob-like structures at locations where the determinant is maximum. In SURF Laplacian of Gaussian is approximated with a box filter. Convolution is applied to an image with varying size box filter for creating the scale space. After constructing the scale space, determinant of the Hessian matrix is calculated for detecting the extremum point. If determinant of the Hessian matrix is positive that means, both the Eigen values are of the same sign either both are negative or both are positive.

In case of the positive response, points will be taken as extrema otherwise it will be discarded. Hessian matrix is represented by,

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad \textbf{(3)}$$

Where, L_{xx} (x, σ) is the convolution of the Gaussian second order derivative with the image I in point x, and similarly L_{xy}(x, σ) and L_{yy}(x, σ). These derivatives are called as Laplacian of Gaussian. The 9 × 9 box filters are approximation of the Gaussian and represent lowest scale for computing the blob response maps. The approximate determinant of the Hessian matrix is calculated by:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad \textbf{(4)}$$

Where 0.9 represents the weights applied to the rectangular regions are simple for computational efficiency. The relative weight of the filter responses is used to balance the expression for the Hessian's determinant. The approximated determinant of the Hessian represents the blob response in the image at the specified location. These responses are stored in a blob response map over different scales, and local maxima are detected.

3) Orientation Assignment:

At first a circular area is constructed around the keypoints. Then Haar wavelets are used for the orientation assignment. It also increases the robustness and decreases the computational cost. Haar wavelet responses are calculated within a circular neighborhood of some radius around the interest point. Haar wavelets are filters that detect the gradients in x and y directions. In order to make rotation invariant, a reproducible orientation for the interest point is identified. Once the wavelet responses are calculated and weighted with the Gaussian centered at the interest points, the responses are represented as points in a space with the horizontal response strength and the vertical response strength.

The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window. The horizontal and vertical responses within the window are summed. The two summed responses then yield a local orientation vector. The longest such vector over all windows defines the orientation of the interest point. A circle segment of 600 is rotated around the interest point. The maximum value is chosen as a dominant orientation for that particular point.

4)Feature Descriptor Generation:

For generating the descriptors, first construct a square region around an interest point, where interest point is taken as the center point. This square area is again divided into 4 × 4 smaller subareas. For each of this cell Haar wavelet responses are calculated. Here d_x, termed as horizontal response and d_y, as vertical response. Horizontal and vertical response represents the selected interest point orientation. The wavelet responses d_x and d_y are summed up over each sub-region and form a first set of entries in the feature vector. And then extract the sum of the absolute values of the responses $|d_x|$ and $|d_y|$. For each of this sub region 4 responses are collected as:

$$V_{subregion} = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|] \quad \textbf{(5)}$$

So each sub region contributes 4 values. Therefore the descriptor is calculated as 4 × 4 × 4 = 64.

C. Matching

A matching operation is performed among the feature vectors to identify similar local patches in the image. In the existing work, Approximate Nearest Neighbor method is used for feature matching. It uses multiple randomized kd-trees for a fast neighbor search. It detects only the single copy-move region. In this paper, matching is done by using g2NN strategy. The generalized 2NN test starts from the observation that in a high dimensional feature space, features that are different from one considered share very high and very similar values among them [7]. It consists of iterating the 2NN test between d_i/d_{i+1} until this ratio is greater than the threshold value T_1.

$$Ratio = \frac{d_i}{d_{i+1}} \quad \textbf{(6)}$$

If k is a value in which the procedure stops, each keypoint in correspondence to a distance in $\{d_{1,\ldots,}dk\}$ (where $1 \leq k < n$) is considered as a match for the inspected feature. It is able to detect the multiple copies of the same region. For further processing, the matched keypoints are used. The unmatched keypoints can be left out.

D. Filtering

Filtering schemes are used to reduce the probability of false matches. Neighboring pixels often have similar intensities, which can lead to false forgery detection. The Euclidean distance that can be calculated between each feature vectors. The pairs can be removed if it is less than the particular threshold value T_2.

E. Clustering

The Agglomerative Hierarchical Clustering is used to cluster the forged regions. It is done on the matched keypoints. This is done in order to avoid the false positives. Hierarchical clustering involves tree like structure.

The hierarchical clustering involves the following steps.

1. Assign each keypoint to a cluster.
2. Compute all the reciprocal spatial distances among the clusters.
3. Finds the closest pair of clusters.
4. Merges them into single cluster.

The clustering is done iteratively until certain threshold is reached. The inconsistency coefficient is compared with threshold, to stop cluster grouping. The linkage method is used to find the distance between the set of observations. Ward's Linkage method is used.

1) Ward's Linkage:

The Error Sum of Squares (ESS) is calculated and the increment or decrement is evaluated when the cluster is joined to a single one.

$$\Delta_{dist}(P,Q) = ESS(PQ) - [ESS(P) + ESS(Q)] \quad \textbf{(7)}$$

If the cluster detected his significant number of matched keypoints, then the cluster is eliminated. The clusters that have more than two matched keypoints are considered to be matched cluster. If more than two such clusters found, the image is considered to be forged image.

III. ERROR MEASURES

It analyzes the performance of evaluating copy-move forgery detection algorithms at image level. It focuses on whether the fact that an image has been tampered or not to be detected. The measures precision and recall can be calculated by

$$p = \frac{T_p}{T_p + F_p} \quad \text{and} \quad r = \frac{T_p}{T_P + F_N} \quad \textbf{(8)}$$

Where T_P - The number of correctly detected forged images.

F_P - The number of images that have been erroneously detected as forged.

F_N - Falsely missed forgery images.

Precision denotes the probability that a detected forgery is truly a forgery; while Recall shows the probability that a forged image is detected. Recall is often also called true positive rate. score as a measure which combines precision and recall in a single value.

$$F_1 = \frac{2 \times p \times r}{p + r} \quad \textbf{(9)}$$

Where p represents precision and r represents recall.

IV. RESULTS AND DISCUSSION

Results for detecting single and multiple copy-move forgery are shown below. Figure 3, 4 and 5 shows the results for detecting copy move image forgery. Some part of the original image has been copied to other areas to get a forged image.

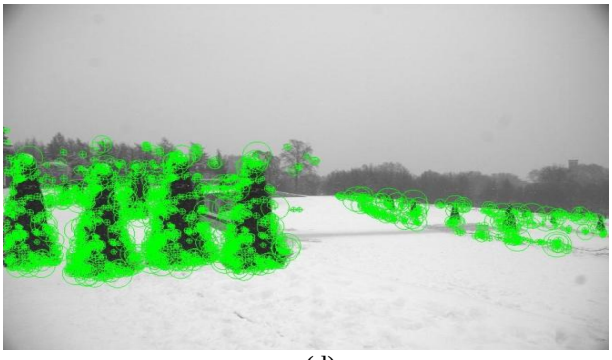
A. Detection of Multiple copies of same region.

Figure 3 shows the result for detecting multiple copies of the same region. The original image and the forged image are shown in figure (a) and (b) respectively.

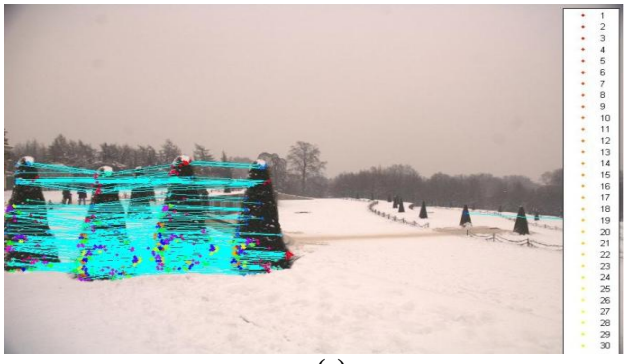
Some part of the original image has been copied and pasted it into multiple times of the same region. Figure (c) is the pre-processed forged image. Figure (d) shows the feature extraction for forged image. Figure (e) shows the location in which the forgery has been identified.

(a) (b)

(c)



(d)



(e)

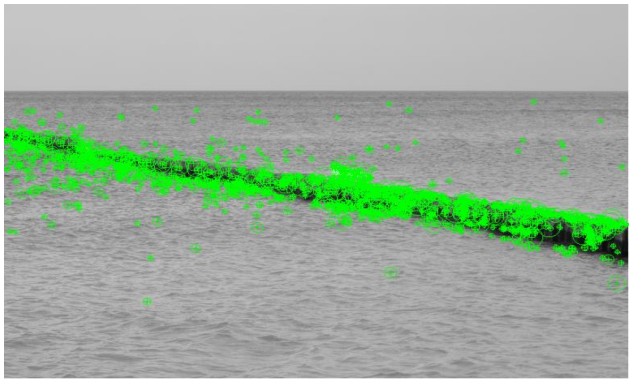
Fig. 3 Detection of multiple copies of same region.

B. Detection of Single Copy-Move Forgery

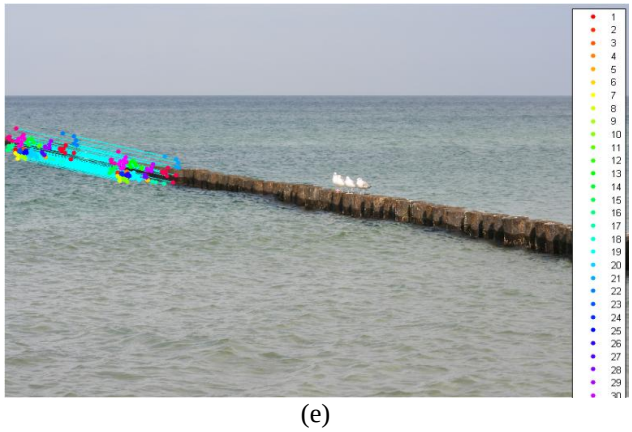
Figure 4 shows the results for detecting single copy move forgery. The original image and the forged image are shown in figure (a) and (b) respectively. Some part of the original image has been modified to get a forged image. Figure (c) is the pre-processed forged image. Figure (d) shows the feature extraction for forged image. Figure (e) shows the location in which the forgery has been identified.

(a) (b)

(c)



(d)



(e)

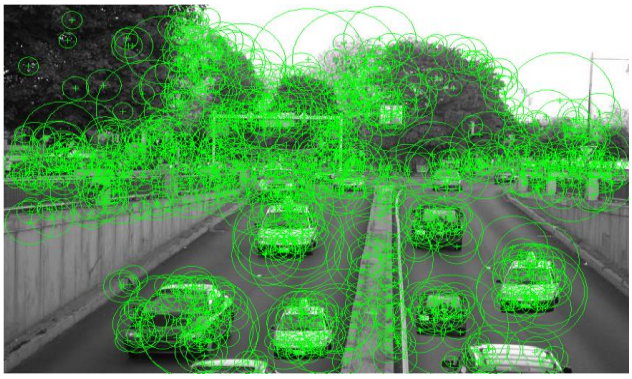
Fig. 4. Detection of Single Copy-Move Forgery

C. *Detection of multiple copies of different regions*

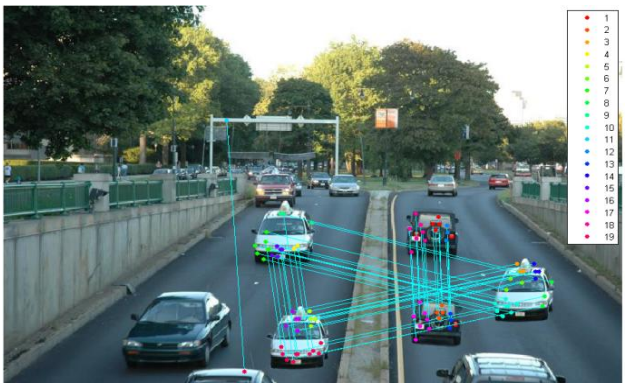
Figure 5 shows the results for detecting multiple copies of different regions. The original image and the forged image are shown in figure (a) and (b) respectively. Two different parts of the original image has been modified to get a forged image. Figure (c) is the pre-processed forged image. Figure (d) shows the feature extraction for forged image. Figure (e) shows the location in which the forgery has been identified

(a) (b)

(c)



(d)



(e)

Fig. 5. Detection of multiple copies of different region

V. PERFORMANCE ANALYSIS

Table 1 shows the performance analysis for SURF method to detect copy-move forgery based on the error measures. The True Positive Rate and False Positive Rate are determined for the tested images and the performance for the existing method is compared with this work.

TABLE I

COMPARISON OF TPR AND FPR FOR COPY-MOVE ATTACK (AVERAGE PER IMAGE)

Methods	TPR (%)	FPR (%)	F1 (%)
SURF and Approximate nearest neighbor	91.49	89.58	90.53
SURF and Generalized 2NN (g2NN) test	94	92	93

The processing time for the two methods is compared in table 2.

TABLE II

COMPARISON OF PROCESSING TIME (AVERAGE PER IMAGE)

Methods	Average Processing Time (seconds)
SURF and Approximate nearest neighbor	31
SURF and Generalized 2NN (g2NN) test	24

VI. CONCLUSION

This work is used to find whether the image is forged one or not. This work deals with the detection of copy-move attack. The paper detects the multiple copies of same region and multiple copies of different region of copy-move forgery. The features can be extracted by using keypoint method called SURF. The g2NN matching is done to match the features. The filtering is done to avoid similar match features. In order to avoid false

positives, agglomerative hierarchical clustering is done and finally to identify the image is forged or not.

This work does not identify other types of image tampering techniques such as enhancing and splicing attack and it identifies only the copy move forgery. The future work is to identify such attacks.

REFERENCES

[1] V Christlein, C Riess, J Jordan, C Riess, and E Angelopoulou, *"An Evaluation of Popular Copy-Move Forgery Detection Approaches,"* IEEE Trans. Inf. Forensics Security, vol. 7, no. 6,p. 1841–1854, Dec. 2012.

[2] J. Redi, W. Taktak, and J.-L. Dugelay, "*Digital image forensics: A booklet for Beginners,"* Multimedia Tools Applications, vol. 51, no. 1, pp. 133–162, Jan. 2011.

[3] H. Farid, "*A survey of image forgery detection*," Signal Process. Mag., vol. 26, no. 2, pp. 16–25, Mar. 2009.

[4] S. Bayram, H. Sencar, and N.Memon, "*An efficient and robust method for detecting copy-move forgery*," in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Apr. 2009, pp. 1053–1056.

[5] A. Popescu and H. Farid, "*Exposing Digital Forgeries by Detecting Duplicated Image Regions*", Department of Computer Science, Dartmouth College, Tech. Rep. TR2004-515, 2004

[6] M. Bashar, K. Noda, N. Ohnishi, and K. Mori, "*Exploring duplicated regions in natural images*," International Journal of Computer Applications (0975 – 8887).

[7] H. Lin, C. Wang, and Y. Kao, "*Fast copy-move forgery detection*," WSEAS Trans. Signal Process., vol. 5, no. 5, pp. 188–197, 2009.

[8] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "*A SIFT-based forensic method for copy-move attack detection and transformation recovery*," IEEE Trans. Inf. Forensics Security, vol. 6, no. 3,p. 1099–1110, Sep. 2011.

[9] J. Fridrich, D. Soukal, and J. Lukas, "*Detection of copy-move forgery in digital images*," Proc. Digital Forensic Research Workshop, Cleveland, OH, August 2003.

[10] J. S. Beis and D. G. Lowe, "*Shape indexing using approximate nearest neighbor search in high-dimensional spaces*," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Jun. 1997, pp. 1000–1006.

[11] B. L. Shivakumar and S. Baboo, "*Detection of region duplication forgery in digital images using SURF*," Int. J. Comput. Sci. Issues, vol. 8, no. 4, pp. 199–205, 2011.

[12] S. Ryu, M. Lee, and H. Lee, "*Detection of copy-rotate-move forgery using Zernike moments*," in Proc. Information Hiding Conf., Jun. 2010, pp. 51–65.

[13] X. Pan and S. Lyu, "*Region duplication detection using image feature matching*," IEEE Trans. Inf. Forensics Security, vol. 5, no. 4, pp.857–867, Dec. 2010.

[14] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou , "*Supplemental Material to an Evaluation of Popular Copy-Move Forgery Detection Approaches*" Aug. 2012.

[15] V. Christlein, C. Riess, and E. Angelopoulou, "*A study on features for the detection of copy-move forgeries*," in Proc. GI SICHERHEIT, Berlin, Germany, Oct. 2010, pp. 105–116.

[16] M. Muja and D. G. Lowe, "*Fast approximate nearest neighbors with automatic algorithm configuration*," in Proc. Int. Conf. Computer Vision Theory and applications, Feb. 2009, pp. 331–340.

[17] J. Zhang, Z. Feng, and Y. Su, "*A new approach for detecting copy move forgery in digital images*," in Proc. Int. Conf. Communication Systems, Nov. 2008, pp. 362–366.

[18] H. Huang, W. Guo, and Y. Zhang, "*Detection of copy-move forgery in digital images using SIFT algorithm*," in Proc. Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Dec. 2008, vol. 2, pp. 272–276.