Exposing postprocessed copy–paste forgeries through transform-invariant features

Sudha, N.; Kakar, Pravin

2012

Kakar, P., & Sudha, N. (2012). Exposing postprocessed copy–paste forgeries through transform-invariant features. *IEEE Transactions on Information Forensics and Security*, 7(3), 1018-1028.

<https://hdl.handle.net/10356/95528>

<https://doi.org/10.1109/TIFS.2012.2188390>

© 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/TIFS.2012.2188390>].

Downloaded on 26 Apr 2023 19:42:34 SGT

Exposing Postprocessed Copy-Paste Forgeries through Transform-Invariant Features

Pravin Kakar, *Student Member, IEEE* and N. Sudha, *Senior Member, IEEE*

Abstract—Image manipulation has become commonplace with growing easy access to powerful computing abilities. One of the most common types of image forgeries is the copy-paste forgery, wherein a region from an image is replaced with another region from the same image. Most prior approaches to finding identical regions suffer from their inability to detect the cloned region when it has been subjected to a geometric transformation. In this paper, we propose a novel technique based on transform-invariant features. These are obtained by using the features from the MPEG-7 image signature tools. Results are provided which show the efficacy of this technique in detecting copy-paste forgeries, with translation, scaling, rotation, flipping, lossy compression, noise addition and blurring. We obtain a feature matching accuracy in excess of 90% across postprocessing operations, and are able to detect the cloned regions with a high true positive rate and lower false positive rate than the state of the art.

Index Terms—Copy-paste forgeries, MPEG-7, image forensics

I. INTRODUCTION

TAMPERING images has become extremely easy due to the easy accessibility of advanced image editing software and powerful computing hardware. Various types of forgeries can be created and in recent years, image forgery detection using passive techniques has become a hot area of research [1], [2].

One of the most common types of image forgeries is the copy-paste (or copy-move or cloning) forgery, where a region from one part of an image is copied and pasted onto another part, thereby concealing the image content in the latter region. Such concealment can be used to hide an undesired object or increase the number of objects apparently present in the image. Although a simple translation may be sufficient in many cases, additional operations are often performed in order to better hide the tampering. These include scaling, rotation, lossy compression, noise addition, blurring, among others. Hence, in order to be able to reliably detect such forgeries, a few techniques have been recently proposed which try to be robust to some of these transformations. As copy-paste forgeries become more convincing, it is necessary to devise techniques which can still detect transformed regions and expose such tampering. Our proposed technique is an attempt to do this.

The key contributions of our technique are listed below:

Copyright © 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are with the Institute for Media Innovation, Nanyang Technological University, Singapore - 637553. E-mails: pkakar@pmail.ntu.edu.sg, sudha@ntu.edu.sg

This work was supported by a Ph.D. scholarship awarded to Pravin Kakar by the Institute for Media Innovation.

Fig. 1: Proposed technique for postprocessed copy-paste forgery detection

- Adapting the MPEG-7 image signature tools for use in the new application of image forgery detection
- Developing an alternative feature matching approach to the one used by the MPEG-7 standard for image signature tools in order to deal with a different problem context
- Employing matching feature constraints to improve cloned region detection via clustering
- Evaluating our novel technique on a variety of images subjected to a significant number of postprocessing operations

The rest of this paper is organized as follows. In Section II, we discuss the existing work concerning the detection of copy-move forgeries. In Section III, we discuss our proposed technique and provide results in Section IV.

II. PRIOR WORK IN COPY-MOVE FORGERY DETECTION

As discussed earlier, copy-move forgeries involve concealing one region in an image by overlaying another region from the same image. The most seemingly obvious way of detecting copied and pasted regions in the same image would be to verify small clusters or blocks of pixels for matches all across the image. However, there are two major issues with this approach. Firstly, this would be a computationally intensive approach, as matching blocks (or other shapes) of pixels would become infeasible with increasing size of the image. Secondly, such an approach would fail in case of minor changes such as addition of noise or multiple image compression. In order to circumvent these drawbacks of this direct approach, researchers have developed various techniques which can be classified into two main categories: block-based and feature-based.

A. Block-based Techniques

Some techniques use representations for dimensionality reduction [3], [4] such as PCA or frequency representation [5] such as DCT in order to efficiently find matching regions. However, they assume that the copied region has not undergone any postprocessing, which is not always the case. Nevertheless, these techniques are invariant to slight noise addition and lossy compression. The work of [6] discusses improved robustness using DCT to noise addition, global blurring and lossy compression, but does not deal with geometrical transformations of the tampered region. The technique of [7] reduces the time complexity of the PCA-based approach by using a Discrete Wavelet Transform (DWT), but again does not address geometrical transformations. In [8], the authors propose using a set of moment invariants, PCA and a kd-tree in order to efficiently detect copied regions. The authors of [9] use log-polar block descriptors to detect rotated, reflected and scaled regions. However, the method is susceptible to noise addition and compression as it uses the pixels directly. The work presented in [10] uses Zernike moments of blocks in order to detect copy-move forgeries with possible rotation, but does not address scaling, and presents results only on a small number of forgeries created with stringent constraints. The authors of [11] proposed a technique based on the Fourier-Mellin Transform which is invariant to small rotation and resizing of the copied regions as well. However, this technique fails when the rotation and/or resizing is significant. This technique was improved upon in [12] where significant rotation invariance was achieved by taking projections along angular directions. However, the scale invariance seems to be valid only over a small range, and the number of false positives yielded is quite high.

B. Feature-based Techniques

Block-based techniques essentially compare blocks in an efficient manner, and provide invariance to some transformations through an appropriate choice of the method of representation. It is seen, however, that this often results in significant false positives, and invariance to other transformations like flipping, brightness changes and blurring is hard to establish. Therefore, recently, interest in feature-based approaches has been spurred, as forgeries have become more convincing with a number of transformations being employed. Feature-based techniques try to avoid these problems by choosing to match features in the image, instead of blocks. By an appropriate selection of features, invariance to a number of transformations can be established. The rationale for this lies in the fact that the features of interest were developed for the purposes of object detection and/or content-based image retrieval, and so needed to be invariant to a large number of transformations. Our proposed technique is an example of this class of copy-move forgery detection techniques as well.

Recently, the authors of [13], [14], [15] proposed techniques to handle various transformations using SIFT features, which are extensively used in the field of computer vision. These features are, however, not robust to many postprocessing operations such as blurring and flipping. The features our

technique uses have been adopted from those which have been shown to be robust to far more operations, and therefore, we expect better performance from our technique.

III. PROPOSED TECHNIQUE

We propose a novel technique for detecting copy-paste forgeries with possible postprocessing. It is based on the MPEG-7 image signature tools [16], which form a part of the MPEG-7 standard. This set of tools was designed for robust and fast image and video retrieval. The main issue in directly applying these tools to image forgery detection is that these tools were designed to find duplicate but separate, *images*, whereas we are trying to find identical *regions* in the same image. We perform modifications in the feature extraction and matching processes to efficiently detect copy-paste forgeries.

Our major modification is in the feature matching process. We have dispensed with the process described in [17] as it is only suitable for separate images and fairly large cloned areas. In the context of CBIR, this may occur when the original version of a cropped image needs to be found. Instead, we have adopted a two-step approach matching features in feature and image spaces. Additionally, we have also modified the feature extraction process to extract a larger number of features with better resolution of components, as compared to [16]. The outline of our proposed method is shown in Fig 1. The various steps in our feature extraction process are described below.

A. Feature Extraction

1) *Feature Point Detection*: First, the given image is represented in scale-space by repeatedly smoothing with a Gaussian filter of increasing size. 12 levels in scale space are extracted. Then, the features' locations and scales are decided by using scale-adapted Laplacian of Gaussian (LoG) and Harris filters, which can indicate gradient changes and corners in the image. For the LoG filter, the output of the standard LoG filter is convolved with a scale-adapted Gaussian filter, whereas a similar approach is adopted for the Harris filter by using a multi-scale Harris operator. A Scharr operator is used as the gradient operator (as it provides more accurate gradient angle estimates than other operators like the Sobel operator) for these filters and is defined as:

$$h_x = \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \text{ and } h_y = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \quad (1)$$

2) *Feature Point Selection*: Likely candidates for feature points are selected based on having certain values resulting from the LoG and Harris filters. For points having a Harris filter response greater than a prespecified threshold, we sort the points in decreasing order of LoG filter response. The N points with the maximum responses (subject to meeting exclusion zone criteria) are selected for signature extraction. An exclusion zone of M pixels is used around each feature to maintain a minimum spatial distance between two features. This is important because it is quite likely that points showing the highest response to the filters tend to come from the same objects or regions, and we wish to avoid clustering of features in certain areas of the image.

Fig. 2: Parameterization used in computing the trace transform

3) *Signature Extraction*: A circular region around each feature point (with a radius dependent on the scale) is scaled to a radius of 32 pixels, allowing for robustness to scale changes. This region is then subjected to the trace transform [18], which is a generalization of the Radon transform. It generalizes the Radon transform by allowing for various functionals (in addition to the integral used in the Radon transform) to be applied over straight lines in the circular region. By a proper choice of a functional, invariance to rotation and scaling can be achieved. The trace transform $T(d, \theta)$ is computed by drawing samples (indexed by t) along lines parameterized by distance d (64 values from $\frac{-31.5}{\sqrt{2}}$ to $\frac{31.5}{\sqrt{2}}$) and angle θ (100 values from 0 to π) as shown in Fig 2, and applying a functional (in our case, a summation) over t . Various identifiers are then extracted from each transformed region. These include:

$$\max(\xi(x)) \quad (2)$$

and

$$\int |\xi(x)'| dx \quad (3)$$

where $\xi(\cdot)$ is some function of parameter x . In the discrete domain, (3) is replaced by a sum-of-differences operation. The functionals in the above two equations are known as invariant functionals, as they are invariant to translation of x , and scale along with x or $\xi(x)$. By considering the application of (2) and (3), along the d -dimension of T , we obtain functions over θ known as circus functions, which are used as identifiers. Further identifiers are extracted by applying (2) and (3) to versions of T subsampled along d (decomposition). By considering a 3-D representation of d , θ and t known as a trace cube, identifiers along θ (trace-annulus) and t (trace-circular) are also extracted. These correspond to circular regions and bands in the image respectively, and contribute to rotational invariance of the features. Details of the extracted identifiers may be found in [16].

For an extracted identifier s , its Fourier transform $\mathcal{S}(\omega)$ is computed and the differences between neighboring components are calculated. ω is the index of the discrete frequency components of $\mathcal{S}$.

$$c(\omega) = |\mathcal{S}(\omega)| - |\mathcal{S}(\omega + 1)| \quad (4)$$

The deviation of each component of $c(\omega)$ from its mean $\bar{c}$ is then computed:

$$b(\omega) = c(\omega) - \bar{c} \quad (5)$$

Finally, $b(\omega_i)$, $\omega_i = 2, \dots, 6$ for each identifier are concatenated to form a descriptor f_i of the feature point i . A maximum of N feature descriptors are extracted from the image in this manner.

4) *Signature Representation*: Although the original tools threshold the components of the feature descriptors to 0 or 1, we found that this caused an unacceptable loss of information for our purpose. Hence, we retain the real-valued components of the descriptors, instead of thresholding them to binary values. This also necessitated a change in the feature-matching process, as explained in the next section. In addition to the descriptors, the pixel coordinates of the feature point are also stored. Fig 1 shows an example of selected feature points after the feature extraction step.

B. Feature Matching

As the descriptor components are no longer binary, we do not employ a Hamming distance measure, as in the original tools. Instead, the Euclidean distance between the components of each pair of features (denoted by $d(\cdot, \cdot)$) is calculated, and the feature pairs below a threshold, T_A , are chosen for further analysis. This threshold is set sufficiently high to remove only the unlikeliest matching feature pairs, without eliminating any true matches. Additionally, feature pairs where the constituent features have a spatial Euclidean distance of less than 20 pixels are ignored. This is important because features which are located close to each other tend to be similar due to the smoothness constraints of natural images, without being part of cloned regions.

One of the persistent problems in the direct application of the matching process from [17] is that we do not know which region each feature point is a part of. A feature pair is unordered and so, directly considering the “first” feature of each pair as belonging to the same region can give erroneous results. On the other hand, in the context of the original problem, the image of origin for each feature was known. Therefore, we employ a different approach and improve the matching methods used in [14], [15] for use with our features. In particular, we match features by selecting pairs of likely matching features and then performing clustering in a modified spatial domain.

1) *Matching in Feature Space*: Likely matching features are selected by applying the Best-Bin-First method [19], which is the same as for SIFT features. The distance in feature space between a feature f_i and a possible match f_j , $d(f_i, f_j)$ is compared against the distances $d(f_i, f_k)$ between f_i and other possible matches $f_k, k = 1, \dots, |\mathcal{K}|, k \neq i, j$, where $\mathcal{K} = \{(f_i, f_k) : d(f_i, f_k) < T_A\}$ and $|\cdot|$ denotes cardinality. The condition tested is

$$\frac{d(f_i, f_j)}{d(f_i, f_k)} < T_B \quad \forall k \neq i, j \text{ and } d(f_i, f_k) < T_A \quad (6)$$

If this condition is true, then the pair of features (f_i, f_j) is declared to be a matching pair. This ratio should ideally be very low for matching features and close to 1 for non-matching features. However, due to factors such as poor compression, small cloned regions, etc. this is not always true. As a compromise, we set T_B to 0.75 in order to avoid eliminating matching features due to a low threshold. Instead, we eliminate false matches by using an improved clustering procedure as described in the next subsection. If the number of matching feature pairs is greater than 100, we select the 100

pairs having the lowest values of the above ratio. This helps avoid false matches in the subsequent clustering process.

2) *Matching in Image Space*: We observe that matching features are likely to cluster in regions of the image if they originate from actually copied regions within the image, whereas spuriously matched features generally do not follow a specific spatial distribution. This indicates that clustering the features in the image space offers a natural way to reject false positives in matching features.

Additionally, we observe that if a feature pair (f_i, f_j) is found to be matching, then the features comprising the pair, f_i and f_j , belong to different regions and hence should belong to different clusters. In other words, f_i and f_j should not be assigned to the same cluster. Therefore, instead of clustering directly in the image space, we modify the image space as follows:

$$g(f_i, f_j) = \begin{cases} \infty & \text{if } (f_i, f_j) \text{ is a matching pair} \\ \|f_i, f_j\| & \text{if } (f_i, f_j) \text{ is not a matching pair} \end{cases} \quad (7)$$

where $\|\cdot, \cdot\|$ denotes the spatial Euclidean distance between the locations of two feature points. The measure $g(\cdot, \cdot)$ used in clustering is no longer a distance measure, but reduces to a dissimilarity measure. Alternatively, this may be viewed as the image space being transformed to a semimetric space, where the distance measure no longer needs to obey the triangle inequality. Allowing $g(\cdot, \cdot)$ to be ∞ for a matching pair of feature points helps maximize the value of the dissimilarity measure between possible clusters and improves the clustering accuracy. Experimental results are provided in Section IV.

The dissimilarities between feature points are then used to perform clustering by using an agglomerative hierarchical clustering algorithm. Note that the above transformation of the image space into a semimetric space adds constraints to the clustering process. Therefore, we are performing constrained hierarchical clustering [20].

In agglomerative hierarchical clustering, clusters are arranged in a hierarchical tree by merging feature points. The number of clusters can be determined by cutting the tree at an appropriate level through various dissimilarity thresholds such as inconsistency threshold, distance threshold, etc. The inconsistency threshold uses an inconsistency coefficient to characterize the dissimilarity of a cluster with respect to other clusters at the same level of the hierarchy, whereas the distance threshold uses a distance measure to quantify the dissimilarity between clusters. These thresholds are used to decide the merging of two different clusters into one. Such merging is known as linkage and can be decided by various methods. For example, single linkage considers the inter-cluster dissimilarity between two clusters, say I and J to be that of the closest (most similar) pair.

$$g_{\text{single}}(I, J) = \min_{f_i \in I, f_j \in J} g(f_i, f_j) \quad (8)$$

Similarly, complete linkage considers the inter-cluster dissimilarity to be that of the least similar pair:

$$g_{\text{complete}}(I, J) = \max_{f_i \in I, f_j \in J} g(f_i, f_j) \quad (9)$$

Average linkage considers the inter-cluster dissimilarity to be

the average of the two clusters:

$$g_{\text{average}}(I, J) = \frac{1}{|I||J|} \sum_{f_i \in I} \sum_{f_j \in J} g(f_i, f_j) \quad (10)$$

We compare the performance of our technique with these linkage methods with various distance thresholds (Section IV-A). Based on the results obtained, we use a distance threshold which is a multiple of the image size on the modified image space with single linkage $g_{\text{single}}(\cdot, \cdot)$. The two largest detected clusters, denoted by i and j , are selected and the matched feature points in the clusters are stored in a set with each pair (f_i, f_j) being stored as an element of the set $\{(f_{ik}, f_{jk})\}$ with the subscript k indicating the k th pair in the set.

C. Estimation of Geometric Transformation

The geometric transformation between the two regions is estimated by using the matching pairs found from the previous steps. In this case, the MPEG-7 image signature tools use features that provide a low number of false positives which allow us to avoid using empirical threshold-dependent RANSAC for the transformation estimation, as is prevalent in the state of the art [14], [15].

Instead, we use the *Least Median of Squares* (LMedS) algorithm [21] which does not require a threshold in order to distinguish between inliers and outliers in the estimation process, but does require the majority of the data to be inliers.

For a set of matched feature pairs $\mathcal{P} = \{(f_{ik}, f_{jk})\}$, a random subset of $\mathcal{P}$ is used to estimate the geometric transformation $\mathbf{H}$ which is modeled as a perspective transformation matrix, having eight degrees of freedom:

$$\mathbf{H} = \begin{bmatrix} t_{11} & t_{12} & | & t_{13} \\ t_{21} & t_{22} & | & t_{23} \\ t_{31} & t_{32} & | & 1 \end{bmatrix} \quad (11)$$

The residual errors $\|f_{jk}, \mathbf{H}f_{ik}\|$ are used to introduce an ordering r on $\mathcal{P}$ such that

$$\|f_{jp}, \mathbf{H}f_{ip}\| \leq \|f_{jq}, \mathbf{H}f_{iq}\| \leftrightarrow r(p) \leq r(q) \quad (12)$$

By repeated sampling of random subsets, the $\mathbf{H}$ for which $\|f_{jm}, \mathbf{H}f_{im}\|$ is lowest, where m is the median of r , is selected as the correct geometric transformation.

For the various geometric transformations in Section IV-B, we consider affine transformations, the general matrix for which can be written as:

$$\mathbf{H} = \begin{bmatrix} \mu\lambda \cos(\theta) & -\lambda \sin(\theta) & | & t_x \\ \mu\lambda \sin(\theta) & \lambda \cos(\theta) & | & t_y \\ 0 & 0 & | & 1 \end{bmatrix} \quad (13)$$

In the above equation, $[t_x, t_y]$ is the translation vector, λ is the scaling factor, θ is the angle of rotation and $\mu = \pm 1$ indicates reflection along the x -axis. Note the correspondence of these terms to the general geometric transformation matrix (11).

Once the geometric transformation $\mathbf{H}$ is estimated, the image in question is warped according to this transformation. Such warping is shown in Fig 1.

D. Image Region Matching

Blockwise correlation is performed between the original and warped images. The correlation coefficient is used as a measure of correlation, as it is invariant to intensity change. The correlation output is smoothed by using a Gaussian filter to reduce the noise present, and thresholded using a prespecified threshold $T_G = 0.4$. Only regions greater than a proportion $T_H = 0.25\%$ of the area of the image are retained. This enables us to only get actual matching regions and reject most regions which only display some correlation due to the nature of the texture in the image. Any holes inside the region are filled by using a flood-fill algorithm, and the boundary of the region is extracted. This boundary is warped by employing the inverse of the estimated geometric transformation, and overlaid on the original image to indicate the copied regions, as in Fig 1. Note that unlike [15], image correlation allows us to further discard falsely detected clusters of matching features which do not yield any correlated regions after applying the geometric transformation.

IV. EXPERIMENTAL RESULTS

In this section, we present the results of our proposed copy-move forgery detection technique. First, we discuss the various parameter values used in our algorithm. We then present quantitative results and examples for the detection of copy-move forgeries subjected to many image processing operations.

A. Selection of Parameter Values

The parameter values that we use in our technique are presented in this section. These values are mostly determined empirically and the reasons for their usage are explained below.

We use a maximum of $N = 7000$ feature points, as compared to 80 in the original standard [16], in order to be able to get feature points even from non-salient regions such as textures like sand and grass, which are often used to hide objects in copy-paste forgeries [5]. An exclusion zone of $M = 2$ or 4 pixels is maintained around each feature point depending on the texture content of the image. We use the sum of the absolute gradient values from (1) as a simple measure of the amount of texture in the image. If this sum is below a threshold of 5×10^5 , then the lower value of M (2 pixels) is chosen to extract more features from the relatively few highly textured regions. The threshold T_A is set to be 10, which allows only the most dissimilar features to be eliminated, without rejecting most actual matches.

We tested the performance of our technique on the MICC-F220 database of [15], the results of which are shown in Fig 3. Following [15], the true positive rate (TPR_i) and false positive rate (FPR_i) for images are defined as follows:

$$TPR_i = \frac{\text{No. of forged images detected as forged}}{\text{No. of forged images}} \quad (14)$$

$$FPR_i = \frac{\text{No. of original images detected as forged}}{\text{No. of original images}} \quad (15)$$

An image is declared as forged if there are two clusters of matching features with at least four features in each cluster. We

Fig. 3: Performance of various linkage methods in constrained hierarchical clustering. (a) True Positive Rate (b) False Positive Rate.

tested the performance of our technique with single, complete and average linkages between clusters with varying distance thresholds. The distance threshold is set to be a fraction of the largest spatial dimension of the image. We also compared this performance by using our features with the unconstrained Ward criterion-based clustering method used in [15] which defines an inconsistency threshold to perform clustering.

As can be seen from Fig 3(a), TPR_i remains relatively high for a significant range of values of the distance threshold. However, FPR_i (Fig 3(b)) is extremely small for low values of the distance threshold with single linkage, which also has relatively high TPR_i values over the same range. We also observed that the inconsistency coefficient employed in [15] for clustering depends on the number of levels of the hierarchical tree that are included in its calculation and often lacks sufficient discrimination between its values for valid and invalid clusters. This results in lower values of TPR_i . Therefore, we use this constrained version of hierarchical clustering in our technique.

B. Invariance to Transformations

We tested our technique on a number of image forgeries subjected to various image processing operations as described below. We explain our detection results on a few example forgeries, before providing results obtained on a database in the next subsection.

1) *Translation*: An example of a copy-paste forgery is shown in Fig 4. As can be seen, we are able to detect the forged region very well with no region other than the tampered one being detected. Block-based techniques, like the ones described in Section II-A, tend to generate false positives, especially when faced with such images containing mostly uniform textures. Our technique did not detect any matching region, or even matching features, when applied to the original image in Fig 4(a), which again indicates the low false positives generated with our technique.

2) *Scaling, Rotation and Flipping*: Often it is necessary to scale, rotate and/or flip an object in order to be able to create convincing forgeries. Our technique is able to detect regions which have undergone such operations as well, as shown in Fig 5. Note that many other features, such as SIFT features, are not invariant to flipping. Therefore, techniques which use such features require additional processing (like searching for feature matches in a mirrored version of the image [14]).

Fig. 4: Detection of copy-paste forgeries. (a) Original image. (b) Tampered image. (c) Detection results.

Fig. 5: Detection of geometrical operations (images from [14]). Left column: Images with scaled and rotated + flipped regions; Right column: Detection results.

The features used in our technique are, however, invariant to flipping and so, no additional processing is required.

3) *JPEG Compression and Noise*: We tested our technique for robustness to addition of Gaussian noise (in all three color channels of the image), and JPEG compression. The results of our technique on a sample image are shown in Fig 6. The original image was a JPEG image saved at quality factor 80. A snippet of the image was copied and pasted to generate a visually convincing forgery. The tampered image was then compressed with various quality factors and subjected to the addition of Gaussian noise with different SNRs. It is to be noted that the compressed tampered image actually has double compression, as the original image itself was a JPEG compressed image. Another interesting observation is that we are able to detect the copied region, even though it constitutes less than 1% of the total image area.

4) *Illumination Change*: In order to conceal image tampering, the illumination of the copied region can be changed. We tested our technique for robustness to such changes by changing the intensity levels of all pixels of the copied region, as shown in Fig 7. The detection result is very good, and shows

Fig. 6: Robustness to noise and JPEG compression (original image from [22]). (a) Tampered image. (b) Result of resaving with JPEG quality factor of 70. (c) Result with SNR of 25 dB.

Fig. 7: Detection output for illumination change. (a) Tampered image. (b) Detection result.

Fig. 8: Detection of copy-paste forgeries with blurring. (a) Image with blurred copied region (b) Matched features. (c) Detected regions.

no significant deviation as compared to what would have been obtained with a simply copy-paste operation, or as with JPEG compression and noise addition. This is due to the correlation measure being the correlation coefficient which is invariant to differences in intensity between identical regions.

5) *Blurring*: Fig 8 shows the detection results when the copied regions are blurred. Such an operation may be applied to make the forgery harder to perceive or to create an artistic effect. We illustrate this latter case, where non-central parts of the image were blurred using a 7×7 Gaussian kernel. We have used this blurring kernel to illustrate the robustness of our technique even under heavy blurring. This is a difficult transform to deal with in general, in copy-paste forgeries, but our technique is able to give a good result with very low false positives. This can especially be seen from the fact that there are no false matches in Fig 8(b).

6) *Multiple Cloning*: In some cases, it may so happen that multiple regions are copied and pasted once within the same image in order to make the forgery difficult to detect. Such an operation is known as multiple cloning, and often attempts to obscure the regularity of the copied regions by changing their relative arrangement upon pasting. As such, a single geometric transform may prove to be insufficient in detecting all the

Fig. 9: Detecting multiple cloning. (a) Original image. (b) Tampered image. (c) First detected region pair. (d) Feature selection mask. (e) Second detected region pair. (f) Combined detection result.

copied regions.

We address this problem by iteratively applying our technique, masking out the detected region for the feature selection stage in each iteration. Consider the original image in Fig 9(a). The tampered version in Fig 9(b) is created by covering one of the dogs with a patch of grass, and a part of the dog in top left of the image is pasted onto the bottom right of the image. Applying our technique to the image in Fig 9(b) detects the large patch of grass as shown in Fig 9(c).

We use the detected region as a mask (Fig 9(d)) and repeat the process, thus ignoring the features present within the already detected regions. The second iteration yields a matching pair of regions corresponding to the partially copied dog (Fig 9(e)). The combined detection result is shown in Fig 9(f).

C. Results on a Database

We tested our technique on images obtained from the Kodak database [23]. The database contains 24 true color lossless images and is often used for research purposes (in compression testing, etc.). We created various types of forgeries by carrying out different operations on copied blocks within the images.

For creating the various forgeries, we began by selecting a block of a randomly chosen size between 32×32 and 96×96 pixels containing at least 10 feature points. This block was then processed according to the desired operation, and pasted at a random location within the image. The requirement of a minimum number of feature points is necessary as a copied block with no feature points is likely to be completely uniform, and thus the detection of its cloning would be impossible through currently available techniques.

For each of the 24 images in the Kodak database, we generated 5 images per operation. Thus, we had 120 images to evaluate our method on for each of 7 operations. Then, we performed lossy compression with 4 different JPEG compression levels and added noise with 3 different intensity levels on each of the 120 images. In the end, our synthetic database consisted of 5880 ($24 \times 5 \times 7 \times 7$) images.

Below, we discuss how the various forgeries were synthesized for evaluation purposes:

- **Translation:** The copied block was pasted on a random location in the image after simple translation, varying t_x and t_y in (13), without overlapping with the original block.
- **Scaling and Flipping:** The copied block was randomly scaled by a factor λ in the interval $(1, 1.5)$ and flipped or left unchanged randomly, according to μ in (13). The new block was then pasted on a random location in the image.
- **Rotation and Flipping:** The angle of rotation θ of the copied block was randomly selected from the interval $(0, 360)$ and flipping was performed as above. The rotated block was then pasted randomly in the image.
- **Illumination Change:** The copied block was pasted in the same manner as in the case of translation, but it also had the intensity of its pixels changed by a random factor in $(0.5, 1.5)$. The synthesized image was then subjected

Fig. 10: Sample detection results. Left column: Tampered images, Right column: Detection results. The numbers in the right column indicate TPR_r (16) and FPR_r (17) (in %).

to the compression and noise addition operations as above.

- **Blurring:** The copied block was blurred by using Gaussian kernels of sizes 3×3 , 5×5 and 7×7 , and then pasted at a random location in the image.

Each image was then compressed with JPEG quality factors of 70, 80, 90 and 100, and subjected to Gaussian noise addition with SNRs of 30, 35 and 40 dB. Some sample results are presented in Fig 10. It can be seen that we are able to detect the copied and pasted regions very well, even when they have been subjected to many image processing operations.

1) *Feature Matching Performance:* We tested the performance of our feature matching process for forgeries where the copied regions have been subjected to various geometrical transformations (translation, scaling, rotation and flipping) and image processing operations (illumination change and blurring). Additionally, varying levels of lossy compression and Gaussian noise addition were performed. The results of our feature matching process are shown in Table I. The purpose of this table is to highlight the excellent performance

Operation	JPEG-70	JPEG-80	JPEG-90	JPEG-100	SNR-30	SNR-35	SNR-40
Translation	0.9840	0.9907	0.9930	0.9981	0.9711	0.9852	0.9957
Scale + Flip	0.9553	0.9641	0.9722	0.9756	0.9251	0.9647	0.9705
Rotate + Flip	0.9610	0.9701	0.9837	0.9812	0.9365	0.9653	0.9812
Ill. Change	0.9762	0.9703	0.9875	0.9929	0.9419	0.9833	0.9834
Blur 3×3	0.9718	0.9841	0.9831	0.9905	0.9406	0.9753	0.9887
Blur 5×5	0.9573	0.9516	0.9562	0.9737	0.9297	0.9486	0.9692
Blur 7×7	0.9489	0.9360	0.9225	0.9493	0.9021	0.9545	0.9565

TABLE I: Feature Matching Accuracy of Proposed Technique

of the features that we have employed. The values reported are the proportions of true positives in the matching feature pairs determined by our technique. We obtain a true positive proportion, referred here as matching accuracy, of over 90% for various image operations, compression levels and noise intensities. Blurring tends to have somewhat lower accuracy than other operations, especially at low quality factors and SNRs, because applying the kernel actually removes information, as opposed to merely transforming it in other cases. Note that false matches in the feature pairs do not necessarily indicate a falsely matched region. This is because false matches tend to generally have a large reprojection error in the geometric transformation estimation process (Section III-C), and are discarded from consideration by the LMedS algorithm. The high accuracy at this stage means that the matched feature points themselves can easily provide a coarse indication of the copied region (without performing the geometric transformation estimation and correlation steps). This is not possible with existing methods due to the high number of false positives at equivalent stages.

2) Geometrical Transformation Estimation Performance:

Table II shows the performance of the LMedS geometric transformation estimation algorithm. For each operation listed, the statistics were calculated from a set of 300 synthesized forged images subjected to the particular operation. Each set contains images with various compression levels and SNRs. The table then lists the mean and standard deviation of the errors between the estimated values of each parameter in (11), and the ground truth values used to generate the synthesized forgeries, according to (13).

It can be seen that our technique is able to achieve quite low errors in estimating the geometric transformation in the forged images. The dark gray cells highlight the cases having the maximum error in the table. The average error for scaling, flipping and rotation with respect to the translation of the block is about a pixel, and the standard deviation is only a few pixels at most. This is very good performance, because the above transformations introduce additional inaccuracies due to the interpolation methods used to align the transformed blocks on the pixel grids. The light gray cells highlight a slight increase in the estimation of the translation parameters with an increase in the blurring kernel size. Lastly, the perspective transformation parameters t_{31} and t_{32} are identified as 0 in most cases, implying that our technique is able to successfully identify the applied geometric transformation as affine.

3) *Cloned Region Matching Performance:* Fig 11 presents some results of our technique on various kinds of image forgeries, in terms of matched regions. Denoting the sets of

Fig. 11: Performance of proposed technique subjected to various image processing operations. (a) Translation under JPEG compression. (b) Scaling+Flipping under JPEG compression. (c) Rotation+Flipping under noise addition (SNRs in dB). (d) Illumination change under noise addition (SNRs in dB). (e) Various blurring kernels under JPEG-70 compression. (f) Various blurring kernels under SNR = 30dB noise addition.

pixels in the ground truth and detected pasted regions by Φ and $\bar{\Phi}$ respectively, the true positive rate (TPR_r) is defined as

$$TPR_r = \frac{|\bar{\Phi} \cap \Phi|}{|\Phi|} \quad (16)$$

and the false positive rate (FPR_r) is defined as

$$FPR_r = \frac{|\bar{\Phi} \setminus \Phi|}{|\bar{\Phi}|} \quad (17)$$

with $|\cdot|$ denoting cardinality. As can be seen, we are able to attain quite high accuracies at low false positive rates. In the case of blurring, it can be seen that the performance tends to suffer with increasing size of the kernel, especially in the case of noise addition. This is probably due to the greater loss of information being caused by larger kernels, which along with the addition of noise, makes it difficult for the correlation process to work. Nevertheless, TPR_r and FPR_r are quite acceptable even with low quality factors and SNRs.

D. Comparisons

A quantitative analysis of various copy-move forgery detection methods has been performed on a database in [22]. Examples of detection of copy-move forgeries by our method for this database are shown in Fig 12(a)-(c). For the image

Operation	Error in Parameter (m = mean, s = standard deviation)															
	t_{11}		t_{12}		t_{13}		t_{21}		t_{22}		t_{23}		t_{31}		t_{32}	
	m	s	m	s	m	s	m	s	m	s	m	s	m	s	m	s
Translation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Scale + Flip	0.0	0.1	0.0	0.1	-0.8	5.0	0.0	0.1	0.0	0.1	-0.3	3.6	0.0	0.0	0.0	0.0
Rotate + Flip	0.0	0.1	0.0	0.1	-1.1	5.3	0.0	0.1	0.0	0.1	-1.2	4.0	0.0	0.0	0.0	0.0
Ill. Change	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blur 3×3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blur 5×5	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blur 7×7	0.0	0.0	0.0	0.0	0.1	2.2	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0

TABLE II: Performance of Geometric Transformation Estimation Technique

in Fig 12(c), the small falsely matched regions along the shoreline arise because of the geometrical transformation between the cloned regions resulting in a high correlation coefficient along the relatively uniform shoreline. However, no matching features are found in the falsely matched regions by our proposed technique which indicates that the false positives are indeed caused by the nature of the image content, and are unavoidable if the cloned regions are to be indicated. Since our method specifically addresses copy-move forgeries that undergo various forms of postprocessing, we compare our method with various aspects of the state-of-the-art methods in [14], [15] having the same capabilities. These use SIFT features to detect copy-move forgeries.

As seen in Table I the feature matching accuracy is considerably high, while SIFT features tend to generate considerable mismatches (arising from clutter, large transformations, etc.) as mentioned in [14], though no quantitative data is provided. Additionally, it is not required to handle pure translation and flipping in a special manner, unlike [14]. This allows us to simplify our algorithm without considering certain operations as special cases. Lastly, we are able to handle additional operations like blurring, which SIFT features are not invariant to.

In order to compare the performance of the features and feature-matching process used by our proposed technique against the state of the art, we tested the feature matching accuracy of SIFT features using the Best-Bin-First and unconstrained agglomerative hierarchical clustering algorithms as in [15]. This matching process quite similar to that in [14], although no detailed information about clustering is provided in the latter. We used the same parameter values reported in [15] for our experiments, and only tested SIFT features on the transformations they are designed to be robust to. The technique of [15] tries to find if a single region has been cloned multiple times and therefore retains all clusters having more than 3 points to use for geometric transformation estimation. In order to ensure a fair comparison, we also considered the case where only the two largest clusters (corresponding to a single cloning operation) are retained, similar to our technique. The feature matching accuracy of SIFT features for both cases is shown in Table III. The results of Table III are generated on the same database as used earlier for Table I. It can be seen that our features and feature-matching process have considerably higher performance (Table I). Moreover, as the proportion of true positives is generally below 50% for SIFT, the LMedS algorithm would be unsuitable. This necessitates the use of

a threshold-dependent RANSAC algorithm for geometrical transformation estimation instead.

The results presented in Fig 11 were obtained for forgeries generated from the Kodak database, which was also used by [14]. Our synthetic forgery database is 20% larger than the one used by the latter. There are also other differences in the method of generation of the forgeries, such as a variable block size with a lower minimum required number of feature points, larger range of illumination change and consideration of flipping along with rotation and scaling (Section IV-C). In general, our method provides comparable or better results as compared to [14]. For example, consider the ROC curves presented in Fig 13 (a) and (b). Under JPEG compression, our method performs significantly better than [14], even though approximately half the images under consideration have their forged regions flipped. Although the difference is lower under noise addition, we get comparable or better performance with TPR_r greater than 80%. As another example, consider Fig 13(c). In [14], the forgeries with illumination changes were created by modulating the values of pixels in the copied region of 64×64 pixels to 80% of their original values. In our case, we allowed for random increments or decrements of up to 50% in intensity in block sizes ranging from 32×32 to 96×96 pixels, which is a considerable deviation. It can be seen that our method performs significantly better at lower JPEG quality factors (around 70) as compared to [14] and provides comparable performance at relatively high quality factors. The slightly worse performance for a quality factor of 90 is caused by the wider range of intensity deviation and the presence of some smaller copied regions in our case. The reason for the overall improved performance is partly due to the fact that the features that we use are also intended to be blur-invariant, and lossy compression causes a loss of high-frequency information similar to blurring.

Our method also does not generate any false positives on authentic images containing somewhat repetitive patterns. The method of [14] declares authentic images similar to those shown in Fig 14 as tampered, which our method does not. Moreover, none of the feature points extracted are declared to be matching which again validates the low false positives generated by our technique.

The work of [15] also uses SIFT features to detect copy-paste forgeries but does not address flipping and illumination change of the cloned regions. Moreover, unlike our method and [14], only the geometrical transformation between the cloned regions is estimated, without indicating the cloned

Fig. 12: Results of detection on sample images from: (a)-(c) Database of [22]; (d) and (e) Database of [15]. Top row: Tampered images; Bottom row: Detection results.

Operation	JPEG-70	JPEG-80	JPEG-90	JPEG-100	SNR-30	SNR-35	SNR-40
Translation	0.2947	0.3145	0.3587	0.3690	0.3089	0.3447	0.3511
	0.4177	0.4639	0.4764	0.5154	0.3997	0.4609	0.4859
Scaling (without flipping)	0.2511	0.2659	0.3083	0.3107	0.2662	0.2951	0.3071
	0.3378	0.4131	0.4457	0.4301	0.3728	0.4205	0.4201
Rotation (without flipping)	0.2801	0.2972	0.3343	0.3330	0.3016	0.3175	0.3220
	0.3582	0.4271	0.3972	0.4545	0.4036	0.4187	0.3929
Illumination Change	0.2873	0.3042	0.3458	0.3567	0.2986	0.3325	0.3575
	0.4022	0.4502	0.4632	0.4600	0.3759	0.4562	0.4605

TABLE III: Matching accuracy of SIFT features with unconstrained hierarchical clustering. Gray cells indicate values with two largest clusters retained; white cells indicate values with all significant clusters retained.

Fig. 13: Comparison of performance of our method and [14]. (a) Rotation (as well as flipping in our method) with JPEG. (b) Rotation (as well as flipping in our method) with noise addition. (c) Illumination change with JPEG compression.

regions themselves. Therefore, we use (14) and (15) in order to compare our technique with [15] on the MICC-F220 database of the latter. As seen from Fig 3, we achieve a TPR_i of around 90% and a FPR_i of less than 3%. In comparison, the technique of [15] achieves a TPR_i of 100% and a FPR_i of 8%. Again, the low false positive rate of our technique while still maintaining a high true positive rate is evident. Examples of detection of copy-move forgeries by our method for this

Fig. 14: Authentic images containing repetitive patterns which are correctly identified as untampered by our method.

database are shown in Fig 12(d) and (e).

Fig 15 shows cases where our technique partially or completely fails to detect the cloned regions. For the image in Fig 15(a), we are able to detect the copied region of grass quite well. However, the cloned bird (located behind the horn) is not detected as no reliable feature points are extracted from that region. For the image in Fig 15(c), the extremely repetitive nature of the image content gives many falsely matched feature points with no regions detected as cloned by our technique. Note that the cloned regions are the lion heads in the center of the image. Although we are not able to detect the cloned region, our proposed technique does not find any other region matched either, which again indicates the low false positive rate of our technique. The techniques of [14] and [15] also result in similar partial or complete failure in these cases.

V. CONCLUSIONS

Copy-move forgeries are a common type of forgery where parts of an image are replaced with other parts from the same image. The copied and pasted regions may be subjected to various image transformations in order to conceal the tampering better. Conventional techniques of detecting copy-paste

Fig. 15: Cases of partial or complete failure for our technique. (a) Tampered image containing multiple cloned regions. (b) Detection result for (a). (c) Image containing small cloned region with extremely repetitive content.

forgeries usually suffer from the problems of false positives and susceptibility to many image processing operations.

In this paper, we have proposed a technique based on the MPEG-7 image signature tools, which have been developed for robust content-based image retrieval, in order to detect copy-move forgeries. We have modified the tools in many ways to deal with copied regions in a single image. We have used a feature matching process that utilizes the inherent constraints in matched feature pairs to improve the detection of cloned regions. We have analyzed the performance of this technique on actual and synthesized forgeries. The results obtained by using these features display high true positive rates and extremely low false positives, and are better than the state of the art, in general. As these descriptors are invariant to a lot of common image processing operations (scaling, rotation, flipping, noise addition, JPEG compression, blurring), their use in dealing with realistic copy-paste forgeries is justified.

In our future work, we will investigate the use of our technique in detecting regions which have undergone non-affine transformations and/or are multiply copied.

REFERENCES

- [1] H. Farid, "A survey of image forgery detection," *IEEE Signal Process. Mag.*, vol. 2, no. 26, pp. 16–25, 2009.
- [2] B. Mahdian and S. Saic, "A bibliography on blind methods for identifying image forgery," *Image Communication*, vol. 25, no. 6, pp. 389–399, 2010.
- [3] A. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Dept. Comput. Sci., Dartmouth College, Tech. Rep. TR2004-515*, 2004.
- [4] H. Lin, C. Wang, and Y. Kao, "Fast copy-move forgery detection," *WSEAS Transactions on Signal Processing*, vol. 5, no. 5, pp. 188–197, 2009.
- [5] J. Fridrich, B. Soukal, and A. Lukáš, "Detection of copy-move forgery in digital images," in *Proceedings of Digital Forensic Research Workshop*, 2003.
- [6] Y. Huang, W. Lu, W. Sun, and D. Long, "Improved DCT-based detection of copy-move forgery in images," *Forensic Science International*, vol. 206, no. 1-3, pp. 178–184, 2011.
- [7] S. Khan and A. Kulkarni, "Reduced Time Complexity for Detection of Copy-Move Forgery Using Discrete Wavelet Transform," *International Journal of Computer Applications IJCA*, vol. 6, no. 7, pp. 31–36, 2010.
- [8] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic science international*, vol. 171, no. 2-3, pp. 180–189, 2007.
- [9] S. Bravo-Solorio and A. Nandi, "Passive forensic method for detecting duplicated regions affected by reflection, rotation and scaling," in *European Signal Processing Conference*, 2009, pp. 824–828.
- [10] S. Ryu, M. Lee, and H. Lee, "Detection of copy-rotate-move forgery using zernike moments," in *International Workshop on Information Hiding*. Springer, 2010, pp. 51–65.
- [11] S. Bayram, H. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1053–1056.
- [12] W. Li and N. Yu, "Rotation robust detection of copy-move forgery," in *IEEE International Conference on Image Processing*, 2010, pp. 2113–2116.
- [13] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using SIFT algorithm," in *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008, pp. 272–276.
- [14] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 857–867, 2010.
- [15] I. Amerini, L. Ballan, R. Caldelli, A. Del 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, pp. 1099–1110, 2011.
- [16] *Image Signature Tools*, ISO/IEC Std. ISO/IEC 15938-3:2002/Amd.3:2009(E). [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50550
- [17] *Extraction and use of MPEG-7 descriptions*, ISO/IEC Std. ISO/IEC 15938-8:2002/Amd.5:2010(E). [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53594
- [18] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 811–828, 2001.
- [19] J. Beis and D. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc, 1997, pp. 1000–1006.
- [20] D. Klein, S. Kamvar, and C. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," in *Proc. Int. Conf. on Machine Learning*, 2002, pp. 307–314.
- [21] P. Rousseeuw and A. Leroy, *Robust regression and outlier detection*, ser. Applied Probability and Statistics. Wiley-Interscience, New York, 1987, pp. 197–206, vol. 3, ISBN 0471852333.
- [22] V. Christlein, C. Riess, and E. Angelopoulou, "On rotation invariance in copy-move forgery detection," in *IEEE International Workshop on Information Forensics & Security*, 2010, pp. 1–6.
- [23] R. Franzen, "Kodak lossless true color image suite." [Online]. Available: <http://r0k.us/graphics/kodak/>

Pravin Kakar (S'10) obtained the B.Tech. degree in Electronics Engineering from the Veermata Jijabai Technological Institute (VJTI), Mumbai, India in 2009. He is currently pursuing the Ph.D. degree in Computer Engineering at the Institute for Media Innovation, Nanyang Technological University (NTU), Singapore. He is also working as a Project Officer at gameLAB in the School of Computer Engineering at NTU.

His research interests include multimedia forensics, computer vision, image processing and multimedia signal processing.

N. Sudha (M'99-SM'07) received her B.E. degree from Madurai Kamaraj University, India in 1992, M.S from Indian Institute of Technology (IIT) Madras in 1997 and Ph.D from IIT Guwahati in 2001. She has been working as an Assistant Professor in the School of Computer Engineering, Nanyang Technological University, Singapore since 2006. Previously she was employed at IIT Madras and IIT Guwahati.

Her research areas include computer vision and image processing, biometrics, embedded systems

and neural networks.