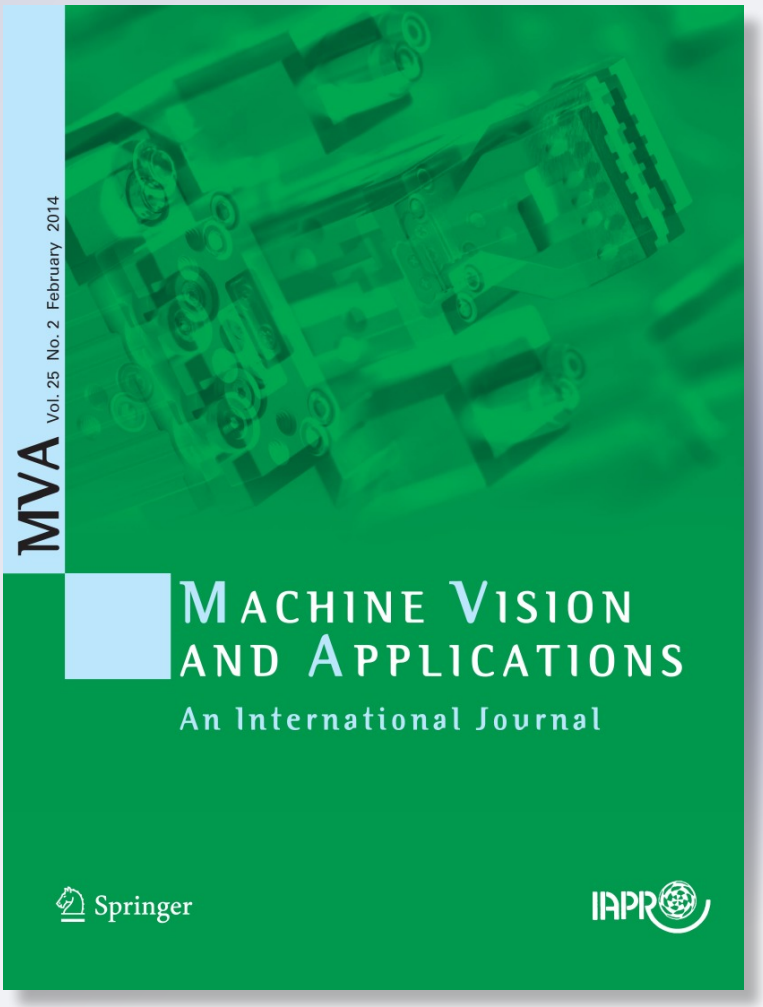
Accurate and robust localization of duplicated region in copy–move image forgery

Maryam Jaberi, George Bebis, Muhammad Hussain & Ghulam Muhammad

Machine Vision and Applications

ISSN 0932-8092
Volume 25
Number 2

Machine Vision and Applications (2014)
25:451-475
DOI 10.1007/s00138-013-0522-0



Springer

Machine Vision and Applications (2014) 25:451–475
DOI 10.1007/s00138-013-0522-0

FULL LENGTH PAPER

Accurate and robust localization of duplicated region in copy–move image forgery

Maryam Jaberi · George Bebis · Muhammad Hussain · Ghulam Muhammad

Received: 22 October 2012 / Revised: 1 March 2013 / Accepted: 15 May 2013 / Published online: 4 June 2013

Abstract Copy–move image forgery detection has recently become a very active research topic in blind image forensics. In copy–move image forgery, a region from some image location is copied and pasted to a different location of the same image. Typically, post-processing is applied to better hide the forgery. Using keypoint-based features, such as SIFT features, for detecting copy–move image forgeries has produced promising results. The main idea is detecting duplicated regions in an image by exploiting the similarity between keypoint-based features in these regions. In this paper, we have adopted keypoint-based features for copy–move image forgery detection; however, our emphasis is on accurate and robust localization of duplicated regions. In this context, we are interested in estimating the transformation (e.g., affine) between the copied and pasted regions more accurately as well as extracting these regions as robustly by reducing the number of false positives and negatives. To address these issues, we propose using a more powerful set of keypoint-based features, called MIFT, which shares the properties of SIFT features but also are invariant to mirror reflection transformations. Moreover, we propose refining the affine transformation using an iterative scheme which improves the estimation of the affine transformation parameters by incrementally finding additional keypoint matches. To reduce false positives and negatives when extracting the copied and pasted regions, we propose using "dense" MIFT features, instead of standard pixel correlation, along with hysteresis thresholding and morphological operations. The proposed approach has been evaluated and compared with competitive approaches through a comprehensive set of experiments using a large dataset of real images (i.e., CASIA v2.0). Our results indicate that our method can detect duplicated regions in copy–move image forgery with higher accuracy, especially when the size of the duplicated region is small.

Keywords Blind image forensics, Copy–move image forgery, SIFT, MIFT, Matching

M. Jaberi · G. Bebis (✉)
Computer Science and Engineering Department,
University of Nevada, Reno, USA
e-mail: bebis@cse.unr.edu

M. Jaberi
e-mail: mjaberi@cse.unr.edu

M. Hussain
Computer Science Department, King Saud University,
Riyadh, Saudi Arabia
e-mail: mhussain@ksu.edu.sa

G. Muhammad
Computer Engineering Department, King Saud University,
Riyadh, Saudi Arabia
e-mail: ghulam@ksu.edu.sa

1 Introduction

Recent advances in imaging technologies, both in hardware (e.g., digital cameras) and software (e.g., image editing applications), have enabled manipulating digital image contents easily in order to hide or create misleading images with no observable trace [1]. Establishing the authenticity of images, however, is of essence in many applications such as criminal investigation, medical imaging, journalism, intelligence services, and surveillance systems [7,24]. Recently, the field of digital forgery detection has been introduced to address this issue and has become a very important field in image processing. Digital altering has already appeared in many disturbing forms [1] and there have been several research studies on improving image forgery techniques [25]. These techniques usually include deleting or hiding a region in the image, adding a new object to the image or representing the

image information in an incorrect way. Based on the operation used to create a tampered image, techniques can be categorized into three main groups: image retouching, copy–paste (i.e., splicing), and copy–move (i.e., cloning) [2].

Image retouching manipulates an image by enhancing or reducing certain features of the image without making significant changes on image content [30]. Image splicing on the other hand utilizes two or more images to create a tampered one. This technique adds a part of an image into another image in order to hide or change the content of the second image [2]. Finally, image cloning creates a forged image by copying a certain portion of an image and moving it to another location of the same image in order to conceal or duplicate some part of the image [4]. The key characteristic of image cloning is that, since the duplicated region is picked from the image itself, the noise components, texture and color patterns are compatible with the rest of the image. Thus, it is not easy to detect the forgery parts. Moreover, there might be post-processing operations that can even make the exposing procedure harder [3]. Figure 1 shows an example of copy–move forgery.

Developing reliable methods for image forgery detection has become an active research topic [7]. Approaches in the literatures can be divided into two main categories: *active* and *passive* [2,6]. Active approaches, like watermarking, try to expose digital tampering by adding prior information to the images (e.g., a signature) [2]. Passive or blind approaches, on the other hand, attempt to detect forgeries in images without assuming any knowledge of the original images or adding any prior information to the images. The aim of these approaches is to demonstrate the possibility of detecting forgeries in the absence of any watermark [7].

In this study, our focus is on detecting copy–move (i.e., cloning) image forgery. Among the blind image forgery detection methods proposed in the literature, pixel-based approaches are the most popular; the key idea is exposing image tampering by analyzing pixel level correlations [2].

Fig. 1 Copy–move image forgery: original image (*left*) and forged image (*right*)

In general, pixel-based approaches for copy–move forgery detection can be classified into categories: *block matching* and *feature matching* [2]. The key idea behind these methods is discovering and clustering similar parts in an image.

In the first category of methods, block matching is used to detect duplicated regions. The simplest approach is using exhaustive search to find and match similar image regions [4]. The main drawback of this approach is high computational complexity and inefficiency. To cope with this challenge, several other approaches have been proposed. In an early approach [4], quantized discrete cosine transform (DCT) coefficients were used assuming overlapped blocks. DCT blocks were stringed in vectors and sorted lexicographically; copied blocks were then detected by finding similar block pair vectors that had an equal offset as well. The method proposed in [5] employed principal component analysis (PCA) to reduce data dimensionality and improve robustness to additive noise. In [1], seven different features were extracted to describe blocks. The first three features were extracted using the average values in the three-color channels. The rest of the features were computed by dividing the block into two equal parts in four directions and finding the proportion of the first part to the sum of the two parts. Sorting the vectors and finding similar blocks were carried out next, similar to other methods.

In [6], nine normalized features in overlapped blocks were extracted. These features were obtained by computing the ratio of sub-averages and total average in each block. Radix sort was applied instead of lexicographic sort. In [7], a blur invariant representation for each overlapped block was employed to extract the feature vectors of each block. Besides, PCA was used to reduce the number of features and kd-trees to find similar blocks. In [8], discrete wavelet transform (DWT) and singular value decomposition (SVD) were used in order to reduce the dimensionality of images before sorting the vectors lexicographically and checking for duplicates. A similar approach was presented in [3] and then completed in [9] where DWT was employed to reduce image dimension and the phase correlation was used to detect duplication zones. In particular, the copied blocks were distinguished at the coarsest level of DWT and verified in finer levels. In [10], a new technique was introduced using one-level DWT. The low-frequency sub-band was selected as a low-dimensional image and the diagonal detail coefficients were considered as a resource to estimate noise in each part of the image. It was assumed that interesting blocks have similarities in the low-frequency band and dissimilarities in the diagonal detail band, which are in fact noise. An extension of this method can be found in [39].

The methods described above perform block matching to detect copy parts in forged images; however, they rarely consider large variations in scaling, rotation and illumination, very common operations in image manipulation. To over-

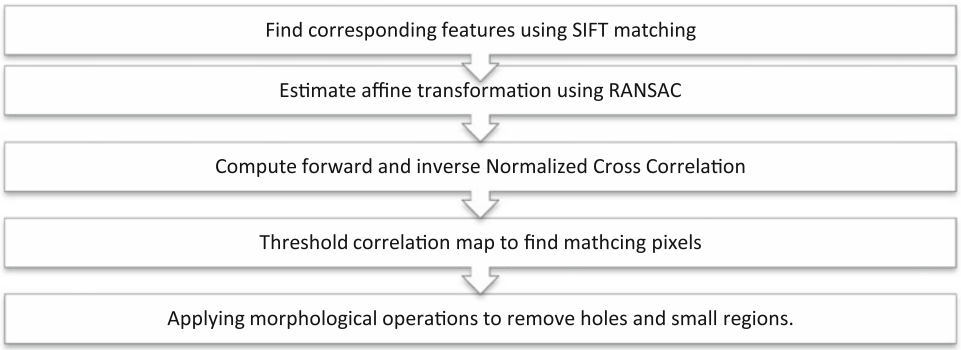


Fig. 2 Main steps of the method of Pan et al. [12]

come this issue, a different category of approaches in copy–move forgery detection try to emphasize the use of feature matching for detecting forged regions in images. The method presented in [11] employed local statistical features, known as scale invariant feature transform (SIFT) [14]. Since, SIFT features are invariant to changes in illumination, rotation, and scaling, looking for similar features in an image could reveal potential image forgery [12]. Huang et al. [11] adopted this idea to detect image forgery. Using SIFT features for image forgery detection has been adopted in several other studies including Pan and Lyu [12] and Amerini et al. [13] where the authors used almost similar techniques to find similar features and potentially interesting areas. An affine transformation between matching regions was estimated using Random Sample Consensus (RANSAC). The method proposed by Pan and Lyu [12] includes a verification step which tries to locate the duplicated regions using the normalized correlation map and thresholding. The steps of this algorithm are summarized in Fig. 2.

As shown in our experimental results, a weakness of Pan's method, as well as similar methods [11,13], is that they cannot localize the forged region very accurately. Moreover, these methods were evaluated on a relatively small number of real forged images.

In this study, we improve on copy–move forgery detection using keypoint-based features (e.g., SIFT) by focusing on the issue of accurate detection and localization of duplicated regions. Specifically, we have made several contributions in this work. First, we employ mirror reflection invariant feature transform (MIFT) features [20] instead of SIFT features for finding similar regions in images. MIFT features share all good properties of SIFT features but are also invariant to mirror reflection transformations. Like in other approaches, we find similar regions by finding corresponding MIFT features and estimate an affine transformation between them using RANSAC [26]. Corresponding MIFT features define the initial detection window. Second, since the quality of the affine transformation computed is critical in localizing the duplicated region accurately, we refine the parameters of the affine transformation iteratively by increasing the detection window slowly and computing new MIFT features. Third, to extract the duplicated region, we use dense MIFT features and apply hysteresis thresholding [12] instead of standard thresholding, to reduce false positives and negatives. To further reduce false positives and negatives, we apply morphological operations on the low and high hysteresis thresholded images and combine the results. We have evaluated the performance of the proposed methodology by performing a comprehensive set of experiments using a large database of real images (i.e., CASIA v2.0). To better understand the strengths and weaknesses of the proposed method, we have analyzed independently the effects of different types of transformations (e.g., rotation, scale, reflection, blur, deformation) as well as combinations of them. Comparisons with competitive approaches show that the proposed method can detect duplicated regions in copy–move image forgery more accurately, especially when the size of the duplicated regions is small.

The rest of this paper is organized as follows: Sect. 2 briefly reviews the problem of local feature extraction. Section 3 describes the steps of the proposed approach in detail. Section 4 presents our experimental results and comparisons. Finally, Sect. 5 concludes our work and discusses directions for future research.

2 Local feature extraction

Recently, significant research has been performed on extracting local invariant features with application to object recognition and categorization [14], image matching and retrieval [15] and video mining. The goal of local feature extraction methods is to find interest points (or keypoints) and to define a distinctive descriptor for the each of them which is invariant to transformations such as scale, rotation, or affine. More precisely, the key characteristics of these methods are their distinctiveness, robustness to occlusion and clutter, and light invariance [16]. Methods for finding local features can be classified as sparse or dense. Sparse methods compute a descriptor for each keypoint by selecting a small patch around

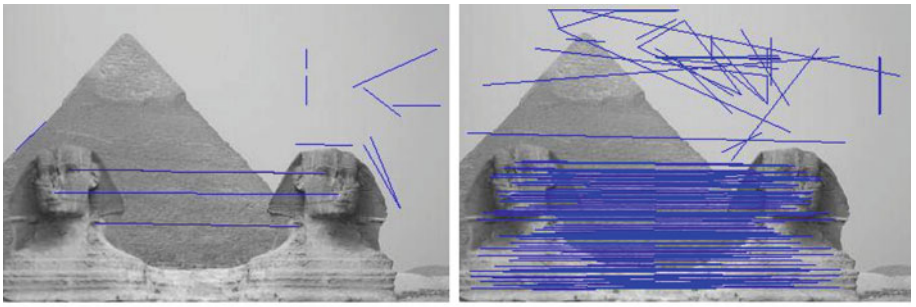


Fig. 3 Comparing MIFT and SIFT assuming mirror reflection

it. Dense methods, on the other hand, do not extract any keypoints explicitly but select a small patch around each pixel and compute a descriptor [17]. The descriptors typically are defined in the form of a vector of measured values inside the patches. These image measurements can emphasize different image properties like pixel intensity changes in a region or curvatures.

Different algorithms have been proposed for keypoint extraction. Among them, the Harris corner detector is one of the most popular algorithms for extracting keypoints invariant to translation, rotation, and partially to illumination [18]. The SIFT, proposed by Lowe [14], extracts a sparse set of keypoints using a similar algorithm. The method involves four main stages: scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor [14]. The descriptor produced is a normalized 128-element vector for each extracted keypoint [14,18]. Due to the success of SIFT, many studies have attempted to improve its performance both in terms of accuracy and time complexity. PCA-SIFT [23] is a variation of SIFT, which projects gradient images to a lower dimension using PCA. Histogram of oriented gradients (HOG) [19] employs normalized local histograms to build the descriptor. RIFT [35] is a rotation-invariant extension of SIFT which is constructed using circular normalized patches divided into concentric rings of equal width. Rotation invariance is achieved by measuring orientation at each point relative to the direction pointing outward from the center. Speeded up robust features (SURF) [36] speeds-up computations using a fast approximation of the Hessian matrix and "integral images".

These methods, however, cannot handle reflection. Mirror reflection invariant feature [20] generalizes SIFT by producing mirror reflection invariant descriptors. The resulted descriptor is invariant to mirror reflection as well as to other transformations such as affine. In general, mirror reflection can be defined in the horizontal or vertical direction as well as a combination of both directions. As explained in [20], to handle mirror reflection, it is adequate to deal with the horizontally or vertically reflected images. While the traditional SIFT approach uses a fixed order to organize the cells, making the descriptor sensitive to mirror reflection, MIFT reorganizes the order of the cells and restructures the order of orientation bins in each cell. We have adopted MIFT descriptors in this work to find duplicated regions with or without mirror reflection. Figure 3 shows an example using SIFT and MIFT in the case of mirror reflection.

As mentioned earlier, dense descriptors are extracted at each pixel location and are usually used in texture and background classification [21]. Local binary pattern (LPB) [33,34], HOG-LBP [22], and Weber local descriptors (WLD) [17] are some popular methods in this category. In this paper, we use dense MIFT features for extracting the duplicated region more accurately.

3 Method overview

The key objectives of the proposed approach are (1) to recognize copy–move manipulated images, (2) to classify images as forged or non-forged, and (3) to accurately locate the duplicated region in the tampered images. Since in copy–move image forgery a part of the image is copied and pasted on another part of the same image, finding similar parts in an image is the key idea explored here as well as in other studies. This is accomplished using feature extraction methods (e.g., SIFT) to extract and match local features from various regions of the image in order to find similar regions. Figure 4 illustrates the main steps of our approach.

Keypoint extraction and matching is the first step of our method; we have experimented both with SIFT and MIFT features for comparison purposes. The next step involves finding corresponding features which allow us to find similar regions and estimate an affine transformation between them. The affine transformation parameters are later refined which allows for localizing the duplicated regions more accurately. To extract the duplicated regions, we employ dense MIFT features, instead of standard correlation, along with hysteresis thresholding and morphological operations for reducing false positives and negatives.

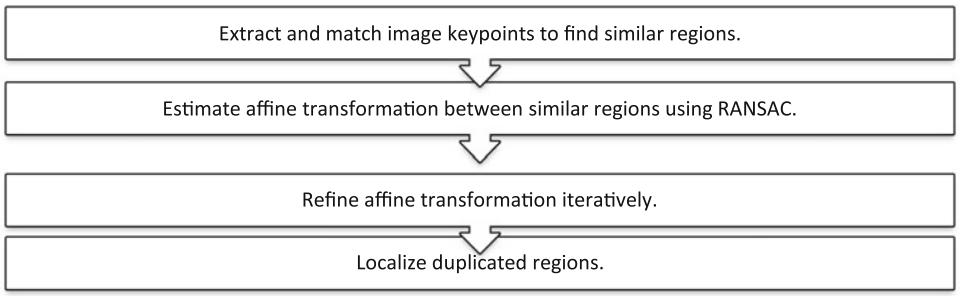


Fig. 4 Main steps of proposed methodology

3.1 Extracting keypoints and establishing correspondences

Copy–move image forgery detection requires detecting the duplicated region in a single image. Feature extraction algorithms find correspondences between similar regions in the same image. As described in Sect. 2, SIFT is a powerful technique, which extracts features invariant to scale, rotation, and brightness. However, SIFT descriptors are not invariant to mirror reflection. To account for this issue, previous approaches proposed extracting SIFT descriptors from horizontally and vertically reflected versions of the original image [12,13]. In this paper, we have adopted MIFT features which are invariant to reflection.

As described earlier, each keypoint is characterized by a feature vector that consists of a set of image statistics collected at a local neighborhood around the keypoint. In general, to find matching keypoints between images, the keypoints from one of the images are first indexed using a kd-tree [32]; then, the matching keypoints from the other image are identified. Due to the high dimensionality of the feature vectors, the search within the kd-tree is performed using the "best bin first" search algorithm [31,32]. In our case, since we search for duplicated regions in a single image, we divide the image into smaller parts and compare the descriptors among them. The search is performed outside a small window centered at the detected keypoint to avoid finding nearest neighbors of a keypoint from the same region [12]. Once a matching candidate has been found, it is accepted as a distinctive matched point if the ratio of the distances from the first and second nearest neighbors is smaller than the threshold [14]. This threshold can vary from zero to one; a threshold closer to zero yields more accurate but fewer matches. Here, a low threshold is utilized since it reduces false matches. Figure 5 shows an image and the extracted keypoints.

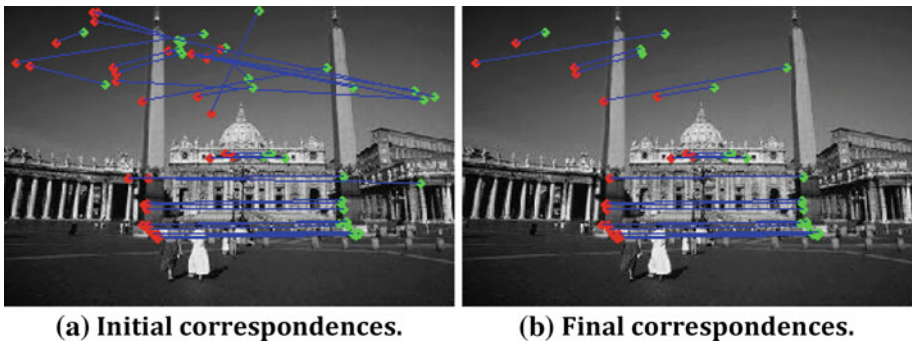
Fig. 5 An example of extracted keypoints in an image

3.2 Estimating affine transformation from keypoint correspondences

Using the keypoint correspondences from the previous step, an affine transformation is estimated. The transformation can be used to verify whether two regions correspond by mapping one region to the other. To eliminate incorrectly matched keypoints before estimating the affine transformation parameters, a pre-processing step is applied using some simple geometric constraints. To further remove incorrect matches, the affine transformation parameters are estimated using RANSAC [26] which can estimate the model parameters with a high degree of accuracy even when a significant number of errors are present.

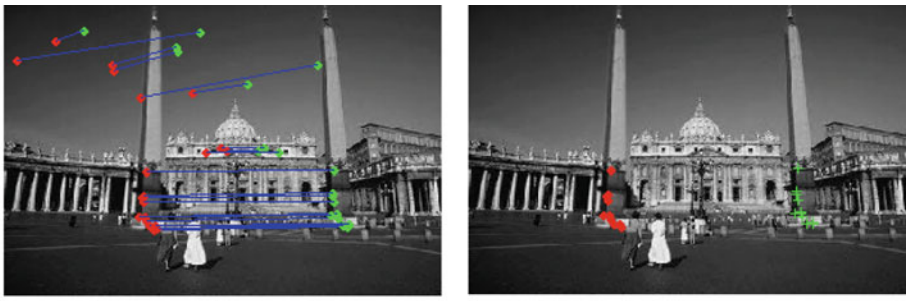
3.2.1 Removing incorrect matches

Corresponding keypoints extracted during matching can lie in different parts of the image. Since in copy–move image forgery a region of an image is duplicated, matching keypoints should lie within two regions; we refer to this as the "location" constraint. Moreover, if we were to connect with lines corresponding keypoints in those regions, then the lines formed should have similar slopes; we refer to this as the "slope" constraint. We take advantage of these two geometric constraints to eliminate incorrect matches between keypoints. To apply the "slope" constraint, we find the slope of all lines connecting corresponding keypoints and cluster them in different groups. The group with the largest number of keypoints is selected as the main group. Then, we compare all other groups to the main group and eliminate any group having a different slope (i.e., within a threshold) from the slope of the main group. Next, we apply the "location" constraint on the remaining groups by eliminating groups containing a small number of correspondences as well as removing corresponding keypoints from groups if the keypoint locations are rather far (i.e., within a threshold) from the average keypoint



(a) Initial correspondences. (b) Final correspondences.

Fig. 6 Removing incorrect correspondences using geometric constraints



(a) Initial matches (i.e., input to RANSAC) (b) Final matches (i.e., RANSAC inliers)

Fig. 7 Removing incorrect matches using RANSAC

location of the group. Figure 6 shows an example of corresponding keypoints before and after removing mismatched keypoints.

3.2.2 Estimating affine transformation

Although the geometric constraints described in the previous section can be used to eliminate many incorrect matches, they cannot eliminate all of them as it is evident from Fig. 6b. To further remove incorrect matches, we apply the RANSAC algorithm [26]. RANSAC is a simple, yet powerful parameter estimation approach designed to cope with a large proportion of outliers in the input data. In essence, RANSAC is a resampling technique that generates candidate solutions using the minimum number data points required to estimate the underlying model parameters. This algorithm estimates a global relation that fits the data, while simultaneously classifying the data into inliers (points consistent with the relation) and outliers (points not consistent with the relation). Due to its ability to tolerate a large fraction of outliers, RANSAC is a popular choice for a variety of robust estimation problems [27,28].

Using RANSAC, the affine transformation is calculated iteratively by selecting three or more non-collinear keypoints from all possible pairs. The affine transformation is then estimated based on these nominated points. The accuracy of the parameters is examined by classifying all available corresponding keypoints into inliers and outliers. Considering the fact that the estimated transformation should map more keypoints to their correspondences with a smaller error, the accuracy of the transformation is estimated by finding how well the keypoints map to their correspondences. If the difference of a mapped point and its correspondence is less than the threshold, then it will be selected as inlier; otherwise, it will become an outlier. As mentioned earlier, the algorithm calculates the transformation iteratively and selects the parameters that yield the largest set of inliers. Figure 7 shows an example of applying RANSAC on the matches found from the previous step. As it can be observed, RANSAC was able to find a highly accurate set of correspondences.

The affine transformation matrix computed by RANSAC can be used to locate the duplicated region in the image. However, to estimate the duplicated region more accurately, we apply one more step to further refine the affine transformation parameters.

3.3 Refining affine transformation

The purpose of this step is to refine the affine transformation parameters estimated from the previous step. As Fig. 7 shows, there are cases where the correspondences selected as inliers do not cover well the region of duplication. Thus, the estimated affine transformation is not precise enough to map the whole duplicated region to the copied region. In this step, we refine the affine transformation parameters iteratively, by slowly increasing the search window around the corresponding regions. Figure 8 shows the main steps of the refinement process.

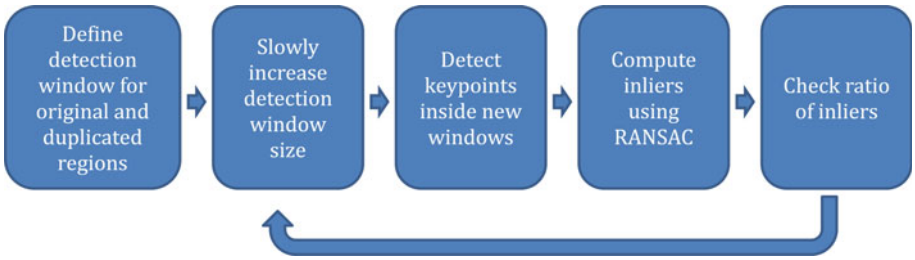


Fig. 8 Mains steps of refining the affine transformation

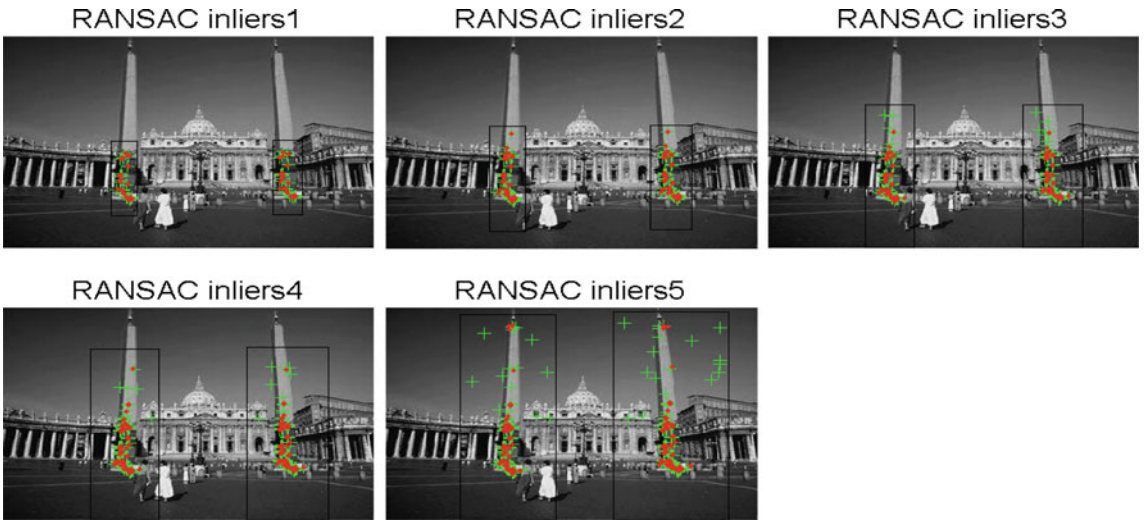


Fig. 9 Refining the affine transformation iteratively; the *green points* show the initial correspondences while the *red points* show the inliers found by RANSAC (color figure online)

Table 1 Number of correspondences and RANSAC inliers at each iteration

	Iterations						
	Initial step	1	2	3	4	5	6
Correspondences		27	28	36	50	71	77
RANSAC inliers	22	23	28	29	31	33	33

Given a pair of corresponding regions, first we define a detection window for each region using the inliers found by RANSAC (see Fig. 9). The detection windows are then slowly resized (i.e., horizontally and vertically). Then, keypoints are detected inside the resized windows and RANSAC is applied to find a new set of inliers. The new inliers are used to re-estimate the affine transformation parameters. Repeating these steps, the affine transformation parameters are refined iteratively until the number of inliers does not increase anymore. Figure 9 shows an example with five iterations. The number of correspondences and inliers at each iteration are shown in Table 1.

As it is evident from the example, the iterative process yields more correspondences, covering a larger area inside the original and duplicated regions; this yields a more accurate affine transformation. It should be mentioned that the threshold used for finding corresponding keypoints during the iterative process is greater than the one used in the initial step. This allows finding more correspondences compared to the initial stage.

3.4 Locating duplicated region

The last step of our algorithm attempts to accurately locate the duplicated region. Cross-correlation has been used before to locate the duplicated region and verify similarity with the original region [12]. In this study, we detect the duplicated region using dense MIFT features.

3.4.1 Dense MIFT feature extraction

To detect as many pixels as possible inside the duplicated region, we employ dense MIFT features. The key idea is computing a MIFT descriptor at each pixel location inside the detection window instead of at the keypoint locations only. This is on contrast to traditional methods employing pixel correlation for finding the duplicated region. Since MIFT descriptors can be matched more accurately than using pixel correlation, the duplicated region can be detected more precisely. Other dense feature descriptors, such as LBP or WLD, could be employed at this stage. Using the estimated

Fig. 10 Detection of original and duplicated regions using dense SIFT descriptors

affine transformation, the correspondences between the original and forged regions can be computed for each pixel location. The similarity between corresponding locations is then calculated using dense MIFT descriptors. Thresholding the distance between corresponding MIFT descriptors can then reveal the duplicated region. Figure 10 shows an example using this process.

3.4.2 Hysteresis thresholding

Using a single threshold to determine the similarity between corresponding MIFT descriptors in the original and duplicated regions might compromise detection results. In this work, we have opted for using hysteresis thresholding [37], a process based on two thresholds, one low and one high, which takes into consideration spatial information. Hysteresis thresholding has been used before in the context of edge detection [37]. The high threshold is used to detect "strong" edges while the low threshold is used to fill in gaps between "strong" edges using "weak" edges. The key idea is to include edge points whose strength exceeds the low threshold but are also adjacent to "strong" edge points. In a similar manner, we use the high threshold to detect "strong" corresponding pixels, that is, corresponding pixels from the original and duplicated region having very similar MIFT descriptors (i.e., very likely to belong to the duplicated region). Additional pixels (i.e., "weak" pixels") are detected if they are adjacent to "strong" pixels and the distance between the corresponding MIFT descriptors in the original and duplicated regions exceeds the low threshold. In our experiments, the low threshold is chosen to be R times lower than the high one, where R is a parameter.

3.4.3 Morphological operations

The output of the previous step is a group of pixels, which might still include holes or contain isolated pixels. To deal with these issues, we apply morphological operations (i.e., dilation and erosion) to remove small holes and eliminate isolated pixels. These operations are applied separately on the images obtained using the high and low thresholds described in the previous section. Then, we simply combine the results to obtain the final duplicated region.

4 Experimental results

In this section, the performance of the proposed approach is analyzed through a comprehensive set of experiments. For comparison purposes, we have also compared our method with the method of Pan and Lyu [12].

4.1 Dataset

To examine digital forgery detection methods, a dataset containing different types of forgery is required. In this study, we have used a realistic dataset, the CASIA tampered image detection evaluation database V2.0 (CASIA, 2010) [29]. CASIA v2.0 includes samples of copy–move and copy–paste digital forgeries applied in color images of different sizes, varying from 240×160 to 900×600. The tampered images have been generated by cutting-and-pasting image region(s). The image region(s) selected for duplication can be transformed before copying them by applying scaling, rotation, reflection, or distortion. The duplicated region can vary in size (e.g., small, medium, or large). The resulted image can be post-processed (e.g., by applying blurring) in order to create the final tampered image. Information about the type of transformations applied to generate a tampered image is encoded in its filename.

In this paper, we have only used images corresponding to copy–move forgery. Since the dataset includes both the original and forged images, we have applied pixel subtraction followed by binary thresholding and morphological closing to extract the duplicated region (i.e., ground truth) for evaluating the accuracy of our method. We have also added a new flag to identify images where the duplicated region has undergone mirror reflection. A sample of forged images and the ground truth indicating the forged area is shown in Fig. 11.

4.2 Implementation details

As mentioned earlier, the first step of our approach is to extract a set of keypoint descriptors. In this study, we extract MIFT features; the window centered at keypoints is defined to be 15×15 pixels. Since our aim in his step is to find quite accurate correspondences, we use threshold equal to 0.2 for comparing MIFT descriptors which gives less but more accurate matches. If the number of correspondences was less than 10, then we increase the threshold to 0.3 with step of 0.05. When removing incorrect matches using geometric constraints, we group corresponding points based on

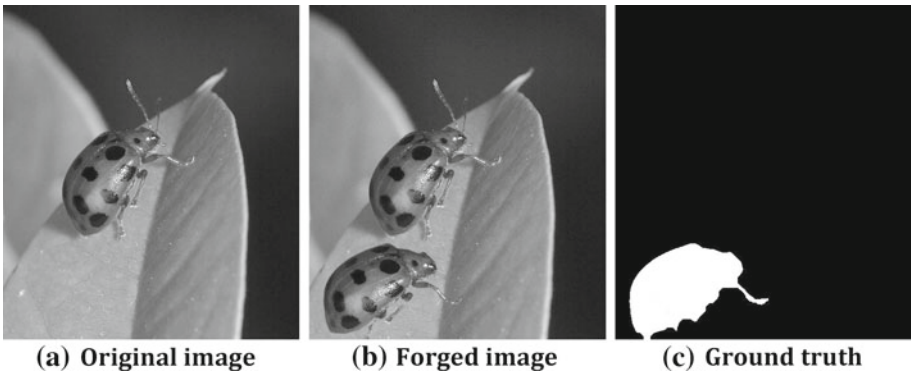


Fig. 11 A sample of images and the ground truth in the CASIA dataset

their slope in 10 groups. In addition, to refine the affine transformation, the search windows are resized with a rate of 0.2 (i.e., both horizontally and vertically) in each iteration. In this step, we match the MIFT descriptors using a threshold equal to 0.3 in order to allow more matches to be found. In hysteresis thresholding, the high threshold is defined to be $R = 2$ times smaller than the low one.[1]

To evaluate the performance of our method, we employ precision–recall (PR) curves [38]. Equations (1) and (2) show how the precision and recall rates are calculated; TP represents the number of true positives while FP represents the number of false positives. The number of pixels selected incorrectly as non-duplicated represents the number of false negatives (FN).

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \quad (1)$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (2)$$

4.3 Detailed results

As mentioned earlier, the CASIA dataset includes various images where operations have been applied on the copied region to generate the duplicated region. To better evaluate the performance of our method, we have classified images into different categories based on the size of the duplicated region and the operations used to create the forgery. Tables 2 and 3 show the different categories and the number of images within each category. We have evaluated our method on the colored entries of the tables since the other entries contain a very small number of images. For each experiment, we report the average PR curves over all the images of the dataset.

[1] Since in finding correspondences, a higher threshold yields a lower number of matches, we define the high and low values of hysteresis thresholding in opposite order compared to their definition in the literature.

Table 2 Image categories when duplicated region size is medium

Operations						Total
Translation	Blurring					0
Translation	Rotate					15
Translation	Rotate	Blurring				2
Translation	Scale					30
Translation	Scale	Blurring				5
Translation	Scale	Rotate				17
Translation	Scale	Rotate	Blurring			3
Translation	Deform					3
Translation	Deform	Blurring				3
Translation	Rotate	Deform				2
Translation	Rotate	Deform	Blurring			0
Translation	Scale	Deform				2
Translation	Scale	Deform	Blurring			1
Translation	Scale	Rotate	Deform			3
Translation	Scale	Rotate	Deform	Blurring		0
Translation	Reflection					0
Translation	Reflection	Blurring				0
Translation	Reflection	Rotate				169
Translation	Reflection	Rotate	Blurring			6
Translation	Reflection	Scale				9
Translation	Reflection	Scale	Blurring			0
Translation	Reflection	Scale	Rotate			48
Translation	Reflection	Scale	Rotate	Blurring		4
Translation	Reflection	Deform				1
Translation	Reflection	Deform	Blurring			0
Translation	Reflection	Rotate	Deform			2
Translation	Reflection	Rotate	Deform	Blurring		1
Translation	Reflection	Scale	Deform			0
Translation	Reflection	Scale	Deform	Blurring		0
Translation	Reflection	Scale	Rotate	Deform		4
Translation	Reflection	Scale	Rotate	Deform	Blurring	1
Sum						331

4.3.1 Effect of thresholding

First, we compare standard thresholding with hysteresis thresholding. Since the output of thresholding is a group of pixels that might contain holes or isolated pixels, we apply morphological operations, as mentioned earlier, to

Table 3 Image categories when duplicated region size is small

Operations						Total
Translation	Blurring					0
Translation	Rotate					168
Translation	Rotate	Blurring				22
Translation	Scale					336
Translation	Scale	Blurring				60
Translation	Scale	Rotate				87
Translation	Scale	Rotate	Blurring			8
Translation	Deform					48
Translation	Deform	Blurring				8
Translation	Rotate	Deform				9
Translation	Rotate	Deform	Blurring			3
Translation	Scale	Deform				48
Translation	Scale	Deform	Blurring			33
Translation	Scale	Rotate	Deform			19
Translation	Scale	Rotate	Deform	Blurring		4
Translation	Reflection					0
Translation	Reflection	Blurring				0
Translation	Reflection	Rotate				50
Translation	Reflection	Rotate	Blurring			6
Translation	Reflection	Scale				12
Translation	Reflection	Scale	Blurring			3
Translation	Reflection	Scale	Rotate			35
Translation	Reflection	Scale	Rotate	Blurring		2
Translation	Reflection	Deform				66
Translation	Reflection	Deform	Blurring			16
Translation	Reflection	Rotate	Deform			12
Translation	Reflection	Rotate	Deform	Blurring		2
Translation	Reflection	Scale	Deform			20
Translation	Reflection	Scale	Deform	Blurring		11
Translation	Reflection	Scale	Rotate	Deform		6
Translation	Reflection	Scale	Rotate	Deform	Blurring	4
Sum						1, 098

reduce false positives and negatives. Hysteresis thresholding includes a low and a high threshold which are applied to threshold the distance between MIFT features; the results are then combined to make the final region. The morphological operations are applied prior to combining the results of the high and low thresholds. Figure 12 shows two examples comparing standard with hysteresis thresholding. The duplicated regions have been produced using scaling in the top image and reflection in the bottom image. Figure 13 shows the corresponding PR curves. Clearly, hysteresis thresholding can locate the duplicated region more accurately. Additional experiments are reported in Sect. 4.5.3.

4.3.2 Effect of scale and rotation

In this set of experiments, we have evaluated the proposed approach assuming that the duplicated regions have been created using rotation and/or scale transformations. We have considered both medium and small size duplicated regions.

Scale or rotation Scale and rotation represent the simplest operations for creating duplicated regions. Figure 14 shows some examples; as it can be observed, the duplicated regions have been located quite accurately. Figures 15 and 16 show the corresponding average PR curves; as the results indicate, the proposed method performs considerably better than the method of [12].

Scale–rotation In this set of experiments, we consider the case where both scale and rotation have been applied to create the image forgery. Figure 17 shows an example along with detection results for our method and the method of [12]. Figure 18 shows the corresponding average PR curves; as the

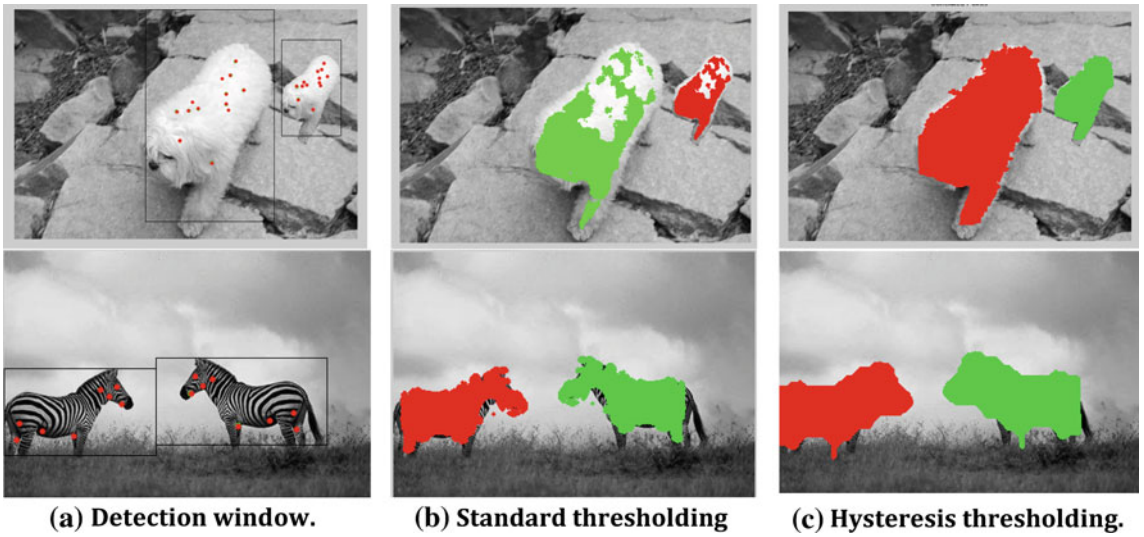


Fig. 12 Comparison between standard and hysteresis thresholding

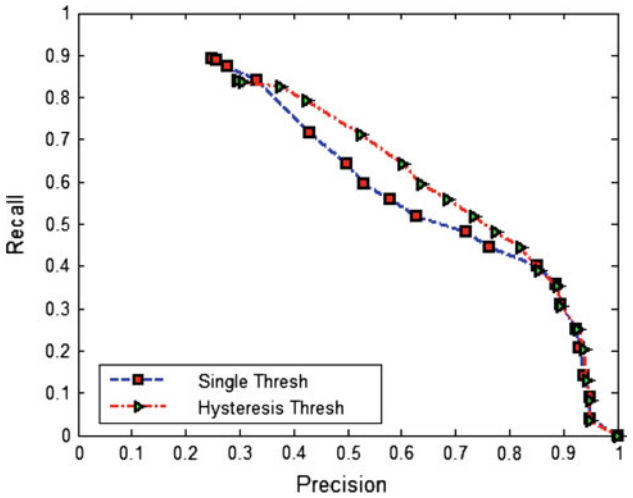


Fig. 13 Comparison between standard (single) and hysteresis thresholding

results indicate, the proposed method performs considerably better than the method of [12], especially when the size of the duplicated region is small.

4.3.3 Effect of reflection

As described earlier, mirror reflection is a common operation used in copy–move image forgery. MIFT is robust to the reflection. However, SIFT which was used in [12] is not robust to reflection. To make SIFT robust to reflection, the feature vector of each keypoint is translated horizontally and vertically before finding the similarities among the vectors. The accuracy of the proposed method and the method of [12] are examined in this set of experiments assuming medium and small duplicated region sizes as well as mirror reflection, rotation and scale. Our results indicate that the accuracy of proposed method is noticeably better than the method of [12], especially when the duplication region size is small.

Reflection–scale and reflection–rotation In the first set of experiments, we considered the case of mirror reflection and scale in creating the image forgery, assuming both medium and small duplication region sizes. In the second set of experiments, we considered the case of mirror reflection and rotation, assuming both medium and small duplication region sizes. Some example images in these two categories are shown in Fig. 19.

The method of [12] uses SIFT features, which are not robust to mirror reflection. As described earlier, to make the SIFT algorithm robust to mirror reflection, we find SIFT correspondences both in the original image as well as in its mirror reflected image (i.e., obtained by filliping the original image horizontally and vertically). The accuracy of the proposed method and the method of [12] are shown in Figs. 20 and 21. For medium size regions, the two methods have similar performance; however, the proposed method outperforms the method of [12] for large size regions.

Reflection–scale–rotation Combining mirror reflection with scale and rotation to create the duplicated region is investigated next. Figure 22 shows an example along with detection results. The accuracy of proposed method and the method of [12] are compared in Fig. 23. The proposed method outperforms the method of [12], especially when the size of the duplicated region is small. When region size is medium, both methods perform about the same.

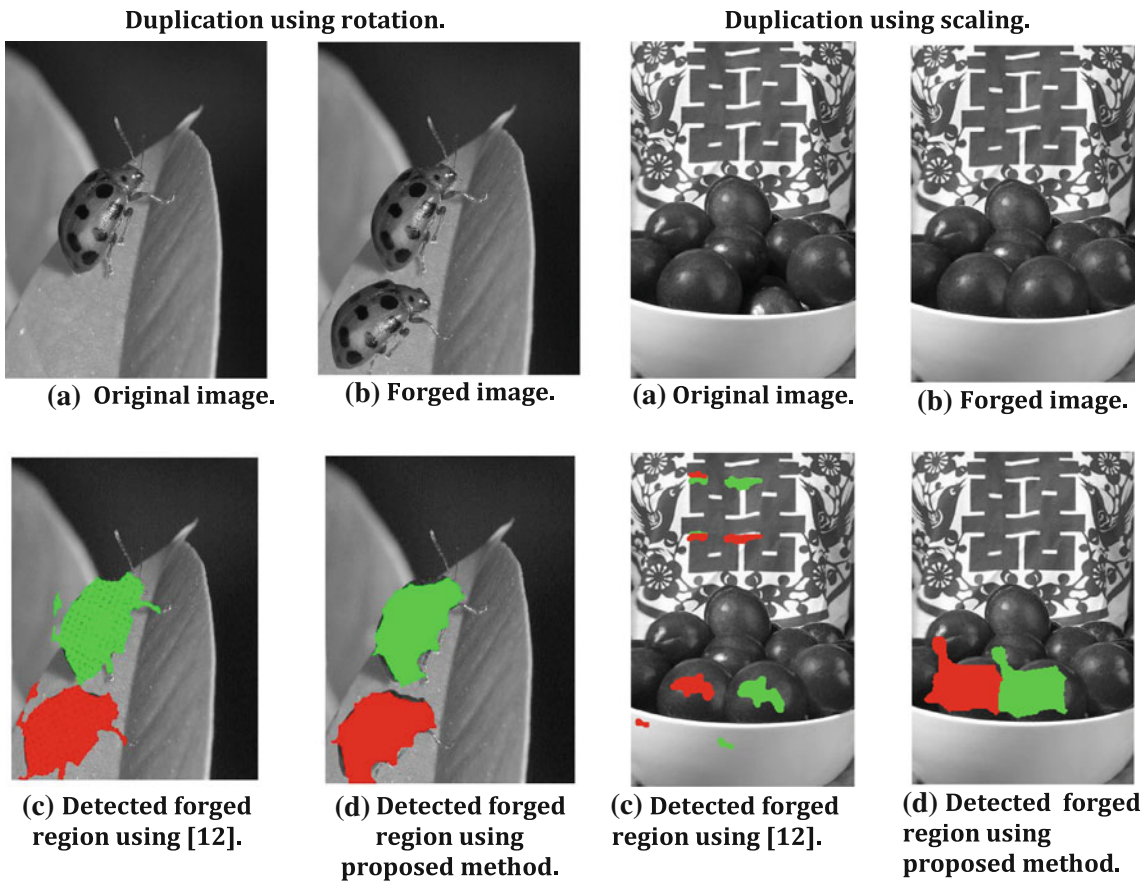


Fig. 14 Detection of image forgery assuming scale or rotation

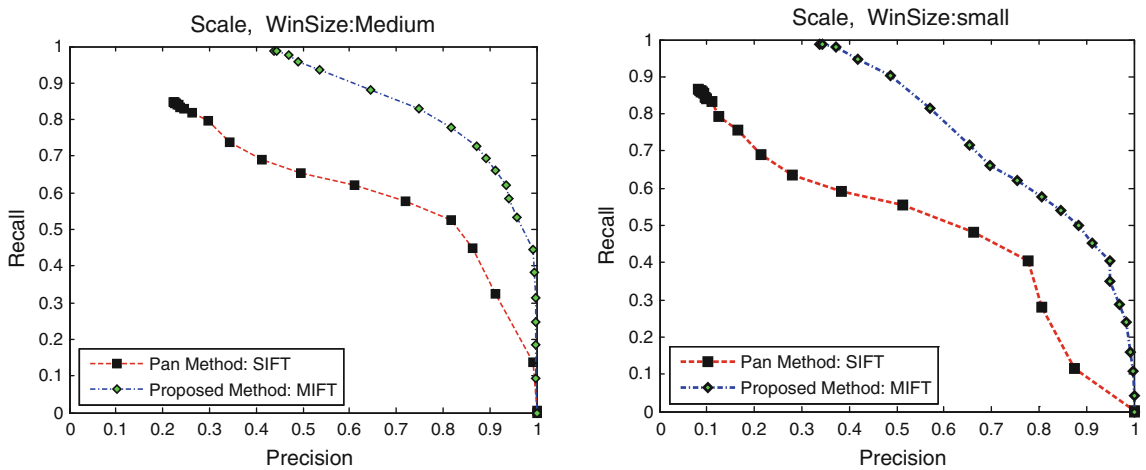


Fig. 15 Comparison between the proposed method and the method of [12] assuming scale differences (average PR curves)

4.3.4 Effect of blurring

Retouching the duplicated region is a common operation for removing inconsistencies and hiding the forgery. To achieve this goal, blurring is used often. In the CASIA dataset, blurring has been applied either on the edges of the duplicated region or on the whole region. This operation is typically combined with other operations such as scale and rotation.

Blurring–scale and blurring–rotation In this section, we consider combining blurring with scale or with rotation for creating the image forgery. We only consider the case

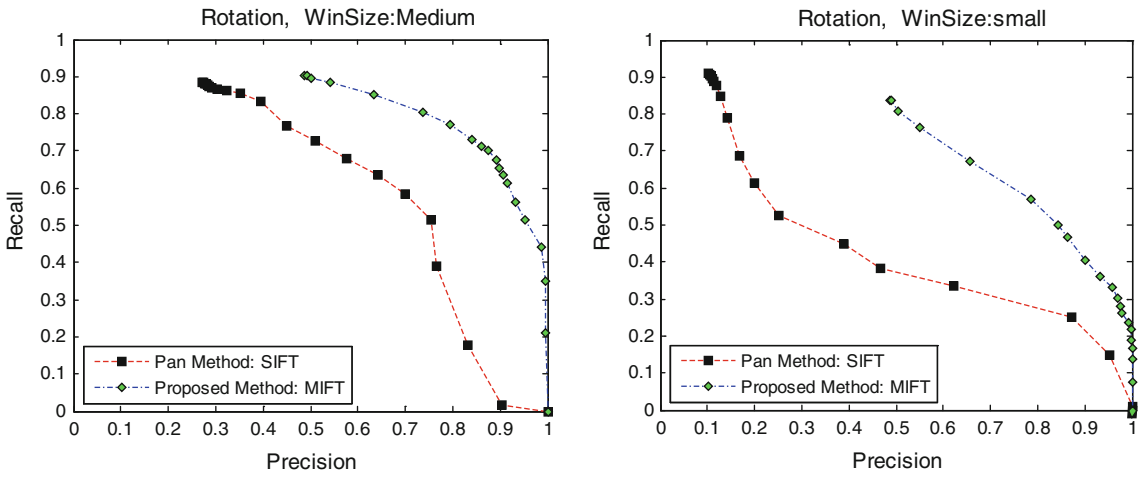
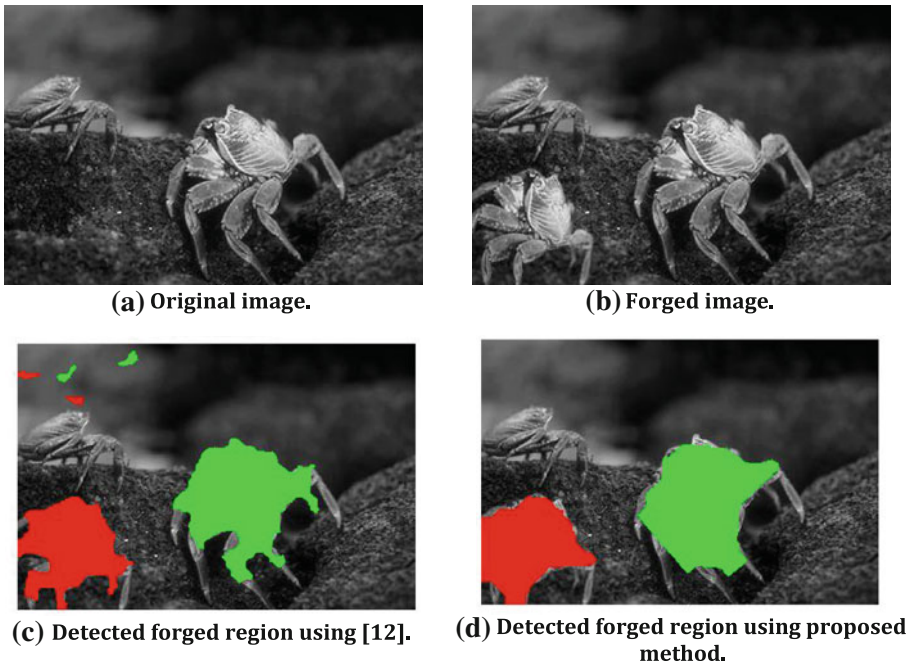


Fig. 16 Comparison between the proposed method and the method of [12] assuming rotation changes (average PR curves)

Fig. 17 Detection of image forgery assuming both scale and rotation



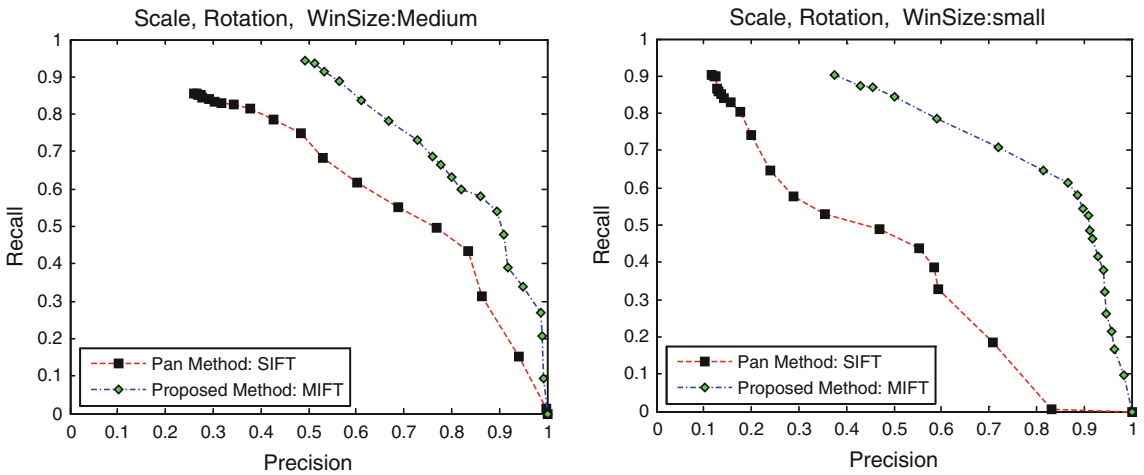


Fig. 18 Comparison between the proposed method and the method of [12] assuming scale and rotation changes (average PR curves)

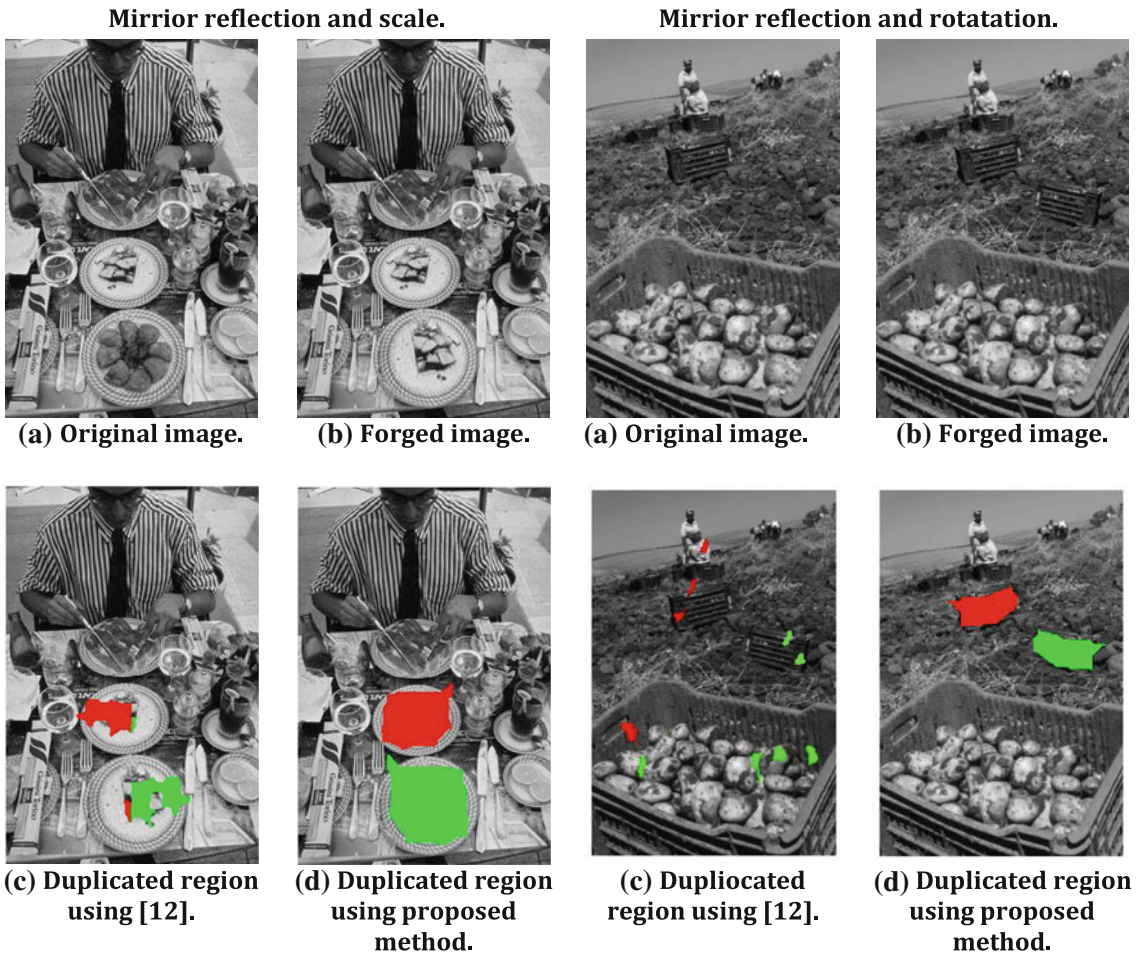


Fig. 19 Detection of image forgery assuming mirror reflection and scale or mirror reflection and rotation

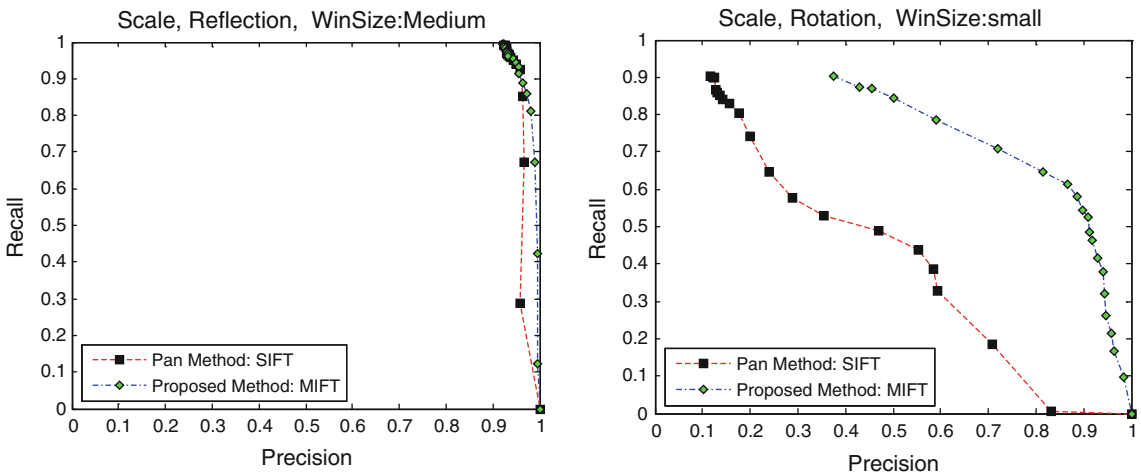


Fig. 20 Comparison between the proposed method and the method of [12] assuming mirror reflection and scale changes (average PR curves)

of small duplicated regions since there are not enough images with medium size duplicated regions for these cases in the CASIA dataset. Figure 24 shows some examples along with detections of duplicated regions. Figures 25 and 26 compare the proposed approach with the method of [12].

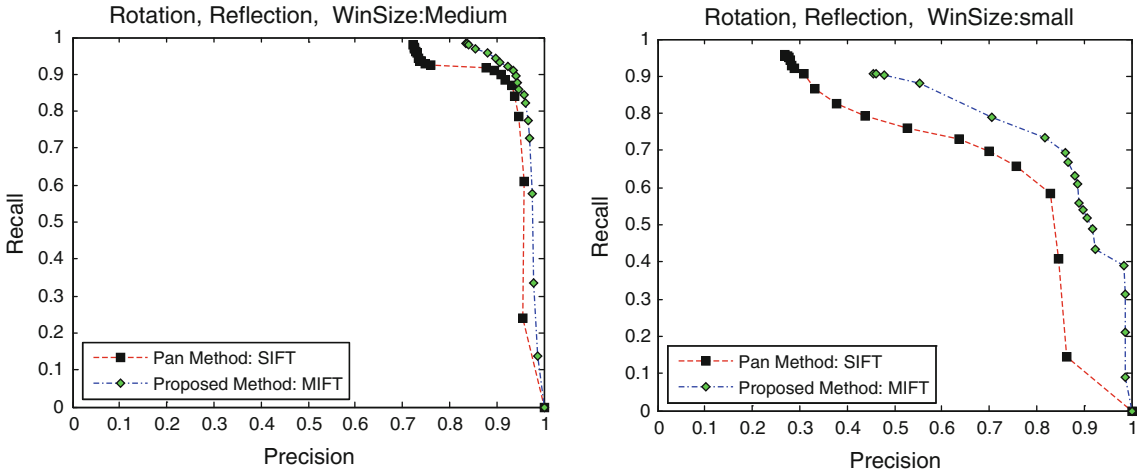


Fig. 21 Comparison between the proposed method and the method of [12] assuming mirror reflection and rotation changes (average PR curves)

Fig. 22 Detection of image forgery assuming mirror reflection, scale, and rotation

(a) Original image.

(b) Forged image.

(c) Duplicated region using [12].

(d) Duplicated region using proposed method.

Blurring–scale–rotation In this set of experiments, blurring, scale, and rotation are combined to create the image forgery. Figure 27 shows an example along with detection of the duplicated region. The accuracy of proposed method and the method presented in [12] are compared in Fig. 28. This comparison was done using small duplicated region sizes only.

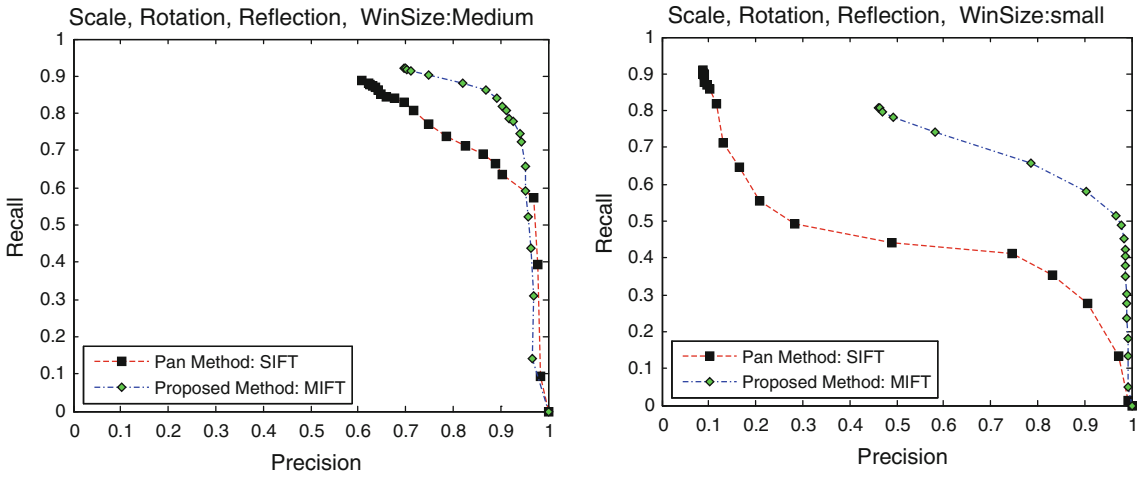


Fig. 23 Comparison between the proposed method and the method of [12] assuming mirror reflection, scale, and rotation changes (average PR curves)

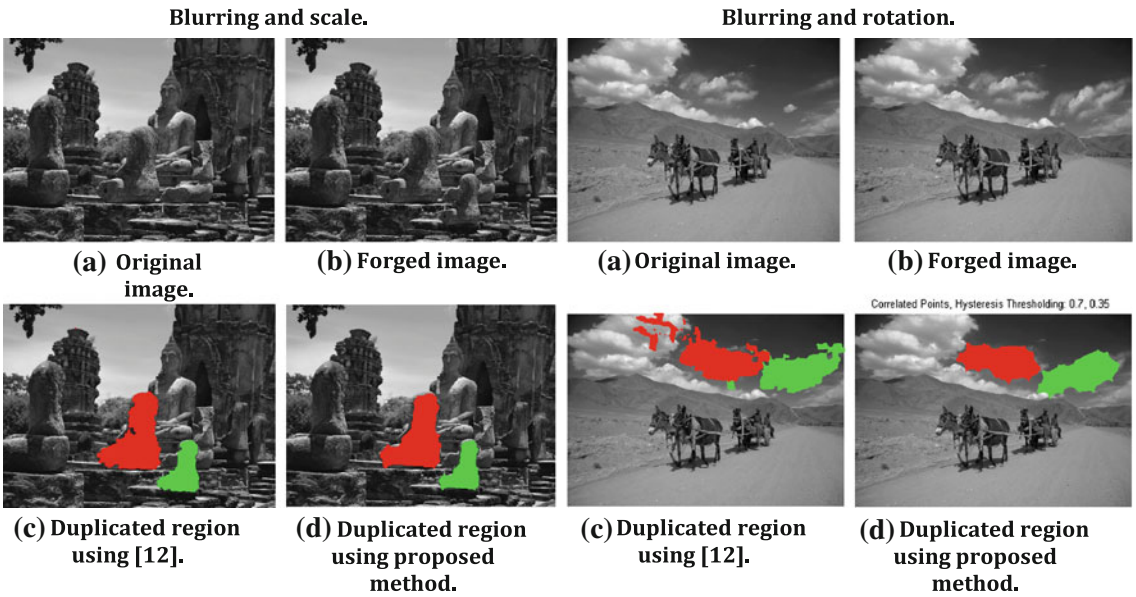


Fig. 24 Detection of image forgery assuming blurring, scale, or rotation

4.3.5 Effect of deformation

Deformation is another operation applied on the images of the CASIA dataset. This operation is typically a non-linear transformation. As shown below, detecting this kind of forgery has lower accuracy than forgery detection in other categories. This is due to the fact that we employ a linear transformation (e.g., affine) to bring similar regions into correspondence. Nevertheless, the proposed method still outperforms the method of [12].

Deformation In this set of experiments, image forgery has been created using deformation only. Figure 29 shows an example along with duplicated region detection results. Figure 30 compares the proposed method with the method of [12]. This comparison was done using small duplicated region sizes only.

Deformation–scale and deformation–rotation In this category of experiments, image forgery was created using deformation and scale as well deformation and rotation. Figure 31 shows an example of deformation and scale along with duplicated region detection results. Figure 33 shows an example of deformation and rotation along with duplicated region detection results. Figures 32 and 34 compare the proposed method with the method of [12]. These comparisons were done using small duplicated region sizes only.

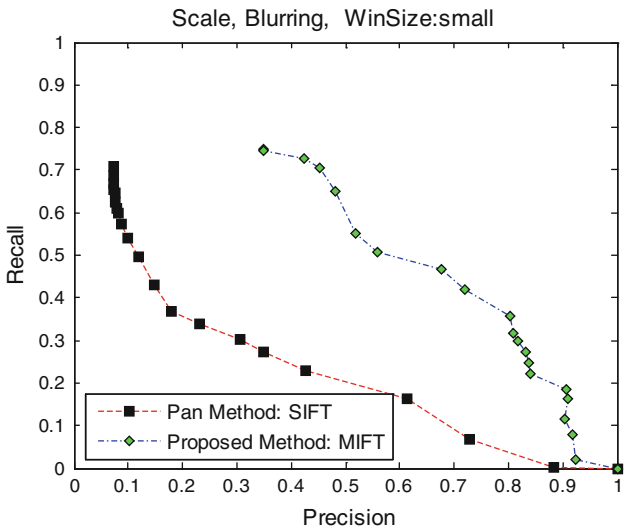


Fig. 25 Comparison between the proposed method and the method of [12] assuming blurring and scale changes (average PR curves)

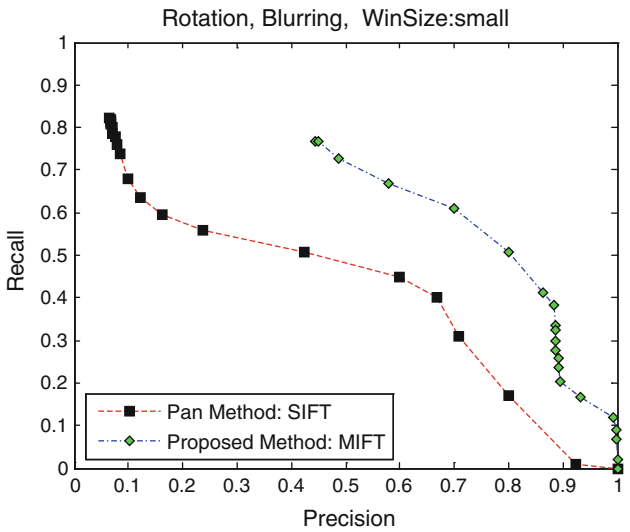


Fig. 26 Comparison between the proposed method and the method of [12] assuming blurring and rotation changes (average PR curves)

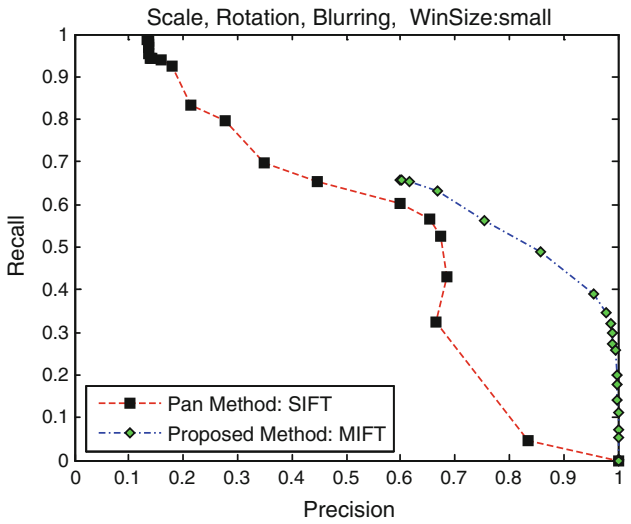
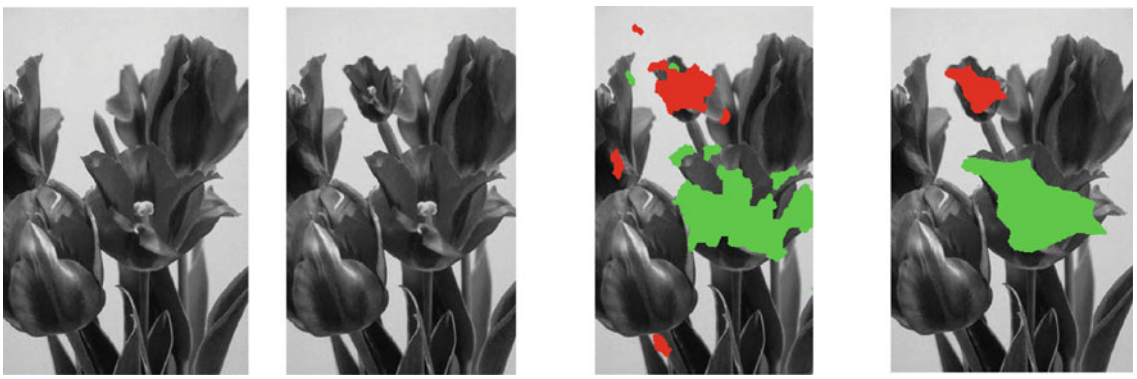


Fig. 28 Comparison between the proposed method and the method of [12] assuming blurring, scale, and rotation changes (average PR curves)

Deformation–scale–rotation In this set of experiments, we considered deformation, scale, and rotation for image forgery. Figure 35 shows an example along with duplicated region detection results. Figure 36 compares the proposed method with the method of [12]. Our method outperforms the method of [12], however, extracting the duplicated region has a lower accuracy overall when combining all three transformations together. This comparison was done using small duplicated region sizes only.

4.3.6 Other combinations

In addition to the combinations investigated above, there are a few more combinations worth of investigating in the dataset. These combinations which include a reasonable number of images are investigated next.



(a) Original image **(b) Forged image** **(c) Duplicated region using [12].** **(d) Duplicated region using proposed method.**

Fig. 27 Detection of image forgery assuming blurring, scale, and rotation

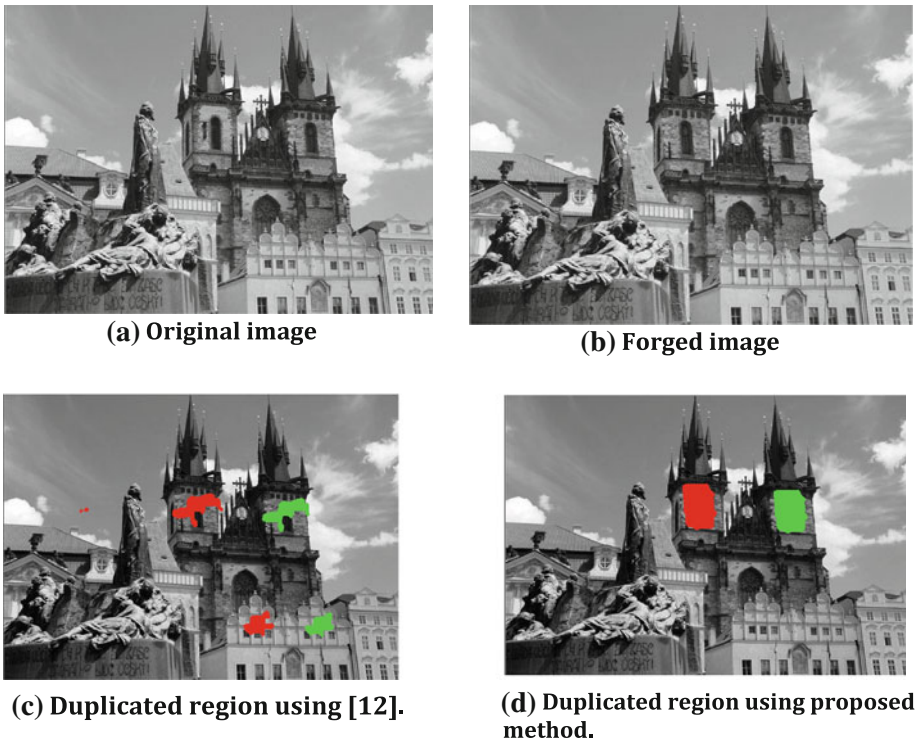


Fig. 29 Detection of image forgery assuming deformation

(a) Original image (b) Forged image

(c) Duplicated region using [12]. (d) Duplicated region using proposed method.

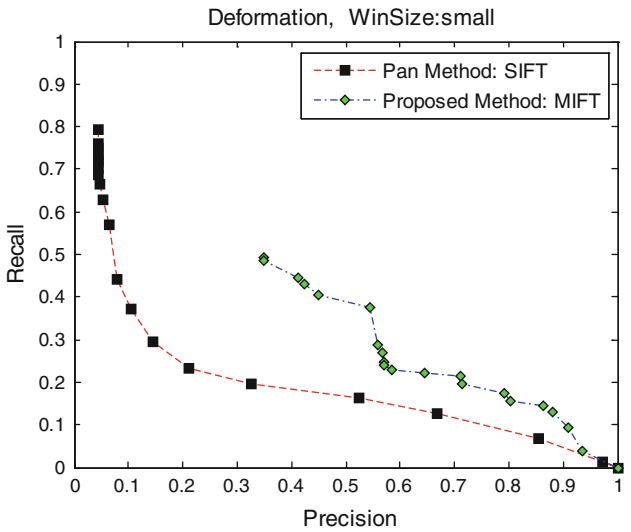
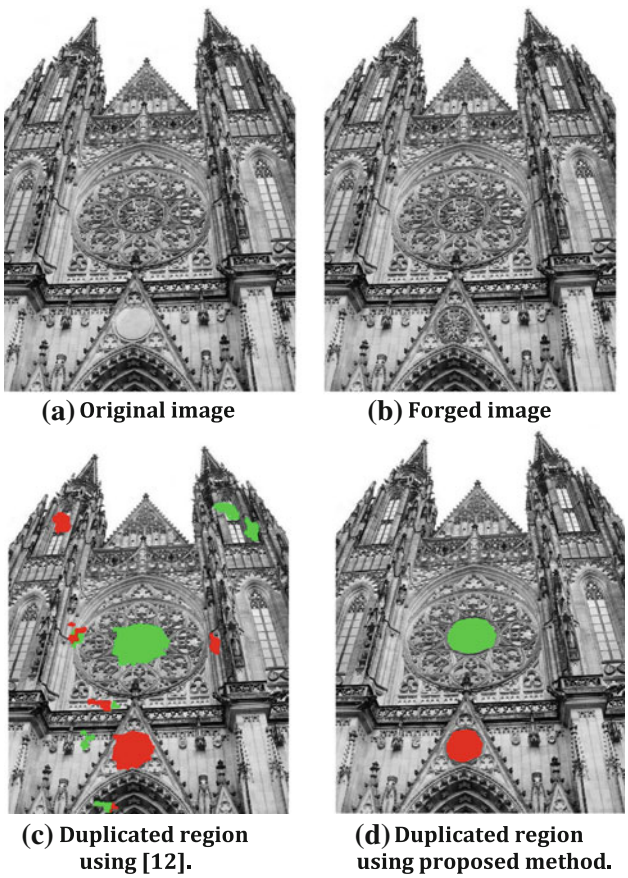


Fig. 30 Comparison between the proposed method and the method of [12] assuming deformation changes (average PR curves)

Deformation–reflection–scale In this set of experiments, we consider applying deformation, reflection, and scale together for image forgery. Figure 37 shows an example along with duplicated region detection results. Figure 38 compares the proposed method with the method of [12]. Our method outperforms the method of [12]. This comparison was done using small duplicated region sizes only.

Deformation–reflection–rotation In this set of experiments, we consider deformation, reflection, and rotation for image



(a) Original image (b) Forged image

(c) Duplicated region using [12]. (d) Duplicated region using proposed method.

Fig. 31 Detection of image forgery assuming deformation and scale

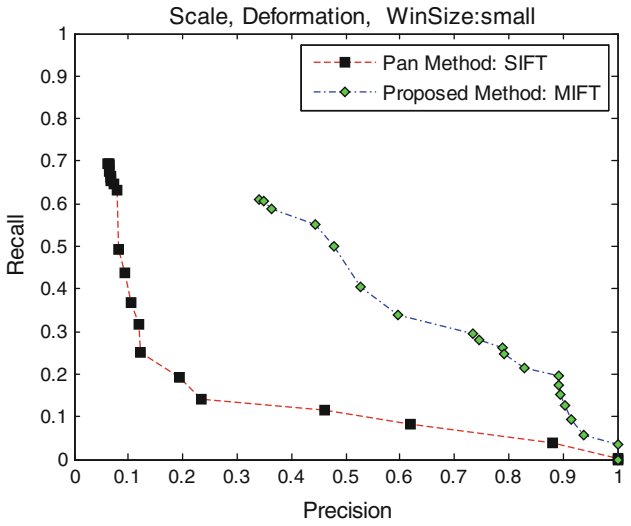


Fig. 32 Comparison between the proposed method and the method of [12] assuming deformation and scale changes (average PR curves)

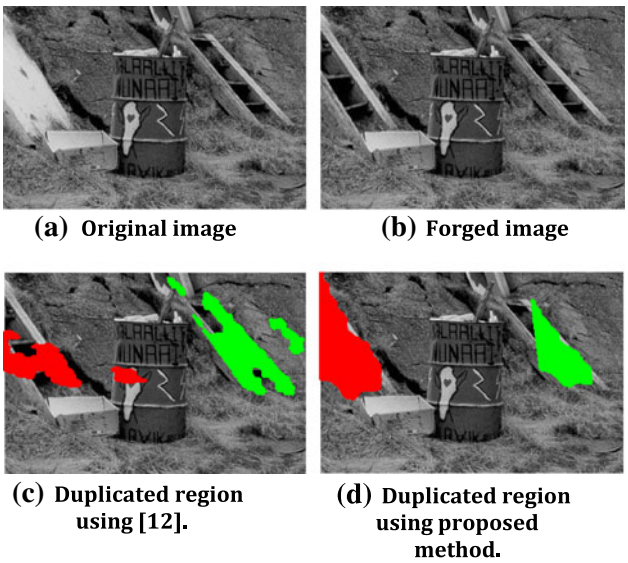


Fig. 33 Detection of image forgery assuming deformation and rotation

forgery. Figure 39 shows an example along with duplicated region detection results. Figure 40 compares the proposed method with the method of [12]. This comparison was done using small duplicated region sizes only.

Deformation–reflection–blurring This category of images combines the deformation, reflection, and blurring transformations to make the forged images. Figure 41 shows an example along with duplicated region detection results. Figure 42 compares the proposed method with the method of [12]. This comparison was done using small duplicated region sizes only.

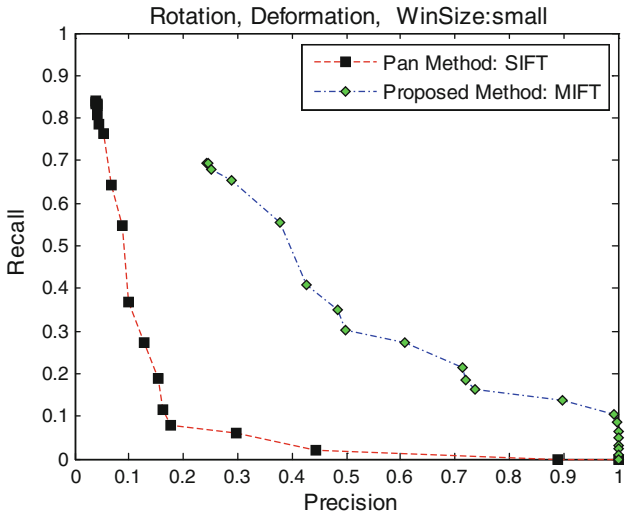


Fig. 34 Comparison between the proposed method and the method of [12] assuming deformation and rotation changes (average PR curves)

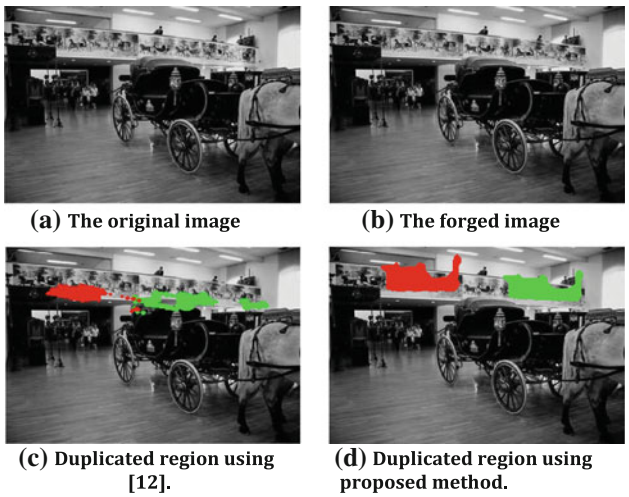


Fig. 35 Detection of image forgery assuming deformation, scale, and rotation

4.4 F-measure

When computing recall and precision rates, it might be useful to define a relative cost α between these quantities, which focuses attention to a specific point on the PR curve. The F-measure [40], defined below, captures this tradeoff as the weighted harmonic mean of P and R.

$$F = PR/(aR + (1 - a)P)$$

The location of the maximum F-measure along the PR curve provides the optimal detector threshold for the application given α. The higher the F-measure is, the better the method performs. Table 4 shows the F-measure for the experiments reported in the previous section, assuming $\alpha = 0.5$. As it can be observed, the proposed method yields a higher F-measure than the method of Pan [12] in all cases.

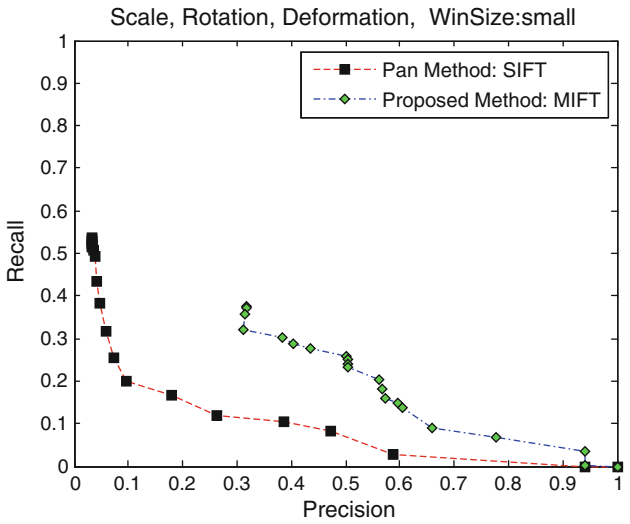


Fig. 36 Comparison between the proposed method and the method of [12] assuming deformation, scale, and rotation changes (average PR curves)

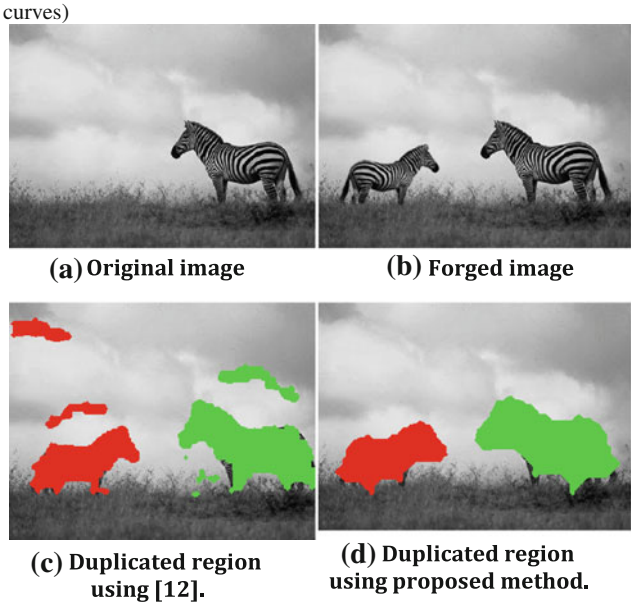


Fig. 37 Detection of image forgery assuming deformation, reflection, and scale

4.5 Effect of different algorithmic steps

In this section, we investigate the effect on performance of different algorithmic steps of the proposed method. In particular, we investigate the effect of MIFT features, iterative affine refinements, and hysteresis thresholding. In these experiments, we have used the 'Translation Reflection, Scale, Rotate' category assuming both medium and small size of forged regions.

4.5.1 Effect of MIFT features

As mentioned earlier, to resolve the issue of mirror reflection and find correspondences in duplicated regions, pervious methods [12] used SIFT descriptors extracted both from horizontally and vertically reflected versions of the original image. However, MIFT descriptors are invariant to mirror reflection transformations. To evaluate the effect of using MIFT features, we have performed experiments comparing MIFT with SIFT features assuming that all other steps are

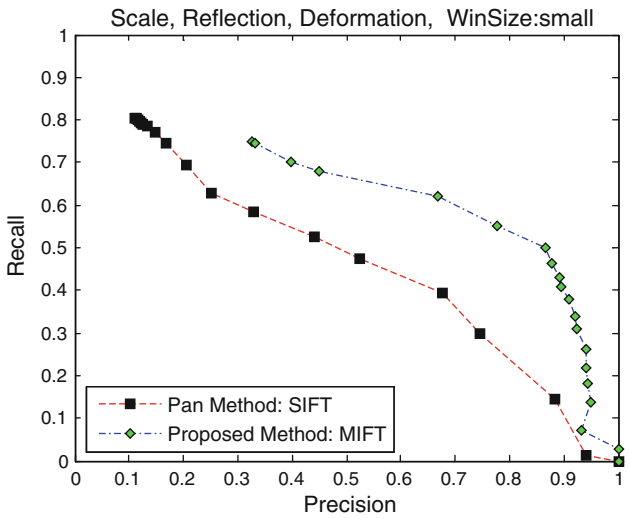


Fig. 38 Comparison between the proposed method and the method of [12] assuming deformation, reflection, and scale changes (average PR curves)

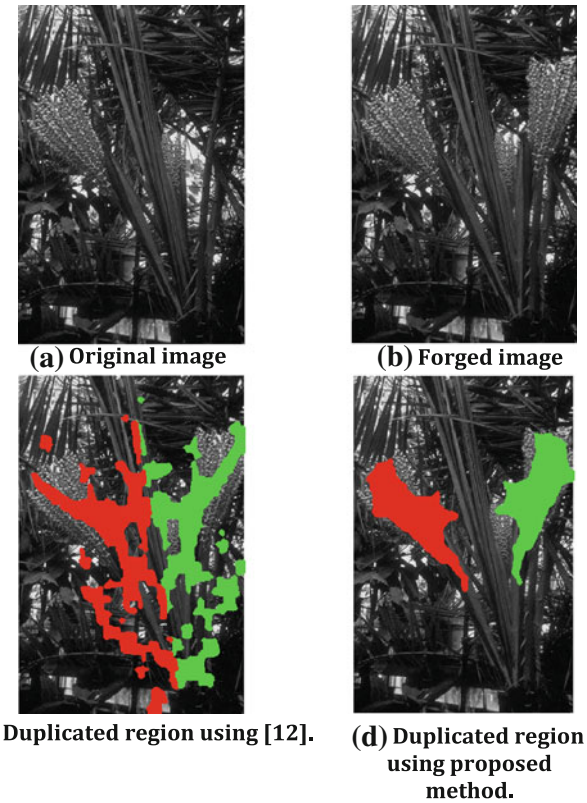


Fig. 39 Detection of image forgery assuming deformation, reflection, and rotation

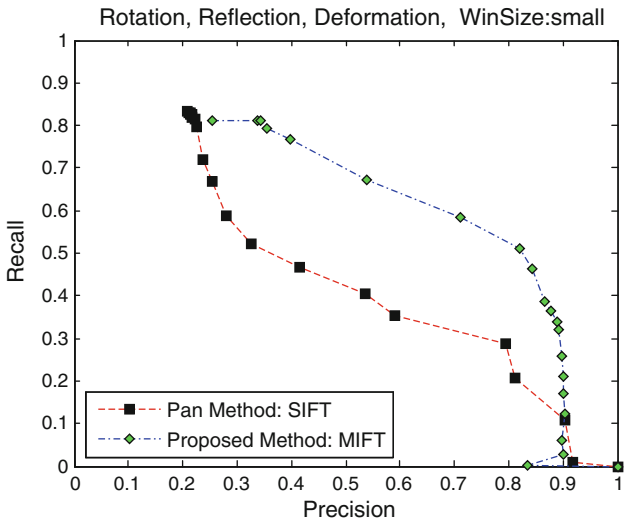


Fig. 40 Comparison between the proposed method and the method of [12] assuming deformation, reflection, and rotation changes (average PR curves)

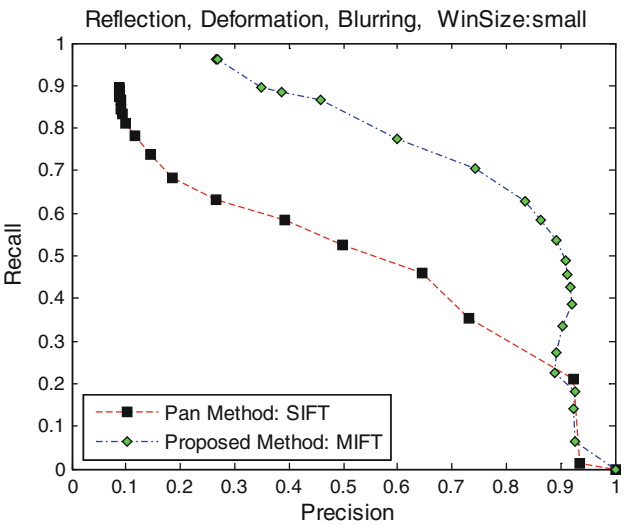


Fig. 42 Comparison between the proposed method and the method of [12] assuming deformation, reflection, and blurring changes (average PR curves)

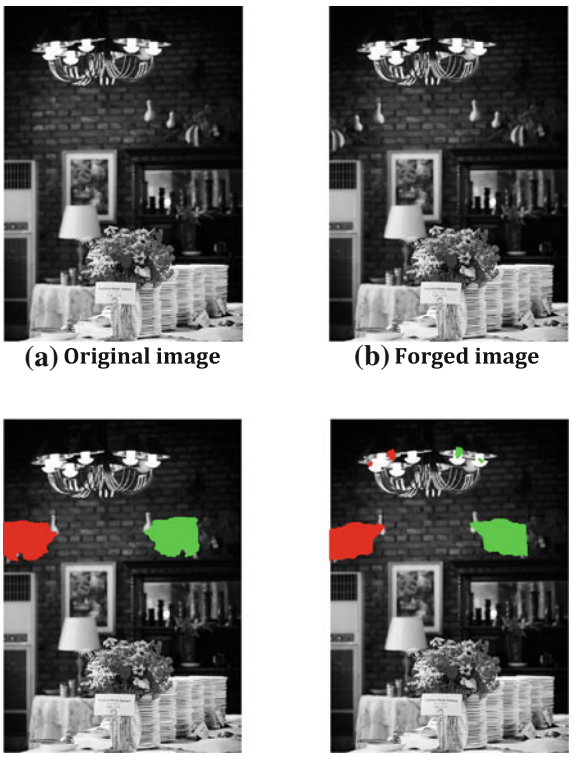


Fig. 41 Detection of image forgery assuming deformation, reflection, and blurring

the same. Figure 43 shows the average PR curves for each case. As it can be observed, the performance using MIFT features does not degrade performance; on the contrary, there is a slight performance improvement using MIFT features.

Moreover, Table 5 shows that MIFT features have lower average time requirements compared to SIFT features.

4.5.2 Effect of iterative affine refinements

As discussed in Sect. 3.3, we iteratively refine the parameters of the affine transformation which is estimated using MIFT matches and RANSAC for removing outliers. In addition to refining the affine transformation parameters, we also determine the size of the search window both for the copy and moved regions. In this experiment, we examine the effect of refining the affine transformation parameters. For this, we compare the performance of the method with and without using affine refinements. Figure 44 shows the average PR curves for each case. As it can be observed, affine refinements improve performance in the case of small-size regions.

4.5.3 Effect of hysteresis thresholding

We have already illustrated the effect of hysteresis thresholding in Sect. 4.3.1. For completeness purposes, we have also tested the effect of hysteresis thresholding on the 'Translation Reflection, Scale, Rotate' category. Figure 45 shows the average PR curves; clearly, hysteresis thresholding can locate the duplicated region more accurately than using single thresholding.

5 Conclusions

In this paper, we have considered the problem of copy–move image forgery detection. Our emphasis was on detecting

Table 4 F-measure for the different datasets used in our experiments

	Transformations			Datasets			
				Proposed method		Pan's method	
				Medium size	Small size	Medium size	Small size
Effect of scale and rotation	Scale			0.7969	0.6834	0.6412	0.5572
	Rotation			0.7831	0.6634	0.6384	0.4369
	Scale	Rotation		0.7299	0.7217	0.6115	0.4893
Effect of reflection	Scale	Reflection		0.9576	0.8127	0.9569	0.5987
	Rotation	Reflection		0.9063	0.7735	0.8792	0.7042
	Scale	Rotation	Reflection	0.8766	0.7206	0.7699	0.5293
Effect of blurring	Scale	Blurring		–	0.5534	–	0.3061
	Rotation	Blurring		–	0.6522	–	0.5139
	Scale	Rotation	Blurring	–	0.6494	–	0.3597
Effect of deformation	Deformation			–	0.4451	–	0.2506
	Scale	Deformation		–	0.4917	–	0.1937
	Rotation	Deformation		–	0.4490	–	0.1747
	Scale	Rotation	Deformation	–	0.3436	–	0.1728
Other combinations	Deformation	Blurring		–	0.4009	–	0.1957
	Scale	Deformation	Blurring	–	0.3433	–	0.1421
	Reflection	Deformation		–	0.6022	–	0.4589
	Reflection	Deformation	Blurring	–	0.7060	–	0.5115
	Rotation	Reflection	Deformation	–	0.6407	–	0.4622
	Scale	Reflection	Deformation	–	0.6458	–	0.4977

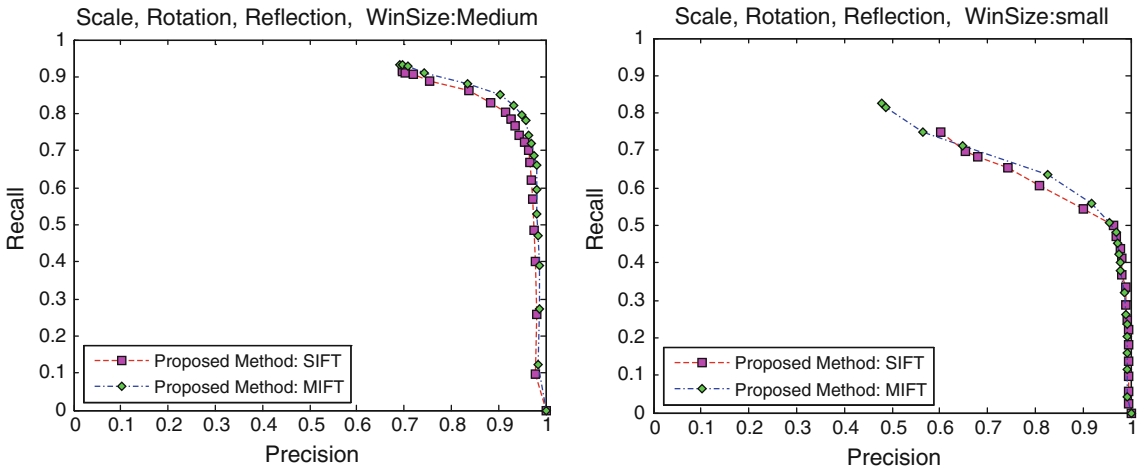


Fig. 43 Comparison between MIFT and SIFT features to resolve mirror reflection (average PR curves)

and extracting duplicated regions with higher accuracy and robustness. The proposed methodology employs a new set of keypoint-based features, called MIFT, for finding similar regions in an image. To estimate the affine transformation between similar regions more accurately, we have proposed an iterative scheme which refines the affine transformation parameter by finding more keypoint matches incrementally. To reduce false positives and negatives when extracting the duplicated region, we have proposed using dense MIFT features in conjunction with hysteresis thresholding and morphological operations. We have performed comprehensive experiments using a large dataset of real images to evaluate the proposed approach. In particular, we have investigated the effect of different transformations in creating the image forgery on detection accuracy. Among all transformation considered, blurring and deformation affect detection results most. Obviously, blurring affects the accuracy of matching keypoint-based features while deformation cannot be modeled well by the affine transformation model being used here for bringing similar regions into correspondence.

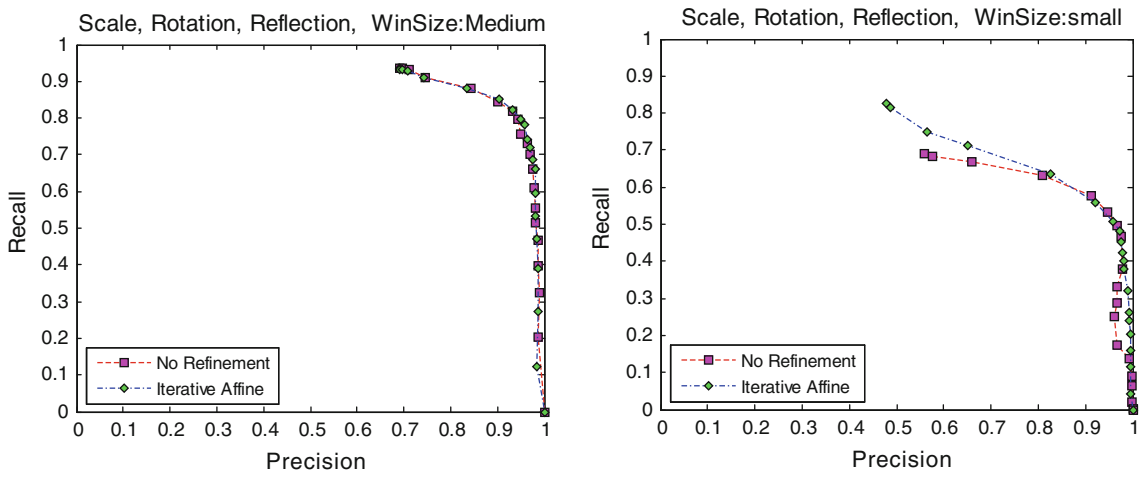


Fig. 44 Comparison between using and not using affine refinements (average PR curves)

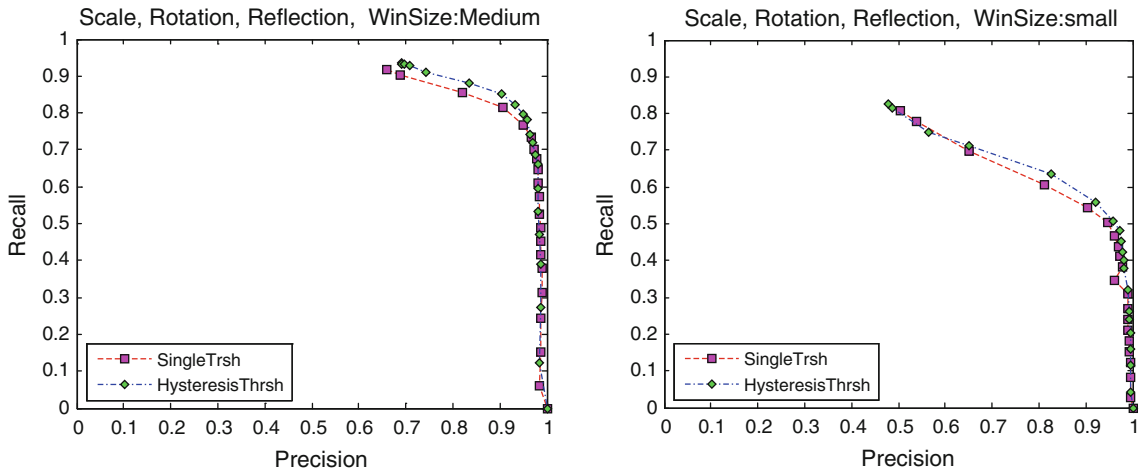


Fig. 45 Comparison between using single and hysteresis thresholding (average PR curves)

Table 5 Time requirements using MIFT and SIFT features to resolve mirror reflection

Feature type	Time (s)	
	Medium size	Small size
SIFT	1.8252	3.0387
MIFT	1.6312	2.6938

Comparisons with competitive methods indicate that the proposed methodology can extract duplicated regions more accurately. It should be mentioned that like similar methods employing keypoint-based features for matching, the proposed approach will not work well if the duplicated region corresponds to a flat surface where no interest points can be detected. For future work, we plan to investigate different types of dense feature descriptors, such as LBP or WLD, for improving detection results.

Acknowledgments This work is supported by grant 10-INF1140-02 under the National Plan for Science and Technology (NPST), King Saud University, Riyadh, Saudi Arabia. George Bebis is a Visiting Professor in the Department of Computer Science at King Saud University, Saudi Arabia.

References

1. Luo, W., Huang, J., Qiu, G.: Robust detection of region-duplication forgery in digital images. In: Proceedings of International Conference on Pattern Recognition, pp. 746–749. Washington, D.C. (2006)
2. Farid, H.: A survey of image forgery detection. IEEE Signal Process. Mag. **2**(26), 16–25 (2009)
3. Zhang, J., Feng, Z., Su, Y.: A new approach for detecting copy-move forgery in digital images. In: IEEE Singapore International Conference on Communication Systems, pp. 362–366. (2008)
4. Fridrich, J., Soukal, D., Lukas, J.: Detection of Copy-Move Forgery in Digital Images. Department of Electrical and Computer Engineering, Department of Computer Science SUNY Binghamton, Binghamton, NY(2003)

5. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting duplicated image regions. In: Technical Report TR2004-515, Dartmouth College, August (2004)
6. Lin, Z., Wang, R., Tang, X., Shum, H.-V.: Detecting doctored images using camera response normality and consistency. In: Proceedings of Computer Vision and Pattern Recognition. San Diego, CA, (2005)
7. Mahdian, B., Saic, S.: Detection of copy move forgery using a method based on blur moment invariants. Forensic Sci. Int. **171**, 180–189 (2007)
8. Li, G., Wu, Q., Tu, D., Sun, S.: A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD. In: IEEE International Conference on Multimedia and Expo, pp. 1750–1753. Beijing, China (2007)
9. Khan, S., Kulkarni, A.: An efficient method for detection of copy-move forgery using discrete wavelet transform. Int. J. Comput. Sci. Eng. (IJCSE) **2**(5), 1801–1806 (2010)
10. Muhammad, G., Hussain, M., Khawaji, K., Bebis, G.: Blind copy move image forgery detection using dyadic uncedimated wavelet transfor. In: 17th International Conference on Digital Signal Processing. Corfu, Greece, July (2011)
11. Huang, H., Guo, W., Zhang, Y.: Detection of copy-move forgery in digital images using SIFT algorithm. In: IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (2008)
12. Pan, X., Lyu, S.: Region duplication detection using image feature matching. IEEE Trans. Inf. Forensics Secur. **5**(4), 857–867 (2010)
13. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G.: A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans. Inf. Forensics Secur. **6**(3), 1099–1110 (Jun. 2011)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
15. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proceedings of Eighth International Conference on Computer Vision, pp. 525–531. (2001)
16. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
17. Chen, J., Shan, S., He, C., Zhao, G., Pietikainen, M., Chen, X., Gao, W.: WLD: a robust local image descriptor. IEEE Trans. Pattern Anal. Mach. Intell **32**, 1705–1720 (2010)
18. Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3D objects. Int. J. Comput. Vis. **73**(3), 800–807 (2007)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition. San Diego, CA, June 20–25 (2005)
20. Guo, X., Cao, X., Zhang, J., Li, X.: Mift: A mirror reflection invariant feature descriptor. In: Proceedings of ACCV (2009)
21. Ojala, T., PietikaÈinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognit. **29**, 51–59 (1996)
22. Chung, Y.C., Tony, H.X., He, Z.: Building recognition using sketch-based representations and spectral graph matching. In: IEEE International Conference on Computer Vision (ICCV 2009), Kyoto
23. Ke, Y., Sukthankar, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. CVPR, Washington, DC (2004)
24. Mahdian, B., Siac, S.: A bibliography on blind methods for identifying image forgery. In: Signal Processing: Image Communication, pp. 389–399. (2010)
25. Kumar, S., Das, P.K.: Copy-move forgery detection in digital images: progress and challenges. Int. J. Comput. Sci. Eng. **3**(2), 652–663 (February 2011)
26. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**, 381–395 (1981)
27. Raguram, R., Frahm, M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: ECCV 2008, Part II, LNCS, vol. 5305, pp. 500–503. Springer, Heidelberg (2008)
28. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceeding of the 15th International Conference on Machine Learning, pp. 445–453. Morgan Kaufmann, San Francisco (1998)
29. CASIA Image Tampering Detection Evaluation Database, ver. 2.0. http://forensics.idealtest.org. (2010)
30. Granty, R., Aditya, T., Madhu, S.: Survey on passive methods of image tampering detection. In: 2010 International Conference on Communication and Computational Intelligence (INCOCCI), Dec., pp. 431–436. (2010)
31. Vedaldi, A., Fulkerson, B.: An Open and Portable Library of Computer Vision Algorithms. http://www.vlfeat.org/. (2008)
32. Beis, J., Lowe, D.G.: Shape indexing using approximate nearest-neighbor search in high-dimensional spaces. In: Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 1000–1006. (1997)
33. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
34. Huang, D., Shan, C., Ardabilian, M., Wang, Y., Chen, L.: Local binary patterns and its application to facial image analysis: a survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **41**(6), 765–781 (2011)
35. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. In: Technical Report CVR-TR-2004-01, Beckman Institute, University of Illinois (2004)
36. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: European Computer Vision Conference (ECCV) (2006)
37. Canny, J.: A computational approach to edge detection. IEEE Trans Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
38. Powers, D.M.W.: Evaluation: from precision, recall and f-factor to roc, informedness, markedness & correlation. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)
39. Muhammad, G., Hussain, M., Bebis, G.: Passive copy move image forgery detection using undecimated dyadic wavelet transform. Digital Investig. **9**, 49–57 (2012)
40. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color and texture cues. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5), 530–549 (2004)

Author Biographies

Maryam Jaberi holds a B.S. degree in Software Engineering from Tarbiat Moallem University of Tehran, Iran (2004), an M.S. degree in Computer Engineering from AmirKabir University of Technolology, Iran (2007), and an M.S. degree in Computer Science and Engineering from the University of Nevada, Reno (2012). Currently, she is a Ph.D. student in the Department of Electrical Engineering and Computer Science at the University of Central Florida, Orlando. Her research interests include Computer Vision and Image Processing, Machine Learning, Knowledge Discovery and Data Mining.

George Bebis received a B.S. degree in mathematics and a M.S. degree in computer science from the University of Crete, Greece, in 1987 and 1991, respectively, and a Ph.D. degree in electrical and

computer engineering from the University of Central Florida, Orlando, in 1996. Currently, he is a Professor in the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR), Director of the UNR Computer Vision Laboratory (CVL), and Visiting Professor at King Saud University. Prior to joining UNR, he was a Visiting Assistant Professor in the Department of Mathematics and Computer Science at the University of Missouri-St. Louis (UMSL). His research interests include computer vision, image processing, pattern recognition, machine learning, and evolutionary computing. His research has been funded by NSF, NASA, ONR, NIJ, and Ford Motor Company. Dr. Bebis is an Associate Editor of the Machine Vision and Applications journal and serves in the editorial boards of the International Journal on Artificial Intelligence Tools and the Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization journal. He has served in the program committees of various national and international conferences and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for Innovation and Entrepreneurship.

Muhammad Hussain is an Associate Professor in the Department of Computer Science, King Saud University, Saudi Arabia. He received both his M.Sc. and M.Phil., from the University of the Punjab, Lahore, Pakistan, in 1990 and 1993, respectively. In 2003, he received a Ph.D. in Computer Science, specializing in Computer Graphics from Kyushu University, Fukuoka, Japan. He worked as a researcher at Japan Science and Technology Agency from April 2003 to September 2005. In September 2005, he joined King Saud University as an Assistant Professor. He worked on a number of funded projects in Kyushu University, Japan and King Saud University, Saudi Arabia. His current research interests include multiresolution techniques in Computer Graphics and Image Processing.

Ghulam Muhammad received his Bachelor degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 1997, and M.E. and Ph.D. degrees in 2003 and 2006, respectively, from Toyohashi University of Technology, Japan. After serving as a JSPS (Japan Society for the Promotion of Science) fellow, he joined as a faculty member in the College of Computer and Information Sciences at King Saud University, Saudi Arabia. His research interests include digital signal processing, automatic speech recognition, and multimedia forensics.