This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

Segmentation-based Image Copy-move Forgery Detection Scheme

Jian Li, Xiaolong Li, Bin Yang, and Xingming Sun *Senior Member, IEEE,*,

***Abstract*—In this paper we propose a scheme to detect the copy-move forgery in an image, mainly by extracting the key-points for comparison. The main difference to the traditional methods is that the proposed scheme first segments the test image into semantically independent patches prior to keypoint extraction. As a result, the copy-move regions can be detected by matching between these patches. The matching process consists of two stages. In the first stage, we find the suspicious pairs of patches that may contain copy-move forgery regions, and we roughly estimate an affine transform matrix. In the second stage, an EM-based algorithm is designed to refine the estimated matrix and to confirm the existence of copy-move forgery. Experimental results prove the good performance of the proposed scheme via comparing it with the state-of-the-art schemes on the public databases.**

***Index Terms*—Copy-move forgery detection, image forensics, segmentation.**

I. Introduction

An image with copy-move forgery (CMF) contains at least a couple of regions whose contents are identical. CMF may be performed by a forger aiming either to cover the truth or to enhance the visual effect of the image. Normal people might neglect this malicious operation when the forger deliberately hides the tampering trace (Figure 1). So we are in urgent need of an effective CMF detection (CMFD) method to automatically point out the clone regions in the image. And CMFD is becoming one of the most important and popular digital forensic techniques currently [1].

In the literature there are mainly two classes of CMFD algorithms [1]. One is based on block-wise division, and the other on keypoint extraction. They both try to detect the CMF through describing the local patches of one image. The former first divides the image into overlapping blocks and then finds the CMF by looking for the similar blocks. In [2] the authors proposed such a kind of method based on DCT describing the block, and they also decreased the complexity of the matching process by means of dictionary sorting. Because the descriptor of the block is important for the algorithm, various description methods like DWT, PCA etc were tested in these papers [3]–[8]. Among them Zernike moment [8], [9] may be the best choice in terms of detection accuracy and robustness. Besides, some post-processing techniques were proposed to improve the CMFD algorithms' efficiency. For example, in [9] the authors provided a method for the selection of duplicated blocks, namely SATS (Same Affine Transformation Selection). This method was able to improve the robustness of the detection algorithm against some attacks like rotation. The second class of algorithms detects the CMF through observing the keypoints in the image [10]–[15]. SIFT [16] and SURF [17] might be the most widely used keypoints for CMFD[1]. In some papers like [13]–[15], the authors estimated the transform matrix between the copying source region and pasting target region as well as detecting CMF in the image. In order to remove the effect of unwanted outliers, RANSAC [20] was often employed to guarantee the robustness of the estimation. In [1] the authors further improved the accuracy of the estimation result obtained by RANSAC via the gold standard algorithm [21, pp.130]. Because the number of the keypoints is much smaller than that of the blocks divided in an overlapping way, the keypoint-based algorithms require less computational resource than the block-based ones. Readers are referred to [1], [22] for some

Fig. 1: CMF examples from Christlein *et al.*'s database [1]. The left column gives the original images, and right column gives the images with CMF.

This work was supported in part by Jiangsu Basic Research Programs - Natural Science Foundation(BK20141006, BK20131004), in part by the Natural Science Foundation of the Universities in Jiangsu Province (14KJB520024), and in part by the NSFC(61173142, 61232016, 61379151 and 61300238), and in part by The Startup Foundation for Introducing Talent of NUIST(2012x053), and in part by Priority Academic Program Development of Jiangsu Higer Education Institutions and in part by Suzhou Science Project - Applied Basic Research (SYG201315) and PAPD fund.

Jian Li and Xingming Sun are with Jiangsu Engineering Center of Network Monitoring and with School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044.

Xiaolong Li and Bin Yang are with Institute of Computer Science and Technology, Peking University, Beijing 100871, China.

[1]As basic computer vision techniques, SIFT and SURF may be used in a large variety of different applications like smart home etc [18], [19].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

survey and evaluation works.

In this paper we propose a new framework for CMFD. The test image is first segmented into non-overlapped patches. Then the mission of CMFD in one image is transfered to partial matching between the obtained patches, which is a problem having been deeply studied in the computer graph research domain [23]–[26]. Based on the EM algorithm [27] we propose a new solution for the problem which has been proved to be an extension of the classic registration method iterative closest point (ICP) [23]. Our solution performs CMFD with two stages. The aim of the first stage is to find the suspicious matches, and a transform matrix between them is roughly estimated. Then in the second stage we confirm the existence of CMF by means of refining the transform matrix. Experimental results show that the proposed CMFD scheme outperforms most prior arts, especially the keypoint-based ones in terms of detection rate.

The rest of the paper is organized as follows. In Section II we first revisit the issues about CMFD and then show the framework of our proposed scheme based on image segmentation. Section III and IV describe the first stage and the second stage of matching process, respectively. The experimental results are given in Section V, followed by conclusion in Section VI.

II. Overview of the Proposed CMFD System and Image Segmentation

In this section, via revisiting the important issues involved in CMFD we first give the framework of our proposed scheme, and then we explain the reason for using image segmentation.

A. CMFD Revisiting and the Framework of the Proposed Scheme

In order to obtain a convincing detection result we would always like to acquire as much forensic information as possible from the test image. So the mission of CMFD is not only to determine if an image has some regions containing identical contents, but also to locate these tampered regions. To this end, we can describe the image with a set of local patches, like the blocks or keypoints in traditional CMFD schemes, and transfer CMFD into a problem of comparison among these local patches. The comparison process may be time-consuming if the number of the patches is too large. For example, the block-based methods [3], [4], [8] usually need a huge amount of time to detect an image. So it is important to decrease the number of patches for comparing. In this regard, the keypoint-based methods are faster and more favorable than the block-based ones, because the number of the image keypoints is smaller than that of the divided blocks.

However, on the other hand, keypoint-based method also has the following two problems. Firstly, the keypoints lying spatially close to each other should not be compared because they may be naturally similar. The determination of the shortest distance between two comparable keypoints is tricky. Most prior arts empirically select this threshold but neglect its relationship with the image size and content. Secondly, it is uneasy to accurately localize and distinguish the copying source region and the pasting target region, because, unlike the overlapping blocks, the keypoints are often not concentrated together. To deal with this problem Amerini *et al.* proposed a method based on clustering the matched keypoints [14], which was also adopted by the CMFD evaluation framework [1]. This method was further improved in [28] where the clustering object became a vector associated to the candidate transform estimation. It is shown that the new clustering-based CMFD scheme significantly raise the accuracy of localization of CMF regions.

We know that an image is seldom forged aimlessly. Hence the copy-move regions should have a certain meaning. In this light, we propose to segment the test image into a number of non-overlapped patches (refer to Figure 2). Then the CMFD can be performed by matching these patches, as long as the pasting target and copying source regions are not in the same patch[2].

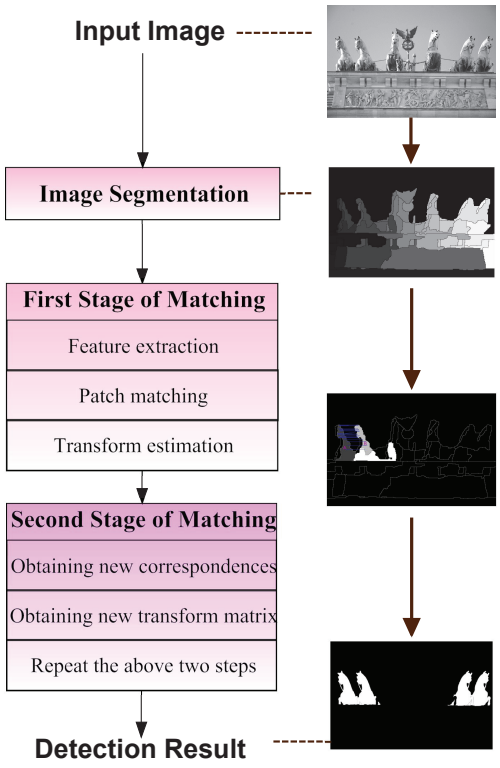


Fig. 2: Flowchart of the proposed CMFD framework

We note that this is not the first CMFD system that employs image segmentation technique. Farid [29] proposed to detect the duplication in science images by grouping the pixels with similar properties. However, being designed for the science images such as gel and micrograph, Farid's method is not efficient and robust enough for normal images that are content rich and contain many different textures. Recently, Liu *et al.* also proposed a forgery detection method using JPEG features and local noises discrepancies [30], where segmentation is

[2]If unluckily the two regions are located in one same patch, we may either perform CMFD within each patch using the traditional method [13], [14], or suitably set the size of the patch to avoid it that one patch includes all the CMF regions. In our implementation, we employ the latter solution which will be discussed in Section II-B.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

proved to be useful to splicing detection. In our proposed CMFD scheme, after segmenting the image, we perform the first stage of affine estimation. During this stage we first extract the keypoints from the whole image and construct a k-d tree. Then the KNN (k-nearest neighbor) search is performed in each region for each keypoint to find a possible correspondence. One region is recorded if it has a certain proportion of keypoints matched with another one. Finally we estimate the affine relationship between the region pairs. The estimated transform matrix is the input to the second stage of matching process, where we iteratively refine the matrix via a probability model based on the EM algorithm.

B. Image Segmentation

In order to separate the copying source region from the pasting target region, the image should be segmented into small patches, each of which is semantically independent to the others. This job is best done by an expert with much experience of digital forensics. In our implementation, however, we only consider the automatic approach and leave the expert interfering method for future work. After testing four famous image segmentation methods [31]–[34], it is observed that the segmentation method does not greatly influence the CMFD's efficiency. Among them the methods in [32], [34] are more favorable owing to their comparatively lower complexity. In most cases, one image sized 800×600 can be segmented in 15 seconds using a personal computer (3.3GHz CPU, 4G RAM). Figure 3 gives an example of image segmentation obtained by [34].

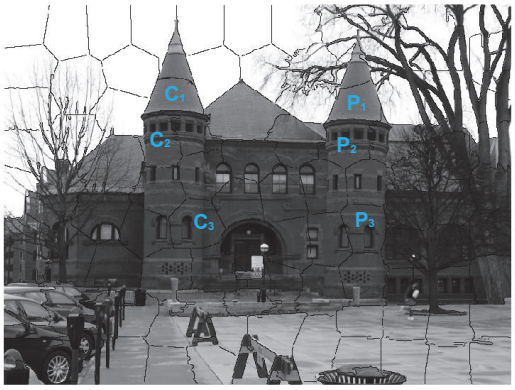


Fig. 3: Example of image segmentation. The two towers in the test image are CMF regions. We segment the image by means of the SLIC algorithm [34]. It can be observed that one CMF region is divided into several patches. Nevertheless, the two CMF regions are segmented in a similar way. In particular, there are some patch pairs with a large proportion of identical contents, say C_1 and P_1, C_2 and P_2, etc.

One may concern the scenario that segmentation cannot help us to separate the CMF regions into different patches. As mentioned above, in order that two CMF regions do not exist in the same patch, we should not coarsely segment the image. In our implementation, each image is empirically segmented into no less than 100 patches (refer to Section V for a further explanation), and thus, a CMF region may be in two or more patches (refer to Figure 3). In consequence the useful information for CMFD is reduced in each patch. However, to obtain a convincing detection result we need not a large number of keypoints (sometimes four is enough). Furthermore, because the CMF region exists in many patches, we meanwhile have more than one chance to find the tampering operation. Extensive experiments prove that the applied segmentation method is able to provide us with satisfying results.

III. First Stage of Matching

In this section we will introduce the first stage of the matching process of our proposed CMFD system. The three steps (refer to Figure (2)) involved in this stage will be detailed in the following three subsections.

A. Keypoint Extraction and Description

In our implementation, we employ vlFeat[3] [35] software to help us to detect and describe the keypoints. There are many kinds of keypoint detection and description methods. The common co-variant keypoint detection and description algorithms, such as difference of Gaussian (DoG), Harris-affine and Hessian-affine [16], [36], can provide similar detection performance. In our implementation we just employ the default setting of vlFeat for keypoints detection and description, namely SIFT [16]. Although the methods of keypoint detection and description are not rather important, note that the number of the keypoints should be larger than 2000 for good performance.

B. Matching between Patches

Next we look for the suspicious pairs of patches that have many similar keypoints. This process is performed by comparing each patch with the rest. Refer to Figure 4, assume that patch A is considered at this time. Define the distance

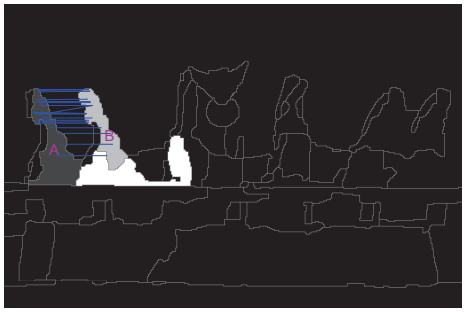


Fig. 4: Find the suspicious pairs of patches. The lines connecting the points in patch A and B represent the matched keypoint pairs.

between two keypoints by the L-2 norm of the difference between their descriptors. In patch A for each keypoint we search its K nearest neighbors that are located in the other

[3]Version 0.9.18

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

patches. Considering there are usually more than one couple of copy-move regions in the image, we set $K = 10$ in our implementation. We should not take all the K searched keypoints into consideration, but only if the difference is smaller than a threshold (0.04 in our implementation), the two keypoints are considered to be matched. In other words, each keypoint in patch A is corresponding to no more than K keypoints in the remaining patches. We know that the target and source regions should have a large proportion of matched keypoints. If a large proportion of the matched correspondences of A are located in another certain patch, say B in Figure 4, A and B are considered to be a suspicious pair of patches where we may find CMF regions. So a threshold φ is defined to find the matched patches. In our implementation, φ is empirically set as 10 times the average number of keypoints per patch, i.e.,

$$\varphi = 10\frac{|\{keypoints\}|}{|\{patches\}|}. \tag{1}$$

With the help of φ, most patches are eliminated from the estimation of transform matrix and, of course, the second stage of matching process. Besides, like the traditional keypoint-based CMFD schemes [13], we decrease the complexity of searching K nearest neighbors for a keypoint from $O(n^2)$ to $O(nlogn)$, by constructing a k-d tree provided by vlFeat software [35].

C. Affine Transform Estimation

After detecting a suspicious pair of patches, we preliminarily know where the copying source region and pasting target region are. Then we estimate the relationship between these two regions in terms of a transform matrix H, such that

$$\vec{x}' = H\vec{x}, \tag{2}$$

where $\vec{x}$ and $\vec{x}'$ are the coordinates[4] of the pixels in the copying source region and pasting target region, respectively. Some proposed CMFD algorithms, especially the block-based ones [2]–[4], only focus on finding the tampering regions and do not further investigate the transform relationship between the copying source region and pasting target region. In fact, it is rather helpful for the CMFD scheme to estimate the transform matrix between the two regions. Firstly, we are able to remove some falsely detected CMF regions as they do not have a set of points with uniform transform relationship. Secondly, more important, the CMFD is enhanced by providing the tampering detail about one image. So most recent CMFD algorithms choose to calculate the transform matrix [9], [13]–[15].

In order to avoid leaving additional forgery traces in an image, the forgers often do not further change the copying source region. As a result, we can simply assume that the error of keypoints extraction only exists in the target regions. And the estimation of transform between the source region and target region can be made by means of a classical method [21, pp.95]. That is, no less than three random non-collinear matched keypoints $(\vec{x}_i, \vec{x}'_i)$ are first used to calculate the transform matrix H by means of minimizing the geometric distance $\sum_i d(\vec{x}'_i, H\vec{x}_i)$. As the existence of noise in the keypoints detection, we also employ the robust estimation method, namely RANSAC [20], to find a transform matrix H that is the best among a certain number of trials. This method is also adopted by some other CMFD schemes [13], [14].

In this transform estimation process, some small sized regions with limited number of keypoints, say 5, influence the detection accuracy. According to Pan *et al.*'s results, it is hard to accurately detect the CMF forgery regions with a size smaller than 32×32. Clearly, the main reason is because the forgery regions are too small, and a limited number of keypoints cannot resist the possible errors in keypoint extraction. So we propose a second stage of matching process, where additional information of the image is employed to improve the accuracy of transform estimation.

IV. Second Stage of Matching

In the first stage of matching process, we have found the suspicious pairs of patches as well as the transform matrix between them. Although RANSAC [20] can provide us with a robust estimation of transform matrix, it is still not accurate enough. Furthermore, some of these detected patches may be just false alarm containing not any CMF regions. In this section, we will introduce our second stage of matching process where the estimation of the transform matrix is refined via an EM-based algorithm. And the false alarm patches might also be eliminated in this stage.

A. CMF Determination Based on Probability

In the first stage of matching process, we made use of the detected keypoints in the copying source region and pasting target region to estimate a transform matrix H. This process follows the traditional way of computer vision [21]. In particular, the pixels not around the keypoints are abandoned. It is mainly because computer vision usually focuses on the research of transform estimation of two distinct images, in which case we are able to obtain a comparatively larger number of matched keypoints. However, in the CMFD case the forgery regions are sometimes so small that only a limited number of keypoints can be detected there. As a result, the detection result of the first stage is not convincing because we do not have enough keypoints.

So in the second stage we propose to exploit all the pixels in the matched patches to find out a more accurate estimation $\bar{H}$. Meanwhile, the pixels belonging to the CMF regions would be more clearly distinguished from the background. Since the really matched pixels in the copying source region and pasting target region should be close to each other, we change the definition of the relationship between them in (2) to

$$f(\vec{x}) = f(H^{-1}\vec{x}'), \tag{3}$$

where $f(\cdot)$ is an image characteristic function with respect to the pixel coordinate, such as the image intensity or some

[4]Please note that the homogeneous coordinates are used when estimating the transform matrix. Specifically, H is a 3×3 matrix with the third row equal to (0, 0, 1). Consequently, $\vec{x}$ is a three-dimensional column vector. The first two elements are the horizontal and vertical coordinates of one pixel while the third equals to 1. The readers are referred to [21, pp.2] for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

other advanced image descriptors. In our implementation, we employ the dense SIFT descriptors [35] for robustness and efficiency. Equation (3) is hard to hold owing to the estimation error etc. Nevertheless, based on (3), we are able to observe the probability of a pixel at $\vec{x}$ located in the CMF region. We introduce a random variable z to indicate that a pixel is located at CMF region ($z = 1$) or not ($z = 0$). Then the probability is given by

$$P(\vec{x}|z=1,H) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(f(\vec{x})-f(H^{-1}\vec{x}'))^T(f(\vec{x})-f(H^{-1}\vec{x}'))}{2\sigma^2}}, \tag{4}$$

Equation (4) is based on the assumption that the difference between the matched pixels follows a Gaussian distribution (zero mean and variance σ^2). Although this assumption is not theoretically strict, extensive experiments show that it is able to provide us with satisfying result.

B. Obtaining the New Correspondences of the Pixels

Denote the transform matrix we estimated in the first stage by H_0 for differentiation here. As H_0 is not accurate enough, the $\vec{x}'$ obtained by (2) may not be the real correspondence of $\vec{x}$. So we search a new correspondence of $\vec{x}$ in the pasting target region, such that the pixel located at the new correspondence position is more similar to the pixel at $\vec{x}$ than the old correspondence in terms of their local feature descriptions.

We first align the image by means of the estimated transform matrix, i.e. a new transformed image is obtained by,

$$\hat{I} = H_0^{-1} \cdot I. \tag{5}$$

Then the process of searching new pixel pairs is illustrated in Figure 5. It is an example where the copying source region is marked by a dashed line and the pasting target region by a black solid line. After transforming the pasting target region via (5), the transformed pasting target region (red solid line) is not well aligned with the copying source region. There is a pixel $\boldsymbol{x}$ in the copying source region and its original matched point $\boldsymbol{x}'$ in the pasting target region. The coordinates of $\boldsymbol{x}$ and $\boldsymbol{x}'$ satisfy the relationship defined in (2). And we indicate the correspondence of $\boldsymbol{x}$ in the transformed image by $\hat{\boldsymbol{x}}$ which has the same coordinate with $\boldsymbol{x}$. Consequently, the pixel $\boldsymbol{x}$ and its correspondence $\hat{\boldsymbol{x}}$ in the transformed image may not have the identical value. In other words, $\boldsymbol{x}$ and its real correspondence are unlikely at the same position. We need to find the new correspondences for these pixels. Specifically, for a pixel coordinate $\vec{x}$ in the copying source region, we find one pixel coordinate $\hat{\vec{x}}$ in the transformed image satisfying

$$\hat{\vec{x}} = \arg\min_{\vec{y}\in\hat{I}} \left(f\left(I\left(\vec{x}\right)\right) - f\left(\hat{I}\left(\vec{y}\right)\right)\right). \tag{6}$$

In order to decrease the computational complexity, the new position could be only searched around $\hat{\boldsymbol{x}}$. In our implementation, five neighboring pixels are considered, i.e. the up, down, left, right and center ones of $\hat{\boldsymbol{x}}$. After obtaining the new correspondence, we transform its coordinate back to the original image via (2), namely

$$\vec{x}' = H\hat{\vec{x}}, \tag{7}$$

and find the new matching pixel $\tilde{\boldsymbol{x}}'$.

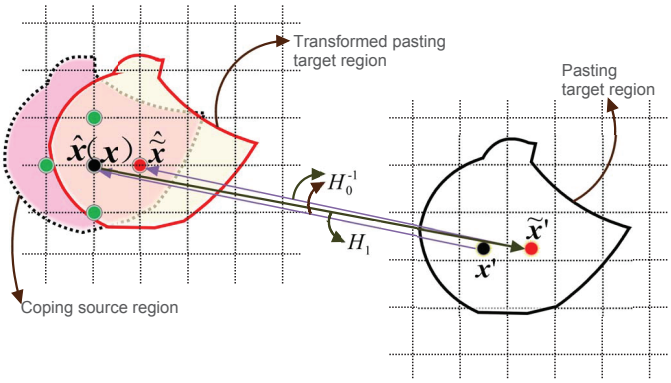


Fig. 5: Alignment of the pasting target region to copying source region. We have $\boldsymbol{x}$ in the copying source region (red dashed line) and $\boldsymbol{x}'$ in the pasting target region (black solid line). The coordinates of $\boldsymbol{x}$ and $\boldsymbol{x}'$ satisfy $\vec{x} = H_0^{-1}\vec{x}'$. After transforming the image by (5), we obtain $\hat{\boldsymbol{x}}$ in the transformed image which is of the same coordinate with $\boldsymbol{x}$ and is corresponding to $\boldsymbol{x}'$ in the original image. Considering the values of $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$ may not be similar, we look for a new pixel in the transformed image that is more close to $\boldsymbol{x}$ according to (6). The search range is four closest neighbors as well as $\hat{\boldsymbol{x}}$ itself. After obtaining the new matched pixel $\hat{\hat{\boldsymbol{x}}}$, we transform it back to $\tilde{\boldsymbol{x}}'$ via H_0.

C. Iterative Re-estimation of the Transform Matrix

Using the newly matched pixel pairs we wish to estimate a more convincing matrix $\bar{H}$. Please note that some of these pixel pairs are outliers that are located outside the CMF region. Furthermore, some correspondences are not accurate enough because they may be at the smooth image regions. One natural solution is RANSAC as it is rather good at handling outliers. However, there usually are a large number of pixel pairs and hence RANSAC is too time-consuming.

We have two classes of pixels in each segmented patch. One is the CMF region, the other is the background. Distinguishing the CMF region from the background is the same problem as classifying these two kinds of pixels. We propose to employ the EM algorithm [27] to this end. The EM algorithm is a useful method for statistical parameter estimation of the samples with underlying distributions. The algorithm repeats a procedure until a target variable converges. The procedure consists of an E-step and an M-step. In the E-step, we calculate the following value which is an expectation of the log likelihood $P(X, z|H_n)$, with respect to the conditional distribution $P(z|X, H_{n-1})$, i.e.,

$$Q(H_n|H_{n-1}) = E_{z|X,H_{n-1}} \ln[P(X, z|H_n)], \tag{8}$$

where X represents all the coordinates of the pixels in the current patch, namely $X = (\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n)$. And z is the same random variable as in (4). Then in the M-step we calculate H_n via maximizing Q. If the new estimated matrix H_n is similar to H_{n-1}, stop the iteration and output $\bar{H} = H_n$. Otherwise, take H_n as the initial estimation and repeat the above two steps.

These two steps can be explained as follows. We have an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

estimation of transform matrix H_{n-1}, and want to obtain a new one H_n which is more accurate than the last one. With the help of H_{n-1}, for the pixels of the current patch we obtain the correspondences in the matched patch according to (2). Then for each pixel a new correspondence around the original one is re-detected as described in the last subsection. After that we calculate the probability of the pixel being in the copy-move region, namely $P(z|X, H_{n-1})$. Then we re-estimate a new transform matrix H_n such that the pixels in the CMF region are able to match their newly obtained correspondences as much as possible. From a statistical point of view, it is to maximize the expectation of the log-likelihood function $P(X, z|H_n)$.

Maximizing (8) needs the definition of the probability $P(X, z|H_n)$. In the case of $z = 1$, we define

$$P(X, z = 1|H_n) = \prod_i e^{(\vec{x}_i - H_n^{-1}\vec{\tilde{x}}'_i)^T(\vec{x}_i - H_n^{-1}\vec{\tilde{x}}'_i)}. \quad (9)$$

This definition is derived from the fact that the pixels in a CMF region are related to their correspondences with the transform matrix H_n. Please note that the definition here is slightly different in comparison with (4). We do not make use of the pixel feature again. The main reason is to simplify the calculation in the maximization step. Besides, (9) requires the coordinate of the current pixel and that of its new correspondence searched by means of (6) and (7), which implies that we will perform the searching process in Section IV-B at each E-step.

In the case of $z = 0$, $P(X, z|H_n)$ gives the probability that a pixel does not belong to forgery regions. Considering the image pixels exist randomly, the probability can be simply considered to follow a uniform distribution, i.e.,

$$P(X, z = 0|H) = \prod_i \frac{1}{U}, \quad (10)$$

where U is a number related to the value range of the difference between two arbitrary pixels.

Because $P(X, z = 0|H_n)$ is defined as a constant, in the maximization step we need not consider the case of $z = 0$. With (9) and (10) we can rewrite (8) as follows,

$$\tilde{Q}(H_n|H_{n-1}) = E_{z=1|X,H_{n-1}} \ln[P(X, z = 1|H_n)] \quad (11)$$

$$= E_{z=1|X,H_{n-1}} \ln[\prod_i e^{(\vec{x}_i - H_n\vec{\tilde{x}}'_i)^T(\vec{x}_i - H_n\vec{\tilde{x}}'_i)}] \quad (12)$$

$$= E_{z=1|X,H_{n-1}} \sum_i (\vec{x}_i - H_n\vec{\tilde{x}}'_i)^T(\vec{x}_i - H_n\vec{\tilde{x}}'_i) \quad (13)$$

$$= \sum_i P(z = 1|\vec{x}_i, H_{n-1})(\vec{x}_i - H_n\vec{\tilde{x}}'_i)^T(\vec{x}_i - H_n\vec{\tilde{x}}'_i). \quad (14)$$

Then the new H_n can be obtained by

$$H_n = \arg\max_{H_n} \tilde{Q}(H_n|H_{n-1}). \quad (15)$$

$P(z = 1|\vec{x}, H_{n-1})$, namely the conditional probability involved in (14), represents the probability of a pixel being in the copy-move region ($z = 1$) given the coordinates of the pixel and its correspondence with respect to a transform matrix H_{n-1}. According to the conditional probability rule, we define

$$P(z = 1|\vec{x}, H_{n-1}) = \frac{P(z = 1|H_{n-1})P(\vec{x}|z = 1, H_{n-1})}{\sum_z P(\vec{x}, z|H_{n-1},)} \quad (16)$$

We have defined $P(\vec{x}|z = 1, H_{n-1})$ in (4). Consistent with (9) and (10), $P(\vec{x}, z = 1|H)$ is defined by

$$P(\vec{x}, z = 1|H) = e^{(\vec{x} - H_n\vec{\tilde{x}}')^T(\vec{x} - H_n\vec{\tilde{x}}')}, \quad (17)$$

and $P(\vec{x}, z = 0|H)$ is defined by $\frac{1}{U}$.

With (16), we can obtain the derivative of $\tilde{Q}$ with respect to H_n. And by making the derivative equal to zero, (15) can be solved, that is

$$H_n = (X * W * X^T)^{-1} * X * W * \tilde{X}'^T, \quad (18)$$

where $X = (\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n)$ and $\tilde{X}' = (\vec{\tilde{x}}'_1, \vec{\tilde{x}}'_2, \cdots, \vec{\tilde{x}}'_n)$, represent the coordinates of the current pixels and its newly obtained correspondences (refer to Fig. (5)), respectively. And W is an $n \times n$ matrix of which the non-diagonal elements are all zeros, i.e.,

$$W = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & \cdots \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}, \quad (19)$$

where $w_i = P(z = 1|\vec{x}_i, H_{n-1})$. If $|H_n - H_{n-1}|$ is smaller than a predefined threshold ϱ ($\varrho = 0.03$ in our implementation), we stop the iteration and output H_n. Following the approach of [1], we transform the test image using H_n and compute the correlation coefficients between the transformed image and the original test image. The generated map of correlation coefficients is post-processed by means of filtering and morphological operations. The copying source region are found in the map where the correlation coefficients are larger than the background. The pasting target region can be obtained by transforming the copying source region with H_n. Otherwise, if $|H_n - H_{n-1}| > \varrho$, we continue the iteration. If the procedure cannot converge even after 70 iterations, we think CMF regions may not exist in this pair of patches and stop the procedure.

As a summary, we note that our second stage of matching process is also an extension of the classic registration algorithm, namely iterative closest point (ICP) [23]. Generally speaking, they both consist of the following three steps.

1) Obtaining the matched points.
2) Calculating the transform matrix.
3) Repeating the above two steps until a convergence condition is satisfied.

V. Experimental Results

A. Test Image Databases and Segmentation Settings

Table I presents two public available image databases involved in evaluation of our proposed CMFD scheme. The first one was constructed by Christlein *et al* [1], consisting of 48 base images and 87 copied snippets that are pasted to the other locations in the same image to make the forgeries. These snippets are carefully selected such that the CMF trace

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

TABLE I: Image databases to examine the performance of the proposed scheme

Databases	**Descriptions**
Benchmark database for CMFD evaluation [1]	Using the base images, copied snippets and software provided by Christlein *et al.*, we generate 48 original images, 48 images with plain CMF, and 1392 images with CMF that the copied snippets are processed (adding noise etc. Refer to Table IV.)
MICC-F600 [28]	A mixture of two other databases, namely MICC-F2000 [14] and the benchmark database above, containing 440 original images and 160 forged images in which the copied snippets are processed in a way different from Table IV.

is almost unnoticeable. The original sizes of the images are rather large (e.g., beach_wood.png, 3264×2488). However, in some cases like the Internet and wireless multimedia applications, we are often faced with small sized images. So in our experiment the width and the height of the test images are set to no larger than 800 by means of resizing. Furthermore, we note that the process of resizing will make it difficult to extract keypoints from the CMF regions, which is rather challenging for the keypoint-based schemes.

The second database MICC-F600 was introduced by Amerini *et al* [28]. Most of its original images (400 ones) are from MICC-F2000 [14]. The forged images in MICC-F600 are also derived from the aforementioned base images and copied snippets provided by Christlein *et al.* Nevertheless, most of the forged images in the two databases are not identical because the attacks performed are different. For instance, in MICC-F600 the copied snippets may be rotated by 30° and then scaled by 120% prior to pasting to the base images. Furthermore, in order to observe the influence of image size on the proposed scheme we do not resize the images in MICC-F600.

The images in the above two databases are segmented by vlFeat software [35]. Because the images in the first database are resized to approximately similar resolution, we segment these images in the same way, i.e., using a vlFeat function vl_quickseg with certain parameters. This function implements the quick shift image segmentation algorithm [32]. We set the two control parameters, *ratio* and *kernelsize*, to 0.7 and 1, respectively, such that each image is segmented to more than 100 patches.

Unlike the first database, we do not change the images in MICC-F600. These images vary from 800×533 to 3888×2592 in size. Thus, they should not be uniformly segmented. Furthermore, we find that vl_quickseg is not fast enough for the large images. So another vlFeat function vl_slic implementing the SLIC algorithm [34] is employed to segment the images in MICC-F600. SLIC is similar to the quick shift algorithm but is more efficient. Function vl_slic requires two parameters as well. One is *regulizer* used to control the regularization of the patches. We set it to 0.8 for all the images. The other parameter *regionsize* is related to the number of segmentation patches. Its value hence should be adaptive to the image size. Table II illustrates the empirical setting of *regionsize* in our experiment.

TABLE II: Segmentation Setting for the Images in MICC-F600

Image size (S)	*regionsize*	Example Image (Number of patches)
$S > (3000 \times 2000)$	200	Hedge(220)
$(3000 \times 2000) > S > (2000 \times 1000)$	150	Three_hundred(175)
$(2000 \times 1000) > S > (1000 \times 600)$	100	Knight_moves(224)
$S < (1000 \times 800)$	50	Giraffe(175)

B. Error Measures

Following the approach in [1], the performance of the CMFD scheme is also tested by detection error at two different levels, namely image level and pixel level. The detection error at the image level is measured by the ratio of the missing detection to the forged images (i.e. false negative rate, F_N), and the ratio of the false alarm to the original images (i.e. false positive rate, F_P). Mathematically,

$$F_N = \frac{|\{\text{Forged images detected as original}\}|}{|\{\text{Forged images}\}|}, \tag{20}$$

$$F_P = \frac{|\{\text{Original images detected as forged}\}|}{|\{\text{Original images}\}|}. \tag{21}$$

The detection error at the pixel level is measured by the common criteria, precision and recall. The precision calculates the ratio of the retrieved CMF pixels in all the retrieved pixels, and the recall calculates the ratio of the retrieved CMF pixels in all of the CMF pixels, mathematically,

$$precision = \frac{|\{\text{CMF pixels}\} \cap \{\text{retrieved pixels}\}|}{|\{\text{retrieved pixels}\}|}, \tag{22}$$

$$recall = \frac{|\{\text{CMF pixels}\} \cap \{\text{retrieved pixels}\}|}{|\{\text{CMF pixels}\}|}. \tag{23}$$

Besides, like [1] we also compute another criterion F_1 that combines both precision and recall, i.e.,

$$F_1 = 2 \cdot \frac{precision * recall}{precision + recall}. \tag{24}$$

C. Results on the First Database

We first examine the ability of the proposed scheme detecting plain copy-move forgery, namely no further attack is performed for the tampered image. This experiment involves the 48 original images as well as the 48 images with plain copy-move forgery. The result is shown by the detection errors in image level with false negative rate F_N and false positive rate F_P (defined in (20) and (21)). According to [1], SIFT and SURF are the most widely used keypoints for CMFD job. So in Table III we compare the proposed scheme with the results associated with these two keypoints. The results are from the CMFD algorithm implemented by Christlein *et al*, but may be different to those in [1] owing to our resizing the test images. From this experimental result we can see that our proposed CMFD scheme is corresponding to the smallest false negative

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

TABLE III: Result of plain CMF detection

Methods	F_N	F_P
SIFT	26/48	9/48
SURF	27/48	8/48
Proposed	8/48	17/48

rate, which means the proposed scheme is good at detecting the tampered images. However, the false positive rate of the proposed scheme is also larger than the others. We think the reason is two-fold. First, the second stage of matching cannot remove all the false alarm from the output of the first stage of matching. On the other hand, when detecting the suspicious pair of patches its threshold φ is set as loose as possible to avoid miss of detection. In consequence, some images are falsely detected especially those with repeated contents, say Statue etc. Secondly, recall that we employ DSIFT to describe the pixels in the second stage of detection. DSIFT descriptor is fast and robust to attacks, but is not discriminative enough. These two problems need to be solved in our future work to essentially improve the efficiency of the proposed scheme. Figure 6 shows two tampered images that can only be detected by the proposed scheme.

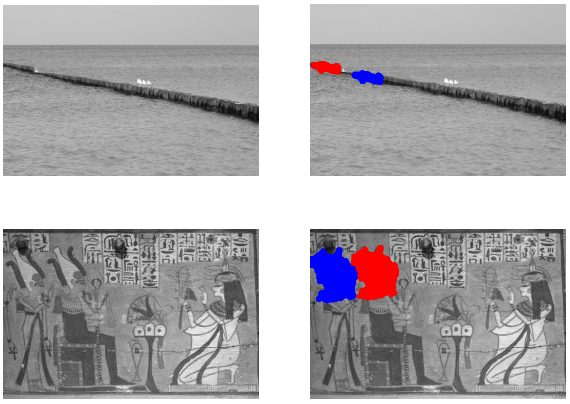


Fig. 6: Two tampered images detected by proposed scheme.

As mentioned above our setting on the threshold φ is one important reason blowing the false positive rate up. Thus the false positive rate may be decreased simply by adjusting φ. We plot the ROC curve in Figure 7 to show the trade-off between false positive and false negative when changing φ. It can be observed that the false positive rate can be smaller than 0.15 when set $\varphi = 20$. However, the false negative rate is increased to 0.33 at the same time. So adjusting the parameter φ only allows us to satisfy different detection requirements, but it does not improve the performance of the proposed scheme essentially. Since the both test databases are not large enough, it is difficult to obtain a parameter setting suitable to every images based on the results in Figure 7. Thus we still set $\varphi = 10$ in the following tests.

Next we test the robustness of our proposed CMFD scheme against various attacks. That is the copied snippets in the plain copy-move images further undergo signal processing

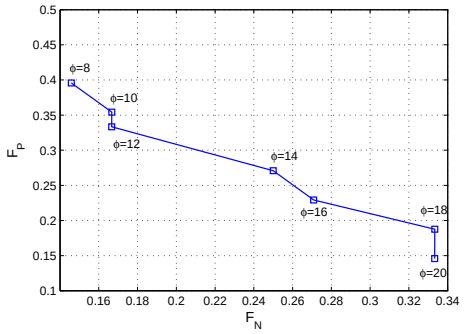


Fig. 7: ROC curve plotted when changing of parameter φ.

and geometric transformations in order to escape the CMFD. In this experiment, we consider 4 kinds of attacks, namely JPEG compression, adding noise, rotation, scaling down and up. Table IV gives the settings for these attacks which are almost the same as in [1]. With these attacks there are totally 1392 (29×48) images for robustness testing.

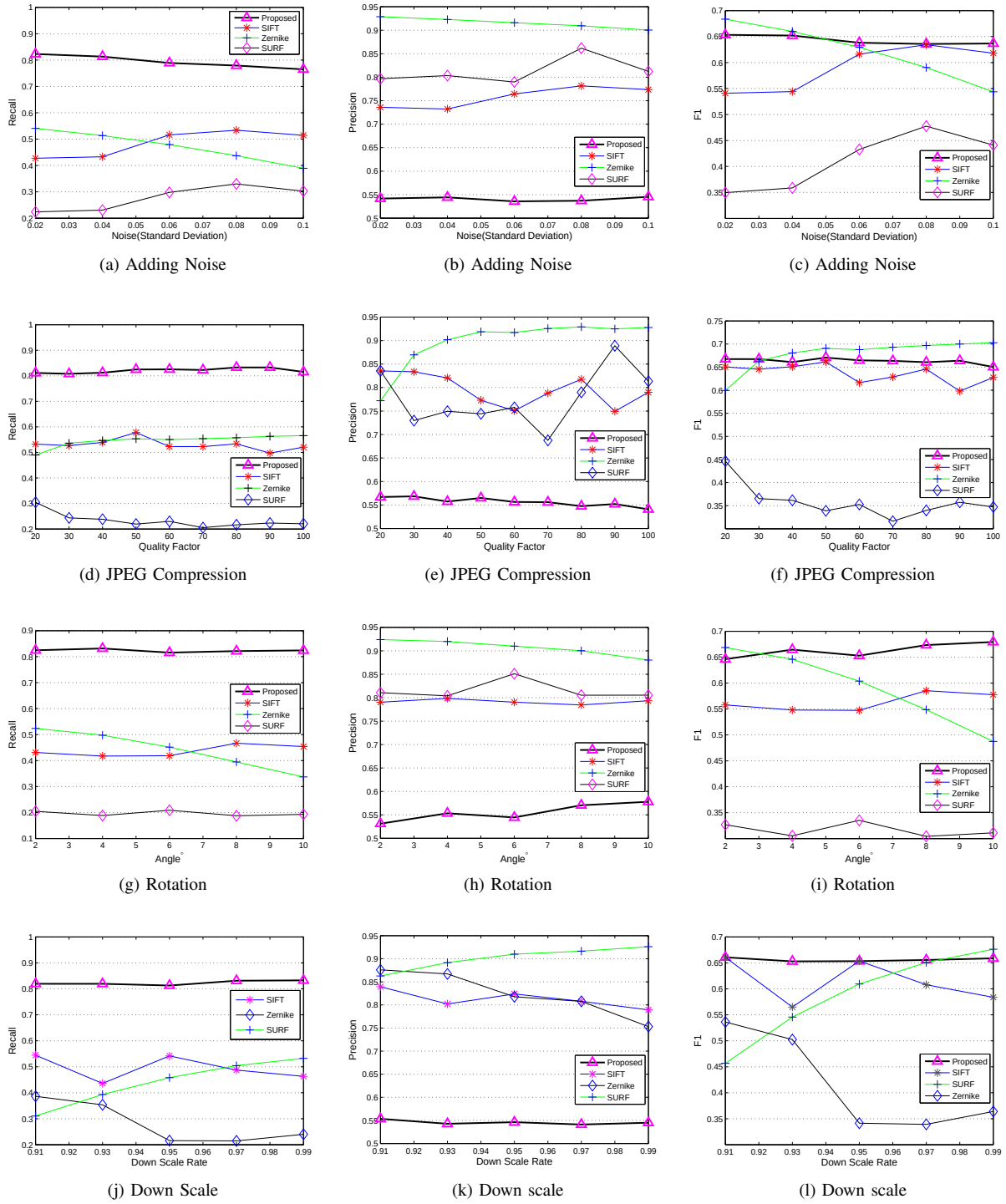
TABLE IV: Setting of the Attacks

Attacks	Parameters
Adding noise	Deviation (20:20:100)
JPEG	Quality Factor (20:10:100)
Rotation	Angle($2^\circ : 2^\circ : 10^\circ$)
Scaling	Ratio(0.91:0.02:1.09)

In this test we evaluate our proposed CMFD scheme by the precision and recall at the pixel level (defined in (22) and (23)). The experimental results are given in Figure 8. Besides SIFT and SURF, the results corresponding to Zernike moment are also illustrated. Please note that the results of the comparison schemes are different to those in [1] because of image resizing. We can see that under signal processing attacks, namely JPEG compression and adding noise, Zernike moment performs best mostly in terms of F_1 criterion owing to its good detection precision. However, its performance clearly drops down when increasing the intensity of attacks. On the contrary, the proposed scheme and the other two keypoint-based schemes are more robust to various attacks. In most cases, our proposed CMFD scheme outperforms the prior arts in terms of F_1 criterion especially under geometric attacks. Besides, our proposed CMFD scheme is with the best recall results among all the tested schemes, which means it is able to find the largest number of CMF regions. However, the precision of the proposed scheme is lower than that obtained by prior arts. We observe that our scheme is likely to detect human-made object (like windows of building) as CMF region. This is also consistent with the results given in Table III and literature [1]. So owing to its properties aforementioned, the proposed scheme may be used in the case that we wish no CMF regions escaping detection even with sacrifice of a little high false alarm rate.

In order to justify the effectiveness of our proposed estimation refinement step (Section IV), we further test the performance of our proposed CMFD scheme without the second stage of matching. Please note that although our proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security



(a) Adding Noise (b) Adding Noise (c) Adding Noise

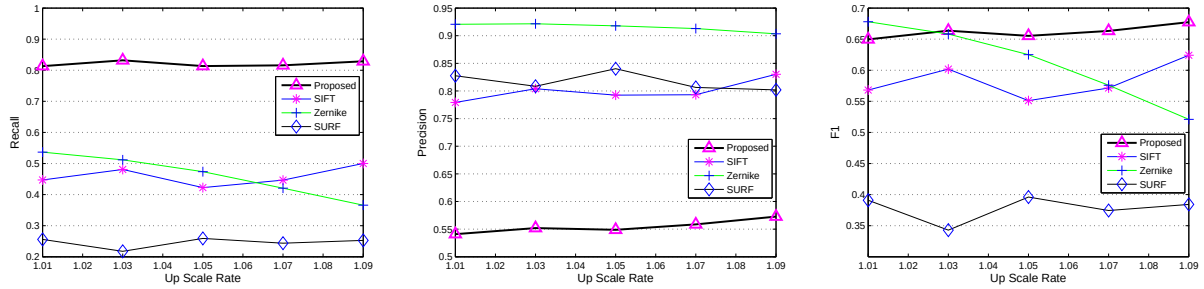
(d) JPEG Compression (e) JPEG Compression (f) JPEG Compression

(g) Rotation (h) Rotation (i) Rotation

(j) Down Scale (k) Down scale (l) Down scale

Fig. 8: Detection Results of different CMFD schemes against 5 kinds of attacks. The three columns give the recall, precision and F_1 results, respectively. Please note that the higher the curve the better the result is. The results of the comparison schemes are different to those in [1] because of image resizing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security



(m) Up Scale (n) Up scale (o) Up scale

Fig. 8: Detection Results of different CMFD schemes against 5 kinds of attacks. The three columns give the recall, precision and F_1 results, respectively. The results of the comparison schemes are different to those in [1] because of image resizing.

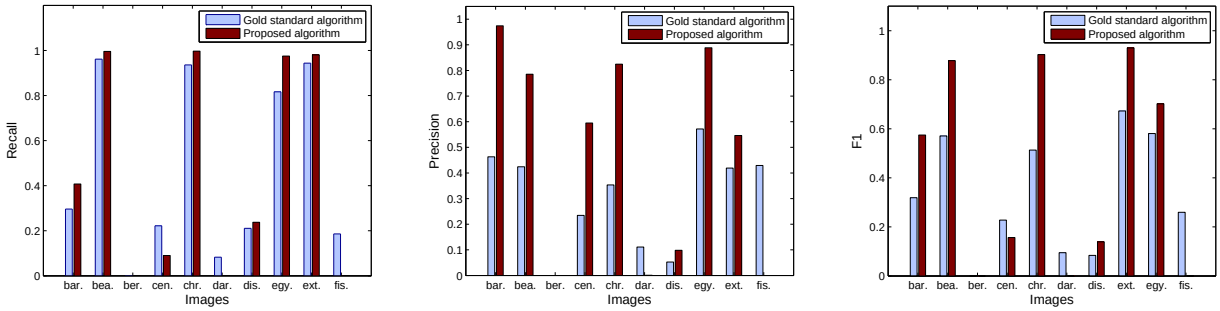


Fig. 9: Detection results of our proposed CMFD scheme with and without the proposed estimation refinement step.

refinement method is not used, the final estimation of the affine matrix here is refined as well using the gold standard algorithm [21, pp. 130] as in [1]. This justification experiment also employs the same criteria and test images as the robustness test above. But unlike Figure 8, Figure 9 presents the test results associated to the images instead of the attacks. Each bar shows an average error measure (precision, recall or F_1) calculated from one base image on which we perform 29 kinds of attacks as described in Table IV. We can find that in most cases the detection performance deteriorates if our proposed estimation refinement step is not taken. These results confirm the effectiveness of the second stage of matching. However, we point out that the second stage of matching requires additional computational cost, which needs our further work to improve it.

D. Test Results on MICC-F600

We also compare our proposed CMFD scheme with two prior arts [14], [28] on the database MICC-F600 (refer to Table I). The detection error at image level is given in Table V. It can be observed that the proposed scheme is with the lowest false negative rate but the highest false positive rate, which is rather consistent with the results on the benchmark database. The detection errors of the proposed scheme at pixel level for all the forged images are also calculated. The average $precision$, $recall$ and F_1 values are 0.86, 0.88 and 0.87, respectively. These results also prove the effectiveness of our segmentation setting in Table II. Figure 10 shows the detection results on the test images with CMF regions fused in the background. It can be observed that the proposed scheme detects most CMF regions.

TABLE V: Detection results on MICC-F600

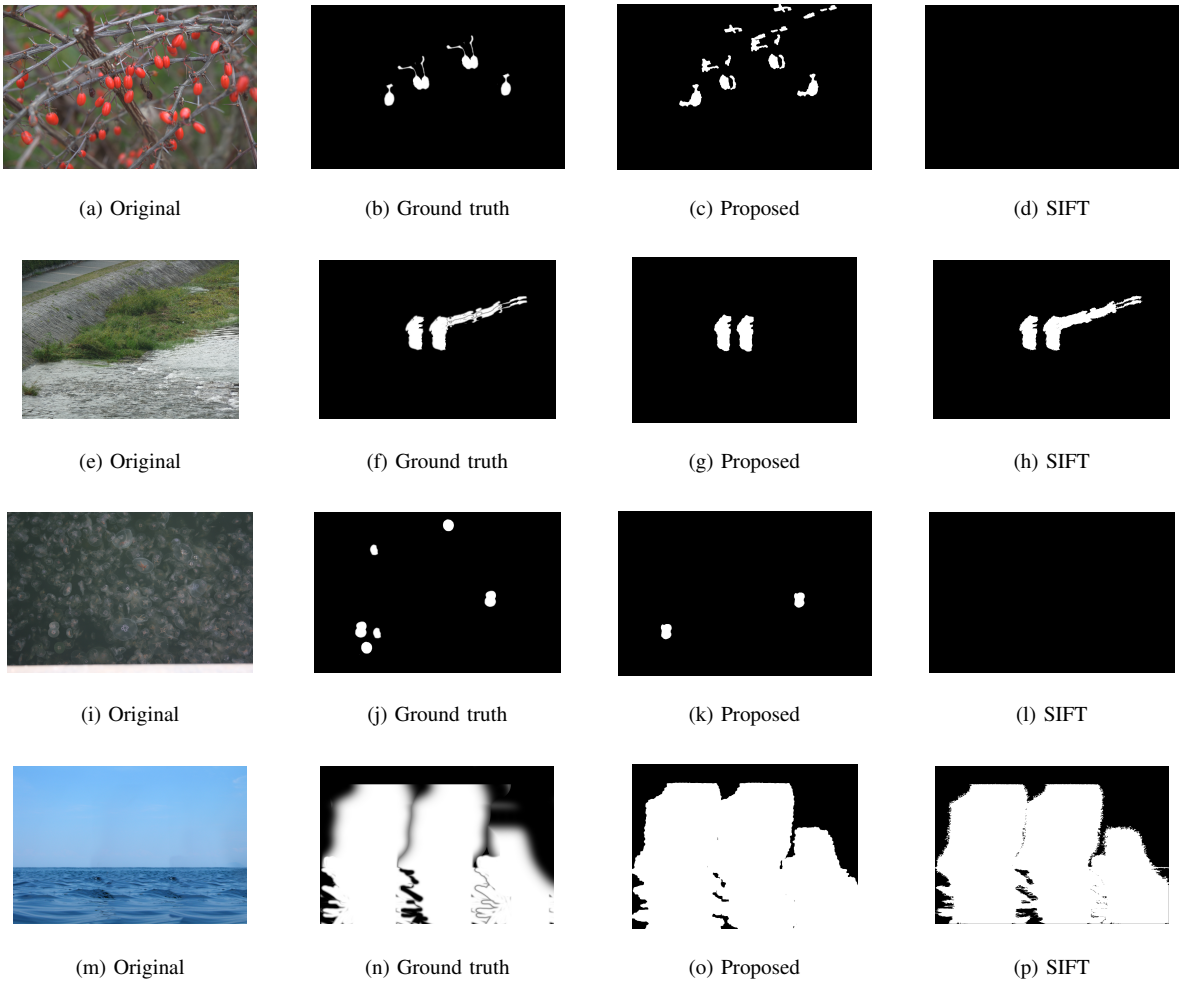
Measures	Amerini *et al.* [14]	Amerini *et al.* [28]	Proposed
F_N	31.0%	18.4%	11.9%
F_P	12.5%	7.27%	13.8%

VI. Conclusion and Discussion

This paper presented a CMFD scheme based on image segmentation. Although the CMF regions are detected mainly by comparing the keypoints extracted in the image, we cannot simply classify the proposed scheme as a keypoint-based one. It can be seen as a combination of both existing schemes because in the two stages of matching process both keypoints and pixel features are employed. Our main contributions can be concluded to the following two aspects.

1) Considering the CMF regions usually have certain meaning, we propose to segment the image into semantically independent patches, such that the CMFD problem can be solved by partial matching among these segmented patches.
2) The matching process between segmented patches consists of two stages. In the second stage, an accurate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security



(a) Original (b) Ground truth (c) Proposed (d) SIFT

(e) Original (f) Ground truth (g) Proposed (h) SIFT

(i) Original (j) Ground truth (k) Proposed (l) SIFT

(m) Original (n) Ground truth (o) Proposed (p) SIFT

Fig. 10: Detection results on the images with CMF regions fused in the background. The first column shows the test images from MICC-600. The second column shows the ground truth of the CMF regions in these images. The third and the fourth columns show the detection results of our proposed scheme and the scheme based on SIFT in [1], respectively.

estimation of transform matrix can be obtained by an EM-based algorithm.

One may concern the computational complexity of the proposed scheme. Compared with the keypoint-based schemes, the proposed scheme mainly needs two more steps, namely the image segmentation and the transform estimation refinement. If using some efficient methods like [32], [34], we are able to segment an image in several seconds. The re-estimation of transform matrix is more complex because it needs an iterative procedure (refer to Section IV-C). However, owing to the threshold set in (1), only a few patches (about one tenth) need the second stage of matching for transform matrix re-estimation. In our future work, we will try to improve the detection speed of the proposed scheme by means of parallel programming.

REFERENCES

[1] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[2] A. J. Fridrich, B. D. Soukal, and A. J. Luk, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensic Research Workshop*, 2003.

[3] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *Proc 18th Int. Conf. Pattern Recognition (ICPR)*, vol. 4, 2006, pp. 746–749.

[4] S. Bayram, H. Taha Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, ser. ICASSP '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1053–1056.

[5] S. Bravo-Solorio and A. K. Nandi, "Exposing duplicated regions affected by reflection, rotation and scaling," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2011, pp. 1880–1883.

[6] M. Ghorbani, M. Firouzmand, and A. Faraahi, "DWT-DCT (qcd) based copy-move image forgery detection," in *Proc. 18th Int. Conf. Syst., Signals and Image Process. (IWSSIP)*, 2011, pp. 1–4.

[7] S. Khan and A. Kulkarni, "Detection of copy-move forgery using multiresolution characteristic of discrete wavelet transform," in *Proc. Int. Conf. & Workshop on Emerging Trends in Technology*, ser. ICWET '11. New York, NY, USA: ACM, 2011, pp. 127–131.

[8] S.-J. Ryu, M. Kirchner, M.-J. Lee, and H.-K. Lee, "Rotation invariant localization of duplicated image regions based on zernike moments," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1355–1370, Aug. 2013.

[9] V. Christlein, C. Riess, and E. Angelopoulou, "On rotation invariance in copy-move forgery detection," in *Proc. IEEE Workshop Int. Inform. Forensics and Security (WIFS)*, 2010, pp. 1–6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2014.2381872, IEEE Transactions on Information Forensics and Security

[10] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using sift algorithm," in *Proc. Pacific-Asia Workshop Computational Intell. and Industrial Applicat. (PACIIA)*, vol. 2, 2008, pp. 272–276.

[11] E. Ardizzone, A. Bruno, and G. Mazzola, "Copy-move forgery detection via texture description," in *Proc. 2nd ACM Workshop Multimedia Forensics, Security and Intell.*, ser. MiFor '10. New York, NY, USA: ACM, 2010, pp. 59–64.

[12] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, "Image copy-move forgery detection based on surf," in *Proc. Int. Conf. Multimedia Inform. Networking and Security (MINES)*, 2010, pp. 889–892.

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

[14] 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, Sept. 2011.

[15] P. Kakar and N. Sudha, "Exposing postprocessed copy–paste forgeries through transform-invariant features," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1018–1028, June 2012.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Comput. Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, June 2008.

[18] Q. Liu, N. Linge, and V. Lynch, "Implementation of automatic gas monitoring in a domestic energy management system," *IEEE Trans. Consum. Electron.*, vol. 58, no. 3, pp. 781–786, Aug. 2012.

[19] Q. Liu, G. Cooper, N. Linge, H. Takruri, and R. Sowden, "Dehems: creating a digital environment for large-scale energy management at homes," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 62–69, Feb. 2013.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *ACM Commun.*, vol. 24, no. 6, pp. 381–395, June 1981.

[21] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2004.

[22] M. Akhin and V. Itsykson, "Clone detection: Why, what and how?" in *Proc. 6th Central and Eastern European Software Eng. Conf. (CEE-SECR)*, 2010, pp. 36–42.

[23] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[24] A. M. Bronstein and M. M. Bronstein, "Regularized partial matching of rigid shapes," in *10th European Conf. Comput. Vision (ECCV)*, A. Z. David Forsyth, Philip Torr, Ed., vol. 5303 2008. Marseille, France: Springer, Oct. 2008, pp. 143–154.

[25] ——, "Not only size matters: regularized partial matching of nonrigid shapes," in *IEEE Comput. Society Conf. Comput. Vision Pattern Recognition Workshop (CVPRW)*. IEEE, 2008, pp. 1–6.

[26] A. M. Bronstein, M. M. Bronstein, Y. Carmon, and R. Kimmel, "Partial similarity of shapes using a statistical significance measure," *IPSJ Trans. Comput. Vision and Applicat.*, vol. 1, no. 0, pp. 105–114, Mar. 2009.

[27] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," ICSI, Tech. Rep. TR-97-021, 1997.

[28] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, L. D. Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with j-linkage," *Signal Process.: Image Commun.*, vol. 28, no. 6, pp. 659 – 669, July 2013.

[29] H. Farid, "Exposing digital forgeries in scientific images," in *Proc. 8th Workshop on Multimedia and Security*, ser. MM&Sec'06. New York, NY, USA: ACM, 2006, pp. 29–36.

[30] B. Liu, C.-M. Pun, and X.-C. Yuan, "Digital image forgery detection using jpeg features and local noise discrepancies," *The Scientific World J.*, vol. 2014, pp. 1–12, Mar. 2014.

[31] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.

[32] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *10th European Conf. Comput. Vision (ECCV)*. Springer, 2008, pp. 705–718.

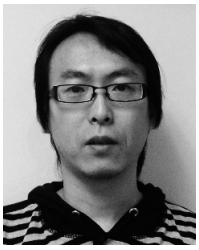
[33] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[34] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[35] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[36] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Int. J. Comput. Vision*, vol. 65, no. 1-2, pp. 43–72, Nov. 2005.

Jian Li is currently a Lecture in the College of Computer and Software at Nanjing University of Information Science & Technology, China. He received the B.S. and M.S. degrees from Shandong University, China, and the Ph. D. degree from Sun Yat-Sen University, China, all in computer science, in 2004, 2007 and 2011 respectively. His research interests include information hiding and forensics.



Xiaolong Li received the B.S. degree from Peking University, Beijing, China, the M.S. degree from Ecole Polytechnique, Palaiseau, France, and the Ph.D. degree in mathematics from ENS de Cachan, Cachan, France, in 1999, 2002, and 2006, respectively. Before joining Peking University as a researcher, he worked as a postdoctoral fellow at Peking University in 2007-2009. His research interests are image processing and information hiding.

Bin Yang received the B.S. and M.S. degrees in computer science from Peking University, Beijing, China, in 1991 and 1994, respectively. Currently, he is a Professor with the Institute of Computer Science and Technology, Peking University. His research interests are image processing and information hiding.

Xingming Sun is currently a Professor in the College of Computer and Software at Nanjing University of Information Science & Technology, China. He received the B.S. degree in mathematics from Hunan Normal University, China, in 1984, the M.S. degree in computing science from Dalian University of Science and Technology, China, in 1988, and the Ph.D. degree in computer science from Fudan University, China, in 2001. His research interests include network and information security, digital watermarking, cloud computing security, and wireless network security.