Region Duplication Detection Using Image Feature Matching

Xunyu Pan, *Student Member, IEEE*, and Siwei Lyu, *Member, IEEE*

***Abstract*—Region duplication is a simple and effective operation to create digital image forgeries, where a continuous portion of pixels in an image, after possible geometrical and illumination adjustments, are copied and pasted to a different location in the same image. Most existing region duplication detection methods are based on directly matching blocks of image pixels or transform coefficients, and are not effective when the duplicated regions have geometrical or illumination distortions. In this work, we describe a new region duplication detection method that is robust to distortions of the duplicated regions. Our method starts by estimating the transform between matched scale invariant feature transform (SIFT) keypoints, which are insensitive to geometrical and illumination distortions, and then finds all pixels within the duplicated regions after discounting the estimated transforms. The proposed method shows effective detection on an automatically synthesized forgery image database with duplicated and distorted regions. We further demonstrate its practical performance with several challenging forgery images created with state-of-the-art tools.**

***Index Terms*—Digital image forensics, image feature matching, region duplication detection.**

I. Introduction

THANKS to the increasing availability and sophistication of digital imaging technology (digital cameras, computers, and photoediting software) and the popularity of the Internet, digital images have become our main information source. However, concomitant with the ubiquity of digital images is the rampant problem of digital forgeries, which has seriously debased the credibility of photographic images as definite records of events. Accordingly, digital image forensics has emerged as a new research field that aims to reveal tampering operations in digital images [13].

A common manipulation in tampering with digital images is known as *region duplication*, where a continuous portion of pixels are copied and pasted to a different location in the same image. To make convincing forgeries, the duplicated regions are often created with geometrical or illumination adjustments.

Fig. 1. Two original digital images and forgery images created based on them using duplicated and distorted regions. The first original image is courtesy of H. Farid, and the second one is from [12].

Fig. 1 exemplifies two main usages of duplicated regions in creating forgery images. For the example shown in the left panel, a rotated duplicated region is used to conceal undesirable contents in the original image. In the other case, two duplicated regions that are scaled, rotated, and mirrored are used to create contents that are not in the original image. These duplicated regions are well blended into the surroundings at the target locations, and become very difficult to detect visually.

In this work, we describe a new method for reliable detection of duplicated and distorted regions in a digital image. Our method is based on image keypoints and feature vectors that are robust to typical image transforms [32]. We formulate region duplication detection as finding transformed identical regions in an image and use robust estimation to obtain correct keypoints matching and transforms between duplicated regions simultaneously. With the estimated transforms, our method further obtains the precise location and extent of the detected duplicated regions. Our method is tested with a comprehensive quantitative performance evaluation on a database of automatically generated forgery images with duplicated and distorted regions. We also report its robustness with regards to different JPEG qualities and additive noise levels. We further demonstrate the effectiveness of our method on several challenging forgery images generated using state-of-the-art image editing tools.

Manuscript received March 18, 2010; revised September 05, 2010; accepted September 06, 2010. Date of publication September 23, 2010; date of current version November 17, 2010. This work was supported by an NSF CAREER Award (IIS09-53373) and by the University at Albany Faculty Research Awards Program (FRAP)-Category A. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Min Wu.

The authors are with the Computer Science Department, State University of New York at Albany, Albany, NY 12222 USA (e-mail: xypan@cs.albany.edu; lsw@cs.albany.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2010.2078506

II. Related Work

Several general techniques in digital image forensics may be applied to detect duplicated regions. For JPEG images, a double JPEG quantization is usually a telltale sign of tampering operations (including region duplication), and can be detected based on the histograms of quantized DCT coefficients [20], [42]. Other intrinsic cues from the imaging process, including color filter array interpolation patterns [44], [46], camera response functions [30], and camera sensor noise [9], [18], or general statistical properties of untampered natural images [34], [35] can also be used to reveal image tampering. However, these general detection methods typically do not provide direct evidence of region duplication or the location of the duplicated regions. Capturing statistical correlation of interpolation [43] or inconsistency in the lighting geometry [23] may be used to locate duplicated regions. But these methods only work reliably with high-quality uncompressed images, and become ineffective in practical scenarios.

One simple approach to locate duplicated regions in an image is to identify off-origin peaks in the image auto-correlation function, which can be computed efficiently using the fast Fourier transform [17]. Due to its running efficiency and simplicity, it has been used for the detection of duplicated regions in videos [47]. However, identifying off-origin peaks in the auto-correlation function becomes difficult when the duplicated region is relatively small, or the image contains noise or other artifacts. Many other existing region duplication detection methods are based on matching blocks of image pixels or transform coefficients (e.g., [4], [15], [24], [26], [28], [33], [36], [41], [48]). While these methods can detect duplicated regions pasted to the target location without any change (a special case known as *copy-move*), they are largely ineffective to detect duplicated regions that are also distorted (such as examples in Fig. 1). To alleviate this problem, a variant of the block matching region duplication method is proposed to handle duplicated regions rotated with 90°, 180°, and 270° [29]. Another vein of works use blocks in the log-polar coordinate system [5], [8], [38], where rotation and scaling become translation and can be detected as copy-move. Another method has been proposed to detect duplicated regions with smoothing operation [27]. However, the flexibilities provided by this method are limited and they cannot be extended for the detection of duplicated regions with general distortions.

As an alternative to the block-matching-based detection methods, several recent methods have explored the use of matched image keypoints to identify duplicated regions. In [21], keypoints and features based on the *scale invariant feature transform* (SIFT) algorithm [32] are used to account for illumination changes in the detection of copy-move region duplication. However, the robustness of SIFT keypoints and features to image distortions is not fully exploited, which prevents this method from being extended to detect affine transformed duplicated regions. In our previous work [39], we describe an SIFT-matching-based detection method that can locate duplicated regions with rotation or scaling. Another recent work [3] uses SIFT keypoint matching to estimate the parameters of the affine transform and recover matched keypoints. But similar to [21], it does not provide the exact extent and location of the detected duplicated region, but only displays the matched keypoints. Furthermore, these detection methods are typically evaluated against simple forgeries where human viewers have no trouble to identify the duplicated regions, and their performance on challenging realistic forgery images is largely unknown.

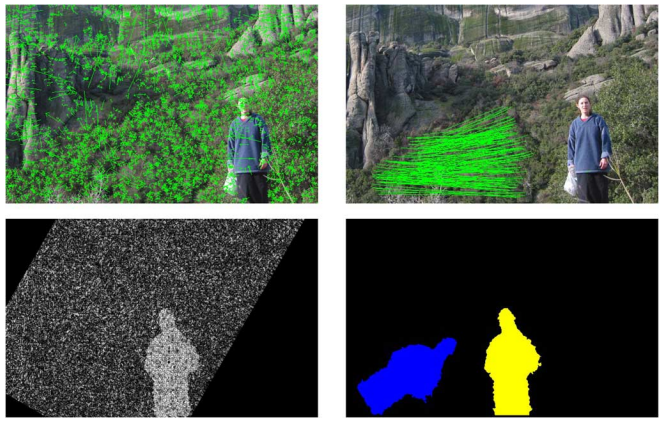

Fig. 2. Main steps of the proposed method to detect duplicated and distorted regions. (top left) Detected SIFT keypoints in an image. (top right) Matched keypoints after the RANSAC algorithm. (bottom left) Region correlation map generated with the estimated affine transforms. Brighter pixel intensity signifies stronger correlation. (bottom right) Detected duplicated regions.

III. Method

In this section, we describe in detail the proposed method to detect duplicated and distorted regions in an image, with Fig. 2 illustrating the main steps of our method using the forgery image shown in Fig. 1.

A. Finding Image Keypoints and Collecting Features

We detect duplicated regions in the illumination domain, so RGB images are first converted to grayscale images using standard color space conversion. The first step in our method is to find image keypoints and collect image features at the detected keypoints.

Keypoints are locations carrying distinct information of the image content. Each keypoint is characterized by a feature vector that consists of a set of image statistics collected at the local neighborhood of the corresponding keypoint. Good keypoints and features should represent distinct locations in an image, be efficient to compute and robust to local geometrical distortion, illumination variations, noise, and other degradations.

Our method is based on an effective keypoint and feature computation algorithm known as the SIFT [32]. SIFT keypoints are found by searching for locations that are stable local extrema in the scale space [31]. At each keypoint, a 128-dimensional feature vector is generated from the histograms of local gradients in its neighborhood. To ensure the obtained feature vector invariant to rotation and scaling, the size of the neighborhood is determined by the dominant scale of the keypoint, and all gradients within are aligned with the keypoint's dominant orientation. Furthermore, the obtained histograms are normalized to

unit length, which renders the feature vector invariant to local illumination changes.

As duplicated regions typically account for only a small fraction of the total area of the image, we limit keypoint detection to a small range of scales. In our experiment, we construct the scale space with Gaussian kernels of initial width of 1.6 pixels up to 7 octaves. The top left panel in Fig. 2 shows the SIFT keypoints detected in an image. The end of each arrow corresponds to the location of one SIFT keypoint. The directions of the arrows show the dominant orientation of each keypoint, and the lengths of the arrows correspond to the dominant scale.

B. Putative Keypoint Matching

The detected SIFT keypoints are then tentatively matched based on their feature vectors using the *best-bin-first* algorithm [6]. For a keypoint at location $\mathbf{x}$ with feature $\mathbf{f}$, we match it with keypoint $\tilde{\mathbf{x}}$, whose corresponding feature vector $\tilde{\mathbf{f}}$ is the nearest neighbor to $\mathbf{f}$ measured with their l_2 (Euclidean) distance. Due to the smoothness of natural images, the best match of a keypoint usually lies within its close spatial adjacency. To avoid searching nearest neighbors of a keypoint from the same region, we perform the search outside an 11×11 pixel window centered at the keypoint. Further, many keypoints can match with each other, but we only keep those with distinct similarities. Specifically, we require that for any other feature vector $\mathbf{f}'$ other than $\mathbf{f}$ and $\tilde{\mathbf{f}}$, the distance between $\mathbf{f}$ and $\tilde{\mathbf{f}}$ has to be smaller than that of $\mathbf{f}$ and $\mathbf{f}'$ by at least a factor of ϵ, as $\|\tilde{\mathbf{f}} - \mathbf{f}\|_2 < \epsilon \|\mathbf{f}' - \mathbf{f}\|_2$, where $\epsilon \in (0, 1)$ is a preset threshold controlling the distinctiveness of the matching. We use a default $\epsilon = 0.5$ to provide a good trade-off between matching accuracy and ratio of outliers.

C. Estimation of Affine Transform Between Matched Keypoints

Next, based on the putative keypoint matching, we estimate the possible geometric distortions of the duplicated regions. To generalize transforms such as rotation, scaling, and shearing that are supported in most photoediting software, we model the distortion as affine transform of pixel coordinates. Given two corresponding pixel locations from a region and its duplicate as $\mathbf{x} = (x, y)^T$ and $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})^T$, respectively, they are related by a 2-D affine transform specified by a 2×2 matrix T and a shift vector $\mathbf{x}_0$ as $\tilde{\mathbf{x}} = T\mathbf{x} + \mathbf{x}_0$, or more explicitly

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}. \tag{1}$$

To obtain a unique solution to the transform parameters, $(t_{11}, t_{12}, t_{21}, t_{22}, x_0, y_0)$, we need at least three pairs of corresponding keypoints that are not *collinear*. In practice, due to imprecise matching, (1) may not be satisfied exactly, and we form the *least squares* objective function using matched keypoints $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$ and $(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_n)$, as

$$L(T, \mathbf{x}_0) = \sum_{i=1}^{N} \|\tilde{\mathbf{x}}_i - T\mathbf{x}_i - \mathbf{x}_0\|_2^2 \tag{2}$$

and searching for T and $\mathbf{x}_0$ that minimize it. The optimal solution is provided and derived in the Appendix.[1]

D. Robust Estimation of Affine Transform

We can use the putative matchings of SIFT keypoints to estimate the affine transform parameters, but the obtained results are inaccurate due to the large number of mismatched keypoints. To prune out unreliable keypoint correspondences and obtain accurate transform parameters simultaneously, we employ a widely used robust estimation method known as the *Random Sample Consensus* (RANSAC) algorithm [14]. The RANSAC algorithm can estimate the model parameters with a high degree of accuracy even when a significant number of mismatched pairs are present. Using the putative matching of SIFT keypoints, we run the following two steps N times:

1) Randomly select three or more pairs of matched keypoints that are not collinear. Using the chosen pairs of keypoints, estimate T and shift vector $\mathbf{x}_0$ by minimizing the objective function given in (2).
2) Using the estimated T and $\mathbf{x}_0$, classify all pairs of matched SIFT keypoints into *inliers* or *outliers*. Specifically, a pair of matched keypoints $(\mathbf{x}, \tilde{\mathbf{x}})$ is an inlier if $\|\tilde{\mathbf{x}} - T\mathbf{x} - \mathbf{x}_0\|_2 \leq \beta$, otherwise, it is an outlier.

The RANSAC algorithm returns with the estimated transform parameters that lead to the largest number of inliers. In our experiment, we choose default values for $N = 100$ and $\beta = 3$ as they lead to better empirical performance. The top right panel in Fig. 2 shows the SIFT keypoint correspondences after pruning with the RANSAC algorithm.

E. Region Correlation Map

With the estimated affine transform, we compare each pixel to its transformation to find identical regions. In practice, because the estimated affine transform can be the inverse of the actual transform (from pixel level, we cannot differentiate which region is the source and which one is the duplicate), we check the correspondence of $\mathbf{x}$ using both the estimated affine transform, $\mathbf{x}_f = T\mathbf{x} + \mathbf{x}_0$ and its inverse, $\mathbf{x}_b = T^{-1}(\mathbf{x} - \mathbf{x}_0)$. Taking the forward transform as example, the similarity between $\mathbf{x}$ and $\mathbf{x}_f$ is evaluated with the *correlation coefficients* between the pixel intensities within small neighboring areas of each location. Denote the pixel intensity at location $\mathbf{x}$ as $I(\mathbf{x})$, and $\Omega(\mathbf{x})$ as the 5×5 pixels neighboring area centered at $\mathbf{x}$, the correlation coefficient between the two pixel locations is computed as

$$c_f(\mathbf{x}) = \frac{\sum_{\mathbf{s} \in \Omega(\mathbf{x}), \mathbf{t} \in \Omega(\mathbf{x}_f)} I(\mathbf{s}) I(\mathbf{t})}{\left[\sum_{\mathbf{s} \in \Omega(\mathbf{x})} I(\mathbf{s})^2\right] \left[\sum_{\mathbf{t} \in \Omega(\mathbf{x}_f)} I(\mathbf{t})^2\right]}.$$

The correlation coefficient $c_b(\mathbf{x})$ for the inverse transformed $\mathbf{x}_b$ is computed in the same manner. The correlation coefficient is in the range of [0, 1], with larger value indicating higher level of similarity. Further, it is invariant to local illumination distortions—any illumination changes consistent within the local

[1]This method can be further generalized to the perspective projection transform, in which we use a general 3×3 homographic matrix in lieu of T, and obtain least squares solutions in a similar fashion [19].

neighborhood will cancel out each other. The computed correlation coefficients $c_f(\mathbf{x})$ and $c_b(\mathbf{x})$ are put into two correlation maps, one example of which is shown in the bottom left panel of Fig. 2.

F. Locating Duplicated Regions

The final step of our method is to process the region correlation maps to obtain the duplicated regions. First, we apply a Gaussian filter of size 7×7 pixels to reduce the noise in the correlation maps. Next, with a threshold $c \in [0, 1]$, the correlation maps are discretized to binary images. A higher c value close to 1 can single out regions that are strongly similar, but can miss detections of duplicated regions that give rise to weaker correlations. On the other hand, a lower c value allows better detection accuracy of duplicated regions, but may increase false detections by including untampered regions that have medium range correlations. In our experiments we choose a default value of $c = 0.3$ for a good trade-off between detection accuracy and false detection rate. The obtained binary maps for c_f and c_b are then combined into a single map by a union of the binary values. Next, we use an area threshold, $A = 0.1\%$ of the total area of the image to remove small isolated regions. By doing so, we assume the minimum duplicated region that can be detected is about 23×23 pixels in size for an image of 800×600 pixels. As a final postprocessing step, we use mathematical morphological operations [45] to smooth and connect the boundaries of the detected duplicated regions. Shown in the bottom right panel of Fig. 2 are the final detected regions.

G. Handling Copy-Move

If the estimated transform matrix T is close to identity, the distortion of the duplicated region is close to a translation, corresponding to a copy-move duplication. In this case, we employ a more efficient and robust method to recover the shift vectors directly. Specifically, we compute the l_2 distances between the location of each pair of matched SIFT keypoints. Though corresponding keypoints in the source and duplicated regions should have the same l_2 distance, due to mismatch, the observed values have a wider range. So we build a histogram of l_2 distances between all pairs of keypoints, and collect keypoint pairs with the distance of maximum frequency of occurrence. After grouping these keypoints into two groups using the k-means clustering, the shift vector is estimated as the difference between the means of these two groups.

H. Handling Reflection

Duplicated regions that are reflected need special treatments. A reflection over a line is a transformation in which each point is mapped to another point that is the same distance from the line of reflection as the original point but is on the opposite side. It can be shown that reflection is a special case of the affine transform. However, SIFT features are not invariant to reflection, as the mirrored keypoint has different dominant orientation. Specifically, we handle reflected duplicated regions by searching corresponding SIFT keypoints in one image, as well as its mirrored version reflected around one image coordinate axis with regards to the geometric center of the image plane. The RANSAC procedure is then run to obtain SIFT keypoint correspondences, and detected matchings in the mirrored image are mapped back to locations in the original image coordinates.

TABLE I
DEFAULT PARAMETER VALUES IN OUR METHOD

$\epsilon = 0.5$	threshold in keypoint matching (Section III-B)
$\beta = 3$	maximum distance of inliers in RANSAC (Section III-D)
$N = 100$	number of RANSAC iterations (Section III-D)
$c = 0.3$	threshold of correlation map (Section III-E)
$A = 0.1\%$	area threshold (in fraction of total image area) for duplicated regions (Section III-E)

I. Detecting Multiple Duplicated Regions

For forgery images containing more than one pair of duplicated regions, we run our detection method iteratively with each iteration selecting one pair of potential duplicated regions. After a pair of duplicated regions are identified, we rerun our detection algorithm, this time with pixels in one of the detected duplicated regions masked out from the search. In other words, any SIFT keypoints from regions that have been detected as duplications are excluded from the next round of detection. This allows more significant duplications to be detected first, followed by ones with smaller area. The whole algorithm stops when there is no duplicated region larger than the area threshold found in the image. As the last step, all recovered duplicated regions are combined together and mapped back to the original image coordinates.

J. Implementation

Our method is implemented using the C language based on the OpenCV platform [7]. On a machine with an Intel Core 2 Duo 3.0-GHz E8400 processor and 4-GB memory, the average time to analyze an 800×600 image is about 10 s. We list the default values of the adjustable parameters in our method in Table I. Unless specified otherwise, the results reported in the following sections are with the default parameters.

IV. EXPERIMENTAL EVALUATIONS

In this section, we evaluate the quantitative performance of the proposed region duplication detection method on a set of automatically generated forgery images with duplicated and distorted regions.

A. Automatically Synthesized Forgery Images

Forgery images are generated based on 25 uncompressed PNG true color images of size 768×512 pixels released by the Kodak Corporation for unrestricted research usage.[2] For each untampered image, randomly chosen square regions that contain more than 50 SIFT keypoints are used for duplication. To test the effect of the sizes of the duplicated regions on detection, we use three different block sizes (32×32, 64×64, and 96×96 pixels) corresponding to 0.26%, 1.04%, and 2.34% of total image area, respectively.

We then distort the duplicated regions using several types of transforms, with the shift vector chosen randomly in the same image. These affine transforms are representatives of the most

[2]Image source: http://r0k.us/graphics/kodak/.

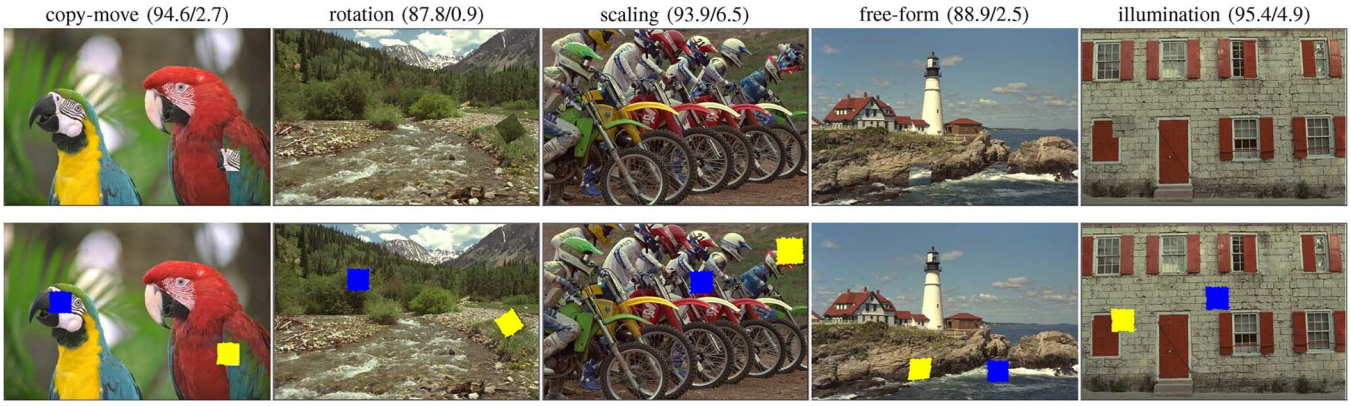

Fig. 3. (Top) Five examples of synthesized forgery images used in our experiments, corresponding to copy-move, rotation, scaling, random free-form linear transform, and illumination distortion to a duplicated region of size 64 × 64 pixels. (Bottom) Detection results using our method. In the parentheses are the PDA/PFP rates of the detected duplicated region in percentage.

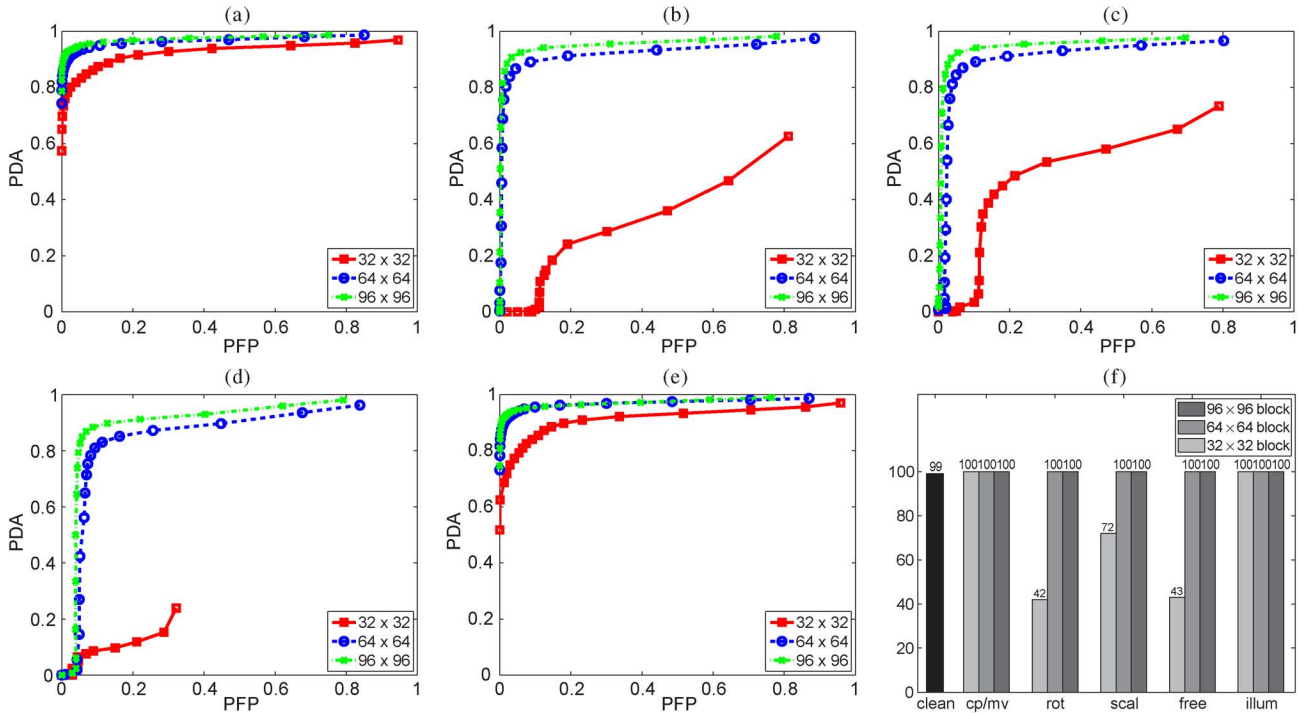

Fig. 4. (a)–(e) ROC curves for different tampering operations and block sizes. Results are averaged over 100 randomly synthesized forgeries with region duplications. (f) Overall image level performance of our method. For clean images, we report the true negative rates. For forgery images of different duplicated region sizes and operations, we report the detection accuracies. Both rates are given in percentage. (a) Copy-move; (b) rotation; (c) scaling; (d) free-form; (e) illumination; (f) image level detection.

frequent manipulations in creating region duplication forgeries provided in most photoediting tools, and are defined with regards to the local coordinate system originating at the geometric center of the corresponding source regions, and implemented with bicubic interpolations using MATLAB.

1) **Copy-move**: The duplicated region is translated to the target location with no distortion.
2) **Rotation**: The duplicated region is rotated with a random angle $\theta \in [0°, 360°)$.
3) **Scaling**: The duplicated region is scaled up or down with a random scaling factor $s \in [0.8, 2.0]$.
4) **Free-form**: The duplicated region is distorted with a linear transform of a random affine matrix T.
5) **Illumination adjustment**: The affine transform is the same as in the case of copy-move, but all pixels in the duplicated region have intensities modulated to 80% of their original values.

For each of the five types of region duplication and size of duplicated regions, we generate four tampered images using each of the 25 images in the Kodak database, resulting in a total 1500 forgery images, five examples of which are shown in the top rows of Fig. 3.

B. Performance Analysis With ROC Curves of PDA/PFP Rates

We use two quantitative measures to evaluate the performance of our method. Denote Ω as pixels in the true duplicated

regions (both the source and its duplicates), and $\tilde{\Omega}$ as pixels in the detected duplicated regions; we define the *pixel detection accuracy* (PDA) rate as the fraction of pixels in duplicated regions that are correctly identified, i.e., PDA $= |\tilde{\Omega} \cap \Omega|/|\Omega|$ and the *pixel false positive* (PFP) rate as the fraction of pixels in untampered regions that are detected as from duplicated regions, i.e., PFP $= |\tilde{\Omega} - \Omega|/|\tilde{\Omega}|$.

Combining the PDA and PFP rates in a *receiver-operator characteristics* (ROC) curve [11] provides a comprehensive evaluation of the detection performance. In our method, different PDA/PFP rates are obtained by adjusting the correlation threshold c (Section III-E) in the range of 0.00 to 0.95 with step size 0.05. To reduce the effect of random samples, each pair of PDA/PFP rates is computed as the averages over all 100 forgery images of each distortion and block size. The resulting ROC curves for each type of distortion and size of the duplicated regions are shown in panels (a)–(f) of Fig. 4.

As these ROC curves show, for block sizes of 64×64 and 96×96, with a PFP rate around 5%, our method achieves a PDA rate greater than 85%. Such a level of PDA/PFP rates are usually sufficient to identify the duplicated regions visually, confirmed with the bottom row of Fig. 3 showing detection results with the corresponding PDA/PFP rates in this range. On the other hand, there is a clear effect of the size of the duplicated blocks to the detection performance, the larger the block size is, the better the overall performance (as the area under the ROC curve is larger). This is expected, as larger duplicated regions include more SIFT keypoints, which makes the matching and transform estimation more reliable. Also, there is a difference in performance for different types of distortion. Especially, the simple case of copy-move and illumination distortion are the easiest to detect, while the most difficult is when the duplicated regions are subject to free-form affine transforms.

For the purpose of comparison, we also implement and apply two previous region duplication detection methods, [41] and [21],[3] to the set of automatically synthesized forgery images. For copy-move forgeries, both methods achieve similar performances to our method, as they are designed for such types of forgeries. On the other hand, for other types of distorted duplicated regions, both methods practically fail to detect any forgery image, reflected by close to zero areas under the ROC curves.

C. Image Level Detection Performance

Area based performance measures such as PDA/PFP rates are useful when we know that the tested image is a forgery, i.e., $\Omega \neq \emptyset$. Yet, in practice, this is usually not known *a priori*. In the next set of experiments, we test the overall image-level detection performance of our method. Specifically, for a forgery image, a successful detection is deemed when our method detects a duplicated region larger than the area threshold A. For an untampered image, a true negative occurs when our method does not detect any duplicated region. Panel (f) of Fig. 4 shows the overall detection performance of our method on the 1500 synthesized forgery images, measured by the rate of successful detections. Also shown is the true negative rate for the 325 untampered images (300 images are collected from Berkeley segmentation dataset [37] and 25 images are collected from Kodak dataset). We use the default parameter values given in Table I.

As these results show, our method correctly identifies all forgery images with duplicated blocks of size 64×64 or 96×96 pixels, and the average PDA/PFP rates for these detections are 83.5% and 8.8%, respectively. The detection performance for duplicated blocks of size 32×32 pixels are significantly inferior, which corroborates the ROC curves in Fig. 4. For untampered images, our method achieves high true negative rates (on average 99.08%), as for most untampered images, intrinsic similar regions (e.g., textures and other repetitive structures) usually lack the high similarity resulted from identical duplicates. This in turn leads to insufficient SIFT keypoint matchings. The threshold in selecting nearest neighbors, the RANSAC matching and threshold on region area then work together to prevent significant false positives in an unmodified image.

D. Robustness to Image Degradations

Our next experiment addresses the robustness and sensitivity of our method with different JPEG qualities, and noise levels. Shown in Fig. 5 are the ROC curves of PDA/PFP for the detection of duplicated regions with rotation, free-form affine transform, and illumination distortion under different JPEG compression qualities (top row) and signal-to-noise ratios (SNRs) of additive white Gaussian noises (bottom row). The forgery images are created with duplicated regions of size 64×64 pixels, and the distortions of these duplicated regions are applied as in the previous experiment. The forgery images are then converted to JPEG images ($Q = 50, 60, 70, 80, 90$) or contaminated with additive white Gaussian noise (SNR $= 20, 25, 30, 35, 40$ dB).

As shown in the ROC curves in Fig. 5, the overall detection performance of our method is relatively robust to these degradations. Even with low image qualities (JPEG $Q = 50$ or SNR $= 20$), in most of the cases, more than 70% of the pixels in duplicated regions can still be detected with less than 20% of PFP rates. In general, the performance tends to decrease for lower image quality. The main reason is that artifacts such as the "blockiness" in low-quality JPEG compression or high level noise interfere with the SIFT algorithm in detecting keypoints. As less reliable keypoints are available in such cases, the detection performance is strongly affected.

E. Transform Parameter Estimation

Last, we evaluate the accuracy of the estimated affine transform using the RANSAC estimation (Section III-D). Using a set of 100 forgery images with randomly chosen affine transformed duplicated regions for each block size, we evaluate the relative estimation errors of each affine transform parameter, which is the ratio of estimation error (the difference between the estimated value and the true value) and the truth parameter value. Shown in Table II are the means and standard deviations (in parenthesis) of the computed relative estimation errors. Note that these errors are relatively small for larger block sizes, but become significant when the block size is 32×32. This is consistent with the results reflected by the ROC curves of PDA/PFP

[3]The original method in [21] only gives matched keypoints. To compare with our method, we extended it in a similar fashion as in Section III-E to return location and extent of detected duplicated regions.

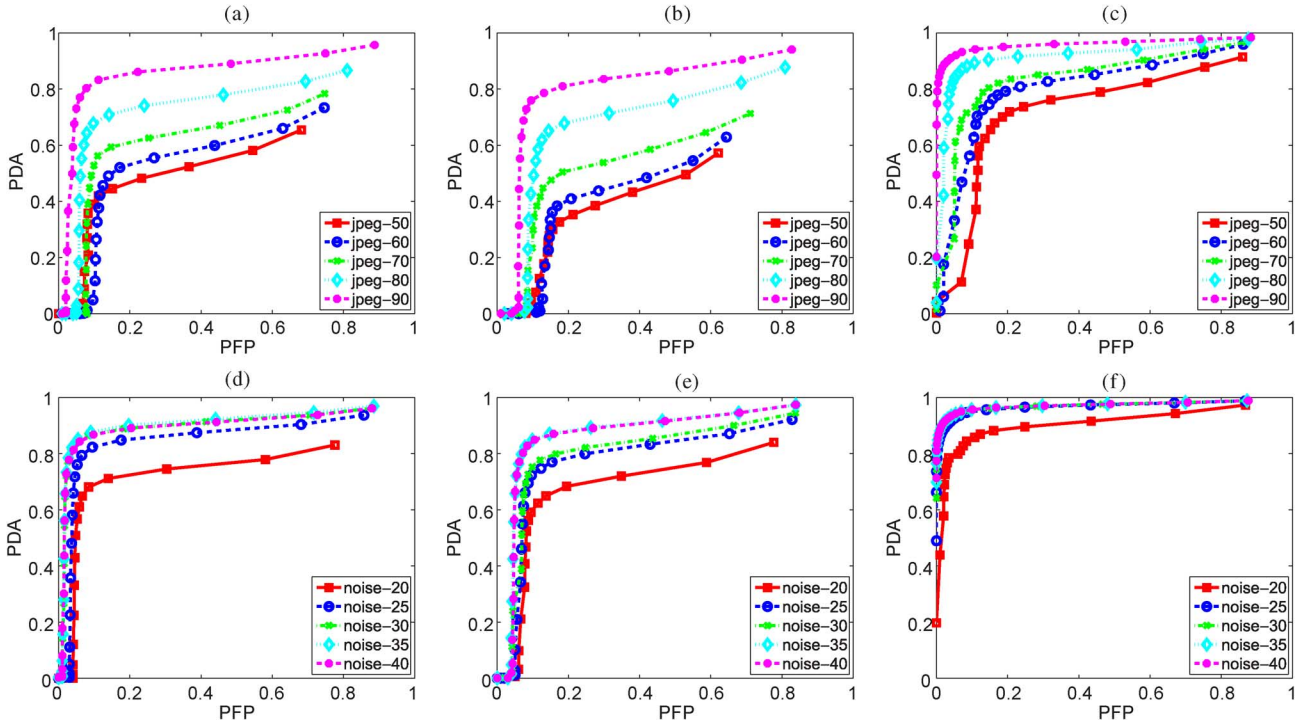

Fig. 5. ROC curves of PDA and PFP rates for different JPEG (top row) and SNRs (bottom row) of region duplications with (a), (d) rotation, (b), (e) free-form transform, and (c), (f) illumination distortion.

TABLE II
MEANS AND STANDARD DEVIATIONS (IN PARENTHESIS) OF THE RELATIVE ERRORS OF THE RANSAC ESTIMATED PARAMETERS FOR THE FREE-FORM LINEAR TRANSFORMED DUPLICATED REGIONS

	matrix T				shift vector $\mathbf{x}_0$	
	t_{11}	t_{12}	t_{21}	t_{22}	x_0	y_0
32×32	-27.7% (37.2%)	11.3% (45.4%)	-26.3% (48.6%)	-21.9% (35.2%)	6.1% (44.8%)	3.9% (20.7%)
64×64	-5.7% (8.3%)	-3.2% (7.3%)	-4.6% (11.2%)	-7.9% (9.5%)	0.4% (1.0%)	0.1% (1.4%)
96×96	-0.2% (4.8%)	-0.2% (4.9%)	0.1% (4.7%)	-0.1% (5.9%)	0.1% (1.3%)	0.1% (0.8%)

rates (Fig. 5), as well as the overall image level detection performance (Fig. 3). This experiment also sheds some light on the relatively low performance for duplicated regions of 32×32 pixels: the smaller number of SIFT keypoints leads to less accurate estimation of the transform parameters, which in turn affects the overall detection performance.

V. REALISTIC DETECTIONS

With the aid of sophisticated photoediting software, such as the *healing brush* in Photoshop [16] and the *smart fill* tool in Image Doctor [2], forgery images can be made with convincing visual appearance using duplicated and distorted regions. And seamless region splicing is an active developments in computer graphics, e.g., [1], [12], [22], and [40]. In this section, we test our region duplication detection method with several convincing digital forgeries made with state-of-the-art image retouching algorithms and tools. All the detection results are made with the default parameter values in Table I.

First, based on the untampered image shown in Fig. 1, we create a set of forgeries of convincing effects using Photoshop. Irregular regions are chosen, distorted with rotation, scaling, free-form affine transform, perspective projection, reflection or illumination adjustment, and then pasted to target locations. The created forgeries, along with the detection results of our method, are shown in Fig. 6. As the visual results and the accompanying PDA/PFP rates show, our method can reliably detect these duplicated regions.

Shown in Fig. 7 are the detection results of our method on some more realistic forgeries. Forgery images in the first two rows are generated with the splicing algorithm developed by Zeev *et al.* [12], which creates natural transition between duplicated region and the surroundings at the target location. This is achieved using numerical solutions to the Laplacian equation defined by the boundary conditions specified by the borders of the duplicated region and its target location, which results in highly convincing tampering results [12]. The duplicated regions in these examples are affine transformed. In the "deer" image, the duplicated region is rotated, scaled, and mirrored; in the "cherry" image, the duplicated region is rotated and overlapped with the source region. Nevertheless, our detection method is able to recover the duplicated regions, and correctly accounts for the geometrical distortions, using the procedure given in Section III-I to detect multiple duplicated regions.

The third row of Fig. 7 shows a forgery created with the *Smart Fill* tool in the Image Doctor software (Alien Skin Software, [2]). The unpublished algorithm used to create this forgery is

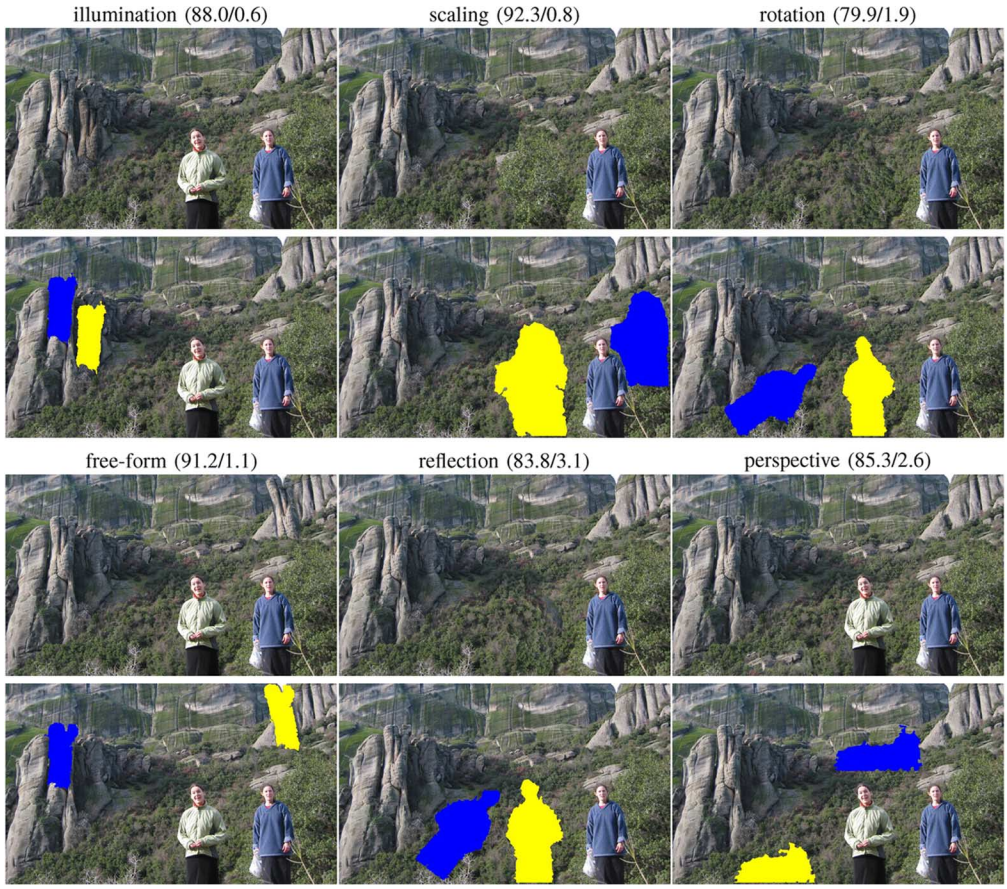

Fig. 6. Detection results (bottom) of our method on a set of forgery images with duplicated regions undergone different types of distortion (top). These images are manually created with the aid of Adobe Photoshop. In the parentheses are the PDA/PFP rates of the detected duplicated region in percentage.

more sophisticated: instead of using a continuous duplicated region of relatively large size, smaller regions containing mostly textures (sometimes the selected region is less than 20 pixels in size) are combined and arranged to cover a larger region at the target location. This makes the visual detection of the duplicated regions significantly more difficult. It also poses as a challenge to our method, especially due to the fact that the smaller identical regions provide less reliable keypoint matchings. However, the detection result of our method as shown in the right column can still provide a considerable clue to draw an inspector's attention for scrutiny.

The last row of Fig. 7 shows the detection result of our method on an alleged forgery image that has recently raised the public's attention. The image shown in the middle panel appeared on the front pages of several internationally important newspapers including *The Los Angeles Times*, *The Financial Times*, and *The Chicago Tribune* and several major news web sites in 2009. Shortly after this image was published, doubts were raised that it had been digitally altered, a fact later confirmed by inspection of photography experts and the appearance of another photograph that was believed to be taken at about the same time (left). Consistent with the analysis of photography experts, our method is able to recover the two major regions that are believed to be duplicated from other parts of the image.

VI. Discussion

Forgery images created with duplicated and distorted regions are challenging to detect visually. In this work, we describe a new method to detect duplicated and distorted regions based on the robust matching of image keypoints and features. We demonstrate the effectiveness of our detection method with a series of experiments, including several sets of highly convincing forgery images created with state-of-the-art image editing tools.

Though having achieved promising performance in detecting sophisticated forgeries with duplicated regions, our method relies on the detection of reliable SIFT keypoints. For some images this may be a limitation. One example is shown as the top left image in Fig. 8, where an obvious duplicated region is not detected by our method. This is because the SIFT algorithm cannot find reliable keypoints in regions with little visual structures. Similarly, as smaller regions have fewer keypoints, they are also hard to detect with our method. Second, there are images that have intrinsically identical areas that cannot be differentiated from intentionally inserted duplicated regions by our method. The three images in Fig. 8 exemplify such cases. As an important future work, we will consider several approaches to improve the detection performance for such cases, including incorporating other features such as PCA-SIFT [25] or histograms

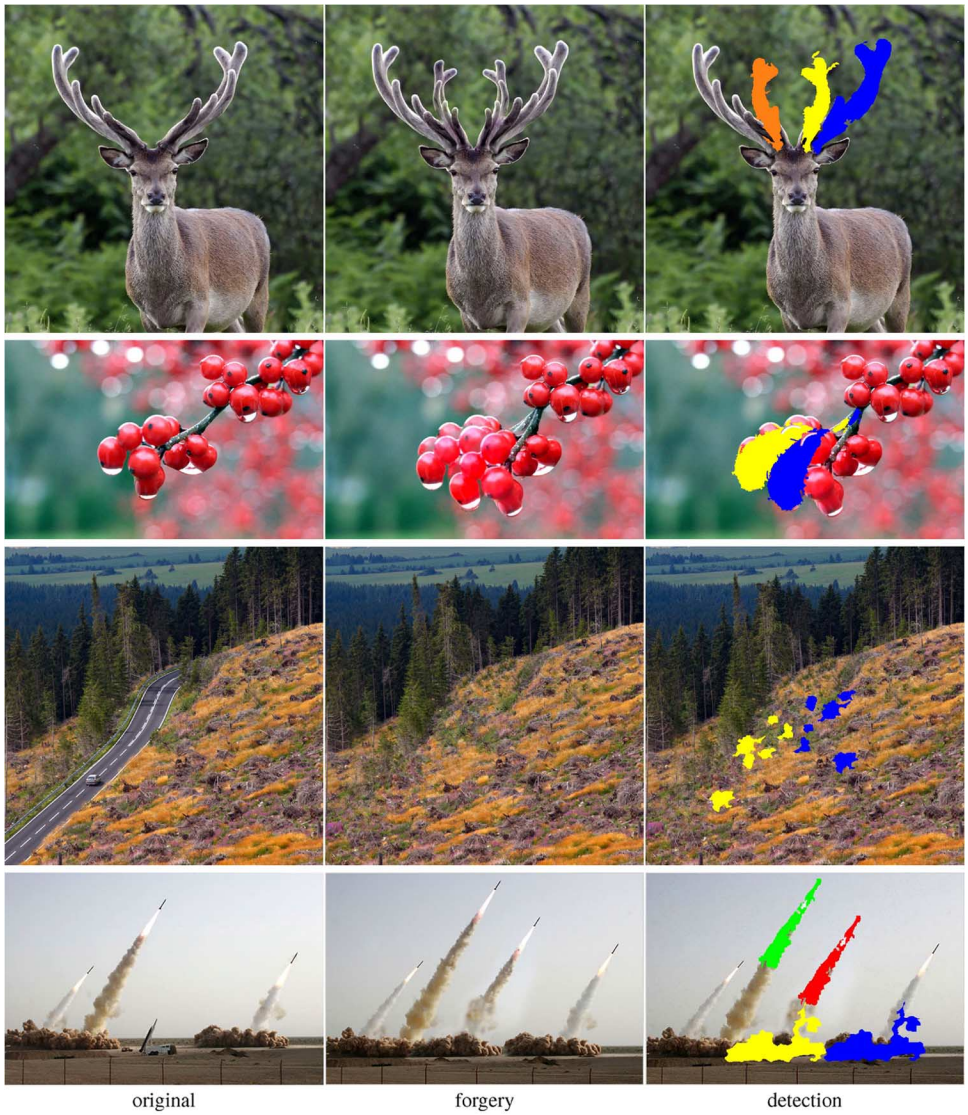

Fig. 7. Detection results of our method on a set of challenging and realistic forgery images with duplicated regions. See text for details.

of oriented gradients [10], and combining with other detection schemes based on intrinsic signal statistics/patterns to provide strong cues when image keypoints and features are not sufficient.

APPENDIX

We provide the detailed derivation of the least squared estimation of the affine transform parameters using matched keypoint pairs in Section III-C. First, taking derivative of the objective function in (2) with regards to $\mathbf{x}_0$, we have

$$\frac{\partial}{\partial \mathbf{x}_0} L(T, \mathbf{x}_0) = \frac{\partial}{\partial \mathbf{x}_0} \sum_{i=1}^{N} \|\tilde{\mathbf{x}}_i - T\mathbf{x}_i - \mathbf{x}_0\|_2^2$$
$$\propto N\mathbf{x}_0 - \sum_{i=1}^{N} (\tilde{\mathbf{x}}_i - T\mathbf{x}_i).$$

Setting this to zero, we obtain $\mathbf{x}_0 = (1/N)\sum_{i=1}^{N}(\tilde{\mathbf{x}}_i - T\mathbf{x}_i) = \tilde{\mu} - T\mu$, where μ and $\tilde{\mu}$ are the mean vectors for

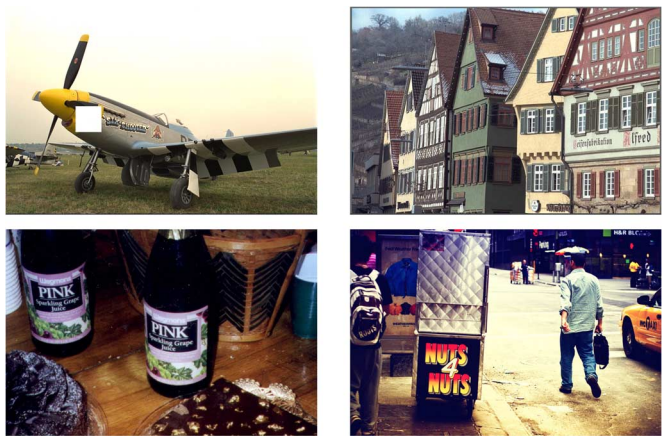

Fig. 8. Top left image is a forgery whose duplicated region is not detected by our method due to the lack of reliable keypoints. The other three images are untampered, but the intrinsic repetitive patterns are regarded as duplicated regions by our method.

$(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$ and $(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_N)$. Furthermore, using the above result, we can further reduce the objective function to

$$L(T) = \sum_{i=1}^{N} ||\tilde{\mathbf{x}}_i - T\mathbf{x}_i - (\tilde{\mu} - T\mu)||_2^2 = \sum_{i=1}^{N} ||\tilde{\mathbf{y}}_i - T\mathbf{y}_i||_2^2$$

where $\mathbf{y}_i = \mathbf{x}_i - \mu$ and $\tilde{\mathbf{y}}_i = \tilde{\mathbf{x}}_i - \tilde{\mu}$ are centered data vectors. Taking derivative with regards to T, we have

$$\frac{\partial}{\partial T} L(T) = \frac{\partial}{\partial T} \sum_{i=1}^{N} ||\tilde{\mathbf{y}}_i - T\mathbf{y}_i||_2^2 \propto T \sum_{i=1}^{N} \mathbf{y}_i \mathbf{y}_i^T - \sum_{i=1}^{N} \mathbf{y}_i \tilde{\mathbf{y}}_i^T.$$

After setting to zero, $T = \left(\sum_{i=1}^{N} \mathbf{y}_i \mathbf{y}_i^T\right)^{-1} \left(\sum_{i=1}^{N} \mathbf{y}_i \tilde{\mathbf{y}}_i^T\right)$.

ACKNOWLEDGMENT

The authors would like to thank S. Tumu for helping with the experiments. The authors would also like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] A. Agarwala, "Efficient gradient-domain compositing using quadtrees," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 94–106, 2007.
[2] Alien Skin Software LLC, Image Doctor 2: Restore, Retouch, Remove and Repair [Online]. Available: www.alienskin.com
[3] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "Geometric tampering estimation by means of a SIFT-based forensic analysis," in *Proc. ICASSP*, Dallas, TX, 2010.
[4] E. Ardizzone and G. Mazzola, "Detection of duplicated regions in tampered digital images by bit-plane analysis," in *Proc. Int. Conf. Image Analysis and Processing*, Vietri sul Mare, Italy, 2009.
[5] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *Proc. ICASSP*, Taipei, Taiwan, 2009.
[6] J. Beis and D. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proc. CVPR*, San Juan, PR, 1997.
[7] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools 2000.
[8] S. Bravo-Solorio and A. K. Nandi, "Passive forensic method for detecting duplicated regions affected by reflection, rotation and scaling," in *Proc. Eur. Signal Processing Conf.*, Glasgow, Scotland, 2009.
[9] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.
[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, San Diego, CA, 2005.
[11] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, 2nd ed. New York: Wiley, 1973.
[12] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski, "Coordinates for instant image cloning," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–9, 2009.
[13] H. Farid, "Photo fakery and forensics," *Adv. Comput.*, vol. 77, pp. 1–55, 2009.
[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[15] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensic Research Workshop*, Cleveland, OH, 2003.
[16] T. Georgiev, "Photoshop healing brush: A tool for seam-less cloning," in *Proc. Workshop on Applications of Computer Vission (ECCV)*, Prague, Czech Republic, 2004.
[17] M. Gonzalez and F. Woods, *Digital Image Processing*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
[18] H. Gou, A. Swaminathan, and M. Wu, "Intrinsic sensor noise features for forensic analysis on scanners and scanned images," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 3, pp. 476–491, Sep. 2009.
[19] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[20] J. He, Z. Lin, L. Wang, and X. Tang, "Detecting doctored JPEG images via DCT coefficient analysis," in *Proc. ECCV*, Graz, Austria, 2006.
[21] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using SIFT algorithm," in *Proc. IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, China, 2008.
[22] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum, "Drag- and-drop pasting," in *Proc. ACM SIGGRAPH*, Boston, MA, 2006.
[23] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 450–461, Jun. 2007.
[24] X. Kang and S. Wei, "Identifying tampered regions using singular value decomposition in digital image forensics," in *Proc. Int. Conf. Computer Science and Software Engineering*, Wuhan, China, 2008.
[25] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. CVPR*, Washington, DC, 2004.
[26] A. Langille and M. Gong, "An efficient match-based duplication detection algorithm," in *Proc. Canadian Conf. Computer and Robot Vision*, Quebec City, Canada, 2006.
[27] D. Letscher, "Detecting filtered cloning in digital images," in *Proc. ACM Workshop on MM&Sec*, Dallas, TX, 2007.
[28] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. ICME*, Beijing, China, 2007.
[29] H.-J. Lin, C.-W. Wang, and Y.-T. Kao, "Fast copy-move forgery detection," *WSEAS Trans. Sig. Proc.*, vol. 5, no. 5, pp. 188–197, 2009.
[30] Z. Lin, R. Wang, X. Tang, and H. Shum, "Detecting doctored images using camera response normality and consistency," in *Proc. CVPR*, San Diego, CA, 2005.
[31] T. Lindeberg, *Scale-Space Theory in Computer Vision*. Norwell, MA: Kluwer, 1994.
[32] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[33] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital images," in *Proc. ICPR*, Hong Kong, China, 2006.
[34] S. Lyu and H. Farid, "How realistic is photorealistic?," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pt. 2, pp. 845–850, Feb. 2005.
[35] S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.
[36] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," in *Proc. Forensic Science Int.*, 2007, pp. 180–189.
[37] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. ICCV*, Vancouver, Canada, 2001.
[38] A. N. Myna, M. G. Venkateshmurthy, and C. G. Patil, "Detection of region duplication forgery in digital images using wavelets and log-polar mapping," in *Proc. Int. Conf. Computational Intelligence and Multimedia Applications*, Sivakasi, India, 2007.
[39] X. Pan and S. Lyu, "Detecting image region duplication using SIFT features," in *Proc. ICASSP*, Dallas, TX, 2010.
[40] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. ACM SIGGRAPH*, San Diego, CA, 2003.
[41] A. C. Popescu and H. Farid, Exposing Digital Forgeries by Detecting Duplicated Image Regions Department of Computer Science, Dartmouth College, Tech. Rep. TR2004-515, 2004.
[42] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. 6th Int. Workshop on Information Hiding*, Toronto, Canada, 2004.
[43] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pt. 2, pp. 758–767, Feb. 2005.
[44] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pt. 2, pp. 3948–3959, Oct. 2005.
[45] S. Suzuki and K. Abe, "Topological structural analysis of digital binary images by border following," *Comput. Vision Graphics Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.
[46] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 101–117, Mar. 2008.

[47] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *Proc. ACM Workshop on MM&Sec*, Dallas, TX, 2007.

[48] Q. Yang and C. Huang, "Copy-move forgery detection in digital image," in *Proc. Pacific Rim Conf. Multimedia*, Bangkok, Thailand, 2009.

Xunyu Pan (S'10) received the B.S. degree in computer science from Nanjing University, China, in 2000, and the M.S. degree in artificial intelligence from the University of Georgia, Athens, GA, in 2004. He is currently working toward the Ph.D. degree in computer science at the State University of New York at Albany.

From 2000 to 2002, he was a lecturer with Department of Computer Science and Technology, Nanjing University, China. His research interests include image processing and forensics, computer vision, multimedia and machine learning.

Mr. Pan is a student member of the ACM.

Siwei Lyu (S'01–M'03) received the B.S. degree in informatics and the M.S. degree in computer science in 1996 and 2000, respectively, both from Peking University. He received the Ph.D. degree in computer science from Dartmouth College in 2005.

He is an Assistant Professor of Computer Science at the State University of New York at Albany. From 2005 to 2007, he was a postdoctoral research associate at New York University and Howard Medical Institute. His research interests include natural image statistics, digital forensics, machine learning, and computer and biological vision.

Dr. Lyu is the recipient of the NSF CAREER Award in 2010.