Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 3215162, 19 pages
http://dx.doi.org/10.1155/2016/3215162

Research Article

A New Copy Move Forgery Detection Method Resistant to Object Removal with Uniform Background Forgery

Guzin Ulutas and Gul Muzaffer

Department of Computer Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey

Correspondence should be addressed to Guzin Ulutas; guzin@ieee.org

Received 7 April 2016; Revised 22 September 2016; Accepted 3 October 2016

Academic Editor: Francesco Soldovieri

Users transfer large number of images everyday over the Internet. Easy to use commercial and open source image editing tools have made intactness of images questionable. Passive methods have been proposed in the literature to determine authenticity of images. However, a specific type of forgery called "Object Removal with uniform Background forgery" becomes a problem for keypoint based methods in the literature. In this paper, we proposed an effective copy move forgery detection technique. The method uses AKAZE features and nonlinear scale space for detection of copied/pasted regions. The proposed method detects "Object Removal with uniform Background" and "Replication" types of forgeries with high precision compared to similar works. Experimental results also indicate that the method yields better discriminative capability compared to others even if forged image has been rotated, blurred, AWGN added, or compressed by JPEG to hide clues of forgery.

1. Introduction

Users transfer many images and video files everyday over the Internet as social networking services have made sharing multimedia content easier. High-resolution images and video files consume a significant part of the Internet bandwidth even if high efficiency coding is used. High quality image and video files are shared not only in social networks but also in news portals, video surveillance transmission and telemedicine image and video transmission. High quality, low cost cameras, and easy to use commercial and open source image editing tools such as Photoshop or GIMP have also made intactness of images questionable. Many methods have been proposed in the literature to determine authenticity of images. These methods can be classified into two groups: active and passive methods.

Digital signatures and watermarking methods fall into the first group. Digital signature requires transmission of not only the image but also a signature created from the image and a private key by an algorithm. Watermarking also requires embedding specially generated watermark data robust to modification attacks into an image for authentication. Both methods require extra information and hence classified as active methods.

The methods in the second group only use selected statistical features of the image to determine possible forgery. Passive methods attract more attention of the researchers recently because these methods do not need extra information to authenticate the image. Substantial amount of research is focused on copy move type of forgery since it is easy to manipulate images by image editing without altering global statistics. Figure 1 shows an example of a copy move forged image where a portion of the image is copied and pasted onto another region to hide or to replicate some part of the image.

The first method to detect copy move forgery proposed by Fridrich and others [1] is based on the Discrete Cosine Transform (DCT). Their method divides the image into 8×8 pixel overlapping blocks. DCT of these blocks forms feature vectors and they are lexicographically sorted to move similar vectors closer. Euclidean distance among set of neighboring vectors is used to determine similarity expected as a result of forged regions. Popescu and Farid used Principal Component Analysis (PCA) to reduce feature vector size of [1]. Their results show that the method can detect forged regions even

(a) (b)

Figure 1: (a) Original image. (b) Forged image.

if Additive Gaussian Noise, JPEG compression, and blurring have been applied to reduce visible clues of forgery [2]. Luo and others used intensity of the blocks to construct feature vectors. Average pixel intensity of R, G, and B channels and some directional information construct 1×7 feature vector corresponding to blocks [3]. The method yields high accuracy ratios even if forged images have been postprocessed. Blur moment invariants are utilized by Mahdian and Saic to make their method robust against blurring operation [4]. They used 1×72 feature vectors to represent the blocks. The authors also exploit the dimension reduction property of the PCA to speed up feature matching time. Bayram and others used Fourier Mellin Transform to represent the blocks [5]. Their method also used Counting Bloom filters to enhance comparison time. Results show that their method can detect slightly rotated forged regions. Rotation invariant Local Binary Patterns (LBP) is used by Li et al. to detect forgery [6]. Zhang and others used Discrete Wavelet Transform (DWT) to extract forged image's subbands [7]. Their method uses phase correlation to test similarity. Ryu and others used Zernike moments of the blocks as rotation invariant features [8]. Bravo-Solorio and Nandi utilized correlation coefficient of the Fourier Transform to test similarity between blocks [9]. Their method discards the blocks with low entropy. Their method also detects flipping as shown in the results. Wang and others used circular blocks and Gaussian Pyramid Decomposition (GPD) while extracting features [10]. GPD decreases computational complexity of the feature matching algorithm. Results show that the method yields high accuracy rates even if forged image has been rotated, blurred, distorted by additive noise, or compressed by JPEG. Wang and others used Hu moments with GPD [11]. GPD is applied on the forged image and Hu moments of each block are calculated to construct feature vectors. In 2011, Huang et al. proposed an improved DCT based method to detect copy move forgery [12]. Their method applies the truncation procedure to reduce the dimension of the features and makes quantization to DCT coefficients make the method more robust against JPEG compression attacks. In 2012, Cao et al. divide overlapping 8×8 DCT transformed blocks into four equal sized regions and calculate mean of DCT coefficients in these regions to construct feature vector [13]. Their results show that the method gives better results when postprocessing operations are applied on the forged images. In 2013, Zhao and Guo apply the DCT transform on 8×8 overlapping blocks and divide the block into 2×2 nonoverlapping subblocks [14]. Singular Value Decomposition is applied on each subblock to construct a feature vector of size 1×16. Results show that the method gives better performance when compared to similar works. Hussain et al. used multiscale Weber's law descriptor and multiscale Local Binary Pattern for copy move forgery detection [15]. Their method also applied Locally Learning Based (LLB) algorithm to reduce the dimension of feature space. Suresh and Rao applied Local Binary Pattern (LBP) on the low frequency content of DWT to extract feature vectors from the blocks [16]. Results show that their method is rotation invariant and correct detection ratio is approximately 99%. In 2016, Zhou et al. used color distribution information to divide the entire search space into smaller pieces [17]. The method assumes that copied and pasted regions will reside on the same cluster. Five image descriptors are utilized by their method to construct feature vectors from the blocks. Results indicate that the method can detect forgery operation even if gamma correction is applied on the forged image. In 2016, Emam et al. used Polar Complex Exponential Transform (PCET) to extract features from the blocks [18]. Approximate Nearest Neighbor (ANN) searching algorithm is also utilized by the method with Locality Sensitive Hashing to determine similar blocks. Results show that the proposed approach is robust to geometric transformations with low computational complexity.

All the methods mentioned above share a common framework. They divide input image into circular or square overlapping blocks, extract spatial or frequency domain features from blocks, compare feature vectors with an appropriate technique to measure similarity, and mark regions as forged if similarity exceeds a certain threshold.

Researchers proposed keypoint feature comparison as an alternative to block feature comparison recently. Huang and others used Scale Invariant Feature Transform (SIFT) to detect and mark forged regions [19]. After their method, Amerini and others realized more comprehensive analysis and used hierarchical clustering to analyze SIFT correspondences [20]. Xu and others used Speeded-Up Feature

Transform (SURF) to detect copied and pasted regions [21]. Their method divides SURF keypoints into two groups and nearest neighbor search is realized in these groups. In 2016, Zhu and others used Oriented Fast and Rotated Binary Robust Independent Elementary Features (ORB) to detect forged regions [22]. Their method also constructs scale space before keypoint extraction to make ORB method robust against scale attacks. Their results show that the method yields better results compared to other keypoint based methods. In [23], Cozzolino et al. used PatchMatch algorithm to compute a high quality approximate nearest neighbor field for the image. Dense linear fitting based post-processing procedure is applied by their method to reduce the complexity. Their results show that the proposed method gives better results compared to state-of-the-art dense field references.

The most important problem of keypoint based methods is their weakness if forged region does not accommodate any keypoint. One realizes "Object removal with uniform Background" type forgery if the operation aims to hide an object in the image. Zhu and others emphasized "Object removal with uniform Background" problem and claim that Scaled ORB (sORB) can detect tampered regions [22]. But keypoint based methods in the literature exhibit weakness if forged region used for removal has smooth characteristic.

SIFT, SURF, and sORB features uses Gaussian scale space either by constructing Gaussian scale space in a pyramidal manner or by approximating Gaussian derivatives through box filters. The most important drawback of these methods is that they do not preserve object boundaries since Gaussian blurring smoothens details and noise characteristic at the same extent. Alcantarilla and others proposed KAZE features to overcome this problem in 2012 [24]. KAZE uses nonlinear diffusion filtering. They also proposed a novel and fast multiscale feature detection and description, Accelarated KAZE (AKAZE) exploiting the benefits of nonlinear scale spaces in 2013 [25]. AKAZE features are faster than KAZE, SIFT, and SURF but slower than ORB features as indicated in [25].

In this work, we utilize AKAZE features to determine the copied and pasted regions on the forged image. Modified Local Difference Binary (M-LDB) is used to extract descriptors from the keypoints found by AKAZE from the input image. Then, Random Sample Consensus (RANSAC) algorithm is applied to eliminate false matches after keypoints are matched. Experimental results show that the proposed method yields better precision results against rotation, blurring, noise addition, and JPEG compression attacks compared to similar works [20–22]. After all, the most important advantage of the method is that it detects "Object Removal with uniform Background forgery" with high precision compared to other works. The proposed method takes the advantage of nonlinear scale space creation to determine the "Object Removal with uniform Background forgery."

The paper is organized as follows. Sections 2 and 3 give the details of the keypoint extraction method and feature extraction technique, respectively. Experimental results are given in Section 4 and conclusion is drawn in Section 5.

2. Keypoint Extraction from the Forged Image

In this work, we adapted AKAZE features that have priority especially to detect keypoints on the uniform regions, in copy move forgery detection. While keypoint based methods in the literature uses Gaussian scale space to detect keypoints on the test images, the proposed method utilizes nonlinear scale space suggested by the AKAZE algorithm to determine keypoints even if the copied and pasted regions are uniform.

The proposed method has five steps: keypoint extraction from the image, feature extraction at the keypoints, matching features, false match elimination with RANSAC (Random Sample Consensus) algorithm, and tamper localization. First two steps of the proposed method use AKAZE keypoint and feature extraction algorithm. Hamming distance is used by the proposed method to match the descriptors of AKAZE features. RANSAC is also utilized by the method at fourth step to eliminate false matches. A new forgery localization method is applied on the coordinates of matched keypoints to localize the forged regions at the last step of the proposed method.

The proposed method is the first in the literature that adapted a nonlinear scale space based keypoint extraction method into copy move forgery detection. In this section and the latter, we give general outline of the AKAZE keypoint and feature extraction methods and the details of the false match elimination and forgery localization procedures.

Popular keypoint extraction methods in the literature such as Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Oriented FAST and Rotated BRIEF (ORB) make use of scale space by filtering the image with a function. For example, SIFT creates scale space of an image with Gaussian kernels. Original image is convolved with Gaussian kernels of increasing standard deviation to construct Gaussian scale space. SURF algorithm creates Gaussian scale space by approximating Gaussian derivatives through box filters. ORB detects keypoints on image using features from Accelerated Segment Test (FAST) algorithm. It also employs Harris corner measure to choose best points. However, FAST does not produce multiscale features. Therefore, the authors used scale pyramid of the image and produce FAST features at each level in the pyramid.

Construction of the scale space with a linear approach has one drawback: Gaussian blurring does not preserve object boundaries as indicated in [24, 25]. Increasing scale in linear scale space eliminates noise effect and prominent structures of image become more dominant. But Gaussian blurring does not take into account the difference between natural boundaries of objects and noise. Blurring smoothens both object details and noise to same extent. Thus, it has a negative effect on localization as scale level increase. Nonlinear scale space is used by AKAZE adaptively blurs image data and reduces noise while object details remain intact, thanks to the nonlinear scale space. The authors used Fast Explicit Diffusion (FED) to build nonlinear scale space [25]. The details of the keypoint extraction phase of AKAZE are given below. The proposed method will extract keypoints from the test image using AKAZE.

Luminance of an image through increasing scale levels can be modeled by nonlinear diffusion. Diffusion process is controlled by the divergence of a flow function. These approaches use nonlinear Partial Differential Equations (PDEs) because nonlinear nature of the differential equations diffuses luminance of the image through the nonlinear scale space. Nonlinear diffusion can be formulated as in

$$\frac{\partial L}{\partial t} = \operatorname{div}\left(c\left(x, y, t\right) \cdot \nabla L\right), \tag{1}$$

where div and ∇ denote the divergence and gradient operations, respectively, and L is the luminance of the image. Conductivity function, c, ensures the applicability of the diffusion according to the local image structure. t denotes the time of the function c and it is also the scale parameter. The image can be represented in a simpler manner for larger t values. Conductivity function c is defined as in

$$c\left(x, y, t\right) = g\left(\left|\nabla L_{\sigma}\left(x, y, t\right)\right|\right). \tag{2}$$

L_{σ} and ∇L_{σ} represent the Gaussian smoothed version of the image and gradient of L_{σ}, respectively. Akaze uses (3) as conductivity function g_2 that supports wide regions.

$$g_2 = \frac{1}{1 + \left|\nabla L_{\sigma}\right|^2 / \lambda^2}. \tag{3}$$

The contrast parameter λ is used for elimination of edges. Histogram of $|\nabla L_{\sigma}|$ denoted by H is constructed to determine the contrast parameter λ. 70% percentile of the histogram will be used to choose the appropriate value of λ.

2.1. Scale Space Construction. The method constructs scale space with O octaves $\{0 \cdots O-1\}$ and S sublevels $\{0 \cdots S-1\}$. Let images in scales generated from the test image L be $L_1 \cdots L_{O \times S}$. First image in each octave, $(L_1, L_{1+S}, L_{1+2S}, \ldots)$, is generated by subsampling the last image in the previous octave and contrast parameter λ used in the previous octave is multiplied by a factor t_{cp} to detect finer edges. The equation given in (4) is used during the generation of the images in the scales, $L_2 \cdots L_{O \times S}$. First image in the first octave is the blurred version of the test image L, $L_1 = G(I, \sigma)$, and the value of $\lambda_1 = \lambda$. The function $G(L, \sigma)$ used in (4) applies Gaussian blurring operation on L with standard deviation σ.

$$\begin{aligned}
&\forall i,\ i \in 2 \cdots O \times S: \\
&\quad \text{if} \quad ((\operatorname{mod}(i, S) \equiv 1)) \\
&\text{then} \quad \lambda_i = \lambda_{i-1} \times t_{\text{cp}} \\
&\qquad L_i \\
&\qquad = \operatorname{FED}\left(L_{i-1}, c\left(G\left(\text{subsample}\left(L_{i-1}\right), \sigma\right), \lambda_i\right), t_i\right), \\
&\quad \text{else} \quad \lambda_i = \lambda_{i-1} \\
&\qquad L_i = \operatorname{FED}\left(L_{i-1}, c\left(G\left(L_{i-1}, \sigma\right), \lambda_i\right), t_i\right).
\end{aligned} \tag{4}$$

Conductivity function c in (4) applies the function proposed by Perona and Malik on (L_{i-1}, σ) as explained in [26]. The function gets the smoothed version of the current scale image and current contrast value as arguments and applies the conductivity function given in (3). Fast Explicit Diffusion (FED) function gets a time value t_i for the ith scale, smoothed version of the previous scale image, and the result of the conductivity function as arguments. Time value t_i for the current scale image L_i is determined using current octave o and scale number s as in (5). Current octave and scale indexes are mapped to their corresponding scale σ as in (5). Scale levels in pixel units σ are converted into time domain by the following equation:

$$\begin{aligned}
\sigma_i\left(o, s\right) &= \frac{1}{2}\left(2^{o+s/S}\right)^2, \\
t_i &= \frac{1}{2}\sigma_i^2.
\end{aligned} \tag{5}$$

Fast explicit diffusion is used by the AKAZE features to create scale images L_i at each scale. FED schemes use iterated box filters to approximate Gaussian kernels. The scheme performs M cycles of n explicit diffusion steps with varying step sizes, τ, to create scale images. The method use one cycle ($M = 1$) and FED implements n steps to find current scale image L_i and determines n step sizes $\tau_1 \cdots \tau_n$ to use with these steps. Each step can get different step sizes and available maximum step size denoted by $\tau_{\max}$ is chosen to be 0.25 (because of the image dimension) by the algorithm.

Appropriate cycle length n for the current image, L_i, is determined according to the time value t_i of the current scale. Cycle length n for the current scale is determined using

$$n = \left\lceil -\frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{12\left(t_i - t_{i-1}\right)}{\tau_{\max}}} \right\rceil. \tag{6}$$

In this regard, step size for the jth step in the FED cycle is calculated using

$$\tau_j = \frac{3\left(t_i - t_{i-1}\right)}{n\left(n+1\right)\tau_{\max}} \times \frac{\tau_{\max}}{2\cos^2\left(\pi\left(\left(2j+1\right)/\left(4n+2\right)\right)\right)}. \tag{7}$$

FED cycle given in (8) is repeated n times with step sizes $\tau_1 \cdots \tau_n$ for $G(L_{i-1}, \sigma)$ considering a priori estimate $L_i^0 = G(L_{i-1}, \sigma)$. L_i^j represents the temporary scale image in the jth cycle of FED. Matrix A is built according to the method described in [27]. Scale image will be created after n cycle, $L_i = L_i^n$

$$L_i^j = \left(I + \tau_j A\left(G\left(L_{i-1}, \sigma\right)\right)\right) L_i^{j-1}, \quad j = 1 \cdots n-1. \tag{8}$$

2.2. Keypoint Extraction. Keypoint extraction is realized on each image in the nonlinear scale space. Hessian matrix of each image is calculated and multiplied by a normalized scale factor. The factor will be different for each image in the scale space. The formula given in (9) is used to calculate current normalized scale factor, sf_i for the ith image L_i in the nonlinear scale space. Let octave index of L_i be o.

$$\text{sf}_i = \frac{\sigma_i}{2^o}. \tag{9}$$

Let Hessian matrix of the current scale image L_i be H_i. The matrix is multiplied by the scaling factor sf_i^2, $H_i = \mathrm{sf}_i^2 \cdot H_i$. Then, keypoint extraction algorithm computes the determinant of the scaled Hessian matrix to find scale space extremes. The values in the matrices are checked to find the points whose values are higher than a predefined threshold and this point is a maxima in its 3×3 neighborhood. Other keypoints in the former determinant image in the scale space are also investigated for maxima determination. The coordinate of the current keypoint is multiplied by 2^i to compare with keypoints residing on lower scale. Points close to other keypoints residing on lower scales with a high response are chosen as keypoints. After keypoints are extracted from the images in the nonlinear scale space according to lower scale levels, they are filtered with upper scale levels. Each keypoint extracted from the ith scale image, L_i, will be consulted with other keypoints reside on the $i+1$th scale image L_{i+1}. A point is selected as a keypoint if it has higher response in L_i compared to other points in a window at L_{i+1}.

3. Extraction of the Descriptors and Matching the Features

The details of descriptor extraction, descriptor matching, and false match elimination are given in this section.

3.1. Descriptor Extraction Algorithm. The method uses Modified Local Difference Binary (M-LDB) method proposed by [25] to extract features at the keypoints. LDB descriptor proposed by [28] uses the same approach introduced by Binary Robust Independent Elementary Features (BRIEF) [29]. However, LDB uses average of areas to make binary comparisons instead of single pixel values used in BRIEF. The means of the horizontal and vertical derivatives in corresponding areas are also used for comparison. Thus, three bits represent binary comparison results. LDB divides the patch into grids of various sizes, 2×2, 3×3, and so forth. Average computation of these subregions is very fast by using the integral images. But using the integral images makes the feature extraction method vulnerable to rotation. In this regard, AKAZE proposed to use main orientation information to make the LDB rotation invariant. M-LDB rotates the grid of LDB according to the main orientation information. M-LDB method steps are given below.

Step 1. Main orientation for the current keypoint is determined.

Step 2. Subsampling step size is determined from pattern size used in the algorithm. Pattern size is 12 for the method and sample step sizes are $\{5, \lceil 5(2/3) \rceil = 4, \lceil 5(1/2) \rceil = 3\}$. Assume that coordinates of the current keypoint scaled with octave value be (k_x, k_y). A 12×12 grid centered at (k_x, k_y) is divided into 5×5, 4×4, and 3×3 subregions. Each subregion is rotated by main orientation and average pixel values and derivatives in both horizontal and vertical directions are calculated. 12, 27, and 48 average values are generated for 5, 4, and 3 step sized subregions, respectively, and they are embedded into a 1×87 temporary vector T.

Step 3. Average values are compared in this step. The temporary vector T has three parts: first 12 for step size 3, from 13 to 39 for step size 4, and from 40 to 87 for step size 5. Elements in each part are compared with each other and a 1×486 descriptor is created for each keypoint as explained in [25].

3.2. Descriptor Matching Algorithm. The method extracts keypoints from the test image using AKAZE as described in Section 2 and obtains corresponding descriptors with the method explained above. Assume that corresponding descriptors at the keypoints be $A = A_1 \cdots A_k$, where k is the number of keypoints for the test image. Each 1×486 descriptor is compared to other vectors in the descriptor list. The method uses Hamming distance to determine similarity between two descriptor vectors. Corresponding binary elements of two vectors are XOR'ed to count the number of elements with different values. The equation given in (10) compares ith and jth descriptor vectors. Let A_i^k be the kth element of the ith descriptor.

$$\mathrm{Hamming}(i, j) = \sum_{k=1}^{468} \mathrm{XOR}\left(A_i^k, A_j^k\right). \tag{10}$$

Keypoints are matched if Hamming distance of two descriptors is smaller than a predefined threshold δ. Coordinate of the keypoints corresponding to matched descriptors is stored in match matrix M. Rows of the match matrix hold coordinates of both of the matched keypoints.

3.3. False Match Elimination. Random Sample Consensus (RANSAC) proposed by Fischler and Bolles estimates general parameters of a certain model with an iterative approach [30]. The method randomly selects a set from the matched keypoints (we use five points in the experiments) and estimates the transformation matrix H as in

$$H \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix}. \tag{11}$$

Matched keypoints will be evaluated according to the transformation matrix H. Each keypoint (x_i, y_i) is transformed by H and compared with its matching keypoint in terms of distance. Matched pair is considered as *inlier* if the distance is smaller than a predefined threshold γ. Otherwise, it is considered as outlier and removed from M. Real matches and false matches in the matrix M are called by inliers and outliers, respectively. A threshold value of $\gamma = 3$ is used by the method.

Figure 2 shows the effect of RANSAC on false matched keypoints. Figures 2(a) and 2(b) show the original image and its tampered version, respectively. The result of the proposed method contains some false matches as can be seen in Figure 2(c). Figure 2(d) indicates that RANSAC eliminates false matched keypoints.

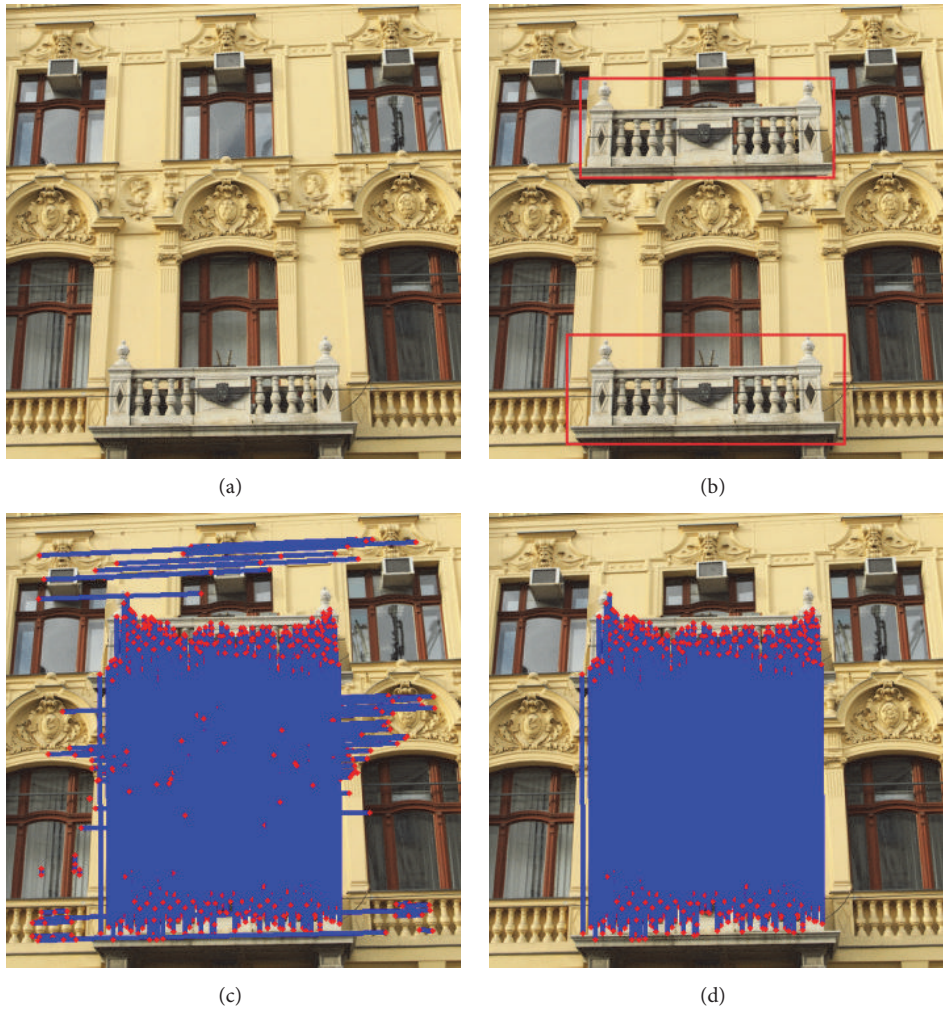


FIGURE 2: (a) Original image. (b) Forged image. (c) The matching result of proposed method without RANSAC. (d) The result of proposed method with RANSAC.

3.4. Tamper Localization. In this step of the proposed method, coordinates of the matched key points are used to localize forged regions. N pairwise matched keypoints are represented by the method as two independent sets: source keypoints in the first set denoted by Source $= \{s_i \mid s_i = (s_{xi}, s_{yi}),\ i \in 1 \cdots N\}$ and corresponding keypoints in the second set denoted by Target $= \{t_i \mid t_i = (t_{xi}, t_{yi}),\ i \in 1 \cdots N\}$. Tamper localization procedure uses corresponding keypoints' coordinates and determines the exact region of forgery. Matlab source code of the algorithm is given in Algorithm 1. Some functions in the code are predefined such as extract_circle(), max_index(), and fill(). The function extract_circle(x, y, r, IM) returns a matrix that represents a circle with center coordinate (x, y) and radius r at the image IM. The function max_index(p) returns the index of maximum valued element in p. The function fill(x, y, r, IM) colors a circular region at (x, y) with radius r on image IM to black.

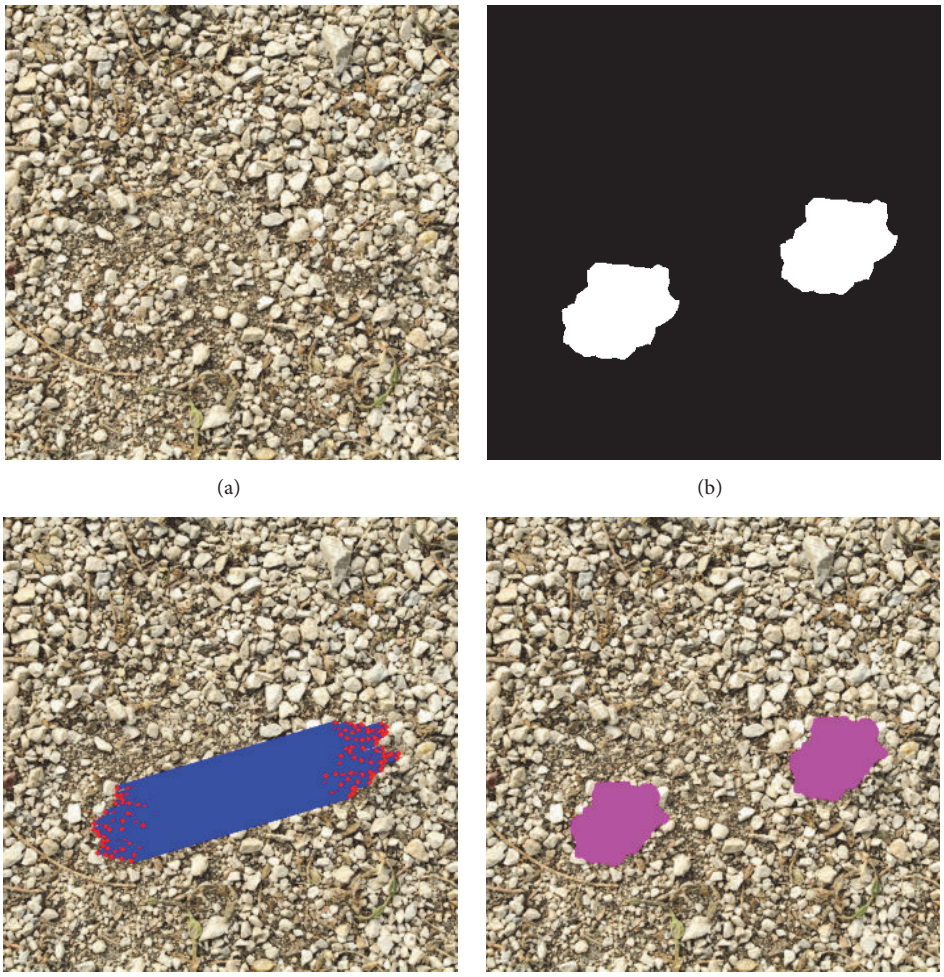
The function extracts the corresponding circular regions with radius r around the corresponding keypoints and calculates Peak Signal to Noise Ratio (PSNR) between them. The function proceeds to calculate PSNR while incrementing radius by one up to r_2. At last, the radius value r that gives maximum PSNR is chosen and corresponding regions around the keypoints are colored to black. Opening operation is applied on the image after all matched N keypoints are processed in the same manner. Figure 3 shows the result of tamper localization algorithm for $r_1 = 3$ and $r_2 = 30$. Figures 3(a) and 3(b) show the forged image and the mask used for tampering process. Matched keypoints after false match elimination and the result of tamper localization algorithm are also given in Figures 3(c) and 3(d), respectively.

4. Experimental Results

Some images are obtained from Google image search and CoMoFoD (http://www.vcl.fer.hr/comofod/) is used to create test dataset [31, 32]. Forged images are created from the test images by GIMP, an open source image editing package, and experiments are carried out on a notebook computer with

```
for j=1: N
      i =1; p=zeros(1, r2-r1+1);
      for r=r1:r2
              circles=extract_circle(sxi, syi, r, IM);
              circlet=extract_circle(txi, tyi, r, IM);
              p[i]=psnr(circles,circlet);
              i = i + 1;
      end
      i=max_index(p);
      r = r1+i-1;
      fill(sxi, syi, r, IM);
      fill(txi, tyi, r, IM);
end
IM = imopen (IM, strel('disk',r));
```

Algorithm 1



(a) (b) (c) (d)

Figure 3: (a) Forged image. (b) The mask of forgery operation. (c) After false match elimination. (d) The result of tamper localization.

Core i7 2.3 GHz processor running OpenCV, another open source package for computer vision. Test dataset is processed to evaluate the effectiveness of the proposed copy move forgery detection method. The results of these experiments are summarized in this section.

There are well known attacks used in the literature to minimize forgery clues making forged images hard to detect. 100 test images denoted by TI are used to create forged image data set to test the robustness of the method. Forged image dataset consists of various partitions as listed below.

Replication Based Forgery (RBF Dataset). 60 test images (60% of the total dataset) are randomly selected from TI and Replication Based Forgery is applied on them. Nonregular regions (that are not larger than %10 of the original image and not smaller than %5 of the original image) are chosen and replicated to create the 60 forged images.

RBFRot30 and RBFRot90. Chosen regions to create the forged images in RBF are rotated by 30° and 90° before pasting and 60 forged images are obtained for RBFRot30 and 60 forged images are obtained for RBFRot90.

RBF-JPG70 and RBF-JPG90. Forged images in RBF are resaved with JPEG quality factor 70 (and quality factor 90) and RBF-JPG70 dataset is created (and RBF-JPG90 is created).

RBF-AWGN25 and RBF-AWGN40. Forged images in RBF are postprocessed by AWGN with 25 dB (and by AWGN with 40 dB) and RBF-AWGN25 dataset is created (and RBF-AWGN40 dataset is created).

RBF-Blur0.5 and RBF-Blur2.0. Forged images in RBF are blurred by Gaussian Function with $\sigma = 0.5$ (and with $\sigma = 2.0$) and RBF-Blur0.5 dataset is created (RBF-Blur2.0 dataset is created).

Object Removal with Uniform Background Based Forgery (ORBF Dataset). Other test images in TI are randomly selected and object removal *with uniform Background* based forgery is applied on them. Nonregular regions (that are not larger than %10 of the original image and not smaller than %5 of the original image) are chosen and removed to create the 40 forged images.

ORBFRot30 and ORBFRot90. Chosen regions to create the forged images in ORBF are rotated by 30° and 90° before pasting and 40 forged images are obtained for ORBFRot30 and 40 forged images are obtained for ORBFRot90.

ORBF-JPG70 and ORBF-JPG90. Forged images in ORBF are resaved with JPEG quality factor 70 (and quality factor 90) and ORBF-JPG70 dataset is created (and ORBF-JPG90 is created).

ORBF-AWGN25 and ORBF-AWGN40. Forged images in ORBF are postprocessed by AWGN with 25 dB (and by AWGN with 40 dB) and ORBF-AWGN25 dataset is created (and ORBF-AWGN40 dataset is created).

ORBF-Blur0.5 and ORBF-Blur2.0. Forged images in ORBF are blurred by Gaussian Function with $\sigma = 0.5$ (and with $\sigma = 2.0$) and ORBF-Blur0.5 dataset is created (ORBF-Blur2.0 dataset is created).

Four scales and four octaves are used to create the scale space in the experiments. The performance of the detection method is measured with True Positive Rate (TPR) and False Positive Rate (FPR). TPR and FPR are calculated by using (12). IDF, FI, IDFO, and OI represent "The number of Images Detected as forged being Forged," "The number of Forged Images," "The number of Images Detected as Forged being Original," and "The number of Original Images," respectively. Receiver Operating Characteristics (ROC) curves are also used in the experiments to make a fair comparison between the proposed method and the others.

$$\begin{aligned} \text{TPR} &= \frac{\text{IDF}}{\text{FI}}, \\ \text{FPR} &= \frac{\text{IDFO}}{\text{OI}}. \end{aligned} \tag{12}$$

The experiments are reported in three sections. Both the proposed and the other methods' matched keypoints are shown on images visually for postprocessed "Replication" and "Object Removal with uniform Background" types of forgery in the first two sections, respectively. Rotation, JPEG compression, blurring, and AWGN are used to minimize traces of forgery. ROC curves are created from TPR and FPR values for both normal and postprocessed images to compare the method with the others in the last section.

4.1. Visual Result of Replication Based Forgery. Visual results of both the proposed and other methods are presented for "Replication" forgery on the test images in this section. Number of matched keypoints is also given in the figures to compare the methods quantitatively.

First experiment gives visual results for randomly selected image from RBF with number of matched keypoints. Forged image is shown in Figure 4(a) where colored rectangular area designates the copied and pasted regions. Visual results for other keypoint based methods are given in Figures 4(c), 4(d), and 4(e). Number of matched keypoints is also reported in the caption of Figure 4. SIFT based forgery detection method matches total 110 keypoints whereas the proposed method matched 324 keypoints. Visual results also support the numeric results.

Second set of experiments in this section applies rotation on copied region before pasting it onto another region as shown in Figure 5(a). Forged image given in Figure 5(a) is chosen from RBFRot30. When SIFT, SURF, and sORB based methods are compared with each other, the method in [20] yields the best result according to the number of matched keypoints as shown in Figure 5(b). The proposed method detects 95 matched keypoints on the forged image, while the method in [20] detects only 28 matched keypoints as indicated in Figures 5(b) and 5(e).

We compare the methods according to average number of keypoints in the third experiment. The proposed

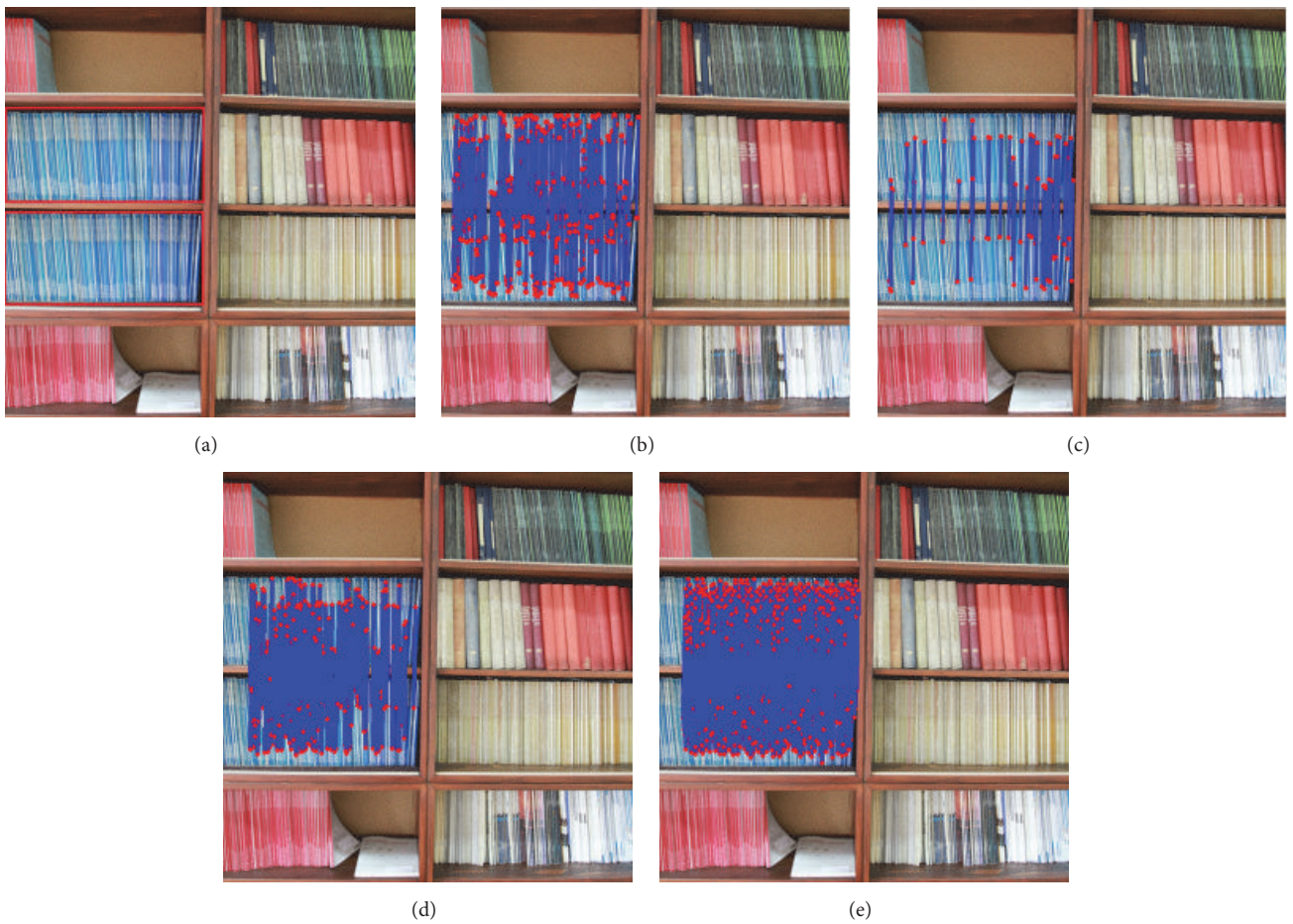


Figure 4: (a) Forged image. (b) The result of SIFT (matched keypoints: 110). (c) The result of SURF (matched keypoints: 41). (d) The result of sORB (matched keypoints: 87). (e) The result of proposed method (matched keypoints: 324).

method and the similar works are applied on the "Replication Based Forgery" datasets (RBF, RBFRot30, RBFRot90, RBF-JPG70, RBF-JPG90, RBF-AWGN25, RBF-AWGN40, RBF-Blur0.5, and RBF-Blur2.0) separately and the average number of matched keypoints is reported in Table 1. When average number of keypoints for RBF-JPG70 and RBF-JPG70 datasets are considered, the proposed method gives better results compared to similar works. The proposed method also gives better results for all datasets. For example, while average number of keypoints for the proposed method is approximately 77 for RBF-Blur2.0 dataset, similar works have approximately 58, 22, and 37 average results. As a result, Table 1 indicates that proposed method detects more matched keypoints on the forged images compared to others [20–22] for various attacks.

Experiments in this section show that the proposed method detects more keypoints compared to similar works in the literature [20–22] on "Replication" type forged regions even if rotation, AWGN, blurring, or JPEG compression is applied.

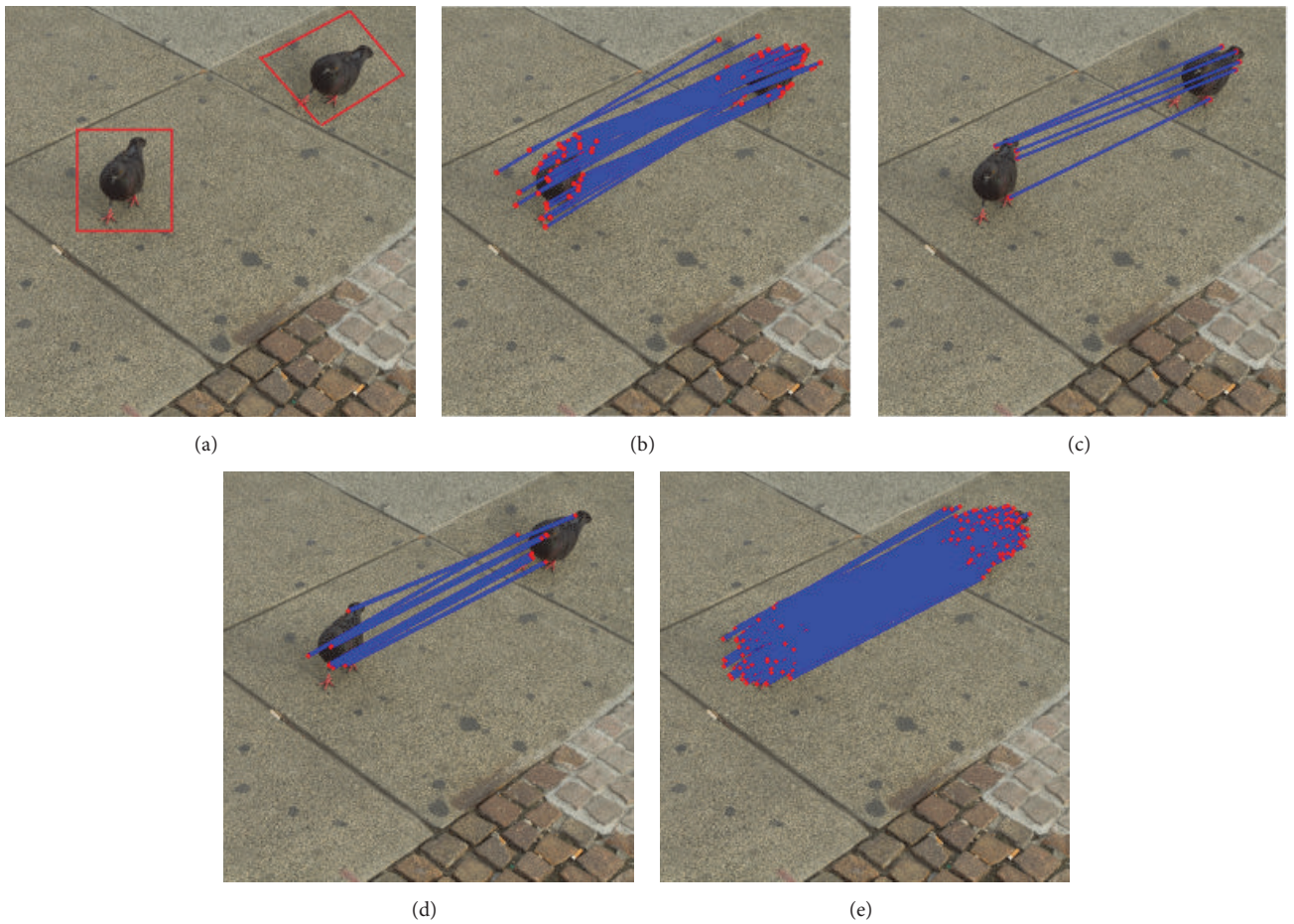
4.2. Visual Result of Object Removal with Uniform Background Forgery. Visual results of both the proposed and other methods are compared for "Object Removal with uniform Background" type forged test images in this section. Keypoint based methods in the literature cannot detect forged regions properly for removal types of forged images. Usually, regions that do not have high frequency components are used for object removal. Thus, keypoint based forgery detection techniques become unsuccessful if copied and pasted regions do not accommodate keypoints. The authors in [22] emphasized this fact and give some results for object removal with uniform Background forgery.

First experiment reports the results of simple object removal with uniform Background forgery operation. Test image in Figure 6(a) is modified to create the forged image as shown in Figure 6(b) that is an example-forged image from ORBF. Figure 6(c) illustrate the result of SIFT, SURF, and sORB based forgery detection methods in the literature. Because regions do not accommodate any keypoint, the methods cannot detect forged regions. However, the proposed method detects 53 matched keypoint pairs as shown in Figure 6(d). AKAZE features based method can extract keypoints from the forged regions thanks to the nonlinear scale space.

Object removal forged image detection performance is also tested for rotated forged region. Figures 7(a) and 7(b)

Table 1: Average number of matching keypoints for each considered method.

		SIFT [20]	SURF [21]	sORB [22]	Proposed method
Rotation	90°	57,41	21,08	32,55	71,35
	30°	52,70	5,38	4,01	55,02
JPEG compression	QF = 90	47,2	22,23	37,61	76,73
	QF = 70	36,97	19,17	29,35	56,52
Gaussian blurring	$\sigma = 0.5$	56,32	20,23	43,41	85,74
	$\sigma = 2$	58,73	22,64	37,1	77,76
Noise addition	SNR = 40 dB	46,14	23,94	43,08	80,91
	SNR = 25 dB	39,85	21,47	34,79	59,11

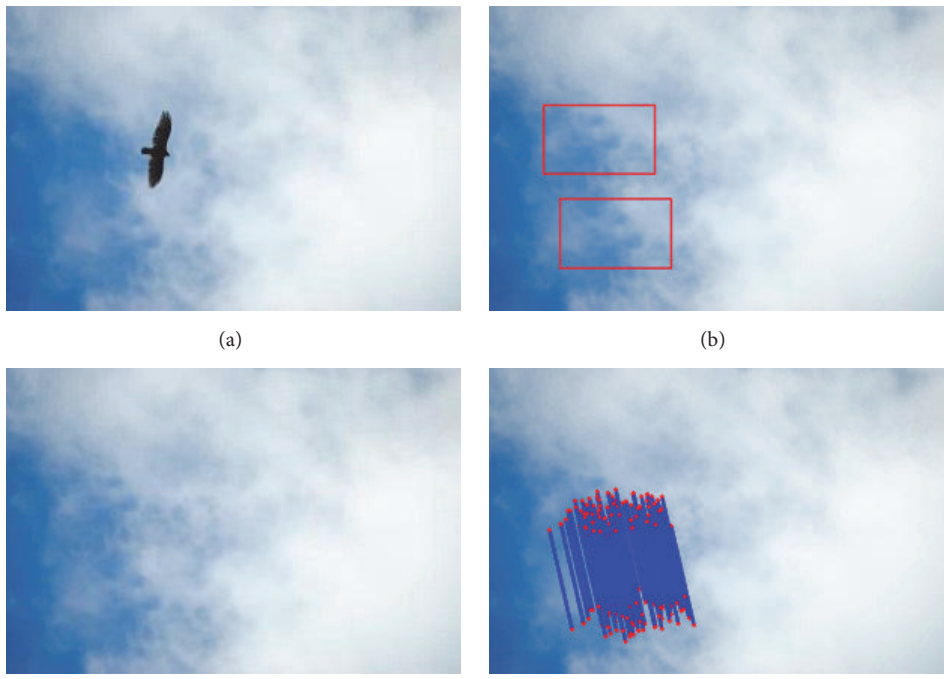


(a) (b) (c) (d) (e)

Figure 5: (a) Forged image where copied region is rotated 30 degrees and then pasted. (b) The result of SIFT (matched keypoints: 28). (c) The result of SURF (matched keypoints: 5). (d) The result of sORB (matched keypoints: 7). (e) The result of proposed method (matched keypoints: 95).

show the original and forged images, respectively, where the region is rotated 30°. Forged image is randomly chosen from ORBF-30. Figure 7(c) designates that other methods in the literature do not detect forgery. The proposed method detects 37 matched keypoint pairs as shown in Figure 7(d), while other methods cannot detect forged regions. Robustness against rotation is also tested by another experiment with 90° rotated forged region. Figures 7(e) and 7(f) show the original and forged images, respectively. Figure 7(g) indicates the results of other methods. SIFT, SURF, and sORB based methods cannot detect any matched keypoints on the forged region as shown in the visual result. One image is shown for all three methods because they do not detect any matched keypoints on the forged region. The method can detect 7 matched keypoint pairs as shown in Figure 7(h).

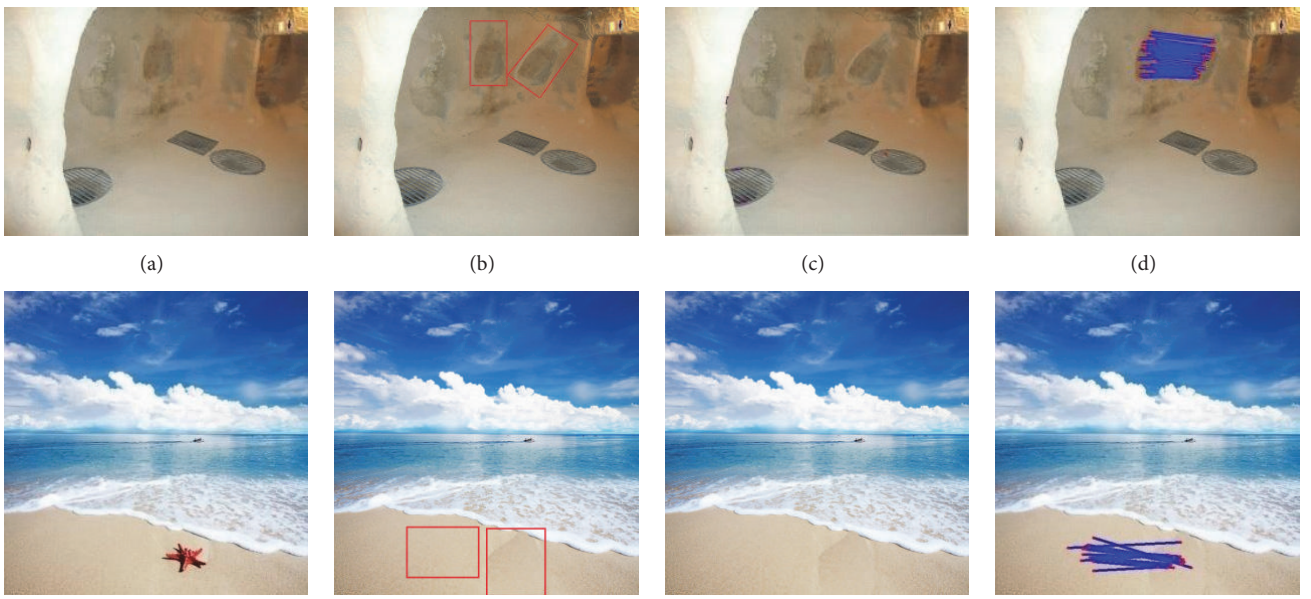
Third experiment shows the effectiveness of the proposed method for JPEG compression on object removal with uniform Background forged image. Quality factors 90



(a) (b)

(c) (d)

Figure 6: (a) Original image. (b) Forged image with smooth region. (c) The result of SIFT, SURF, and sORB method (matched keypoints: 0). (d) The result of proposed method (matched keypoints: 53).



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: (a) Original image. (b) Forged image where copied region is rotated 30 degrees and then pasted. (c) The result of SIFT, SURF, and sORB method (matched keypoints: 0). (d) The result of proposed method (true matched keypoints: 37). (e) Original image. (f) Forged image where copied region is rotated 90 degrees before pasting. (g) The result of SIFT, SURF, and sORB method (matched keypoints: 0). (h) The result of proposed method (true matched keypoints: 7).

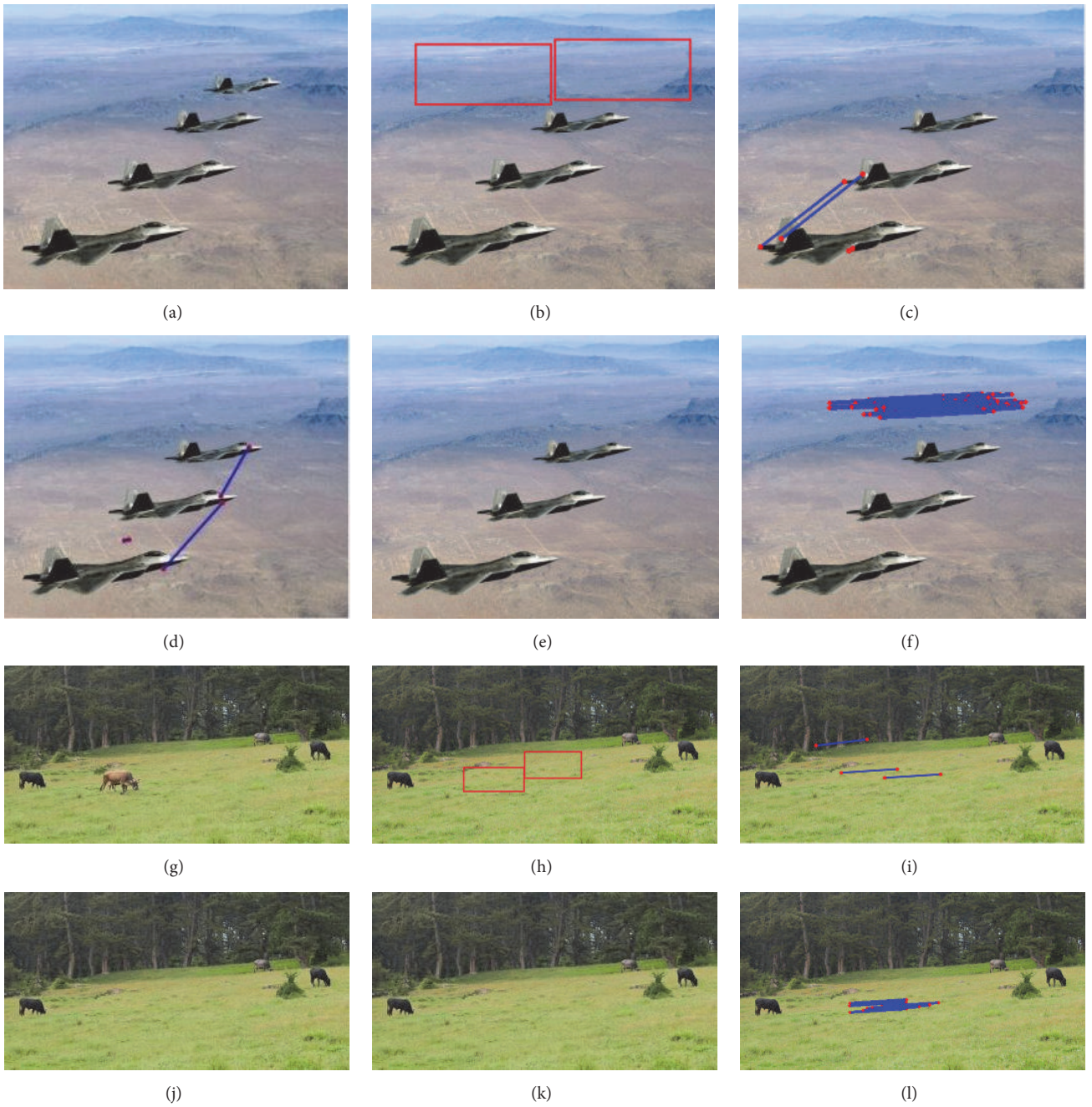


Figure 8: (a) Original image. (b) Forged image with a JPEG quality factor of 90. (c) The result of SIFT (true matched keypoints: 0). (d) The result of SURF (true matched keypoints: 0). (e) The result of sORB (true matched keypoints: 0). (f) The result of proposed method (true matched keypoints: 15). (g) Original image. (h) Forged image with a JPEG quality factor of 70. (i) The result of SIFT (true matched keypoints: 2). (j) The result of SURF (true matched keypoints: 0). (k) The result of sORB (true matched keypoints: 0). (l) The result of proposed method (true matched keypoints: 8).

and 70 are used for testing. Figure 8(a) shows the original image and Figure 8(b) is the forged and JPEG recompressed image by quality factor 90. Forged image is randomly chosen from ORBF-JPG90. SIFT and SURF based methods detect false matches as illustrated in Figures 8(c) and 8(d). Figure 8(e) shows that sORB based method cannot detect any matched keypoint on the forged regions. However, the proposed method matches 15 keypoint pairs as can be seen in Figure 8(f). To create an even lower quality forged image, quality factor 70 is also used in the next experiment to create Figure 8(h) from the test image, Figure 8(g). Numbers of true matched keypoints for the three works in the literature

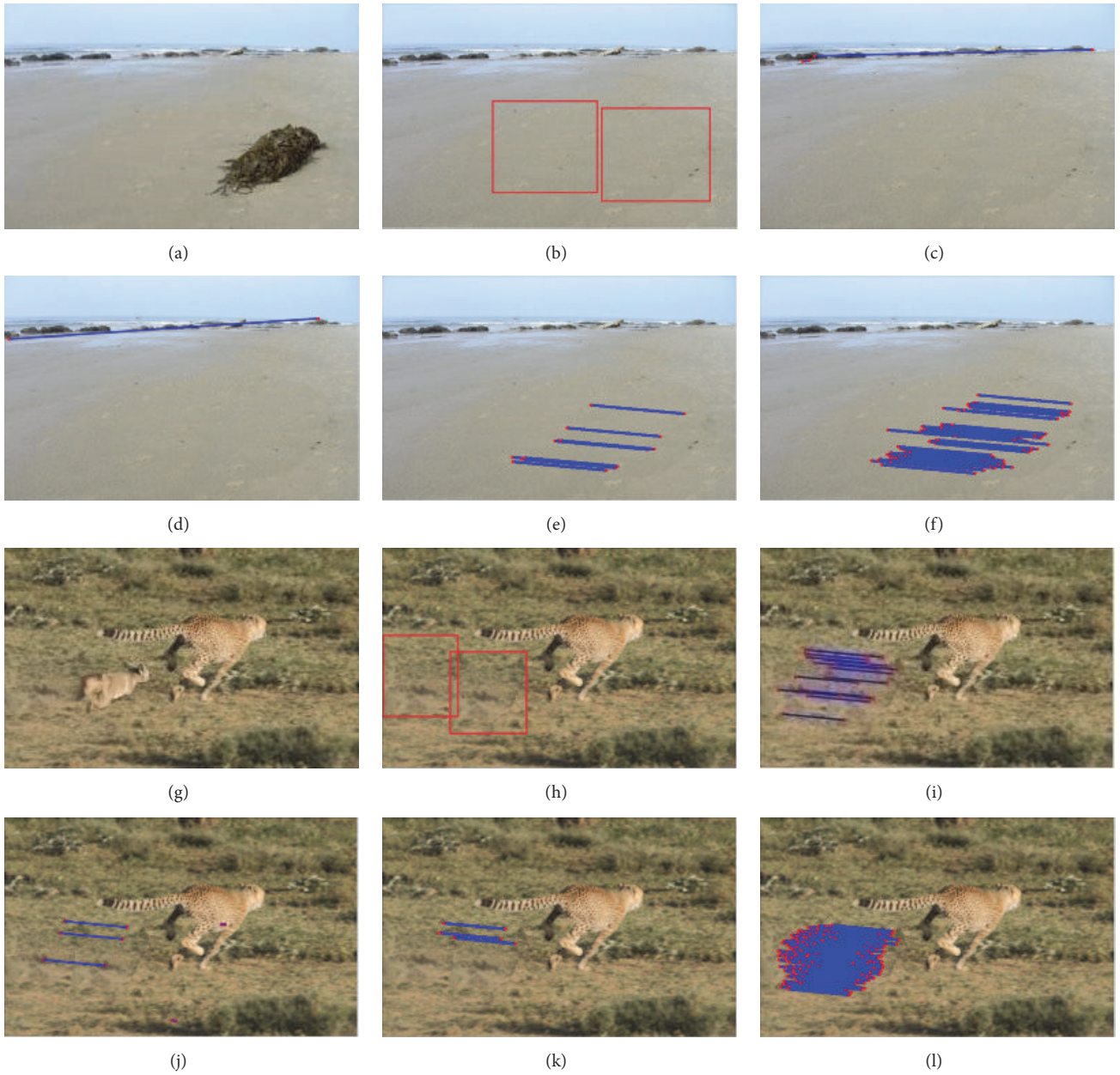


Figure 9: (a) Original image. (b) Forged image is added with Gaussian blur whose kernel is $[3 \times 3]$ and $\sigma = 0.5$. (c) The result of SIFT (true matched keypoints: 0). (d) The result of SURF (true matched keypoints: 0). (e) The result of sORB (true matched keypoints: 9). (f) The result of proposed method (true matched keypoints: 41). (g) Original image. (h) Forged image is added with Gaussian blur whose kernel is $[3 \times 3]$ and $\sigma = 2$. (i) The result of SIFT method (true matched keypoints: 15). (j) The result of SURF (true matched keypoints: 3). (k) The result of sORB (true matched keypoints: 3). (l) The result of proposed method (true matched keypoints: 84).

[20–22] are 2, 0, and 0, respectively. Visual results of [20–22] are given in Figures 8(i), 8(j), and 8(k). The proposed method matches 8 keypoint pairs as shown in Figure 8(l).

The performance of the method for blurring is also evaluated with two different tests. Kernel size is 3×3 and σ is 0.5 and 2 for the tests, respectively. Figure 9(a) shows the test image. Figure 9(b) shows the forged image blurred by a 3×3, $\sigma = 0.5$ Gaussian kernel that is selected randomly selected from ORBF-Blur0.5. SIFT and SURF based methods cannot detect any keypoint on the forged region and thus they do not mark the forged regions as shown in Figures 9(c) and 9(d), respectively [20, 21]. The methods find also false matches. The method in [22] that uses sORB detects 9 matched keypoint pairs as shown in Figure 9(e). However, Figure 9(f) shows that the proposed method detects 41 matched keypoint pairs in the forged region. The methods are also tested for $\sigma = 2$. Figures 9(g) and 9(h) show the test and forged images, respectively. SIFT, SURF, and sORB based methods can detect 15, 3, and 3 matched keypoint pairs as shown in Figures 9(i), 9(j), and 9(k). Figure 9(l) shows that the proposed method

Table 2: Average number of matching keypoints.

		SIFT [20]	SURF [21]	sORB [22]	Proposed method
Rotation	90°	11,28	3,28	3,5	48,21
	30°	10,5	1,92	1,03	42,78
JPEG compression	QF = 90	9,14	3,21	2,92	46,78
	QF = 70	6,64	2,78	1,21	20,2
Gauss blurring	$\sigma = 0.5$	7,23	1,84	5,92	61,38
	$\sigma = 2$	8,31	1,84	3,15	52,92
Noise addition	SNR = 25 dB	5,42	2,5	1,57	23,57
	SNR = 40 dB	7,35	2,42	5,01	49,85

can detect 84 matched keypoint pairs. The method detects more keypoints compared to SIFT based methods.

Figure 10 will show the comparison of the visual results for the proposed and the other methods after white Gaussian noise is added onto the forged image. Figures 10(a) and 10(b) show the original image and forged image that is randomly chosen from ORBF-AWGN40. The method in [20] detects 15-matched keypoint on the forged regions as shown in Figure 10(c). SURF and sORB based methods in [21, 22] do not detect any keypoint on grass as shown in Figures 10(d) and 10(e). Figure 10(f) shows that the proposed method detects 119 matched keypoints thanks to the nonlinear scale space. 20 dB WGN is also added on the forged test image given in Figure 10(g) to obtain Figure 10(h) that is an example image from ORBF-AWGN20. SIFT based method detects 2 matched keypoints on the forged region whereas 28 matched keypoint pairs are obtained by the proposed method as shown in Figures 10(i) and 10(l), respectively. Methods in [21, 22] do not find any matched keypoint on the forged regions as shown in Figures 10(j) and 10(k).

Visual experiments show that the proposed method detects more keypoints compared to similar works [20–22] in the literature on forged regions for "Object Removal with uniform Background" type of forgery even if rotation, AWGN, blurring, or JPEG compression is used to reduce traces of forgery.

At last experiment, we apply the prosed method and similar works on the ORBF, ORBFRot30, ORBFRot90, ORBF-JPG70, ORBF-JPG90, ORBF-AWGN25, ORBF-AWGN40, ORBF-Blur0.5, and ORBF-Blur2.0 datasets separately. Average numbers of matched keypoints for all methods are reported in Table 2. The experiment indicates that the proposed method matches more keypoints on the forged images when compared to similar works under various attacks. For example, while the average number of keypoints for the proposed method is approximately 42 for RBFRot30 dataset, similar works have approximately 10, 2, and 1 average results. When ORBF-AWGN40 dataset is considered, while proposed method has approximately 49 average results, the others have approximately 7, 2, and 5 average results [20–22].

4.3. Evaluation of the Method by TPR/FPR Values and ROC Curves. First experiment in this section compares the proposed method with similar works according to TPR/FPR values. A threshold value of 11 is used to classify a test image to be forged or original. This threshold value is determined to be the best value for the dataset according to TPR/FPR values when all the previous works and the proposed method are considered [20–22]. When the matched number of keypoints is greater than 11, the method classifies the image as forged. Otherwise the image is labeled as authentic.

Table 3: TPR and FPR for each considered method with threshold value 11.

SIFT [20]		SURF [21]		sORB [22]		Proposed method	
TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
0,59	0,02	0,31	0,02	0,45	0,01	0,80	0,02

Forged images in the datasets (RBF, RBFRot30, RBFRot90, RBF-JPG70, RBF-JPG90, RBF-AWGN25, RBF-AWGN40, RBF-Blur0.5, RBF-Blur2.0, ORBF, ORBFRot30, ORBFRot90, ORBF-JPG70, ORBF-JPG90, ORBF-AWGN25, ORBF-AWGN40, ORBF-Blur0.5, and ORBF-Blur2.0) are used to report the average TPR. On the other hand, Original 100 test images in TI are also used to report the average FPR values. Table 3 shows the average TPR/FPR values for the proposed method and for the others in the literature. TPR values show that proposed method can detect forged images with higher accuracy. While TPR for the proposed method is 0.8, the SIFT, SURF, and ORB based methods have approximately 0.6, 0.3, and 0.4 TPR values, respectively. The proposed method also yields the same FPR value compared to SIFT and SURF based methods [20, 21] and approximately equal result to ORB based method.

ROC curves indicate Sensitivity or TPR as a function of Specificity or FPR for varying threshold values. $x = y$ line divides the ROC space into two sections. Points above the line and points below the line represent the good classification results and bad results, respectively. Comparison of ROC curves for the proposed method and others in the literature is also given in this section.

Figures 11(a) and 11(b) represent the ROC curves of the methods for both "Replication" and "Object Removal with uniform Background" types of forged images with rotated regions. Forged images in RBFRot30, RBFRot90, ORBFRot30, and ORBFRot70 are used with their original versions during the experiment. ROC curves of the methods in the literature are given along with the result of proposed method in Figure 11. The proposed method exhibits better

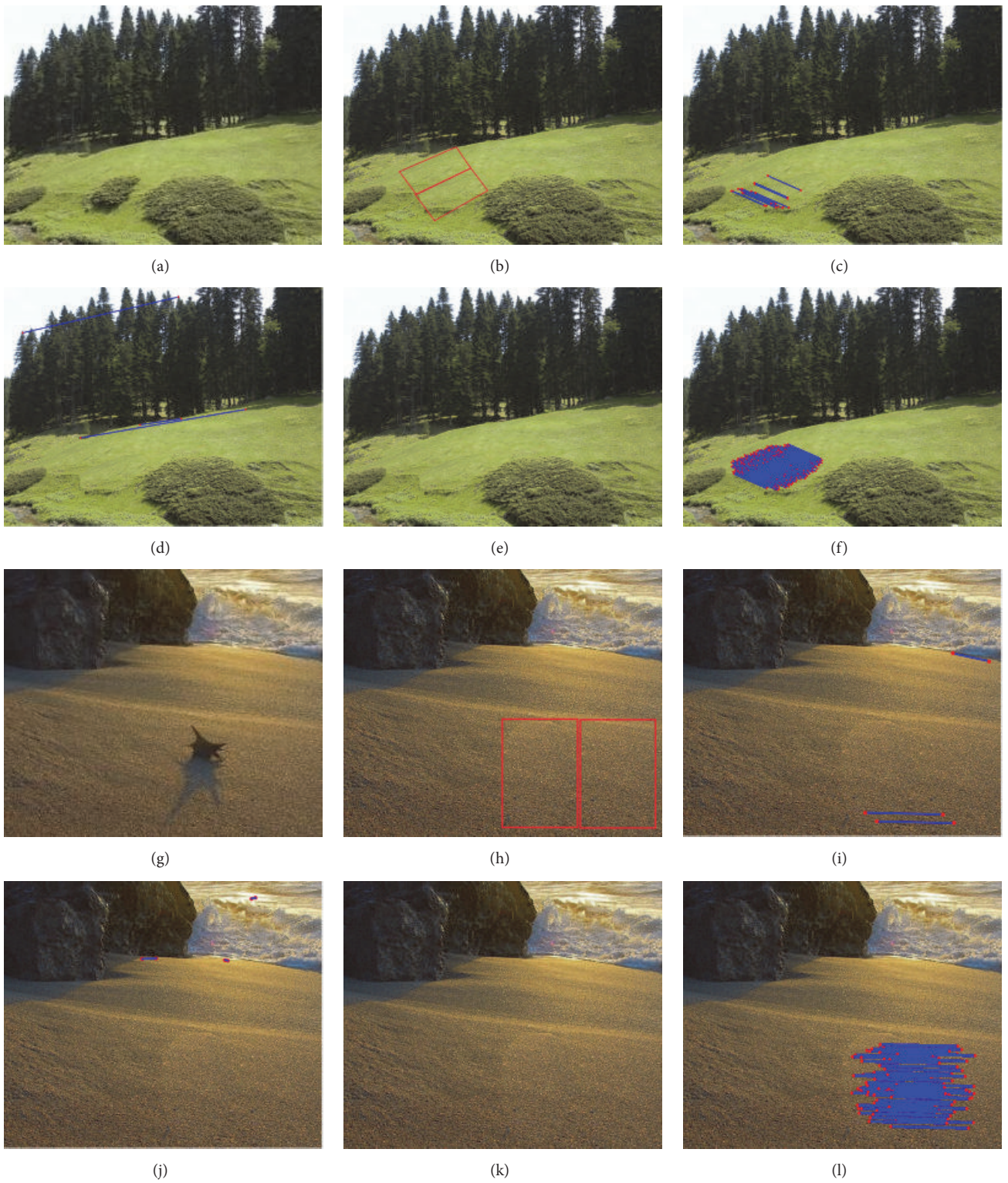


FIGURE 10: (a) Original image. (b) Forged image is added with 40 dB adaptive white Gaussian noise. (c) The result of SIFT method (true matched keypoints: 15). (d) The result of SURF (true matched keypoints: 0). (e) The result of sORB (true matched keypoints: 0). (f) The result of proposed method (true matched keypoints: 119). (g) Original image. (h) Forged image is added with 20 dB adaptive white Gaussian noise. (i) The result of SIFT method (true matched keypoints: 2). (j) The result of SURF (true matched keypoints: 0). (k) The result of sORB (true matched keypoints: 0). (l) The result of proposed method (true matched keypoints: 28).

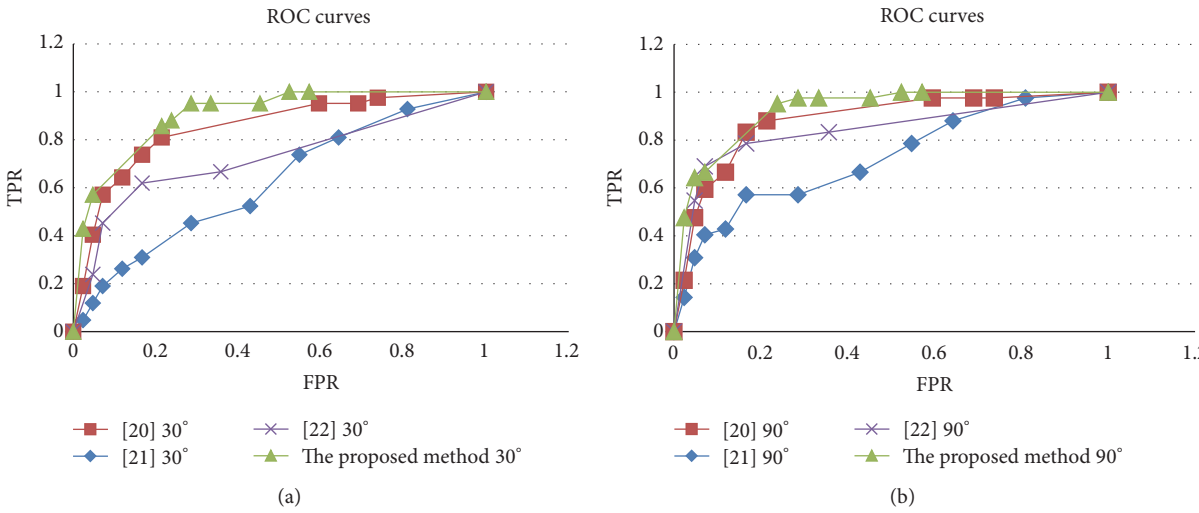


Figure 11: (a) ROC curves of 30° rotation. (b) ROC curves of 90° rotation based on SIFT [20], SURF [21], sORB [22], and the proposed method.

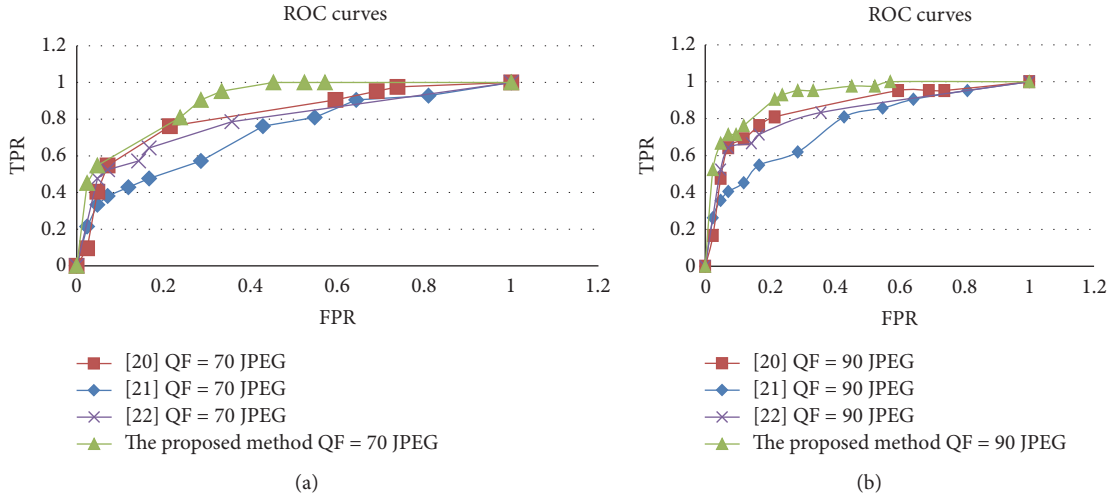


Figure 12: (a) The ROC curves of compression with a JPEG quality factor of 70. (b) The ROC curves of compression with a JPEG quality factor of 90 based on SIFT [20], SURF [21], sORB [22], and the proposed method.

classifying performance compared to other methods for both 30° and 90° rotated forged regions.

Impact of JPEG compression on discriminative capability of the proposed method is evaluated with two different JPEG quality factors in the second set experiments. Forged images in RBF-JPG70, RBF-JPG90, ORBF-JPG70, and ORBF-JPG90 are used with their original versions during the experiment to evaluate the methods and the corresponding ROC curves are given in Figures 12(a) and 12(b), respectively. ROC curves in Figure 12 show that the proposed method classifies forged images with better performance compared to similar works. The method also yields better ROC curve when quality factor 70 is used to recompress the forged images.

ROC curves for blurred forged images in RBF-Blur0.5, RBF-Blur2.0, ORBF-Blur0.5, and ORBF-Blur2.0 are also given in Figures 13(a) and 13(b), respectively. 80 forged images are used in this experiment. The proposed method has better discrimination compared to similar works for both $\sigma = 0.5$ and $\sigma = 2$ as shown in Figure 13. The method's discrimination improves as the value of σ increase.

Robustness of the method is compared with others using the ROC curves under noise addition attack as the last experiment. Forged images in RBF-AWGN25, RBF-AWGN40, ORBF-AWGN25, and ORBF-AWGN40 and their original versions are used during the test. Figure 14 shows the ROC curves of the methods for two-noise level. Figure 14(b) shows that the proposed method can classify the images with better accuracy even if the noise is 25 dB SNR.

ROC curves are used to show that the proposed method has better classification compared to similar works even if various attacks are performed to hide traces of forgery. It indicates that AKAZE features based method can detect forged

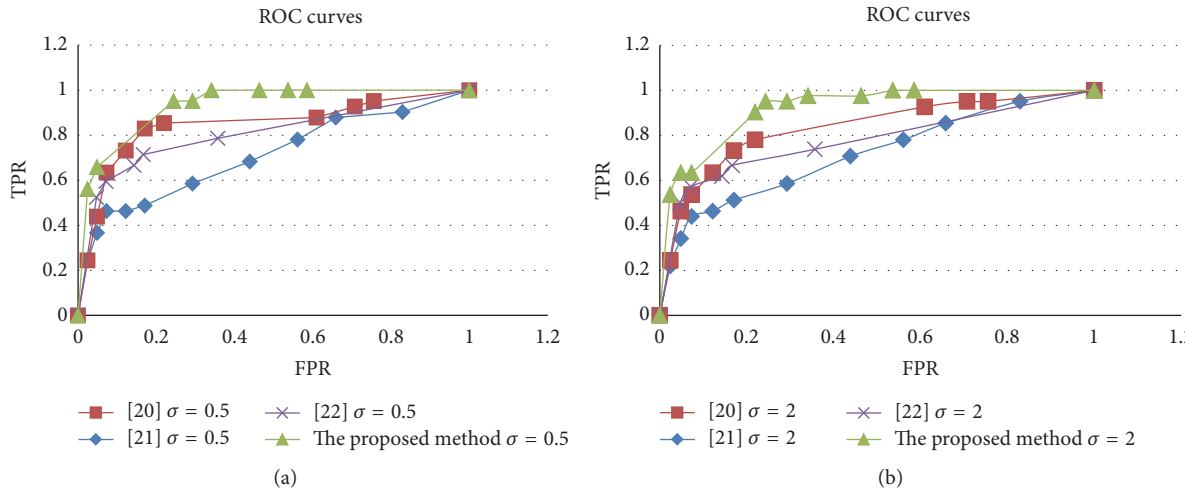


FIGURE 13: (a) The ROC curves of Gaussian blurring with $\sigma = 0.5$. (b) The ROC curves of Gaussian blurring with $\sigma = 2$ based on SIFT [20], SURF [21], sORB [22], and the proposed method.

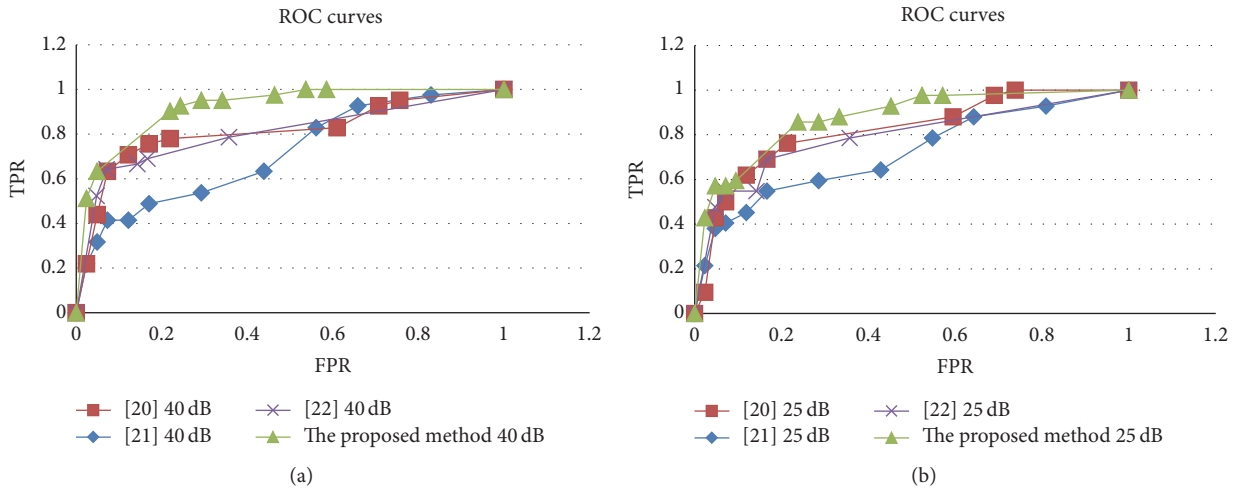


FIGURE 14: (a) The ROC curves of adaptive white Gaussian noise with 40 dB SNR. (b) The ROC curves of adaptive white Gaussian noise with 25 dB SNR based on SIFT [20], SURF [21], sORB [22], and the proposed method.

regions even in object removal with uniform Background type of forgery. Other methods cannot detect keypoint in forged region used for object removal with uniform Background type of forgery. Nonlinear scale space based AKAZE features applies various scales on the image and preserves the object boundary. AKAZE features can be calculated faster than SURF and SIFT features but slower than ORB features as indicated in [25].

5. Conclusion

In this paper, an effective copy move forgery detection technique is proposed. The method uses AKAZE features for detection of copied and pasted regions and uses RANSAC for elimination of the false matches. Two kinds of forgery used in the literature are "Replication" and "Object Removal with uniform Background" types of forgeries. Latter forgery technique becomes a problem for keypoint based methods and the authors in [22] reported the problem in their work. The proposed method detects "Object Removal with uniform Background" and "Replication" types of forgeries with high precision compared to similar works thanks to the nonlinear scale creation [20–22]. Experimental results also indicate that the method yields better discriminative capability compared to others even if forged image has been rotated, blurred, WGN added, or JPEG compressed to hide clues of forgery.

Disclosure

The authors further confirm that the order of authors listed in the manuscript has been approved by all of them.

Competing Interests

There is no conflict of interests regarding the publication.

Authors' Contributions

The authors confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed.

References

[1] J. Fridrich, D. Soukalm, and J. Lukas, "Detection of copy-move forgery in digital images," in *Proceedings of the Digital Forensic Research Workshop*, pp. 19–23, Cleveland, Ohio, USA, August 2003.

[2] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Tech. Rep. 2004-515, Dartmouth College, 2004.

[3] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 4, pp. 746–749, August 2006.

[4] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2-3, pp. 180–189, 2007.

[5] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 1053–1056, IEEE Press, Taipei, Taiwan, April 2009.

[6] L. Li, S. Li, and H. Zhu, "An efficient scheme for detecting copy-move forged images by local binary patterns," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 1, pp. 46–56, 2013.

[7] J. Zhang, Z. Feng, and Y. Su, "A new approach for detecting copy-move forgery in digital images," in *Proceedings of the 11th IEEE Singapore International Conference on Communication Systems (ICCS '08)*, pp. 362–366, Singapore, November 2008.

[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 Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1355–1370, 2013.

[9] S. Bravo-Solorio and A. K. Nandi, "Exposing duplicated regions affected by reflection, rotation and scaling," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, pp. 1880–1883, Prague, Czech Republic, May 2011.

[10] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang, "Detection of image region duplication forgery using model with circle block," in *Proceedings of the 1st International Conference on Multimedia Information Networking and Security (MINES '09)*, pp. 25–29, IEEE, Hubei, China, November 2009.

[11] J.-W. Wang, G.-J. Liu, Z. Zhang, Y.-W. Dai, and Z.-Q. Wang, "Fast and robust forensics for image region-duplication forgery," *Acta Automatica Sinica*, vol. 35, no. 12, pp. 1488–1495, 2009.

[12] Y. Huang, W. Lu, W. Sun, and D. Long, "Improved DCT-based detection of copy-move forgery in images," *Forensic Science International*, vol. 206, no. 1–3, pp. 178–184, 2011.

[13] Y. Cao, T. Gao, L. Fan, and Q. Yang, "A robust detection algorithm for copy-move forgery in digital images," *Forensic Science International*, vol. 214, no. 1–3, pp. 33–43, 2012.

[14] J. Zhao and J. Guo, "Passive forensics for copy-move image forgery using a method based on DCT and SVD," *Forensic Science International*, vol. 233, no. 1–3, pp. 158–166, 2013.

[15] M. Hussain, S. Qasem, G. Bebis, G. Muhammad, H. Aboalsamh, and H. Mathkour, "Evaluation of image forgery detection using multi-scale weber local descriptors," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 4, Article ID 1540016, 2015.

[16] G. Suresh and C. S. Rao, "RST invariant image forgery detection," *Indian Journal of Science and Technology*, vol. 9, no. 22, Article ID 89227, 2016.

[17] H. Zhou, Y. Shen, X. Zhu, B. Liu, Z. Fu, and N. Fan, "Digital image modification detection using color information and its histograms," *Forensic Science International*, vol. 266, pp. 379–388, 2016.

[18] M. Emam, Q. Han, and X. Niu, "PCET based copy-move forgery detection in images under geometric transforms," *Multimedia Tools and Applications*, vol. 75, no. 18, pp. 11513–11527, 2016.

[19] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using sift algorithm," in *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA '08)*, pp. 272–276, Computer Society, December 2008.

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

[21] B. Xu, J. Wang, G. Liu, and Y. Dai, "Image copy-move forgery detection based on SURF," in *Proceedings of the 2nd International Conference on Multimedia Information Networking and Security (MINES '10)*, pp. 889–892, IEEE, Nanjing, China, November 2010.

[22] Y. Zhu, X. Shen, and H. Chen, "Copy-move forgery detection based on scaled ORB," *Multimedia Tools and Applications*, vol. 75, no. 6, pp. 3221–3233, 2016.

[23] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2284–2297, 2015.

[24] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Proceedings of the European Conference on Computer Vision (ECCV '12)*, Fiorenze, Italy, October 2012.

[25] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proceedings of the 24th British Machine Vision Conference (BMVC '13)*, Bristol, UK, September 2013.

[26] P. Perona and J. Malik, "Scale-space and edge detection using annisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 1651–1686, 1990.

[27] J. Weickert, B. M. Ter Haar Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 398–410, 1998.

[28] X. Yang and K.-T. Cheng, "LDB: An ultra-fast feature for scalable Augmented Reality on mobile devices," in *Proceedings of the 11th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '12)*, pp. 49–57, Atlanta, Ga, USA, November 2012.

[29] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 778–792, Springer, Berlin, Germany, 2010.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis

and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] Google Image Search, http://images.google.com/.

[32] CoMoFoD database, http://www.vcl.fer.hr/comofod.