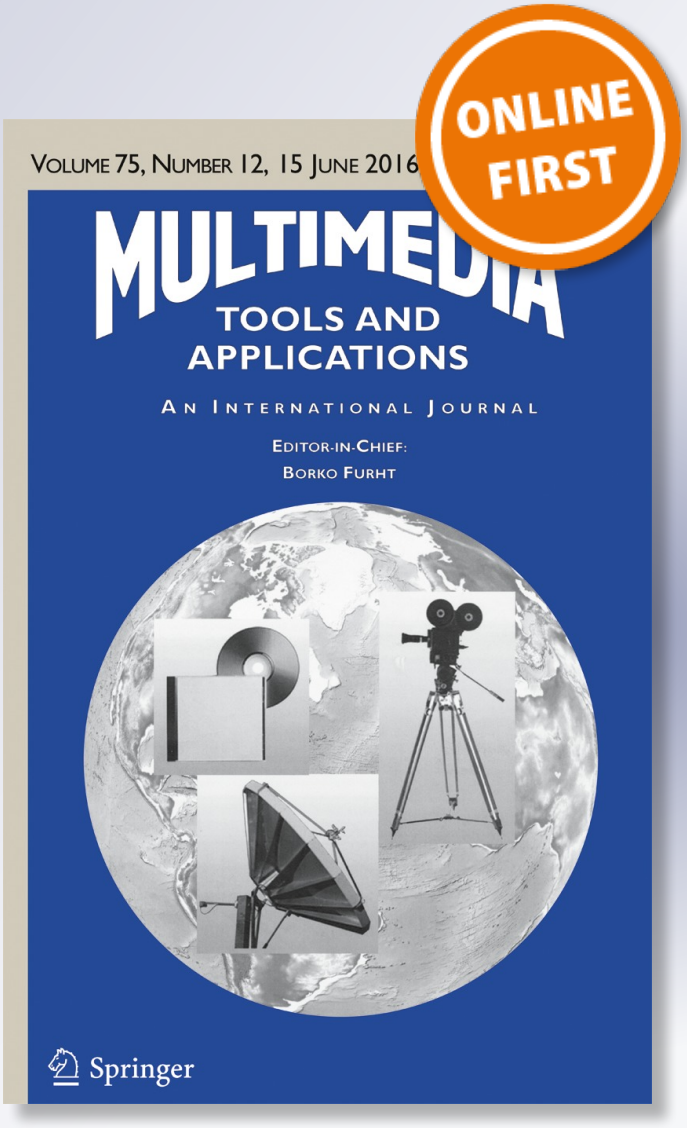
Image splicing localization using PCA-based noise level estimation

Hui Zeng, Yifeng Zhan, Xiangui Kang & Xiaodan Lin

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-016-3712-8



Springer

Multimed Tools Appl
DOI 10.1007/s11042-016-3712-8

Image splicing localization using PCA-based noise level estimation

Hui Zeng[1] · **Yifeng Zhan**[1] · **Xiangui Kang**[1] · **Xiaodan Lin**[1]

Received: 28 September 2015 / Revised: 7 June 2016 / Accepted: 24 June 2016

Abstract Image splicing is one of the most common image tampering operations, where the content of the tampered image usually significantly differs from that of the original one. As a consequence, forensic methods aiming to locate the spliced areas are of great realistic significance. Among these methods, the noise based ones, which utilize the fact that images from different sources tend to have various noise levels, have drawn much attention due to their convenience to implement and the relaxation of some operation specific assumptions. However, the performances of the existing noise based image splicing localization methods are unsatisfactory when the noise difference between the original and spliced regions is relatively small. In this paper, through incorporation of a recent developed noise level estimation algorithm, we propose an effective image splicing localization method. The proposed method performs blockwise noise level estimation of a test image with principal component analysis (PCA)-based algorithm, and segments the tampered region from the original region by k-means clustering. The experimental results demonstrate the superiority of the proposed method over several state-of-the-art methods, especially for practical image splicing, where the noise difference between the original and spliced regions is typically small.

Keywords Image splicing · Image splicing localization · Noise level · Principal component analysis (PCA) · K-means clustering

✉ Xiangui Kang
isskxg@mail.sysu.edu.cn

Hui Zeng
zengh5@mail2.sysu.edu.cn

Xiaodan Lin
xd_lin@hqu.edu.cn

[1] Guangdong key laboratory of information security, School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Published online: 01 July 2016

1 Introduction

Owing to the increasing availability of low cost, and sometimes free-of-cost, powerful image editing software, even a regular user can easily tamper digital images without leaving obvious visual artifacts. The ease of image tampering challenges the credibility of image contents as evidences in varied fields. As a consequence, forensic tools aiming to reveal various types of image tampering have recently drawn increasing attention.

There are two most commonly used image-tampering techniques that can significantly change the content of the original image, namely copy-move and image splicing. In the copy-move tampering, a region of an image is duplicated and pasted elsewhere on the same image to hide undesired objects or to replicate objects, which usually results in duplicated regions in the tampered image [1, 16, 30]. The current paper focuses on the other type of tampering, i.e., image splicing, where a forged image is generated by splicing regions from different source images. Existing image splicing localization methods can be classified into two categories:

1) The first category generally assumes that a specific operation has been introduced in the spliced region or the boundary between the spliced region and the original region. Hence the problem of locating image spicing is reduced to the detection of the specific operation locally. For example, the works in [2, 4, 17, 18, 27] assume that the spliced region has been blurred, median filtered, resampled, contrast enhanced and double JPEG compressed, respectively.
2) The second category is based on certain intrinsic fingerprints of the original image. A typical example of this category is the PRNU based method [6, 20], in which the tampered region is revealed by the absence of the camera PRNU. This PRNU based method is improved in [7] with an advanced denoising filter BM3D [11] in extracting the fingerprint, and in [9] by using a Markov random field prior to enforce regularity constraints. Guided filtering [14] is used to improve the resolution of the localization results in [8], which achieves a better performance for small forgeries. The most important advantage of the PRNU based methods is that they make no assumption of a specific operation involved in the splicing process. However, the PRNU based method requires the camera fingerprint or that the camera itself is available to the forensic investigator.

Due to the complexity of the image splicing localization problem itself, all of the reviewed methods are based on some certain assumptions or some prior knowledge of the questioned image, and the practical splicing localization system usually involves a large set of features or techniques to reveal various splicing artifacts [10]. Moreover, the two categories above are not strictly mutual exclusion, e.g., the noise based image splicing localization methods [21–24, 26] can be classified into either category. The noise based methods are motivated by the fact that the intrinsic noise level is usually consistent across an original image and images from different origins usually have different noise levels. Such differences could either be introduced during the image capturing process, or caused by intentional noise addition to conceal the traces left by image splicing. Figure 1 shows two image splicing examples. For the example on the left, a bird from another image is spliced into the original image. During the forgery, the bird is added with mild Gaussian noise to conceal the spliced region [26] or for the anti-forensic purpose [29]. For the example on the right, a toy duck from another image is spliced into the target image. As will be illustrated later, the source image of the toy duck has different intrinsic noise level from the target image, probably due to different camera settings, e.g., ISO speed.

Fig. 1 Two examples of image splicing. (left) A bird from another image with noise addition is spliced into the original image. (right) A toy duck from another image is spliced into the original image

In [26], noise variance is estimated by computing the second and fourth moments for each local image block. However, the method requires the kurtosis of the original signal to be known. In [22], the authors apply the median based noise variance estimator, median absolute deviation (MAD) [12], to local image blocks, and use the local noise level inconsistencies to reveal the image splicing. In [23], the authors apply the noise level estimation method [31] to local image blocks, and the spliced region is segmented from the original image via coarse-to-fine clustering of the blocks. In [21, 24], the same authors of [23] extend the noise level estimation method [31], and achieve pixel-level detection precision when applied to image splicing localization. However, our preliminary tests show that such a method is sensitive to image texture, which in turn may introduce false positives in the localization results. In general, existing noise-based methods achieve satisfactory performance when the noise difference between the original and spliced regions is large enough, e.g., the noise difference is larger than 5. However, such difference in most image forgeries of practical interest is relatively small e.g., the noise difference is smaller than 2, and existing noise-based methods are less accurate in this case. This performance gap is our motivation for seeking a novel method that can handle a case where the noise difference is small.

In this paper, we propose a novel noise-based method to effectively locate splicing regions, which improves the existing methods through a better noise level estimation algorithm and a more reasonable treatment along the splice boundary. We segment the questioned image into blocks and estimate the local noise level blockwise using a principal component analysis (PCA)-based noise level estimation method [28]. K-means algorithm is adopted to cluster image blocks into an original cluster and a spliced cluster based on their noise levels. To get a more precise localization of the spliced regions, a coarse-to-fine strategy is employed and the regions along the boundary are carefully handled. Extensive experiments on various databases demonstrate the performance improvements of the proposed method, especially when the noise difference between the original and spliced regions is relatively small, which significantly improves the practicability of the noise-based image splicing detection methods.

This paper is organized as follows. In the next section, we give detail about the proposed image splicing localization method. Section 3 shows the experimental results and the conclusion is given in Section 4.

2 Proposed method

An image is first segmented into non-overlapping blocks. The local noise level of each block is estimated using a state-of-the-art noise level estimation method [28]. Then k-means clustering

is performed to categorize these blocks into two clusters according to their noise levels. The cluster that has fewer blocks is regarded as spliced region.

2.1 Image noise level estimation

There are several noise level estimation methods in the image processing area. When applying these methods for the splicing localization purpose, the following requirements should be met.

Firstly, to better separate the spliced region from the original region, the estimation should be as accurate as possible. To our best knowledge, the noise level estimation methods based on PCA are the state-of-the-art [19, 28]. Secondly, to obtain a precise localization, the method should be effective on small blocks, e.g., 32×32 pixels or 64×64 pixels. In view of this, the method which only uses weak textural areas is not an ideal choice [19], since there may be no such areas in some small blocks.

In our preliminary study, to provide an intuitive comparison of the noise level estimation methods, we estimate the noise level of 12,000 original image blocks and that of 12,000 noise-added image blocks with three state-of-the-art methods [12, 28, 31]. The source images are from the BOSSraw database [3], which includes both homogeneous and texture areas. Figure 2 shows the capability of separating original image blocks with a block size of 64×64 from noise-added ones using these three methods respectively. From left to right, are the results from the method [12], the method [31], and the method [28], respectively. The standard deviation of the added noise is 2 for the top row and 5 for the bottom row. It is observed that the dynamic ranges of the results of [12, 31] are obviously larger than that of the results of [28], and cause more overlap between the original image blocks and noise-added ones. Therefore, the method in [28] outperforms the others in distinguishing a noise-added image block from an original one. Hence, in the following, we employ the noise level estimation method [28] in our proposed splicing localization method.

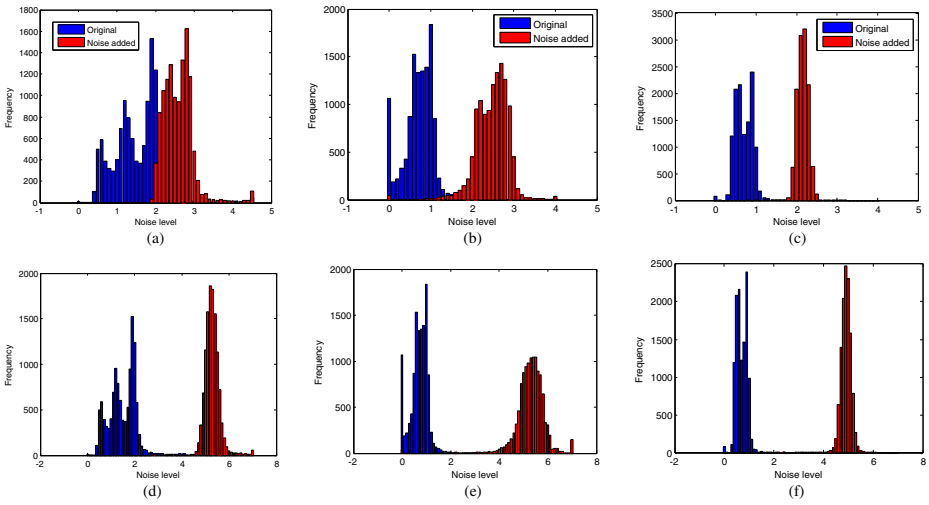


Fig. 2 Histogram plots of the estimated noise levels for original image blocks and noise-added ones. Left column: results of the method [12], $\sigma = 2$(**a**) and $\sigma = 5$ (**d**). Middle column: results of the method [31], $\sigma = 2$(**b**) and $\sigma = 5$ (**e**). Right column: results of the method [28], $\sigma = 2$(**c**) and $\sigma = 5$(**f**)

Let $\boldsymbol{x}$ be a noise-free image (or an image block) of size $S_1 \times S_2$, and $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ be the noisy image corrupted with Gaussian noise $\boldsymbol{n}$. Images $\boldsymbol{x}$ and $\boldsymbol{y}$ both contain $N = (S_1 - M + 1) \times (S_2 - M + 1)$ overlapping patches of size $M \times M$. These patches are vectorized as $\boldsymbol{x}_i, \boldsymbol{y}_i$, $i = 1, \ldots, N$ and their covariance matrices are denoted as $\sum_{\boldsymbol{x}_i}, \sum_{\boldsymbol{y}_i}$, which have $M^2 \times M^2$ entrances. The key assumption in [28] is that:

Assumption 1: $\boldsymbol{x}$ can be sparsely represented by applying PCA, i.e., all $\{\boldsymbol{x}_i\}$ lie in the subspace V_{M^2-m}.

When the assumption above holds, the authors in [28] proved that

$$\lim_{N \to \infty} E\left(\left|\lambda_{\boldsymbol{y},\min} - \sigma^2\right|\right) = 0 \tag{1}$$

where $\lambda_{\boldsymbol{y},\min}$ is the minimum eigenvalue of $\sum_{\boldsymbol{y}_i}$ and σ^2 is the variance of the added noise. Therefore, we can estimate the noise level σ as $\sqrt{\lambda_{\boldsymbol{y},\min}}$, and the estimation accuracy is proportional to N according to [28]. The noise level estimation algorithm can be summarized as follows (please refer to [28] for details):

1. $\boldsymbol{y}$ is decomposed into overlapping patches $\boldsymbol{y}_i$, $i = 1, \ldots, N$. The default patch size is 5×5 pixels, i.e., $M = 5$.
2. An initial estimate is computed based on the image patch variance distribution, which also serves as the upper bound of the whole estimation. Let $s^2(\boldsymbol{y}_i)$ be the sample variance of $\boldsymbol{y}_i$, and $Q(p)$ be the p-quantile of $\{s^2(\boldsymbol{y}_i), i = 1, \ldots, N\}$. The initial noise level estimate is computed as $C_0 Q(p_0)$. In [28], the authors experimentally set $C_0 = 3.1$ and $p_0 = 0.0005$.
3. A subset of the image patches $\boldsymbol{Y}_p$ is selected with (2) by recursively discarding the patches with largest variance until Assumption 1 is satisfied.

$$\boldsymbol{Y}_p = \left\{\boldsymbol{y}_i \middle| s^2(\boldsymbol{y}_i) \le Q(p), i = 1, \ldots, N\right\} \tag{2}$$

Assumption 1 is checked by the following condition:

$$\lambda_{Y_p,m} - \lambda_{Y_p,\min} < 49\sigma^2 / M \tag{3}$$

where $\lambda_{Y_p,\min}$, $\lambda_{Y_p,m}$ are the minimum and the m^{th} smallest eigenvalue of $\sum_{Y_p}$ respectively, and σ is the estimated noise level in the previous iteration. In [28], m is set to 7 and p is decreased from 1 to 0.1 with a step of 0.05.

4. $\sqrt{\lambda_{\boldsymbol{Y}_p,\min}}$ is regarded as the noise level estimation of current iteration. Steps 3 and 4 are iterated until convergence is achieved, e.g., the difference between two successive estimations is less than 1e-6 in [28].

Although the noise level estimation methods [12, 28, 31] were developed for additive Gaussian noise estimation, they can also be used to depict the noise level of other practical noise sources. This is because the noise from an actual camera can be regarded as a mixture of different noise sources, e.g., non-uniformities in sensor structure and thermal noise, and can thus be regarded as Gaussian-like according to the central limit theorem [13]. The experimental results in Section 3.3 will support this statement.

2.2 Image splicing localization

To locate suspicious splicing regions, a test image $\boldsymbol{y}$ is decomposed into N non-overlapping $B \times B$ image blocks for local noise level estimation. As stated earlier, the estimation accuracy is proportional to N, which here equals to $(B-5+1) \times (B-5+1)$. Take $B = 32$ for example: There are a total of $N = (32-5+1) \times (32-5+1) = 784$ patches in each block. Therefore, it is preferred to use larger blocks for better noise level estimation performance. However, for localization precision, smaller blocks are preferred. Hence, we need to make a tradeoff between estimation accuracy and localization precision in selecting an appropriate block size B.

To evaluate the estimation performance on different block size B, we randomly crop image blocks from BOSSraw database with the size of 16 × 16, 32 × 32, 64 × 64 and 128 × 128 pixels, respectively. For each block size, we produce 12,000 noise-added image blocks by adding zero mean Gaussian noise with standard deviation $\sigma = 5$. The means and standard deviations of the estimated noise levels with various sizes are shown in Table 1. It is observed that the mean value for $B = 16$ has clearly deviated from the ground truth$\sigma = 5$. We therefore choose $B = 32$ or 64 to balance noise level estimation accuracy and splicing localization precision in this paper.

To get a more precise localization of the spliced region, we follow a two-phase coarse-to-fine strategy as [23]. The test image $\boldsymbol{y}$ is first segmented into 64 × 64pixel blocks for initial detection. K-means algorithm is adopted to classify the image blocks into two clusters, and the cluster with fewer blocks is treated as the spliced region. In the second phase, each 64 × 64pixel block is further segmented into four32 × 32pixel blocks to estimate local noise level respectively. Therefore, for each pixel in $\boldsymbol{y}$, there are two versions of noise level estimation. NL_{64} is the estimate when $B = 64$ and NL_{32} is the estimate when $B = 32$. For the unsuspicious regions located in the initial detection, NL_{64} is regarded as the final estimate. For the suspicious regions, the final estimate is a weighted sum of the two estimates as in [23].

$$NL = 0.8\, NL_{64} + 0.2\, NL_{32} \tag{4}$$

The idea behind such a weight allocation is quite intuitive since the estimation with larger block size is more reliable. However, we find that this is not the case along the boundary, where the blocks usually include both the original and the spliced regions. As a result, the estimation with small block size is favorable along the boundary. Hence, we take the estimate of $B = 32$ as the final estimation for the blocks along the boundary. The blocks along the boundary are defined as the blocks whose detection result of the first phase is different from any of its eight neighbors, i.e., four HV-neighboring blocks and four diagonal-neighboring blocks. Figure 3 shows an example of the blocks along the boundary. Figure 3(a) is the detection result of the first phase, in which the white blocks indicate the suspicious region and

Table 1 The means and standard deviations of the estimated noise levels with different block size. The ground truth $\sigma = 5$

Block Size B	16	32	64	128
Mean	3.405	4.563	4.908	5.021
Standard deviation	0.457	0.316	0.221	0.140

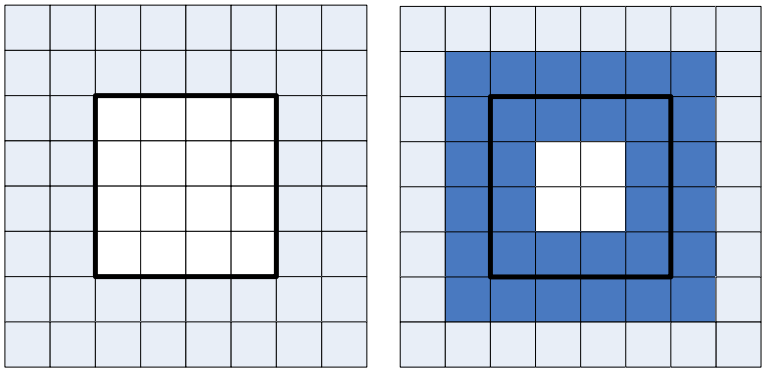


Fig. 3 Illustration of the blocks along the boundary. **a** The detection result of the first phase. **b** The blocks along the boundary (*blue blocks*). The *bold line* denotes the boundary obtained in the first phase

gray blocks indicate the unsuspicious region. The blue blocks in Fig. 3(b) are the blocks along the boundary, for which NL_{32} is used as the final estimate in our scheme. After the final local noise level estimation for each block is obtained, these blocks are classified into two clusters using k-means algorithm based on their noise levels again.

Figure 4 illustrates the necessity of our modification. Figure 4(a) is a spliced image, where a bird is spliced from another image. During the splicing, the bird is processed by adding zero mean Gaussian noise with standard deviation $\sigma = 5$ (please see Fig. 5 for more information). Figure 4(b) shows the detection results for $B = 64$ when zoomed in on the head of the bird. It is observed that the eye and beak of the bird are undetected. Figure 4(c) shows the detection results using (4), where the final estimations on the boundary are the weighted sum of the two estimates with $B = 64$ and $B = 32$. It is observed the beak of the bird is still missing detected. Figure 4(d) shows the detection results when only NL_{32} is used along the boundary, where the beak of the bird is better detected.

3 Experimental results

In this section, we evaluate the proposed method experimentally. First, the proposed method is compared with three state-of-the-art image splicing localization methods on images where artificial noise is added in spliced regions. Second, as a more realistic test, we will show that

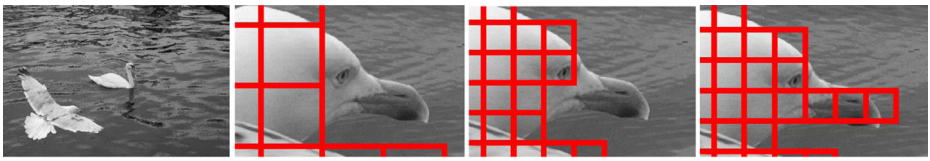


Fig. 4 Illustration of the coarse-to-fine splicing detection. **a** The spliced image. **b** Initial detection with $B = 64$ zoomed in on the head of the bird, (**c**) refined detection using (4), and (**d**) refined detection when only NL_{32} is used along the boundary

the proposed method is also effective in detecting image splicing when the image noise is introduced unintentionally.

3.1 Comparison with related works

We first compare the proposed method with three state-of-the-art noise-based splicing localization methods [21–24] on two spliced images created by *Adobe Photoshop*. The images involved in this subsection are from the BOSSraw database with the original size larger than 8 megapixels (Mp). During the splicing, the cropped region is processed by adding zero mean Gaussian noise with standard deviation $\sigma = 1$, 2 and 5 respectively. We do not consider the case that $\sigma > 5$, since it may introduce visual artifacts which can be easily detected by human inspection. For the method in [22], we choose $B = 32$. For the proposed method, we choose $B = 64$ in the initial detection and choose $B = 32$ in the refined detection as in [23]. To make the localization results more practical, small isolated suspicious regions whose size is less than a threshold T are removed in all the four methods. We set $T = 4$ blocks for the methods in [22, 23] and the proposed method, and set $T = 100$ pixels for the method in [21, 24]. Note that the methods in [22, 23] and the proposed method achieve block-level localization precision, and the method [21, 24] achieves pixel-level precision.

Figure 5 shows an image splicing example, in which a bird from one image is spliced into another image. The original images involved in such a forgery are shown in the top row. From the second row to the bottom row, the left column shows the spliced images with $\sigma = 1$, 2 and 5, respectively. The detection results of the proposed method, the method [22], the method [23] and the method [21, 24] are shown in the second, third, fourth and fifth column, respectively. The true positives, where the spliced regions (bird) are correctly detected, are marked in green. The false positives, where the original regions (background) are falsely detected, are marked in red. Overall, all four methods are able to provide meaningful clues about the forgeries when $\sigma = 5$, but the detection results of the proposed method have fewer false positives. When

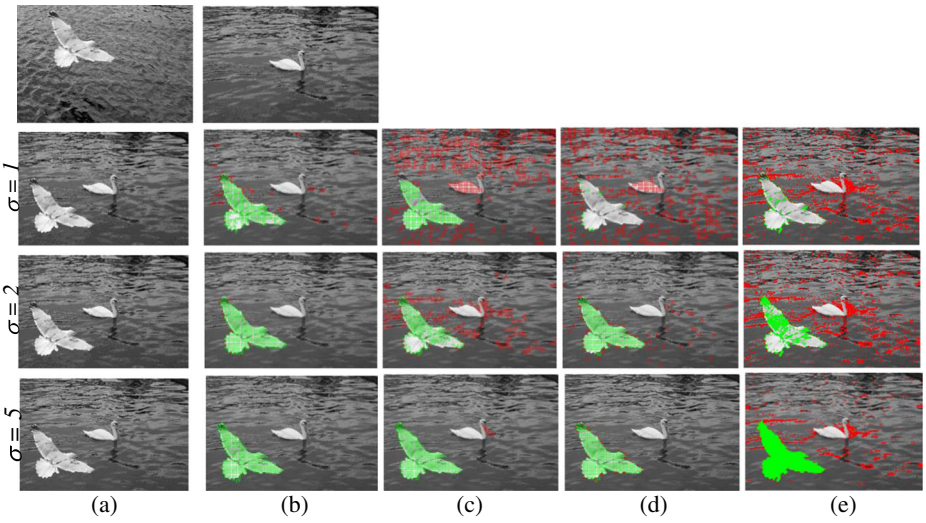


Fig. 5 Image forgeries tampered with different levels of noise addition (**a**), and the detection results using the proposed method (**b**), the method in [22] (**c**), the method in [23] (**d**), and the method in [21, 24] (**e**) respectively. The true positives are marked in *green* and the false positives are marked in *red*

the added noise is small, e.g., $\sigma = 1$, the method [22] has a higher number false positives than that of the proposed method, and the methods in [21, 23, 24] deteriorate more significantly and fail to locate the spliced region.

Figure 6 shows another example, in which a climber from one image is spliced into another image. The original images involved in such a forgery are shown in the left column. From top to bottom, the second column shows the spliced images with $\sigma = 1$, 2 and 5, respectively. The detection results with the proposed method, the method [22], method [23] and the method [21, 24] are shown, respectively, in the third, fourth, fifth and sixth column. All four methods tend to have false positives in the bush region as the rich textures in this region would impair the reliability of the noise-based methods. Nonetheless, it can be observed that the proposed method provides more accurate information about the splicing location in all the cases, especially when the added noise is small.

Secondly, we make a quantitative comparison between the proposed method and other state-of-the-art methods on 200 images chosen from the BOSSbase database [3],[1] with an original size of 512×512 pixels. The spliced images are generated by adding zero mean Gaussian noise to a randomly selected square region of size 192×192 pixels, which corresponds to 14 % of the original image size. The used standard deviation σ is from 1 to 10 with step size of 1, resulting in a total 2000 simulated images.

We use the *block detection accuracy* (BDA) and the *block false positive* (BFP) to evaluate the performance of the splicing localization method, as in [23]. BDA is the probability of images blocks in the spliced region that are correctly detected and BFP is the probability of image blocks in the original region that are falsely detected. For the method [21, 24], which achieves pixel-level precision, we take each pixel as a block in computing BDA and BFP. An effective splicing localization method is expected to achieve high BDA and low BFP simultaneously. Figure 7 shows the average BDA/BFP rates over all 200 images at each noise level. The red square lines represent the proposed method, the black circle lines represent the method in [22], the blue rhombus lines represent the method in [23] and the green star lines represent the method in [21, 24]. It appears that the method in [22] performs better than the method in [23] for large values of$\sigma$and is inferior to it for small values ofσ. The method in [21, 24] tends to have more false positives than other methods. The proposed method consistently performs the best among the four methods, especially when $\sigma$is low. Take $\sigma = 3$as an example: The proposed method achieves BDA = 91.0 % and BFP = 31.2 %, while the method in [22] achieves BDA = 74.6 % and BFP = 51.0 %, the method in [23] achieves BDA = 77.6 % and BFP = 44.0 % and the method in [21, 24] achieves BDA = 76.8 % and BFP = 50.5 %.

3.2 Robustness against common post-processing operations

In this subsection, we evaluate the proposed method on forged images that have undergone some common image post-processing operations, e.g., JPEG compression, geometrical transformation and blurring of the splicing boundary. The standard deviation of the added noise is fixed at $\sigma = 5$.

Figure 8(a) and (b) show the results for the same forged image in Fig. 5 after JPEG compression with different quality factors (QF). It is observed that the proposed method performs satisfactorily when QF = 95, and that the performance degrades significantly when QF = 85. This is because the noise that the noise-based methods based on is significantly

[1] In [3], the authors constructed two uncompressed databases for steganalysis test. One is called BOSSraw, which contains 10,000 original size color images. The other is called BOSSbase, which contains 10,000 grayscale images with a size of 512 × 512 pixels.

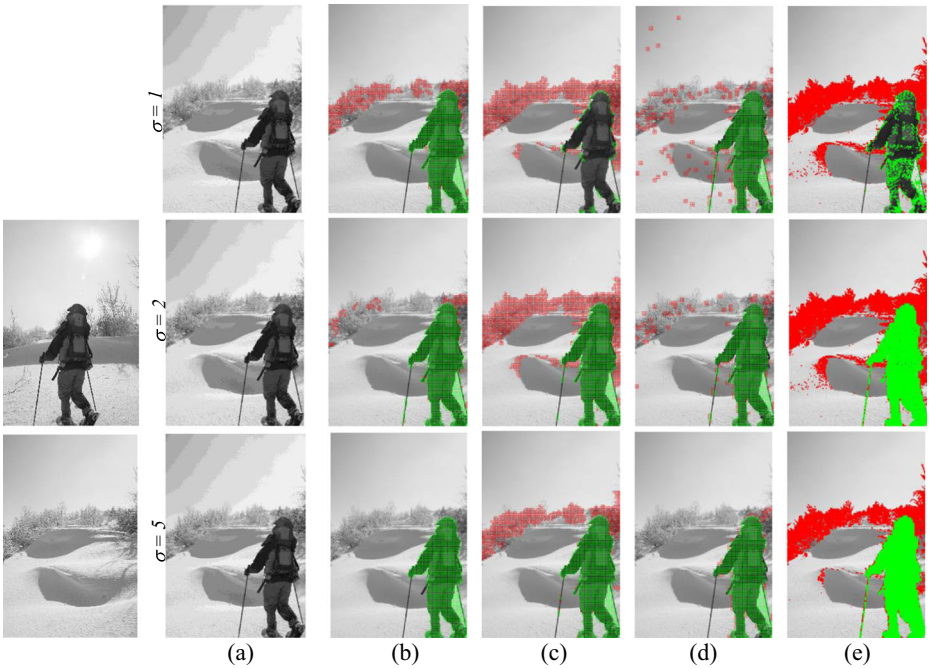


Fig. 6 An image splicing example with different levels of noise addition (**a**), and the detection results using the proposed method (**b**), the method in [22] (**c**), the method in [23] (**d**), and the method in [21, 24] (**e**) respectively. The true positives are marked in *green* and the false positives are marked in *red*

affected by heavy compression. However, by studying various JPEG-based forensic methods available in the literature [4, 5], we suggest that this limitation can be overcome by combining the noise-based methods with the JPEG-based methods. Figure 8(c) and (d) show the results after rescaling using bicubic interpolation with different factors. Figure 8(e) and (f) show the results after rotation with different degrees. It can be observed that the proposed method is only marginally affected by geometrical transformations. In practice, the boundary between the original region and the spliced region is usually blurred by the forger to keep the forged image visually pleasant. Hence, in Fig. 8(g) and (h), we also tested the proposed method on the forged images with Gaussian blurring on the splicing boundary. It is observed that the proposed method is resistant to such post-processing.

3.3 Evaluation on public image splicing dataset

As a more realistic test, we applied the proposed method to detect image splicing on the *Columbia uncompressed image splicing detection evaluation dataset* [15]. The authentic images of this database are taken with four cameras: Canon G3, Nikon D70, Canon 350D Rebel XT and Kodak DCS330. In total, 180 spliced images are created from the authentic ones by splicing salient objects in Adobe PhotoShop. The spliced images are of a size of approximately 1000 × 700pixels.

Since the spliced regions are taken with different cameras and camera settings compared to the original regions, they may have different noise levels that can be used to detect the splicing. Four examples of the spliced images, together with their detection results using different methods, are shown in Fig. 9. Pixel-level quantitative comparison is provided in Table 2,

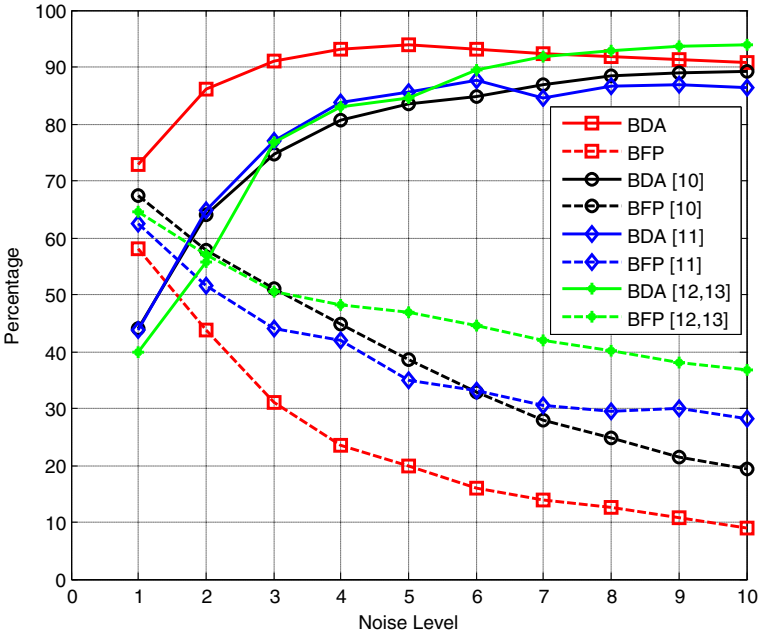


Fig. 7 BDA/BFP rates comparison for image splicing localization. The introduced noise levels on the tampered regions are from 1 to 10. The *red square lines* represent the proposed method and the other lines represent state of the art methods

where *TP* denotes the true positive rate and *FP* denotes the false positive rate. Experimental results show that the proposed method provides accurate information of the splicing forgery in all four cases, whereas the other methods may fail in some cases. This is because the noise level difference between the original region and the spliced region in this dataset is typically relatively small, making it difficult for the preceeding methods to distinguish between them.

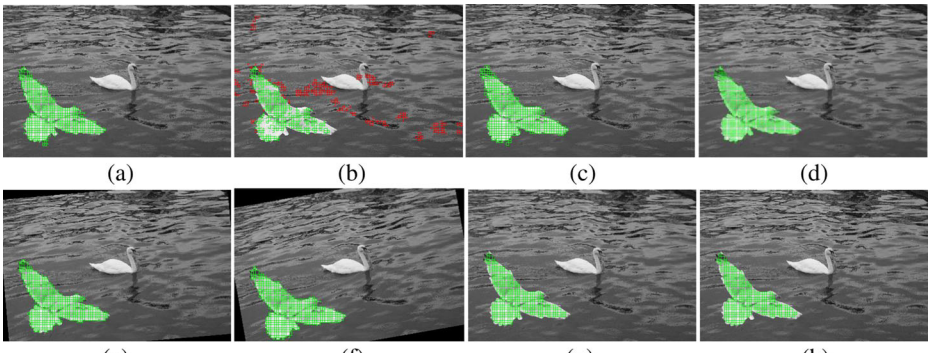


Fig. 8 Splicing localization results of the proposed method for different image post-processing. **a** JPEG compression, QF = 95, (**b**) JPEG compression, QF = 85, (**c**) down-sampling by a factor of 20 %, (**d**) up-sampling by a factor of 20 %, (**e**) rotated by 5°, and (**f**) rotated by 10°. **g** Gaussian blurring on the splicing boundary, 3 × 3 window size and the standard deviation is 0.5, (**h**) Gaussian blurring on the splicing boundary, 5 × 5 window size and the standard deviation is 1. In this subsection, we evaluate the proposed method on forged images that have undergone some common image post-processing operations, e.g., JPEG compression, geometrical transformation and blurring of the splicing boundary. The standard deviation of the added noise is fixed at $\sigma = 5$

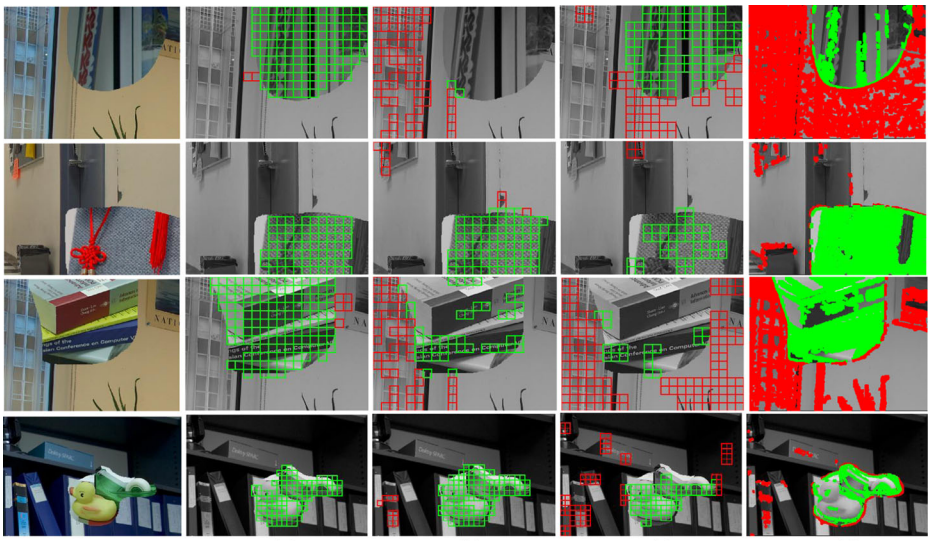


Fig. 9 Detection results for four image splicing samples from the *Columbia uncompressed image splicing detection evaluation dataset* [15] (numbered from top to bottom, No.1, No. 2, No.3, and No. 4). **a** the proposed method, (**b**) the method in [22], (**c**) the method in [23], (**d**) the method in [21, 24]. The true positives are marked in *green* and the false positives are marked in *red*

Note that no artificial noise is involved in the splicing process. These results verified our hypothesis that images from different sources tend to have different intrinsic noise levels, which can be used as a clue for image splicing localization. We also compared the related methods through application to the whole *Columbia uncompressed image splicing detection evaluation dataset* before and after common post-processing. The average pixel-level quantitative comparison is reported in Table 3. Note all the noise-based methods may fail on some forged images where there is no distinguishable noise level difference from the source images, and that it is often difficult to make a fair comparison for the failed cases. Nonetheless, the proposed method outperforms the other noise-based methods in general, according to the experimental results.

Upon careful examination, we find that the differences in noise levels are not necessarily associated with the differences in the cameras used. The authentic images from the same camera can have distinct noise levels, whereas the authentic images from different cameras may have similar noise levels. Hence, the noise level differences may attribute to different

Table 2 Pixel-level performance comparison (%) for the four test images in Fig. 9. Best results are in bold

	No. 1 image		No. 2 image		No. 3 image		No. 4 image	
	TP	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*
Proposed	**97.0**	**4.1**	70.2	**0**	**93.4**	**5.1**	80.2	**1.0**
[22]	0.9	32.2	**75.2**	4.0	26.4	25.0	**87.8**	3.2
[23]	80.7	25.1	38.4	2.2	11.5	66.3	61.6	10.0
[21, 24]	16.5	59.3	69.3	3.8	60.1	32.0	32.0	2.6

Table 3 Pixel-level performance comparison (%) for the *Columbia uncompressed image splicing detection evaluation dataset*, before and after post-processing. Best results are in bold

	No post-processing		JPEG, QF = 95		down-sampling 20 %	
	TP	*FP*	*TP*	*FP*	*TP*	*FP*
Proposed	**47.9**	**18.5**	22.0	**10.8**	**46.0**	23.5
[22]	30.8	21.3	29.6	14.9	36.2	**18.9**
[23]	40.9	31.6	32.2	24.4	29.7	19.1
[21, 24]	36.8	23.0	**37.7**	20.7	32.0	20.9

camera settings, e.g., ISO number [25]. Unfortunately, most of the EXIF information is unavailable for the images in the *Columbia uncompressed image splicing detection evaluation dataset* and we thus can not further confirm our hypothesis for this dataset. Alternatively, we create a spliced image with two source images from the BOSSraw database. The images used are taken with a same Canon EOS 400D camera but distinct ISO settings. Figure 10(a) shows one source image taken with ISO100, and Fig. 10(b) shows the other source image taken with ISO1600. Figure 10(c) shows the forged image where part of the floor in Fig. 10(b) is spliced into Fig. 10(a). Though no artificial noise is added in this splicing process, the proposed method can locate the forgery accurately, as observed in Fig. 10(d).

It is worth pointing out that we cannot expect such noise-based methods to detect all potential forgeries in a real scenario, due to the variety of the image splicing. However, as stated in [21–24], the noise-based methods can give the forensic investigator important clues about the potential forgery when combined with other detection methods.

3.4 Time complexity

Finally, the time complexity of the noise-based methods is compared in terms of the running time. The simulations were performed with Matlab on a computer with a 3.1 GHz CPU and with 4 GB RAM. Average running time per image from the *Columbia uncompressed image splicing detection evaluation dataset* is shown in Table 4. Due to the low-complexity nature of the MAD noise level estimator, the method [22] is the fastest. The proposed method ranks

Fig. 10 An image splicing sample from two source images with distinct ISO settings. **a**, **b** two source images from the BOSSraw database, (**c**) the spliced image, (**d**) detection results with the proposed method

Table 4 Time comparison (s) of the image splicing localization methods

Method	Proposed	[22]	[23]	[21, 24]
Running time	4.7	0.42	250.5	94.2

second. However, several seconds for testing an image is completely acceptable in practice. The method [23] is time-consuming as its noise level estimator sometimes becomes divergent for low noise level case, which has also been illustrated in [21, 24]. The efficiency bottleneck of the method [21, 24] lies in the clustering phase, which makes it time consuming as well. It is worthwhile to remind that the first three methods are implemented at block-level, whereas the method [21, 24] is implemented at pixel-level. If we apply the block-level methods on the pixel-level with overlapping blocks, the time complexity would be much higher.

4 Conclusions

In this work, we have proposed a simple yet effective image splicing localization method. Based on the assumption that spliced regions and original regions in a forged image tend to have different noise levels, we perform a blockwise local noise level estimation in the questioned image. The block size B can be adjusted according to the specific requirement, e.g., small block size for high splicing localization precision. In this study, we choose $B = 32$ or $B = 64$ to balance the noise level estimation accuracy and the splicing localization precision. The adopted noise level estimation method is based on a sparse representation assumption over noise-free images (or image blocks in our case), which is typically true due to local similarity of natural images. K-means algorithm is used to categorize the blocks into two clusters. The cluster including fewer blocks usually provides a meaningful clue about the spliced region. A coarse-to-fine strategy is used and the regions along the boundary are carefully handled, which significantly improves the localization precision. Experimental results on different datasets have shown the superiority of the proposed method.

Compared to the operation oriented image splicing localization methods [2, 4, 17, 18, 27], the proposed method has fewer assumptions and can be applied in more forensic scenarios. Compared to the existing noise-based image splicing localization works [21–24], the proposed method achieves improved performance, especially when the noise difference between the spliced region and the original region is small. This improvement is critical because such difference in image forgeries of practical interest usually is relatively small. However, the proposed method shares some limitations with other noise-based methods in that it is not robust enough for JPEG compression, and it does not work when the forged image is spliced with source images of similar noise levels.

In this study, k-means algorithm is used to cluster the noise level estimation results without taking the geometric information into consideration. Fully exploiting the geometric information in clustering is likely to further improve the image splicing localization performance and this will be our future work.

Acknowledgments We would like to thank the authors of [21, 23, 24] for kindly sharing codes and discussing the implementation detail.

This work was supported by NSFC (Grant nos. 61379155, U1536204, 61332012, and 61502547), and NSF of Guangdong province (Grant no. s2013020012788).

References

1. Al-Qershi OM, Khoo BE (2013) Passive detection of copy-move forgery in digital images: state-of-the-art. Forensic Sci Int 231:284–295
2. Bahrami K, Kot AC, Li L, Li H (2015) Blurred image splicing localization by exposing blur type inconsistency. IEEE Trans Inf Forensics Secur 10(5):999–1009
3. Bas P, Filler T, Pevny T (2011) 'Break our Steganographic System': the ins and outs of organizing BOSS. Filler T et al. (Eds.): IH 2011. LNCS 6958:59–70
4. Bianchi T, Piva A (2012) Image forgery localization via block-grained analysis of JPEG artifacts. IEEE Trans Inf Forensics Secur 7(3):1003–1017
5. Chen YL, Hsu CT (2011) Detecting recompression of jpeg images via periodicity analysis of compression artifacts for tampering detection. IEEE Trans Inf Forensics Secur 6(2):396–406
6. Chen M, Fridrich J, Goljan M, Lukas J (2008) Determining image origin and integrity using sensor noise. IEEE Trans Inf Forensics Secur 3(1):74–90
7. Chierchia G, Parrilli S, Poggi G, Sansone C, Verdoliva L (2010) On the influence of denoising in PRNU based forgery detection. In Proc. of the 2nd ACM workshop on Multimedia in Forensics, Security and Intelligence, pp. 117–122
8. Chierchia G, Cozzolino D, Poggi G, Sansone C, Verdoliva L (2014a) Guided filtering for PRNU-based localization of small-size image forgeries, in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process., pp. 6272–6276
9. Chierchia G, Poggi G, Sansone C, Verdoliva L (2014b) A bayesian-MRF approach for PRNU-based image forgery detection. IEEE Trans Inf Forensics Secur 9(4):554–567
10. Cozzolino D, Gragnaniello D, Verdoliva L (2014) Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques, In Proc. of IEEE Int. Conf. Image Processing, pp. 5302–5306
11. Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095
12. Donoho D, Johnstone I (1994) Ideal spatial adaption by wavelet shrinkage. Biometrika 8:425–455
13. Faraji H, MacLean WJ (2006) Ccd noise removal in digital images. IEEE Trans Image Process 15(9):2676–2685
14. He K, Sun J, Tang X (2013) Guided image filtering. IEEE Trans Pattern Anal Mach Intell 35(6):1397–1409
15. Hsu YF, Chang SF (2006) Detecting image splicing using geometry invariants and camera characteristics consistency. In International Conference on Multimedia and Expo, pp. 549–552
16. Huang DY, Huang CN, Hu WC, Chou CH (2015) Robustness of copy-move forgery detection under high JPEG compression artifacts, Multimed. Tools Appl., published online
17. Kang X, Stamm MC, Peng A, Liu KJR (2013) Robust median filtering forensics using an autoregressive model. IEEE Trans Inf Forensics Secur 8(9):1456–1468
18. Lin X, Li CT, Hu Y (2013) Exposing image forgery through the detection of contrast enhancement, In Proc. of IEEE Int. Conf. Image Processing, pp. 4467–4471
19. Liu X, Tanaka M, Okutomi M (2012) Noise level estimation using weak textured patches of a single noisy image. In Proc. of IEEE Int. Conf. Image Processing, pp. 665–668
20. Lukas J, Fridrich J, Goljan M (2006) Detecting digital image forgeries using sensor pattern noise. In SPIE Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents VIII, 6072: 362–372
21. Lyu S, Pan X, Zhang X (2014) Exposing region splicing forgeries with blind local noise estimation. Int J Comput Vis 110(2):202–221
22. Mahdian B, Saic S (2009) Using noise inconsistencies for blind image forensics. Image Vis Comput 27(10): 1497–1503
23. Pan X, Zhang X, Lyu S (2011) Exposing image forgery with blind noise estimation. In the 13th ACM Workshop on Multimedia and Security, pp. 15–20
24. Pan X, Zhang X, Lyu S (2012) Exposing image splicing with inconsistent local noise variances. In IEEE International Conference on Computational Photography (ICCP), pp. 1–10
25. Petteri M (2008) Dependence of the parameters of digital image noise model on ISO number, temperature and shutter time. Project work report, Dec. 2008, Tampere, Finland. [online] http://www.cs.tut.fi/~foi/MobileImagingReport_PetteriOjala_Dec2008.pdf
26. Popescu C, Farid H (2004) Statistical tools for digital forensics, in 6th International Workshop on Information Hiding. Toronto, Canada, pp. 128–147
27. Popescu C, Farid H (2005) Exposing digital forgeries by detecting traces of resampling. IEEE Trans Signal Process 53(2):758–767
28. Pyatykh S, Hesser J, Zheng L (2013) Image noise level estimation by principal component analysis. IEEE Trans Image Process 22(2):687–699
29. Stamm MC, Liu KJR (2011) Anti-forensics of digital image compression. IEEE Trans Inf Forensics Secur 6(3):1050–1065

30. Zhu Y, Shen X, Chen H (2016) Copy-move forgery detection based on scaled ORB. Multimed Tools Appl 75:3221–3233
31. Zoran D, Weiss Y (2009) Scale invariance and noise in nature image. In Proc. of IEEE International Conference on Computer Vision, pp. 2209–2216

Hui Zeng received the B.E. and M.S. degree in communication engineering from Nanjing University of Posts and Telecommunications (NUPT), Nanjing, China, in 2004 and 2007 respectively. He received the Ph.D. degree in School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, in 2016. His research interests include multimedia forensics and game theory.

Yifeng Zhan received the Bachelor degree in electronic engineering from South China University of Technology (SCUT), Guangzhou, China, in 2014. He is currently pursuing the Master Degree in School of Data and Computer science, Sun Yat-sen University, Guangzhou, China. His research interests include multimedia forensics and deep learning.

Xiangui Kang is currently a professor with the School of Data and Computer science, Sun Yat-sen University, Guangzhou, China. He visited University of Maryland, College Park, USA during Aug. 2011 – Aug. 2012, and New Jersey Institute of Technology during Aug. 2004-Sept. 2005. His research interests include information forensics and security, game theory, multimedia communications and security.

Xiaodan Lin is currently pursuing the Ph.D. degree in School of Date and Computer science, Sun Yat-sen University, Guangzhou, China. Her research interests include multimedia forensics.