J. Vis. Commun. Image R. 38 (2016) 195–206

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Multi-scale noise estimation for image splicing forgery detection ☆

Chi-Man Pun *, Bo Liu, Xiao-Chen Yuan

Department of Computer and Information Science, University of Macau, Macau

ARTICLE INFO

Article history:
Received 23 October 2015
Revised 14 January 2016
Accepted 2 March 2016
Available online 4 March 2016

Keywords:
Splicing forgery
Multi-scale noise estimation
Noise level function
SLIC superpixels
Optimal Parameter Combination Searching (OPCS)

ABSTRACT

Noise discrepancies in multiple scales are utilized as indicators for image splicing forgery detection in this paper. Specifically, the test image is initially segmented into superpixels of multiple scales. In each individual scale, noise level function, which reflects the relation between noise level and brightness of each segment, is computed. Those segments not constrained by the noise level function are regarded as suspicious regions. In the final step, pixels appears in suspicious regions of each scale, after necessary morphological processing, are marked as spliced region(s). The Optimal Parameter Combination Searching (OPCS) Algorithm is proposed to determine the optimal parameters during the process. Two datasets are created for training the optimal parameters and to evaluate the proposed scheme, respectively. The experimental results show that the proposed scheme is effective, especially for the multi-objects splicing. In addition, the proposed scheme is proven to be superior to the existing state-of-the-art method.

1. Introduction

Due to the wide availability of picture editing software, normal individual can manipulate the digital image quickly and easily. Therefore anyone can hardly tell if an image is original or fake when seen in the Internet. On most circumstances it does not matter, but in some special cases, it makes difference. A famous fake picture can affect public opinions and even worse a fake picture as an evidence in the court may lead to irreparable damage. Accordingly an effective and robust image forensic method is of great importance. However there are so many kinds of operations can be added to a digital image such as contrast enhancing [1], copy-move [2] and splicing [3], and a single algorithm to deal with various types of attacks is very challenging though there are some attempts [4]. In order to achieve better detection accuracy, in this paper, we will focus on a common operation, splicing, which refers one or more regions of a picture come(s) from the other pictures. This type of manipulation is widely seen in our daily lives, because these fake pictures often contains huge information which people will be interested in. Existing splicing forgery detection algorithms will be briefly reviewed in the following.

In order to make tampered pictures look real, some operations such as geometrical distortions and contrast altering of spliced area, will be applied to the image, and these manipulations can be detected as evidence of splicing forgery. Mahdian and Saic [5] analyzed statistical changes of signal by interpolation process which is involved with scaling, rotation and skewing of inserted regions. Wei et al. [6] extended Mahdian and Saic's work and their method can estimate the image rotation angles. However, the interpolation-related methods are vulnerable to low quality JPEG compressions and cannot locate very small spliced areas. Detection of contrast enhancement was proposed by Cao et al. [1], however, it is not robust to JPEG compressions as well.

Since JPEG format is widely used and composing two images together usually involve the operation of double JPEG compression, methods based on detecting double compression have been proposed. Farid [7] proposed the method to detect if the part of an image is firstly compressed at a lower quality. And algorithm proposed by Bianchi and Piva [8] was to deal with more scenarios, as first and second quantization table are different. And Valenzise et al. [9] proposed forward quantization noise in JPEG compression to deal with high-quality compressed images in forensics. These JPEG-based methods rely on comparatively strict hypothesis. And multi-compression is common but too complex to analyze. Most of these methods become ineffective when pictures are compressed more than two or three times. In addition, block artifact grids (BAG) caused by block processing during JPEG compression can be used in splicing detection [10]. Besides JPEG compression, contents of pictures also provide traces for forensics. Inconsistent of motion blur, which is produced by camera vibrations or insufficient shutter speed, provides evidence for detecting splicing forgery [11,12]. Illumination environment in pictures also present

☆ This paper has been recommended for acceptance by Susanto Rahardja.

* Corresponding author.
E-mail addresses: cmpun@umac.mo (C.-M. Pun), yb57413@umac.mo (B. Liu), xiaochen_yuan@ieee.org (X.-C. Yuan).

http://dx.doi.org/10.1016/j.jvcir.2016.03.005
1047-3203/

some consistency: directions of lights [13], shadows [14] and illumination colors [15] can be estimated and used as cues.

Apart from user's operations and image contents, the digital camera including optical lens will leave some information in the final picture and they can be used for exposing forgery. Modern digital camera uses Color Filter Array (CFA) overlaid on image sensor to catch RGB colors and utilizes demosaicing algorithm to reconstruct full color images. The local absence of demosaicing artifacts is regarded as tampering in [16]. Chromatic Aberration (CA), and produced by imperfect optical system, can be modeled and inconsistent CA across the image will be a forgery indicator [17]. The property that mapping input irradiance to output image intensity, called as Camera Response Function (CRF) can also be used in forgery detection [18]. The image is randomly segmented and CRF of each segment is estimated, and inconsistent across the image is regarded as splicing. But according to the experimental results, the accuracy of locating forged area is modest. Besides the fingerprints of image sensor, the characters of curtain lenses can also be used. Fu and Cao [19] studied the geometric distortions of wide angle lenses and objects without consistent distortion is detected as inserted regions. However, the work is limited to images captured with very wide angle lens since normal lenses shows little distortion which is hardly calculated. Besides CA and CRF, Photo Response Non-Uniformity (PRNU), which contributes to pattern noise in digital cameras [20,21], can also be formulated to identify a series of sensors. However it is very time-consuming and unpractical to compute PRNU for each camera model. Therefore many researchers investigated the noise patterns in the image to discriminate splicing forgery [22,23].

While most of the previous algorithms as stated above rely on specific and strict hypothesis, our goal is to detect more accuracy splicing forged regions without much assumptions. Considering that noise is widely existed in natural pictures taken by digital camera, and each picture contains certain level and type of noise, which is produced during the whole process when photons coming into the sensor until the camera output the picture [24,25], in this paper, we propose to detect the forgery based on noise estimation. To estimate the noise level of a given picture, the phenomenon of kurtosis concentration [26,27] which kurtosis value of natural images in band-pass filtered domains concentrates to a constant value helps estimate noise variance [28]. However this method is based on the assumption that the intrinsic noise is similar across the original image. The truth is the noise level is not similar within the image, and it is always affected by brightness and textures [29]. And the omission of these trivial but important noise difference has bad impact on the detection result. Noise level function is thus proposed to deal with the noise fluctuations [30]. Since the brightness will influence the noise level in the digital image, and the noise level function can describe the relations of the brightness and noise level, inconsistent with noise level function will be considered as forged.

The paper is organized as follows. In Section 2, the multi-scale estimation method for detecting splicing forgery is introduced. In Section 3, a training dataset is used to find optimal parameters for algorithm termination and a forgery dataset is created to evaluate the proposed method and lots of experiments are conducted to present the performance of the proposed scheme. Besides, comparisons with existing state-of-the-art method are demonstrated to show the superiority of the proposed scheme. Section 4 concludes the paper and states the future work.

2. Multi-scale noise estimation for splicing detection

The existing noise estimation methods are not always reliable and accurate in dealing with small image segments [31,32], since local noise estimation is affected by image segmentation and content. And therefore inaccurate noise estimation influences the following steps of forgery detection. Accordingly we propose multi-scale segmentation scheme to compensate such effects. Instead of segmenting the image with fixed initial size [33], in our method we propose to segment the host image in multiple scales, where a minimum initial size is defined and the initial sizes for the multiple scales are accordingly increased progressively at regular intervals. In this way the noise estimator will be applied into all scales respectively to estimate the noises and finally the algorithm is designed to automatically composite results from all scales together. Apart from more accurate noise estimation results, the multi-scale segmentation plays an important role in detecting multi-objects forgery and especially when the objects are of various sizes. Fig. 1 demonstrates the better detection results of our algorithm comparing with the existing state-of-the-art method proposed by Lyu et al. [33], where the noise estimation method was proposed in single scale to detect the splicing forgeries. In Fig. 1, the 1st row shows the example where multiple objects of various sizes are pasted into the host image (a1) to generate the forged image, as shown in (a2); and the 2nd row shows another example where a pretty large object is pasted into the host image (b1) to form the forged image, as shown in (b2). (a3) and (b3) in the 3rd column show the detected results of Lyu's method, where the *precision* and *recall* rates are evaluated: *precision = 77.45%*, *recall = 18.99%* and (a4) and (b4) in the 4th column show the detected results of the proposed scheme: *precision = 81.14%*, *recall = 94.84%*. It can be easily seen from these two typical examples that the proposed scheme outperforms Lyu's method [33] a lot in both *precision* and *recall*.

Fig. 2 presents the overview of the proposed method. First, the multi-scale image segmentation algorithm is proposed to segment the input image. Given the host image as shown in (a), the multi-scale segmented images are then generated using the multi-scale image segmentation algorithm, as shown in (b). Next, in each scale, the noise level is estimated to build the noise level function, with which, outliers, the segments with comparatively higher or lower noise level, are detected and indicated as suspicious forged area. Afterwards, in each scale, the outliers map which has recorded all outlier segments is updated by combining previously constructed outliers map, and only regions appears in both maps are remained as forged area. Finally, a termination criteria is defined by judging the stability of the outliers maps or whether the iteration times has reached the preset maximum times i_{max}. The following Sections 2.1,2.2 and 2.3 will elaborate, respectively, the multi-scale image segmentation algorithm, the outlier detection using noise level function, and termination condition, especially the method to measure the stability of outliers map.

2.1. Multi-scale image segmentation

This paper introduces a novel multi-scale image segmentation method to segment an image. The traditional image segmentation algorithms usually segment a host image into small clusters with a given initial size. In the proposed method, the multi-scale segmentation method is proposed to segment a host image into successive scales, and the segmentation result in each scale corresponds to that of traditional image segment algorithms, with independent and different initial sizes. During the segmentation, the Simple Linear Iterative Clustering (SLIC) algorithm is applied to the input image to segment it in multiple scales. The reason why we adopt multi-scale segmentation is that the proposed algorithm is designed to detect as many types of forgery as possible. Specifically, the multi-scale image segmentation method can detect forged regions of various size. It is fairly difficult to detect and locate

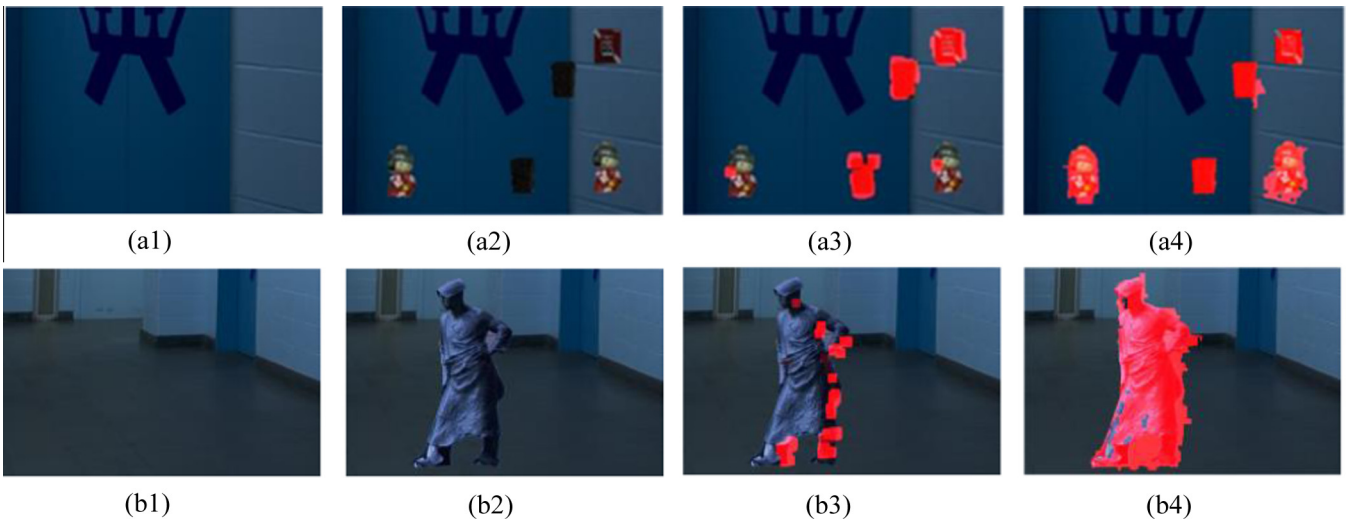

Fig. 1. Demonstration of detection results of Lyu's method and the proposed scheme. (a1) (b1) Original images. (a2) (b2) Forged images. (a3) (b3) Detected results of Lyu's method [33], *precision = 77.45%, recall = 18.99%.* (a4) (b4) Detected results of the proposed scheme, *precision = 81.14%, recall = 94.84%.*

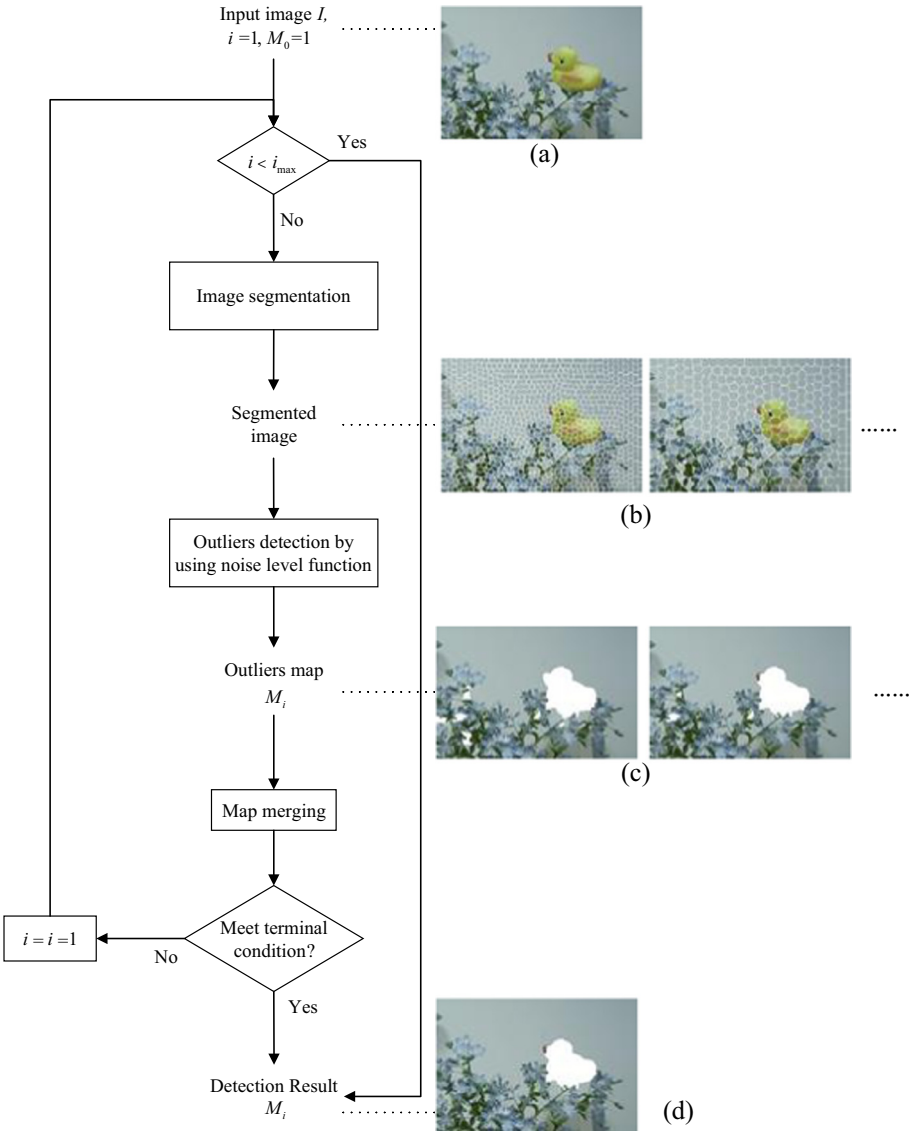

Fig. 2. Flowchart of the proposed scheme.

those objects with various size with the traditional segmentation methods which adopt segmentation algorithm only once. Take SLIC as example, where the initial cluster size is the parameter to control the final size of each segment, when the initial size is too large, the small spliced regions will be submerged in a larger segment, and thus it cannot be detected in the following steps; on the contrary, if the initial size is set to be small enough to decompose all the possible forged regions, noise estimation will be inaccurate and inefficient since small segments contains inadequate pixels and the number of segments is excessive. Therefore, the proposed multi-scale image segmentation can obviously help to detect and locate the forged objects of different size in proper scale.

Formally, given a host image I, to segment I into n scales, the segments satisfy

$$\begin{cases} \bigcup_{j=1}^{J_i} S_j^i = I \\ \bigcap_{j=1}^{J_i} S_j^i = 0 \end{cases} \quad \text{for} \quad i = 1, 2, \ldots, n \tag{1}$$

where S_j^i indicates a segment in the ith scale, where there are J_i segments in total.

Note that segments from different scale should not intentionally the same, namely:

$$S^p \neq S^q \quad \text{for} \quad p \neq q. \tag{2}$$

Many image segmentation algorithms have been proposed in the past year, for example, graph-based segmentation, normalized cuts, turbopixels and so on, and most of them can be used in our method to segment the image in a given scale. In our proposed method, SLIC superpixels [34] is employed for its efficiency and effectiveness, in addition, its segmentation results satisfy both condition (1) and (2). By using SLIC to segment the host image, its initial size can be easily adjusted and thus can yield a series of scales as needed. Two steps are involved in SLIC segmentation algorithm, initialization step the assignment step. In the initialization step, within each regular grid cluster center $C_s = (l_s, a_s, b_s, x_s, y_s)^T$ is assigned by sampling pixels. Then cluster center is relocated to the lowest gradient position in 3×3 neighborhood. In the assignment step, each pixel is associated with its nearest cluster center, which is adjusted as the mean vector of all pixels within the cluster, notating as $(l, a, b, x, y)^T$. Once residual error between the previous and new cluster center is smaller than a given threshold, the algorithm stops.

2.2. Outlier detection using noise level function

Noise level of each segment is used as an indicator in our paper, to expose image forgery. Even if we assume that noise level of spliced regions is different from which of the original regions, separating these regions relying on their noise level only is challenging, because it is hard to choose an appropriate threshold when there are a wide range of various images, and the noise level of which differs tremendously. Although adaptive threshold can be applied, segregating spliced area still remains difficult to be detected since the noise level is not constant across the whole frame even in untouched image. On the other hand, the noise distribution vibrates in the image and relates to the regional brightness. Accordingly, setting a hard threshold will produce many false positive results, and at the same time, the truly forged area may be submerged. For a better description of this property that the noise level varies across the image, the Noise Level Function (NLF) [30] is used in our method.

Noise level function is the standard deviation of the noise as a function of brightness. The ideal noise level function depicts the relationship between local noise and brightness. However, ideal noise level function is hardly to obtain because the accurate estimation of noise is a tough task, and most of time, is impossible. Local noise estimation is vulnerable to improper image segmentation, and complex structures in the area. Thus the noise level function can only be approximated via following two steps: 1. segmental noise estimation, and 2. the curve fitting. The overview of proposed forgery detection by using noise level function is given below.

Algorithm 1. Forgery detection by using noise level function

Input: Segmented image $I = \bigcup S_j$
Output: Outliers map M_i
1: **for** each image segment S_j
2: noise estimation using Weiner filter, stored in $n(S_j)$
3: segment's average brightness calculation, stored in $b(S_j)$
4: **end for**
5: create noise level function f by curve fitting using $n(S_j)$ and $b(S_j)$
6: **for** each image segment S_j
7: calculate probability of genuineness with f, $n(S_j)$ and $b(S_j)$, stored in $P(S_j)$
8: **end for**
9: Cluster $P(S_j)$ into two clusters
10: Mark image segments in smaller cluster, stored in M_i

In the step of local noise estimation, standard deviation of the noise in each segment is computed. In order to successfully locate small spliced objects, the segment size of initial scales cannot be very large. However accurate noise estimation from those limited number of pixels is difficult. In addition, in our proposed multi-scale method, the analysis of the outlier detection will be conducted multiple times accordingly. Therefore, considering that the complex noise estimator is time-consuming, it is not necessary at the expense of efficiency. To make the proposed method more efficient, a single local noise estimation algorithm across all scales is proposed. Therefore a simple and common Weiner filter is used to estimation the local noise level of each segment. To be specific, the standard deviation of noise in each segment is estimated by (3).

$$n\left(S_j^i\right) = \sqrt{\frac{1}{\left|S_j^i\right|} \sum_{n(x,y) \in S_j^i} \left(n(x,y) - \overline{n\left(S_j^i\right)}\right)^2} \tag{3}$$

where $\left|S_j^i\right|$ is the pixel count of the corresponding jth segment in the ith scale; $n(x,y)$ indicates the estimated noise of the host image, and it can be calculated using (4); $\overline{n\left(S_j^i\right)}$ denotes the mean value of the estimated noise of the corresponding segment S_j^i, as defined in (5); and $n\left(S_j^i\right)$ indicates the standard deviation of the corresponding segment S_j^i.

$$n(x,y) = I(x,y) - w * I(x,y) \tag{4}$$

$$\overline{n\left(S_j^i\right)} = \frac{\sum_{n(x,y) \in S_j^i} n(x,y)}{\left|S_j^i\right|} \tag{5}$$

where $I(x,y)$ is the host image, and w is used to indicate the Weiner filter, $*$ denotes the convolution operator, and $n\left(S_j^i\right)$ denotes estimated noise of the segment.

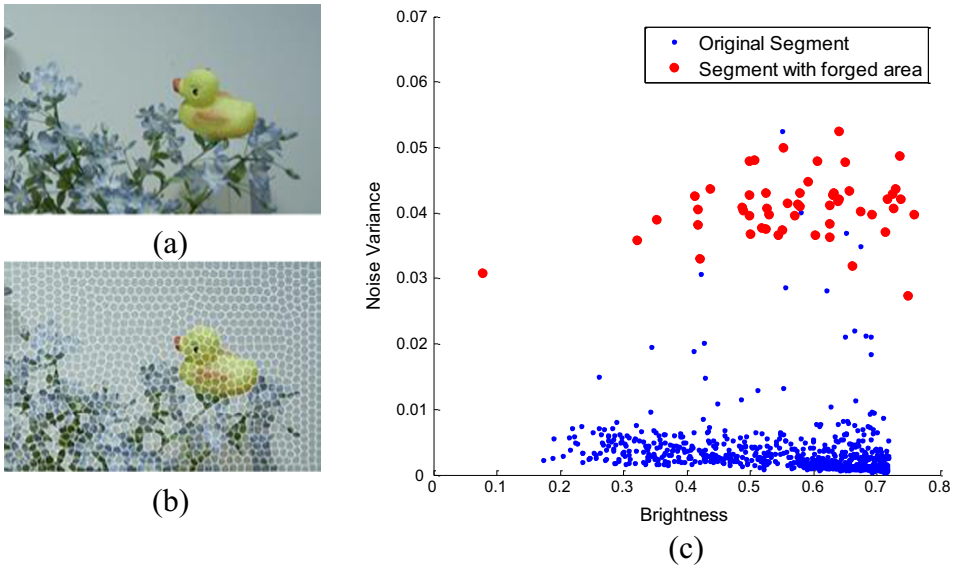

Fig. 3. The difference noise distribution of original and spliced regions. (a) A forged picture with the spliced duck; (b) the segmented image in one scale; and (c) distribution of relationship between noise variance and average brightness of the given segments.

Similarly, the average brightness of the segment are defined in (6).

$$b\left(S_j^i\right) = \frac{\sum I(x,y)}{\left|S_j^i\right|} \tag{6}$$

where $I(x,y) \in S_j^i$ is the pixel grayscale value, $\left|S_j^i\right|$ is the pixel count of the corresponding jth segment in the ith scale, and $b\left(S_j^i\right)$ is average grayscale of the segment.

After calculating the noise level and the average brightness of each segment, we can project these data into the 2-D coordinate to present the relations between noise and brightness within the given image. Fig. 3 displays an example, where (a) shows a forged image with the spliced duck, (b) shows the segmented image in one scale, and (c) shows the distribution of relationship between noise variance and average brightness, and the results notated in red indicates which of the segments with forged regions, while the results notated in blue indicates which of the original segments. It can be seen from the distribution that most of the points appears within the range, while some of the points scatters far from the most. These faraway points denote the segments with significantly higher noise level than they should be, therefore, they are going to be separated in this step.

To reveal the relationship between noise and brightness of segments, noise level function is approximated from the points. Due to the accuracy of noise estimation, measured noise vibrates in a small range as illustrated in Fig. 4. Then, the least squares polynomial curve fitting technique [35] is used to construct noise level function, and five degree fitting shows good result in the experiment. Given the form of the estimated noise level function f as in (7). For simplicity, we perform curve fitting within one scale where there are n segments in total. The residual R is given in (8).

$$f(b) = a_0 + a_1 b + a_2 b^2 + a_3 b^3 + a_4 b^4 + a_5 b^5 \tag{7}$$

where $a_0, a_1, \ldots, a_5$ are coefficients to be calculated, and b is the average grayscale value of the segment.

$$R = \sum_{i=1}^{n} \left[f_i - \left(a_0 + a_1 b_i + \ldots + a_5 b_5^5 \right) \right]^2 \tag{8}$$

By setting partial derivatives to zero, as in (9), the coefficients $a_0, a_1, \ldots, a_5$ in noise level function can be calculated.

$$\frac{\partial(R)}{\partial a_0} = \frac{\partial(R)}{\partial a_1} = \cdots = \frac{\partial(R)}{\partial a_5} = 0 \tag{9}$$

After computing the noise level function, outliers can be separated easily. According to former analysis, the longer distance from the point to the curve, the smaller possibility for originality (namely not spliced) would be. The noise level of segments in certain brightness is assumed to be subject to Poisson distribution, with the greatest probability assigned on the curve and reduced gradually away from the center. For an arbitrary segment S_i in one scale with average brightness $b(S_i)$ and estimated noise $n(S_i)$, the probability of its originality is given in (10).

$$P(S_i) = \frac{e^{-\lambda} \cdot \lambda^{n(S_i)}}{n(S_i)!}, \quad \lambda = f[b(S_i)] \tag{10}$$

Then, all segments are clustered into two sets regarding to their probability of originality and those in the smaller cluster are regarded as potentially spliced regions of the scale, thus they will be recorded in the corresponding outliers map M_i. The proposed method has two main advantages: first, it avoids setting a hard threshold to filter out the outliers; second, the outliers map in each scale contains sufficient information for the following steps. Finally, the outlier map M_i will be updated by intersecting with outlier map M_{i-1} from its previous scale, by $M_i = M_i \cap M_{i-1}$. In this way, those outliers appears successively in multiple scales will be kept.

2.3. Terminal conditions of the algorithm

As growing number of scales have been analyzed, forged objects are located, and false positive regions remaining in the outliers map are eliminated gradually. To preserve truly spliced areas and remove falsely marked regions, two conditions are defined to terminate the algorithm: 1. when outliers map becomes stable, which means that the outlier map of successive scales does not varies largely; and 2. when number of iteration reaches the preset maximum iteration value. In summary, three parameters are defined to judge the stability of the outliers map: the number of clusters,

K; the threshold of the change ratio, τ; and the number of successive scales, C. First, the outliers map M_i is clustered into K clusters, and the distance from each pixel in the cluster to its corresponding cluster center are summed, then, K inner sums are accumulated as $SUMD_i$, with which, the change rate of outliers map between two successive scales can be determined by (11).

$$\tau_i = \frac{|SUMD_i - SUMD_{i-1}|}{SUMD_{i-1}} \tag{11}$$

where $SUMD_i$ and $SUMD_{i-1}$ indicate the accumulated value of the successive ith and $(i-1)$th scale, respectively; and τ_i is the change rate of the successive ith and $(i-1)$th scale.

When τ_i is smaller than the threshold τ in C successive scales, as indicated in (12), the outliers map M_{i+C-1} will be regarded as stable.

$$\tau_i < \tau, \tau_{i+1} < \tau, \ldots, \tau_{i+C-1} < \tau \tag{12}$$

To determine the appropriate values for the three parameters K, τ, and C, we create a training dataset to train the proposed method, thus the proper values can be obtained experimentally. The training helps to select the best combination of parameters from prescribed K, τ, and C values. Since there are a number of potential relationships between the three parameters, we employ the Receiver Operating Characteristic (ROC) curve to show the *recall* and *precision* on each subset, thus to determine in which combination we can get the best results. On normal circumstances, optimal parameter can be easily obtained from ROC. However in our case, each subset has its own curve and optimal parameter combination. Therefore it is very important and necessary that we should find a global parameter combination which maximizes the *precision* and *recall* of all the subsets. Consequently, we propose a new algorithm, the Optimal Parameter Combination Searching (OPCS) Algorithm, to find the optimal combination of K, τ, and C. The detail steps of OPCS are explained as follows in Algorithm 2.

Algorithm 2. Optimal Parameter Combination Searching (OPCS) Algorithm

Input: ROC curves A_j
Output: Optimal Parameters $Cmb = \{K, C, \tau\}$
1: **for** each ROC A_j
2: **for** each point on ROC $a_j^i = \{p, r | K, C, \tau\} \in A_j$
3: Calculate the distance from $a_j^i\left(p_j^i, r_j^i\right)$ to $Idl(p_{idl}, r_{idl})$, as $Dist_j^i$
4: **end for**
5: **end for**
6: **for** each A_i
7: Sort A_i in ascending order according to $Dist_j^i$, and store as $A\_sort_i = [a_{1,j}, a_{2,j}, \ldots, a_{N\_Cmb,j}]$
8: **end for**
9: **for** $j = 1 : N\_Cmb$
10: **if** $\bigcap_{j=1}^{N\_S}\bigcap_{i=1}^{N\_Cmb} a_{i,j}(K, C, \tau) \neq \varnothing$
11: Optimal Parameter Combination $\bigcap_{j=1}^{N\_S}\bigcap_{i=1}^{N\_Cmb} a_{i,j}(K, C, \tau) \rightarrow \{K, C, \tau\}$
12: **return** $Cmb = \{K, C, \tau\}$
13: **end if**
14: **end for**

In OPCS algorithm, first, for ROC curve of each subset, A_j, $j = 1, 2, 3, \ldots, N\_S$, the distance of point $a_j^i\left(p_j^i, r_j^i\right)$ in the jth subset to the ideal condition $Idl(p_{idl}, e_{idl})$, which means both *precision* and *recall* reach 100%, $p_{idl} = 1$ and $r_{idl} = 1$, are calculated using

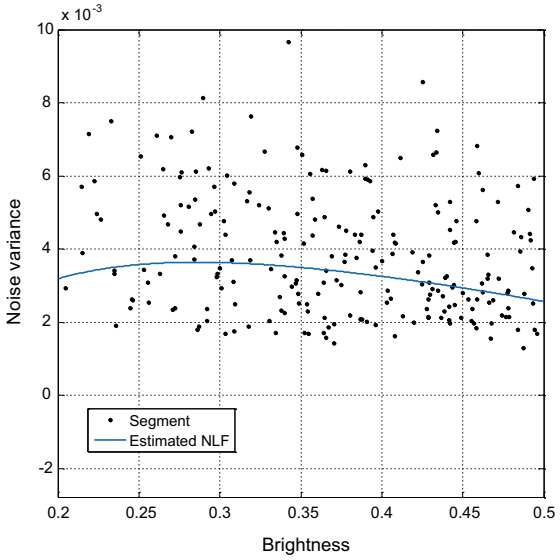

Fig. 4. A close look of a part of segmental noise distribution and approximated noise level function.

(13). Then, the distances $Dist_j^i$ in each subset are sorted in ascending order, as demonstrated in Fig. 5, where the rows, denoted with j, indicate the corresponding subset, and the total number of subsets is denoted with N_S; the columns, denoted with i, indicate the corresponding parameter combination, and the total number of combinations is denoted with N_Cmb. Afterwards, the elements in each sorted array are checked in dictionary order to obtain the optimal parameters with which each all rows will have a relative minimum distance. As indicated in Fig. 5, after sorting, the first column is firstly checked in the first iteration, as denoted in red; then the first two columns are checked in the second iteration, as denoted in yellow; and in this way, the first i columns will be checked in the ith iteration, as denoted in blue; and the iterations will continue until the optimal parameters are obtained.

$$Dist_j^i = \sqrt{\left(p_{idl} - p_j^i\right)^2 + \left(r_{idl} - r_j^i\right)^2} \tag{13}$$

where $Dist_j^i$ indicates the distance of the ith point in the jth subset; p_j^i and r_j^i indicates the *precision* and *recall* rates of the corresponding point, respectively; and (p_{idl}, r_{idl}) represents the ideal condition, $p_{idl} = 1$ and $r_{idl} = 1$.

Fig. 6 illustrates the detection results in different scales and the final detection results when outliers map becomes stable. In Fig. 6, (a) shows the forged image with four spliced objects; (b) shows the corresponding ground truth image; (c) and (d) show the detection results in the 9th and 11th scale, respectively; and (e)–(h) show the detection results in the 13th, 14th, 15th, and 16th scale, respectively. It can be easily seen that when in the 9th and 11th scale, there are some false positive values in the detection results, and the false positive area becomes smaller as outliers map in more scales are created; while in the 13th, 14th, 15th, and 16th scale, the outliers map changes little in successive scales, and the detection results are becoming stable, in this case, the algorithm stops and gives the final result.

3. Experimental results and comparisons

In this section, a lot of experiments are conducted to evaluate the proposed scheme. In following experiments, we first create a forgery dataset which is composed of 35 different splicing

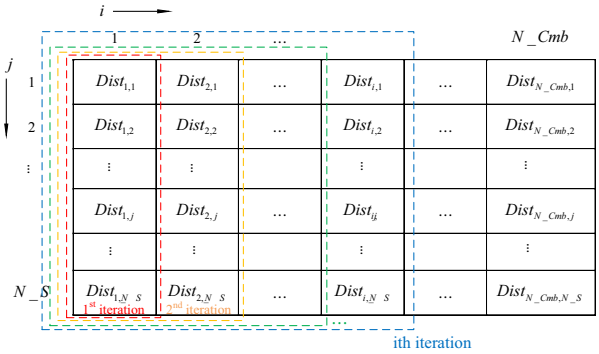

Fig. 5. Demonstration of OPCS algorithm.

situations and thousands of forged images, to test the proposed algorithm. The source images are from [36] and each subset is created according to different combinations of spliced region size and different variance of Gaussian noise. The Gaussian noise is added to spliced region in each tampered picture. Table 1 shows detailed specification of the created dataset, including the number of images in each subset, the number of spliced objects in images of each subset, the average proportion of spliced region size in image of each subset, and the variance of Gaussian noise added in the spliced region in each subset.

To approximate the real image forgery scenarios, in our dataset, the spliced regions which are pasted into the host images are mostly individual objects, for example, a piece of toy, other than visually meaningless area, like some public datasets. Besides, in our dataset, as introduced in Table 1, multiple number of objects are pasted into the host images to form the forgery, and the size of the pasted object is about 2–8% coverage of the host image. Specifically, in different subsets, the average proportion of spliced region varies from about 4% in subset #1–#5, to about 20% in subset #31–#35. In addition, the Gaussian noise of different variance is added into the dataset, and the last column in Table 1 indicates variance of the noise in different subset.

Table 1
Specification of forgery database.

Subset #	Number of images	Number of spliced objects	Average proportion of spliced region size (%)	Added Gaussian noise (variance)
#1	49	1 object	4.73	0.001
#2	40		4.43	0.003
#3	45		5.13	0.005
#4	45		4.31	0.007
#5	49		5.00	0.01
#6	47	2 objects	10.20	0.001
#7	45		7.95	0.003
#8	44		8.32	0.005
#9	45		9.51	0.007
#10	42		10.48	0.01
#11	44	3 objects	13.05	0.001
#12	44		12.97	0.003
#13	49		13.80	0.005
#14	49		12.84	0.007
#15	47		14.06	0.01
#16	46	4 objects	17.23	0.001
#17	45		16.08	0.003
#18	51		17.77	0.005
#19	42		18.30	0.007
#20	42		16.42	0.01
#21	49	5 objects	18.95	0.001
#22	43		21.33	0.003
#23	46		19.62	0.005
#24	45		21.01	0.007
#25	42		20.90	0.01
#26	53	6 objects	22.43	0.001
#27	43		23.39	0.003
#28	40		26.49	0.005
#29	42		23.72	0.007
#30	47		23.25	0.01
#31	52	7 objects	26.42	0.001
#32	30		21.82	0.003
#33	29		21.30	0.005
#34	31		22.47	0.007
#35	28		20.45	0.01

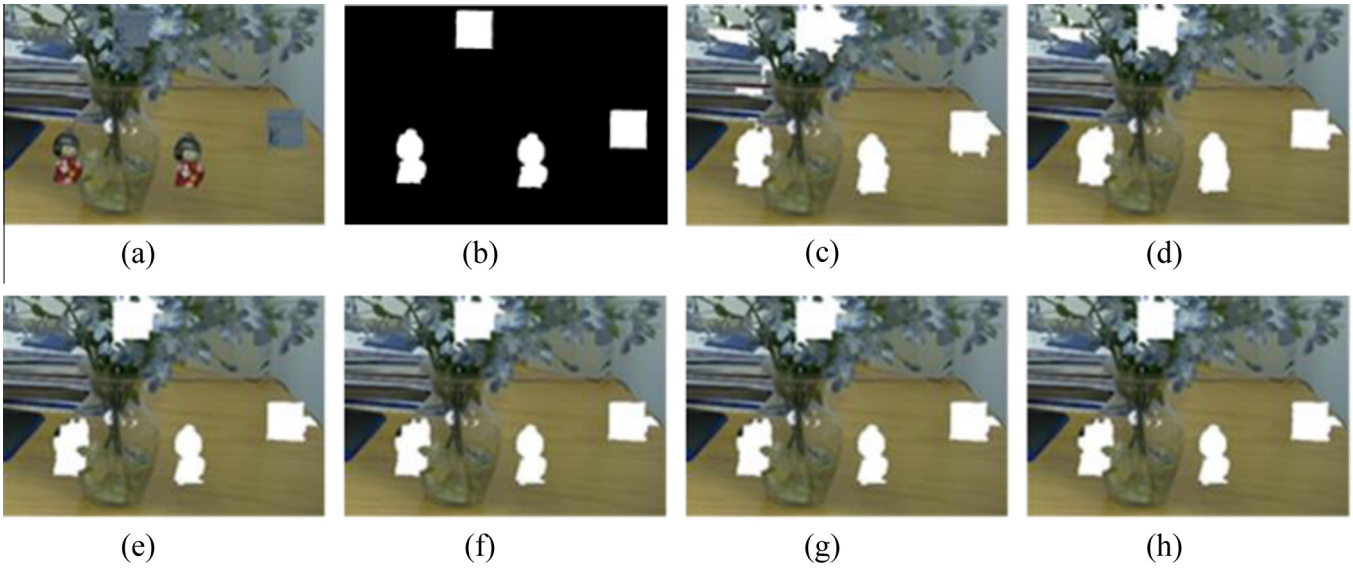

Fig. 6. Detection results in different scales and the final detection results when outliers map becomes stable. (a) A forgery image with four spliced objects. (b) Ground truth image. (c) Detection result in the 9th scale with some false positive results. (d) Detection result in the 11th scale with some false positive results. (e)–(h) Detection results become stable in the 13th, 14th, 15th, and 16th scale.

Table 2
Specifications of training dataset.

Subset #	Number of images	Number of spliced objects	Average proportion of spliced region size (%)	Added Gaussian noise (variance)
#1	157	1 small object	2.17	0.001
#2	138		2.08	0.003
#3	155		2.16	0.005
#4	153	1 medium object	7.97	0.001
#5	141		7.74	0.003
#6	148		7.79	0.005
#7	95	1 large object	25.14	0.001
#8	110		23.75	0.003
#9	105		22.66	0.005
#10	150	2 small objects	4.09	0.001
#11	150		4.07	0.003
#12	148		4.04	0.005
#13	145	3 small objects	6.25	0.001
#14	156		6.27	0.003
#15	164		6.39	0.005
#16	157	4 small objects	8.17	0.001
#17	145		8.39	0.003
#18	152		8.63	0.005
#19	151	5 small objects	10.23	0.001
#20	148		10.49	0.003
#21	151		10.34	0.005

3.1. Parameter settings by training

As introduced in Section 2.3, the algorithm terminates when outliers map becomes stable, which is described by three parameters K, C, and τ. The optimal combination of the parameters is obtained by training, and this section will explain the details of the training process. Similar to forgery dataset, the training dataset which contains subsets representing different scenarios of forgery is created, as shown in Table 2. To create the training dataset, multiple number of objects, which are of various sizes, are pasted into host images; besides, Gaussian noise of various variances are added. Table 2 shows the specifications and according to which, the training dataset is composed of 21 subset. In subset #1–#3 and #10–#21, small forged objects account for about 2% of the host image; in subset #4, #5 and #6, the spliced objects account for about 8%, and they are defined as medium size spliced objects; while large objects come out in subset #7, #8 and #9, accounting for more than 20% of the host image.

After creating the training dataset, the candidate values of parameters$K$, C, and τ are given by experiences as follows:

$$K = \{3, 5, 7, 10, 15, 20\};$$
$$C = \{2, 3, 4, 5, 7, 10\};$$
$$\tau = \{0.005, 0.01, 0.03, 0.05, 0.1\}.$$

With the given parameters candidates and the OPCS proposed in Section 2.3, Fig. 7 presents the corresponding ROC curves with optimal parameter combination $\{K = 20, C = 4, \tau = 0.05\}$ indicated. In Fig. 7, (a)–(g) display the ROC curves of subset #1–#21 of the training dataset, respectively, and different colors indicate results of different subset; while (h) displays the corresponding *precision* and *recall* of the training dataset when using optimal parameters combination, where the different colors indicate results of different subset, respectively, and different symbols, O, □, and Δ denote the results in conditions with different noise variances.

In other words, the optimal parameter combination $\{K = 20, C = 4, \tau = 0.05\}$ means that the algorithm stops when sum of inner distance of 20 clusters from outliers map changes smaller than 5% in 4 successive scales. Although the combination may not be the best parameters for a certain situation represented by a subset, it is the optimal choice for all subsets of our training dataset. For practical use, the observer does not know size of the objects, noise discrepancy and the number of spliced objects, therefore a combination of parameters should be effective for all possible situations. The *precision* and *recall* of chosen parameters in the figures are acceptable, albeit not best for a single subset.

3.2. Performances and comparisons

In this section, the performance of the proposed scheme is evaluated using the image forgery database which we have created and is composed of 35 subsets, specifications of which were displayed in Table 1. The *precision* and *recall* are defined and thus used to measure the effectiveness of the proposed method. In the following experiments, *precision* is defined as the ratio of correctly detected forged pixels over all detected pixels, while *recall* refers to the ratio of detected spliced pixels over all spliced pixels. Accordingly, the F score is defined based on *precision* and *recall*, as in (14), to evaluate the experimental results. In addition, the existing state-of-the-art method proposed by Lyu et al. [33] is implemented and conducted on the same image forgery database which we have created and the results are compared with ours to reveal the better performance of the proposed scheme. Table 3 shows the *precision, recall*, and F results of the 35 subsets, respectively, to show the performance of the proposed scheme; furthermore, the *precision, recall*, and F results of Lyu's [33] method are calculated on the same dataset and displayed for comparison, to reveal the effectiveness of the proposed scheme. The experimental results demonstrates our proposed method can successfully detect multiple spliced objects. The comparison results are given to demonstrate the superiority of the proposed scheme, in most of the cases, the proposed scheme outperforms the existing state-of-the-art method, while in some exceptional cases, although the *recall* of the proposed method is lower, the *precision* is high. It means that the observer can still identify the forged area despite they are not fully marked in the detection results.

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{14}$$

In Table 3, the results highlighted in italic indicate results of our proposed scheme, while the results highlighted in bold indicate the better results. The comparison results indicate that our *precision* is comparatively high and stable across all of the subsets in the dataset; in addition, the advantage of our method is outstanding especially when there are more spliced objects in the image. It's obvious that the variance of added noise affects the detection results which were based on noise estimation: in situation that the spliced area is similar, performance is better in higher level of added noise, as indicated in Table 3. Another special case is that when there are more spliced objects in the image, the *recall* rate of our method will be lower, as indicated in Table 3, however, we can still maintain the *precision* rate. Therefore the detection results are acceptable although each forged object is not fully marked.

Besides Table 3, we also demonstrate the results in Figs. 8 and 9. Fig. 8 shows the *precision* and *recall* results of the proposed scheme, changing with number of spliced regions; where the results indicated with solid lines show the *precision* rate, and the results indicated with dash lines show the *recall* rate. In Fig. 8, the x-axis indicates the number of spliced regions in the corresponding image, while y-axis indicates the *precision*/*recall* rates. Different colors[1] and symbols show the results in different situations, where

[1] Please note that 'Figs. 8 and 9' will appear in B/W in print and color in the web version. Based on this, please approve the footnote 1 which explains this.

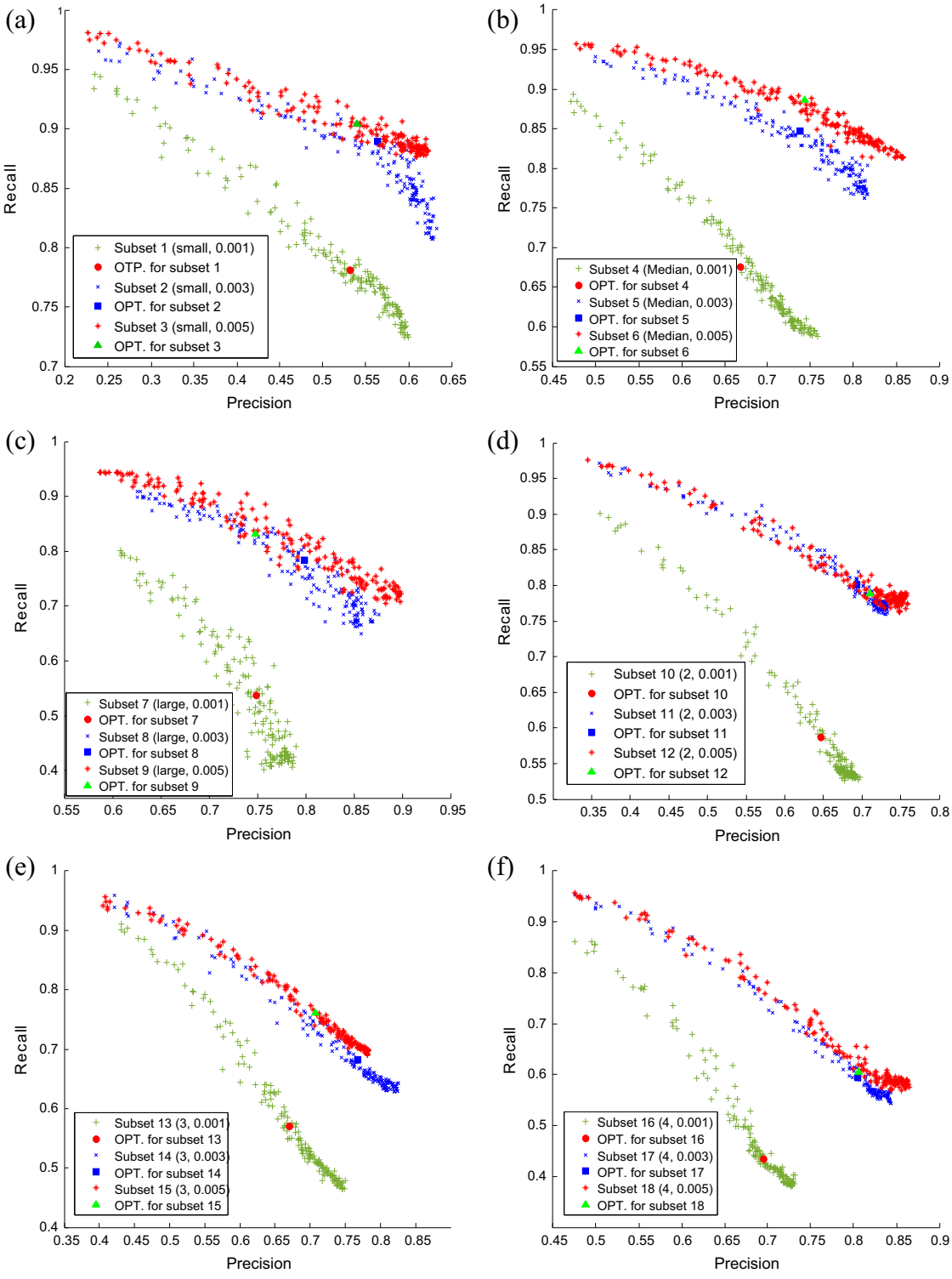

Fig. 7. ROC curves of all subsets, with the optimal parameter combination indicated. (a)–(g) ROC curves of subset #1–#21 of the training dataset; and (h) the corresponding *precision* and *recall* of the training dataset when using optimal parameters combination.

the variances of added noise are different: yellow lines with symbols 'O' indicate the results when variance $\sigma = 0.001$, blue lines with symbols '□' indicate the results when variance $\sigma = 0.003$, red lines with symbols '∇' indicate the results when variance $\sigma = 0.005$, gray lines with symbols 'Δ' indicate the results when variance $\sigma = 0.007$, and green lines with symbols '▷' indicate the results when variance $\sigma = 0.01$. It is obviously that when the number of spliced regions becomes larger, the *precision* rates of the proposed scheme remain stable, even when variance of added noise being different; on the other hand, the *recall* rates will become lower as the number of spliced regions increase.

Taken together, Fig. 9 shows the comparison of the proposed scheme and the Lyu's [33] method. In Fig. 9, the results indicated in blue line and symbols '∗' show the *F* score of the

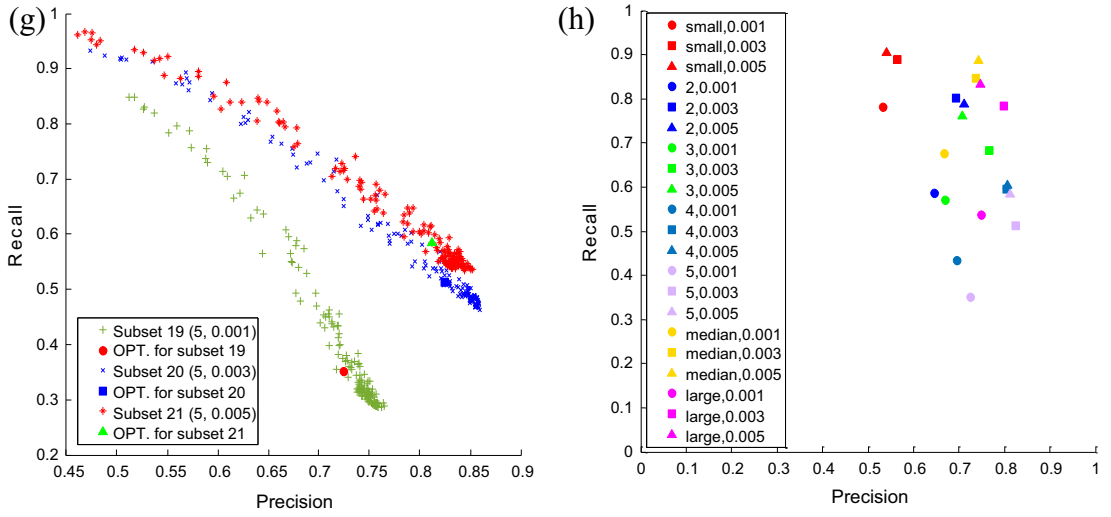

Fig. 7 (*continued*)

Table 3
Comparison of the proposed scheme with the state-of-the-art method.

Subset #	*Precision* Lyu's [33] (%)	*Precision* Our's (%)	*Recall* Lyu's [33] (%)	*Recall* Our's (%)	*F* Lyu's [33] (%)	*F* Our's (%)
#1	37.82	***68.12***	24.30	***62.10***	29.59	***64.97***
#2	60.19	***75.27***	26.57	***87.00***	36.87	***80.71***
#3	64.10	***78.50***	50.89	***88.12***	56.74	***83.04***
#4	53.33	***66.37***	71.20	***80.34***	60.98	***72.69***
#5	11.12	***75.44***	**99.89**	*93.16*	20.02	***83.37***
#6	54.23	***68.88***	10.45	***46.45***	17.52	***55.48***
#7	60.67	***87.71***	20.71	***68.83***	30.88	***77.13***
#8	68.94	***80.52***	41.11	***78.13***	51.51	***79.30***
#9	48.99	***77.39***	69.25	***77.80***	57.38	***77.59***
#10	11.66	***75.27***	**97.41**	*64.27*	20.83	***69.34***
#11	41.87	***68.95***	9.59	***38.42***	15.61	***49.35***
#12	58.63	***89.85***	20.39	***54.81***	30.26	***68.08***
#13	69.60	***88.24***	34.05	***51.67***	45.72	***65.18***
#14	56.90	***81.02***	63.89	***67.89***	60.20	***73.88***
#15	14.11	***77.37***	**82.50**	*59.01*	24.10	***66.95***
#16	54.10	***85.32***	4.75	***34.33***	8.73	***48.97***
#17	63.82	***82.65***	12.01	***45.31***	20.22	***58.53***
#18	72.00	***84.05***	32.41	***43.58***	44.70	***57.40***
#19	56.64	***83.18***	**65.14**	*54.78*	60.59	***66.06***
#20	16.46	***78.31***	**78.62**	*46.75*	27.22	***58.55***
#21	54.59	***79.13***	2.86	***22.56***	5.44	***35.11***
#22	70.12	***86.74***	5.78	***33.23***	10.68	***48.05***
#23	74.67	***87.57***	27.45	***42.75***	40.14	***57.45***
#24	43.63	***83.69***	**77.50**	*45.48*	55.83	***58.93***
#25	20.94	***66.64***	**82.50**	*45.86*	33.41	***54.33***
#26	48.09	***80.68***	1.31	***16.32***	2.54	***27.15***
#27	66.95	***89.14***	15.28	***27.78***	24.88	***42.36***
#28	78.62	***74.28***	23.75	***29.06***	36.48	***41.78***
#29	50.94	***76.97***	**71.42**	*29.30*	**59.46**	*42.45*
#30	23.29	***68.74***	**88.26**	*40.16*	36.86	***50.70***
#31	46.97	***76.45***	0.56	***20.12***	1.11	***31.91***
#32	59.03	***84.56***	5.77	***25.01***	10.51	***38.60***
#33	74.07	***77.39***	**36.99**	*26.74*	**49.34**	*39.75*
#34	49.75	***84.67***	**70.99**	*34.82*	**58.50**	*49.35*
#35	20.47	***78.22***	**97.42**	*38.37*	33.83	***51.48***

proposed scheme, and the results indicated in red line and symbols '•' show the *F* score of Lyu's method [33]. The comparison results show the better performance of the proposed scheme. In addition, a special thing to note is in (14), where during the calculation of *F*, the weight of *precision* and *recall* is equivalent; however, in the application of splicing forgery detection, the goal is to discriminate whether or not a picture has been spliced by locating the forged area. This fact demonstrates that locating the forged

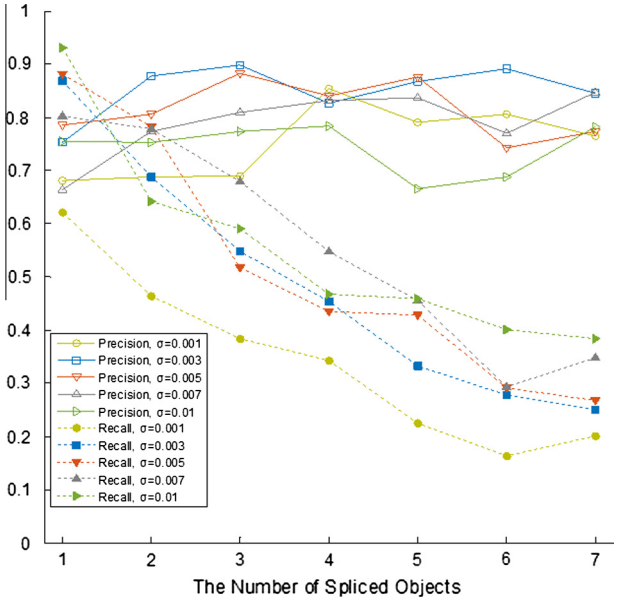

Fig. 8. Demonstration of *precision* and *recall* results changing with number of spliced regions.

regions accurately is more important, and thus, *precision* is crucial. Low *precision* leads to misjudge since many regions will be marked as possible spliced area so that the observer cannot judge if the test picture is fake and which part is fake. Compared to *precision*, *recall* is not as important as *precision* in this situation, because partial locating of spliced area will not influence the observer's judgement if the *precision* remains high. In terms of the value of *F*, the performance of our algorithm is better than Lyu's method [33], except for subsets #29, #33 and #34, still and all, our *precision* is better in these exceptional subsets.

Considering the spliced picture may be undergone some post-processing such as contrast enhancement and JPEG compression, the following experimental results demonstrate the performance of the proposed method when the images are contrast enhanced and JPEG compressed. The test database is derived from the former forgery database which includes 35 subsets. Each picture in the original forgery database is altered to ten versions: five copies with different effect of contrast enhancement and five with different

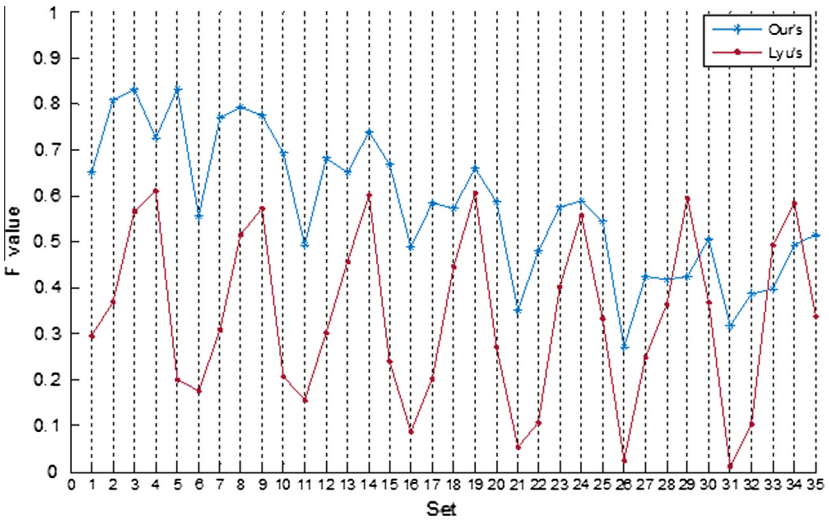

Fig. 9. Comparison of *F* value between our algorithm and Lyu's method.

Table 4
Performance of proposed method against contrast enhancement and JPEG compression.

Subset#	*Original* (%)		*Contrast Enhancement* (%)		*JPEG Compression* (%)	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
#1	68.12	62.10	53.63	81.28	54.25	69.82
#2	75.27	87.00	52.37	94.25	52.10	80.26
#3	78.50	88.12	62.61	90.26	64.29	76.18
#4	66.37	80.34	51.21	90.57	45.42	84.60
#5	75.44	93.16	54.95	92.57	57.59	88.17
#6	68.88	46.45	58.72	75.42	58.96	61.80
#7	87.71	68.83	67.84	88.98	68.44	76.09
#8	80.52	78.13	66.51	86.22	64.99	69.95
#9	77.39	77.80	64.68	87.61	67.99	73.16
#10	75.27	64.27	59.85	90.75	64.62	76.74
#11	68.95	38.42	57.68	76.22	55.21	63.84
#12	89.85	54.81	71.55	83.65	80.45	58.95
#13	88.24	51.67	71.60	77.83	69.13	68.64
#14	81.02	67.89	63.46	82.53	69.99	69.35
#15	77.37	59.01	61.25	86.23	66.05	79.75
#16	85.32	34.33	70.22	53.04	75.59	48.91
#17	82.65	45.31	61.72	82.03	65.97	62.56
#18	84.05	43.58	60.12	83.06	66.67	62.86
#19	83.18	54.78	61.61	81.16	75.73	63.67
#20	78.31	46.75	52.15	89.19	66.96	65.19
#21	79.13	22.56	55.37	62.27	64.52	43.88
#22	86.74	33.23	74.79	66.76	74.62	50.22
#23	87.57	42.75	60.70	76.85	70.73	56.28
#24	83.69	45.48	51.41	83.20	67.08	57.05
#25	66.64	45.86	49.16	82.18	62.84	67.61
#26	80.68	16.32	74.44	43.28	58.41	45.55
#27	89.14	27.78	62.46	66.33	67.15	47.34
#28	74.28	29.06	62.66	69.54	68.85	47.92
#29	76.97	29.30	54.30	78.44	59.65	63.04
#30	68.74	40.16	50.65	73.77	68.38	55.32
#31	76.45	20.12	45.56	37.25	42.65	29.50
#32	84.56	25.01	54.64	83.69	52.74	50.27
#33	77.39	26.74	57.12	86.18	66.65	60.06
#34	84.67	34.82	59.10	82.35	53.49	65.56
#35	78.22	38.37	55.82	79.17	63.15	47.47

JPEG quality of 10, 30, 50, 70 and 90 respectively. Accordingly the dataset used in this experiment is ten times as large as the former one. Table 4 shows the average *precision* and *recall* rates of five scenarios of contrast enhancement and five of compression quality in each 35 subset. Not much loss in *precision* shows our method is robust to common post-processing like contrast enhancement and JPEG compression.

4. Conclusions

This paper introduced a novel method to expose splicing forgery in digital images. The primary cue utilized is the noise discrepancy between the spliced image and host image. Noise level function is used to separate potential spliced segments whose noise level is inconsistent and multi-scale analysis finally presents visually distinguishable results to observers. The significant contribution is that forensic algorithm is expanded into multi-scale and our method achieves satisfactory results. Experimental results show our proposed method retains good detection accuracy in diverse situations: spliced area with different noise variance, spliced area of different size and different number of spliced objects. Comparing to the existing single scale noise discrepancy based algorithm, our method outperforms them in detection accuracy.

There is still room for improvement. Noise estimation process is acceptable but not good enough, accordingly the false positive rate is higher than we expected. A valid and robust noise estimation scheme will reduce the scale needed and thus improve efficiency. Apart from noise discrepancy, other forgery detection scheme may be extended into multi-scales for better performance.

Acknowledgment

This research was supported in part by the Research Committee of the University of Macau (MYRG2015-00011-FST, MYRG2015-00012-FST) and the Science and Technology Development Fund of Macau SAR (008/2013/A1, 093-2014-A2).

References

[1] G. Cao, Y. Zhao, R. Ni, X. Li, Contrast enhancement-based forensics in digital images, IEEE Trans. Inform. Forensics Secur. 9 (2014) 515–525.
[2] S. Bravo-Solorio, A.K. Nandi, Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics, Signal Process. 91 (8) (2011) 1759–1770.
[3] H. Farid, Image forgery detection, IEEE Signal Process. Mag. 26 (2009) 16–25.
[4] S.D. Lin, T. Wu, An integrated technique for splicing and copy-move forgery image detection, in: 2011 4th International Congress on Image and Signal Processing (CISP), 2011, pp. 1086–1090.
[5] B. Mahdian, S. Saic, Blind authentication using periodic properties of interpolation, IEEE Trans. Inform. Forensics Secur. 3 (2008) 529–538.
[6] W. Wei, S. Wang, X. Zhang, Z. Tang, Estimation of image rotation angle using interpolation-related spectral signatures with application to blind detection of image forgery, IEEE Trans. Inform. Forensics Secur. 5 (2010) 507–517.

[7] H. Farid, Exposing digital forgeries from JPEG ghosts, IEEE Trans. Inform. Forensics Secur. 4 (2009) 154–160.
[8] T. Bianchi, A. Piva, Image forgery localization via block-grained analysis of JPEG artifacts, IEEE Trans. Inform. Forensics Secur. 7 (2012) 1003–1017.
[9] G. Valenzise, M. Tagliasacchi, S. Tubaro, Revealing the traces of JPEG compression anti-forensics, IEEE Trans. Inform. Forensics Secur. 8 (2013) 335–349.
[10] W. Li, Y. Yuan, N. Yu, Passive detection of doctored JPEG image via block artifact grid extraction, Signal Process. 89 (9) (2009) 1821–1829.
[11] M.P. Rao, A.N. Rajagopalan, G. Seetharaman, Harnessing motion blur to unveil splicing, IEEE Trans. Inform. Forensics Secur. 9 (2014) 583–595.
[12] K. Bahrami, A.C. Kot, L. Li, H. Li, Blurred image splicing localization by exposing blur type inconsistency, IEEE Trans. Inform. Forensics Secur. 10 (2015) 999–1009.
[13] M.K. Johnson, H. Farid, Exposing digital forgeries in complex lighting environments, IEEE Trans. Inform. Forensics Secur. 2 (2007) 450–461.
[14] Q. Liu, X. Cao, C. Deng, X. Guo, Identifying image composites through shadow matte consistency, IEEE Trans. Inform. Forensics Secur. 6 (2011) 1111–1122.
[15] T.J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, A. de Rezende Rocha, Exposing digital image forgeries by illumination color classification, IEEE Trans. Inform. Forensics Secur. 8 (2013) 1182–1194.
[16] P. Ferrara, T. Bianchi, A. De Rosa, A. Piva, Image forgery localization via fine-grained analysis of CFA artifacts, IEEE Trans. Inform. Forensics Secur. 7 (2012) 1566–1577.
[17] M.K. Johnson, H. Farid, Exposing digital forgeries through chromatic aberration, in: Proceedings of the 8th Workshop on Multimedia and Security, 2006, pp. 48–55.
[18] Y.F. Hsu, S.F. Chang, Camera response functions for image forensics: an automatic algorithm for splicing detection, IEEE Trans. Inform. Forensics Secur. 5 (2010) 816–825.
[19] H. Fu, X. Cao, Forgery authentication in extreme wide-angle lens using distortion cue and fake saliency map, IEEE Trans. Inform. Forensics Secur. 7 (2012) 1301–1314.
[20] M. Chen, J. Fridrich, M. Goljan, J. Lukáš, Source digital camcorder identification using sensor photo response non-uniformity, in: Electronic Imaging 2007, 2007, pp. 1–12.
[21] C.-T. Li, Y. Li, Digital camera identification using colour-decoupled photo response non-uniformity noise pattern, in: Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), 2010, pp. 3052–3055.
[22] X. Pan, X. Zhang, S. Lyu, Exposing image splicing with inconsistent local noise variances, IEEE International Conference on Computational Photography (ICCP) 2012 (2012) 1–10.
[23] X. Pan, X. Zhang, S. Lyu, Exposing image forgery with blind noise estimation, in: Proceedings of the Thirteenth ACM Multimedia Workshop on Multimedia and Security, 2011, pp. 15–20.
[24] G.E. Healey, R. Kondepudy, Radiometric CCD camera calibration and noise estimation, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 267–276.
[25] Y. Tsin, V. Ramesh, T. Kanade, Statistical calibration of CCD imaging process, in: Proceedings, Eighth IEEE International Conference on Computer Vision, 2001, ICCV 2001, 2001, pp. 480–487.
[26] S. Lyu, E.P. Simoncelli, Nonlinear extraction of independent components of natural images using radial gaussianization, Neural Comput. 21 (2009) 1485–1519.
[27] M. Bethge, Factorial coding of natural images: how effective are linear models in removing higher-order dependencies?, JOSA A 23 (2006) 1253–1268
[28] D. Zoran, Y. Weiss, Scale invariance and noise in natural images, in: 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 2209–2216.
[29] G. Cao, Y. Zhao, R. Ni, B. Ou, Y. Wang, Forensic detection of noise addition in digital images, J. Electron. Imag. 23 (2014) 1–9.
[30] C. Liu, R. Szeliski, S.B. Kang, C.L. Zitnick, W.T. Freeman, Automatic estimation and removal of noise from a single image, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 299–314.
[31] S. Pyatykh, J. Hesser, Z. Lei, Image noise level estimation by principal component analysis, IEEE Trans. Image Process. 22 (2013) 687–699.
[32] W. Liu, W. Lin, Additive white Gaussian noise level estimation in SVD domain for images, IEEE Trans. Image Process. 22 (2013) 872–883.
[33] S. Lyu, X. Pan, X. Zhang, Exposing region splicing forgeries with blind local noise estimation, Int. J. Comput. Vis. 110 (2014) 202–221.
[34] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 2274–2282.
[35] P. Lancaster, K. Salkauskas, Curve and Surface Fitting, Academic Press, 1986.
[36] Y.-F. Hsu, S.-F. Chang, Detecting image splicing using geometry invariants and camera characteristics consistency, in: 2006 IEEE International Conference on Multimedia and Expo, 2006, pp. 549–552.