CFA Artifacts Analysis for Image Splicing Detection

A A Varlamova[1] and A V Kuznetsov[1, 2]

[1]Samara National Research University, Moskovskoe Shosse 34, Samara, Russia, 443086
[2]Image Processing Systems Institute - Branch of the Federal Scientific Research Centre "Crystallography and Photonics" of Russian Academy of Sciences, Molodogvardeyskaya str. 151, Samara, Russia, 443001

Abstract. Image splicing is one of the widespread image forgery techniques. It represents pasting in image parts of other images. The paper is devoted to one of the methods of image splicing localization based on analysis of CFA artifacts that appear in an image during the capturing process. The proposed method is based on measuring a feature that characterizes the presence/absence of CFA artifacts for each image block. Obtained values of the feature define the probability of each block to be pasted. In the experimental part of the paper, we analyse the accuracy of the splicing detection method and its robustness against different types of distortions such as additive Gaussian noise, JPEG compression, and linear enhancement. The results showed that the suggested method reveals pasted regions of different shape, size, and nature in images. The method possesses stability against additive Gaussian noise and linear enhancement, but it is not steady against JPEG compression. The advantage of the method is the ability to localize splicing regions even at the smallest 2×2 block level.

1. Introduction

Today the appearance of a large number of digital devices, which allow capturing images has led to a decrease in their cost and, as a consequence, to their wide availability for each person. At the same time, a large number of software tools for image editing has also increased significantly. These tendencies have resulted in widespread image forgeries.

Nowadays, any user can make changes to the image which can be visually imperceptible. Besides, if it comes to professional forgery, the majority of existing services for verifying the authenticity of images cannot reveal it too.

There are many examples from the military and political spheres, the media, litigation, the activities of insurance companies and many other areas when forged images were used for the purpose of committing crimes or concealing any facts, or for copyright infringement, or to cause public outcry [1]. In this regard, image protection and verification of image authenticity have become the urgent problems.

Depending on the purpose of forgery, images can be subjected to such modifications as retouching and embedding of duplicates (copying areas of the image and pasting them into other areas of the same image). Methods for retouching and duplicates detection in images are considered in [2, 3] and [4, 5], respectively.

This work is devoted to detection of another commonly applied type of forgeries – photomontage. Photomontage represents pasting areas into an image that were taken from another image [6].

In some cases to protect images from this type of forgery embedding a digital watermark into an image can be done. [7]. However, this approach has several substantial disadvantages. Its application is limited: authentication can only be done by an owner of the data.

There are other solutions which do not require embedding additional information into an image. They include the methods that use the characteristics of device's sensor by which the image was obtained in order to detect the forgeries. One of such characteristics is color filter array (*CFA*) artifacts. These are local artifacts in an image caused by the presence of a CFA filter in the camera that captured the image. [8]. CFA is a mosaic of tiny color filters placed over the pixel sensors of the image sensor to capture color information. It is presented in most modern cameras. Note that CFA artifacts are unique for each camera model.

In [9] the authors describe the method of CFA artifacts detection. Under paradigm, they calculate the probability map of presence/absence CFA artifacts in the image and calculate its Fourier transform (FT). Spikes in the Fourier domain are the evidences of the map periodicity which means that the image contains CFA artifacts. With a small modification, the method can be used for detection of 256×256 forged image blocks.

Similarly, based on the fact that CFA artifacts have a periodic structure, the authors in [10] proposed an algorithm for determining the nature of images (whether they were obtained with a digital device or artificially generated). This method is also based on the analysis of the FT. The absence of CFA artifacts in an image area indicates that area has been altered or artificially generated, or a CFA filter was not used in the registering device. Due to the use of the FT, this method is applicable for detection of tampering regions on images of size 64 × 64 or greater.

Methods of this group are not robust against the forgeries when the image was reinterpolated after splicing was made. This problem will be solved in the course of further research. In this paper, this case is not considered, since it is another kind of forgeries and goes beyond the scope of the task. Also, we do not consider the case when the original image and the pasted region were obtained using the same recording device – in this case, CFA artifacts are the same.

This paper is devoted to the investigation of one of the methods for detecting pasted regions in images based on the analysis of CFA artifacts [8]. It allows detecting forgeries on areas with a minimum size of 2 × 2. The result of applying the method is a tampering probability map – a two-dimensional array, each element of which contains a probability of tampering of a corresponding local area in an image.

2. Image splicing detection based on CFA artifacts analysis

Despite the fact that image forgeries can be visually imperceptible they alter its statistical characteristics. In particular, they destroy the inter-pixel connections that arise during the process of obtaining an RGB image [11].

In most modern cameras CFA is used to produce an RGB image. There are a lot of types of CFA filters, but the most commonly used is the Bayer filter, which is shown in Figure 1.

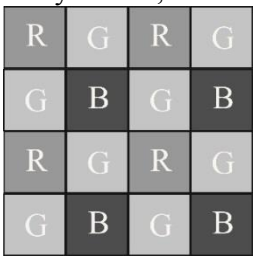


Figure 1. The Bayer filter.

Once the light passes through a CFA and a camera's sensor a RAW image is generated. A value of each pixel of a RAW image is defined for only one channel, whereas the other two values are not known. Hence only the third part of the color information is presented in a RAW image. A RAW file also contains the EXIF data – information on the date and time of photo capturing, a model of the capture device, and other parameters of recording the photo, etc.

Since for each sample of the RAW image the value of only one channel of three is determined, a demosaicing algorithm (an interpolation) is used to obtain a three-channel image. This leads to a correlation between the samples within each channel, and, as a consequence, CFA artifacts appear in the image due to the characteristics of a used camera [12].

The task of the demosaicing algorithm (demosaicing is interpolation of Bayer's templates) consists in obtaining an RGB image from the Bayer pattern. In other words, demosaicing algorithm is an interpolation of each of the three color channels in those samples where the value of the corresponding color component is unknown.

When obtaining a three-channel image by interpolation, in each channel the missing pixels values are calculated from the values of known, neighboring pixels. This process can be interpreted as a filtering process, in which the interpolation kernel (mask) is periodically applied to the original RAW image to obtain a resulting three-channel image.

There are many interpolation algorithms. The most detailed review of them is given in [9]. When calculating the missing values of pixels, the values from all channels can also be used for calculations, which lead to the appearance of interchannel connections. Further, for simplicity, we will consider the algorithm without interchannel connections. It means that missing channel values will be calculated based on known samples from the same channel only. All the calculations given in the paper are performed for the green channel, for the other two they can be produced in a similar way.

The simplest interpolation algorithm is the bilinear interpolation algorithm. Propose the bilinear interpolation algorithm was applied by the camera during the capturing process. Figure 2 shows a schematic view of the green image channel after interpolation. The pixels with known values (acquired with camera) are located in the positions A, whereas the pixels with interpolated values are located in the positions I.

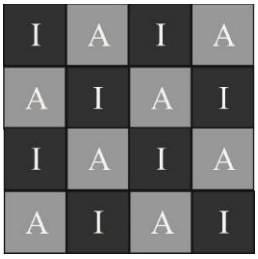


Figure 2. The green channel of the image (A– acquired green samples, I – interpolated green samples).

When using bilinear interpolation, the values of the green channel $s(x,y)$ are determined by (1):

$$s(x,y)=\begin{cases} G_A(x,y), & (x,y)\in A \\ G_I(x,y)=\sum_u\sum_v h(u,v)\times G_A(x-u)(y-v), & (x,y)\in I \end{cases}. \quad (1)$$

where $G_A(x,y)$ – values of the acquired samples in the green channel; $G_I(x,y)$ – values of the interpolated samples in the green channel; $h(u,v)$ – the interpolation kernel.

In this paper, the interpolation kernel is defined as:

$$h(u,v)=\frac{1}{4}\times\begin{bmatrix}0 & 1 & 0\\ 1 & 4 & 1\\ 0 & 1 & 0\end{bmatrix}. \quad (2)$$

Usually, color filter arrays (including the Bayer filter) have a periodical structure. Hence the correlation between samples has a periodical structure too. Forgeries destroy or alter the correlation between image samples. Thus, by analyzing the correlation of the pixels in local areas, whether the image was tampered or not can be determined.

2.1. CFA Modeling

For simplicity, the one-dimensional case will be considered, since the conclusions drawn from the calculations are also valid for the two-dimensional case.

Let $s(x)$ be a one-dimensional green channel of image that was obtained by interpolation using the Bayer filter. Then its values are determined by the equation (3):

$$s(x)=\begin{cases} G_A(x), & x(\mathrm{mod}2)=0 \\ G_I(x)=\sum_u h(u)G_A(x+u), & x(\mathrm{mod}2)\neq 0 \end{cases}, \quad (3)$$

where $G_A(x)$ – values of the acquired samples in the green channel; $G_I(x)$ – values of the interpolated samples in the green channel; $h(u)$ – the interpolation kernel.

In practice, only odd values of u (values of the acquired samples) contribute to the above summation, hence, $h(u)=0$ for odd values of u. Otherwise, $G_A(x+u)=0$ and the prediction error for the green channel can be determined by the equation (4):

$$e(x)=s(x)-\sum_u k(u)s(x+u), \quad (4)$$

where $k(u)$ – the prediction kernel.

Note that, in case the interpolation kernel $h(u)$ used by the camera is known, the prediction kernel coincides with the interpolation kernel, i.e. $k(u)=h(u)$ and there is no prediction error. In case the type of the used filter is not known, the prediction error occurs.

After the substitution of (3) in (4), the prediction error can be written as:

$$e(x)=\begin{cases} G_A(x)-\sum_u k(u)s(x+u), & x(\mathrm{mod}2)=0 \\ \sum_u h(u)G_A(x+u)-\sum_u k(u)s(x+u), & x(\mathrm{mod}2)\neq 0 \end{cases}.$$

Assume $k(u)=h(u)$, then the prediction error is equal to zero in the odd positions of x (interpolated) and differs from zero in the even positions of x (acquired with the camera). Hence, the variance of the prediction error is identically zero in the interpolated samples, whereas it differs from zero in the acquired samples.

In general, the exact interpolation coefficients may not be known, however, we can assume that $k(u)=0$ for odd u. Moreover, the equality $\sum_u k(u)=\sum_u h(u)=1$ usually holds for any used interpolation kernels.

Since only the values corresponding to odd values are meaningful, we consider only them to estimate the prediction error. Therefore, the prediction error can be expressed as follows:

$$e(x)=\begin{cases} G_A(x)-\sum_u k(u)\sum_v h(v)G_A(x+u+v), & x(\mathrm{mod}2)=0 \\ \sum_u (h(u)-k(u))G_A(x+u), & x(\mathrm{mod}2)\neq 0 \end{cases}. \quad (5)$$

By assuming that the values of the acquired samples are independent and identically distributed (i.i.d.) with the mean μ_G and the variance σ_G^2, the prediction error of the mean can be evaluated as (6):

$$E[e(x)]=\begin{cases} \mu_G-\mu_G\sum_u k(u)\sum_v h(v), & x(\mathrm{mod}2)=0 \\ \mu_G\left(\sum_u h(u)-\sum_u k(u)\right)=0, & x(\mathrm{mod}2)\neq 0 \end{cases}. \quad (6)$$

The variance of the prediction error in even samples x is calculated as (7):

$$Var[e(x)]=\sigma_G^2\left[\left(1-\sum_u k(u)h(-u)\right)^2+\sum_{t\neq 0}\left(\sum_u k(u)h(t-u)\right)^2\right]. \quad (7)$$

The variance of the prediction error in odd samples x is calculated as (8):

$$Var[e(x)]=\sigma_G^2\sum_u (h(u)-k(u))^2. \quad (8)$$

According to the above calculations, the variance of the prediction error is proportional to the variance of the acquired signal. If the prediction kernel is close to the interpolation kernel, the variance of the prediction error will be much higher at the positions of the acquired pixels than at the positions of the interpolated pixels.

2.2. Modeling of the method for image splicing detection

Thus the variance of the prediction error is higher in the acquired samples (samples A) than in the interpolated samples (samples I). This statement is also true for two-dimensional case. If the image was not obtained by applying the demosaicing algorithm or was forged, the variance of the prediction error for both types of samples will have close values within a range ε. Therefore, in order to identify the presence/absence of artifacts that arise after the application of the interpolation, it is necessary to calculate the variance of the prediction error for the samples A and I.

Let *s(x, y)* – the green channel of image, then the prediction error can be calculated by (9):

$$e(x,y) = s(x,y) - \sum_{u\neq 0}\sum_{v\neq 0} k(u,v)s(x+u, y+v). \quad (9)$$

where $k(u,v)$ – the two-dimensional prediction kernel.

Assume the used demosaicing algorithm is unknown, thus $k(u,v) \neq h(u,v)$, where $h(u,v)$ – the two-dimensional prediction kernel that was used by the camera to obtain the image.

Note that the values of the acquired samples usually are independent and identically distributed only locally, so the estimation of the local variance of the prediction error for both I and A samples is locally calculated.

Let the prediction error be local stationary within a range $(2K+1)\times(2K+1)$, $c = 1 - \sum_{i=-K}^{K}\sum_{j=-K}^{K}\alpha^2(i,j)$ – a scale factor that makes the estimator unbiased, $\mu_e = \sum_{i=-K}^{K}\sum_{j=-K}^{K}\alpha(i,j)e(x+i, y+j)$ – a local weighted mean of the prediction error, $\alpha'(i,j) = W(i,j)$ if $e(x+i, y+j)$ and $e(x,y)$ belong to the same class of samples, else $\alpha'(i,j) = 0$, W – a $(2K+1)\times(2K+1)$ Gaussian window with standard deviation $\sigma_W^2 = \frac{K}{2}$. A Gaussian window is a two-dimensional smoothing filter whose elements are distributed in accordance with the normal law of distribution. The value of the variance of W was chosen experimentally by comparison with other values from the following set: $\sigma_W^2 = \left\{K, \frac{K}{2}, \frac{K}{4}, \frac{K}{8}\right\}$.

Hence, the local weighted variance $\sigma_e^2(x,y)$ is defined by the equation (10):

$$\sigma_e^2(x,y) = \frac{1}{c}\left(\sum_{i=-K}^{K}\sum_{j=-K}^{K}\alpha(i,j)e^2(x+i, y+j) - \mu_e^2\right), \quad (10)$$

where $\alpha(i,j) = \frac{\alpha'(i,j)}{\sum_{i=-K}^{K}\sum_{j=-K}^{K}\alpha'(i,j)}$ – weights.

2.3. Feature modelling

After finding the locally-weighted variance of the prediction error, a feature characterizing the ratio between of variances of the prediction error in the acquired and interpolated samples is calculated. From the obtained values of the measure, it is possible to determine the presence/absence of CFA artifacts in the image.

Let the size of the analyzed image be $N\times N$, then we can calculate the feature for each of the disjoint image blocks of the $B\times B$ size. The size of block value must be related to the period of the Bayer filter, the smallest period and block size is 2×2. The matrix of the obtained values of the

variance of the predictor error is divided into blocks of size $B \times B$. Each block $B_{k,l}$ contains the values of variance of the acquired and interpolated samples, which we denote as $B_{Ak,l}$ and $B_{I\,k,l}$, respectively, where $k,l = \overline{0,\left(\frac{N}{B}\right)-1}$.

To calculate the feature for each image block, we use the geometric mean of the locally weighted variances of the prediction errors within the selected image fragment. It is worth noting that any other averaging measure can be used to get some characterization of the "tampering" of the fragment, for example, the arithmetic mean.

Let $GM_A(k,l)$ be the geometric mean of the prediction error for A samples within the block $B_{k,l}$ and defines by (11):

$$GM_A(k,l) = \left[\prod_{i,j \in B_A(k,l)} \sigma_e^2(i,j)\right]^{\frac{1}{|B_{Ak,l}|}}, \tag{11}$$

$GM_I(k,l)$ – the geometric mean of the prediction error for I samples within the block $B_{k,l}$ and defines by (12):

$$GM_I(k,l) = \left[\prod_{i,j \in B_I(k,l)} \sigma_e^2(i,j)\right]^{\frac{1}{|B_{I_{k,l}}|}}, \tag{12}$$

then the measure characterizing the ratio between prediction errors in the acquired and interpolated samples can be calculated by the (13):

$$L(k,l) = \ln\left[\frac{GM_A(k,l)}{GM_I(k,l)}\right]. \tag{13}$$

If CFA artifacts are present in the image block $B_{k,l}$, which means that this block was obtained using the demosaicing algorithm, the variance will be higher in A samples. Thus, the value of measure $L(k,l)$ will be positive. However, if the image was obtained in a different way, the prediction errors of variances for the two types of samples will have close values within a range ε, since the sample values will be equally distributed and will have the same statistical characteristics. Hence, the value of $L(k,l)$ will be close to zero within the range ε.

2.4. The tampering probability map estimation. Expectation-maximization algorithm (EM algorithm)

If in the image were pasted frames from other images, in order to make the insertion more realistic, it is usually accompanied by other processes: smoothing, compression, etc. These processes destroy the traces caused by the interpolation process, that is, leads to the destruction of CFA artifacts. Therefore, the values of the feature L in the image are be non-uniform: in some areas its values are much higher than zero, which is a consequence of the presence of CFA artifacts, and in other areas where CFA artifacts are absent, the feature values are close to zero within the range ε. This fact can be used to detect forgeries in images by using values of L to find the probability of the presence of CFA artifacts in each image block $B_{k,l}$. Thus, using the obtained measure values, it is possible to determine the probability map of the presence of CFA artifacts. For this aim, the EM algorithm is used [13].

Let there are two hypotheses: M_1 – CFA artifacts are present in the image; M_2 – CFA artifacts are absent in the image. Assume the $L(k,l)$ values is Gaussian distributed under both hypotheses and for any possible size of the blocks $B_{k,l}$. For a fixed $B \times B$, we can characterize the feature using the following conditional probability density functions:

$$P\{L(k,l)|M_1\} \sim N(\mu_1, \sigma_1^2),$$

where μ_1 – mean under the truth of the hypothesis M_1, $\mu_1 > 0$;

σ_1^2 – the variance under the truth of the hypothesis M_1;

$$P\{L(k,l)|M_2\} \sim N(\mu_2, \sigma_2^2),$$

where $\mu_2 = 0$ – mean under the truth of the hypothesis M_2; σ_2^2 – the variance under the truth of the hypothesis M_2.

Assume the distribution parameters in both cases are constant. If the image obtained with the demosaicing algorithm was modified, both hypotheses will be truth for each sample, but with different probabilities. This allows to represent the feature $L(k,l)$ as a mixture of two Gaussian distributions with mean $\mu_1 > 0$ in the regions where the artifacts are present, – in the intrinsic regions, and with mean $\mu_2 = 0$ in the regions where CFA artifacts are absent, – in the forged ones.

To estimate the distribution parameters of the feature $L(k,l)$: μ_1, σ_1^2, σ_2^2, we use the EM algorithm. It is an iterative algorithm consisting of two steps at each iteration. It allows dividing the mixture of several distributions and determining their latent variables by maximizing the likelihood ratio. Knowing the parameters for each sample, the posterior probabilities of each of the hypotheses $P\{M_1|L(k,l)\}$ and $P\{M_2|L(k,l)\}$ can be determined.

At the E-step of the algorithm, the probabilities of belonging samples to each of the models are calculated. In this case, we will consider the a priori probabilities of each of the hypotheses as equals: $P\{M_1\} = P\{M_2\} = \frac{1}{2}$. Then, the probability that the block was not changed and CFA artifacts are present in it, i.e. the probability of the hypothesis M_1 is determined by the Bayes rule (14):

$$P\{M_1|L(k,l)\} = \frac{P\{L(k,l)|M_1\}}{P\{L(k,l)|M_1\} + P\{L(k,l)|M_2\}}, \quad (14)$$

where $P\{L(k,l)|M_1\}$ – the probability of the $L(k,l)$ with the truth of the hypothesis M_1; $P\{L(k,l)|M_2\}$ – the probability of the $L(k,l)$ with the truth of the hypothesis M_2.

Similarly, according to the Bayes rule, the probability of the hypothesis M_2 – $P\{M_2|L(k,l)\}$ can be calculated. It is the probability that the block was tampered and CFA artifacts are absent.

Using the calculated probabilities, the distribution parameters: μ_1, σ_1^2, σ_2^2 can be estimated. These variables are fixed at the M-step, which makes it possible to calculate the likelihood ratio by the (15):

$$\Lambda(L(k,l)) = \frac{P\{L(k,l)|M_2\}}{P\{L(k,l)|M_1\}}. \quad (15)$$

The parameters providing the maximum value of the likelihood ratio are the required parameters, and the calculated likelihood ratio values represent a tampering probability map in which each sample $\Lambda(L(k,l))$ defines a probability of presence CFA artifacts in block $B_{k,l}$, so a small value is a sign that the block was tampered.

2.5. Model validation

The performance of the method can be measured by the true positive rate R_{TP}, characterizing the rate of correctly detected tampered blocks according to the formula (16), and false positive rate R_{FP}, characterizing the rate of falsely detected blocks according to the formula (17) [14].

$$R_{TP} = \frac{N_{m_{R2}}}{N_{R_2}}, \quad (16)$$

where R_2 – the forgered region of the image; N_{R_2} – the total amount of blocks in the forgered region R_2; $N_{m_{R2}}$ – the amount of blocks detected as tampered in the region R_2.

$$R_{FP} = \frac{N_{m_{R1}}}{N_{R_1}}, \tag{17}$$

where R_1 – the untampered region of the image; N_{R_1} – the total amount of blocks in the untampered region R_1; $N_{m_{R1}}$ – the amount of blocks detected as tampered in region R_1.

Further R_{TP} and R_{FP} are used to estimate the quality of detection of tampered areas.

3. Experimental research

For the experiments, four RAW files were taken from the database [15]. The selected images were obtained with four different cameras using the Bayer filter, namely: Canon EOS 450D, Nikon D50, Nikon D90, Nikon D7000. Type of a filter used in a camera can be learned from its technical characteristics. To obtain a three-channel TIFF image from a RAW file, we used the dcraw application [16]. Two types of images were pasted into the images to form forgeries: artificially created images, whose samples are not correlated with each other and images taken from sources [17, 18], which differ in interpolation properties from the source images.

First of all, in the course of experiments, the ability of algorithm to detect artificially generated built-in regions of various nature and shape was verified. Figure 3 shows an example of an image with a forged region of an arbitrary shape and the corresponding tampering probability map computed by 8 × 8 blocks. The pasted region was obtained by the camera. It should be noted that the algorithm makes it possible to detect tampering of very small sizes, so the tampering map can be calculated by blocks with a minimum size of 2 × 2.

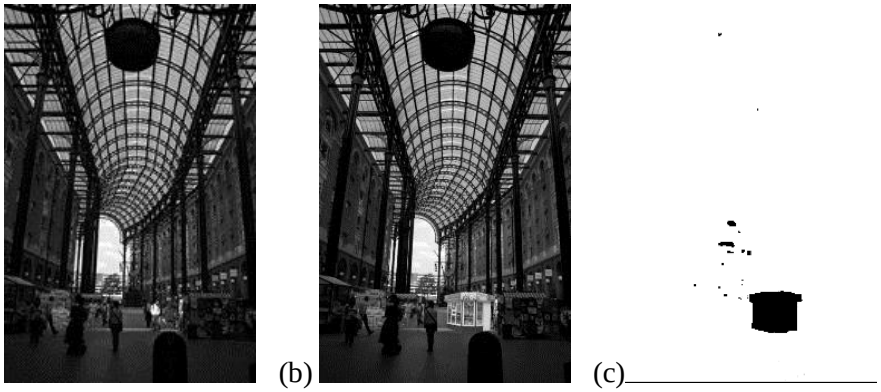


Figure 3. Examples of the algorithm performance with the size of the processed block 8 × 8: a) the source image, b) the image with a forged region of an arbitrary shape, c) tampering probability map.

Let the size of the embedded region – $M \times M$. The graphs of the dependency of rates $R_{TP}(M)$ and $R_{FP}(M)$ on the size of the built-in area are shown in Figure 4.

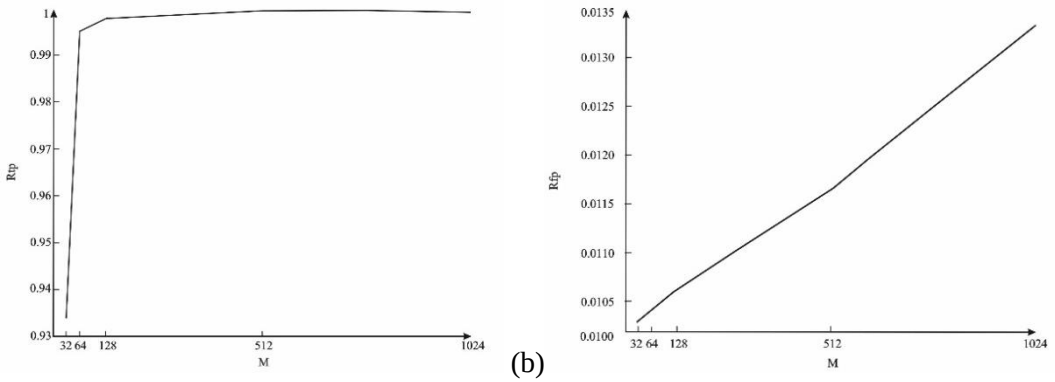


Figure 4. The dependency of quality rates on the size of the embedded region a) $R_{TP}(M)$, b) $R_{FP}(M)$.

The results of the experiment showed that with the increase in the size of the pasted area, the detection quality improves and reaches a maximum value – 1 at a size of 512 × 512. Note that with a minimum explored size of pasted region – 32 × 32 $R_{TP} = 0{,}93$, that characterizes the high quality of detection ability. In this case, the number of falsely detected unforged image blocks grows insignificantly and at a size of pasted area of 1024 × 1024 $R_{FP} = 0{,}0133$.

3.1. Investigation of method robustness against different types of distortions

To study the stability of the algorithm to various types of distortions, we used previously obtained 40 test images with a fixed size of the built-in area of 128 × 128.

As a part of the research, additive Gaussian noise with the signal-to-noise ratio (SNR (dB)): 30, 35, 40, 45, 50 was added to each of them. An example of the experiment is shown in Figure 5.

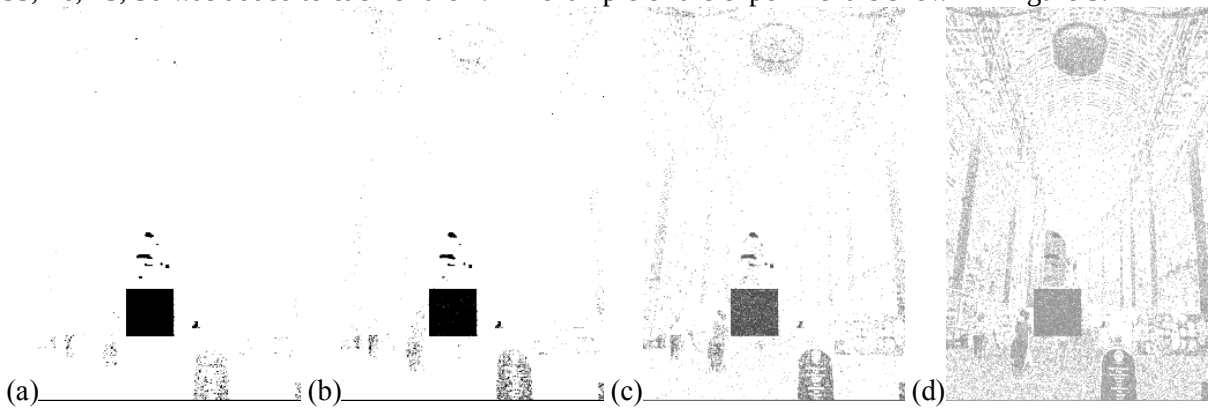


Figure 5. Tampering probability map of the forged image after adding additive Gaussian noise with SNR (dB): a) SNR = 50, b) SNR = 45, c) SNR = 40, d) SNR = 35.

To improve the visual perception of the obtained results, contrast enhancement was applied to all of the tampering probability maps.

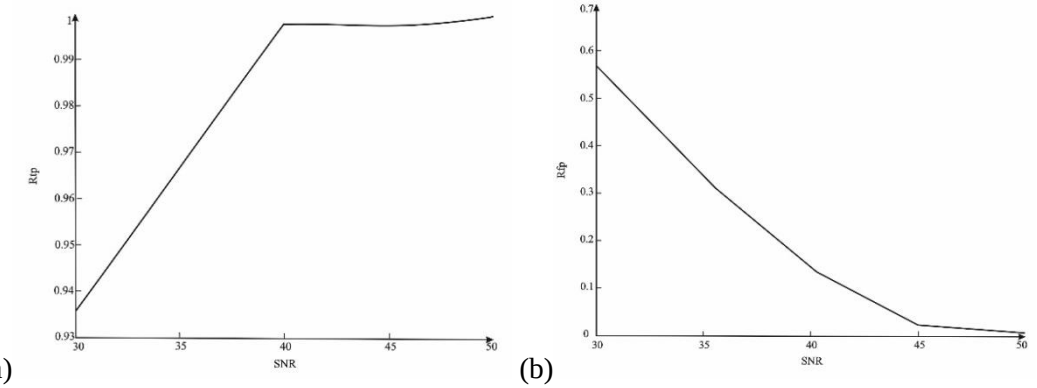
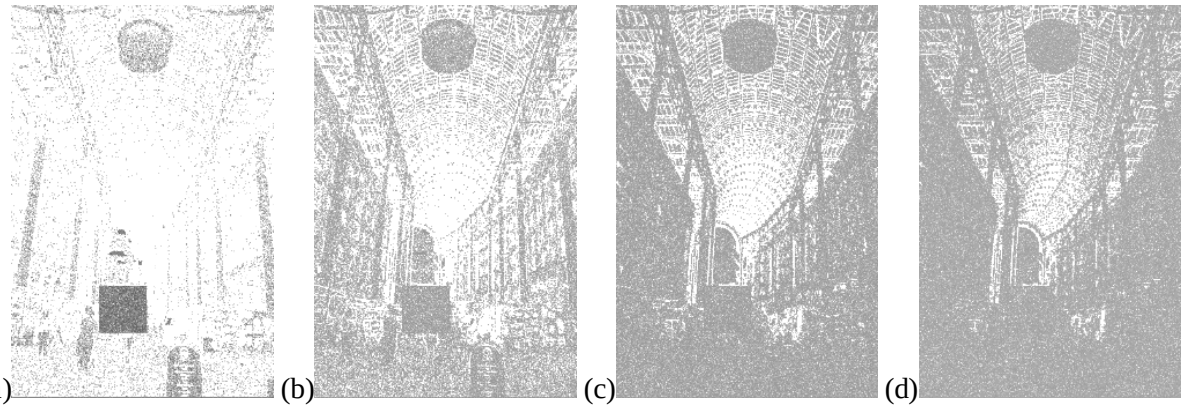


Figure 6. The dependency of quality rates on the SNR (dB): a) $R_{TP}(SNR)$, b) $R_{FP}(SNR)$.

Figure 6 shows the dependency of the quality rates of detection $R_{TP}(SNR)$ and $R_{FP}(SNR)$ on the values of SNR. The results of the experiment showed that the number of correctly detected forged blocks in the image is large for a given range of parameters, but at SNR = 35 dB or less, the number of false alarms of the algorithm increases, so we can say that the method works at values of SNR = 35 dB and above.

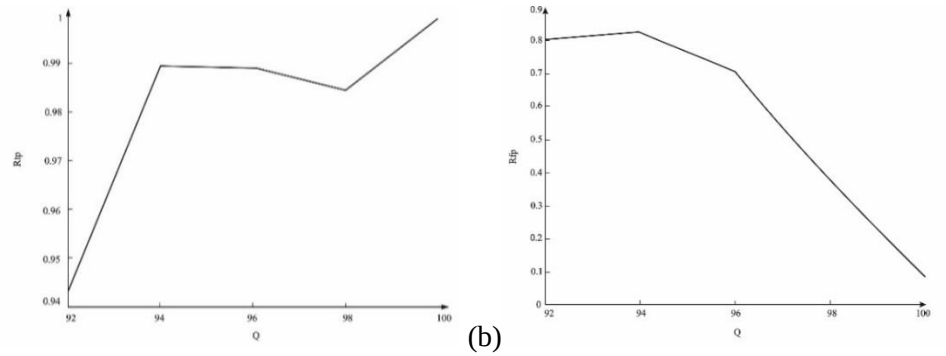
Further, as a part of research, JPEG compression with different values of the quality parameter Q, varying from 0 to 100, was applied to the same set of images. Example of the algorithm performance with the quality parameter values Q = 100, 98, 96, 94 is shown in Figure 7.



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

Figure 7. Tampering map of the forged region after applying JPEG compression noise with the quality parameter values Q : a) Q = 100, b) Q = 98, c) Q = 96, d) Q = 94.

Figure 8 shows the dependency of the quality rates of the detection on the value of the compression quality parameter JPEG Q: $R_{TP}(Q)$ and $R_{FP}(Q)$. From the obtained results, it can be seen that the method does not have the resistance to JPEG compression – even at high quality parameter values the number of false detections is large and already at Q = 92 $R_{FP}(Q) = 0{,}803$. Such a result can be considered a confirmation of the obvious assumptions. The use of the JPEG algorithm destroys the interpolation properties in the image, which leads to a sharp increase in the falsely detectable image fragments.



(a) (b)

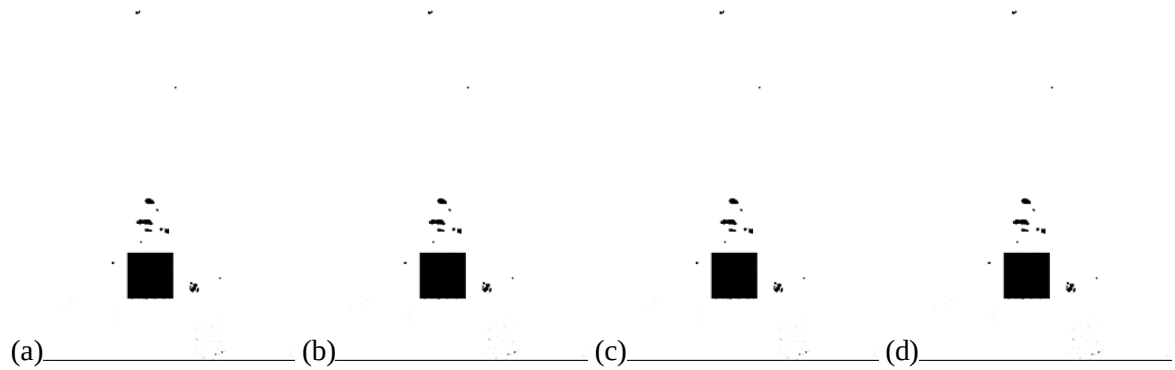
Figure 8. The dependency of quality rates on the values of Q : a) $R_{TP}(Q)$, b) $R_{FP}(Q)$.

As a part of experiments, an investigation of the stability of the method in case the JPEG compression was applied only to a distorted image region was made. It did not affect the detection result, which proves the fact that the algorithm allows to detect pasted regions of a different nature.

As a part of research, we also investigated the robustness of the method against linear contrast. The images entered into the computer are often low contrast, i.e. the variances of brightness function are small in comparison to its average value. Real dynamic range of brightness $[f_{\min}, f_{\max}]$ for such images is much less than the permissible range of brightness scale. The task of contrasting is to "stretch" the real dynamic range to the entire scale. The contrasting was implemented with the help of linear element-by-element conversion: $g = af + b$, where a, b – conversion parameters.

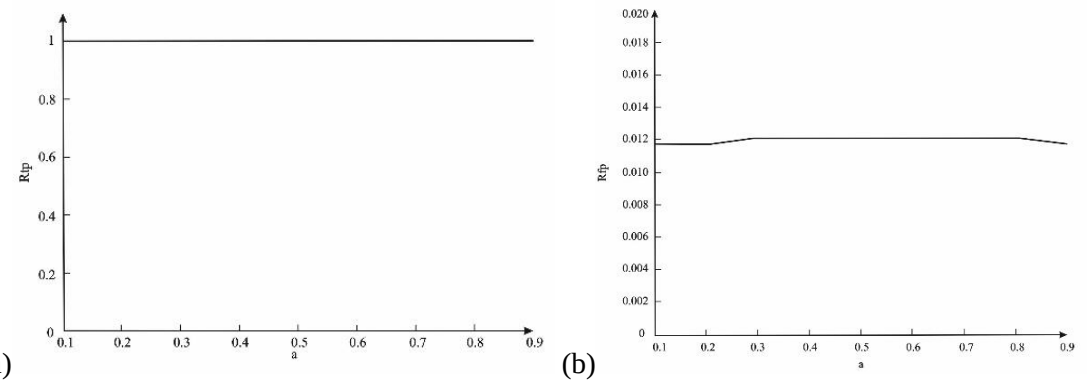
Figure 9 shows an example of how a method works if a linear contrast was applied to the forged image with different values of the transformation parameters. In the presented figure, value of the parameter b was fixed: b = 20. Figure 10 shows the dependency of the quality rates of detection on the value of the transformation parameter a: $R_{TP}(a)$ and $R_{FP}(a)$.

Similarly, an experiment was conducted in which the value of the parameter a was fixed and the parameter values b changed. The results showed that the algorithm is robust against linear contrast and the result of detection of the built-in region does not depend on the values of the parameters of linear contrast.



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

Figure 9. Tampering probably map of the forged image after applying linear contrast with different values of the transformation parameter a : a) a = 0,2; b) a = 0,4; c) a = 0,6; d) a = 0,8.



(a) (b)

Figure 10. The dependency of quality rates on the values of a : a) $R_{TP}(a)$, b) $R_{FP}(a)$.

4. Conclusion

In this paper we consider the method of photomontage detection. It was established that the method allows detecting the built-in areas of various nature and form in images. As the size of the pasted area increases, the quality of detection increases, but the number of false positives increases slightly. The minimum size of the built-in area that can be detected is 2×2.

Experimental studies also showed that the algorithm is robust against such distortions as additive white Gaussian noise at values above 35 dB and linear contrast for any values of the transformation parameters. However, the method proved to be unstable to JPEG compression. Even at high values of the quality parameter, the number of false positives is large.

The method can be used to verify the authenticity of images. It allows to find pasted areas of even very small sizes, but its use is limited (it does not work for detecting pasted regions in compressed images).

5. References

[1] How to deal with fake photo reports (Access mode: https://club.esetnod32.ru/articles/analitika/kak-borotsya-s-poddelkami-fotootchetov/)
[2] Choi C, Lee H and Lee H 2013 Estimation of color modification in digital images by CFA pattern change *Forensic Science International* **226** 94-105
[3] Chakraverti A K and Dhir V 2017 Review on Image Forgery & its Detection Procedure *Journal of Advanced Research in Computer Science* **8(4)** 440-443
[4] Evdokimova N I and Kuznetsov A V 2017 Local patterns in the copy-move detection problem solution *Computer Optics* **41(1)** 79-87 DOI: 10.18287/2412-6179-2017-41-1-79-81
[5] Glumov N I, Kuznetsov A V and Myasnikov V V 2013 The algorithm for copy-move detection on digital images *Computer Optics* **37(3)** 360-367
[6] Burvin P S and Esther J M 2014 Analysis of Digital Image Splicing Detection *Journal of Computer Engineering (IOSR-JCE)* **16(2)** 10-13

[7] Snigdha K M and Ajay A G 2015 Image Forgery Types and Their Detection *Advanced Research in Computer Science and Software Engineering* **5(4)** 174-178
[8] Ferrara P, Bianchi T, Rosa A and Piva A 2012 Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts *IEEE Transactions on Information Forensics and Security* **7(5)** 1566-1577
[9] Popescu A and Farid H 2005 Exposing Digital Forgeries in Color Filter Array Interpolated Images *IEEE Transactions on Signal Processing* **53(10)** 3948-3959
[10] Gallagher A and Chen T 2008 Image authentication by detecting traces of demosaicing *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 1-8 DOI: 10.1109/CVPRW.2008.4562984
[11] Li L, Hue J, Wang X and Tian L 2015 A robust approach to detect digital forgeries by exploring correlation patterns *Pattern Analysis and Applications* **18(2)** 351-365 DOI: 10.1007/s10044-013-0319-9
[12] Bayram S, Sencar H, Memon N and Avcibas I 2005 Source camera identification based on CFA interpolation *IEEE Image Processing* **3** 63-72
[13] Bishop C M 2006 *Pattern Recognition and Machine Learning* (Springer Verlag)
[14] Fawcett T 2006 An introduction to ROC analysis *Pattern Recognition Letters* **27** 861-874 DOI: 10.1016/j.patrec.2005.10.010.
[15] *The original RAW-Samples* (Access mode: http://rawsamples.ch)
[16] *Dcraw* (Access mode: http://www.centrostudiprogressofotografico.it/en/dcraw/)
[17] *Photo database. Zermatt Matterhorn* (Access mode: http://www.zermatt.ch/ru/Media/Media-corner/Photo-database) (30.08.2017)
[18] *Columbia University Image Library (COIL-100)* (Access mode: http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php)

Acknowledgments
This work was supported by the Federal Agency of scientific organization (Agreement 007-GZ/43363/26) in part "The proposed forgery detection method" and by the Russian Foundation for Basic Research (#17-29-03190 - ofi_m) in parts "Experimental results".