Automated Image Splicing Detection from Noise Estimation in Raw Images

Thibaut Julliand*, **Vincent Nozick** † **and Hugues Talbot***

Laboratoire d'Informatique Gaspard-Monge, Equipe A3SI, UMR 8049,
* Université Paris-Est ESIEE, France: thibault.julliand@esiee.fr hugues.talbot@esiee.fr
† Université Paris-Est Marne-la-Vallée, France: vincent.nozick@u-pem.fr

Keywords: Image Forgery, noise, raw image

Abstract

Splicing is a common image manipulation technique in which a region from a first image is pasted onto a second image to alter its content. In this paper, we use the fact that different images have different noise characteristics, according to the camera and lighting conditions during the image acquisition. The proposed method automatically detects image splicing in raw images by highlighting local noise inconsistencies within a quadtree scan of the image. The image noise is modelized by both Gaussian and Poisson noise components. We demonstrate the efficiency and robustness of our method on several images generated with an automated image splicing.

1 Introduction

Over the past few decades, there has been an important rise in the production of digital images. Most of the pictures we see nowadays (press, fashion industry, politics, scientific journals) usually come from a digital camera. The number of falsified photographs has risen as well, becoming more frequent and sophisticated, and created with increasingly powerful tools.

During the past ten years, digital forensics has emerged to help detect falsifications and restore trust towards digital imagery, as explained by Farid in [5]. In this paper, we study a way to counter a specific alteration called splicing, based on the noise characteristics of the image. Our approach especially focuses on raw images where the noise is supposed to be unaltered. Indeed, raw images correspond to the information coming directly from the camera sensors, without any processing like white balance, gamma correction, noise correction, etc. These raw images are harder to tamper with, though it is still possible to alter them, especially via the open file format DNG (Digital Negative) from Adobe. As a consequence, some press and media agencies have started to require raw images as a proof of authenticity. It is therefore timely and important to detect tamperings in this kind of images before using one of them.

1.1 Image splicing

Splicing is a common image manipulation consisting of the insertion in a first image of regions coming from a second, different image. There are various ways to detect such an alteration, some of them dependent on the format of the image. In [19], Popescu and Farid present a method based on the Color Filter Array (CFA) used to transform a raw image into a color image. Producing a full color image from CFA input requires image interpolation, which can be perturbed by splicing. This method, of course, can not be used on raw images.

Some other methods are restricted exclusively to JPEG images: JPEG ghosting, introduced by Farid [6], is based on the potential difference of JPEG quality between the spliced element and the rest of the image. It is also possible to detect an altered image by analyzing the quantization histogram of JPEG images, either on the whole image as proposed by Popescu and Farid [18] or by patch, as performed by He et al. [9] and Lin et al. [12].

There also exist some methods that are either very high-level or have hard to meet prerequisites. Lukas et al. [13] propose a detection method based on the camera fingerprint. However this method requires some unaltered images taken by the camera, or access to the camera itself. It is also possible to use machine learning, such as presented by Bayram et al. [2] or Fu et al. [8]: spliced regions in an image can be detected using a classifier of image features learned from training sets of authentic and forged images. However, this approach is rather high level, while in this article we choose to remain at the level of the camera imaging system.

Finally, some methods use the image noise to detect splicing regions, as detailed in the next section.

1.2 Splicing detection from image noise

All digital sensor capture images perturbed by noise. This noise may be reduced by the camera post-processing or by a lossy compression. However, the remaining noise have consistent parameters throughout the image and this noise can take various form according to the camera model and the lighting conditions during the image acquisition. Thus, noise parameters discrepancy in a zone of the image can be a strong indicator of splicing in this zone.

Mahdian and Saic [15] present a block-based noise inconsistencies approach that relies on the homogeneity of the standard deviation of the Gaussian noise over the image. However, this relies on the assumption that the noise standard deviation is homogeneous over an image which is usually not true, especially due to Poisson noise component. Pan and et al. [16, 17] also propose a block based approach and consider that kurtosis values of natural images in band-pass filtered domains tend to concentrate around a constant value. Although very efficient at detecting noise inconsistencies, their method is not robust against textures and low noise values, making the detection of splicing in natural images difficult. Popescu and Farid [18] in part 5, "Signal to Noise Ratio", introduce another way to identify spliced zones in an image from a noise estimation. However, this method requires a-priori information about the unaltered noise, which is impossible to do with a blind analysis.

Finally, some methods are a combination of various tools. For example, Mahdian and Saic [14] combine resampling detection and noise analysis to highlight suspicious areas.

Most noise estimation methods used in digital forensics simplify the noise as a zero-mean white Gaussian process that is additive and independent of the pixel intensity and localization. Therefore it is possible to reduce the noise estimation to the estimation the variance of this Gaussian process. This ignores the Poisson component of the noise, which is usually dominant in high-intensity areas of an image. Our approach, which is also block-based, aim to get a good estimate of the parameters of the Gaussian and Poisson parts of the noise. The measure of those parameters in sub-images allows to isolate discrepancies, and thus possible splicings. Our study mainly focuses on raw images where noise is supposed to be unaltered.

2 Digital Image Noise estimation

2.1 State of the art

Image denoising is a very active research field in signal processing community. Moreover, most existing denoising methods requires noise parameter estimation before denoising. Hence, some noise parameter estimation studies also exist.

Digital images are subject to noise of two kinds: a Gaussian noise due to some thermal cues of the camera sensor and a Poisson noise from the photonic nature of light. The Gaussian noise affects all the pixels with a constant variance, hence it is more apparent in the low pixel intensities, whereas the Poisson noise component has variance proportional to the intensity, and so ends up dominating in the high pixel intensities. Most noise estimation and removal methods mainly focus on Gaussian noise. However, our tests showed that the Poisson component also contains significant informations useful to distinguish images from different cameras or from different shooting situations. Therefore, the following state of the art presents the methods that can handle both types of noise.

Foi et al. [7] presents a noise estimation method that identifies the Gaussian and the Poisson noise component of an image. They include a clipping stage for higher and lower pixel intensity for robustness purpose. However, this method is also subject to an homogeneous image region search that discards a large part of the pixels of a natural image. Thus this method is not well suited to our purpose, where small region of the image where homogeneous part are too small are also to be considered. Jezierska et al. [10] distinguish pixels that are more subject to Gaussian noise or to Poisson noise from an iterative Expectation-Maximization process. Nevertheless, this method is extremely slow, even for a 80 × 80 pixels, and thus is impracticable for a raw image. It is important to note that the main purpose of the noise estimators is to get a good noise estimation in order to denoise. Our purpose is different and does not require the same characteristics: the discarded areas should be minimal and the noise estimation should be local.

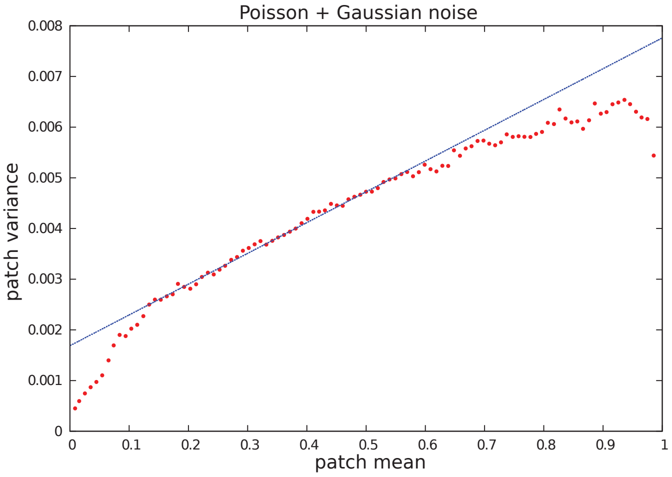


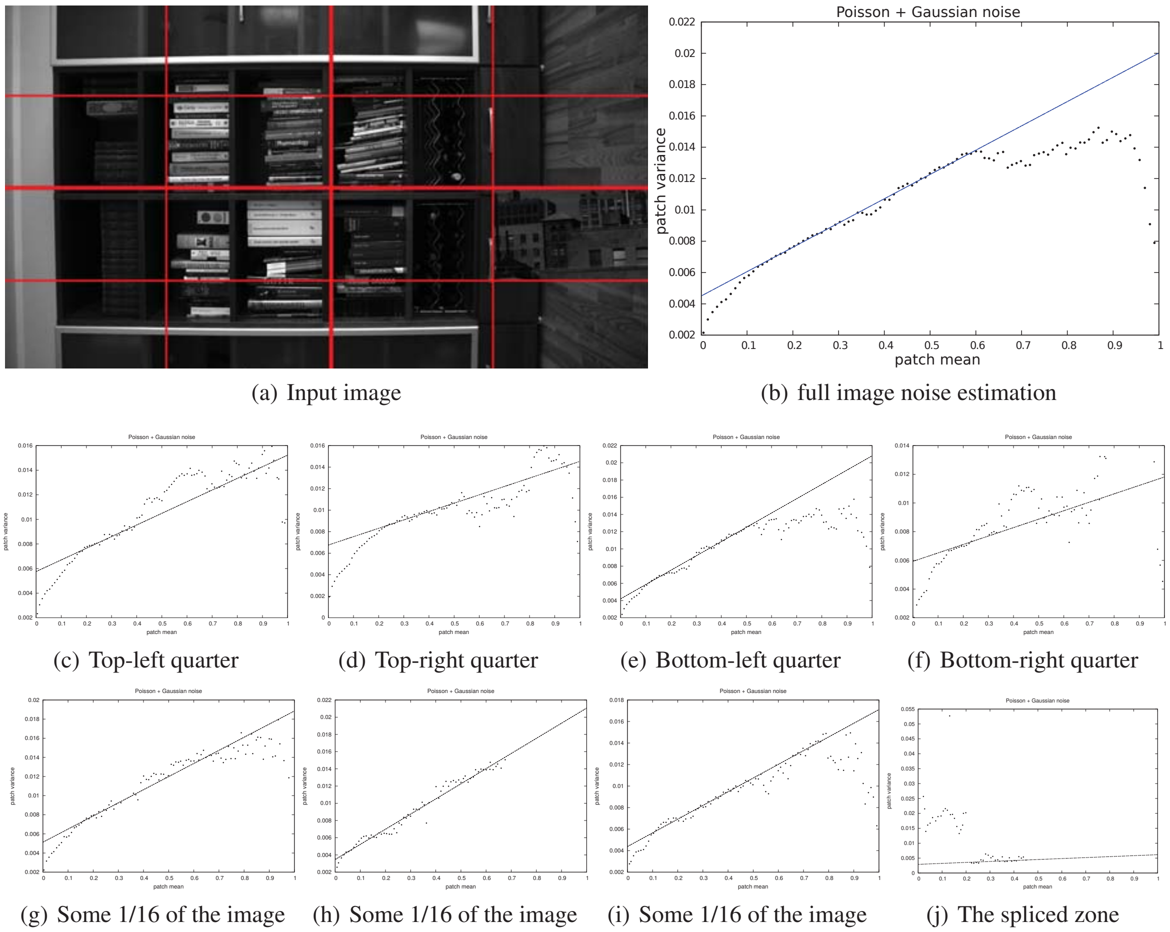
Figure 1. Patches variances according to the patch mean. The lower and higher pixel intensities are discarded. The line is fitted with RANSAC.

Another possibility to estimate the image noise consists in first denoising an image and then subtracting the result from the original noised image. Among the large variety of denoising methods, the non-local means, proposed by Buades et al. [3], performs well. However our tests shown that the method lacks of accuracy in highly textured zones. Moreover, this method only denoises the Gaussian noise component. Jezierska et al. [11] follows their previous work [10] and still suffer from time computation issue. Dabov et al. [4] introduce the so-called BM3D algorithm that performs a 3D collaborative filtering. This technique performs excellent denoising for both homogeneous and textured regions, and denoises the Gaussian noise component as well as the Poisson component.

Our noise estimation process follows the latter approach that provides a noise image.

2.2 Poisson and Gaussian noise parameters

The Poisson and Gaussian noise parameters are estimated from the difference image between the denoised image by BM3D and the original noised image. We first divide the pixel luminance range into n equal intervals $\mathcal{I}_i$, $i \in [1, n]$. The pixels of the denoised image with intensity in $\mathcal{I}_i$ are grouped together to compute a variance σ_i of these pixels in the noise image and



(a) Input image

(b) full image noise estimation

(c) Top-left quarter

(d) Top-right quarter

(e) Bottom-left quarter

(f) Bottom-right quarter

(g) Some 1/16 of the image

(h) Some 1/16 of the image

(i) Some 1/16 of the image

(j) The spliced zone

Figure 2. 2(a): a spliced image. The grid represents the quadtree sub-images. For each curve, the vertical axis represents the patch variance and the horizontal axis the patch mean. 2(b): Line-fitting of the noise data for the full image. From 2(c) to 2(i): Line-fitting of the noise data for various 1/16 subimages. 2(i) Line-fitting of the noise data for the spliced 1/16 sub-image.

a mean value m_i in the denoised image. As specified by [7], the noise that appear in the lower and higher pixel intensities is not reliable. Thus, these pixels are discarded from our noise estimation process. The plot of the variance as a function of the mean gives a line that can be fitted with RANSAC line fitting. The line slope corresponds to the Poisson noise parameters whereas the "y-intercept" corresponds to the Gaussian noise component. An example of this fitting is depicted in Figure 1. For numerical robustness purpose, pixel groups with only a few pixels are discarded. In the rare case where a line have negative parameters, either Poisson or Gaussian, this line is discarded and not taken into account for the rest of the process. This can occur when the denoising is suboptimal or when the line fitting fails.

3 Block based approach

The purpose of this paper is to detect a splicing from the difference of noise between the two images (or more) used in the forgery. Our approach consists in the computation of a quadtree of the image regions. The statistics of the falsified region may differ from those of the rest of the image. This section describes how to build this quadtree data and how to detect the falsified zones.

3.1 Image block noise estimation

A significant constraint of BM3D and most of the denoising methods is the requirement of a good noise estimation of the input image before processing. Indeed, the denoising strength will be chosen according to the estimated noise in the image to process. Thus, a falsified region of the image may not be denoised properly during the denoising process.

In practice, the falsified region can be correctly denoised when the image is processed block per block, according to the respective proportion of original and falsified part in the block. In that case, the Poisson and Gaussian component of the falsified part can be estimated properly and compared to the rest of the image.

In this purpose, our method uses a quadtree to decompose the falsified image into sub regions. The quadtree approach is motivated by the noise statistic computation constraints. Indeed, very small regions provide an accurate spatial information of the forgery coordinates but may not have a wide enough

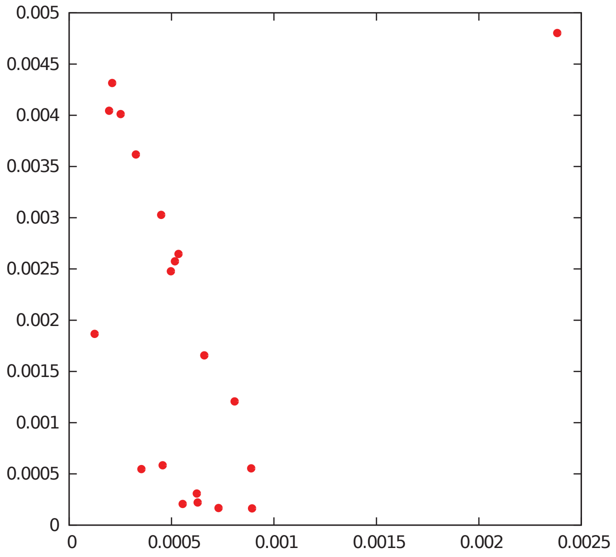


Figure 3. Each point represents the noise of a sub-region of the image. The horizontal axis represents the Gaussian standard deviation of the subimage and the vertical axis the Poisson standard deviation. The isolated point on the top right corner corresponds to the splicing area, its noise parameters are different from the parameters of the rest of the image.

variety of pixel intensity to compute the Poisson and Gaussian noise parameters. On the other hand, large image region can provide robust noise statistics but are not precise enough for forgery localization. It is noticeable that a small non falsified block of the image will not be denoised in exactly the same way it is denoised when the process is performed on the full image. A sample image with some subimages noise fitting is shown in Figure 2.

3.2 Gaussian-Poisson space

The quadtree processed on the falsified image generates a set of image regions. Usually, 3 levels of quadtree, that generate a total of 21 images, are enough. Each image is used to estimate a Poisson and a Gaussian noise component. The blocks corresponding to non altered parts of the image will provide similar noise parameters whereas the region corresponding to the splicing area may provide different noise parameters.

We use a clustering method on the Gaussian-Poisson space to identify the potential suspicious regions. This Gaussian-Poisson space expresses each noise estimation in term of coordinates ($x = gaussian$, $y = poisson$). Similar noise should have similar position. An example of a usual point distribution is shown in Figure 3.

At this step, the point distribution is discriminative enough to visually identify the splicing area but the automatic detection is not straightforward. The next section describes how to identify an outlier.

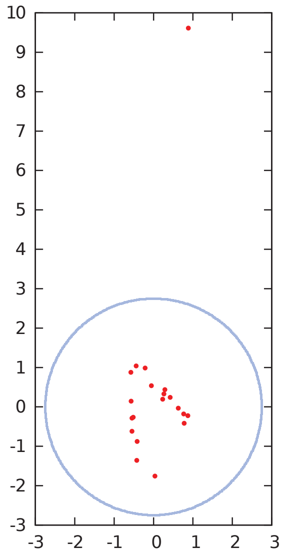


Figure 4. After the robust Principal Component Analysis, the point cloud corresponding to the non-falsified sub-images are grouped in a disc shape. The outlier, in the top can be detected with a σ-clipping.

3.3 Clustering

The point cloud obtained from the Gaussian-Poisson space should be transformed so that the outliers can be detected easily. Ideally, the point cloud of the non-altered sub-image should be grouped in a more compact form. A Principal Component Analysis (PCA) would perform well in a data set without outliers, but in our situation, an outlier located far from the main point group will alter the PCA transformation, due to the L_2-norm sensitivity of the PCA computation.

3.3.1 Robust Principal Component Analysis

To perform a robust PCA, we first remove a large set of potential outliers candidates. A point is considered as an outlier if the distance to its nearest neighbor is the highest of the group. To make our method robust to 2 splicings from the same image, with their representation in the Gaussian-Poisson space with two points near from each other, we better select an outlier as a point whose distance to its second nearest neighbor is the highest of the group.

Then our process removes not only one, but many isolated points with an iterative form of this method. Indeed, the PCA will still perform well if we remove much more points than the expected number of outliers.

Finally, the data normalization along the axis of the PCA space is computed only from the remaining points. Then this normalization is apply to all the data points, including the potential outliers. The initial point distribution that initially roughly looks like an ellipse should finally look like a compact group of points, except for the outliers, as depicted in Figure 4.

3.3.2 Outlier detection

From the compact form of the data described in the previous section, the outliers can easily be extracted by a σ-clipping: all points with distance to the origin higher that twice the average distance from the origin is considered as suspicious.

Moreover, for robustness purpose, the average distance from the origin σ is computed without the 3 points whose distance from the origin is the higher.

Finally, we use a progressive σ-clipping. Data points with a distance to the center of the disc shorter than two σ are considered unaltered. From two σ onward, the probability of a patch being altered increases linearly, until it reaches one when the most distant point or four σ, whichever is greater, is reached. This allows us to evaluate the impact of false positive: while datapoints that are not representative of an altered patch can be found outside of the two-σ boundary, they are often fairly close to it, and thus can be disregarded in comparison with more distant, and thus suspect, points. The boundary values have been chosen empirically, and are the ones that give the best results.

4 Results

4.1 Implementation

The raw images are loaded using LibRaw [1]. We could test our method on various raw image format files, like DNG, RAW, NEF, etc. The denoising process is performed by BM3D using the Matlab code provided by [4]. We empirically chose 100 value intervals to extract the noise statistics in section 2.2, inducing a line fitting over a maximum of 100 points.

4.2 Experiments

We conducted our tests on real data with an automatic process. The database consists in 400 raw images with splicing, and 20 non spliced images to test the robustness of the method against false positive. The spliced images were generated automatically by copying random areas from other images in a random 1/16 of the test image. We conduced the tests on the green channel of every raw image.

On spliced images, we detect a splicing in 84% of the cases, and in 70.6% of those detection the localisation of the splicing is correctly identified. On unspliced images, we have a rate of false positive of 30%. A sample splicing detection is shown on Figure 5. This method, however, has a few weaknesses: first of all, on the size of the spliced areas. An area that is too small will not affect much the noise characteristics of the subimage containing it, and thus will not be detectable. It is also impossible for this method to detect spliced areas with noise characteristics similar to the ones of the image being altered. We have also tested the method in the case of two simultaneous splicing from two different images, with success, see Figure 6. The discrimination method adopted for the PCA offer an important robustness in the case of multiple splicings. We conducted some tests on JPEG files. The results are promising but not convincing yet because of the lossy compression characteristic of JPEG. Indeed, the method seems to measure a kind of JPEG noise rather than the Gaussian and Poisson noise considered in this paper.

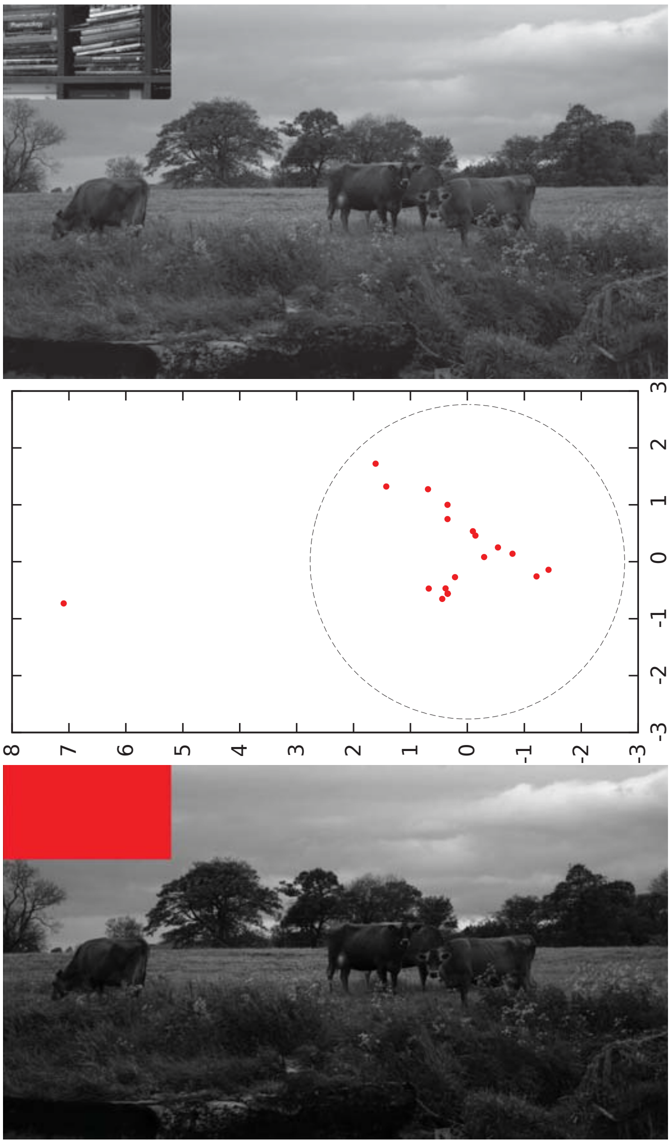


Figure 5. Top: a spliced image. Middle: the robust PCA and the σ-clipping. Bottom: the spliced zone detection (in red).

5 Conclusion

We present a fully automated method to detect splicing in raw digital images. This approach is based on the relative consistency of noise parameters throughout an unaltered image. By looking for inconsistencies in those noise parameters in a quadtree decomposition of the image, which are representative of both the Gaussian and Poisson components of the noise, we show that it is possible to highlight spliced areas in an image. The method is fully automatic, based on a robust PCA of the noise parameters and a subsequent σ-clipping, which detects the potential image splicings, and returns an image showing the probability of falsification by patch.

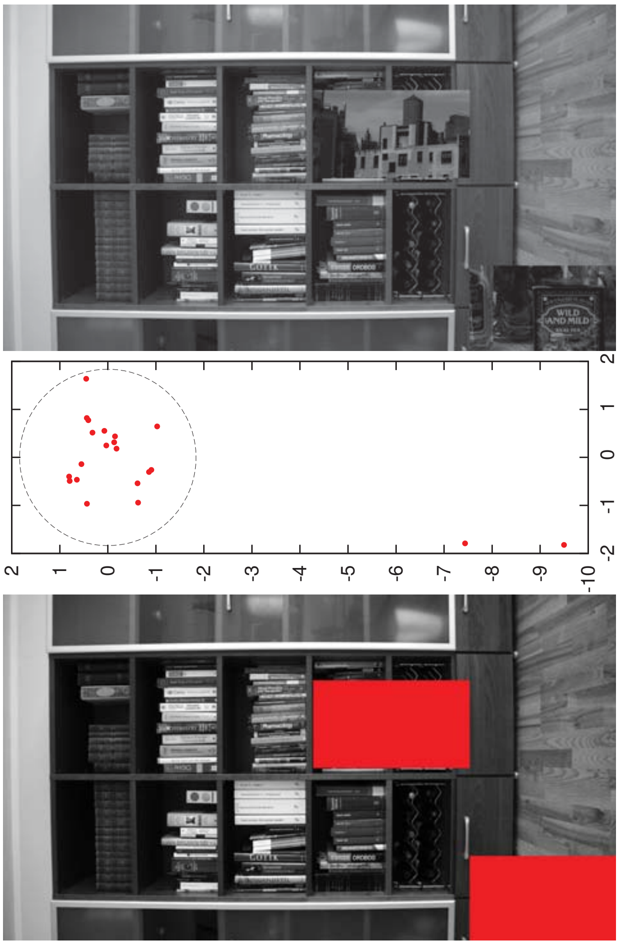


Figure 6. Top: a double spliced image. Middle: the robust PCA and σ-clipping. Bottom: the spliced zones detection (in red).

References

[1] LibRaw 0.17. Image decoder library, 2015.

[2] S. Bayram, I. Avcibas, B. Sankur, and N. D. Memon. Image Manipulation Detection. *Electronic Imaging*, 15(4):1–17, 2006.

[3] A. Buades, B. Coll, and J.-M. Morel. Non-Local Means Denoising. *Image Processing On Line*, 2011.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising By Sparse 3D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, August 2007.

[5] H. Farid. A Survey Of Image Forgery Detection. *IEEE Signal Processing Magazine*, 26(2):26–25, 2009.

[6] H. Farid. Exposing Digital Forgeries From JPEG Ghosts. *IEEE Trans. Inf. Forensics Security*, 4(1):154–160, March 2009.

[7] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian Noise Modeling And Fitting For Single Image Raw-Data. *IEEE Trans. Image Process.*, 17(10):1737–1754, 2008.

[8] D. Fu, Y. Q. Shi, and W. Su. Image Splicing Detection Using 2D Phase Congruency And Statistical Moments Of Characteristic Function. *Proceedings of SPIE Security, Steganography, and Watermarking of Multimedia Contents IX*, 2007.

[9] J. He, Z. Lin, L. Wang, and X. Tang. Detecting Doctored JPEG Images Via DCT Coefficient Analysis. *ECCV*, 2006.

[10] A. Jezierska, C. Chaux, J.-C. Pesquet, and H. Talbot. An EM Approach For Poisson-Gaussian Noise Modeling. In *EUSIPCO*, pages 2244–2248, August 2011.

[11] Anna Jezierska, Emilie Chouzenoux, Jean-Christophe Pesquet, and Hugues Talbot. A Primal-Dual Proximal Splitting Approach For Restoring Data Corrupted With Poisson-Gaussian Noise. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, March 2012.

[12] Z. Lin, J. He, X. Tang, and C. Tang. Fast, Automatic And Fine-Grained Tampered JPEG Images Detection Via DCT Coefficient Analysis. *Pattern Recognition*, 42(11):2492–2501, 2009.

[13] J. Luk, J. Fridrich, and M. Goljan. Detecting Digital Image Forgeries Using Sensor Pattern Noise. *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, 6072:0Y1–0Y11, 2006.

[14] B. Mahdian and S. Saic. Detection Of Resampling Supplemented With Noise Inconsistencies Analysis For Image Forensics. In *International Conference on Computational Sciences and Its Applications*, page 546556, July 2008.

[15] B. Mahdian and S. Saic. Using Noise Inconsistencies For Blind Image Forensics. *Image and Vision Computing*, 2009.

[16] X. Pan, X. Zhang, and S. Lyu. Exposing Image Forgery With Blind Noise Estimation. *The 13th ACM Workshop on Multimedia and Security, Buffalo, NY*, 2011.

[17] Xunyu Pan, Xing Zhang, and Siwei Lyu. Exposing Image Splicing With Inconsistent Local Noise Variances. In *International Conference on Computation Photography (ICCP)*, pages 1–10, April 2012.

[18] A. C. Popescu and H. Farid. Statistical Tools For Digital Forensics. *6th International Workshop on Information Hiding*, 2004.

[19] C. Popescu and H. Farid. Exposing Digital Forgeries In Color Filter Array Interpolated Images. *Signal Processing, IEEE Transactions*, 53(10):1948–3959, September 2005.