Automatic JPEG grid detection with controlled false alarms, and its image forensic applications

Tina Nikoukhah, Rafael Grompone von Gioi, Miguel Colom and Jean-Michel Morel
CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France

Abstract

With the progress of image manipulation tools and the proliferation of fake news and images posted online on social networks, automatic identification of fake content is becoming indispensable. Lossy image compression leaves traces which can be used to recover the history of an image and to help decide about its authenticity. We propose a new JPEG grid detection algorithm. This operation is the first step of many forensic, anti-forensic, and deblocking algorithms. Our analysis is based on the detection of the blocking artifacts and is global and local at the same time. It retrieves the origin of the JPEG grid in all image regions and detects suspicious discrepancies. Our work is based on the a-contrario *framework which reins in the over-detections caused by multiple testing. It also yields a number of false alarms (NFA) which gives extremely secure guarantees for tampering detection. We demonstrate the performance of the proposed method with both quantitative and visual results from well-known image databases.*

1 Introduction

With the growth of social networks, the need for views and ratings has been increasing in the past decade. The evolution of technology has made it possible to publish false content in form of multimedia elements. More particularly, with image editing tools becoming more and more efficient and easy to use, forged images have become a great way to attract viewers.

Doctored photographs are very difficult to identify by visual examination [11]. More than 3.2 billion images are shared each day, which is 100 times more than the amount in 2011. Therefore, the credibility and trustworthiness of digital images has become an important matter, which has led experts to work on image forensic techniques. Indeed, finding digital fingerprints left by image processing and tampering can be used to determine whether an image has undergone modifications [5]. Many studies have been conducted to detect forgery in images [1, 14]. Most of them assume that almost no *a priori* information is available. Therefore, authentication of the image's history needs to be done from only the image itself and without the analysis of its metadata, headers, or file extension.

The digital life cycle of an image can be separated in three phases [22]: acquisition, coding and editing. We will focus here on the coding phase. Indeed, the convenience of desiring smaller amounts of data to store and transmit leads to most digital cameras exporting in JPEG format [26], which is the most common format found online. JPEG images are involved in many forensics situations and the compression history is anyway interesting to recover.

The JPEG procedure starts by partitioning the image into 8×8 non-overlapping blocks. The Discrete Cosine Transform (DCT) is then applied to each block. The DCT coefficients are quantized and finally losslessly encoded.

Figure 1: **Close view of block artifacts for JPEG compressed image "Lena" with quality factor $Q = 10$.**

Since the quantization process of JPEG compression is performed on each image block independently, blocking artifacts, as shown in figure 1, appear at block boundaries between adjacent blocks in the image. These characteristic compression traces can be analyzed both in the pixel domain and the transform domain. These artifacts, which degrade the quality of the image [17], yield very informative cues in forensic analysis.

The outline of the article follows. Section 2 reviews the state of the art on the techniques which are used to infer the JPEG compression history. Section 3 presents our method

in the context of the *a-contrario* theory. This is our main theoretical contribution to the topic. Section 4 presents our results assuming different applications.

2 State of the Art

A high number of forensic techniques have been designed to authenticate JPEG compression history. For the most part, these techniques have the same three step structure [24, 14]:

- grid detection, or block detection;
- quantization estimation;
- double JPEG compression detection.

Some techniques skip the first step because of a lack of a solid theoretic framework. Most methods are based on the analysis of DCT coefficients [3, 19] to estimate the original quantization table which is subsequently used locally to identify tampered areas. In the transform domain, block-based image coding schemes modify the histogram of transformed coefficients. In consequence several methods analyzing the shapes of these functions are proposed in the literature [20, 23].

The blocking artifacts have a regular pattern, since the quantization of the DCT coefficients is done separately on 8×8 disjoint blocks. Image content and dynamic makes these discrepancies hard to detect in the Fourier domain [2]. This leads to work directly in the spatial domain, and in particular, on the luminance component I. Denoting by R, G and B the color components of the pixel, the standard definition [25] of I is

$$I = 0.299\ R + 0.587\ G + 0.114\ B.$$

The authors of [9] and [10] proposed algorithms which are based on the idea that if the image has been compressed, then the pixel differences across block boundaries are significantly different from those within blocks. The energy differences are compared to a threshold to deduce the presence of a prior compression. However, their method is a statistical estimate over the whole image and therefore gives a global result, not a local JPEG grid estimate.

The approach in [16] is capable of detecting and localizing tampered areas but is, nevertheless, sensitive to image content and suffers from high false detection rates. This is problematic for an automatic analysis.

Lin et al. [17] present a robust grid extraction method with an estimation based on a maximum likelihood method, introduced by [18]. Their forgery detection technique is based on two passes: one to estimate the main grid of the image and one to identify blocks which do not coincide with the global estimation. Their two-step technique is less sensitive to the image content and is capable of localizing tampered areas. Nevertheless their method depends on the tuning of several parameters (thresholds and attenuation values). This makes it difficult to render the detection method fully automatic.

A recent survey [14] states that "*all the approaches and methodologies [...] have the capacity to recognize fraud. In any case, a few algorithms are not viable regarding identifying actual forged regions. On the other hand some algorithms have a time complexity problem. So, there is a need to develop an effective (efficient) and accurate image forgery detection algorithm.*" A solid theoretical mathematical framework describing the statistical behavior of the quantities involved is also desired for each image forensic technique [21, 24]. Indeed, although some methods achieve excellent results in certain experimental settings, the absence of a generalized model might result in non-controllable performance when the setting is modified since the parameters of the methods change as well.

In this paper, we present an accurate method to estimate the grid origin of a JPEG image (globally and locally), which in most cases is, as mentioned previously, the first step of image forgery techniques. The method is based on three steps: extracting the block artifacts, decomposing the image into several voters, and evaluating the accuracy of the statistical estimation based on the *a-contrario* method. The estimation is controllable with an *a priori* number of false alarms for each detection which will be detailed in section 3.3. Furthermore, the proposed method does not suffer of a complexity problem since it can be parallelized. We show several applications for the method.

3 An *A-Contrario* Detector

3.1 Grid extraction

Let I be the $X \times Y$ luminance component of the input image and $I(x, y)$ the intensity value of pixel (x, y), with $0 \leq x \leq X - 1$ and $0 \leq y \leq Y - 1$.

The method [18] detects the presence of block artifacts by computing the absolute value of the gradient magnitude image. Indeed, the block artifacts are represented by horizontal and vertical abrupt changes in the luminance value image. The difference filter used is defined as follows

- horizontally:

$$dI(x, y) = |I(x, y) - I(x - 1, y)|$$

- vertically:

$$dI(x, y) = |I(x, y) - I(x, y - 1)|$$

Other authors [16] use second order differences defined as

- horizontally:

$$dI(x, y) = |2I(x, y) - I(x + 1, y) - I(x - 1, y)|;$$

- vertically:

$$dI(x,y) = |2I(x,y) - I(x,y+1) - I(x,y-1)|.$$

As can be seen in figure 2, the first difference and second difference filters are highly affected by the edges and textures in the image. The latter image dynamics is neither vertical nor horizontal. To reduce these interferences, a cross difference filter proposed in [4] is defined by

$$dI(x,y) = |I(x,y)+I(x+1,y+1)-I(x+1,y)-I(x,y+1)|.$$

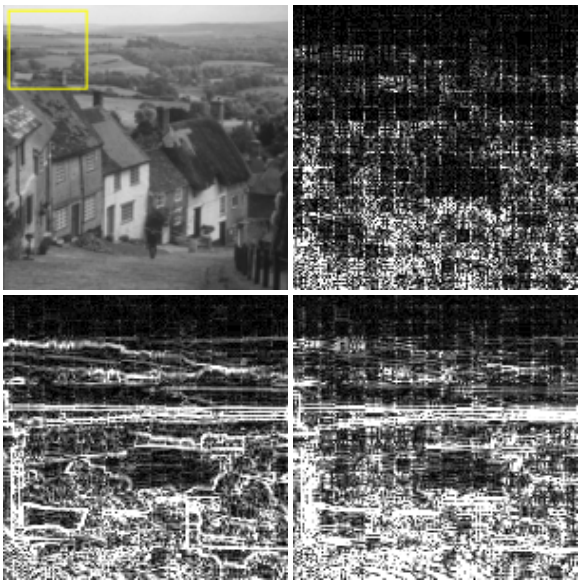

Figure 2: **Compressed image "Goldhill" with quality factor $Q = 85$ and close view of its cross difference, first order difference and second order difference.**

As figure 3 shows, the higher the compression quality, the dimmer the JPEG grid. This explains the limits of grid extraction methods. Other techniques [16, 18] add a nonlinear correction to enhance the JPEG artifact over the strong edges in the image. Thanks to the locality of our method, we shall not need this correction.

3.2 The voting process

The voting process consists in decomposing the cross difference image into overlapping test blocks. Each block has a say and votes, independently, for its grid origin. The blocks are of size multiples of B_s (block size), which results in N independent (therefore parallelizable) tests to perform.

$$N = \frac{1}{4}\frac{X}{B_s}\left(\frac{X}{B_s}+1\right)\frac{Y}{B_s}\left(\frac{Y}{B_s}+1\right)$$

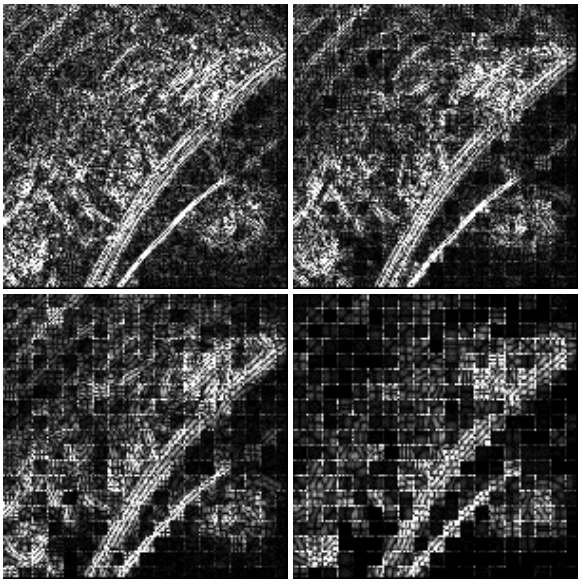

Figure 3: **Comparison of cross images for different JPEG compression quality factors: 95, 85, 50 and 15.**

B_s makes sense from $B_s > 32$ so that each test block has at least 4 repetitions of the JPEG 8×8 blocking artifact. This also implies that tampered regions as small as 32×32 can already be detected. Of course if B_s is chosen too high for the application of forgery detection, it will not detect small modifications. Using overlapping test blocks is a form of multi-scale approach.

Each block b votes for the main grid by looking at the horizontal and vertical local maxima *separately*. Each direction (horizontal or vertical) has 8 different possible grid origins, since a typical JPEG block is of size 8×8. Algorithm 1 describes the voting process.

Algorithm 1: Block voting

Data: Block b
Result: Vote
forall *horizontal local maxima* **do**
  $x \leftarrow$ first coordinate of local maximum
  vote_x [x `mod` 8] ++ ;
forall *vertical local maxima* **do**
  $y \leftarrow$ second coordinate of local maximum
  vote_y [y `mod` 8] ++ ;
n_x, n_y $\leftarrow$ sum(vote_x), sum(vote_y): total number of local maximums horizontal, vertical;
k_x, k_y $\leftarrow$ max(vote_x), max(vote_y): number of votes of the elected coordinates ;

Algorithm 1 returns the total number of local maxima (n_x and n_y) which represent the number of voters and the

number of votes (k_x and k_y) given for each block. Let us take the example of the "Goldhill" image, a 512×512 greyscale image compressed with JPEG quality factor $Q = 85$, seen on the top left of figure 2. In this example B_s was set to 100 resulting in 225 test blocks. Table 1 shows the votes obtained on one block. The grid origin abscissa 0 has a strong relative majority. The second row shows the result for the horizontal extrema, with again a significant vote for the 0 position. The next section will explain the proposed detection criterion based on these statistics and controlling the number of false alarms.

Table 1: **Statistics extracted from an image block. The first row shows the number of local vertical extrema at each pixel abscissa, modulo 8.**

	0	1	2	3	4	5	6	7
x	**604**	142	244	239	253	244	213	156
y	**509**	181	240	255	259	233	210	152

3.3 Statistical validation

The proposed validation step is based on the *a-contrario* framework [7]: an event of interest is called meaningful if its occurrence is non-accidental, in the sense that the relation between its parts is too regular to be the result of an accidental arrangement of independent parts. Indeed, the grid estimation is based on the regularity of the pattern left by the JPEG compression.

The mathematical setting corresponds to a multiple testing procedure to control the expected number of false detections under a null model H_0 [12]. The Number of False Alarms (NFA) of the event e is defined as

$$\mathtt{NFA}(e) = N_{test} P_{H_0}(e)$$

where N_{test} is the number of events to be tested and $P_{H_0}(e)$ is the probability of observing an event e (or better) under the stochastic model H_0. An event e is called ϵ-meaningful if and only if $\mathtt{NFA}(e) < \epsilon$.

In our situation, the *a-contrario* framework determines whether a block's vote is significant or not. Each block has two events to test: the horizontal and the vertical JPEG fingerprints. A test block is called significant when both of these events are ϵ-meaningful, *i.e.*, $\mathtt{NFA}_x < \epsilon$ and $\mathtt{NFA}_y < \epsilon$.

With $N_{test} = N$ being the number of blocks, we have

$$\mathtt{NFA}_x = N_{test} \mathcal{B}(n\_x, k\_x, 1/8),$$

$$\mathtt{NFA}_y = N_{test} \mathcal{B}(n\_y, k\_y, 1/8),$$

where $\mathcal{B}(n, k, p)$ is the binomial tail

$$\mathcal{B}(n, k, p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}.$$

As a simple convention, Desolneux et al. [7] suggest using $\epsilon = 1$, which is done in other fields using *a-contrario* methods [13]. Indeed, setting the value of ϵ to 1 implies getting, on average, one false detection (one wrong block) per image.

In summary, for each test block we obtain a pair of NFA values for the most voted grid position. If both values are less than 1, we consider that the grid position is significant. This ensures that in an uncompressed image, there should be less than 1 false detection. The ensuing detection algorithm is summarized in Algorithm 2. Regarding the image "Goldhill" taken as an example, the result is an overwhelming vote for the grid position $(0, 0)$. Indeed, 225 blocks over 225 blocks voted significantly for it. But the main point here is the NFA value of this detection. For the "most significant" block, with lowest NFA, this value is $\mathtt{NFA} = 10^{-785.487}$. Most test blocks having extremely significant NFA values, ensures that the detection could "never happen by chance".

Algorithm 2: Summary of the proposed algorithm

```
Data: Input RGB image
Block size N
Result: Images with different main grids
Compute greyscale image;
Compute cross difference image ;
Decompose into blocks ;
forall blocks do
    Vote, see Algorithm 1 ;
    NFA_x, NFA_y ← horizontal NFA, vertical NFA;
    if NFA_x < 1 and NFA_y < 1 then
        Status(b) ← significant;
Gather all votes and present result;
```

4 Results on Several Applications

4.1 Grid detection

Grid detection is our main application as it represents the first step of most forgery detection algorithms. But this is not the only application. In image restoration, grid detection is also used to remove grid artifacts by a deblocking procedure [6]. To do so, it is useful to detect the grid in every case, and the hardest cases are when the compression level is low.

The challenge here is to detect a grid even for high quality compressed images. Table 2 shows that a very significant detection is possible up to $Q = 95$. Reliable detections for Q values up to 98 are also observed. On the other hand, the original uncompressed images do not produce detections (as expected). In contrast, the method proposed by [10], and applied to the same images than in Table 2, does not work for Q values over 90. On another hand, the method proposed in [17] gets similar results to ours, yet requires to adjust properly two parameters, an attenuation value α and a threshold θ, while our method is parameter-free.

Table 2: **Results on standard images for several quality factors using the proposed algorithm.**

Image	Original	Q98	Q95	Q90
Barbara	$10^{0.822}$	$10^{0.016}$	$\underline{10^{-42.29}}$	$\underline{10^{-280.4}}$
Lena	$10^{1.257}$	$10^{1.335}$	$\underline{10^{-48.32}}$	$\underline{10^{-418.3}}$
Cameraman	$10^{0.159}$	$\underline{10^{-0.058}}$	$\underline{10^{-0.338}}$	$\underline{10^{-120.17}}$
Goldhill	$10^{0.428}$	$10^{0.695}$	$\underline{10^{-39.16}}$	$\underline{10^{-383.82}}$
Peppers	$10^{2.203}$	$10^{0.056}$	$\underline{10^{-2.658}}$	$\underline{10^{-131.47}}$

4.2 Crop detection

Table 3 reports the overall results, described in terms of correct percentage of the cropping position detection, depending on the compression ratio. The Kodak standard dataset [15] was used for that purpose, as it is good quality and compression-free. The detection rate decays significantly for $Q \geq 90$. Notice that cropping might have occurred just by chance with an origin compatible with the original grid. This actually happens one out of 64. Hence, there is a minimal 1/64 false negative rate, which is clearly unavoidable.

Table 3: **Results of the proposed method on cropped images of the Kodak database.**

Quality factor	Accuracy
≤ 80	100 %
90	91 %
95	70 %
99	41 %

4.3 Copy-paste tampering detection

Again for this task, a detection based on a disparity in grid position in some block may fail with probability 1/64, when the copied area is placed so that its grid is aligned with the global grid. To test how detection can be based on JPEG grid misplacement, we used the database of tampered images [8]. Figures 4 and 5 represent tampered images and their ground truth. They come from the folder *C1_panasonic* folder of the benchmark data [8], and were created by coping and pasting. The copied area is taken from the same image and its borders hidden in a smooth transition. However, we do not use this information for our detection, which would work equally well if the copied area came from a different JPEG image.

Figure 4: **Image "Hedge" from database [8]: original, forged and results: over** 1980 **blocks,** 1952 **voted significantly for** $0, 0$ **and** 21 **for** $(5, 4)$**.**

Figure 5: **Image "Supermarket" from database [8]: original, forged and results: over** 1980 **blocks,** 1974 **voted significantly for** $(0, 0)$ **and** 6 **for** $(0, 5)$**.**

5 Conclusion

In this paper we have presented an accurate method for grid origin detection from a given image with no prior

information. We proposed a way to validate this step with a fully unsupervised parameter-less algorithm. Our grid only method is local enough to detect tampering such as crop and copy-paste, without any further step. It does not require extra information on the JPEG quantization and does not require involving the computation of DCT coefficients. In future work, we will aim at extending the method to detect locally double compression with a shifted JPEG grid (and therefore detecting a principal and a shifted grid).

Our method is only one of the steps of a tampering detection chain. For JPEG images, tampering detection attempts can go on, even if no discrepancy has been found in the grid origin throughout the image. But knowing the grid origin enables an accurate analysis of the statistics of block DCT coefficients, which is the classic next step in tampering detection. Being only one (significant) step in the detection chain, grid detection must be fully automatic and offer strong guarantees. For this reason, we also believe that the *a-contrario* methods, able to attach very small NFAs to detections, should also be used for the other detection tasks.

References

[1] G. K. Birajdar and V. H. Mankar. Digital image forgery detection using passive techniques: A survey. *Digital Investigation*, 10(3):226 – 245, 2013.

[2] A. R. Bruna, G. Messina, and S. Battiato. *Crop Detection through Blocking Artefacts Analysis*, pages 650–659. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[3] Y. Cao, T. Gao, L. Fan, and Q. Yang. A robust detection algorithm for copy-move forgery in digital images. *Forensic Science International*, 214:33–43, 2012.

[4] Y.-L. Chen and C.-T. Hsu. Image tampering detection by blocking periodicity analysis in jpeg compressed images. In *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pages 803–808, Oct 2008.

[5] Z. Chen, Y. Zhao, and R. Ni. Detection of operation chain: JPEG-resampling-JPEG. *Signal Processing: Image Communication*, 57(Supplement C):8 – 20, 2017.

[6] J. Chou, M. Crouse, and K. Ramchandran. A simple algorithm for removing blocking artifacts in block-transform coded images. *IEEE Signal Processing Letters*, 5(2):33–35, Feb 1998.

[7] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis*. Springer, 2008.

[8] F.-A.-U. Erlangen-Nurnberg. Image Manipulation Dataset. `http://www5.cs.fau.de/research/data/image-manipulation/`. [Online; accessed 20-December-2017].

[9] Z. Fan and R. L. de Queiroz. Maximum likelihood estimation of JPEG quantization table in the identification of bitmap compression history. *ICIP*, pages 948–951, 2000.

[10] Z. Fan and R. L. de Queiroz. Identification of bitmap compression history: JPEG detection and quantizer estimation. *IEEE TIP*, 12(2), 2003.

[11] H. Farid. Digital doctoring: how to tell the real from fake. *Significance*, 3(4):162–166, 2006.

[12] A. Gordon, G. Glazko, X. Qiu, and A. Yakovlev. Control of the mean number of false discoveries, Bonferroni and stability of multiple testing. *The Annals of Applied Statistics*, 1(1):179–190, 2007.

[13] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: a line segment detector. *IPOL*, 2012.

[14] A. Kashyap, R. S. Parmar, M. Agarwal, and H. Gupta. An evaluation of digital image forgery detection approaches. *CoRR*, abs/1703.09968, 2017.

[15] Kodak Lossless True Color Image Suite. `http://r0k.us/graphics/kodak/`. [Online; accessed 20-December-2017].

[16] W. Li, Y. Yuan, and N. Yu. Passive detection of doctored JPEG image via block artifact grid extraction. *Signal Processing*, 89(9):1821 – 1829, 2009.

[17] C.-S. Lin and J.-J. Tsay. Passive forgery detection for JPEG compressed image based on block size estimation and consistency analysis. *Applied Mathematics and Information Sciences*, 9(2):1015–1028, 2015.

[18] W. Lin, S. Tjoa, H. Zhao, and K. Ray Liu. Digital image source coder forensics via intrinsic fingerprints. *IEEE Trans. on Information Forensics and Security*, 4(3):460 – 475, Sept 2009.

[19] Z. Lin, J. He, X. Tang, and C.-K. Tang. Fast, automatic and fine-grained tampered JPEG image detection via dct coefficient analysis. *Pattern Recognition*, 42:2492 –2501, 2009.

[20] W. Luo, J. Huang, and G. Qiu. JPEG error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 5(3):480–491, Sept 2010.

[21] C. Pasquini, G. Boato, and F. Pérez-González. Statistical detection of JPEG traces in digital images in uncompressed formats. *IEEE Transactions on Information Forensics and Security*, 12(12):2890 – 2905, Dec. 2017.

[22] A. Piva. An overview on image forensics. *ISRN Signal Processing*, 2013(496701), 2013.

[23] S.M.Ye, Q. Sun, and E. Chang. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. *IEEE International Conference on Multimedia and Expo, Beijing, China*, pages 12 – 15, 2007.

[24] T. H. Thai, R. Cogranne, F. Retraint, and T.-N.-C. Doan. JPEG quantization step estimation and its applications to digital image forensics. *IEEE Trans. Inf. Forensics Security*, 12(1), Jan. 2017.

[25] M. Tkalcic and J. F. Tasic. Colour spaces: perceptual, historical and applicational background. In *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, volume 1, pages 304–308 vol.1, Sept 2003.

[26] G. K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 1991.