

## Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps

Tiziano Bianchi and Alessandro Piva

**Abstract**—In this paper, a simple yet reliable algorithm to detect the presence of nonaligned double JPEG compression (NA-JPEG) in compressed images is proposed. The method evaluates a single feature based on the integer periodicity of the blockwise discrete cosine transform (DCT) coefficients when the DCT is computed according to the grid of the previous JPEG compression. Even if the proposed feature is computed relying only on DC coefficient statistics, a simple threshold detector can classify NA-JPEG images with improved accuracy with respect to existing methods and on smaller image sizes, without resorting to a properly trained classifier. Moreover, the proposed scheme is able to accurately estimate the grid shift and the quantization step of the DC coefficient of the primary JPEG compression, allowing one to perform a more detailed analysis of possibly forged images.

**Index Terms**—Image forensics, JPEG artifacts, nonaligned double JPEG compression, threshold detector.

### I. INTRODUCTION

Almost everybody today has the possibility of recording and sharing a large amount of digital images, enabling the virtually instantaneous diffusion of visual information regarding people, events, and so on. At the same time, the large availability and the ease of use of commercial image processing tools make it extremely simple to alter the content of digital images, so that the message or the emotion conveyed by an image can be easily manipulated. This can be a serious problem when the content of an image is used to influence the opinion of viewers, like in tribunals, in journals, and in medical applications.

Image forensics is a new research area aiming at assessing the credibility of digital images by looking for possible inconsistencies in statistical or geometrical features, like JPEG quantization artifacts, interpolation effects, and demosaicing traces [1]. An advantage of this approach is that it is passive, in the sense that it is not necessary to embed an explicit fingerprint in the digital image after its acquisition, making forensics tools applicable even in the case of images captured by standard commercial cameras.

Due to the fact that a vast amount of digital images is stored in JPEG format, several forensics tools have been developed to detect the presence of tampering in this class of images. The presence of tampering can be revealed by analyzing some artifacts introduced by JPEG recompression occurring when the forgery is created; the tools can work in presence of an image exhibiting an aligned double JPEG compression (i.e., where the discrete cosine transform (DCT) grids of successive JPEG compressions are perfectly aligned), or a nonaligned double

JPEG compression. Approaches working in the first scenario include [2]–[5]. In [2], the author proposes to detect areas which have undergone a double JPEG compression by recompressing the image at different quality levels and looking for the presence of so-called *ghosts*; in [3], double JPEG compression is detected by computing a tampering probability map of the image according to a proper statistical model of DCT coefficients. In [4] and [5], the statistical distribution of first digits in quantized DCT coefficients is analyzed for detecting double JPEG compression.

The scenario of nonaligned double JPEG compression has been investigated in [6]–[9]. In [6], a blocking artifact characteristics matrix (BACM) is computed in the pixel domain to measure the symmetrical property of the blocking artifacts introduced by JPEG compression. The method is based on the observation that when a JPEG image is recompressed with a nonaligned DCT grid, the regular symmetry of BACM will be destroyed. In [7] and [9] also, the blocking artifacts in the pixel domain are investigated, but in this case their periodic property is measured by devising two proper sets of features: indeed, when an image exhibits a nonaligned double JPEG compression, this blocking periodicity will be perturbed. In [8], the statistics of DCT coefficients is represented as a noisy convolutive mixing model; starting from the observation that nonaligned JPEG recompression weakens the independency between DCT coefficients, an independent component analysis (ICA)-based method is designed to derive the asymmetry of an independent value map (IVM) as a normalized criteria of the coefficient independency.

In this paper, we propose a novel technique to detect the presence of nonaligned double JPEG (NA-JPEG) compression. Differently from [6]–[9], in which NA-JPEG compression is detected by training a classifier on a set of features, our approach relies on a single yet powerful feature derived from the statistics of DCT coefficients, allowing us to apply a simple threshold detector. Moreover, the proposed approach is able to estimate both the grid shift and the quantization step of the DC coefficient of the primary compression. Such information can be used to perform a more detailed analysis of a possibly forged image.

The rest of this paper is organized as follows. In Section II, the possible scenarios where an algorithm for the detection of nonaligned double JPEG compression can be useful are briefly reviewed. In Section III, the proposed algorithm is described. The results of the experimental analysis carried out to assess the performance of the proposed scheme are discussed in Section IV. Finally, in Section V, our conclusions are drawn.

### II. FORENSIC SCENARIO AND PREVIOUS WORK

The classical scenario in which NA-JPEG double compression occurs is that of image splicing. In this kind of forgery, it is assumed that a region from a JPEG image is pasted onto a host image and that the resulting image is JPEG recompressed. Assuming a random placement of the forged region, there is a probability of 63/64 that the forged region will exhibit NA-JPEG compression artifacts. This is the same scenario considered in [6] and [8].

An alternative scenario for NA-JPEG compression could consider the following forgery model: an original JPEG image is locally modified using an image processing technique which disrupts JPEG compression statistics, then randomly cropped and recompressed in JPEG format. Examples of local tampering which destroys JPEG statistics could be a cut and paste from either a noncompressed image or a resized image, or the insertion of computer generated content. This is similar to the scenario considered in [3], with the difference that in our scenario the nontampered region exhibits NA-JPEG compression instead of aligned double JPEG compression.

Manuscript received March 11, 2011; revised September 20, 2011; accepted September 24, 2011. Date of publication October 10, 2011; date of current version March 08, 2012. This work was supported in part by the REWIND Project funded by the Future and Emerging Technologies (FET) programme within the 7FP of the European Commission, under FET-Open Grant 268478, and by the project LivingKnowledge-Facts, Opinions and Bias in Time funded by the European Commission under Contract 231126. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alex ChiChung Kot.

The authors are with the CNIT Research Unit and the Department of Electronics and Telecommunications, University of Florence, I-50139 Florence, Italy (e-mail: tiziano.bianchi@unifi.it; alessandro.piva@unifi.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIFS.2011.2170836

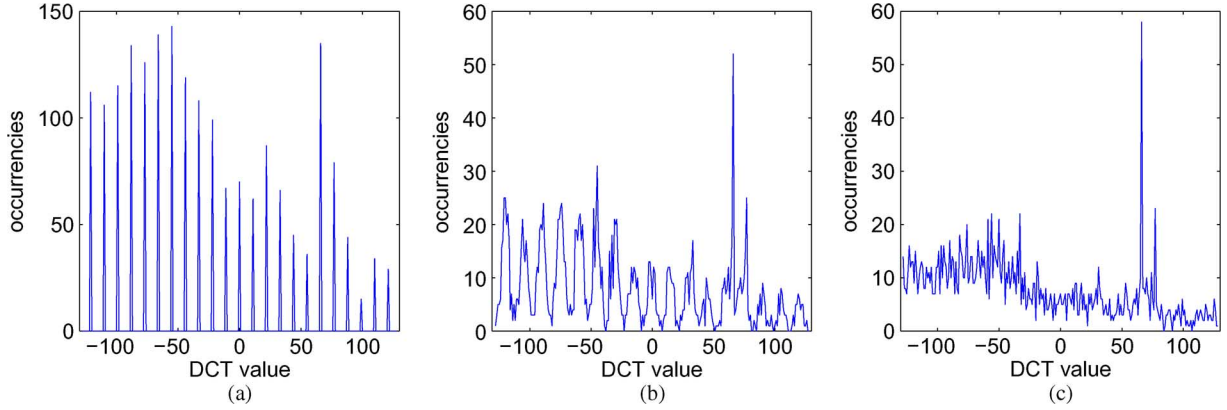


Fig. 1. Histogram of DC coefficients for different shifts of the DCT grid: (a)  $(i, j) = (0, 0)$ , (b)  $(i, j) = (y, x)$ , and (c) random shift.

In both cases, a tool able to identify NA-JPEG double compression could be used to discern original regions from tampered ones, for example using a segmentation of the image under test, like the approach proposed in [10]. As a matter of fact, the two forensics scenarios are complementary: the only difference is in the interpretation of the NA-JPEG compressed regions as either tampered (in the first scenario) or original (in the second one). We note that existing forensic tools can be used in both situations, so that they can be directly compared with the proposed approach.

### III. PROPOSED METHOD

Let us assume that an original image  $\mathbf{I}_1$  is JPEG compressed with a quality factor  $QF_2$  and then decompressed. The obtained image  $\mathbf{I}_2$  can be modeled as follows:

$$\mathbf{I}_2 = \mathbf{D}_{00}^{-1} \mathcal{Q}_2(\mathbf{D}_{00} \mathbf{I}_1) + \mathbf{E}_2 = \mathbf{I}_1 + \mathbf{R}_2 \quad (1)$$

where  $\mathbf{D}_{00}$  models an  $8 \times 8$  block DCT with the grid aligned with the upper left corner of the image,  $\mathcal{Q}_2(\cdot)$  models quantization and de-quantization processes with JPEG quantization table corresponding to a quality factor  $QF_2$ , and  $\mathbf{E}_2$  is the error introduced by rounding and truncating the output values to eight bit integers. The last quantity  $\mathbf{R}_2$  can be thought of as the overall error introduced by JPEG compression with respect to the original image.

Let us now suppose that the original image  $\mathbf{I}_1$  was previously JPEG compressed, starting from an uncompressed image  $\mathbf{I}_0$ , with a quality factor  $QF_1$  and with a grid shifted by  $(y, x) \neq (0, 0)$ ,  $0 \leq x \leq 7$  and  $0 \leq y \leq 7$ , with respect to the upper left corner, i.e.,

$$\mathbf{I}_1 = \mathbf{D}_{yx}^{-1} \mathcal{Q}_1(\mathbf{D}_{yx} \mathbf{I}_0) + \mathbf{E}_1. \quad (2)$$

Then the image  $\mathbf{I}_2$  is doubly compressed image, and we can express it as

$$\mathbf{I}_2 = \mathbf{D}_{yx}^{-1} \mathcal{Q}_1(\mathbf{D}_{yx} \mathbf{I}_0) + \mathbf{E}_1 + \mathbf{R}_2. \quad (3)$$

If a block DCT with grid alignment  $(i, j)$  is applied to  $\mathbf{I}_2$ , we can have three possible cases, according to the values assumed by this shift. If the grid is aligned to that of the last JPEG compressions, i.e.,  $i = 0$ ,  $j = 0$ , it happens that  $\mathbf{D}_{ij} \mathbf{I}_2 = \mathbf{D}_{00}(\mathbf{D}_{00}^{-1} \mathcal{Q}_2(\mathbf{D}_{00} \mathbf{I}_1) + \mathbf{E}_2) = \mathcal{Q}_2(\mathbf{D}_{00} \mathbf{I}_1) + \mathbf{D}_{00} \mathbf{E}_2$ . If the grid is aligned to that of the first JPEG compressions, i.e.,  $i = y$ ,  $j = x$ , we have  $\mathbf{D}_{ij} \mathbf{I}_2 = \mathbf{D}_{yx}(\mathbf{I}_1 + \mathbf{R}_2) =$

$\mathcal{Q}_1(\mathbf{D}_{yx} \mathbf{I}_0) + \mathbf{D}_{yx}(\mathbf{E}_1 + \mathbf{R}_2)$ . If the grid is misaligned with the two previous ones, we obtain  $\mathbf{D}_{ij} \mathbf{I}_2 = \mathbf{D}_{ij}(\mathbf{D}_{00}^{-1} \mathcal{Q}_2(\mathbf{D}_{00} \mathbf{I}_1) + \mathbf{E}_2)$ . In summary, the three previous cases can be collected together as

$$\mathbf{D}_{ij} \mathbf{I}_2 = \begin{cases} \mathcal{Q}_2(\mathbf{D}_{00} \mathbf{I}_1) + \mathbf{D}_{00} \mathbf{E}_2, & \text{if } i = 0, j = 0 \\ \mathcal{Q}_1(\mathbf{D}_{yx} \mathbf{I}_0) + \mathbf{D}_{yx}(\mathbf{E}_1 + \mathbf{R}_2), & \text{if } i = y, j = x \\ \mathbf{D}_{ij} \mathbf{D}_{00}^{-1} \mathcal{Q}_2(\mathbf{D}_{00} \mathbf{I}_1) + \mathbf{D}_{ij} \mathbf{E}_2, & \text{elsewhere.} \end{cases} \quad (4)$$

Since the codomains of the functions  $\mathcal{Q}_2(\cdot)$  and  $\mathcal{Q}_1(\cdot)$  are two lattices defined by the respective quantization tables, equation (4) shows that when the DCT grid is aligned with the grid of either the last compression or the first compression, the DCT coefficients tend to cluster around the points of such lattices, with a spread due to the presence of the error terms  $\mathbf{D}_{00} \mathbf{E}_2$  and  $\mathbf{D}_{yx}(\mathbf{E}_1 + \mathbf{R}_2)$ , respectively. Conversely, when the DCT grid is aligned with neither of the two compressions, DCT coefficients usually do not cluster around any lattice [11]. An example is given in Fig. 1, in which the histogram of doubly compressed DC coefficients is shown: for grid shifts  $(i, j) = (0, 0)$ ,  $(i, j) = (y, x)$  the DC coefficients tend to cluster around the points of two monodimensional lattices defined by the respective quantization steps, whereas for a random shift, representing the case of a single JPEG compressed image, such a periodic clustering cannot be observed.

In the case of alignment to the last compression, if we assume that rounding errors are uniformly distributed in  $[-0.5, 0.5]$  and  $\mathbf{D}_{00}$  is unitary, according to the Central Limit Theorem the error term  $\mathbf{D}_{00} \mathbf{E}_2$  is approximately Gaussian distributed with zero mean and variance  $1/12$ . In the case of alignment to the first compression, if we assume that rounding errors on DCT coefficients at a given frequency are uniformly distributed in  $[-Q_2/2, Q_2/2]$ , being  $Q_2$  the quantization step used by the second JPEG compression, and independent from  $\mathbf{E}_1$ , then the error term  $\mathbf{D}_{yx}(\mathbf{E}_1 + \mathbf{R}_2)$  at the same frequency is approximately Gaussian distributed with zero mean and variance  $(Q_2^2 + 1)/12$ . The clustering of the DCT coefficient in the case of NA-JPEG will be evident only if the standard deviation of this error term is small compared to the corresponding quantization step of the primary compression  $Q_1$ , meaning that the presence of NA-JPEG is usually difficult to detect when  $Q_1 < Q_2$ .

The main idea behind the proposed algorithm is that of detecting the presence of NA-JPEG double compression by measuring how DCT coefficients cluster around a given lattice for any possible grid shift. When NA-JPEG is detected, the parameters of the lattice also allow us to derive the primary quantization table. Even if the effect described above can be measured in theory for each DCT coefficient within an  $8 \times 8$  block, we observed that it is more evident in the case of the DC coefficient, when most of the analyzed DCT coefficients are different

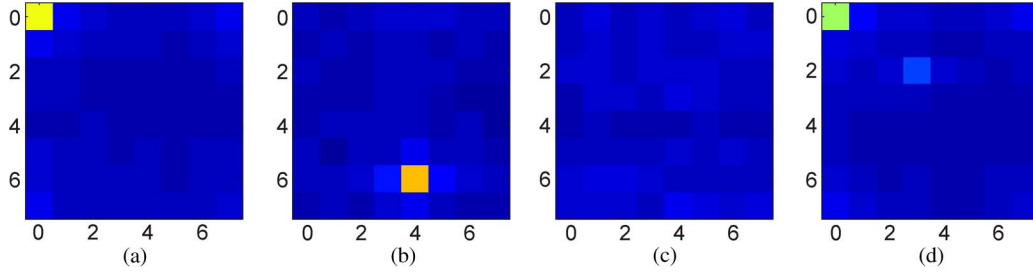


Fig. 2. Examples of IPMs for quantization steps (bright/dark points correspond to high/low values): (a)  $\mathbf{M}(Q)$  for  $Q = Q_2$ ; (b)  $\mathbf{M}(Q)$  for  $Q = Q_1 \neq Q_2$ , in presence of NA-JPEG with shift between first and second compression  $(y, x) = (6, 4)$ ;  $H_\infty = 2.56$ ; (c)  $\mathbf{M}(Q)$  for  $Q \neq Q_2$ , in absence of NA-JPEG;  $H_\infty = 5.23$ ; and (d)  $\mathbf{M}(Q)$  for  $Q = Q_1 = Q_2$ ,  $(y, x) = (2, 3)$ .

from zero. Hence, in order to keep the detection simple, in the following we will take into account only the DC coefficient of each block.

When a single DCT coefficient for each image block is considered, clustering around a lattice can be measured by analyzing the periodicity of the histogram computed on these coefficients for an integer period, as shown in Fig. 1. The periodicity of the histogram can be evaluated by considering its Fourier transform at frequencies which are reciprocal of an integer value, i.e., by evaluating the following values:

$$f_{ij}(Q) \triangleq \sum_k h_{ij}(k) e^{-j \frac{2\pi k}{Q}}, \quad Q \in \mathbb{N} \quad (5)$$

where  $h_{ij}$  is the histogram of DC coefficients computed for a grid shift equal to  $(i, j)$ , and  $Q$  is the quantization step with which the coefficients have been compressed.

According to (4), in the presence of NA-JPEG, where two compressions with a grid shift equal to  $(y, x)$  have been computed, we will expect both  $f_{00}(Q_2)$  and  $f_{y,x}(Q_1)$  to have higher magnitude than the other values,  $Q_2$  and  $Q_1$  being the quantization steps of the DC coefficient in the second and first JPEG compressions, respectively. In the absence of NA-JPEG, instead, only  $f_{00}(Q_2)$  will have higher magnitude; moreover, for each quantization value  $Q \neq Q_2$  we can assume that  $f_{ij}(Q)$  varies very little with  $(i, j)$ , since the overall histogram of the DCT coefficients remains quite similar for different shifts being it mainly dependent on the image content only.

In order to capture this behavior of the coefficients  $f_{ij}(Q)$  we resort to the integer periodicity map (IPM) at the quantization step  $Q$  defined as

$$M_{ij}(Q) \triangleq \frac{|f_{ij}(Q)|}{\sum_{i',j'} |f_{i'j'}(Q)|}, \quad 0 \leq i, i' \leq 7, \quad 0 \leq j, j' \leq 7. \quad (6)$$

The map  $\mathbf{M}(Q_2)$  will show a peak at the location  $(0, 0)$  due to the last compression. Moreover, in the case of an NA-JPEG image,  $\mathbf{M}(Q_1)$  will exhibit a single entry much greater than the others at the location  $(y, x)$  corresponding to the shift of the primary compression, whereas in the absence of NA-JPEG  $\mathbf{M}(Q)$  will be nearly uniform for every  $Q \neq Q_2$ . Examples of IPMs for the cases just described are shown in Fig. 2.

In the proposed algorithm, the uniformity of each IPM is measured by its min-entropy, defined as

$$H_\infty(Q) \triangleq \min_{ij} (-\log M_{ij}(Q)). \quad (7)$$

It is easy to verify that a high min-entropy corresponds to a mostly uniform IPM, whereas an IPM with a high peak will be characterized by a low min-entropy.

A JPEG image whose DC coefficients have been compressed with quantization step  $Q_2$  will then be classified as NA-JPEG if there exists a  $Q \neq Q_2$  such that  $H_\infty(Q) < T_1$ , where  $T_1$  is a suitable threshold, and the relative shift, computed as  $(y, x) = \arg \max_{(i,j)} M_{ij}(Q)$ , is different from  $(0, 0)$ . In practice, we test all  $Q$  values between  $Q_{\min} = 2$  and  $Q_{\max} = 16$ . When more than one  $Q$  satisfies the above condition, the  $Q$  achieving the lowest min-entropy is selected as the quantization step of the primary compression  $Q_1$ .

#### A. Coping With Case $Q_1 = Q_2$

The above strategy works well when  $Q_1$  is different from  $Q_2$ , that is when the two compressions were carried out with different quality factors. When  $Q_1 = Q_2$ ,  $\mathbf{M}(Q_1)$  already shows a high peak at  $(0, 0)$  due to the last compression, which makes the peak due to the first compression less evident, as shown in Fig. 2(d), so that the detection of the presence of the primary compression will fail.

To cope with this problem, we observed experimentally that in the absence of NA-JPEG,  $\mathbf{M}(Q_2)$  is approximately symmetric with respect to the shift  $(4, 4)$ ; this can be explained by observing that the distribution of the DC coefficients computed using different grid shifts will appear more close to that of a quantized signal when most of the pixels within the block comes from a single block of the original JPEG image, that is, when we have a shift like  $(0, 1)$ , or  $(7, 0)$ , whereas it will be less close otherwise. In summary, we can assume that when the proportion of pixels coming from different adjacent blocks is the same, as it appears for shifts symmetric with respect to  $(4, 4)$  (for example  $(1, 1)$  and  $(7, 7)$ ), the histogram of DC coefficients has similar resemblance to that of a quantized signal, so that similar IPM values can be expected.

Hence, we devise to detect the secondary peak by observing the asymmetry of the map that can be studied by defining a differential IPM (DIPM) as follows:

$$M'_{ij}(Q) \triangleq K^{-1} \cdot \max(M_{ij}(Q) - \mathcal{P}(M_{ij}(Q)), 0) \quad (8)$$

where  $\mathcal{P}(M_{ij}(Q))$  is the prediction of  $M_{ij}(Q)$  according to the symmetry of the IPM, and  $K$  is a constant such that the entries of the DIPM sum up to unity.  $\mathcal{P}(M_{ij}(Q))$  is computed as follows (we omit the quantization scale  $Q$  for clarity):

$$\mathcal{P}(M_{ij}) = \begin{cases} \frac{M_{i,8-j} + M_{8-i,j} + M_{8-i,8-j}}{3} & (i, j) \in \mathcal{R}_S \\ M_{i,8-j} & (i, j) \in \mathcal{R}_H \\ M_{8-i,j} & (i, j) \in \mathcal{R}_V \\ \frac{M_{0,3} + M_{0,5}}{2} & (i, j) = (0, 4) \\ \frac{M_{3,0} + M_{5,0}}{2} & (i, j) = (4, 0) \\ \frac{M_{4,3} + M_{4,5} + M_{3,4} + M_{5,4}}{4} & (i, j) = (4, 4) \\ 0 & (i, j) = (0, 0) \end{cases} \quad (9)$$

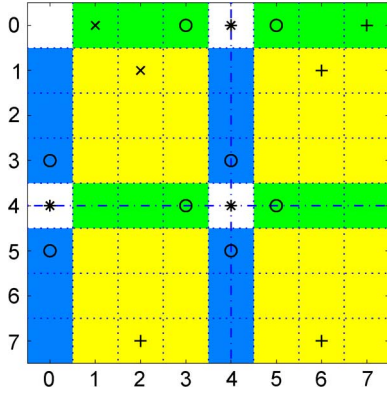


Fig. 3. Symmetric regions of IPM around the  $i = 4$  and  $j = 4$  axes. Yellow regions have both horizontal and vertical symmetry ( $\mathcal{R}_S$ ), green regions have horizontal symmetry ( $\mathcal{R}_H$ ), blue regions have vertical symmetry ( $\mathcal{R}_V$ ). White values have no symmetric counterparts. Values indicated by “x” are predicted according to their symmetric counterpart(s) indicated by “+”. Values indicated by “\*” are predicted according to the neighboring values indicated by “o”.

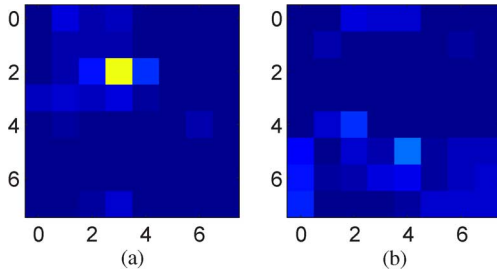


Fig. 4. Examples of DIPMs (bright/dark points correspond to high/low values): (a) DIPM of the map in Fig. 2(d),  $H'_\infty = 1.26$ . (b) DIPM of the map in Fig. 2(a),  $H'_\infty = 3.14$ .

where  $\mathcal{R}_S = \{(i, j) : i \neq 0, 4, j \neq 0, 4\}$  is the set of IPM values showing both horizontal and vertical symmetry around the  $i = 4$  and  $j = 4$  axes, whereas  $\mathcal{R}_H = \{(i, j) : i = 0, 4, j \neq 0, 4\}$  and  $\mathcal{R}_V = \{(i, j) : i \neq 0, 4, j = 0, 4\}$  are the sets of IPM values showing only horizontal or vertical symmetry around the  $j = 4$  or  $i = 4$  axes, respectively. The above regions are shown for clarity in Fig. 3. The rationale is that IPM values belonging to symmetric regions are predicted according to their symmetric counterparts, whereas IPM values having no symmetry are predicted according to their neighbors.

In the presence of NA-JPEG, the DIPM will show a peak in correspondence with the shift of the primary compression (see Fig. 4). Again, we will define the min-entropy of the DIPM as

$$H'_\infty(Q) \triangleq \min_{ij} (-\log M'_{ij}(Q)). \quad (10)$$

### B. Detection Algorithm

Hence, the detection algorithm proceeds as follows: first, it looks for a  $Q \neq Q_2$  such that  $H_\infty(Q) < T_1$ ; if there is one value satisfying this condition, the image will be classified as NA-JPEG; if there is no such a value, it checks if  $H'_\infty(Q_2) < T_2$ : if true, then the image will be classified as NA-JPEG, otherwise as non-NA-JPEG.<sup>1</sup>

<sup>1</sup>Here, by “non-NA-JPEG” we mean either a singly compressed image or a doubly compressed image with aligned grid: concerning the proposed algorithm, the two cases are indistinguishable.

A pseudo-code of the complete detection algorithms, summarizing the above steps, is shown in Algorithm 1.

---

**Algorithm 1** Pseudocode of the proposed algorithm for detecting NA-DJPG compression.

---

```

input  $I_2$ 
for  $i, j = 0 \rightarrow 7$  do
    compute  $D_{ij}I_2$ 
    compute  $h_{ij}$  as histogram of DC coefficients
    for  $Q = Q_{\min} \rightarrow Q_{\max}$  do
        compute  $f_{ij}(Q)$  as in (5)
    end for
end for
for  $Q = Q_{\min} \rightarrow Q_{\max}$  do
    for  $i, j = 0 \rightarrow 7$  do
        compute  $M_{ij}(Q)$  as in (6)
        if  $Q = Q_2$  then
            compute  $M'_{ij}(Q_2)$  as in (8)
        end if
    end for
    compute  $H_\infty(Q)$ 
    if  $Q = Q_2$  then
        compute  $H'_\infty(Q_2)$ 
    end if
end for
select  $H_\infty = \min_Q H_\infty(Q)$ ,  $Q_1 = \arg \min_Q H_\infty(Q)$ ,  $(y, x) = \arg \max_{(i,j)} M_{ij}(Q_1)$  such that  $(y, x) \neq (0, 0)$ 
if  $H_\infty < T_1$  then
    return NA-DJPG,  $Q_1, (y, x)$ 
else if  $H'_\infty(Q_2) < T_2$  then
    return NA-DJPG,  $Q_2, (y, x) = \arg \max_{(i,j)} M'_{ij}(Q_2)$ 
else
    return non-NA-DJPG
end if

```

---

## IV. EXPERIMENTAL RESULTS

For the experimental validation of the proposed work, we built an image dataset composed by 1000 noncompressed TIFF images, having heterogeneous contents, coming from three different digital cameras (namely Nikon D90, Canon EOS 450D, Canon EOS 5D) and each acquired at its highest resolution; each test was performed by cropping the central portion using four different image sizes ( $128 \times 128$ ,  $256 \times 256$ ,  $512 \times 512$ , and  $1024 \times 1024$ ).

For simulating NA-JPEG, each original image was JPEG compressed with a quality factor  $Q_{F1}$ , decompressed, cropped by a random shift  $(i, j) \neq (0, 0)$ , with  $0 \leq i \leq 7, 0 \leq j \leq 7$ , and

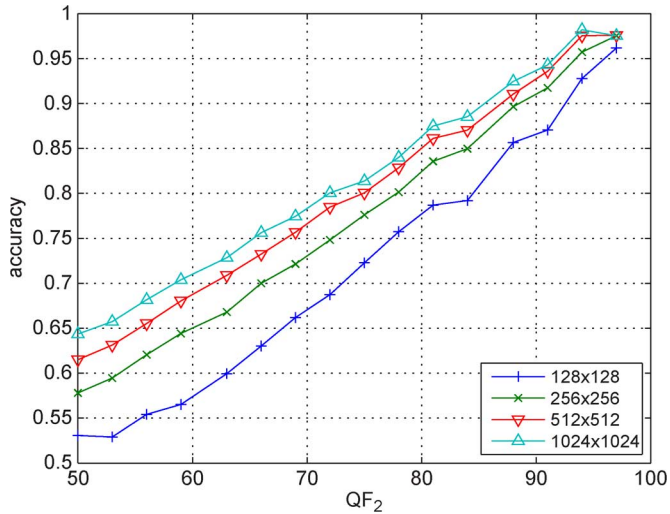


Fig. 5. Accuracy of proposed detector for different JPEG qualities  $QF_2$  and different image sizes.

JPEG compressed with another quality factor  $QF_2$ . The absence of NA-JPEG was simulated by simply compressing the original image with a quality factor  $QF_2$ . The quality factor of the first compression ( $QF_1$ ) was chosen so that the quantization step of the DC coefficients ( $Q_1$ ) ranges from 2 to 16 with step 1, whereas the quality factor of the second compression ( $QF_2$ ) was chosen so that the quantization step of the DC coefficients ( $Q_2$ ) ranges from 1 to 16 with step 1. The case  $Q_1 = 1$  was avoided because it is undetectable with the proposed method.

This resulted in 240 possible combinations of ( $QF_1, QF_2$ ) for each tampered image and 16 different  $QF_2$  for each original image, yielding a total of 240 000 tampered images and 16 000 original images for each of the four image sizes.

The performance of the proposed detector has been investigated by estimating the ROC curves for different combinations of ( $QF_1, QF_2$ ), using a 5-fold cross validation strategy: the distribution of the statistics  $H_\infty(Q)$ ,  $H'_\infty(Q_2)$  is recorded over a training set of 800 images, optimal thresholds  $T_1, T_2$  are set according to such statistics, and the performance is measured on the remaining 200 images. The above procedure is repeated five times, using mutually disjoint sets of images for testing, and the average performance measures are recorded. Different optimal thresholds are chosen according to  $QF_2$  and the image size.

In Fig. 5, we show the maximum accuracy of the detector for different values of  $QF_2$  and different image sizes. The maximum accuracy is defined as the point on the ROC curve corresponding to the maximum number of correctly classified images and is averaged over all possible  $QF_1$  values.

In order to make comparisons with other methods, we computed on the same database the features described in [6] and [9] and we fed them to a support vector machine (SVM) using a radial basis function kernel [12]. A different SVM was trained for each  $QF_2$  and image size, considering every possible  $QF_1$ . Optimal kernel parameters are found via grid search and 3-fold cross validation, whereas the accuracy is evaluated through 5-fold cross validation. To avoid the effects of imbalance [13]—for each  $QF_2$  we have 15 000 tampered images and 1000 original images—we used an ensemble of undersampled SVMs [14] and we measured the overall accuracy as the arithmetic mean of the accuracy on each class.

The accuracy of the features of [6] and [9] is shown in Figs. 6 and 7, respectively. Compared to [6], our detector is from 5% to 15% more accurate for similar image sizes. Noticeably, the higher improvement

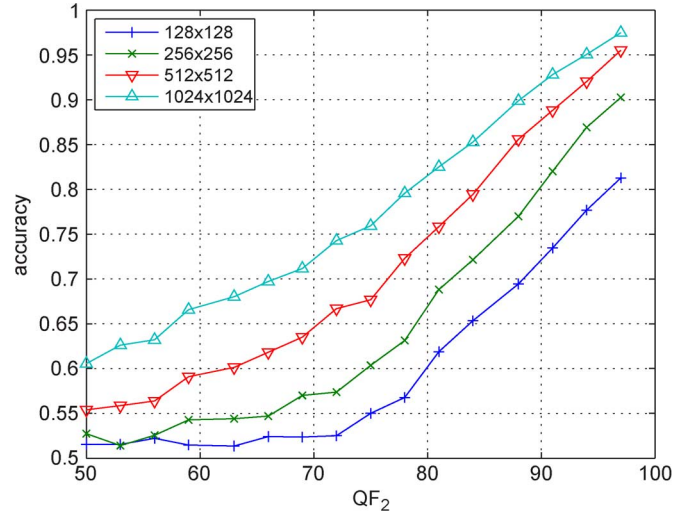


Fig. 6. Accuracy of detector in [6] for different JPEG qualities  $QF_2$  and different image sizes.

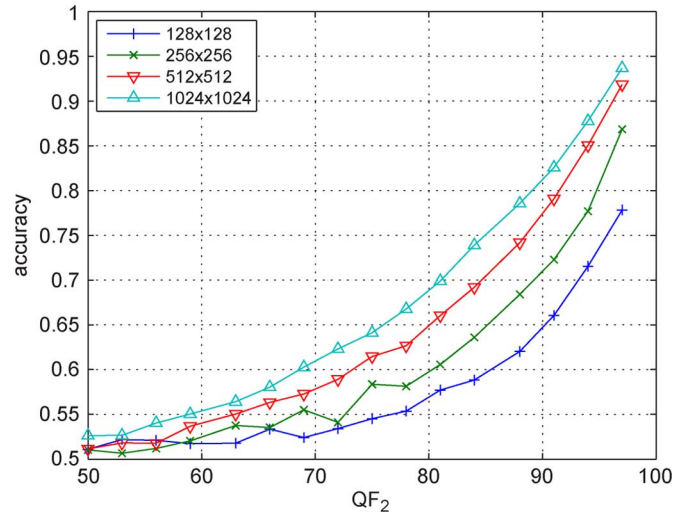


Fig. 7. Accuracy of detector in [9] for different JPEG qualities  $QF_2$  and different image sizes.

TABLE I  
ACCURACY OF PROPOSED DETECTOR (%) FOR IMAGE SIZE  $1024 \times 1024$

$QF_1 \backslash QF_2$	50-57	58-67	68-76	77-85	86-95	96
50-57	<b>88.5</b>	<b>92.6</b>	<b>93.7</b>	94.7	95.7	95.3
58-67	<b>80.7</b>	<b>90.7</b>	<b>94.4</b>	<b>96.3</b>	<b>97.2</b>	96.5
68-76	60.8	<b>77.7</b>	<b>92.5</b>	<b>96.5</b>	<b>98.0</b>	98.2
77-85	50.2	53.6	<b>67.6</b>	<b>91.0</b>	<b>98.6</b>	<b>98.8</b>
86-95	50.2	50.1	49.8	54.8	<b>85.4</b>	<b>98.9</b>

in performance is achieved for the smaller image sizes: our detector needs only a  $256 \times 256$  image to achieve the best performance of [6]. Compared to [9], our detector is from 10% to 25% more accurate for similar image sizes and similar  $QF_2$ .

In Table I, we show the maximum accuracy of the proposed detector for different combinations of  $QF_1$  and  $QF_2$ , when the image size is  $1024 \times 1024$ . For a comparison, the accuracy of the detectors in [6] and [9] is shown in Tables II and III, respectively. To make the evaluation of the methods easier, the best results for each combination ( $QF_1, QF_2$ ) are highlighted in bold. Even though the case of  $1024 \times 1024$  images is

TABLE II  
ACCURACY OF DETECTOR IN [6] (%) FOR IMAGE SIZE 1024 × 1024

$QF_1 \backslash QF_2$	50-57	58-67	68-76	77-85	86-95	96
50-57	73.6	82.6	89.5	<b>95.9</b>	<b>97.1</b>	<b>98.6</b>
58-67	67.2	77.3	85.4	94.8	97.2	<b>98.4</b>
68-76	<b>61.0</b>	67.8	76.8	89.9	97.3	<b>98.2</b>
77-85	<b>57.4</b>	<b>60.0</b>	63.9	75.3	94.8	97.6
86-95	<b>51.4</b>	<b>52.8</b>	<b>53.4</b>	<b>56.4</b>	76.5	94.5

TABLE III  
ACCURACY OF DETECTOR IN [9] (%) FOR IMAGE SIZE 1024 × 1024

$QF_1 \backslash QF_2$	50-57	58-67	68-76	77-85	86-95	96
50-57	56.3	62.4	73.7	82.4	91.5	96.3
58-67	55.1	60.0	67.6	79.3	90.6	96.4
68-76	53.2	56.7	61.9	72.7	88.6	95.7
77-85	51.4	53.5	57.1	63.1	80.7	93.5
86-95	49.5	49.8	50.8	53.3	63.5	86.5

TABLE IV  
PROBABILITY OF DETECTION OF PROPOSED DETECTOR (%) FOR A PROBABILITY OF FALSE ALARM EQUAL TO 1%, FOR IMAGE SIZE 1024 × 1024

$QF_1 \backslash QF_2$	50-57	58-67	68-76	77-85	86-95	96
50-57	78.1	86.9	88.6	90.4	92.4	92.1
58-67	62.3	82.6	90.5	93.8	95.5	94.1
68-76	20.3	55.9	86.0	94.4	97.1	97.5
77-85	1.5	7.9	34.7	82.3	98.1	98.6
86-95	1.3	1.3	0.8	9.9	71.7	98.6

that in which we have the smallest performance improvement, the proposed approach outperforms both previous methods, especially when  $QF_2$  is similar to  $QF_1$ : when  $QF_2 - QF_1 > 10$  NA-JPEG is detected with very high accuracy (>95% in most cases), while it is still detected with about 90% accuracy when  $QF_2$  is similar to  $QF_1$  and with about 80% accuracy when  $QF_1 - QF_2 < 10$  and  $QF_1 < 76$ . Noticeably, the proposed method maintains a very high probability of detection even if we force a 1% probability of false alarm, as shown in Table IV, confirming the robustness of the min-entropy feature.

In order to evaluate the ability of the algorithm to estimate the correct quantization step  $Q_1$  and the correct shift  $(y, x)$ , we also performed a test by setting the thresholds so as to achieve a probability of false alarm equal to 1% and we recorded the estimated parameters. The accuracy, defined as the percentage of correctly identified  $Q_1$  and  $(y, x)$  over the images detected as NA-JPEG, is shown in Fig. 8, for different values of  $QF_2$  and different sizes. For  $QF_2 > 75$ , the proposed method identifies the correct  $Q_1$  and  $(y, x)$  in over 98% of the images recognized as NA-JPEG irrespective of the image size, while for 1024 × 1024 images the correct identification rate is always greater than 96% irrespective of  $QF_2$ .

V. CONCLUSION

In this paper, a simple and reliable algorithm to detect into a digital image the presence of nonaligned double JPEG compression has been proposed. The method is based on the observation that the DCT coefficients exhibit an integer periodicity when the blockwise DCT is computed according to the grid of the primary JPEG compression. Such a behavior can be efficiently detected by measuring the nonuniformity of a suitably defined integer periodicity map (IPM), in which every entry of the map depends on the DCT statistics for a particular grid

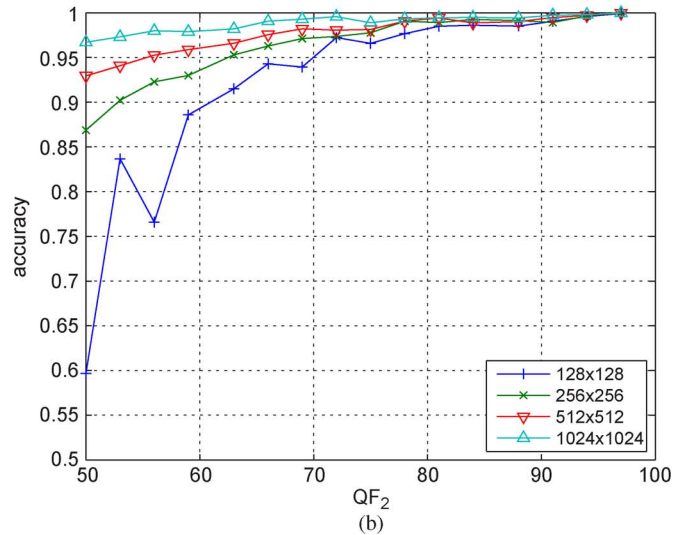
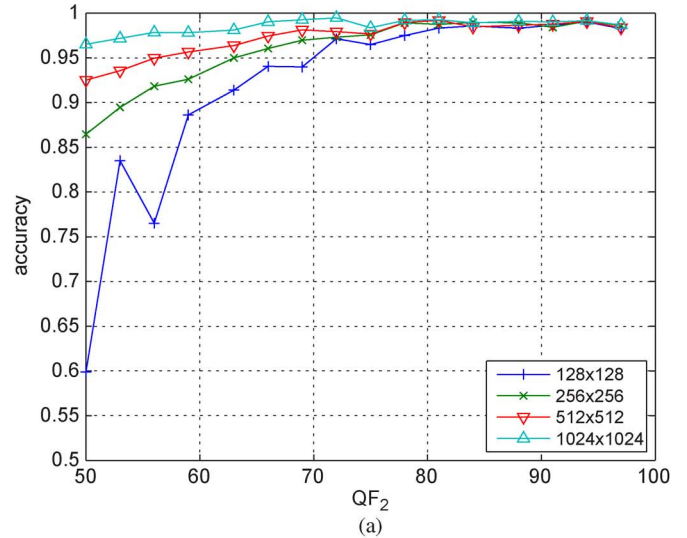


Fig. 8. Percentage of correctly estimated primary compression parameters: (a) quantization step  $Q$  and (b) shift  $(y, x)$ .

shift. A slightly modified map is required when the second compression uses the same quantization step as the primary one. The presence of NA-JPEG is detected by applying a threshold detector to the min-entropy of the IPM, measuring its uniformity. Experimental results show that the proposed detector achieves a higher detection accuracy than previously proposed methods and is able to analyze smaller images. Moreover, the proposed method is able to accurately estimate both the grid shift and the quantization step of the DC coefficient of the primary JPEG compression, which can be used to perform more advanced analyses. Indeed, we are currently studying the possibility of using the estimated parameters to derive a statistical model of DCT coefficients of NA-JPEG images, which can be used for the automatic localization of tampered regions following an approach similar to the one proposed in [3].

REFERENCES

[1] H. Farid, "A survey of image forgery detection," *IEEE Signal Process. Mag.*, vol. 2, no. 26, pp. 16–25, 2009.  
 [2] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Trans. Inform. Forensics Security*, vol. 4, no. 1, pp. 154–160, Mar. 2009.

- [3] 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*, vol. 42, no. 11, pp. 2492–2501, Nov. 2009.
- [4] D. Fu, Y. Q. Shi, and W. Su, "A generalized Benford's law for JPEG coefficients and its applications in image forensics," in *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX*, P. W. Wong and E. J. Delp, Eds., San Jose, CA, Jan. 2007, vol. 6505, pp. 1L1–1L11.
- [5] B. Li, Y. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *Proc. IEEE 10th Workshop Multimedia Signal Processing*, Oct. 2008, pp. 730–735.
- [6] W. Luo, Z. Qu, J. Huang, and G. Qui, "A novel method for detecting cropped and recompressed image block," in *Proc. ICASSP 2007, 2007*, vol. 2, pp. II-217–II-220.
- [7] Y.-L. Chen and C.-T. Hsu, "Image tampering detection by blocking periodicity analysis in JPEG compressed images," in *Proc. IEEE 10th Workshop Multimedia Signal Processing*, Oct. 2008, pp. 803–808.
- [8] Z. Qu, W. Luo, and J. Huang, "A convolutive mixing model for shifted double JPEG compression with application to passive image authentication," in *Proc. ICASSP*, 2008, pp. 1661–1664.
- [9] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Trans. Inform. Forensics Security*, vol. 6, no. 2, pp. 396–406, Jun. 2011.
- [10] M. Barni, A. Costanzo, and L. Sabatini, "Identification of cut & paste tampering by means of double-JPEG detection and image segmentation," in *Proc. ISCAS*, 2010, pp. 1687–1690.
- [11] J. Lukáš and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Digital Forensic Res. Workshop*, 2003.
- [12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [13] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowledge Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [14] P. Kang and S. Cho, "EUS SVMs: Ensemble of under-sampled SVMs for data imbalance problems," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, I. King, J. Wang, L. Chan, and D. Wang, Eds. Berlin, Germany: Springer, 2006, vol. 4232, pp. 837–846.