Reverse Engineering of Double JPEG Compression in the Presence of Image Resizing

Tiziano Bianchi [#1], Alessandro Piva [#2]

Department of Electronics and Telecommunications, University of Florence
via S. Marta 3, 50139, Firenze, Italy
[1] `tiziano.bianchi@unifi.it`, [3] `alessandro.piva@unifi.it`

Abstract—In this paper, we propose a forensic technique for the reverse engineering of double JPEG compression in the presence of image resizing between the two compressions. Our approach is based on the fact that previously JPEG compressed images tend to have a near lattice distribution property (NLDP), and that this property is usually maintained after a simple image processing step and subsequent recompression. The proposed approach represents an improvement with respect to existing techniques analyzing double JPEG compression. Moreover, compared to forensic techniques aiming at the detection of resampling in JPEG images, the proposed approach moves a step further, since it also provides an estimation of both the resize factor and the quality factor of the previous JPEG compression. Such additional information can be used to reconstruct the history of an image and perform more detailed forensic analyses.

I. INTRODUCTION

In recent years, the possibility of acquiring a large amount of multimedia data in digital format and sharing it through the Internet has brought a dramatic change in the way such information is used. With a very little effort, multimedia data such as images can be copied, manipulated, or combined together, making it extremely difficult to maintain a link between the original acquisition and the final digital copy accessed by the users. This fact has stimulated the development of a new discipline, image forensics, aiming at detecting clues regarding the history of a digital image, by looking for distinctive patterns in statistical and geometrical features, including JPEG quantization artifacts, interpolation, demosaicing traces [1].

Since a vast amount of digital images is available today in JPEG format, several forensic tools try to obtain clues by analyzing artifacts introduced by JPEG recompression. When the discrete cosine transform (DCT) grids of successive JPEG compressions are perfectly aligned, areas which have undergone a double JPEG compression can be detected by recompressing the image at different quality levels [2], or by analyzing the statistics of DCT coefficients [3]. Recent results demonstrate that even multiple JPEG recompressions can be detected using first digit features of DCT coefficients [4]. Alternatively, non aligned double JPEG compression can be revealed by considering blocking artifacts [5], or by evaluating

the integer periodicity of DCT coefficients when different shifts are applied to the examined image [6]. In both cases, a careful examination of DCT coefficient statistics can also permit automatic localization of doubly compressed areas [7].

A classical application scenario for the aforementioned tools is image tampering detection. However, a much more ambitious goal could be that of moving a step further, in order to collect information about the processing chain which led to a specific observed image. In this respect, a first example can be provided by forensic tools that not only detect double JPEG compression, but also estimate previous compression parameters, like [2] or [6]. Unfortunately, the above tools suffer from severe limitations in real life scenarios. For example, if the image is resized between successive JPEG compressions, which is often the case when digital images are posted on photo sharing applications, the models the above methods rely on are no more valid.

In this paper, we propose a forensic technique for the reverse engineering of double JPEG compression in the presence of image resizing between the two compressions. Our approach aims at exploiting the fact that previously JPEG compressed images tend to be distributed near the points of a lattice and is based on the extension of the technique proposed in [6] for non aligned double JPEG compression. The proposed approach represents an improvement with respect to existing techniques analyzing double JPEG compression. Moreover, compared to forensic techniques aiming at the detection of resampling in JPEG images like [8], the proposed technique moves a step further, since it also provides an estimation of both the resize factor and the compression parameters of the previous JPEG compression. Such additional information is important, since it can be used to reconstruct the history of an image and perform a more detailed forensic analysis.

II. PROBLEM STATEMENT

Let us then assume that an original uncompressed image $\mathbf{x}$ is JPEG compressed with a quality factor QF_1, and then decompressed. The image obtained after JPEG decompression can be modeled as follows:

$$\mathbf{x}_1 = \mathbf{D}_{00}^{-1}\mathcal{D}(\mathcal{Q}(\mathbf{D}_{00}\mathbf{x})) + \mathbf{e}_1. \tag{1}$$

In the above equation, $\mathbf{D}_{00}$ models an 8×8 block DCT with the grid aligned with the upper left corner of the image, $\mathcal{Q}(\cdot)$ and $\mathcal{D}(\cdot)$ model quantization and dequantization processes, respectively, and $\mathbf{e}_1$ is the error introduced by rounding and truncating the output values to eight bit integers.

In the above chain, each 8×8 block of the decompressed image can be modeled as a point in a 64-dimension lattice, generated by the composition of the JPEG quantization matrix and the block DCT matrix, perturbed by a noise due to rounding/truncation (R/T) errors. The distribution of such a signal can be modeled as a mixture given by several components distributed as the R/T errors, each one having mean equal to a lattice point. Such a behavior can be considered as a very powerful feature for detecting JPEG compression. If we consider a common high SNR model for signal quantization, the probability distribution of a generic signal within each lattice cell can be assumed uniform. Hence, the probability of a generic signal to fall within the support of the near lattice distribution (NLD) can be computed as the ratio of the volume of the support of R/T errors over 64 pixels to the volume of a lattice cell, i.e.,

$$p = \left(\frac{q^{64}}{\mathcal{V}_\Lambda}\right)^N = \left(\prod_{k=1}^{64}\frac{1}{Q_1(k)}\right)^N \tag{2}$$

where q is the quantization step for the pixel values, $\mathcal{V}_\Lambda$ is the volume of the lattice cell, and N is the number of 8×8 blocks of the image. In the above equation, we have assumed that pixels are rounded to the nearest integer ($q = 1$) and that the volume of the lattice cell is given by the product of the JPEG quantization matrix entries $Q_1(k)$, $k = 1, \ldots, 64$, since the DCT matrix is unitary. It is clear that for $Q_1(k) > 1$ and for moderately large images ($N \approx 10^4$) the above probability quickly becomes negligible.

The near lattice distribution property (NLDP) of JPEG images is expected to hold even if the decompressed image undergoes subsequent processing steps. Let us consider the case we are interested in, i.e., image resizing followed by JPEG recompression.

When the decompressed image is resized, the whole process can be modeled as the convolution of the image pixels by an interpolation kernel $\phi(\mathbf{s})$, followed by resampling on the desired grid. More formally, the resized image is given by

$$\mathbf{x}_2(\mathbf{m}) = \sum_{\mathbf{n}} \mathbf{x}_1(\mathbf{n})\phi(\mathbf{A}\mathbf{m} + \mathbf{b} - \mathbf{n}) \tag{3}$$

where $\mathbf{n} = (n_1, n_2)$ and $\mathbf{m} = (m_1, m_2)$, $n_1, n_2, m_1, m_2 \in \mathbb{N}$, are the pixel coordinates of the original and resized image, respectively, whereas $\mathbf{A}$ and $\mathbf{b}$ are a 2×2 matrix and a 2×1 shift vector defining the resampling grid. For simple image resizing, we have $\mathbf{A} = \gamma^{-1}\mathbf{I}$, where γ is the resize factor and $\mathbf{I}$ is the identity matrix. Moreover, in the following we will assume that $\mathbf{b} = \mathbf{0}$, i.e., image resizing preserves the position of the upper left pixel. It is worth noting that the above formalism can be used to model other generic affine transformations of the image plane, like rotation and shearing.

Observing the NLDP directly on the resampled pixels is not an easy task. However, if we are able to estimate an approximation of the intermediate continuous image surface $\xi(\mathbf{s}) = \sum_{\mathbf{n}} \mathbf{x}_1(\mathbf{n})\phi(\mathbf{s} - \mathbf{n})$, it is evident that by resampling it according to the original pixel grid we will again observe the NLDP of the resulting pixels. In order to quantify to what extent the NLDP is preserved after image resizing, it is important to assess how well the continuous image $\xi(\mathbf{s})$ can be approximated starting from the observed resampled pixels. When the interpolation kernel is an ideal sinc pulse, $\xi(\mathbf{s})$ can be exactly reconstructed if $\gamma > 1$, or when $\gamma < 1$ and the spectrum of the original signal is zero outside $[-\gamma\pi, \gamma\pi]$. For practical interpolation kernels, like the bilinear or the bicubic spline, there will be in general some residual aliasing terms that can be modeled as signal dependent noise. However, for natural images such aliasing terms will usually be negligible and we do not expect them to significantly affect the NLDP.

In the case of JPEG recompression, the second JPEG compression will introduce an approximation term $\mathbf{e}_3$ with respect to the decompressed and resized image $\mathbf{x}_2$. Such an error term will add up to the R/T errors and will make the detection of the NLDP more difficult. The exact distribution of $\mathbf{e}_3$ may be complex to derive, however by assuming again the high SNR model for the quantization of DCT coefficient and invoking the central limit theorem (CLT) it can be approximated by a Gaussian distribution having zero mean and variance

$$\sigma_e^2(\mathbf{n}) = \sum_{k=1}^{64} \tilde{d}_k(\mathbf{n})^2 Q_2(k)^2 \tag{4}$$

where $\tilde{d}_k(\mathbf{n})$ is the kth frequency 8×8 IDCT vector evaluated at pixel position $\mathbf{n}$ and $Q_2(k)$ models the quantization matrix of the second JPEG compression. Hence, the probability of a generic signal to fall within the support of the NLD can be approximated as the ratio of the volume of the typical set of $\mathbf{e}_3$ over an 8×8 block to the volume of a lattice cell, i.e.,

$$p \approx \left(\frac{2^{h(\mathbf{e}_3)}}{\mathcal{V}_\Lambda}\right)^N = \left[\frac{\prod_{\mathbf{n}\in\mathcal{B}_8}(\sigma_e(\mathbf{n})\sqrt{2\pi e})}{\prod_{k=1}^{64} Q(k)}\right]^N \tag{5}$$

where $h(\mathbf{e}_3) = \frac{1}{2}\sum_{\mathbf{n}\in\mathcal{B}_8}\log[2\pi e\sigma_e(\mathbf{n})^2]$ is the differential entropy of $\mathbf{e}_3$ over the generic 8×8 block $\mathcal{B}_8$. As long as $2^{h(\mathbf{e}_3)}/\mathcal{V}_\Lambda < 1$, the above equation says that the support of a signal exhibiting NLDP and that of a generic signal are disjointed with a probability tending to one.

A. Practical Measures for NLDP

Although the NLDP is a very discriminative feature, detecting it in practice can be a difficult task. Firstly, the lattice defined by the first JPEG compression depends on hidden parameters. In some cases, these parameters can be estimated by exploiting the fact that they can assume only a limited number of possible values. For example, in the case of non aligned double JPEG compression there are only 64 possible grid shifts between the first and second compressions [7]. Nevertheless, estimating the quantization matrix of the first JPEG compression may still be a problem. Moreover, when

the image has been resized after the first compression, the parameter γ may have a very large set of possible values (even a continuous range), in which case the above approach may become unpractical.

Secondly, the high SNR assumption may no longer hold when quantizing DCT coefficients at higher frequencies. In practice, this can be solved by checking only a sublattice corresponding to the DCT coefficients at lower frequencies. Even if this approach reduces the discriminative power of the NLDP, it also has the merit of reducing the number of hidden parameters to be estimated.

A suboptimal yet computationally practical approach to detect the NLDP in the case of non aligned double JPEG without resizing is that proposed in [6]. The aforementioned method analyzes the integer periodicity of the histogram of the DC coefficients when the block DCT is applied with different grid shifts. The periodicity of the histogram can be evaluated by considering its Fourier transform at frequencies which are reciprocal of an integer value, i.e., we define

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

where h_{ij} is the histogram of the DC coefficient for the (i, j) shift. When the NLDP is present, we expect to find a strong periodic component for a particular integer period and a particular grid shift. This behavior can be captured by defining an integer periodicity map (IPM) at the quantization step Q

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

In the presence of NLDP, $\mathbf{M}(Q_1)$ will have a single entry much greater than the others at the location (r, c) corresponding to the shift of the primary compression, whereas in the absence of NLDP $\mathbf{M}(Q)$ will be nearly uniform. In [6], this property is measured by computing the min-entropy of the IPMs, defined as

$$ H_\infty \triangleq \min_Q \left[\min_{ij} (-\log M_{ij}(Q)) \right]. \quad (6) $$

A high min-entropy corresponds to a mostly uniform IPM, whereas a IPM with a high peak will be characterized by a low min-entropy. If for a particular Q we have a low min-entropy value, this can be associated to the presence of the NLDP. Moreover, the Q achieving the lowest min-entropy value provides an estimate of the quality factor of the previous compression.

III. REVERSE ENGINEERING OF DOUBLE JPEG COMPRESSION

As discussed in the previous section, the NLDP can be a very reliable feature in order to detect the presence of a previous JPEG compression in the processing chain of the observed image. In the following, we will propose a simple algorithm exploiting the NLDP for the reverse engineering of a processing chain composed by a first JPEG compression, an image resizing, and a final JPEG compression.

The proposed algorithm can be summarized by the following steps:
1) estimate a number of candidates for the resizing factor;
2) for each candidate, reverse the resizing step;
3) for each counter-resized image, compute a measure of the NLDP;
4) if the measure over one of the counter-resized images is greater than a given threshold, label the image as doubly compressed with resizing factor equal to that yielding the maximum value of the measure, otherwise label the image as singly compressed.

The rationale of the above algorithm is that, in the presence of NLDP, the measure of NLDP obtained for the correct resizing factor will be much higher than the other ones. Conversely, in the absence of NLDP the probability of finding a particular γ for which the measure is higher than the threshold is negligible, since the probability of verifying the NLDP for a generic signal is very low.

The implementation of the proposed algorithm depends on the solution of three specific sub-problems. In the following sections, we will discuss how we propose to solve the above tasks.

A. Finding the Candidate Resizing Factors

This is probably the most critical step, since estimating the correct γ is fundamental for the observability of the NLDP. In this paper, we propose two alternative approaches based on different assumptions.

The first approach assumes that we have no prior knowledge about the correct resizing factor. This is the most general case, and also the most challenging one. We propose to estimate the candidate resizing factors by using the following algorithm, inspired by the methods in [9], [8]:
1) Compute approximate horizontal and vertical second derivatives by applying the filter $\mathbf{g} = [\ -\frac{1}{2} \quad 1 \quad -\frac{1}{2} \ ]$ along rows and columns

$$ \mathbf{x}_h''(n_1, n_2) = \sum_m \mathbf{g}(m) \mathbf{x}(n_1, n_2 - m) $$
$$ \mathbf{x}_v''(n_1, n_2) = \sum_m \mathbf{g}(m) \mathbf{x}(n_1 - m, n_2); $$

2) Average the magnitudes of the rows of horizontal derivatives and of the columns of vertical derivatives, yielding two one-dimensional vectors

$$ \mathbf{a}_h(n_2) = \sum_{n_1} \left| \mathbf{x}_h''(n_1, n_2) \right| $$
$$ \mathbf{a}_v(n_1) = \sum_{n_2} \left| \mathbf{x}_h''(n_1, n_2) \right|; $$

3) Compute the magnitude of the discrete Fourier transform (DFT) of the above vectors $\mathbf{f}_{h,v}(k) = |\text{DFT}\{\mathbf{a}_{h,v}(n)\}|$, divide them by their median filtered versions in order to obtain two equalized magnitude vectors, i.e. $\tilde{\mathbf{f}}_{h,v}(k) = \mathbf{f}_{h,v}(k) / \mathbf{f}_{h,v,\text{median}}(k)$, and add them together

$$ \tilde{\mathbf{f}}(k) = \tilde{\mathbf{f}}_h(k) + \tilde{\mathbf{f}}_v(k); $$

4) Estimate from $\tilde{\mathbf{f}}(k)$ the positions $\tilde{t}_i$ of the peaks having higher magnitude. For the above step, we consider only peaks whose value is greater than the median value times a constant, i.e,

$$\tilde{t}_i = \left\{ k \;\middle|\; \tilde{\mathbf{f}}(k) > \tilde{\mathbf{f}}(k-1) \;\wedge\; \tilde{\mathbf{f}}(k) > \tilde{\mathbf{f}}(k+1) \right.$$
$$\left. \wedge\; \tilde{\mathbf{f}}(k) > \beta \cdot \mathrm{median}\big[\tilde{\mathbf{f}}\big] \right\}.$$

The accuracy of the estimated positions is increased by fitting a periodic sinc function to the peak value and the immediately previous and following values, using the least squares method, yielding refined position estimates

$$t_i = \psi\left[\tilde{\mathbf{f}}(\tilde{t}_i - 1), \tilde{\mathbf{f}}(\tilde{t}_i), \tilde{\mathbf{f}}(\tilde{t}_i + 1)\right];$$

5) Estimate the candidate resizing factors as

$$\gamma_i = \frac{N_{\mathrm{DFT}}}{8 t_i} \tag{7}$$

where N_{DFT} is the length of the DFT.

The rationale of the above algorithm is that JPEG blocking artifacts left by the previous compression, after resizing by a factor γ, will result in periodic horizontal and vertical patterns, having period equal to 8γ.

The second approach assumes to have some prior knowledge about the possible resizing factors. This may be justified in several scenarios. For example, most photo sharing applications use a limited number of possible image sizes when resizing the original uploaded pictures. Since also commercial cameras are based on image sensors providing a set of standard image sizes, the respective image sizes can be paired in order to enumerate a set of possible resizing factors. Hence, we can output a set of candidate resizing factors by choosing all the image size pairs which result in the image size of the observed image.

B. Reversing the Resizing Step

In order to reverse the resizing step, we propose to simply apply an image resizing with factor γ^{-1}, using a standard image interpolation algorithm. As discussed in Section II, even if this will in general introduce aliasing on the counter-resized image, we assume that the effects are negligible for what concerns the detectability of the NLDP.

C. Measuring the NLDP

As a measure of the NLDP, we propose to compute the min-entropy H_∞ of the IPM according to (6). Since images exhibiting the NLDP yield lower min-entropy values, we choose to employ $-H_\infty$ in order to be consistent with the definition of measure given in the algorithm of Section III. The IPM approach also allows us to obtain an estimate of the quantization step of the first JPEG compression, given as $Q_1 = \arg\min_Q[\min_{ij}(-\log M_{ij}(Q))]$, which can be used to infer the quality factor of the first compression.

IV. EXPERIMENTAL RESULTS

The proposed algorithm has been tested on a dataset of 500 uncompressed images having heterogeneous contents, coming from three different digital cameras (namely Nikon D90, Canon EOS 450D, Canon EOS 5D). From each image, we extracted a portion of size 1024×1024 pixels and we compressed it with six different JPEG quality factors, namely 50, 60, 70, 80, 90, and 99, using the Matlab function `imwrite`. The resulting 3000 images represent the dataset of singly compressed images. Then, each of the singly compressed image, except those at quality 99, has been resized using the Matlab function `imresize` with bicubic interpolation, using a set of ten possible resizing factors and compressed again using the same JPEG quality factors as the singly compressed images. This resulted in 10 datasets of 15000 doubly compressed and resized images.

We tested both the version of the algorithm based on the *estimated* resizing factor, described in Section III-A, and the version based on *prior* knowledge of γ. In the second case, the algorithm used as candidate γs all ten possible resizing factors. In order to have a performance upper bound, we also tested an *oracle* version of the proposed algorithm in which the correct γ was assumed to be known. In all cases, the algorithms have been applied to a 512×512 portion of the luminance channel of the resized and recompressed image. The counter-resized images has been obtained by using bicubic interpolation. As to the parameters of the resizing factor estimator, $\mathbf{f}_{h,v,\mathrm{median}}(k)$ has been obtained using an 11 point window and we set $\beta = 1.2$. Moreover, we considered only positions t_i satisfying $5N_{\mathrm{DFT}}/64 < t_i < N_{\mathrm{DFT}}/4$.

The ability of the proposed algorithm to discriminate doubly JPEG compressed an resized (DCR) images from singly compressed (SC) images has been measured by estimating the true positive rate (TPR) and false positive rate (FPR) of a detector based on thresholding the min-entropy feature. TPR is defined as the number of images detected as DCR over all DCR images, whereas FPR is defined as the number of images detected as DCR over all SC images.

As to the detection performance, the results obtained by the proposed algorithm have been compared to those obtained by the algorithm of Kirchner et al. in [8] and those obtained by using the blocking artifact characteristics matrix (BACM) features developed in [5]. In order to apply a threshold detector to the BACM features, we projected each feature vector along an optimal direction chosen according to Fisher's linear discriminant analysis [10].

In Fig. 1, we plot the TPR achieved by the different detectors at FPR < 1% for different resizing factors γ, when the quality factor of the second compression (QF_2) was equal to 90 and the quality factor of the first compression (QF_1) was in the interval $[50, \dots, 90]$, whereas in Fig. 2, we plot the TPR achieved by the different detectors at FPR < 1% for different QF_2s, considering $\gamma = 1.2$ and $QF_1 \in [50, \dots, 90]$. The results show that the detector based on prior knowledge of γ is clearly superior to the detector that estimates the
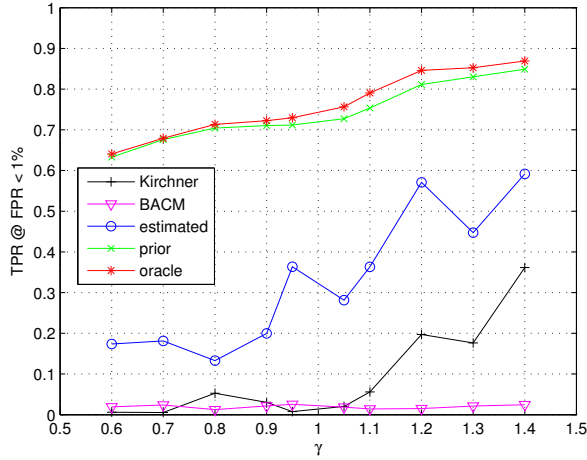
Fig. 1. TPR of the different detectors at FPR $< 1\%$ for different resizing factors γ, considering $QF_2 = 90$ and $QF_1 \in [50, \ldots, 90]$.
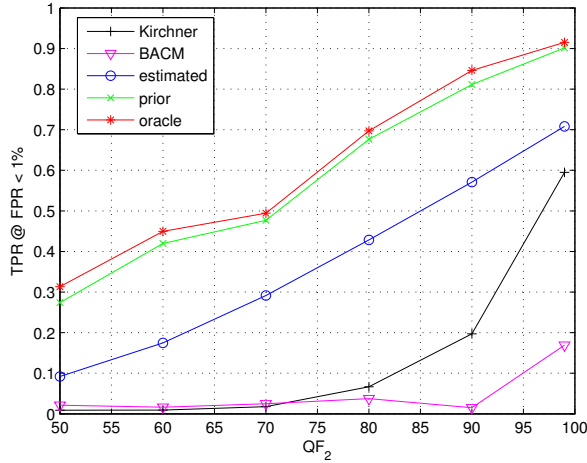

Fig. 2. TPR of the different detectors at FPR $< 1\%$ for different quality factors QF_2, considering $\gamma = 1.2$ and $QF_1 \in [50, \ldots, 90]$.

		QF_2					
		50	60	70	80	90	99
QF_1	50	0.014	0.008	0.050	0.190	0.352	0.722
	60	0.012	0.006	0.014	0.110	0.302	0.680
	70	0.008	0.014	0.002	0.022	0.194	0.646
	80	0.002	0.008	0.008	0.006	0.130	0.506
	90	0.008	0.010	0.014	0.006	0.008	0.420

TABLE I
TPR AT FPR $< 1\%$ ACHIEVED BY THE ALGORITHM IN [8], FOR $\gamma = 1.2$.

		QF_2					
		50	60	70	80	90	99
QF_1	50	0.278	0.538	0.646	0.728	0.766	0.792
	60	0.124	0.258	0.524	0.692	0.770	0.822
	70	**0.018**	0.050	0.242	0.522	0.712	0.822
	80	**0.026**	**0.016**	**0.016**	0.184	0.498	0.702
	90	**0.014**	**0.012**	**0.030**	**0.016**	0.108	0.402

TABLE II
TPR AT FPR $< 1\%$ ACHIEVED BY THE ALGORITHM USING ESTIMATED γ, FOR $\gamma = 1.2$.

		QF_2					
		50	60	70	80	90	99
QF_1	50	**0.658**	**0.798**	**0.794**	**0.830**	**0.834**	**0.836**
	60	**0.678**	**0.712**	**0.836**	**0.872**	**0.868**	**0.876**
	70	0.012	**0.572**	**0.734**	**0.882**	**0.920**	**0.932**
	80	0.010	0.006	0.014	**0.788**	**0.934**	**0.962**
	90	0.012	0.010	0.006	0.010	**0.500**	**0.902**

TABLE III
TPR AT FPR $< 1\%$ ACHIEVED BY THE ALGORITHM USING PRIOR KNOWLEDGE ABOUT γ, FOR $\gamma = 1.2$.

resizing factor, achieving a performance close to that of the oracle detector. Both detectors have better performance than Kirchner's detectors and the BACM detector. Particularly, the BACM features do not appear suitable at all in the case of resized and recompressed images. The results also show that resizing factors greater than one (upsampling) are more easily detectable than resizing factors smaller than one (downsampling). From Fig. 2, it is also evident that DCR images can be reliably detected only when the quality factor of the second compression is sufficiently high ($QF_2 > 80$).

In order to assess the influence of QF_1 on the detection performance, in Tables I-III we report the TPR obtained at FPR $< 1\%$ by Kirchner's detector, the proposed algorithm with estimated γ, and the proposed algorithm with prior knowledge, respectively, for each (QF_1, QF_2) pair and $\gamma = 1.2$. For ease of reading, the best TPRs among the three detectors have been highlighted in bold. The above results show that on our dataset the detector of Kirchner is able to detect DCR images

only when QF_2 is much greater than QF_1. Conversely, the detector based on estimated γ is able to distinguish DCR images when $QF_2 - QF_1 \geq 10$, whereas the detector based on prior knowledge usually work also when $QF_2 = QF_1$, and, for $QF_1 \leq 70$, even when QF_2 is slightly less than QF_1. Surprisingly, lower values of QF_1 seem to be more difficult to detect when QF_2 is very high.

With respect to previous detectors, the proposed algorithm is also able to estimate some parameters of both the resizing step and the previous JPEG compression. In order to evaluate the estimation performance, we measured the mean absolute error (MAE) between the estimated and the true resizing factor and the probability of correctly identifying the first quality factor (P_c), which depends on the estimation of the quantization step of the DC coefficient of the first JPEG compression [6]. The above measures have been computed over the images correctly detected as DCR when FPR was less than 1%. In Fig. 3 we plot the MAE and P_c values, respectively, for different quality factors QF_2. In both cases, results are averaged over all possible resizing factors and quality factors QF_1. From the above results, we can observe that while the algorithm based on prior knowledge of γ has a good estimation performance irrespectively of QF_2, the algorithm estimating γ tends to
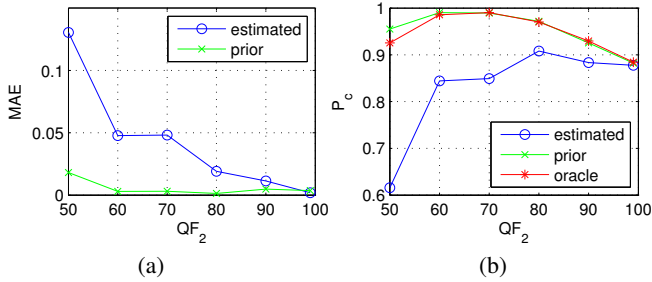
Fig. 3. Estimation performance for different quality factors QF_2. (a) mean absolute error of the estimated resizing factor γ; (b) probability of estimating the correct quality factor QF_1. Results are evaluated over the correctly detected images at FPR $< 1\%$ and averaged over all possible resizing factors and quality factors QF_1.

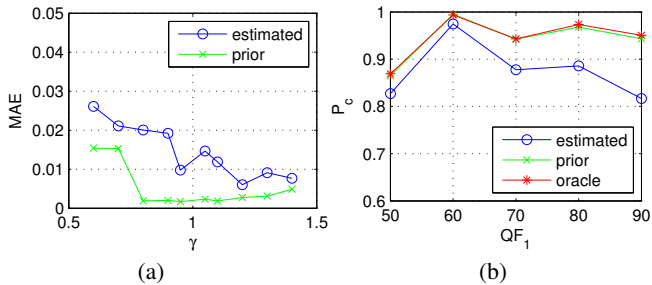
Fig. 4. Estimation performance for different values of the estimated parameters. (a) mean absolute error of the estimated γ; (b) probability of estimating the correct QF_1. Results are evaluated over the correctly detected images at FPR $< 1\%$ and averaged over all possible values of the other parameters.

produce worse estimates for lower QF_2 values.

In order to verify the influence of the to-be-estimated parameter on the estimation performance, in Fig. 4 we plot also the MAE and P_c values for different values of γ and QF_1, respectively. In both cases, results are averaged over all possible values of the other parameters. The results show that resizing factors less than one are usually more difficult to estimate. Moreover, some quality factors of the first JPEG compression are more difficult to estimate than others. This behavior seems related to the actual value of the quantization step Q of the DC coefficient. For example $QF_1 = 60$, corresponding to $Q = 13$, is more easy to identify than $QF_1 = 50$, corresponding to $Q = 16$. However, in the latter case we verified that most of the wrong estimates reported $Q = 8$. Hence, we suppose that quantization steps corresponding to prime numbers may be more easily identifiable.

V. CONCLUSIONS

In this paper, we have presented a practical algorithm for the reverse engineering of a doubly compressed JPEG image when a resizing step has been applied between the two compressions. The method aims at exploiting the fact that previously JPEG compressed images tend to have a near lattice distribution property (NLDP), and that this property is usually maintained after a simple image processing step and subsequent recompression. The results demonstrate that in the presence of prior knowledge regarding the possible

resizing factors, the proposed algorithm is usually able to detect a resized and recompressed image, provided that the quality of the second compression is not much lower than the quality of the first compression. When the resizing factor has to estimated, the detection performance is usually lower, however it remains comparable to or better than that of previous approaches. Differently from existing techniques, the proposed approach is also able to estimate with a reasonably good performance some parameters of the processing chain, namely the resizing factor and the quality factor of the previous JPEG compression. This represents an important novelty with respect to the state of the art, since it may open the way to more detailed forensic analyses.

We are aware that there are still a number of points to be further investigated. First of all, some results (like those in Table III), suggest that the IPM feature may not be the best way to fully exploit the NLDP. Further research is needed both to establish the fundamental limits of the NLDP and devise practical methods achieving those limits. Moreover, the proposed technique relies heavily on the estimation of some hidden parameters of the processing chain. Future research should be devoted to finding computationally efficient methods for this task. It would also be interesting to evaluate the sensitiveness of the algorithm to various parameters, such as different interpolation algorithms, different quantization matrices than those used in Matlab, and the use of chrominance channels.

REFERENCES

[1] H. Farid, "A survey of image forgery detection," *IEEE Signal Processing Mag.*, vol. 2, no. 26, pp. 16–25, 2009.

[2] ——, "Exposing digital forgeries from JPEG ghosts," *IEEE Trans. on Information Forensics and 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] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compression using first digit features," in *Proc. of ICASSP 2012*, Mar. 2012.

[5] W. Luo, Z. Qu, J. Huang, and G. Qui, "A novel method for detecting cropped and recompressed image block," in *Proc. of ICASSP 2007*, vol. 2, 2007, pp. II–217–II–220.

[6] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 2, Apr. 2012.

[7] ——, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, June 2012.

[8] M. Kirchner and T. Gloe, "On resampling detection in re-compressed images," in *First IEEE International Workshop on Information Forensics and Security, 2009*, December 2009, pp. 21–25.

[9] A. C. Gallagher, "Detection of linear and cubic interpolation in JPEG compressed images," in *Second Canadian Conference on Computer and Robot Vision*, 2005, pp. 65–72.

[10] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Human Genetics*, vol. 7, no. 2, pp. 179–188, 1936.