DEMOASAICING STRATEGY IDENTIFICATION VIA EIGENALGORITHMS

Simone Milani, Paolo Bestagini, Marco Tagliasacchi, Stefano Tubaro

Dipartimento di Elettronica, Informazione, e Bioingegneria - Politecnico di Milano,
e-mail: {milani,bestagini}@elet.polimi.it, {marco.tagliasacchi,stefano.tubaro}@polimi.it

ABSTRACT

The identification of the camera that has acquired a specific image can be performed via several device-related footprints. Among these, it is possible to look for the traces left by the adopted color demosaicing strategy, which varies according to the camera model and vendor.

The paper presents an identification strategy that re-processes the analyzed image with a set of distinctive CFA interpolation algorithms (eigenalgorithms) and, according to the correlation of the output with the original image, builds a set of features that permits identifying the algorithm. The proposed solution performs well with respect to other state-of-the-art solutions also when the analyzed image is severely compressed.

Index Terms— demosaicing identification, CFA interpolation, Bayer mask, device identification, image forensics.

1. INTRODUCTION

The identification of the acquisition device is an important clue in the authentication and validation of multimedia content such as images and videos [1].

As for images, several different footprints can be considered as revealing clues useful for the device identification [2]. For example, Photo Response Non Uniformity (PRNU) noise is one of the distinctive feature of the CCD acquisition matrix employed in a camera and, when compression distortion is limited, it permits identifying the originating device [3]. Since most of the images are available in compressed format, several coding artifacts can be detected in an image and used to identify the specific codec implementation since many non-normative aspects are implemented differently by each vendor [4].

In this paper, we focus on the specific Color Filter Array (CFA) interpolation strategy (also known as demosaicing [5]) employed by the camera [6]. In most cameras it is possible to acquire a single color component per pixel, and therefore, color samples are positioned according to a specific acquisition mask (Bayer mask). The missing colors are generated via an interpolation of this mask (before compressing the image), and each camera vendor has developed its own algorithms. Therefore, identifying the demosaicing strategy is an important task in image forensics [7].

Among the possible identification approaches available in the literature [8, 9, 4, 10, 11, 7], some of the proposed blind techniques rely on the fact that some operators are “idempotent” [12]. This means that, if an operator was used to process the signal under analysis, reprocessing the signal with the same operator will generate output data that proves to be highly-correlated with the analyzed data themselves [13]. Unfortunately, the identification of the demosaicing strategy via idempotence is severely conditioned by image compression level (as it will be shown in Section 4). The performance dramatically collapses whenever lossy compression (e.g., using JPEG) is included in the processing chain. Moreover, idempotence-based detectors imply that the investigated method is included in the analysis set, i.e., an implementation of the demosaicing strategy to be identified is available to the analyst. This assumption might not be true in practice.

In this paper, we try to identify the demosaicing strategy by analyzing the interpolated image via eigenalgorithms [14]. The analyzed image is re-processed with a set of distinctive strategies (called eigenalgorithms), which enable to characterize it in an “eigen-correlation” space. The position assumed in the space is a revealing trace of the specific demosaicing approach. The proposed solution permits a correct identification of the interpolation strategy even in the presence of lossy JPEG compression, and does not imply that the algorithm to be identified must be available to the analyst. Experimental results show that the proposed analysis technique can be applied to images taken by consumer cameras and leads to their correct identification. In the following, Section 2 describes the idempotence property, while Section 3 extends this concept introducing eigenalgorithms. Section 4 reports the detection performance and Section 5 draws the final conclusions.

2. THE IDEMPOTENCE PRINCIPLE

A conceptual illustration of the proposed identification scheme based on the idempotent property is presented in Fig. 1. Let \( J_1 \) denote the image under analysis, which is the result of processing the acquired color samples with the unknown algorithm \( X \). The analyst has a set of processing units \( Y \in \mathcal{A} \) at hand. For each of them, he generates
the output data $I_2$ by resampling $I_1$ according to the Bayer mask and interpolating it with algorithm $Y$.

The correlation between $I_2$ and $I_1$ is a distinctive feature of the unknown algorithm $X$. In this research work, it is measured via the PSNR value between pixels of $I_1$ and $I_2$. Assuming that $I_2$ was obtained using the algorithm $Y$, the PSNR value between $I_1$ and $I_2$ is $p_{I_Y}$.

The demosaicing strategy that leads to the highest PSNR is identified with the unknown strategy $X$, i.e.,

$$
\hat{X} = \arg \max_{Y \in A} p_{I_Y}
$$

In the literature, the idempotent property has been successfully exploited for the identification of the quantizer [15], the traces left by JPEG compression antiforensics [16], and the adopted video coding architecture [12, 17].

Experimental results in Section 4 will show that when dealing with CA interpolation strategy detection, the performance is severely conditioned by compression artifacts. Moreover, the set $A$ must include strategy $X$ to be identified; otherwise, it is only possible to identify the algorithm in $A$ that is the closest to $X$ (but does not necessarily correspond to $X$). The proposed method, presented in the next section, is able to circumvent these shortcomings.

3. INTERPOLATION STRATEGY IDENTIFICATION VIA EIGENALGORITHMS

Experimental data shows that some of the strategies in $A$ present different correlation levels with the unknown strategy $X$. This suggests the idea that it is possible to characterize the unknown interpolation algorithm to be identified via the PSNR values obtained by an adequate set $A$ of demosaicing solutions (which does not necessarily include the unknown algorithm $X$). The strategy in this set will be called eigenalgorithms (similarly to eigenvector or eigenfunctions) [14].

Given the set of algorithms in Table 1, Fig. 2 reports the PSNR values obtained by processing a set of images $I$ (which has been demosaiced with strategy $X$) with algorithms A, B, and C. The possible interpolators $X$ are reported in Table 1. Results are plotted in a couple of graph displaying $p_A$ vs. $p_B$ and $p_B$ vs. $p_C$. Note that since no compression was performed, $p_X \rightarrow +\infty$ (with $X = A, B, C$) since the mentioned strategies do not alter the acquired reference color samples. In order to allow a better visualization and computation, values $p_X = +\infty$ are replaced with $p_X = 60$ dB. It is possible to notice that points related to a specific algorithm $X$ occupy a definite region $R_X$ in the three-dimensional feature space (for the sake of clarity, this feature space is split into two 2-D plots in Fig. 2).

Similar results were obtained when the image was compressed. Figure 3 reports the points $(p_A, p_B, p_C)$ obtained from JPEG images coded with QF=85.

In the following, it will be explained how to characterize the decision regions $R_X$ and determine the interpolation strategy given the input vector $p$. More precisely, we distinguished two experimental setups. A first setting consists in a controlled scenario where the whole chain (acquisition, demosaicing, compression) is simulated and all the processing steps are known. A second setting considers images taken by different commercial cameras. In the first case regions of $p$ that characterize each algorithm are more regular, while in the latter, regions have more complex shapes. This fact is mainly due to the influence of other processing units (like white-balancing, gamma correction, compression, rate-distortion optimization, etc.) on the final images. Therefore, the regions $R_X$ can be correctly characterized by training a set of classifiers based on a Support Vector Machine (SVM) [18].

The vectors $p$ related to a specific image is fed to a set of binary SVM classifiers that report whether the processed image has to be related to a specific strategy $Y$, i.e., each classifier tries to state whether $p$ refers to $Y \in A$ or not. Each classifier (associated to the algorithm/camera $Y$) outputs a real value $w_Y \geq 0$, whose absolute value corresponds to the distance from the separating hyperplane.

Combining the outputs of the different classifiers in the variables

$$
W_Y = w_Y - \sum_{Z \in A, Z \neq Y} w_Z
$$

where signs are combined according to the algorithm we are considering, it is possible to estimate the adopted algorithm via the maximum

$$
\hat{X} = \arg \max_{Y \in A} W_Y
$$

Fig. 2. Diagrams of points $p = (p_A, p_B)$ and $p = (p_C, p_B)$ for different demosaicing algorithms. Points labelled pts $X$ denotes values $p$ related to unknown algorithm $X$. Points belonging to different algorithms cluster in regions $R_X$. a) Points $p = (p_A, p_B)$ b) Points $p = (p_C, p_B)$.

Fig. 3. Diagrams of points $p = (p_A, p_B)$ and $p = (p_C, p_B)$ for different demosaicing algorithms on compressed images. For the sake of conciseness, the convention of notation is the same of Fig. 2. The JPEG quality factor of images is 85. a) Points $p = (p_A, p_B)$ b) Points $p = (p_C, p_B)$.
A possible explanation for this is provided by Fig. 4, which reports the absolute coefficient values of DFT-transformed image peppers averaged along the columns (vertical frequencies) and along the rows (horizontal frequencies). The image has been sampled using RGGB Bayer mask and reinterpolated with algorithms A,B,E. It is possible to notice that the different algorithms differs mostly in processing the input signal in the transient band between low and high frequencies (approximately between 0.07 and 0.1). Low frequency information can be easily estimated by all the proposed solutions. High frequency components prove to be more difficult to reconstruct and different interpolation strategies lead to different amounts of artifacts (e.g., aliasing, rainbows, etc.). In the reported figure, this differentiation is quite evident in the vertical interpolation (Fig. 4b) since the spectra of the three signals are quite separated. As a matter of fact, these differences provide some footprints for the specific interpolation strategy.

These can be acquired by resampling and reinterpolating the image $I_1$, followed by evaluating the differences with respect to the original data (as described before). From Fig. 4 (c) and (d), it is possible to notice that when the process interpolated with $A$ is reprocessed with $E$ the difference between absolute frequency response is quite close to 1 dB. In case, the process is reprocessed with $B$ the average difference is lower.

### 4. EXPERIMENTAL RESULTS

In a first set of experiments, we considered the identification of known demosaicing algorithms. To this purpose, images from the UCID dataset [19] were sampled, interpolated, and compressed with different quality factors. In this work we considered the demosaicing strategies reported in Table 1. These strategies were selected since they are clearly described in the literature, results are reproducible, and their performance proves to be quite satisfactory. Moreover, they present a good degree of diversity that makes the different py values more informative with respect to using similar interpolation strategies. This fact was further validated by an extensive set of experiments on the UCID dataset involving a wider set of demosaicing algorithms. Results are omitted here for the sake of conciseness.

The adopted JPEG codec is the one implemented in MATLAB. On the coded images we tested the proposed identification strategy, a detector based on idempotence (i.e., using eq. (1) under the assumption that all the demosaicing strategies are available to the analyst) and that proposed by Gao et al. in [20].

In order to evaluate their efficiencies, we tested the detectors in two scenarios. A first scenario considers a set of non compressed images [19], where color components are sampled according to a RGGB Bayer mask, interpolated using one of the algorithms in Table 1, and eventually, compressed (see Fig. 1).

Similarly to the approaches in [24, 20], for every image we select an area of $352 \times 288$ pixels (CIF resolution) where the analysis is performed. In our analysis we skipped those blocks with too stationary behavior or too much varying since the outputs of different demosaicing strategy is not discriminative in the first case or significant in the latter (as done in [24]). As a matter of fact, we process the image via a sliding window of CIF resolution computing the average absolute gradients (vertical and horizontal) on the acquired color samples. For every image, we select the CIF window with the highest gradient included in the range 5 and 12. In this way it is possible to identify those parts where the peculiarities of the different interpolation strategies are highlighted. Source code for this selection is available at [25]

Table 1. Demosaicing strategies

<table>
<thead>
<tr>
<th>Letter</th>
<th>Algorithm</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>DLMMSE</td>
<td>[21]</td>
</tr>
<tr>
<td>B</td>
<td>Alternating Projections</td>
<td>[22]</td>
</tr>
<tr>
<td>C</td>
<td>hold</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>bilinear</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Self-similarity Driven</td>
<td>[23]</td>
</tr>
</tbody>
</table>

Fig. 4. Fourier transform of image peppers sampled and interpolated with different algorithms. Graphs report $|DFT(I_1)|_{dB}$ averaged along columns (a) and rows(b), together with $|DFT(I_1)|_{dB} - |DFT(I_2)|_{dB}$ (with $I_1 = A$) averaged along columns (c) and rows (d). Frequency $f$ is normalized.
based approach can be explained considering that the energy of the compression noise is mainly located at high frequencies. By reinterpolating the signal, part of this energy is removed (since the interpolator is generally a low pass operator) leading to a less corrupted version of the image. This permits a more accurate detection. The detailed confusion matrices are reported at \[25\].

A second set of experiments was performed considering real cameras. To this purpose, a set of JPEG images was downloaded from the Dresden dataset \[26\] as in \[20\]. Four central areas of \(512 \times 512\) pixels were used in this case (as described in \[20\]). Since the classification outcomes is found combining the classification results of these 4 images, the average precision of the different algorithms is a bit higher with respect to the first scenario (where a single feature array was computed on the whole image). Note also that image resolution is higher, and therefore, frequency spectra of coding blocks proves to fit a low-pass model better than images with smaller resolution. As a consequence, the amount of interpolation noise introduced by the different analysis algorithms is lower, leading to a more accurate identification. We downloaded images acquired with FujiFilm, Nikon Coolpix, and Nikon D70s cameras. The list of adopted images is available at \[25\]. In order to vary the amount of compression noise applied to images, we recompressed the decoded JPEG images with different quality factors \(QF\)’s. Since a lot of additional noise sources (derived from the applied processing steps like white-balancing, gamma correction, etc.) affect the image, experimental results showed that using vectors of 5 features \(\mathbf{p} = (p_A, p_B, p_C, p_D, p_E)\) permits an accurate characterization.

Each feature vector \(\mathbf{p}\) is classified via a set of SVM classifiers (one per camera). Fig. 6 reports the detection results for the approach in \[20\] and the proposed one. Table 2 reports the complete confusion matrices obtained from uncompressed images and from images coded with \(QF=90, 80\). All the data are available at \[25\]. It is possible to see that the robustness to compression is confirmed also in this case. The accuracy is higher than 80% with \(QF > 85\) for the eigenalgorithms-based approach, while the solution in \[20\] has a lower performance when strong compression is introduced. It is worth noticing that the performance of the algorithm in \[20\] is slightly better than the proposed solution whenever no compression is applied. This is due to the adoption of 69 feature values in place of the 5 feature values adopted by the eigenalgorithms-based approach.

The paper presented a new strategy for the identification of the CFA interpolation strategy on static images that relies on reprocessing the analyzed image with some known demosaicing strategies, called “eigenalgorithms”. The correlation between the obtained output and the original image under analysis permits identifying the original interpolation strategy with a nice accuracy (> 80%) even when the signal has been corrupted by compression noise. Future work will be devoted to investigate the influence of the number of features/eigenalgorithms on the detection performance. Moreover, we are planning to extend the approach to the identification of a wider set of algorithms in the presence of different noise sources. Preliminary results have shown that it is also possible to reverse the approach using the eigenalgorithms to generate forged images with consistent CFA-related footprints. Implementing this possibility, it will be possible to design antiforensic strategies for the synthesis of demosaicing algorithms that could be used to fool CFA-based camera detectors.

**Table 2. Confusion matrix for strategies eigen-based/Gao et al.**

<table>
<thead>
<tr>
<th></th>
<th>Fuji</th>
<th>CoolPix</th>
<th>D70s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuji</td>
<td>100.00/100.00</td>
<td>0.00/0.00</td>
<td>0.00/0.00</td>
</tr>
<tr>
<td>CoolPix</td>
<td>25.00/0.00</td>
<td>75.00/100.00</td>
<td>0.00/0.00</td>
</tr>
<tr>
<td>D70s</td>
<td>0.00/8.33</td>
<td>0.00/8.33</td>
<td>100.00/83.33</td>
</tr>
</tbody>
</table>

**No loss**

<table>
<thead>
<tr>
<th></th>
<th>Fuji</th>
<th>CoolPix</th>
<th>D70s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuji</td>
<td>100.00/91.67</td>
<td>0.00/0.00</td>
<td>0.00/8.33</td>
</tr>
<tr>
<td>CoolPix</td>
<td>33.33/25.00</td>
<td>66.67/41.67</td>
<td>0.00/33.33</td>
</tr>
<tr>
<td>D70s</td>
<td>0.00/8.33</td>
<td>0.00/0.00</td>
<td>100.00/91.67</td>
</tr>
</tbody>
</table>

**QF=90**

<table>
<thead>
<tr>
<th></th>
<th>Fuji</th>
<th>CoolPix</th>
<th>D70s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuji</td>
<td>91.67/91.67</td>
<td>8.33/0.00</td>
<td>0.00/8.33</td>
</tr>
<tr>
<td>CoolPix</td>
<td>33.33/33.33</td>
<td>66.67/16.67</td>
<td>0.00/50.00</td>
</tr>
<tr>
<td>D70s</td>
<td>0.00/8.33</td>
<td>0.00/0.00</td>
<td>100.00/91.67</td>
</tr>
</tbody>
</table>

**QF=85**

5. CONCLUSIONS
6. REFERENCES


