Patch-Based Desynchronization of Digital Camera Sensor Fingerprints

John Entrieri and Matthias Kirchner
Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902, USA

Abstract

This paper explores image self-similarity as a means to impede forensic camera identification based on sensor noise. We follow the tradition of patch replacement attacks against robust digital watermarks, putting particular emphasis on the use of PatchMatch, an efficient algorithm for finding approximate nearest neighbor patches. Experimental results suggest that sensor noise fingerprints can be desynchronized reasonably well without the need of geometric image transformations and without explicit knowledge of the fingerprint, while maintaining a tolerable visual quality.

Introduction

Scalable tests for digital camera identification based on highly robust sensor noise fingerprints link digital images to their source device with remarkable reliability [1, 2]. This quality is commonly attributed to photo-response non-uniformity (PRNU), a camera-specific multiplicative noise pattern caused by inevitable material imperfections and variations in the manufacturing process of sensor elements. PRNU occurs very similarly for images captured with the same camera, but differs substantially between images from different cameras. With access to enough images taken by a certain digital camera, it is possible to estimate the noise pattern resulting from these imperfections and to establish a fingerprint unique to the device. The ability to identify the source camera from an image and to link seemingly unrelated images from the same camera holds great potential for forensic applications. Yet it also brings to the surface many concerns for the *anonymity* of photographers, who may become identifiable through the analysis and combination of information derived from one or multiple images. This is not always desired [3]. *Counter-forensic techniques* [4] to suppress traces of origin in digital images thus become a relevant building block for ensuring unlinkability [5] for anonymous image communication, e. g., in the case of journalism, activism, or legitimate whistle-blowing.

Since the camera sensor fingerprint is a spatially varying noise pattern, irreversible desynchronization is one approach to impede camera identification—a strategy with traditions in digital watermarking and image forensics [6, 7]. Recent works based on seam-carving follow this line of thought [8, 9], yet require a substantial reduction of image resolution to be successful. This paper explores the potential of *patch-replacement attacks* [10, 11] as a tool to desynchronize sensor noise fingerprints instead. Also this approach has its origin in attacks against watermarking systems, where it achieved watermark removal with considerably less image degradation than simple image processing primitives that have been discussed in the context of sensor fingerprint robustness as well (strong JPEG compression or denoising, for instance). We combine the idea of distortion-constrained patch-replacement with PatchMatch [12], a modern approximate nearest neighbor search scheme. The result is a heuristic, yet effective algorithm to suppress sensor noise fingerprints while maintaining a surprisingly high visual quality, specifically so in *textured* image regions.

Before we present our approach in more detail in the remainder of this text, the following section reviews the basics of sensor noise forensics and corresponding countermeasures. We continue with a discussion of potential patch-replacement strategies and experimental results, before we conclude the paper.

Notation Without loss of generality, we consider grayscale images of size $U \times V$, which we express as vectors $\mathbf{x}$ of 8-bit intensity values, $\mathbf{x} = (x_i)$, $0 \le i < UV$, $x_i \in [0, 255]$. Indices i are organized in column-major order. At times it will be more convenient to refer to spatial indices (u, v) explicitly, with row index $u = u(i) = i - U\lfloor i/U \rfloor$ and column index $v = v(i) = \lfloor i/U \rfloor$. We will frequently consider *patches* $\mathbf{x}^{(m)}$, defined as small contiguous $P \times P$ image regions, $\mathbf{x}^{(m)} = \left(x_p^{(m)}\right)$, $0 \le p < P^2$, where patch index m denotes the m-th patch in the image $\mathbf{x}$. Patches may overlap by $0 \le O < P$ pixels in horizontal and vertical direction, such that the upper-left corner of the m-th patch corresponds to pixel $i = i(m)$,

$$i(m) = (P - O) \cdot \left(m + \left(U - \left\lfloor \frac{U-O}{P-O} \right\rfloor \right) \cdot \left\lfloor \frac{m}{\lfloor \frac{U-O}{P-O} \rfloor} \right\rfloor \right),$$

$0 \le m < \lfloor \frac{U-O}{P-O} \rfloor \cdot \lfloor \frac{V-O}{P-O} \rfloor$. Patches are said to be fully overlapping, if $O = P - 1$. Overlap $O = 0$ results in non-overlapping patches. Patch elements are indexed relative to $i(m)$, i. e.,

$$x_p^{(m)} = x_{i(m)+p-(U-P)\lfloor p/P \rfloor}, \qquad 0 \le p < P^2.$$

We define the convenience functions $\mathsf{left} : m \mapsto m - \lfloor \frac{U-O}{P-O} \rfloor$ and $\mathsf{up} : m \mapsto m - 1$, which return the indices of patches $\mathbf{x}^{(\mathsf{left}(m))}$ and $\mathbf{x}^{(\mathsf{up}(m))}$ to the left and above of patch $\mathbf{x}^{(m)}$, respectively (assuming that such neighbors exist). Similar functions can be defined to obtain patches to the right or below. These functions may also be applied to pixel indices, which implicitly implies $P = 1$ and $O = 0$ in the definitions above. Operators $\odot$ and $\oslash$ denote element-wise multiplication and division; $[\mathbf{x}]$ denotes element-wise rounding and truncation according to the dynamic range of $\mathbf{x}$.

Background

Sensor noise is among the best-studied device characteristics in digital image forensics. This section briefly summarizes the estimation of sensor noise fingerprints, their application to digital camera identification, as well as countermeasures to impede successful identification.

Camera Identification

Estimating the sensor noise fingerprint of a digital camera c requires access to a sufficiently large number of images $\mathbf{x}_1, \ldots, \mathbf{x}_L$ from that camera. Each of the L images is denoised to obtain noise residuals $\mathbf{r}_l = \mathbf{x}_l - \mathsf{denoise}(\mathbf{x}_l)$, commonly modeled as [1]

$$\mathbf{r}_l = \mathbf{x}_l \odot \mathbf{k}_c + \boldsymbol{\Theta}_l \,. \tag{1}$$

Multiplicative factor $\mathbf{k}_c$ is the camera-specific PRNU term, i. e., the sensor noise fingerprint. The noise term $\boldsymbol{\Theta}$ subsumes a variety of other noise sources. It is assumed to be i.i.d. Gaussian. We use the maximum likelihood estimator in [1] to obtain an estimate $\hat{\mathbf{k}}_c$ of a camera's sensor noise fingerprint. The estimates require post-processing to remove non-unique artifacts, e. g., due to demosaicing or lens distortion correction [1, 13, 14]. For a given query image $\mathbf{x}_q$ and its noise residual $\mathbf{r}_q = \mathbf{x}_q - \mathsf{denoise}(\mathbf{x}_q)$, digital camera identification can be established by evaluating the peak-to-correlation energy (PCE) [1],

$$s_c = \mathsf{PCE}(\mathbf{r}_q, \mathbf{x}_q \odot \hat{\mathbf{k}}_c) \,. \tag{2}$$

A similarity score $s_c > 60$ has been suggested to be a robust indicator that image $\mathbf{x}_q$ was captured by camera c [1].

Countermeasures

The reliability and the robustness of camera identification based on sensor noise have been under scrutiny ever since seminal works on sensor noise forensics surfaced about a decade ago. Two major goals of countermeasures can be distinguished [15, 16]. *Fingerprint removal* concerns the suppression of a camera's fingerprint to render source identification impossible. *Fingerprint copying* attempts to make an image plausibly appear as if it was captured by a different camera. The latter strictly implies the suppression of the original fingerprint and is generally a much harder problem [17, 18]. The success of such counter-forensic techniques is to a large degree bound by the admissible visual quality of the resulting image. If anonymity is of utmost priority, strong measures that go along with a severe loss of image resolution are more likely acceptable.

Existing fingerprint removal methods can be categorized under two general approaches [15]. Methods of the first category are *side-informed* in the sense that they use an estimate of the sensor noise fingerprint to ensure a detector output below the identification threshold. Flatfielding is known to remove the multiplicative noise term $\mathbf{k}_c$, yet ideally requires access to the raw image data [15, 16]. Adaptive fingerprint removal techniques explicitly attempt to minimize Equation (2) by finding a noise sequence that cancels out the multiplicative fingerprint term in Equation (1) [19], ideally by having exact knowledge of the detector (and thus the images used to estimate $\hat{\mathbf{k}}_c$). *Uninformed* techniques make less assumptions and directly address the robustness of the sensor noise fingerprint. Methods of this category apply post-processing to the image until the noise pattern is too corrupted to correlate with the fingerprint. No specific knowledge of the camera c, the camera's fingerprint $\mathbf{k}_c$, or the detector is necessary. However, due to the high robustness of the sensor fingerprint, this is generally a non-trivial problem. The drawback of existing approaches is a more immediate loss of image quality compared to side-informed methods. It has been reported repeatedly that even strong JPEG compression or repeated denoising are generally not sufficient to remove sensor noise fingerprints [20]. Irreversible desynchronization is a more promising approach. A recent work proposes the use of seam-carving, a form of content-adaptive resizing [21], to impede camera identification [8]. A major limitation of the seam carving method is that a considerable amount of seams must be removed to successfully desynchronize the sensor fingerprint [9], with a high potential for the removal of "important" seams, thus degrading image quality and resolution.

Patch Replacement Strategies

One of the most characteristic properties of natural digital images is their redundancy. Small image patches recur in very similar form numerous times across an image [22], and also across different scales [23]. This *self-similarity* has been exploited in various forms, for instance for image coding [24], texture synthesis [25], denoising [26], or super-resolution [23]. Patch recurrence has also driven the design of *patch replacement attacks* against robust watermarking schemes for still images [11]. The key idea of this type of attacks is to find for each image patch $\mathbf{x}^{(m)}$ a replacement patch $\tilde{\mathbf{x}}^{(m)}$ that is as similar as possible to the original patch yet does not contain the watermark. Such attacks have been demonstrated to be highly effective against spread-spectrum watermarking and eventually led to the call for signal-coherent watermarking [27].

Given the conceptual similarity of robust watermarks and sensor noise camera fingerprints, we expect that a suitable patch replacement strategy may also serve as counter-forensic technique to impede camera identification. In the following we first review a projection-based strategy proposed in [11], before we discuss our approach based on the PatchMatch algorithm [12].

PCA-Based Patch Replacement

Doërr et al. [11] discuss a number of variants of patch replacement strategies for attacks against robust watermarks. The authors conclude that an approach based on the principal component analysis (PCA) offers the best trade-off between removal effectiveness and visual quality in their setup.

Specifically, the algorithm constructs for every $P \times P$ patch $\mathbf{x}^{(m)}$ a code book $\mathscr{Q}_m = \{\mathbf{x}^{(q)}\}$, $|\mathscr{Q}_m| = Q$, $q \neq m$, by collecting Q patches of size $P \times P$ from the neighborhood around $\mathbf{x}^{(m)}$. Computing the PCA of the code book[1] results in Q unit-length eigenpatches $\mathbf{e}_q$, which are sorted in descending order such that $\mathbf{e}_0$ is associated with the largest eigenvalue and $\mathbf{e}_{Q-1}$ corresponds to the smallest eigenvalue. Denoting $\mathbf{c}$ as the centroid of the code book, the replacement patch is obtained by projecting $\mathbf{x}^{(m)} - \mathbf{c}$ onto the subspace spanned by the first $K \leq Q$ eigenpatches,

$$\tilde{\mathbf{x}}^{(m)} = \mathbf{c} + \sum_{k=0}^{K-1} \left((\mathbf{x}^{(m)} - \mathbf{c}) \cdot \mathbf{e}_k \right) \mathbf{e}_k \,, \tag{3}$$

such that the mean squared error (MSE) satisfies

$$\mathsf{MSE}\left(\mathbf{x}^{(m)}, \tilde{\mathbf{x}}^{(m)}\right) = \frac{1}{P^2} \sum_{p=0}^{P^2-1} \left(x_p^{(m)} - \tilde{x}_p^{(m)} \right)^2 \geq \tau \tag{4}$$

for some threshold $\tau \geq 0$. We refer to [11] for a more detailed description of the algorithm. In general, the MSE between the original patch and the replacement patch decreases as the number K of

[1] It is advised to employ a photometric correction of the code book before the PCA, see [11] for details.

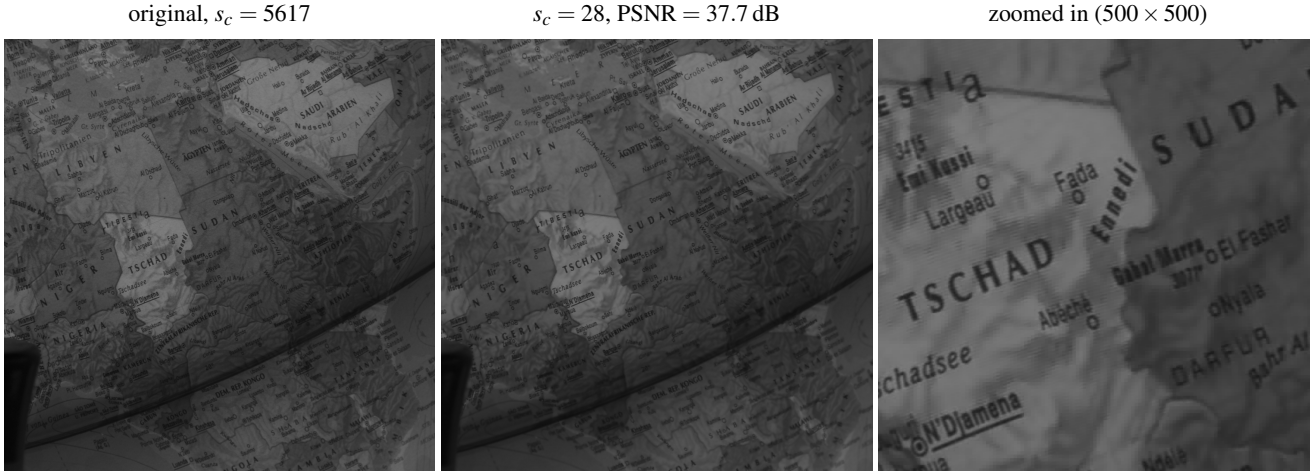

Figure 1. *Results of PCA-based patch replacement (center) and close-up of a* 500×500 *region (right). The original image (shown on the left) is of size* 2000×2000*. Half-overlapped* 8×8 *patches,* $\tau = 20$*. PCE and PSNR values are reported above the images.*

eigenpatches increases. In contrast to [11], we opt for a conservative strategy and allow $K = 0$, if even a single eigenpatch brings the replacement patch too close to the original (thus increasing the likelihood of successful camera identification). In this case, we replace $\mathbf{x}^{(m)}$ with the centroid $\mathbf{c}$.

Since the replacement strategy computes the optimal projection for each patch independently, it is computationally expensive to run the algorithm on fully overlapping patches for large images. Following [11], we work with half-overlapped patches ($O = P/2$) and average spatially corresponding portions of overlapping replacement patches to obtain the final image. Figure 1 gives an example of a typical outcome for patch size $P = 8$ and distortion threshold $\tau = 20$, with code books obtained from the 64×64 regions around the patches. The original image (size 2000×2000) is from the Dresden Image Database [28]. The PSNR of the final image is 37.7 dB, the PCE value drops from 5617 to 28.

PatchMatch-Based Replacement

PatchMatch [12] is an efficient correspondence algorithm for approximate nearest neighbor search, specifically suited for image analysis and processing. Given two images $\mathbf{x}$ and $\mathbf{y}$, the algorithm determines a dense approximate nearest neighbor field $\mathbf{z} = (z_m)$, which specifies for each patch $\mathbf{x}^{(m)}$, $0 \le m < M$, the index n of a similar patch $\mathbf{y}^{(n)}$, $0 \le z_m, n < N$. Patch similarity can be measured in the pixel domain or in a suitable feature space. PatchMatch inherently relies on the fact that sufficiently small patches recur in similar form numerous times across an image, and that patches in close proximity tend to share common characteristics. After a randomized initialization of the nearest neighbor field $\mathbf{z}$, the algorithm operates in an iterative manner. Each iteration consists of a propagation step followed by a random search step. In the *propagation step*, PatchMatch scans image $\mathbf{x}$ from left to right, top to bottom, and inspects for each patch index m whether patches $\mathbf{y}^{(\mathsf{right}(z_{\mathsf{left}(m)}))}$ or $\mathbf{y}^{(\mathsf{down}(z_{\mathsf{up}(m)}))}$ are more similar to $\mathbf{x}^{(m)}$ than $\mathbf{y}^{(z_m)}$. If so, z_m gets updated correspondingly. After a full scan of the image, the *random search step* mitigates effects of local minima by extending the search for better matches to a randomized sequence of candidate patches. The next iteration repeats those steps with a reversed scan order. The propagation step is the key to the success of PatchMatch. It will quickly collect larger contiguous regions of good matches. A relatively small number of iterations is thus typically sufficient to obtain reasonable mappings.

More general formulations of the algorithm consider matching across rotations and scale [29] or explicit smoothness constraints [30], amongst others. We refer to [12, 29] for an overview of applications, yet it is worth pointing out that a PatchMatch-based algorithm has recently also been applied in the field of copy-move forgery detection [31].

Adjustments

We utilize PatchMatch to desynchronize camera sensor noise fingerprints by replacing patches with suitable content from elsewhere. Our focus is on information available in the image $\mathbf{x}$ that should be anonymized, i. e., we attempt to find for each patch $\mathbf{x}^{(m)}$ a replacement patch $\tilde{\mathbf{x}}^{(m)} = \mathbf{x}^{(z_m)}$ (and thus $\mathbf{y} = \mathbf{x}$). PatchMatch is efficient enough to handle fully overlapping patches. Running PatchMatch "off the shelf" would result in too good replacement patches. In fact, with $\mathbf{y} = \mathbf{x}$, PatchMatch will most likely converge to the identity mapping, $z_m = m$. With MSE as patch similarity measure, a straightforward adjustment of the basic matching procedure imposes an additional constraint

$$\mathsf{MSE}\left(\mathbf{x}^{(m)}, \mathbf{x}^{(z_m)}\right) = \frac{1}{P^2} \sum_{p=0}^{P^2-1} \left(x_p^{(m)} - x_p^{(z_m)}\right)^2 \ge \tau, \tag{5}$$

such that z_m is only updated, if a candidate patch gives a lower MSE than the candidate from the previous iteration *and* if the new MSE remains above a threshold τ at the same time.

Once the approximate nearest neighbor field (z_m) is available, we obtain the final image by averaging spatially corresponding portions of the replacement patches. We parametrize the averaging process by weighting each patch pixel-wise with a symmetric $P \times P$ Gaussian mask $\mathbf{h} = (h_p) = (h_{u,v})$ centered at the patch center,

$$h_{u,v} \propto \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\left(u - \frac{P-1}{2}\right)^2 + \left(v - \frac{P-1}{2}\right)^2}{2\sigma^2}\right). \tag{6}$$

Larger values of σ result in an overall higher visual quality.

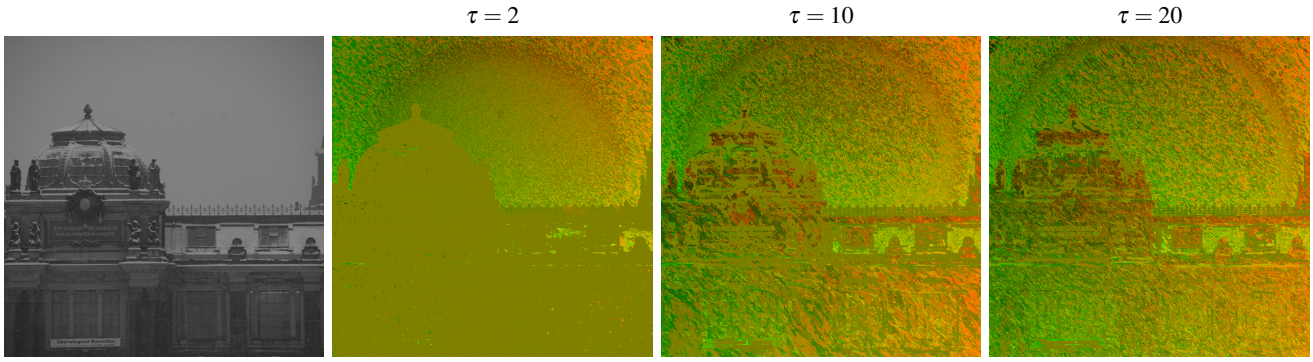

Figure 2. *PatchMatch displacements* $\mathbf{d} = (m - z_m)$ *for thresholds* $\tau \in \{2, 10, 20\}$. *Coordinate displacements in the horizontal direction are encoded in the red color channel, vertical displacements in the green channel. The original image (on the left) is of size* 2000×2000. *Fully overlapping* 8×8 *patches, eight iterations.*

Note that setting $\tau = 0$ in Equation (5) results in the original formulation of PatchMatch, whereas a threshold $\tau > 0$ will affect the propagation properties of the algorithm. Although it can still be expected that a pair of patches $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(z_m)}$ with $\mathsf{MSE}\big(\mathbf{x}^{(m)}, \mathbf{x}^{(z_m)}\big) \approx \tau$ will also have neighbors with similar properties, there exist many more valid candidate patches that fulfill the MSE constraint. Due to the randomized nature of the algorithm, this leads to a less smooth approximate nearest neighbor field as τ increases. This effect can be observed in Figure 2, which presents the outcome of MSE-constrained PatchMatch for one example image from the Dresden Image Database [28] in the form of approximate nearest neighbor displacement fields $\mathbf{d} = (m - z_m)$, obtained from fully overlapping patches of size 8×8 after eight iterations for three different thresholds $\tau \in \{2, 10, 20\}$.

While a less smooth approximate nearest neighbor field is generally advantageous for sensor noise desynchronization, it will also affect the visual quality of the resulting image. We found that artifacts are particularly visible in dark image regions, yet annoying streaking is generally present in predominantly smooth areas where adjacent patches may be replaced with content from different positions. This is illustrated in the center panel of Figure 3, which compares the results of MSE-constrained PatchMatch (fully overlapping 8×8 patches, $\tau = 20$, $\sigma = 1$) for a scene from the Dresden Image Database [28], shot once with camera flash off and once with flash on. We mitigate these effects by adopting a replacement strategy that replaces too smooth patches with an i.i.d. Gaussian noise vector:

$$\tilde{x}_p^{(m)} = \begin{cases} x_p^{(z_m)} & \text{if } \mathrm{Var}(\mathbf{x}^{(m)}) \geq \nu \\ r_p \sim \mathcal{N}(\bar{\mathbf{x}}^{(m)}, \sigma_r) & \text{else} \end{cases}, \tag{7}$$

where $\mathbf{z} = (z_m)$ is the approximate nearest neighbor field obtained from the MSE-constrained PatchMatch algorithm, $\bar{\mathbf{x}}^{(m)}$ and $\mathrm{Var}(\mathbf{x}^{(m)})$ are the empirical mean and variance of the the m-th patch, ν is a suitable threshold, and σ_r is the standard deviation of the noise source. We found $\sigma_r = 1$ to work sufficiently well in our experiments. The right panel of Figure 3 depicts the results with $\nu = 5$ for the two images from above. The higher visual quality is also reflected in the PSNR values, which increase from 37.3 (36.2) dB to 41.9 (38.2) dB. The PCE values are well below 60 for both images. Figure 4 gives another example, applying the same settings as above to the image in Figure 1. Both PCA and PatchMatch give acceptable results, yet we observed that the latter tends to yield less ringing artifacts in textured regions. This can be attributed both to a higher resolution due to fully-overlapping blocks and to a less constrained patch-replacement strategy that does not depend on the quality of a relatively small code book.

Experiments

We work with a random subset of 500 never-compressed Adobe Lightroom images from the Dresden Image Database [28] (Nikon D70, Nikon D70s, Nikon D200, two devices each). All images were synchronized to landscape orientation, cropped to a common size of 2000×2000 pixels, and converted to grayscale before any further processing. Noise residuals were computed with the standard Wavelet denoising filter [32]. Clean sensor fingerprint estimates $\hat{\mathbf{k}}$ were obtained from 25 homogeneously lit flat field images per camera, applying the post-processing suggested in [1]. All six cameras by and large gave similar results, so we aggregate our outcomes over these devices.

The patch size is $P = 8$ in all our experiments. If not stated otherwise, the PCA-based approach works with overlap $O = 4$ and code books $\mathscr{Q}_m$ constructed from the 64×64 neighborhood around $\mathbf{x}^{(m)}$; PatchMatch[2] runs with fully overlapping patches ($O = 7$) and eight iterations, with a smoothness threshold of $\nu = 5$.

Evaluation Criteria

We report ROC curves to measure the performance of digital camera identification. PCE values s_c obtained from images taken with camera c define the set of true positives. The set of true negatives comprises PCE values s_c from all unprocessed images not taken with camera c. The area under the ROC curve (AUC), expressed as "detection reliability" $\rho = 2 \cdot \mathrm{AUC} - 1$ [33], and the true positive rate at a false positive rate of 1 %, $\mathrm{TPR}_{0.01}$, serve as scalar performance measures. Larger values imply higher identification rates. Image quality is reported in terms of peak-signal-to-noise ratio (PSNR). Larger values suggest better visual quality.

Baseline

Our baseline experiment confirms the high reliability of camera identification based on sensor noise, so we refrain from plotting any curves. We found that true positive PCE values fall in the range $119 < s_c < 20955$, all true negatives lie in $22 < s_c < 60$. The true negative 95 % quantile is 43, the median is 28.9.

[2] `http://gfx.cs.princeton.edu/pubs/Barnes_2009_PAR`

original, $s_c = 3512$ | $v = 0$, $s_c = 30$, PSNR $= 37.3$ dB | $v = 5$, $s_c = 44$, PSNR $= 41.9$ dB

original, $s_c = 5578$ | $v = 0$, $s_c = 28$, PSNR $= 36.2$ dB | $v = 5$, $s_c = 30$, PSNR $= 38.2$ dB

Figure 3. *Results of MSE-constraint patch replacement with PatchMatch for smoothness thresholds $v = 0$ (center) and $v = 5$ (right). The original images (shown on the left) are of size 2000×2000. Fully overlapping 8×8 patches, eight iterations, $\tau = 20$, $\sigma = 1$. PCE values s_c and PSNR values are reported above the images.*

original, $s_c = 5617$ | $s_c = 32$, PSNR $= 38.3$ dB | zoomed in (500×500)

Figure 4. *Results of MSE-constraint patch replacement with PatchMatch (center) and close-up of a 500×500 region (right). The original image (shown on the left) is of size 2000×2000. Fully overlapping 8×8 patches, eight iterations, $\tau = 20$, $\sigma = 1$, $v = 5$. PCE and PSNR values are reported above the images.*

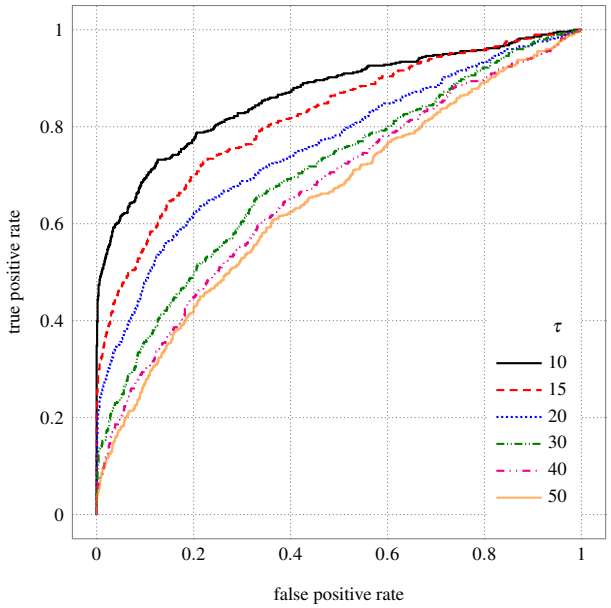

Figure 5. *Camera identification ROC curves after PCA-based patch replacement for different distortion settings τ.*

PCA-Based Patch Replacement

Figure 5 reports ROC curves for patch replacement following the PCA-based projection strategy with overlapping patches. We tested distortion constraints $\tau \in \{10, 15, 20, 30, 40, 50\}$. The curves clearly indicate that patch replacement has the desired effect. Stronger distortion will have a stronger impact on camera identification. Yet it also becomes apparent that the impact saturates as τ increases. This is inherently due to the nature of the replacement strategy, which will resort to the code book centroid for large τ more and more frequently. Our results thus suggest that the projection-based approach has limits and may not be sufficient to impede camera identification in all cases. Figure 6 sheds more light on the visual quality of the resulting images by plotting ρ and $\text{TPR}_{0.01}$ for different values of τ as functions of median PSNR values. Observe that the color of the symbols corresponds to the color of the respective ROC curves in Figure 5. As to be expected, the graphs indicate that lower values of τ give higher visual quality, yet against the backdrop of higher identification reliability. For comparison, we also plot the corresponding results for patch replacement with non-overlapping blocks, which reduces the ρ and $\text{TPR}_{0.01}$ measures but tends to yield median PSNR values that are at least 1 dB lower across all settings of τ.

PatchMatch-Based Patch Replacement

Figure 7 summarizes the camera identification results after patch replacement with PatchMatch for distortion constraints $\tau \in \{10, 15, 20, 30\}$. Each panel depicts ROC curves for various Gaussian blur settings, $\sigma \in \{0.1, 0.75, 1, 2, 4\}$. Symbol '*' indicates non-overlapping patches, which we consider for comparison. We find that larger values of τ and smaller values σ have stronger impact on camera identification reliability. Corresponding PSNR values, along with ρ and $\text{TPR}_{0.01}$ measures are reported in Figure 8, which adopts the color coding of the ROC curves. Overall, we find that there is no single best parameter setting. The PatchMatch

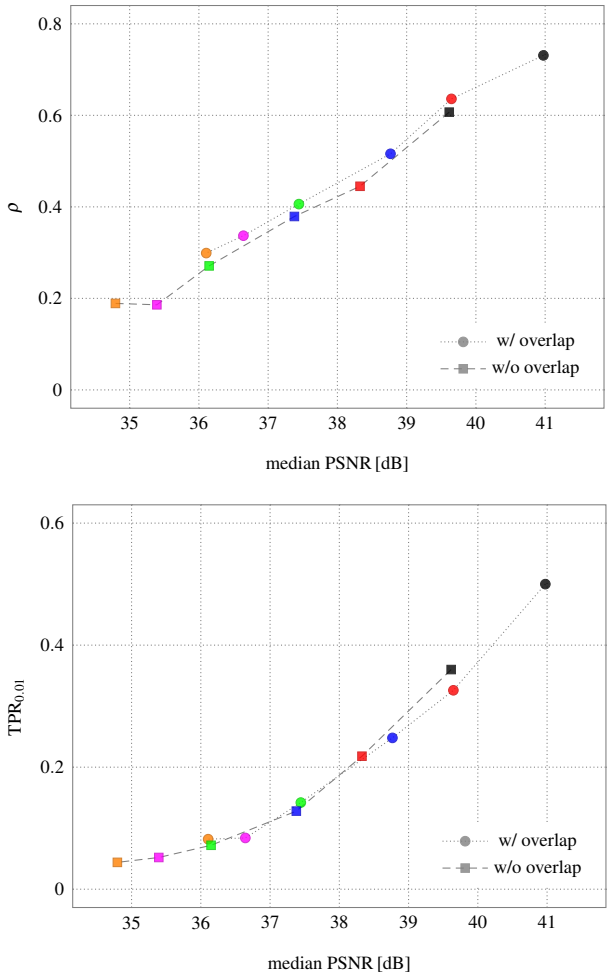

Figure 6. *Camera identification ρ and $\text{TPR}_{0.01}$ measures vs. median PSNR values after PCA-based patch replacement for different distortion settings τ with overlapping and non-overlapping patches. Symbol colors encode distortion settings (from $\tau = 50$ in orange to $\tau = 10$ in black) and match the colors of the ROC curves in Figure 5.*

approach yields satisfactory results for instance for combination $\tau = 20$, $\sigma = 0.75$. At a median PSNR of 37.8 dB (inter-quartile range 5 dB), ρ drops to 0.26, and we get $\text{TPR}_{0.01} = 0.07$. Only 3 % of the tested true positives have PCE values $s_c > 60$ (median 30.8).

Observe that in terms of image quality, the PatchMatch-based approach performs worse than the PCA-based approach. Comparing results for non-overlapping patches with $\tau = 30$, we find for instance that the former gives a median PSNR of 34 dB (inter-quartile range 3.8 dB), while the latter yields 36.1 dB (inter-quartile range 3 dB). The strength of the PatchMatch-based approach is the weighted averaging of overlapping patches, which boosts image quality to a median of 37.1 dB (inter-quartile range 4.2 dB) for $\sigma = 1$ while keeping the camera identification reliability reasonably low ($\rho = 0.24$ and $\text{TPR}_{0.01} = 0.06$). A view back at Figure 6 indicates that a comparable performance was not achievable with the PCA-based approach in our setup.

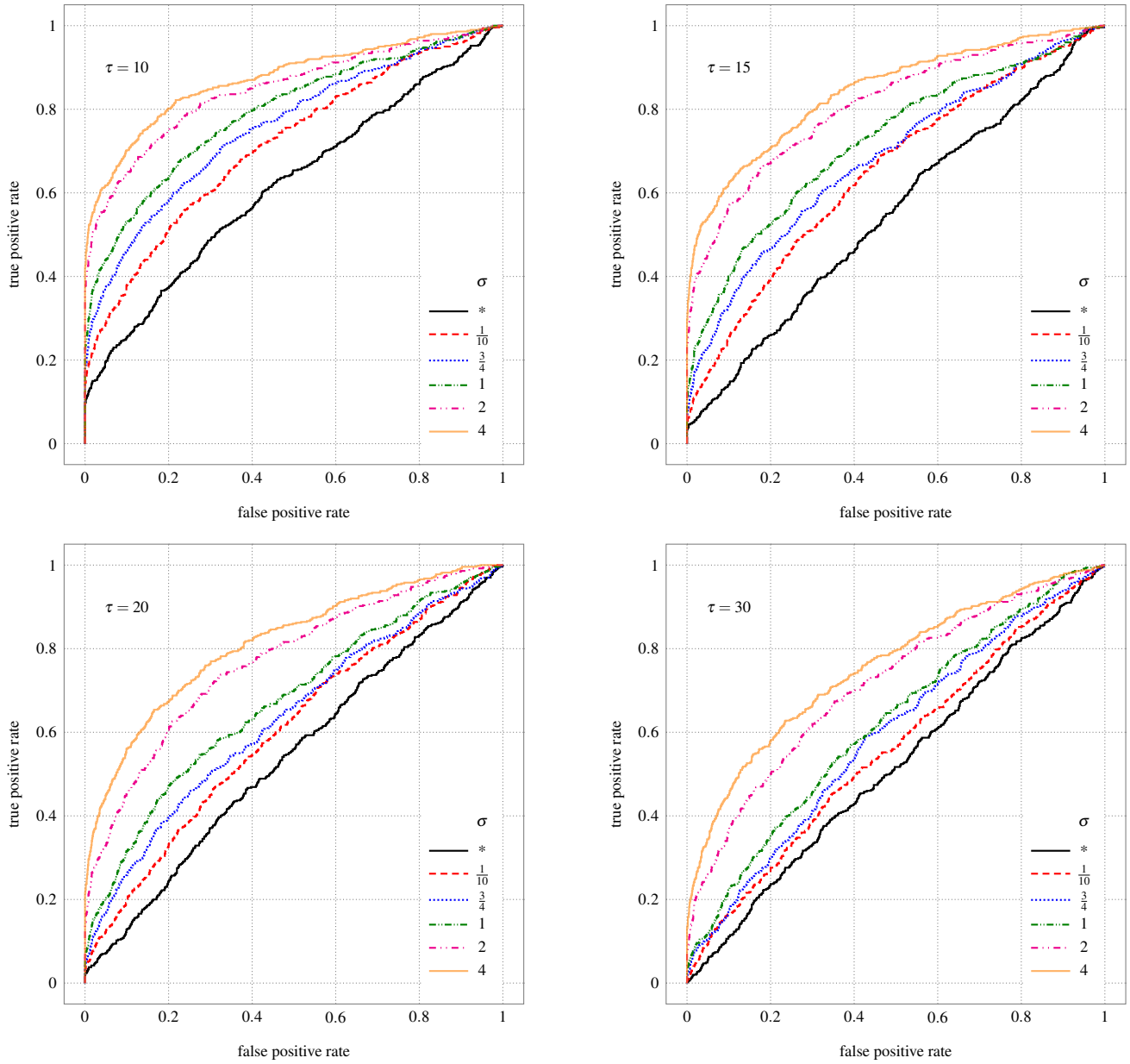

Figure 7. *Camera identification ROC curves after PatchMatch-based patch replacement ($\nu = 5$) for different distortion settings τ and Gaussian blurs σ. Black curves correspond to non-overlapping patches.*

Concluding Remarks

We have studied the application of patch replacement strategies to impede digital camera identification based on sensor noise. Inspired by similar attacks against robust watermarks, we have exploited the inherent self-similarity of digital images to replace small image patches with content from elsewhere in the image. The key to successful image anonymization is a careful tradeoff between image quality and distortion strength. Our results indicate that a PatchMatch-based heuristic is particularly promising. To the best of our knowledge, the attack discussed here is the first with sufficient potential to remove sensor fingerprints while maintaining a reasonably high image quality through simple image processing (i. e., without taking knowledge about the fingerprint into account) that does not rely on geometrical transformations of the image.

Future research will have to show to what degree an exhaustive local computation of fingerprint similarity can still establish camera identification (possibly driven by a content-based presegmentation of the image). The displacement field in Figure 2 suggests that PatchMatch's propagation property yields (relatively small) contiguous regions in highly textured areas also for stronger distortion constraints. A potential countermeasure is to randomize replacement patches through a k-nearest neighbors variant of PatchMatch [29]. In this context, it will also be interesting to see whether broadening the search space to different scales or affine transformations contributes to a better overall performance. Patch merging beyond simple weighted averaging is another direction

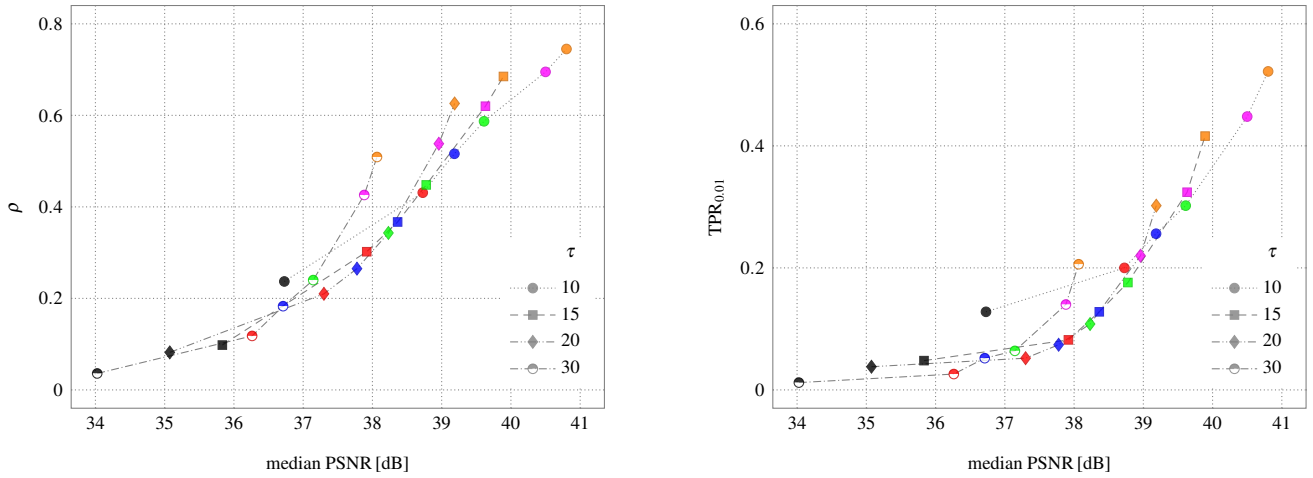

Figure 8. *Camera identification ρ and $TPR_{0.01}$ measures vs. median PSNR values after PatchMatch-based patch replacement ($\nu = 5$) for different distortion settings τ and Gaussian blurs σ. Symbols refer to distortion settings, symbol colors encode Gaussian blur strength (from non-overlapping blocks in black to $\sigma = 4$ in orange). Colors match the colors of ROC curves in Figure 7.*

worth looking into. Finally, we mention that the proposed approach naturally benefits from the abundance of candidate patches in large images. Future work will thus also have to investigate the influence of image size and content.

We close by emphasizing that the objective of this paper was not to make an image appear unsuspiciously authentic. Covering up the complete processing history of an image is generally a significantly more involved procedure. It is very likely that the proposed processing can be detected rather trivially. We believe that this is not a major concern when anonymization is the primary goal, for instance in the case of legitimate whistle-blowing. Along these lines, techniques that allow the photographer to prove to a third party that the image is authentic without revealing its source may be a welcome addition.

References

[1] J. Fridrich, "Sensor defects in digital image forensics," in *Digital Image Forensics: There is More to a Picture Than Meets the Eye*, H. T. Sencar and N. Memon, Eds. Springer, 2013, pp. 179–218.

[2] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Media Forensics and Security*, ser. Proceedings of SPIE, E. J. Delp, J. Dittmann, N. D. Memon, and P. W. Wong, Eds., vol. 7254, 2009, 72540I.

[3] S. Nagaraja, P. Schaffer, and D. Aouada, "Who clicks there!: anonymising the photographer in a camera saturated society," in *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2011, pp. 13–22.

[4] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics: There is More to a Picture Than Meets the Eye*, H. T. Sencar and N. Memon, Eds. Springer, 2013, pp. 327–366.

[5] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," 2010,

[6] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems," in *Information Hiding, Second International Workshop*, ser. Lecture Notes in Computer Science, D. Aucsmith, Ed., vol. 1525, 1998, pp. 219–239.

[7] M. Kirchner and R. Böhme, "Tamper hiding: Defeating image forensics," in *Information Hiding, 9th International Workshop*, ser. Lecture Notes in Computer Science, T. Furon, F. Cayre, G. Doërr, and P. Bas, Eds., vol. 4567, 2007, pp. 326–341.

[8] S. Bayram, H. T. Sencar, and N. Memon, "Seam-carving based anonymization against image and video source attribution," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 272–277.

[9] A. E. Dirik, H. T. Sencar, and N. Memon, "Analysis of seam-carving-based anonymization of images against PRNU noise pattern-based source attribution," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2277–2290, 2014.

[10] D. Kirovski and F. A. P. Petitcolas, "Blind pattern matching attack on watermarking systems," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1045–1053, 2003.

[11] G. Doërr, J.-L. Dugelay, and L. Grangé, "Exploiting self-similarities to defeat digital watermarking systems: a case study on still images," in *ACM Multimedia and Security Workshop (MM&Sec)*, 2004, pp. 133–142.

[12] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, 2009.

[13] M. Goljan and J. Fridrich, "Sensor-fingerprint based identification of images corrected for lens distortion," in *Media Watermarking, Security, and Forensics 2012*, ser. Proceedings of SPIE, N. Memon, A. M. Alattar, and E. J. Delp, Eds., vol. 8303, 2012, 83030.

[14] T. Gloe, S. Pfennig, and M. Kirchner, "Unexpected artefacts in PRNU-based camera identification: A 'Dresden Image Database' case-study," in *ACM Multimedia and Security Workshop*, 2012, pp. 109–114.

[15] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[16] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can we trust digital image forensics?" in *15th International Conference on Multimedia*, 2007, pp. 78–86.

[17] M. Goljan, J. Fridrich, and M. Chen, "Defending against fingerprint-

copy attack in sensor-based camera identification," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 227–236, 2011.

[18] E. Quiring and M. Kirchner, "Fragile sensor fingerprint camera identification," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.

[19] A. Karaküçük and A. E. Dirik, "Adaptive photo-response non-uniformity noise removal against image source attribution," *Digital Investigation*, vol. 12, pp. 66–76, 2015.

[20] A. Karaküçük, A. E. Dirik, H. T. Sencar, and N. D. Memon, "Recent advances in counter PRNU based source attribution and beyond," in *Media Watermarking, Security, and Forensics*, ser. Proceedings of SPIE, A. M. Alattar, N. D. Memon, and C. D. Heitzenrater, Eds., vol. 9409, 2015, 94090N.

[21] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 26, no. 3, 2007.

[22] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 977–984.

[23] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 349–356.

[24] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 18–30, 1992.

[25] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 1999, pp. 1033–1038.

[26] A. Buades, B. Coll, and J.-M. More, "A non-local algorithm for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 60–65.

[27] G. Doërr, J.-L. Dugelay, and D. Kirovski, "On the need for signal-coherent watermarks," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 896–904, 2006.

[28] T. Gloe and R. Böhme, "The Dresden Image Database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, pp. 150–159, 2010.

[29] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6313, 2011, pp. 29–43.

[30] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "PMBP: PatchMatch belief propagation for correspondence field estimation," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 2–13, 2014.

[31] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy–move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2284–2297, 2015.

[32] M. K. Mıhçak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of Wavelet coefficients," *IEEE Signal Processing Letters*, vol. 6, no. 12, pp. 300–303, 1999.

[33] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Information Hiding, 6th International Workshop*, ser. Lecture Notes in Computer Science, J. Fridrich, Ed., vol. 3200, 2004, pp. 67–81.