https://doi.org/10.2352/ISSN.2470-1173.2020.4.MWSF-077

Semi-Blind Image Resampling Factor Estimation for PRNU Computation

Morteza Darvish Morshedi Hosseini, Miroslav Goljan, and Hui Zeng, Department of ECE, SUNY Binghamton, NY, USA
{mgoljan, mdarvis1}@binghamton.edu

Abstract

Camera sensor fingerprints for digital camera forensics are formed by Photo-Response Non-Uniformity (PRNU), or more precisely, by estimating PRNU from a set of images taken with a camera. These images must be aligned with each other to establish sensor location pixel-to-pixel correspondence. If some of these images have been resized and cropped, the transformations need to be reversed. In this work we deal with estimation of resizing factor in the presence of one reference image from the same camera. For this problem we coin the term semi-blind estimation of resizing factor. We post two requirements that any solution of this problem should meet. It needs to be reasonably fast and exhibit very low estimation error. Our work shows that this problem can be solved using established image matching in Fourier-Mellin transform applied to vertical and horizontal projections of noise residuals (also called linear patterns).

Introduction

Photo-response non-uniformity (PRNU) is a result of imperfections in imaging sensors. It appears as offsets in measured luminance and they are different in every pixel of the sensor. This offset or bias is then passed through the signal and image processing pipeline, thus becoming the “fingerprint” of the sensor and consequently of the camera itself. Because PRNU almost does not change over time and does not repeat itself from one sensor to another, it is useful in many forensic applications including source camera verification. PRNU can be estimated from a single image or even a video frame, although accuracy of the estimate is low due to its very low signal-to-noise ratio. In order to increase SNR of the estimate, more than one image is typically used. PRNU estimation methods proceed by averaging noise obtained from a reference set of images or video frames [1, 2]. Averaging improves signal-to-noise ratio of the estimate since random noise tends to average out. All these “training” images must be aligned w.r.t. the imaging sensor. This requirement is easily met if the camera is at hand and all the training images can be taken with the same camera setting. Flat field shots, bright, yet without saturation, high resolution, and digital zoom turned off are recommended for the best result.

If the camera is not available and all training images for PRNU estimation are given to us, a robust procedure that guaranties geometrical alignment is needed. In this work, we address the case when some images in the training set are cropped and resized by an unknown factor and eventually JPEG compressed. We will refer to this case as *inconsistent training set problem*. This is not just an academic problem. For example, smart phone cameras often offer HDR image capturing mode. We have shown that many cameras slightly crop and upsample the images during HDR processing in order to deal with insufficient data near the image borders [3]. Recently, some camera models make HDR a default mode (such as LG G4).

Previously, resampling factor estimation has been studied mostly in two basic scenarios. The first, when the original image before transformation is available, the estimation in the more general context of scaling, rotation, and translation can be efficiently achieved using Fourier-Mellin transform [4] or Radon transform [5]. In the second, blind detection, i.e. without having the original image for matching procedure, relies on resampling artifacts in Fourier domain [6, 7, 8, 9]. Bianchi and Piva exploit near lattice distribution property of double JPEG images to estimate both the resizing factor and previous JPEG quantization parameters [10]. They first estimate the scaling factor before determining the original JPEG quality factor. This method requires the image to be already JPEG compressed before resizing. Their approach to scaling factor estimation has some similarities to ours. We discuss them in detail in the next section.

In the context of source camera identification, the problem of resized images was first addressed by Goljan *et al.* in 2008 [11]. Having the camera PRNU estimated, one can use it as a reference signal and launch a brute force search over a range of scaling factors while maximizing output of a correlation based detector. The method can find the scaling factor with very high precision but computational cost can be high. Furthermore, it requires a number of non-resized reference images for obtaining a sufficiently good PRNU estimate, also known as *camera fingerprint*, that are not always available. However, blind detection methods typically do not achieve such precision.

We introduce a *semi-blind detection and estimation* of image resizing. In this scenario, the original image is not known, yet one other non-resampled image from the same camera and the same imaging sensor is available for reference. Both images contain traces of the same PRNU, and sensor pattern noise (SPN) in general, that are utilized in a signal matching technique for detection of the resampling factor and for PRNU spatial alignment. JPEG compression artifacts known as “JPEG dimples” introduced to images by certain JPEG in-camera implementations are utilized as well if detected in the reference image. The proposed method is put under scrutiny during PRNU estimation from inconsistent training sets, where a portion of traning images is resized by unknown scaling factors and cropped.

In the next section, we introduce the semi-blind method of resampling detection and estimation that is based on pattern registration, describe the inputs for the matching process, recall the LP definition for grayscale images, and explain how JPEG dimples are utilized in this method. The following section details an algorithm for PRNU computation from images inconsistent in scaling. A large experimental section includes comparison with a blind method as a baseline. The paper is concluded in the last section.

Semi-blind scaling factor estimation

The problem setup includes (i) one image $\mathbf{I}$ (*probe image*) that has been uniformly resized by unknown factor γ starting from either a JPEG image produced by a digital camera (including those on mobile devices) or an uncompressed image in a raster format and (ii) one reference image $\mathbf{J}$ (*reference image*) from the same camera in the same resolution as was $\mathbf{I}$ before resizing. Image $\mathbf{I}$ could have been cropped as well. The goal is to estimate the factor γ and do it with such precision that allows for the camera PRNU estimation by averaging noise residuals of both images and of more such resized images $\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_n$,when such “scaling inconsistent” set of images serves for PRNU training.

We propose using Fourier-Mellin transform (originally described by Reddy and Chatterji [12]) for matching projections of sensor pattern noise in the resized image and the reference image from the same camera. Successful matching procedure outputs both relative scaling factor and translation vector. Direct application of this registration technique to two different images that contain the same camera SPN cannot work correctly because the image content overwhelms the common SPN. On the other hand, our attempts to perform the registration to the images noise residuals were successful for many image pairs but not satisfactory for the target application of PRNU estimation. Inspired by the resilience of the linear pattern (LP) in the blind method for rotation angle detection [13], we extract LP from images noise residuals and apply the above matching method to the two-dimensional representations of LPs. Linear patterns can be interpreted as horizontal and vertical projections that, functioning as directional low-pass filters, eliminate most random noise. This fact explains why the matching method developed for images with structured content can work for the weak SPN signal. After the scaling factor is determined, correlation test with the PRNUs from both images, after the estimated resizing is reversed, verifies whether or not the resampling factor is determined correctly.

Linear pattern computation

Representing a grayscale $m \times n$ image with $m \times n$ matrix $\mathbf{I}$, $\mathbf{I} \in \{0, \dots, 255\}^{m \times n}$, its noise residual is defined as

$$\mathbf{W}^{(0)} = \mathbf{I} - F(\mathbf{I}), \tag{1}$$

where F is a denoising filter. We use the wavelet-based Daubechies 8-tap denoising method described in [14].

Before computing linear pattern from the noise residual, normalization steps are applied to $\mathbf{W}^{(0)}$ to make it zero-mean, unit sample variance:

$\overline{\mathbf{W}} = \mathbf{W}^{(0)} - \overline{w}$, where $\overline{w} = \frac{1}{mn}\sum_{i,j=1}^{m,n} w_{ij}^{(0)}$ is the sample mean of $\mathbf{W}^{(0)}$.

$\mathbf{W} = \frac{\overline{\mathbf{W}}}{\sqrt{\frac{1}{mn}\sum_{i,j=1}^{m,n} \overline{w}_{ij}^2}}$, where the denominator is the sample standard deviation of $\overline{\mathbf{W}}$.

The (normalized) *linear pattern* of $\mathbf{W}$ is the $m \times n$ matrix $\mathbf{L} = \mathbf{L}(\mathbf{W})$ with elements

$$L_{ij} = r_i + c_j, \tag{2}$$

where r_i and c_j are the averages of the ith row and jth column of $\mathbf{W}$,

$$r_i = \frac{1}{n}\sum_{j=1}^{n} w_{ij}, \quad c_j = \frac{1}{m}\sum_{i=1}^{m} w_{ij}. \tag{3}$$

Once the choice of the filter F is fixed, we call $\mathbf{L}$ the *linear pattern of image* $\mathbf{I}$. Notice that the linear pattern of $\mathbf{L}$ is $\mathbf{L}$ itself [13].

From FFT-based image registration to LP-method

Image registration problem including determining scaling, rotation, and translation parameters of one transformed image with respect to its original copy can be efficiently solved by transforming the images into log-polar space of Fourier-Mellin transform. Phase correlation method is then applied in order to find translation in the log-polar space. This translation in vertical and horizontal directions translates to scaling and rotation of the image in spatial domain. This method was introduced by Reddy and Chatterji [12] in 1996. In this paper, it will be referred to as “FM registration”. While the transformed image is affected by interpolation and rounding noise, and possibly by JPEG compression, the method works under random noise attack as well.

In our semi-blind scenario, the common signal in the probe and in the reference image is the sensor SPN rather than the image content. The content of the images should be viewed as a special type of strong noise on top of the SPN. In order to make the above image registration method work, even when rotation is absent, the amount of noise has to be reduced. For this purpose, we apply the denoising filter (1) to both, image $\mathbf{I}$ and image $\mathbf{J}$, obtaining $\mathbf{W}_\mathbf{I}$ and $\mathbf{W}_\mathbf{J}$. This step alone can in some cases be sufficient to make matching of SPN and scaling factor estimation possible. Nevertheless, taking linear patterns of the two images, $\mathbf{L}(\mathbf{W}_\mathbf{I})$ and $\mathbf{L}(\mathbf{W}_\mathbf{J})$, yields much more desirable outcome, i.e. correctly estimating the transform parameters with higher probability. We determined that the high-pass emphasis filter (HPEF) (Formula (23) in [12]) is not required in our application. We stress that rotation is not considered in this work. The main reason is that the LP obtained from the probe image does not carry information about rotation. We refer to the previously mentioned work [13] that can be used to estimate the rotation angle with high precision (without the reference image).

LP-method*:*

1. Compute $\mathbf{W_I}$ and $\mathbf{W_J}$(according to Equation 1)
2. Compute $\mathbf{L}(\mathbf{W_I})$ and $\mathbf{L}(\mathbf{W_J})$ from images $\mathbf{I}$ and $\mathbf{J}$.
3. Estimate scaling factor γ and translation vector $[t_1, t_2]$ by running FM registration (with HPEF removed) on $\mathbf{L}(\mathbf{W_I})$ and $\mathbf{L}(\mathbf{W_J})$.

Here, we point out differences from the approach by Bianchi *et al.* [10]. Inspired by Kirchner e*t al.* [8], they apply the simplest high-frequency filters $g = \left[-\frac{1}{2}, 1, \frac{1}{2}\right]$ to image columns and rows. This step can be viewed as an attempt to remove the image content, which is what we do using filter $\mathbf{1} - F$. They only work with the magnitude of the filtered image. After forming row and column sums, they arrive at vectors $\mathbf{a}_h$ and $\mathbf{a}_v$, an analogy to our vectors $\mathbf{r}$ and $\mathbf{c}$. The next steps are different from ours since the method is blind, without any side information. They stay in one-dimension, taking magnitude of DFT of $\mathbf{a}_h$ and $\mathbf{a}_v$, inserting median filtering step, and computing the final vector $\tilde{\mathbf{f}}$ as the sum of horizontal and vertical components (only defined for square images). Further analysis of peaks in $\tilde{\mathbf{f}}$ introduced by the JPEG blockiness artifacts existing in the image before resizing allows computation of the likely scaling factor.

Implied from their definition, linear patterns capture biases in pixel intensity values in columns and rows of the images. These biases are transformed but not destroyed by image resizing. Another prominent bias was discovered in JPEG images by Agarwal and Farid [15]. JPEG dimples are positive or negative biases in one out of 64 pixels in every 8x8 block. They are caused by a “faulty” implementation of JPEG compression. Due to their repetitive character they can be easily detected. If JPEG dimples are detected in the reference image $\mathbf{J}$ they likely have been present in the probe image $\mathbf{I}$ before resizing as well. For this reason, we can expect that some combination of LP and dimples obtained from the reference image will make the registration procedure more reliable than using LP only. In the following, we determine the presence and type of JPEG dimples according to the method published in [15]. The dimples will be represented with binary matrix $\mathbf{D}$ of the same dimensions as the image, omitting the (content dependent) prominence map.

LPD-method*:*

1. Compute $\mathbf{W_I}$ and $\mathbf{W_J}$(according to Equation 1)
2. Compute $\mathbf{L}(\mathbf{W_J})$ from the reference image $\mathbf{J}$.
3. Detect dimples $\mathbf{D}$ from the reference image $\mathbf{J}$. If not detected, use LP-method.
4. Estimate scaling factor γ and translation vector $[t_1, t_2]$ by running FM registration (with HPEF removed) on $\mathbf{W_I}$ and $\mathbf{L}(\mathbf{W_J}) + \mathbf{D_J}$.

Notice that we do not work with JPEG dimples $\mathbf{D_I}$ from the probe image. Such dimples were introduced to the probe image after resizing operation. The JPEG dimples that had been present before resizing are displaced, which prevents us from localizing them before estimating the scaling factor γ. Thus, we work with entire $\mathbf{W_I}$ and a linear combination of LP and dimples from the reference image. Although a general linear combination of the form $\alpha\mathbf{L}(\mathbf{W_J}) + (1-\alpha)\mathbf{D_J}$ for other choices of α than $\frac{1}{2}$ may yield better performance in particular cases, our limited experiments suggested that FM registration is quite insensitive to α, $0 \leq \alpha \leq 0.8$ and the dependence varies from camera to camera and across reference images. Therefore, we set $\alpha = 0.5$ (see step 4) in all our experiments.

Algorithm 1

```
Input: $\mathbf{I}_i, i \in [1..n]$ ,all originated from the same source camera, "inconsistent in scaling."

1. Take $m$ random pairs of images $\mathbf{I}_i, \mathbf{I}_j,\ i \neq j$
2. Extract noise residuals $\mathbf{W}_i, \mathbf{W}_j$
3. $[u, v] = \arg\max_{i,j}\left(corr\left(\mathbf{W}_i, \mathbf{W}_j\right)\right)$
4. Form initial fingerprint $\mathbf{F} = \mathbf{W}_u + \mathbf{W}_v$
5. For the remaining $n-2$ images $\mathbf{I}_k$, $k \in [1..n]$, $k \neq i$, $k \neq j$

    • $[PCE_1, s_1] = PCE(\mathbf{F}, \mathbf{I}_k\mathbf{W}_k)$, where $s_1$ is a possible small spatial translation
    • Invert scaling for $\mathbf{I}_k$ and $\mathbf{W}_k$ using the proposed LPD-method $\longrightarrow \mathbf{I}'_k\mathbf{W}'_k$
    • $[PCE_2, s_2] = PCE\left(\mathbf{F}, \mathbf{I}'_k\mathbf{W}'_k\right)$, where $s_2$ is a possible spatial translation
    • If $PCE_2 > PCE_1$ and $PCE_2 > 20$
        – Correct $\mathbf{W}'_k$ for spatial translation and add it to $\mathbf{F}$.
    • Else if $PCE_1 > 20$
        – Add $\mathbf{W}_k$ to $\mathbf{F}$.
```

Estimation of PRNU from inconsistent sets of reference images

Having the semi-blind resizing factor estimation method developed, we present a straightforward algorithm for PRNU estimation from a set of images some of which may have been resized (Algorithm 1).

The number m should be chosen based on prior information about the minimum number of non-resized images in the set, and such that the probability of taking at least one pair of non-resized images is sufficiently large. Furthermore, the threshold of 20 guaranties skipping images that are very likely not correctly spatially synchronized with other images.

Experiments

For experimental evaluation of the proposed method we use the SDR_HAND subset of the dataset provided by a research group from the University of Florence [16] and expanded by test images from three phone cameras listed as A18-20 in Table 1. All these images are in JPEG format as produced by the cameras in their default settings. The total number of cameras represented in this dataset is 26. Table 1 lists all cameras in this dataset, JPEG quality factors QF_1 of all images (italic font emphasizes “closest” quality factors of custom quantization), image dimensions and detected presence of JPEG dimples.

Table 1. List of the cameras in "UNIFI HDR+3" dataset. JPEG quality factors in italic represent the standard qualities that are "closest" to the custom quantization tables. Presence of JPEG dimples in images from each camera is denoted in column 'Dimp'.

Camera name/model	QF_1	Resolution	Dimp
A01 Gionee S55	95	3120×4208	Yes
A02 Huawei P8	*94*	4160×3120	No
A03 Huawei P9	95	3264×1840	Yes
A04 Huawei P10	95	3968×2976	Yes
A05 Huawei Mate Pro10	*92*	3968×2976	Yes
A06 Huawei Y5	95	3264×2448	Yes
A07 Galaxy S7	96	4032×3024	No
A08 Galaxy S7	96	4032×2268	No
A09 Galaxy Note 5	*93−95*	5312×2988	No
A10 Galaxy J7	97	4128×3096	No
A11 Xiaomi 5	97	3456×4608	No
A12 Huawei RY6	*94*	2448×3264	No
A13 Huawei RY6	*94*	2448×3264	No
A14 Xiaomi 5A	97	4160×2340	No
A15 Xiaomi 3	97	4608×2592	No
A16 OnePlus 3t	95	4640×3480	No
A17 Asus Zenfone-2	90	3264×1836	No
A18 Xiaomi 3	97	4608×2592	No
A19 Xiaomi Note 4 #0	87	3120×4160	No
A20 Xiaomi Note 4 #1	87	3120×4160	Yes
I01 iPhone 8	*92*	3024×4032	No
I02 iPhone SE	*92,94*	4032×3024	Yes
I03 iPhone 7	*94*	4032×3024	Yes
I04 iPad Air	*95*	2592×1936	Yes
I05 iPhone 6	*92*	2448×3264	Yes
I06 iPhone 5S	*95*	3264×2448	Yes

For experiments involving controlled JPEG compression before image resizing, one hundred of uncompressed images from Bossbase [17] were used. Both proposed methods were implemented in Matlab. In particular, the Fourier-Mellin transform image registration method *imregcorr* was adopted with a small modification, in which the high frequency filter HPEF was removed for better performance in our settings. Since we have not attempted to estimate image rotations in this work we disabled unnecessary computations regarding the angle estimation in order to gain processing speed.

Experimental setup

Having around 20 images from each camera, the first 10 images serve as probe images and the rest as reference images (one *test pair* of the probe and the reference image in every single test). Wavelet denoising filter as in [1] is used to obtain noise residuals $\mathbf{W_I}$ and $\mathbf{W_J}$. Each probe image $\mathbf{I}_i$ is subject to resizing using bilinear interpolation method to produce image $\mathbf{I}_i(\gamma)$. The range of γ in our experiments is limited to $0.7 < \gamma < 1.3$. Particularly chosen scaling factors are 4th roots of {0.25, 0.4, 0.55, 0.7, 0.85, 0.95} symmetrically laid around 1, i.e. $\gamma \in$ {0.707, 0.795, 0.861, 0.915, 0.960, 0.987, 1, 1.013, 1.040, 1.085, 1.139, 1.205, 1.293} (rounded to three decimal points). This choice makes sure that more tests include scaling factors near 1 than far from 1. We have not attempted to test and optimize the method for much wider range of scaling factors. The next step in producing probe images is JPEG compression with standard quality factor varying it from 60 to 100, $QF_2 \in \{60, 65, 70, 75, 80, 85, 90, 92, 94, 96, 98, 100\}$. The option of no compression is included as well, denoted in tables as "NC". A small off-center image cropping by [10,30] pixels is applied at the end in order to introduce additional spatial shift between the test images and reference images, removing 40 rows and 40 pixels in total from each. The purpose of this cropping is to remove implicit information about scaling factor. Relative scaling between $\mathbf{I}_i(\gamma)$ and $\mathbf{J}$ is estimated using LP-method and LPD-method. This experiment is repeated for each camera in the dataset.

Performance of the proposed methods over a range of scaling factors

Our first test is designed to show how often the LPD-method correctly determines the scaling factor with high precision with respect to the camera model, true scaling factor, and JPEG compression factor QF_2. For a test pair, the success is reported if the estimation error is below precision level ϵ. The default precision in the tests is $\epsilon = 0.002$. Tables 2, 3, and 4 show the success rate w.r.t. scaling factor, averaged over all quality factors QF_2. These tables depict cameras that produced images with JPEG dimples. For these 11 cameras the tables show how the success rate changed as the method evolved from registration of noise residuals only ($\mathbf{W_I}$ with $\mathbf{W_J}$), LPs only in the LP-method and to registration of LP+dimples with $\mathbf{W_J}$ in the LPD-method. Success rates for the other 15 cameras can be found in Table 5. The plots in Figure 1 summarize how these three methods compare overall for all 26 cameras mixed together. Using JPEG dimples along with LP improves the performance of resampling estimation significantly whenever the test image before resizing was JPEG compressed by dimples introducing JPEG module.

The LPD method works surprisingly good for some cameras yet it mostly fails for certain camera models, namely Galaxy-S7, Huawei-RY6, and iPhone 8. Interestingly, recognizing non-resized images (scaling factor equal to 1) has the highest accuracy, except Huawei P8, for which upsizing is detected much better than downsizing. Assuming the camera model is known, for example from the metadata of the reference image, it is possible to make automated decision whether to launch the LP-method or any other alternative method for resizing estimation.

Note: It is not a mistake that the JPEG dimples were detected in images from Xiaomi Note 4 #1, yet not detected in those from Xiaomi Note 4 #0. EXIF headers of the images reveal that each camera features a different software that is responsible for image processing. The camera #0 (A19) has 'mido-user 7.0 NRD90M V9.6.2.0.NCFCNFD release-keys' software while the camera #1 (A20) has 'MediaTek Camera Application' software installed.

Robustness to JPEG compression

The previous tables do not tell us how the final JPEG compression of the resized images reduces the capability of

Table 2. Success rate (%) of the registration method using noise residuals only, averaged over all JPEG quality factors (precision $\epsilon = 0.002$).

W-only	Scaling Factor												
Camera name/model	0.71	0.8	0.86	0.91	0.96	0.99	1	1.01	1.04	1.09	1.14	1.2	1.29
A01 Gionee S55	74	85	86	95	99	100	100	95	99	98	100	99	100
A03 Huawei-P9	35	64	62	67	69	80	98	75	78	72	79	88	82
A04 Huawei-P10	46	82	85	89	92	90	100	94	95	99	99	100	100
A05 Huawei-Mate 10 Pro	99	100	100	100	100	100	100	100	100	100	100	100	100
A06 Huawei-Y5	14	18	21	37	27	52	100	56	30	35	61	72	51
A20 Xiaomi-Note 4 #1	40	62	55	81	82	79	100	78	89	77	75	87	94
I02 iPhone SE	35	43	54	65	61	64	100	65	68	72	72	70	72
I03 iPhone 7	45	57	65	65	68	70	99	72	73	78	76	69	78
I04 iPad-Air	1	0	1	0	1	0	81	3	0	3	0	1	8
I05 iPhone 6	18	18	31	28	38	55	98	57	48	45	56	68	58
I06 iPhone 5S	0	0	0	0	1	2	89	1	1	0	1	1	1

Table 3. Success rate (%) of the LP-method for listed cameras averaged over all JPEG quality factors (precision $\epsilon = 0.002$).

LP-Method	Scaling Factor												
Camera name/model	0.71	0.8	0.86	0.91	0.96	0.99	1	1.01	1.04	1.09	1.14	1.2	1.29
A01 Gionee S55	82	93	55	100	94	90	100	95	86	98	88	74	88
A03 Huawei-P9	45	98	64	100	98	100	100	32	15	81	67	92	79
A04 Huawei-P10	1	77	87	85	92	78	100	100	62	81	99	100	100
A05 Huawei-Mate 10 Pro	100	100	100	100	100	100	100	100	94	100	100	100	100
A06 Huawei-Y5	2	0	58	24	41	97	100	75	81	85	66	59	89
A20 Xiaomi-Note 4 #1	42	73	83	94	89	87	98	76	88	90	88	95	97
I02 iPhone SE	82	80	92	92	95	96	100	96	98	96	96	95	98
I03 iPhone 7	60	78	86	77	78	94	100	94	95	93	93	94	95
I04 iPad-Air	0	2	0	1	29	32	85	28	32	18	7	26	25
I05 iPhone 6	43	69	87	98	91	92	95	94	99	93	92	94	86
I06 iPhone 5S	0	22	2	8	34	8	99	63	63	67	45	65	62

Table 4. Success rate (%) of the LPD-method for listed cameras averaged over all JPEG quality factors (precision $\epsilon = 0.002$).

LPD-Method	Scaling Factor												
Camera name/model	0.71	0.8	0.86	0.91	0.96	0.99	1	1.01	1.04	1.09	1.14	1.2	1.29
A01 Gionee S55	98	99	99	100	100	100	100	100	100	100	100	100	100
A03 Huawei-P9	91	99	98	100	100	100	100	100	99	100	100	100	100
A04 Huawei-P10	29	80	86	88	93	79	100	100	65	97	100	100	100
A05 Huawei-Mate 10 Pro	100	100	100	100	100	100	100	100	100	100	100	100	100
A06 Huawei-Y5	68	89	97	100	96	100	100	100	100	100	100	100	100
A20 Xiaomi-Note 4 #1	86	98	100	100	100	100	100	100	100	100	100	99	100
I02 iPhone SE	91	92	94	95	95	98	100	97	100	99	98	98	100
I03 iPhone 7	68	90	96	97	96	98	100	98	98	99	98	99	100
I04 iPad-Air	4	25	21	32	48	53	100	56	29	21	28	68	67
I05 iPhone 6	77	95	99	100	100	100	100	94	93	94	100	88	100
I06 iPhone 5S	8	15	18	42	32	48	100	57	62	73	52	63	68

Table 5. Success rate (%) of LPD-method for listed cameras averaged over all JPEG quality factors (precision $\epsilon = 0.002$).

LP-Method	Scaling Factor												
Camera name/model	0.71	0.8	0.86	0.91	0.96	0.99	1	1.01	1.04	1.09	1.14	1.2	1.29
A02 Huawei P8	13	61	3	0	3	2	58	72	69	66	64	74	68
A07 Galaxy-S7	12	7	8	12	14	15	83	36	22	39	30	35	42
A08 Galaxy-S7	2	2	0	10	2	8	29	3	2	4	0	0	0
A09 Galaxy-Note 5	27	51	37	55	60	74	89	78	60	68	56	67	49
A10 Galaxy-J7	62	65	69	69	78	79	100	54	42	68	68	58	52
A11 Xiaomi 5	55	78	88	72	63	82	99	86	82	83	82	82	83
A12 Huawei-RY6	0	0	0	0	0	0	46	22	15	23	35	10	28
A13 Huawei-RY6	0	2	0	0	0	0	52	36	5	18	36	5	37
A14 Xiaomi-5A	37	29	17	39	45	33	91	69	65	70	72	71	48
A15 Xiaomi-3	76	100	100	97	100	87	99	75	89	89	75	85	58
A16 OnePlus-3t	68	82	83	85	82	75	95	76	87	69	68	67	65
A17 Asus Zenfone-2	82	97	93	91	100	98	100	100	100	95	99	89	57
A18 Xiaomi-3	88	100	91	90	92	99	100	91	98	82	83	88	75
A19 Xiaomi-Note4 #0	94	98	100	100	100	100	100	78	99	100	95	100	100
I01 iPhone 8	0	8	4	5	5	17	85	21	6	5	3	1	1

the LPD-method. This is best depicted in Figure 2. Here, the rates are averaged over all scaling factors from 0.7 to 1.3. The performance drops very slowly with compression quality with two exceptions, iPad Air and iPhone 5S, for which compression quality has a stronger impact on the method performance. More detailed success rates for images from just two cameras are shown in Figure 3, where it is clearly noticeable that the LPD-method starts breaking down mostly at the low end of scaling and JPEG quality factors.

Histograms of estimation errors in Figure 4 provide some insight into the precision LPD-method achieves. Errors smaller than 0.001 make 66% when images were downsized in the range of scaling factors $0.7 < \gamma < 1$. This number increases to 71% when images were upsized ($1 < \gamma < 1.3$). These histograms also suggest that relaxing the precision threshold from 0.002 to 0.003 would increase success rates in the previous tables only negligibly.

The role of prior JPEG compression

Intuitively, the stronger the LP in the image prior to resizing, the better the registration method should work. Because JPEG compression has a "side effect" in creating blockiness, one can expect that it might add to the LP strength. On the other hand, the LP itself changes with JPEG compression. It is therefore interesting to see how the JPEG compression applied prior to the image resizing effects the proposed LPD-method. In order to gain control over the level of compression we start with a set of 100 uncompressed images from Bossbase. The images are first JPEG compressed with one of the quality factors QF_1 and then resized. The range for QF_1 is the same as for QF_2 before. No second compression follows this time. Each such test image is paired with a fixed reference image ('1339.tif') for running the LPD-method. The range of scaling factors remains within the interval [0.7 1.3]. The test results plotted in Figure 5 show the pattern. Best success rates are for quality factor QF_1 between 90 and 94, decreasing slowly as QF_1 decreases. High quality $QF_1 > 95$ means lower success rate, up to 16 percentage points for $QF_1 = 100$. If we require twice higher precision, $\epsilon = 0.001$, the success rates drop about 7 percentage points regardless of the quality factor of the prior JPEG compression.

Comparison of the LPD method with the blind method

Although blind methods for estimating resizing factors have disadvantage of having no access to side information about the source camera and the images it produces, we believe it is worth it to show superiority of the proposed semi-blind method to a blind method. For simplicity of its implementation we chose the blind method proposed by Kirchner in 2008 [6]. In our test, limitations of the blind method are leveraged. Since it cannot differentiate between downsizing and upsizing we ran separate tests for downsized and upsized images. Non-scaled images (scaling factor equal to 1) were not subject of this test.

Test setup: 100 uncompressed images from Bossbase, original resolution images, all from one camera. The reference image is fixed as '1339.tif'.

Parameters: Precision $\epsilon = 0.002$.

The plotted results in Figure 6 show that the blind method cannot compare to the LPD-method at least at the same precision level. The performance of the blind method varies wildly with the scaling factor that needs to be estimated.

Computation of PRNU from inconsistent set of reference images

The purpose of the last experiment is to test Algorithm 1. The scaling inconsistent sets are formed by 10 images in JPEG format "right from the camera" and other 10 (or less when 20 images are not available in the SDR subset) images resized by randomly generated scaling factor λ, $0.7 \leq \lambda \leq 1.3$

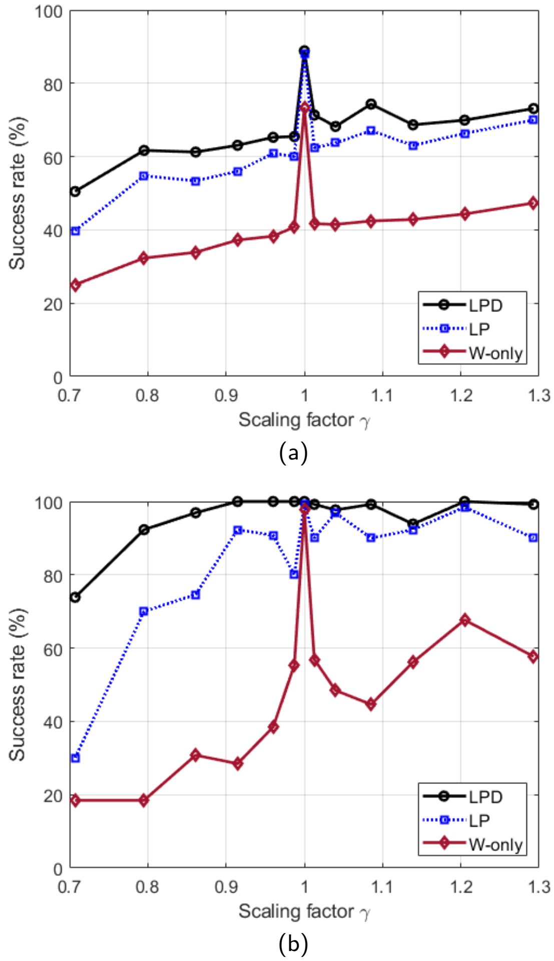
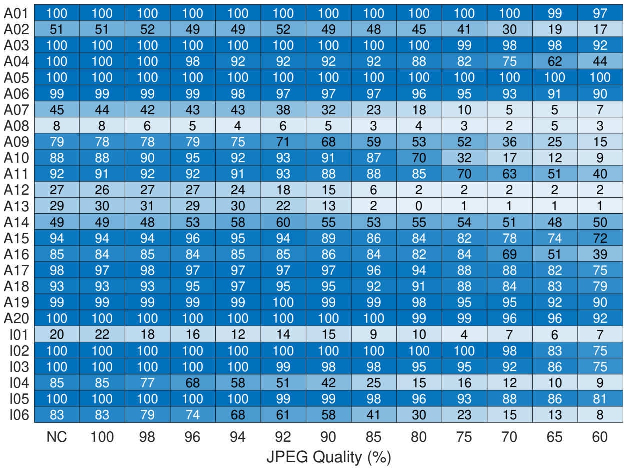
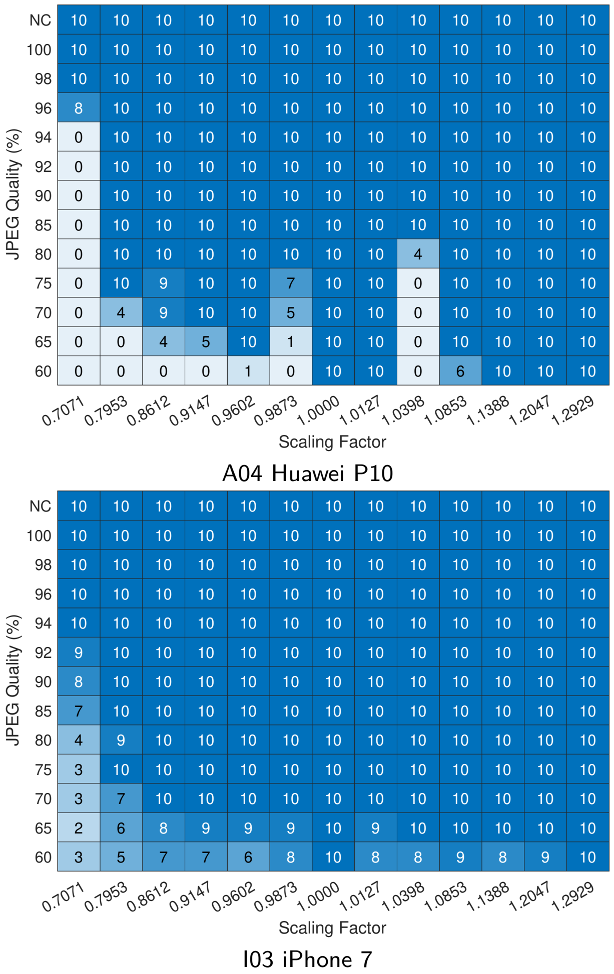


Figure 1. *Success rates of the two proposed methods in comparison to scaling factor estimation by direct registration of noise residuals (W-only). All 26 cameras (a), iPhone 6 (b).*



	NC	100	98	96	94	92	90	85	80	75	70	65	60
A01	100	100	100	100	100	100	100	100	100	100	100	99	97
A02	51	51	52	49	49	52	49	48	45	41	30	19	17
A03	100	100	100	100	100	100	100	100	100	99	98	98	92
A04	100	100	100	98	92	92	92	92	88	82	75	62	44
A05	100	100	100	100	100	100	100	100	100	100	100	100	100
A06	99	99	99	99	98	97	97	97	96	95	93	91	90
A07	45	44	42	43	43	38	32	23	18	10	5	5	7
A08	8	8	6	5	4	6	5	3	4	3	2	5	3
A09	79	78	78	79	75	71	68	59	53	52	36	25	15
A10	88	88	90	95	92	93	91	87	70	32	17	12	9
A11	92	91	92	92	91	93	88	88	85	70	63	51	40
A12	27	26	27	27	24	18	15	6	2	2	2	2	2
A13	29	30	31	29	30	22	13	2	0	1	1	1	1
A14	49	49	48	53	58	60	55	53	55	54	51	48	50
A15	94	94	94	96	95	94	89	86	84	82	78	74	72
A16	85	84	85	84	85	85	86	84	82	84	69	51	39
A17	98	97	98	97	97	97	97	96	94	88	88	82	75
A18	93	93	93	95	97	95	95	92	91	88	84	83	79
A19	99	99	99	99	99	100	99	99	98	95	95	92	90
A20	100	100	100	100	100	100	100	100	99	99	96	96	92
I01	20	22	18	16	12	14	15	9	10	4	7	6	7
I02	100	100	100	100	100	100	100	100	100	100	98	83	75
I03	100	100	100	100	100	99	98	98	95	95	92	86	75
I04	85	85	77	68	58	51	42	25	15	16	12	10	9
I05	100	100	100	100	100	99	99	98	96	93	88	86	81
I06	83	83	79	74	68	61	58	41	30	23	15	13	8

Figure 2. *Success rate (%) of the LPD-method vs. JPEG quality factor QF_2 averaged over all scaling factors (precision $\epsilon = 0.002$).*



JPEG Quality (%)	0.7071	0.7953	0.8612	0.9147	0.9602	0.9873	1.0000	1.0127	1.0398	1.0853	1.1388	1.2047	1.2929
NC	10	10	10	10	10	10	10	10	10	10	10	10	10
100	10	10	10	10	10	10	10	10	10	10	10	10	10
98	10	10	10	10	10	10	10	10	10	10	10	10	10
96	8	10	10	10	10	10	10	10	10	10	10	10	10
94	0	10	10	10	10	10	10	10	10	10	10	10	10
92	0	10	10	10	10	10	10	10	10	10	10	10	10
90	0	10	10	10	10	10	10	10	10	10	10	10	10
85	0	10	10	10	10	10	10	10	10	10	10	10	10
80	0	10	10	10	10	10	10	10	4	10	10	10	10
75	0	10	9	10	10	7	10	10	0	10	10	10	10
70	0	4	9	10	10	5	10	10	0	10	10	10	10
65	0	0	4	5	10	1	10	10	0	10	10	10	10
60	0	0	0	0	1	0	10	10	0	6	10	10	10

JPEG Quality (%)	0.7071	0.7953	0.8612	0.9147	0.9602	0.9873	1.0000	1.0127	1.0398	1.0853	1.1388	1.2047	1.2929
NC	10	10	10	10	10	10	10	10	10	10	10	10	10
100	10	10	10	10	10	10	10	10	10	10	10	10	10
98	10	10	10	10	10	10	10	10	10	10	10	10	10
96	10	10	10	10	10	10	10	10	10	10	10	10	10
94	10	10	10	10	10	10	10	10	10	10	10	10	10
92	9	10	10	10	10	10	10	10	10	10	10	10	10
90	8	10	10	10	10	10	10	10	10	10	10	10	10
85	7	10	10	10	10	10	10	10	10	10	10	10	10
80	4	9	10	10	10	10	10	10	10	10	10	10	10
75	3	10	10	10	10	10	10	10	10	10	10	10	10
70	3	7	10	10	10	10	10	10	10	10	10	10	10
65	2	6	8	9	9	9	10	9	10	10	10	10	10
60	3	5	7	7	6	8	10	8	8	9	8	9	10

Figure 3. *Correctly-estimated scaling factors out of 10 tests for each scaling and JPEG quality factor. Tables shown for two representative cameras, Huawei P10 and iPhone 7 (precision $\epsilon = 0.002$).*

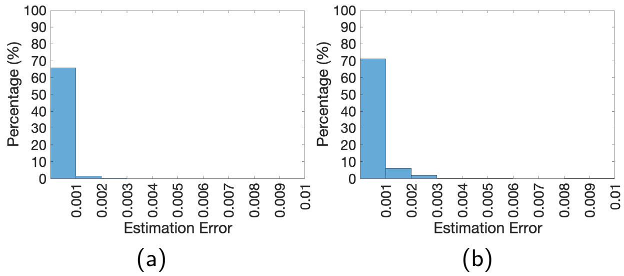


Figure 4. *Histogram of absolute estimation errors for 10×26 test images from 26 cameras, JPEG quality factors QF_2ranging from 80 to 100: (a) for errors collected over the range of scaling factors $0.7 < \gamma < 1$ and (b) for errors collected over the range of scaling factors $1 < \gamma < 1.3$ (The histogram plots are clipped at 0.01).*

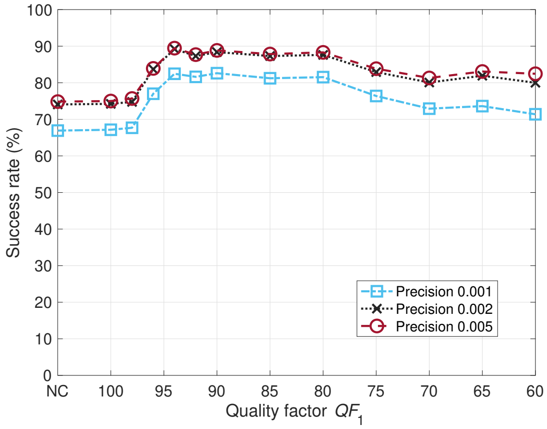


Figure 5. *Success rates of the LPD-method evaluated at three precision levels, $\epsilon = 0.001, 0.02, 0.005$, from tests with 100 images from Bossbase. Each data point reflects 10 tests for each of 13 scaling factors from range [0.7 1.3], i.e. 130 tests.*

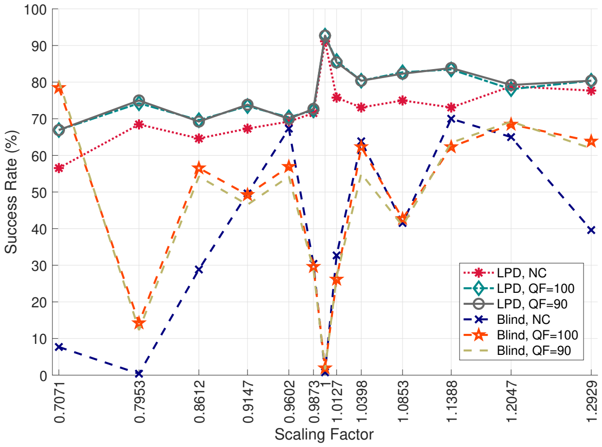


Figure 6. *Success rate of the LPD-method compared to the blind method [6] at $QF_1 = 90, 100$ and no prior compression.*

(same for every camera). Such sets are prepared for each camera listed in Table 1. For comparison, we run the direct form of PRNU estimation using maximum likelihood (ML) formula as in [1] that is not aware of scaling performed to some images.

For a relative measure of fingerprint quality we take one flat image from the same camera and run camera identification test for this extra image with each computed fingerprint. The higher PCE ([18]) indicates the higher quality PRNU estimation. Alternatively, the larger correlation coefficient can also point to the better fingerprint quality. The measured PCEs for both algorithms and their ratio are shown in Table 6. In order to reduce a potential bias that the extra image might have on the results, the gain ratio is averaged over three pairs of PCE measurements obtained from tests with three extra images.

The gain obtained from Algorithm 1 is significant. It increases even more as the portion of resized images in the training gets larger (not shown in this table). Interestingly, what appears to be the largest gain seen among the 26 cameras for cameras Huawei P8, Huawei RY6, and One Plus 3t, cannot be fully attributed to the scaling correction. Some of the gain Algorithm 1 provides is due to the small proper spatial translation correction unexpectedly needed for some images from these cameras.

Table 6. Improvement in PRNU estimation through Algorithm 1. PCE in the middle columns is computed using one flat image. The gain in fingerprint quality is due to the LPD-method. The gain ratio is computed as the average ratio $PCE_{\mathrm{LPD}}/PCE_{\mathrm{ML}}$ for three flat images entering the camera identification test.

Camera name/model	PCE (ML)	PCE (LPD)	Gain ratio
A01 Gionee S55	1459	3366	2.253
A02 Huawei P8	469	2496	5.589
A03 Huawei P9	303	899	2.948
A04 Huawei P10	3833	5511	1.437
A05 Huawei Mate Pro10	5319	13695	2.594
A06 Huawei Y5	19859	47440	2.292
A07 Galaxy S7	879	2080	2.336
A08 Galaxy S7	5843	13723	2.383
A09 Galaxy Note 5	7114	19187	2.622
A10 Galaxy J7	14663	31227	2.108
A11 Xiaomi 5	2423	5111	2.114
A12 Huawei RY6	1480	5922	4.098
A13 Huawei RY6	1680	7022	4.089
A14 Xiaomi 5A	12017	18463	1.550
A15 Xiaomi 3	6206	12518	2.029
A16 OnePlus 3t	418	1634	4.763
A17 Asus Zenfone-2	8413	16901	1.948
A18 Xiaomi 3	21474	37841	1.751
A19 Xiaomi Note 4 #0	3446	7652	2.128
A20 Xiaomi Note 4 #1	452	1315	2.788
I01 iPhone 8	1838	2515	1.350
I02 iPhone SE	13684	29170	2.155
I03 iPhone 7	432	1111	2.379
I04 iPad Air	2732	4864	1.762
I05 iPhone 6	7667	17986	2.360
I06 iPhone 5S	1322	2733	1.831

Conclusions

Image resizing detection and estimation has mostly been studied in "blind" scenario, where no side information is available. We introduced "semi-blind" scenario, in which one other image from the same digital camera is provided as a reference. We show that vertical and horizontal projections (linear patterns) of sensor pattern noise derived from the reference image can be used in an established image registration procedure to determine scaling factor and translation with high precision. This method is applicable to most images from mobile devices that have later been resized and JPEG compressed again. Presence of any JPEG compression is not a necessary condition for the proposed method to work. On the other hand, if the resized image had been JPEG compressed, which is a typical practice, the performance of the proposed method increases. Moreover, a presence of so-called JPEG dimples caused by prior JPEG compression is accounted for to boost the probability of successful estimation at high precision. The proposed method is shown to improve PRNU estimation from training sets consisting of scaled images, without adding much computational complexity.

The proposed method has its limitations. It may not work for cameras that produce images with very weak or random linear patterns. Future efforts in the semi-blind scenario will include the case when rotation of the image is added on top of scaling and translation.

Acknowledgments

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

The authors would like to thank Hany Farid for providing Matlab code for detecting JPEG dimples.

Author Biography

Morteza Darvish Morshedi Hosseini is a PhD candidate in Electrical and Computer Engineering at Binghamton University. Morteza received his M.S. in Information Technology Engineering with a minor in Information Security from University of Isfahan in 2016. His research is centered around practical problems in the area of digital media forensics.

Miroslav Goljan received the Ph.D. degree in Electrical Engineering from Binghamton University in 2002 and the M.S. in Mathematical Informatics from Charles University in Prague, Czech Republic, in 1984. He is a Research Scientist at the Dept. of Electrical and Computer Engineering at Binghamton University, Binghamton, NY. His research focuses on digital image and digital camera forensics, steganography, steganalysis, and reversible data hiding in digital media.

Hui Zeng received the B.E. and M.S. degrees from Nanjing University of Posts and Telecommunications (NUPT) in 2004 and 2007 respectively, and the Ph.D. degree from Sun Yat-sen University in 2016. He is currently an Associate Professor with the Southwest University of Science and Technology, and a visiting scholar at Binghamton University. His research interests include multimedia forensics and game theory.

References

[1] M. Goljan, T. Filler, and J. Fridrich, “Large scale test of sensor fingerprint camera identification,” in *Proc. SPIE, Electronic Imaging, Media Forensics and Security XI* (N. Memon, E. Delp, P. Wong, and J. Dittmann, eds.), vol. 7254, (San Jose, CA), pp. 0I 01–12, January 19–21, 2009.

[2] C.-T. Li and Y. Li, “Color-decoupled photo response non-uniformity for digital image forensics,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 260–271, Feb 2012.

[3] M. Darvish Morshedi Hosseini and M. Goljan, “Camera identification from HDR images,” in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec'19, (New York, NY, USA), p. 6976, Association for Computing Machinery, 2019.

[4] B. Dasgupta and B. N. Chatterji, “Fourier-Mellin transform based image matching algorithm,” *IETE Journal of Research*, vol. 42, no. 1, pp. 3–9, 1996.

[5] J. You, W. Lu, J. Li, G. Gindi, and Z. Liang, “Image matching for translation, rotation and uniform scaling by the Radon transform,” in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, vol. 1, pp. 847–851, Oct 1998.

[6] M. Kirchner, “Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue,” in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, MM&Sec '08, (New York, NY, USA), pp. 11–20, ACM, 2008.

[7] L. Li, J. Xue, Z. Tian, and N. Zheng, “Moment feature based forensic detection of resampled digital images,” in *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, (New York, NY, USA), pp. 569–572, ACM, 2013.

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

[9] S. Pfennig and M. Kirchner, “Spectral methods to determine the exact scaling factor of resampled digital images,” in *International Symposium on Communications Control and Signal Processing, ISCCSP 2012*, 2012.

[10] T. Bianchi and A. Piva, “Reverse engineering of double JPEG compression in the presence of image resizing,” in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 127–132, Dec 2012.

[11] M. Goljan and J. Fridrich, “Camera identification from scaled and cropped images,” in *Proceeedings of SPIE Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X* (E. Delp, P. Wong, J. Dittmann, and N. Memon, eds.), vol. 6819, (San Jose, CA), pp. 0E 1–13, January 28–30, 2008.

[12] B. S. Reddy and B. N. Chatterji, “An FFT-based technique for translation, rotation, and scale-invariant image registration,” *IEEE Transactions on Image Processing*, vol. 5, pp. 1266–1271, Aug 1996.

[13] M. Goljan, “Blind detection of image rotation and angle estimation,” in *Media Watermarking, Security, and Forensics 2018, Burlingame, CA, USA, 28 January 2018 - 1 February 2018*, 2018.

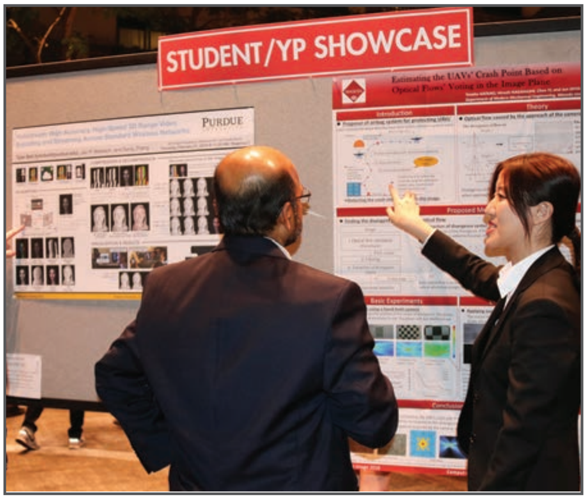
[14] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, “Low-complexity image denoising based on statistical modeling of wavelet coefficients,” *IEEE Signal Processing Letters*, vol. 6, no. 12, pp. 300–303, 1999.

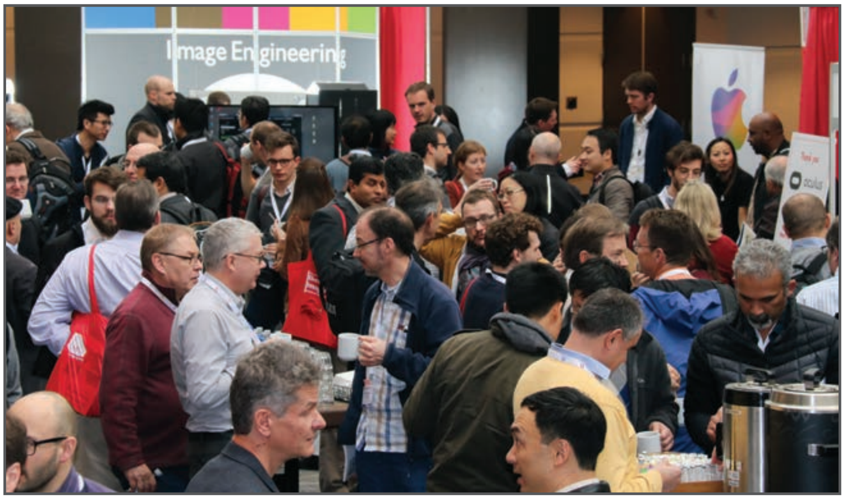
[15] S. Agarwal and H. Farid, “Photo forensics from JPEG dimples,” in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, Dec 2017.

[16] O. A. Shaya, P. Yang, R. Ni, Y. Zhao, and A. Piva, “A new dataset for source identification of high dynamic range images,” *Sensors*, vol. 18, no. 11, 2018.

[17] P. Bas, T. Filler, and T. Pevný, “"Break our steganographic system": The ins and outs of organizing BOSS,” in *International workshop on information hiding*, pp. 59–70, Springer, 2011.

[18] M. Goljan, J. Fridrich, and T. Filler, “Large scale test of sensor fingerprint camera identification,” in *Proceedings of SPIE, Electronic Imaging, Media Forensics and Security XI* (N. Memon, E. Delp, P. Wong, and J. Dittmann, eds.), vol. 7254, (San Jose, CA), pp. 0I 1–12, SPIE, January 2009.





- AR is a fundamental and transformative interface and platform.
- 20+ years from my initial working demo to today's AR displays
- Optimistic that it won't take another 20 to reach ubiquitous consumer AR!