Camera Fingerprint Extraction via Spatial Domain Averaged Frames

Samet Taspinar, Manoranjan Mohanty, and Nasir Memon

***Abstract*—Photo Response Non-Uniformity (PRNU) based camera attribution is an effective method to determine the source camera of a visual object (an image or a video). To apply this method, images or videos need to be obtained from a camera to create a "camera fingerprint" which then can be compared against the PRNU of the query media whose origin is under question. The fingerprint extraction process can be time consuming when a large number of video frames or images have to be denoised. This may need to be done when the individual images have been subjected to high compression or other geometric processing such as video stabilization. This paper investigates a simple, yet effective and efficient technique to create a camera fingerprint when so many still images need to be denoised. The technique utilizes Spatial Domain Averaged (SDA) frames. An SDA-frame is the arithmetic mean of multiple still images. When it is used for fingerprint extraction, the number of denoising operations can be significantly decreased with little or no performance loss. Experimental results show that the proposed method can work more than 50 times faster than conventional methods while providing similar matching results.**

***Index Terms*—PRNU, video forensics, camera fingerprint extraction, image forensics.**

I. Introduction

Photo Response Non-Uniformity (PRNU) based source camera attribution is a well-studied and successful method in media forensics for finding the source camera of an anonymous image or video [1]. The method is based on the unique PRNU noise of a camera sensor array stemming from manufacturing imperfections. This PRNU noise can act as a camera fingerprint. The PRNU approach is often used in two scenarios: camera verification and camera identification. Camera verification aims to establish if a suspect camera takes a given query image or a video. This is done by correlating the noise estimated from the query image or video with the fingerprint of the camera. The camera fingerprint is usually computed by taking pictures from the camera under controlled conditions. For camera identification, the potential source camera of the query image or video is determined from a large database of camera fingerprints. One can view camera identification as essentially the same as performing n camera verification tasks where n is the number of camera fingerprints in the database. However, when performing identification, it is assumed that the camera fingerprints are pre-computed.

In both verification and identification, it is often the case that there is no camera available to create fingerprints under controlled conditions. Instead, camera fingerprints are estimated from a set of publicly available media assumed to be from the same camera. Such media can have a very diverse range of quality and content and often lacks metadata.

For efficient fingerprint matching in large databases, various approaches have been proposed. Fridrich et al. [2] proposed the use of fingerprint digests in which a subset of fingerprint elements having the highest sensitivity are used instead of the entire fingerprint. Bayram *et al.* [3] introduced binarization where each fingerprint element is represented by a single bit. Valsesia *et al.* [4] proposed the idea of applying random projections to reduce the fingerprint dimension. Bayram et. al. [5] introduced group testing via composite fingerprint that focuses on decreasing the number of correlations rather than decreasing the size (storage) of a fingerprint. Recently, Taspinar et al. [6] proposed a hybrid approach that utilizes both decreasing the size of a fingerprint and the number of correlations. All these methods were designed and tested for images, however, they can also be used for videos.

Although the image-centric PRNU-based method can be extended to video [7]–[9], source camera attribution with video presents several new challenges. First, a video frame is much more compressed than a typical image. Therefore, the PRNU signal extracted from a video frame is of significantly lower quality than one obtained from an image. As a result, a more significant number of video frames are required to compute the fingerprint. Chuang et. al. [7] found that it is best to use all the frames instead of using only the I- or P-frames to compute a fingerprint. Using a large number of frames can introduce significant computation overhead. For example, calculating a fingerprint from 60 I-frames of a one-minute HD video requires one to two minutes, whereas 30 to 40 minutes is required if all frames are used.

In the case of camera identification, the amount of computation can be prohibitive in practical scenarios. For example, for computing fingerprints from a thousand one-minute Full HD videos (i.e., ≈ 1800 frames each) using a PC may take more than $3-4$ days when all resources are used. Clearly, with billions of visual objects uploaded every day on the Internet, large scale camera source identification becomes quickly infeasible. Although camera fingerprints stored in a database may have to be computed just once by a system, computing a fingerprint estimate at run-time from a query video can be prohibitive when faced with a reasonable number of query videos presented to the camera identification system in a day.

Besides source camera identification, digital stabilization operations performed within modern cameras also present a

Samet Taspinar (st89@nyu.edu) is with Center for Cyber Security, New York University, New York, USA, Manoranjan Mohanty (manoranjan.mohanty@uts.edu.au) is with Center for Forensic Science, UTS, Australia, and Nasir Memon (memon@nyu.edu) is with Department of Computer Science and Engineering, New York University, New York, USA.

significant challenge for PRNU-based source camera verification for video [8], [10], [11]. Video stabilization results in sensor-pixel misalignments between individual frames of the video as the geometric transformations performed to compensate for camera motion and spatially align each frame are different. An accurate camera fingerprint cannot be obtained using misaligned frames as is done with non-stabilized video even if the video quality is very high. Although some preliminary methods address source camera verification for stabilized video, [8], [10], these methods are either limited in scope or have low performance (low true positive rate) and high computation overhead. An alternate approach to address the stabilization issue for a fairly long video (at least a couple of minutes) [12] is to use a large number of frames for computing the fingerprint. The idea is that with a large number of frames, there will be a sufficient number of aligned pixels at each spatial location that can result in the computation of an accurate fingerprint. As discussed above, this approach, however, can again introduce high computation overhead unsuitable for practical use.

As a third example, modern devices such as smartphones capture different types of media with different resolutions. For example, most cameras don't use the full sensor resolution when capturing a video and downsize the sensor output to a lower resolution by proprietary and often unknown in-camera processing techniques. For such a challenging task, PRNU based source camera matching may often fail if only I-frames are used.

This paper proposes a computationally efficient way to compute a camera fingerprint from a large number of visual objects, such as individual frames of a video or highly compressed images taken from a social media platform. In contrast to the three-step conventional fingerprint computation method (which first estimates PRNU noise from each frame using a denoising filter and then averages several estimated individual PRNU noise estimates to get a reliable fingerprint estimate. Finally applies a post-processing step to reduce the non-unique artefacts [13], [14]), the proposed method uses a four step approach: frame averaging, denoising, noise averaging and post-processing. The frame averaging step gets the arithmetic mean of the frames in the spatial domain, resulting in *Spatial Domain Averaged frames (SDA-frames)* (Fig. 2). Then, in the second step, each SDA-frame is denoised, and an averaging of the estimated PRNU noise is done to arrive at the fingerprint estimate. A post-processing step is applied to the fingerprint estimate to remove non-unique artifacts such as the JPEG blockiness artifact. These post-processing steps are normalizing the PRNU noise with the zero-mean operation and applying Wiener filtering [13], [14]. The goal here is to minimize the number of denoising operations (as denoising is the most expensive step) and also get rid of scene dependent noise by averaging multiple frames. Experiments with VISION dataset [15] and NYUAD-MMD [16] show that the proposed method provides significant speedup in computing fingerprints. It achieves a significantly higher true positive rate than a fingerprint computed by I-frames only and much lower computation cost than a fingerprint obtained from all available frames while yielding similar performance.

The rest of the paper has been organized as follows. Section II summarizes the PRNU-based method and provides an overview of how digital video stabilization works. Section III explains the proposed fingerprint extraction method using SDA-frames as well as an analysis comparing it with the conventional approach. The insights obtained from the analysis are experimentally validated in Section IV. Section V examines applications for which SDA-frames based technique can be used and reports the improvement that can be achieved using an SDA-based method for those cases. Section VI provides a discussion on future work and concludes the paper.

II. Background and Related Work

In this section, we provide a brief review of PRNU-based source camera attribution and video stabilization.

A. PRNU-based Source Camera Attribution

PRNU-based camera attribution is established on the fact that the output of the camera sensor, I, can be modeled as

$$I = I^{(0)} + I^{(0)}K + \psi \tag{1}$$

where $I^{(0)}$ is the noise-free still image, K is the PRNU noise, and ψ is the combination of additional noise, such as readout noise, dark current, shot noise, content-related noise, and quantization noise. The multiplicative PRNU noise pattern, K, is unique for each camera and can be used as a camera fingerprint which enables the attribution of visual objects to its source camera. Using a denoising filter F (such as a Wavelet filter) on a set of images (or video frames) of a camera, we can estimate the camera fingerprint by first getting the noise residual, W_k, (i.e., the estimated PRNU) of the k^{th} image as $W_k = I_k - \hat{I}_k^{(0)}, \hat{I}_k^{(0)} = F(I_k)$, and then averaging the noise residuals of all the images. For determining if a specific camera has taken a given query image, we first obtain the noise residual of the query image using F and then correlate the noise residual with the camera fingerprint estimate.

For images, the PRNU-based method has been well studied. Following the seminal work in [1], much research has been done to improve the scheme [17]–[21], and also make camera identification effective in practical situations [2], [3], [5], [6], [22]. Researchers have also studied the effectiveness of the PRNU-based method by proposing various counter forensics and anti-counter-forensics methods [23], [24]. It has also shown that the PRNU method can withstand a multitude of image processing operations, such as cropping, scaling [25], compression [26], [27], blurring [26], and even printing and scanning [28].

In contrast, there has been lesser work dedicated to PRNU-based camera attribution from a video [29]. Mo Chen et al. [30] first extended PRNU-based approach to camcorder videos. They used Normalized Cross-Correlation (NCC) to correlate fingerprints calculated from two videos, as the videos may be subject to translation shift, e.g., due to letter-boxing. To compensate for the blockiness artifacts introduced by heavy compression (such as MPEG-x and H26-x compression), they discard the boundary pixels of a block (e.g., a JPEG block). In [31], McCloskey proposed a confidence weighting scheme that can improve PRNU estimation from a video by minimizing the contribution from regions of the scene that are likely to distort PRNU noise (e.g., excluding high-frequency content). Chuang et al. [7] studied PRNU-based source camera identification with a focus on smart-phone cameras. Since smart-phones are subject to high compression, they considered only I-frames for fingerprint calculation and correlation. Chen et al. [9] proposed a method to find PRNU noise from wireless streaming videos, which are subject to blocking and blurring. In their approach, they divided a video frame into multiple blocks and did not consider the blocks having significant blocking or blurring artifacts. Chaung et al. [7] showed that the best possible fingerprint could be computed when all the frames are considered (instead of using only the I- or P-frames). However, to the best of our knowledge, efficient computation of fingerprint from a given video is a relatively unexplored area.

B. Affine Transformation in Video Stabilization

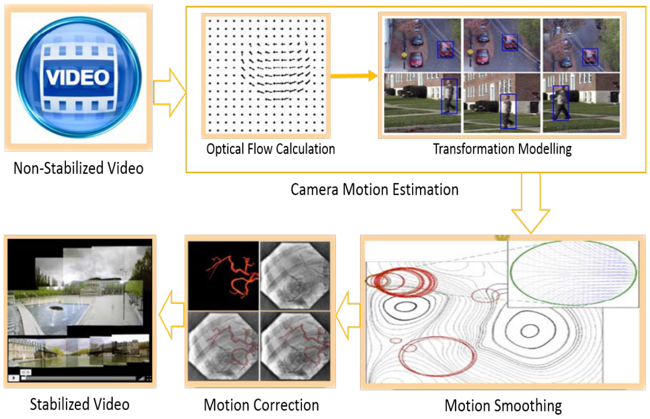


Fig. 1: Video Stabilization Pipeline. This figure is a modified version of a figure that appeared in [32].

An out-of-camera digital video stabilization process contains three major stages: camera motion estimation, motion smoothing, and motion correction (Fig. 1) [33] [32]. In the motion estimation step, the global inter-frame motion between adjacent frames of a non-stabilized video is modeled from the optical flow vectors of the frames using an affine transformation. In the motion smoothing step, unintentional translations, rotations, shearing, are filtered out from the global motion vectors using a low pass filter. Finally, in the motion correction step, a stabilized video is created by shifting, rotating, shearing, or zooming frames according to the parameters in the filtered motion vector. Since each video frame can use different parameters, pixels can be misaligned with the sensor array. For example, one frame can be rotated with an angle -1 degree while another by 0.5 degrees.

Digital video stabilization presents a big challenge for PRNU-based camera attribution. The frame specific affine transformations described above make the PRNU method ineffective as there is a misalignment between frames. The brute-force methods [10], [24] proposed to address the stabilization issue have had limited success and resulted in low performance. These brute-force methods try to overcome the desynchronization issue by first finding the stabilization parameters through an exhaustive search and then performing the corresponding inverse affine transformation. Such methods, therefore, have very high computation overhead. Recently, Mandelli et al. [11] improved over brute-force approaches by using a *best-fit reference frame* in the parameter searching process rather than using the first frame of the given video. The *best-fit reference frame* is obtained by looking for a frame that matches the largest number of frames. Their approach also has high computation overhead.

III. Spatial Domain Averaging

As mentioned in the introduction, this paper proposes spatial domain averaging for computing camera fingerprints, which reduces the number of denoising operations when many visual objects are available. In the proposed method, efficient computation of a fingerprint is achieved by first creating averaged frames from a large collection, and using these averaged frames for computing the fingerprint. For example, given a video with m frames, g non-intersecting equal-sized subgroups are formed each with $d = \frac{m}{g}$ frames. A *Spatial Domain Averaged frame (SDA-frame)* is created from each subgroup by getting the mean of the d frames in the subgroup. Then, in the second step, each SDA-frame is denoised, and an averaging of the estimated PRNU noise patterns is done to arrive at the final camera fingerprint estimate. In this manner, the number of frames that are denoised gets reduced by a factor of d. An SDA-frame obtained from three different images is shown in Fig. 2.

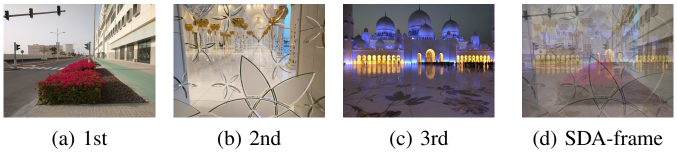


Fig. 2: SDA-frame is the average of 1^{st}, 2^{nd}, and 3^{rd} frames.

The proposed method is inspired by the fact that although the denoising filter is designed to remove random noise from an image originating from the camera sensors (e.g., readout noise, shot noise, dark current noise etc.), as well as noise caused by processing (e.g., quantization and compression), it is not able to do a perfect job. Therefore, some scene content leaks into the extracted noise pattern. Averaging in the spatial domain acts as a preliminary filter that smoothens the image and potentially reduces the content noise that leaks into the extracted noise pattern. Of course, the effectiveness of the approach then depends on the nature of the two noise signals. Below we analyzed this fact and characterized the relationship between the noise signal arrived at by using the conventional approach and the SDA-approach.

Further, when using the proposed approach, many questions arise. First, does frame-averaging lead to a drop in the accuracy of the fingerprint computed as compared to the conventional method, assuming the same number of images are used for both? If so, what is the trade-off between the decrease in computation and the loss in accuracy? Can accuracy be increased by utilizing more images in the SDA method? If so, what is the optimal combination of averaging and denoising that leads to the least computation while yielding the best performance? Then, we investigated these questions, both theoretically and experimentally. We first provide a mathematical analysis using a simple framework in the two subsections below. We then validate our study in the next section by providing experimental results. The results show that spatial domain averaging strategy can indeed result in significant savings in computation while maintaining performance and, in some cases, improving it.

The rest of this section provides an analysis of spatial domain averaging. To this end, we first give a summary of the conventional method and then analyze the SDA method.

A. Conventional method

As discussed in Section II, in the conventional method, the camera fingerprint is estimated from n images from a known camera. Each image I can be modeled as $I = I^{(0)} + I^{(0)}K + \psi$, where ψ is the random noise accumulated from a variety of sources (as in (1)) and K is the PRNU noise.

To estimate K, a denoising filter, F, such as [34], BM3D [35], is used to estimate the noise free signal $I^{(0)}$. Using such a filter, we denote the noise residual as $W = I^{(0)}K + \psi + \xi$, where ξ is the content noise. This noise is essentially due to sub-optimal denoising filter that is unable to completely eliminate the content from PRNU noise. Then, from n known images, the camera fingerprint estimate, $\hat{K}$, can be obtained using Maximum Likelihood Estimation (*MLE*) as

$$\hat{K} = \frac{\sum_{i=1}^{n} W_i . I_i}{\sum_{i=1}^{n} I_i^2} \tag{2}$$

where W_i is noise pattern extracted from I_i.

Note that in the estimated camera fingerprint, $\hat{K}$, ψ and ξ are the unwanted noise. The quality of $\hat{K}$ can be assessed from its variance $Var(\hat{K})$ [36]. The lower the variance is (i.e., images with smooth content), the higher the quality becomes. Assuming that ψ and ξ are independent White Gaussian Noise with variances σ_1^2 and σ_2^2 respectively, $Var(\hat{K})$ can be found as (using Cramer-Rao Lower Bound as shown by Fridrich et al. [36])

$$Var(\hat{K}) \geq \frac{\sigma_1^2 + \sigma_2^2}{\sum_{i=1}^{n} I_i^2}. \tag{3}$$

Thus a better PRNU is obtained from lower σ_1^2 and σ_2^2 (i.e., high luminance and low textured image [36]).

B. Proposed SDA method

In this subsection, we derive the variance of the estimated camera fingerprint obtained using frame averaging. We then compare this variance with that obtained by the conventional approach (in (3)).

Suppose $I_1, I_2, \ldots, I_m$ are m images used to compute the camera fingerprint using the SDA method. With frame averaging, these m images are divided into $g = \frac{m}{d}$ disjoint sets of equal size with d pictures in each set. From each set, an SDA-frame is computed. Thereafter, the process is similar to the conventional approach. Each SDA-frame is denoised, and the camera fingerprint is computed from g noise residuals using MLE.

Suppose, I_i^{SDA} is the SDA-frame obtained from the i^{th} image set. Then

$$\begin{aligned} I_i^{SDA} &= \frac{\sum_{j=(i-1)d+1}^{id} I_j}{d} \\ &= \frac{\sum_{j=(i-1)d+1}^{id} (I_j^{(0)} + I_j^{(0)}K + \psi_j)}{d} \end{aligned}$$

We can write the above equation as

$$I_i^{SDA} = I_i^{(0),SDA} + I_i^{(0),SDA}K + \psi_i^{SDA}, \tag{4}$$

where $I_i^{(0),SDA}$ is the noise free image, and ψ_i^{SDA} is the random noise (from pre-filtering sources) in the SDA-frame. This noise can be written as

$$\psi_i^{SDA} = \frac{\sum_{j=(i-1)d+1}^{id} \psi_j}{d}.$$

Suppose σ_1^2 is the variance of ψ's (which is assumed to be White Gaussian Noise). Then, the variance of ψ_i^{SDA} turns out to be $\frac{\sigma_1^2}{d}$.

Suppose W^{SDA} is the noise residual of each SDA-frame, I^{SDA}. Then,

$$\begin{aligned} W^{SDA} &= I^{SDA} - F(I^{SDA}) \\ &= I^{(0),SDA}K + \psi^{SDA} + \xi', \end{aligned}$$

where F is the denoising filter, and $\xi' = I^{(0),SDA} - F(I^{SDA})$ is the content noise due to the sub-optimal nature of the denoising filter. Note that ξ' is assumed to be independent of PRNU signal $I^{(0),SDA}K$ (although ξ' contains content layover $I^{(0),SDA} - F(I^{SDA})$ as ξ' is negligible compared to $I_0^{SDA}K$ [36].

We know that ξ' is dependent on the smoothness of the SDA-frames. If the frames contain textured content, ξ' is high. Assuming that SDA-frames have similar smoothness to the input frames from which they are created, we consider that ξ' and ξ have the same variance σ_2^2.

Using MLE, the camera fingerprint can now be estimated from g SDA-frames $I_1^{SDA}, I_2^{SDA}, \ldots, I_g^{SDA}$ as

$$K^{\hat{SDA}} = \frac{\sum_{i=1}^{g} W_i^{SDA} . I_i^{SDA}}{\sum_{i=1}^{g} \left(I_i^{SDA}\right)^2}.$$

Using Cramer-Rao Lower Bound, the variance of the estimated fingerprint $K^{\hat{SDA}}$ becomes

$$Var(K^{\hat{SDA}}) \geq \frac{\frac{\sigma_1^2}{d} + \sigma_2^2}{\sum_{i=1}^{g} \left(I_i^{SDA}\right)^2}. \tag{5}$$

In an ideal case, we want that the averaging operation does not degrade the quality of the estimated PRNU from the SDA-frames. In other words, we want that $Var(K^{\hat{SDA}})$ is approximately equal to the variance from the conventional method $Var(\hat{K})$. That is, in other words, using the results from (3) and (5), it is desired that

$$\frac{\frac{\sigma_1^2}{d} + \sigma_2^2}{\sum_{i=1}^{g} \left(I_i^{SDA}\right)^2} \approx \frac{\sigma_1^2 + \sigma_2^2}{\sum_{i=1}^{n} I_i^2}.$$

By simplifying the above equation, we get

$$\frac{\frac{\sigma_1^2}{d}+\sigma_2^2}{\sigma_1^2+\sigma_2^2} \approx \frac{\sum_{i=1}^{g}\left(I_i^{SDA}\right)^2}{\sum_{i=1}^{n} I_i^2}.$$

Suppose

$$\frac{\sum_{i=1}^{g}(I_i^{SDA})^2}{\sum_{i=1}^{n} I_i^2} = \frac{g}{n} \times k$$

where

$$k = \frac{(\sum_{i=1}^{g}(I_i^{SDA})^2)/g}{(\sum_{i=1}^{n} I_i^2)/n}.$$

Note that the value of k is a temporary variable that is less than or equal to 1 as the numerator $\sum_{i=1}^{g}(I_i^{SDA})^2)/g$ is less than equal to the denominator $\sum_{i=1}^{n} I_i^2)/n$. Putting these values in the above equation, we get

$$\frac{g}{n} \times k \approx \frac{\frac{\sigma_1^2}{d}+\sigma_2^2}{\sigma_1^2+\sigma_2^2}.$$

Putting $g = \frac{m}{d}$ in the above equation, we get

$$\frac{m \times k}{d \times n} \approx \frac{\sigma_1^2 + d \times \sigma_2^2}{d \times (\sigma_1^2+\sigma_2^2)}.$$

or,

$$m \approx \frac{n}{k} \times \frac{\sigma_1^2 + d \times \sigma_2^2}{\sigma_1^2+\sigma_2^2} \tag{6}$$

We then discard the temporary variable, k, from the equation. Since $0 < k \leq 1$, the final equation becomes

$$m \leq n \times \Big(\frac{\sigma_1^2 + d \times \sigma_2^2}{\sigma_1^2+\sigma_2^2}\Big) \tag{7}$$

From (7), we can derive the following concluding remarks:

- Since $d \geq 1$, the right-hand side of the equation is at least 1. Therefore, the number of images required in the proposed SDA method (i.e., m) will be more than or equal to the number of images required in the conventional method (i.e., n).
- For smooth images σ_2^2 is close to zero. So, the impact of SDA-depth, d, will be negligible for such images. Therefore, SDA and conventional approaches will have similar performance. However, the SDA technique will be d times faster in the best case.
- For textured images, when the number of for both techniques is equal (i.e., $m = n$), because σ_2 is greater than zero, the conventional approach is expected to outperform SDA approach.
- Since σ_2^2 is greater than zero for textured images, the ratio of images for SDA divided by the conventional approach, $\frac{m}{n}$, will increase as the SDA-depth, d, increases. Therefore, the SDA approach is expected to require more images to achieve the same performance for textured images.

Notice that it is hard to characterize the relationship of σ_1 and σ_2. Also, σ_1 depends on various factors such as shot noise, exposure time, temperature, illumination, image content, and so on. Therefore, we are not focusing on their relationship in this research. In the following section, we experimentally validate the observations listed above.

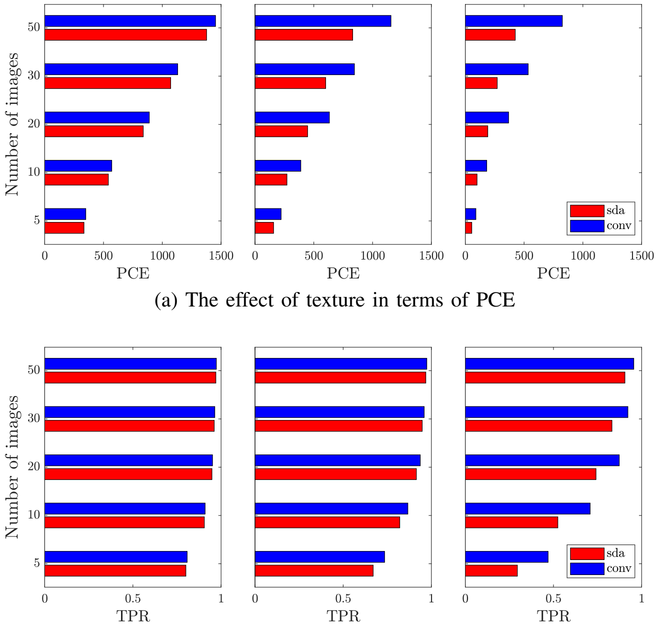
IV. Validation of analysis

In this section, we experimentally verify the main conclusions arrived at by the analysis performed in the previous section. In this section, we used images shared in VISION dataset [15] which contains visual objects from 35 cameras of 11 brands and 27 models. The experiments conducted in this section used only the pristine images, specifically, those tagged as "nat" and "flat" which indicate the images are textured or flatfield, respectively. This dataset contains a total of 11732 images and 650 videos. The implementations were done using Matlab 2018a on Windows 7 PC with 32 GB memory and Intel Xeon(R) E5-2687W v2 @3.40GHz CPU. The wavelet denoising algorithm [34] was used to obtain fingerprint and PRNU noise. PCE and NCC methods were used for comparison. A preset threshold of 60 [37] was used for PCE values for estimating the True Positive Rates (TPR). Values higher than this threshold were taken to conclude that the two visual objects originated from the same camera. All experiments conducted in this section used a block-wise correlation approach. The details of this approach will be given below.

A. Studying the effect of smoothness

To verify the observations of the analysis related to the smoothness of the images used to compute a camera fingerprint, we created three classes of texture level: low(i.e., flatfield), medium (mid), high texture. We used SURF features to do this clustering with respect to the texture level [38]. Although texture level is hard to characterize in an image, we estimated the number of SURF features in each image, and grouped training images accordingly. For each camera in the VISION dataset, we first sorted the images with respect to the number of SURF features. Then three training classes (i.e., low, mid, and high) were created, all of which contained 50 images. The average numbers of SURF features in these classes are $2, 1312$, and 13736 for low, mid, and high, respectively. The rest of the images of each camera are considered as test images. For each of these types, five experiments were conducted by using a random set of 5, 10, 20, 30, and 50 images for computing the fingerprint. So, for example, when we chose 30 flatfield images, we created one fingerprint using the conventional approach by denoising each of the 30 images and then averaging the PRNU noise patterns which is followed by a post-processing step to create the final fingerprint estimate. Another fingerprint estimate using the SDA approach was computed by averaging the same 30 images in the spatial domain and then denoising this SDA-frame of depth 30 along with the post-processing to directly arrive at another fingerprint estimate. Therefore, a total of 30 fingerprints were obtained for each camera (3 categories of images with different texture level; 2 fingerprint extraction techniques; 5 different cardinalities of image sets used for fingerprint computation).

Each of these 30 fingerprints was correlated with the PRNU noise obtained from the test images in the dataset taken with the same camera. To create an abundance of test cases and diversity of FEs, we divided each full-resolution fingerprint into 500×500 disjoint blocks. We correlated them with the corresponding blocks in the test images to match the PRNU noise. As a result, a total of $131,614$ comparisons were made as "true cases".



(a) The effect of texture in terms of PCE

(b) The effect of texture in terms of TPR for $\tau = 60$

Fig. 3: Performance for varying number of low, mid, and high textured images

Fig. 3a shows how image content affects the PCE for fingerprints obtained from $5, 10, 20, 30$ or 50 low-, mid- and high-textured images. The figure shows that with flatfield images, despite the significantly lower number of denoising operations performed by the SDA approach, the results obtained are similar to the conventional approach. This observation holds, regardless of the number of images averaged for fingerprint extraction. The performance of the SDA approach drops as the texture level of images increases. However, this difference can be overcome by increasing the number of images used for the SDA technique but still keeping the number of denoising operations lower than the conventional approach. We investigate this issue in the next subsection.

If we consider the above results in terms of TPR, the SDA approach starts doing better as the PCE is thresholded around a set value (60 in our case) to arrive at the attribution result. So a drop in PCE does not necessarily result in a wrong decision. This improvement can be observed in Fig. 3b, which shows TPR for the same experiments when the threshold is set to 60 as proposed in [37]. The other implications of these figures are already well-known in the field (i.e., flatfield images are better than textured, and as the number of images increases, the quality of fingerprint also increases, which results in a higher PCE and TPR.)

For the sake of clarification, in Table I, we present the results of Fig. 3 for only fingerprints extracted from 50 images. The table shows that the difference between SDA and conventional method increases as the texture level increases. This figure supports the first implication of the theoretical analysis in Section III.

TABLE I: Performance of the proposed SDA method and the conventional method for low, mid and high textured images when a fingerprints is computed from 50 images.

		Texture Level		
		low	mid	high
PCE	SDA	1374	834	425
	conv	1451	1160	824
	SDA/conv	0.947	0.719	0.516
TPR	SDA	97.2	96.7	90.5
	conv	97.5	97.2	95.5
	SDA/conv	0.997	0.995	0.948

To see the full picture, we have correlated images of i^{th} camera with $(i+1)^{th}$ camera in VISION dataset. Hence, a total of $180,384$ correlation of 500×500 blocks were made. This experiment helps us to see if the SDA method harms the false positive rate (FPR). Fig. 4 shows two ROC curves obtained for flat and textured images. As seen, for both flatfield and textured images, the results are similar to the ones in Fig. 3.

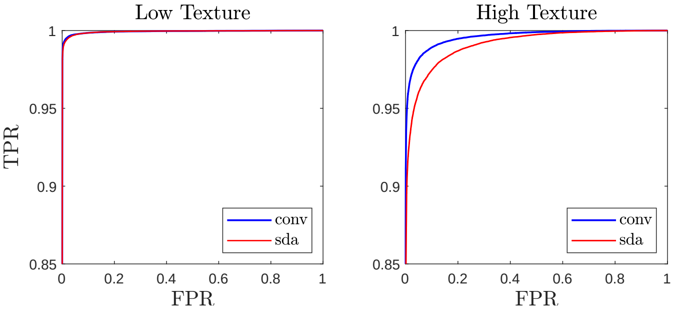


Fig. 4: ROC curve for textured and flatfield images when 10 images are used for FEs

Finally, we estimate how much speedup can be achieved using the SDA method compared to the conventional one. Table II shows the average time it takes to extract a "full resolution" fingerprint estimate by the two methods in the above experiment. For each of the 35 cameras in VISION dataset, we created 10 fingerprints (i.e., for both SDA and conventional methods using $5, 10, 20, 30,$ and 50 images). We then estimated the average time for each fingerprint. Moreover, to avoid further complications in this estimation, we used a single-threaded implementation in this experiment. Notice that in both cases, the same number of images, m, is read from the disk but for the SDA technique only one denoising operation is needed.

In contrast, for the conventional way, m denoising operations are done. This implies that as the training images increase, the speedup also increases. As an example, a speedup of 13.5 times can be achieved by averaging 50 images using the SDA method.

TABLE II: Average time for extracting fingerprint using SDA and conventional methods (in sec)

	5	10	20	30	50
SDA	5.0	6.0	8.2	10.4	14.5
Conventional	21.6	40.8	80.0	118.8	196.6
Speedup	4.3	6.8	9.7	11.5	13.6

Notice that the reason why time requirement for the SDA method is increasing as the number of images increases is due to the I/O time.

B. Fingerprint equivalence for textured images

In the previous subsection, we have shown that when the training images are flatfield, both SDA and conventional methods are performing similarly. Hence, the speedup gained in Table II can be used as a reference for that case. However, for textured images, the TPRs and FPRs are not equivalent. Therefore, better estimates are required for that case.

Section III has shown that more images are needed by the SDA method than conventional for textured images. Thus other estimations are required to understand the correspondence of the methods for textured images. In this experiment, our goal is to investigate the relationship between the number of images required by SDA compared to the number needed by the conventional approach to yield similar performance for textured images while still retaining a speedup in fingerprint computation. This experiment was again performed using images from the VISION dataset [15].

We created a training set from 50 textured images for each camera in the VISION dataset. 19 fingerprints were created using $2, 3, \ldots 20$ images using the conventional approach. We also created 49 fingerprints using the SDA method using $2, 3, \ldots 50$ images. As done in the previous experiment, each fingerprint was partitioned into disjoint 500×500 blocks, and correlations were computed with the corresponding blocks of the test PRNU noise pattern.

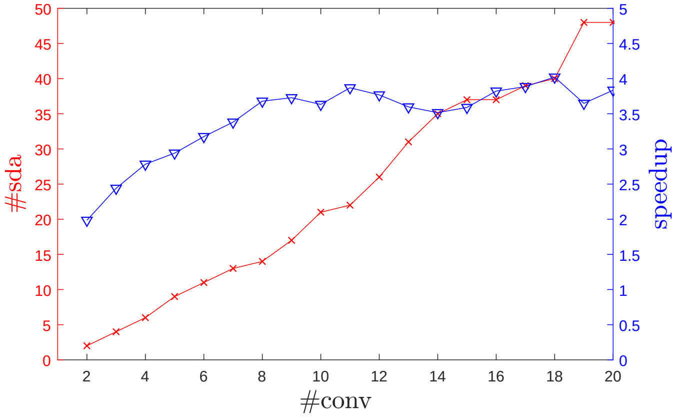


Fig. 5: Fingerprint equivalence for SDA and conventional approaches. x-axis indicates number of images for conventional. The left of y-axis (red) is the number of images required for SDA and the right one (blue) is the speedup gained in this case.

Fig. 5 shows the number of images required by the SDA approach to achieve at least the same TPR as the conventional approach. Moreover, it shows the speedup gained in these cases. For example, when a fingerprint is created from 20 textured images using the conventional way, the same TPR can be achieved using 48 images in the SDA approach. In this way, the fingerprint extraction is approx. 3.85 times faster for the SDA approach. The figure shows that using $2-3$ times more images for the SDA method, up to 4 times speedup can be achieved with no loss in TPR when the images are textured.

C. Effect of SDA-depth on image fingerprint

In Section III, we have shown that as the SDA-depth increases, when the number of images for fingerprint extraction is constant, the TPR is expected to drop. To verify this remark, we used 50 textured images for fingerprint extraction. We didn't include any flatfield image in this set, as flatfield images result in a negligible difference in performance between SDA and conventional fingerprints.

We then created fingerprints using 50 textured images from each camera in the VISION dataset. We set SDA-depth to $1, 2, 5, 10, 25$ and 50. Therefore, we created $50, 25, 10, 5, 2$, and 1 SDA-frames, respectively. The SDA-frames were denoised and then averaged. Finally, a post-processing step is applied to each SDA-FE to obtain the final fingerprint estimate. So, each fingerprint is computed from 50 training images and the rest of the images were used as test images. We correlated each fingerprint with the PRNU noise extracted from the test images in a block-wise manner as done in previous experiments. Notice that SDA-1 is the same as the conventional approach.

TABLE III: TPR and PCE for various SDA depths

	SDA-1	SDA-2	SDA-5	SDA-10	SDA-25	SDA-50
PCE	893.9	701.5	536.1	460.2	395.7	350.7
TPR %	96.5	95.2	92.9	91.3	89.4	87.3

Table III shows that as the SDA-depth increases, the average PCE decreases. For textured images, the more images we combine to create an SDA-frame, the lower the PCE and TPR values that will result. This supports the third observation of the analysis in Section III.

This section has provided a validation of Section III by experimentally supporting all three observations derived from the analysis. Namely, when images are not textured, hence resulting in low post-filtering noise, both the SDA and conventional fingerprints from the same images perform similarly, which can lead up to 13.5 times speedup. On the other hand, textured images and larger SDA-depth results in requiring a higher number of images to achieve the same performance as the conventional method. Yet, a speedup by a factor of 4 can still be achieved in most cases.

In the next section, we apply the proposed approach to more practical problems, and show that SDA fingerprints can perform with significantly higher accuracy or result in significant speedup compared to state-of-the-art fingerprint extraction techniques.

V. Application to computing video fingerprints

In this section, we investigate a more practical use case of the proposed SDA technique, where it makes a more significant impact that is its usage for videos. As Section II explains, two of the most common ways to extract a fingerprint from a video are using only I-frames or using all frames (or the first n frames). While the former results in low performance, the latter can be impractical in many real-life applications due to very high computational needs. For example, fingerprints from 50 1−minute videos (i.e., approximately 1800 frame per video) using a single-thread may take up to a day to compute. In this section, we provide experimental results that demonstrate how using the SDA approach can deliver significant improvements to the time needed for computing fingerprint estimates from video while retaining at least the same performance as conventional approaches.

In each experiment below, three different types of fingerprints (i.e., I-frames only, SDA-frames and ALL-frames) were obtained from each video. For the sake of simplicity, we refer to them as *I-FE* (i.e., *Fingerprint Estimate*), *SDA-FE*, and *ALL-FE*, respectively. Moreover, in some cases, we add an indication of the SDA-depth when we need to highlight it. For example, SDA-50-FE indicates that the video frames were divided into groups of 50, and each group averaged to create an SDA-frame.

In the first experiment, we examine source matching for videos. That is given two videos, can we determine if they are from the same camera. Next, we investigate a more complicated case that involves mixed media. In this subsection, we also analyze an important question related to mixed media: "What is a good balance of SDA-depth which optimizes speed and performance?". In the next two subsections, we examine the performance achieved with video and images obtained from social media such as Facebook and YouTube. Finally, we show how the proposed technique can be used for source attribution with moderate length stabilized videos (i.e., up to 4 minutes) from which obtaining a "reliable" FE might take a couple of hours each using all frames.

Two datasets were used in all the experiments, the NYUAD-MMD, and VISION datasets. The NYUAD-MMD contains visual objects from 78 smartphone cameras (19 brands, 62 different models). From these cameras, a total of 6892 images, and 301 non-stabilized videos (most of them being 40+ seconds) of different resolutions, as allowable by the camera settings, were collected. This makes it a challenging dataset for mixed media attribution. Moreover, we added 5 more cameras that have stabilized videos that are longer than 4 minutes. Hence, we used this dataset for experiments using mixed media and stabilized video. All the visual objects in this dataset are pristine (i.e., no out-camera operations are applied.) The VISION dataset, on the other hand, contains visual objects from 35 cameras. It contains both stabilized and non-stabilized videos. Depending on the experiment, we used a different subset of the database. The videos in this dataset are generally high quality with high luminance, and most of them are at least 1-minute long. Along with these original visual objects, this dataset contains copies of those objects compressed by social media such as Facebook and YouTube. Hence, we used this dataset in experiments involving social media.

Note that, as opposed to the previous section, which used a block-wise correlation, these experiments were done using full frames of the videos and images.

A. Matching Two Non-Stabilized Videos

In the first experiment, we examine source matching for videos using FE computed from the three different approaches that have been presented. Our goal was to estimate the length of videos and the resulting computation time needed to achieve greater than 99% TPR for I-FEs, SDA-FEs, and ALL-FEs. This way, a clear comparison of the three approaches could be made.

FE from the non-stabilized videos of the VISION dataset was first created. As [15] presents, 19 cameras have non-stabilized videos, whereas the rest contain stabilized videos. There is a total of 351 videos from those 19 cameras that are tagged as one of nine different "categories". Those are "flat", "indoor", and "outdoor" videos under three different movements: "still", "move", and "panrot". These videos are typically high quality with HD or Full HD resolutions. Because every camera in VISION dataset only had a single video resolution, cropping and resizing is not involved in estimating the "true cases".

An FE was extracted from the first $5, 10, \dots 40$ seconds of each video using the two techniques mentioned in Section II and the proposed method. On average, each video had approximately one I-frame per second. We selected an SDA-depth of 30 resulting in an SDA-frame from each second of a video.

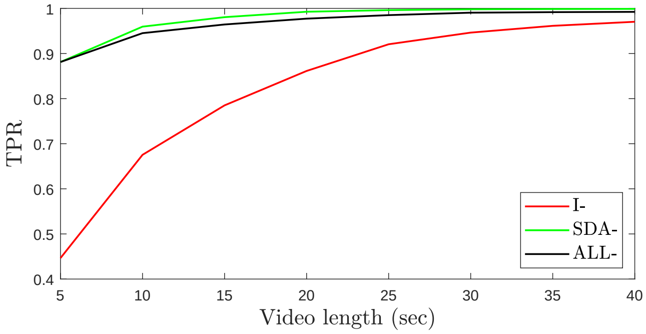


Fig. 6: TPR for different lengths of video using I-FEs, SDA-FEs, and ALL-FEs

Fig. 6 shows TPR using I-FE, SDA-FE, and ALL-FE with respect to increasing video length. As seen, SDA-FEs outperforms ALL-FEs in this setting for all video lengths. The difference varies between 0.5 (for 5 sec videos) and 1.7%(for 15 sec videos). Both FEs achieve significantly higher TPR than I-FEs. For example, for 10 seconds video, SDA-FEs and ALL-FEs result in 94.1% and 95.6% TPR, respectively, whereas I-FEs can only reach 67.2% TPR.

While a TPR of more than 95% can be achieved with 10-second videos using SDA- and ALL-FEs, at least 30-second videos are needed to accomplish the same TPR for I-FEs. This difference is because SDA-FE and ALL-FEs use all the 300 frames in a 10-second video (i.e., I-, B- or P-frames), whereas the I-FEs use only 30 I-frames on average and "waste" the rest of the frames. Hence, for this setting, I-FEs fail to reach a comparable accuracy as the other two methods. Thus, it is fair to say SDA- and ALL-FEs outperform the I-FEs in terms of TPR for the preset PCE threshold.

In addition to Fig. 6, we added ROC curves for FEs obtained from 15-second videos that compares all three methods.

For estimating the correlation of the false cases, we selected the cameras having the non-stabilized videos. We sorted the cameras with respect to their brand and models. For example, after the sorting, the two "Samsung Galaxy S3 Mini" VISION dataset (i.e., "D01" and "D26") came together so that we can correlate them. The reason behind this sorting is when we correlate the same model cameras, and it may better reflect the effect of non-unique artifacts. Hence, it may help us better understand the limitations of the techniques. The same strategy we mentioned above is then followed (i.e., correlating FEs from i^{th} camera with the $(i+1)^{th}$ camera in the sorted camera list.)

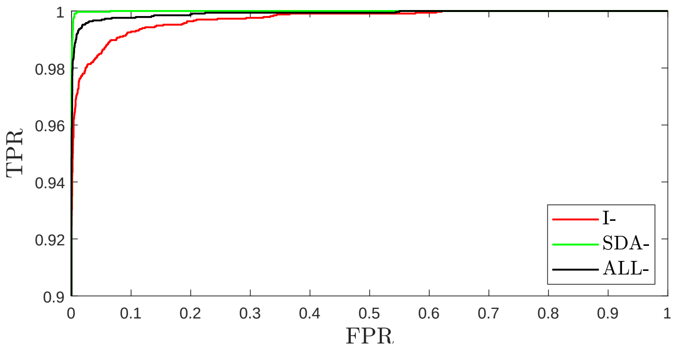


Fig. 7: ROC curves for correlation of two FEs computed from 20-second videos

These results are similar to Fig. 6 that both SDA-FEs perform slightly better than ALL-FEs and both outperform I-FEs.

TABLE IV: Computation time for video fingerprint extraction (in sec)

type	I/O	denoising	total
I-FE	13*	37	50
SDA-FE	25	37	62
ALL-FE	25	1382	1407

In addition to the performance estimation, we compared the required time for the extraction of each FE from a 40-second Full HD video captured @30 FPS. Table IV compares the average times for them. It takes 50, 62, and 1407 seconds for an I-FE, SDA-FE, and ALL-FE, respectively. Notice that, although a lower number of frames are read for I-FEs, the I/O time is quite high compared to the others. This is because finding and extracting I-frames from a video is taking extra time.

When we evaluate the required time to achieve $\approx 90\%$ TPR, we need 5 seconds of video for SDA-FEs and ALL-FEs, whereas I-FEs require more than 20 seconds of video. This suggests that the times required for SDA-FEs and ALL-FEs are approx. 8 and 176 seconds, respectively. Hence, the SDA technique is at least 3 times faster than I-FEs and requires 8 times shorter videos, yet still achieves a higher TPR. Moreover, it performs up to 1.7% higher than ALL-FEs in terms of TPR and speeds up approximately 22.5 times in this setting. Additionally, while SDA-FEs can achieve 99% TPR with 20 seconds videos, the same can be achieved with 30 seconds for ALL-FEs. Therefore, close to 34 times speedup can be achieved in this case when SDA-depth is set to 30.

Notice that these results involve videos that did not undergo any processing such as scaling, compression in social media, and so on. Therefore, it is possible to have lower performance with more difficult datasets, such as when videos are dark or processed. However, our intention here was to demonstrate the effectiveness of the SDA approach first for the simplest of cases. We examine more challenging situations in further experiments below.

B. *Mixed Media Attribution*

As we have seen in the previous subsection, using I-FEs may cause a significant drop in TPR, whereas $20-30$ seconds of video is enough to achieve more than 99% TPR for both SDA-FEs or ALL-FEs. In this subsection, we investigate a more challenging scenario where a video FE needs to be matched with a single query image. In [16], source attribution with mixed-media was investigated using the NYUAD-MMD dataset, which is a very challenging dataset containing images and videos of various resolutions from 78 of cameras. Here, we performed "Train on videos and test on images" experiment for I-FEs, SDA-FEs, and ALL-FEs. That is, a camera FE was computed from the video, and the query image was cropped and resized. Then, its PRNU matched with the FE. The resizing and cropping parameters to perform the matching were obtained from the "Train on images, test on videos" experiment done in [16].

The videos in this dataset were typically around 40 seconds long, each having approximately 1200 frames. The dataset contains a total of 301 non-stabilized videos and 6892 images from those cameras. Each video FE was correlated with the PRNU noise of all the test images from the same camera to estimate "true cases", which ended up with 23571 correlations. Then, each video FE from i^{th} camera was compared with the PRNU noise of images from $(i+1)^{th}$ camera for resizing and cropping parameters that maximize the PCE for the image FE (i.e., the FE obtained from all images of the camera using conventional approach). This way, we estimated the "false cases" resulted in 17755 correlations.

In the previous experiment, we had used a fixed SDA-depth, d, of 30. In this experiment, we used different SDA-depths to investigate its impact on performance and speed. Given a video of m frames (in our case approximately 1200 frames), we divided the frames into groups of $d = 1, 5, 10, 30, 50, 200, 1200$. Therefore, the number of SDA-frames, g, became $1200, 240, 120, 40, 24, 6, 1$ respectively. When $g = 1$, the technique becomes the same as using all frames, whereas when $p = 1200$, only a single SDA-frame is created by averaging all 1200 frames. After obtaining the PCE of the "true" and "false" cases, we created a ROC curve for each video FE type/depth. Fig. 8 shows the ROC curves for each of the SDA-FEs of different depths, as well as I-FE and ALL-FE. The results show that ALL-FE results in the highest performance, whereas I-FE performs significantly poorer compared to others. The proposed SDA method performs close to the ALL-FE method for all depths.

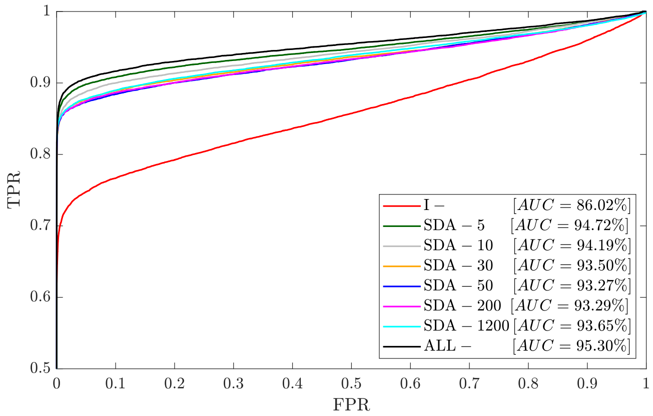


Fig. 8: The ROC curves for varying SDA-depths

Table V shows more detailed results. $|\overline{PCE}|$ stands for the average of the PCE ratios with respect to I-FEs. For example, when an ALL-FE from i^{th} video is correlated with the noise of j^{th} image, its PCE is, on average, 3.2 times higher compared to the I-FE obtained from the same video. The reason we used such a normalization instead of average PCE is that outliers have a big impact on average PCE. Moreover, the table shows the TPR for the PCE threshold of 60, the average time to extract a FE, and the speedup compared to ALL-FEs. As seen, the results indicate that the TPR of the SDA method is very close to ALL-FE. However, a speedup of up to 52 times can be achieved using the SDA method.

TABLE V: Detailed information for mixed media attribution

	I-	ALL-	5	10	30	50	200	1200
$\lvert\overline{PCE}\rvert$	1.0	**3.2**	3.1	2.9	2.6	2.6	2.5	2.4
TPR(%)	64.0	**83.1**	82.3	81.3	80.0	79.8	80.1	79.8
time(s)	50	1407	276	142	62	48	32	**27**
speedup	28.1	1.0	5.1	9.9	22.7	29.3	44.0	**52.1**

Similar to the previous experiment using I-FEs have significantly lower accuracy (at least 16% lower TPR). Moreover, when SDA-depth $\geq$ 30, SDA-FEs are faster to extract as compared to I-FEs. Notice that when ALL-FEs are used, it takes approximately five days to create all the FEs from the 301 videos in the NYUAD-MMD dataset using a single-threaded implementation. This type of performance will be impractical for many applications.

C. *Train and test on YouTube videos*

This experiment explores the performance achieved when two video FEs from YouTube are correlated. Although this experiment is essentially the same as the Section V-A, it is relevant in practice as high compression is involved. Note that a key motivation of the SDA approach is that when high compression is used, a large number of frames are needed for computing a reliable FE. We created FE from all non-stabilized YouTube videos in the VISION dataset (i.e., the ones labeled flatYT, indoorYT, and outdoorYT) using only I-frames, SDA-50, SDA-100, SDA-200, and ALL-frames. Here, we used the first $10, 20, \ldots 60$ seconds of the YouTube videos to extract FEs. Each 60 second video had approximately 1800 frames that were used for SDA- or ALL-FEs, whereas they contained 31.3 I-frames on average. After fingerprint extraction, we correlated each video FE with others of the same type and same length taken by the same camera. For example, an I-FE from 20 seconds of video is correlated with all I-FEs obtained from the rest of the 20 seconds videos from the same camera. The same was done for SDA- and ALL-FEs. This way, a total of 3124 correlations were done for each type.

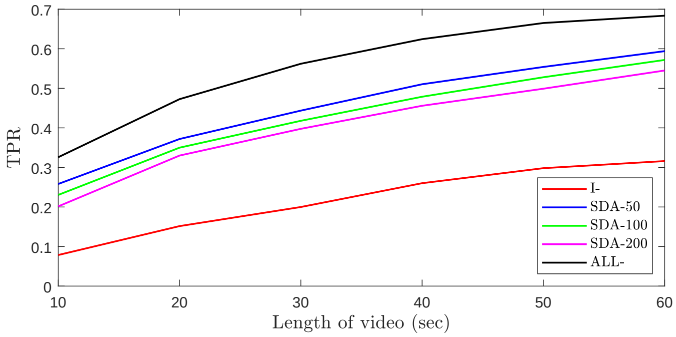


Fig. 9: The effect of FE type and video length on TPR for YouTube videos

Fig. 9 shows the TPR for varying lengths of video for each FE type. The figure shows that I-FEs perform very poorly for all cases, and any FE type created from a video of more than 20 seconds outperforms I-FEs. While ALL-FEs perform better than SDA-FEs for the same-length videos, this difference can be overcome by increasing the video length but still using much fewer denoising operations. For example, SDA$-$50 obtained from 50 second videos or SDA$-$100 from 60 seconds videos, perform approximately the same as ALL-FEs obtained from 30 seconds (within $\pm 1\%$ TPR range). Hence, instead of using 900 frames for ALL-FEs, using 1800 frames for SDA$-$100 can result in significant speedup with no loss in TPR. While an ALL-FE from 900 frame of a Full HD video takes 1045 seconds to compute, and SDA$-$100 FE from 1800 frames, which only does 18 denoising instead of 900, takes 56 seconds to compute. Therefore, a speedup of close to 19 times can be achieved with SDA$-$100 with a 1% increase in TPR. Notice that, because most videos are around 60 seconds in the VISION dataset, it limits the maximum length we could use in our experiments.

Along with the above experiment, we compared all videos with the videos of the next camera. This way, we were able to obtain the correlation of false cases, which were a total of 6283 correlations. Fig. 10 shows the ROC curve obtained from the comparison of these true and false cases. In terms of AUC, SDA-FE, and ALL-FE methods perform similarly, and they are superior compared to I-FEs. However, there is a small difference between ALL-FEs compared to SDA-FEs for very low FPRs which can also be seen in Fig. 9.

D. *Train on Facebook images, test on YouTube videos*

From the previous experiments, we know that the SDA method can help achieve a significant speedup for both videos and images with a small loss in performance, which can be

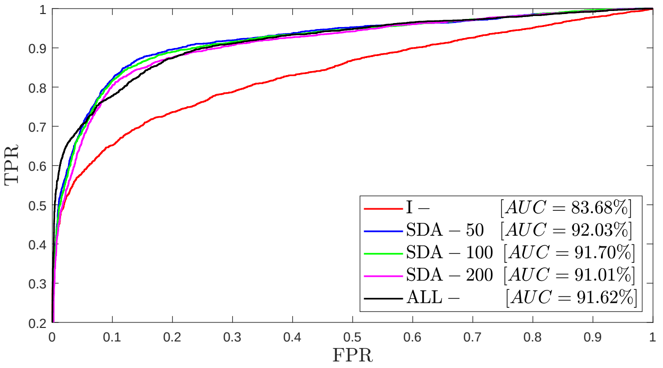


Fig. 10: ROC curve for correlation of 60 seconds YouTube videos

overcome by increasing the number of still images used for fingerprint extraction if available. In this experiment, our goal was to show that the proposed method can be successfully applied to other social media. Specifically, in this subsection, we extract FEs from Facebook images and match them with the FE of YouTube videos. We call this the "Train on Facebook images, test on YouTube videos" experiment. The importance of this experiment is both media sharing services contain billions of visual objects, and computing ALL-FEs from these collections can have very high time complexity. Therefore, faster fingerprint extraction methods (along with search techniques) that speed up attribution are badly needed.

In this experiment, for the cameras in the VISION dataset that had non-stabilized videos, we created a FE from 100 Facebook images (i.e., the ones labeled FBH) using conventional fingerprint computation method. We then used the FEs from non-stabilized YouTube videos (those created in the previous experiment). We again used I-frames, SDA-50, SDA-200, SDA-600, and ALL-frames that were computed from the first 60 seconds of YouTube videos. We then correlated the image FE of a camera with the FE of each video of each type using the efficient search proposed in [16], and a total of 343 pairs were compared for each FE type. Table VI shows the TPR of these correlations. Similar to "Train on videos, test on images" experiment, these results show that for FEs obtained from Facebook images matches with 81.34% TPR with the YouTube videos for SDA-50 which is higher than both ALL-FEs and I-FEs. On the other hand, FEs from I-frames yield approximately 30% lower TPR. These results show that the SDA approach is a good replacement overusing I-FEs or ALL-FEs for this scenario.

TABLE VI: TPR of FEs extracted from Facebook images vs FEs from YouTube videos extracted using different methods

	I-FE	SDA-50	SDA-200	SDA-600	ALL-FE
TPR %	51.6	81.4	79.9	78.1	79.6

E. Matching two stabilized videos

A recent work [12] has shown that a FE obtained from a long stabilized video can successfully be matched with other videos from the same camera. However, thousands of frames must be denoised, which may not be practical in many circumstances. A potential alternative for this problem is the use of the SDA method, which may lead to a significant speedup. To evaluate this, we captured stabilized videos from 5 cameras (not included in NYUAD-MMD). These cameras are Huawei Honor, Samsung S8, Samsung S9, iPhone 6plus, and iPhone 7plus. A total of 37 videos were captured, which added up to 260 minutes.

We extracted FEs from the frames of $20, 40, \ldots 240$ second video lengths using conventional (I-frame and ALL-Frame) method as well as SDA method for SDA-depths of $30, 50,$ and 200. These depths were deemed to be reasonable choices from previous experiments. As shown in [8], [10], [11], the first frame of the videos are typically not geometrically transformed. Since we divide the video into pieces, some video pieces do not have an untransformed frame. So, we discarded the first frame of each video to avoid inconsistencies. We correlated each FE with the other FEs of different videos from the same camera that is created using the same number of frames. For example, SDA-30-FEs of 20 second videos are correlated with the same type FEs from the same camera.

Fig. 11 shows the TPR for three cameras (i.e., Huawei Honor, Samsung S8, and iPhone 6plus) and the total average of all the five cameras.

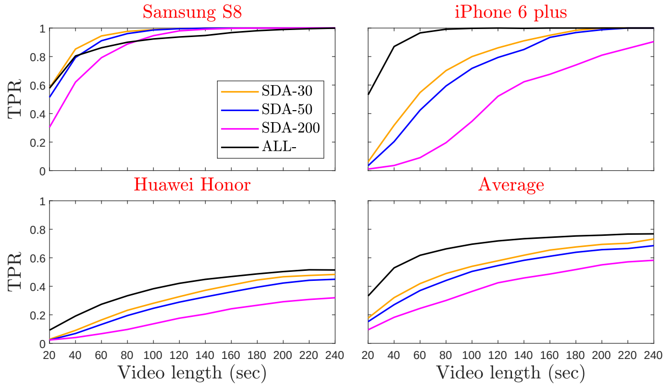


Fig. 11: TPR for stabilized videos for varying SDA-depths (All subfigures share the same axis limits)

The results show that as videos get longer, ALL-FEs and SDA-FEs achieve higher TPR. Moreover, the effect of increased SDA-depth is more significant for this case in comparison to non-stabilized videos. While for some cameras, ALL-FEs and SDA-FEs perform similarly (e.g., Huawei and Samsung cameras), for others (e.g., iPhone cameras), there is a significant difference between the two. For example, for Samsung S8 SDA-200-FE from 120 seconds video, perform similarly as 180 seconds ALL-FE. Therefore, for this particular case, SDA-200 can speedup 66 times $\left(\text{i.e. } \frac{180}{120} \times \frac{1407}{32}\right)$(see Table V for times). On the other hand, for iPhone 6 plus, ALL-FEs from 60 seconds video and 160 seconds SDA-50 have similar TPR. Therefore, 11 times $\left(\text{i.e., } \frac{60}{160} \times \frac{1407}{48}\right)$ speedup can be achieved in this case. Hence, a speedup between these numbers (i.e., 11 and 60) can be achieved without any loss in TPR if a long video is available.

Furthermore, we also performed an additional experiment with the stabilized videos from the VISION dataset. In this experiment, we estimated both TPR and FPR. In the VISION dataset, there are 16 cameras with stabilized videos. Each stabilized video is approximately one minute long. Since we needed a longer stabilized video, we combined four one-minute videos and produced one four-minute video. A camera fingerprint was then computed from each four-minute video. For true cases, the fingerprint was then correlated with a fingerprint computed from the same camera (two fingerprints do not contain common video). For false cases, the fingerprint was correlated with a fingerprint computed from a different camera. There were a total of 3200 correlations for both true and false cases (i.e., 200 correlations for each of the 16 cameras.).

Fig. 12 shows how ALL-FE and SDA-FE perform. As shown in the figure, both ALL- and SDA-FEs have a higher area under curve (AUC) than I-FEs. However, the SDA-FE method requires a significantly lower computation cost than the ALL-FE method.

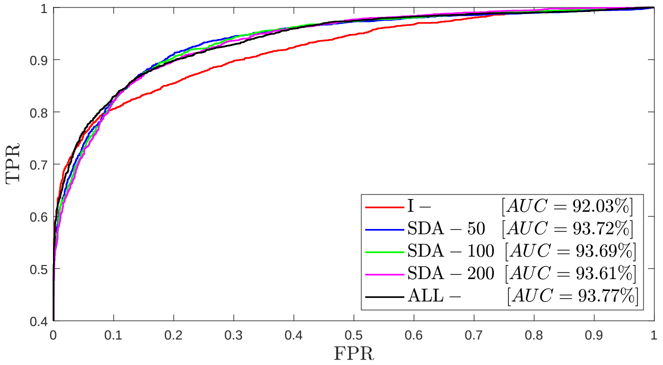


Fig. 12: ROC curves for 4-minute stabilized videos taken from VISION dataset

Overall, this section shows that the proposed SDA-FEs outperforms the commonly used I-frame-only technique in all the cases for videos. These include mixed media, stabilized videos, and social media. On the other hand, the SDA-FEs achieves comparable results as ALL-FEs with up to 52 times speedup in these experiments. We also show the impact of SDA-depth on the performance that can be achieved in various cases.

VI. Conclusion and Future Work

This paper has investigated camera fingerprint extraction using Spatial Domain Averaged frames, which are the arithmetic mean of multiple still images. By adding one extra step of averaging before denoising, a significant speedup can be achieved for fingerprint extraction. We show that this technique can successfully be used for images, non-stabilized videos as well as stabilized video to speedup the fingerprint extraction process. The proposed method is especially useful when the number of denoising operations needed can be very high. For example, when dealing with non-stabilized or highly compressed stabilized videos or images from social media.

It is often considered that for video source attribution, using only I-frames for fingerprint extraction (I-FEs) is "enough" to achieve high performance. However, in this research, we have shown that I-FEs performs poorly compared to ALL-FEs in all cases. On the other hand, using ALL-FEs is impractical due to the substantial computation time needed for practical scenarios where thousands of videos can be available. The proposed SDA approach comes into play here to resolve the problem of I-FEs (i.e., accuracy) and ALL-FEs (i.e., speed). Both SDA- and ALL-FEs perform similarly in most cases. When the SDA method performs worse, this can be overcome by using more of the available frames, if any.

The proposed technique can be used for other source attribution related problems where many denoising operations are needed. For instance, this method can be applied when many "partially misaligned" still images, and a suspect camera is available. For example, a seam carved video contains many partially misaligned frames with its source camera. In such a scenario, instead of denoising all frames of the video, the SDA technique can be used as a way to speed up this process. Moreover, determining whether a video is stabilized or not is another issue which requires a number of denoising operations. As an alternative to using only I-frames, the proposed SDA technique could successfully work with only 2 denoising operations.

Another avenue for future research is to create an SDA-FE in a weighted manner such that performance achieve with the SDA method can be increased. Two of the potential ways to accomplish this are weighting I-, P- and B- frames differently, and weighting the frames in a block-by-block manner. For example, it has been shown that flatfield images perform better with the SDA method compared to textured ones. Using this idea, one may weight textured regions differently from the smooth areas.

References

[1] J. Lukas, 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.

[2] M. Goljan, J. Fridrich, and T. Filler, "Managing a large database of camera fingerprints," in *Media Forensics and Security II*, vol. 7541. International Society for Optics and Photonics, 2010, p. 754108.

[3] S. Bayram, H. T. Sencar, and N. Memon, "Efficient sensor fingerprint matching through fingerprint binarization," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1404–1413, 2012.

[4] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Compressed fingerprint matching and camera identification via random projections," *IEEE Transactions of Information Forensics and Security*, vol. 10, no. 7, pp. 1472–1485, July 2015.

[5] S. Bayram, H. T. Sencar, and N. Memon, "Sensor fingerprint identification through composite fingerprints and group testing," *IEEE Transactions of Information Forensics and Security*, vol. 10, no. 3, pp. 597–612, March 2015.

[6] S. Taspinar, H. T. Sencar, S. Bayram, and N. Memon, "Fast camera fingerprint matching in very large databases," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4088–4092.

[7] W.-H. Chuang, H. Su, and M. Wu, "Exploring compression effects for improved source camera identification using strongly compressed video," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1953–1956.

[8] S. Taspinar, M. Mohanty, and N. Memon, "Source camera attribution using stabilized video," in *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 1–6.

[9] S. Chen, A. Pande, K. Zeng, and P. Mohapatra, "Video source identification in lossy wireless networks," in *IEEE INFOCOM*, 2013, pp. 215–219.

[10] M. Iuliani, M. Fontani, D. Shullani, and A. Piva, "Hybrid reference-based video source identification," *Sensors*, vol. 19, no. 3, p. 649, 2019.

[11] S. Mandelli, P. Bestagini, L. Verdoliva, and S. Tubaro, "Facing device attribution problem for stabilized video sequences," *IEEE Transactions on Information Forensics and Security*, 2019.

[12] J. Lubin, M. Isnardi, C. Spence, I. Sur, and A. Chaudhry, "Joint sensor fingerprinting and processing history recovery for visual media forensics," Private conversation, 2018.

[13] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Transactions on information forensics and security*, vol. 3, no. 1, pp. 74–90, 2008.

[14] T. Gloe, S. Pfennig, and M. Kirchner, "Unexpected artefacts in prnu-based camera identification: a'dresden image database'case-study," in *Proceedings of the on Multimedia and security*. ACM, 2012, pp. 109–114.

[15] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, 2017.

[16] S. Taspinar, M. Mohanty, and N. Memon, "Source camera attribution of multi-format devices," *arXiv preprint arXiv:1904.01533*, 2019.

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

[18] Y. Sutcu, S. Bayram, H. T. Sencar, and N. Memon, "Improvements on sensor noise based source camera identification," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 24–27.

[19] 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, no. 2, pp. 260–271, 2012.

[20] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva, "On the influence of denoising in PRNU based forgery detection," in *ACM Multimedia in Forensics, Security and Intelligence*, 2010, pp. 117–122.

[21] C. T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, 2010.

[22] W. Yaqub, M. Mohanty, and N. Memon, "Towards camera identification from cropped query images," in *25th ICIP*. IEEE, 2018, pp. 3798–3802.

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

[24] S. Taspinar, M. Mohanty, and N. Memon, "Prnu based source attribution with a collection of seam-carved images," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 156–160.

[25] M. Goljan and J. Fridrich, "Camera identification from scaled and cropped images," *Proc. SPIE, Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819, pp. 68 190E–68 190E–13, 2008.

[26] E. J. Alles, Z. J. Geradts, and C. J. Veenman, "Source camera identification for low resolution heavily compressed images," in *Computational Sciences and Its Applications, 2008. ICCSA'08. International Conference on*. IEEE, 2008, pp. 557–567.

[27] K. Rosenfeld and H. T. Sencar, "A study of the robustness of prnu-based camera identification," in *Media Forensics and Security*, ser. SPIE Proceedings, E. J. Delp, J. Dittmann, N. D. Memon, and P. W. Wong, Eds., vol. 7254. SPIE, 2009, p. 72540.

[28] M. Goljan, J. Fridrich, and J. Lukáš, "Camera identification from printed images," *Proceedings of SPIE*, vol. 6819, p. 68190I, 2008. [Online]. Available: http://www.ws.binghamton.edu/fridrich/Research/Printed.pdf

[29] S. Milani, M. Fontani, and P. B. et. al., "An overview on video forensics," *Signal Processing Systems*, vol. 1, pp. 1–18, June 2012.

[30] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Source digital camcorder identification using sensor photo response non-uniformity," in *SPIE Electronic Imaging*, 2007, pp. 1G–1H.

[31] S. McCloskey, "Confidence weighting for sensor fingerprinting," in *IEEE CVPR Workshops*, 2008, pp. 1–6.

[32] N. Ejaz, W. Kim, S. I. Kwon, and S. W. Baik, "Video stabilization by detecting intentional and unintentional camera motions," in *IEEE International Conference on Intelligent Systems, Modelling and Simulation*, 2012, pp. 312–316.

[33] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1150–1163, July 2006.

[34] 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.

[35] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," in *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[36] J. Fridrich, "Sensor defects in digital image forensic," *Digital Image Forensics*, pp. 1–43, 2013.

[37] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Media forensics and security*, vol. 7254. International Society for Optics and Photonics, 2009, p. 72540I.

[38] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.