GPU-ACCELERATED SIFT-AIDED SOURCE IDENTIFICATION OF STABILIZED VIDEOS

Andrea Montibeller[⋆] *Cecilia Pasquini*[⋆] *Giulia Boato*[⋆]
Stefano Dell'Anna[⋆] *Fernando Pérez-González* [†]

[⋆]University of Trento, Department of Information Engineering and Computer Science, Italy
[†]atlanTTic, University of Vigo, Department of Signal Theory and Communications, Spain

2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897579

ABSTRACT

Video stabilization is an in-camera processing commonly applied by modern acquisition devices. While significantly improving the visual quality of the resulting videos, it has been shown that such operation typically hinders the forensic analysis of video signals. In fact, the correct identification of the acquisition source usually based on Photo Response non-Uniformity (PRNU) is subject to the estimation of the transformation applied to each frame in the stabilization phase. A number of techniques have been proposed for dealing with this problem, which however typically suffer from a high computational burden due to the grid search in the space of inversion parameters. Our work attempts to alleviate these shortcomings by exploiting the parallelization capabilities of Graphics Processing Units (GPUs), typically used for deep learning applications, in the framework of stabilised frames inversion. Moreover, we propose to exploit SIFT features to estimate the camera momentum and identify less stabilized temporal segments, thus enabling a more accurate identification analysis, and to efficiently initialize the frame-wise parameter search of consecutive frames. Experiments on a consolidated benchmark dataset confirm the effectiveness of the proposed approach in reducing the required computational time and improving the source identification accuracy. The code is available at `https://github.com/AMontiB/GPU-PRNU-SIFT`.

***Index Terms*—** Video Source Identification, GPU, SIFT, PRNU, Video Stabilization

1. INTRODUCTION

Digital devices, social media platforms and multimedia data have an increasingly relevant role in our daily life. In order to avoid content improper usage and spreading, it is necessary to develop techniques to prove their origin [1, 2] and identify their source. By source identification we refer to the procedure, mostly required in investigations and courtroom cases, in which the device (i.e., the camera or smartphone) that acquired an image or a video under investigation is identified. With this regard, the research literature is particularly rich when it comes to techniques dealing with images [1, 3, 4], among which the most reliable ones use the so-called Photo Response non-Uniformity (PRNU) [1], a scant and unique residual introduced by the camera sensor every time an image or a video is taken. In fact, by extracting the PRNU from two different images and comparing them in terms of Peak of Correlation Energy (PCE), it can be verified whether they were taken with the same device or not.

However, the PRNU is highly sensitive to spatial transformations (such as radial corrections [5], digital zoom [6], HDR correction [7]), as they cause a misalignment of such noise patterns. In this case, the inverse spatial transformation must be applied in order to restore the original pattern and reliably compare it with reference ones; this requires the estimation of the transformation parameters applied in the first place. Such issues are even more impactful when applied to video source identification, due to stronger compression and more complex spatial transformations such as the Electronic Image Stabilization (EIS) [8], applied by modern devices to improve video quality.

To invert EIS transformations and restore the reliability of the PRNU, many works propose to use a combination of grid searches, predicting methods and parallel CPU processing [9, 10, 11]. However, a common trait of such approaches is the rather high computational burden they entail. In [11], Iuliani et. al. check every possible combination of scaling, rotation and shift parameters by means of a grid search on each frame; to reduce the computational cost, some of the parameters are estimated offline and used as a-priori information. In [9], Mandelli et. al. propose a faster algorithm for the inversion of the EIS, which however implies significant hardware requirements to be computationally efficient. Finally, in [10] Mandelli et. al. propose a method based on a modified version of the Fourier-Mellin transform for efficient estimation of the rotation parameter and the inversion of the EIS. The algorithm obtains promising results in terms of accuracy and computational cost but (just like [9, 11]) it exploits only the information coming from the video intra-frames (I), while fully discarding P and B frames. Furthermore, although the EIS transformations are typically modelled through 8 parameters, all the cited methods ([9, 10, 11]) target the estimation of only three of them in order to avoid combinatorial explosion, thus decreasing the inversion accuracy.

In performing source identification, an alternative approach to the conventional PRNU extraction is the computation of a proper residual by means of deep neural networks, whose weights are learned through a training procedure. It is the case of Noiseprint [2], which has been successfully applied for digital images and recently extended to videos [12], although not dealing with video stabilization issues.

Given the current limitations of existing approaches, in this paper, we propose an innovative solution for source identification of stabilized videos using the PRNU. Our algorithm inverts the EIS by pre-selecting the less stabilized frames through a blind camera momentum estimator before estimating the inversion parameters via grid search. In this phase, we leverage the higher computational and parallelization capabilities of the GPU architectures, which particularly fit our needs as they are optimized for similar point-wise operations arising in computer graphics applications and act here as computing accelerators. Our pipeline includes the use of SIFT features-already used in forensics [13, 14] but never for source identification - in order to exploit the temporal correlation between neighbour frames and efficiently initialize their search parameters.

The theoretical background behind our work is described in Sec-

tion 2 and the proposed method is detailed in Section 3. Experimental results and comparison with the literature are discussed in Section 4, while future directions and conclusions are drawn in Section 5.

2. THEORETICAL BACKGROUND

2.1. Notation

In this paper, $M \times N$ matrices will be denoted with uppercase boldface letters $\mathbf{X}$, their (i^{th}, j^{th}) elements as $X_{i,j}$, and their mean value over all elements as $\overline{\mathbf{X}}$. Similarly, M-dimensional vectors are denoted as lowercase boldface letters $\mathbf{x}$ with mean value $\overline{\mathbf{x}}$. We indicate with $\odot$ the dot product of vectors and with $\mathbb{1}$ the identity matrix.

For a video under analysis, the u-th I-frame appearing in the video stream is denoted as $\mathbf{I}_u$. The normalized cross-correlation (NCC) between two matrices $\mathbf{X}$ and $\mathbf{Y}$ (of the same size $M \times N$) is defined as follows [15]:

$$\rho(\mathbf{X}, \mathbf{Y}) = \frac{(\mathbf{X} - \overline{\mathbf{X}}) \odot (\mathbf{Y} - \overline{\mathbf{Y}})}{||\mathbf{X} - \overline{\mathbf{X}}|| \cdot ||\mathbf{Y} - \overline{\mathbf{Y}}||} \tag{1}$$

If $\mathbf{X}$ and $\mathbf{Y}$ have different sizes we apply zero-padding to the smaller one [6].

2.2. Photo Response non-Uniformity

The PRNU (Photo-Response Non-Uniformity) is a scant residual introduced by the acquiring sensor when the light hits its components, caused by small variations of the output signal from pixel to pixel. Such aberration is unique, related to the materials and manufacturing process of the sensor and more detectable in brighter flat areas [16, 6]. The PRNU is a very weak signal modelled as multiplicative noise [17] and has to be separated from other noise components when extracting it from the image or frame under analysis. Typically, the *noise residual* containing the PRNU is estimated from a single image as $W(\mathbf{I}) \doteq \mathbf{I} - F(\mathbf{I})$ where $F(\cdot)$ is the denoiser [18, 19], which in our case is the Mihcak's wavelet-based denoiser [20]. Instead, the *reference fingerprint* for a given device is computed starting from L images $\mathbf{I}^{(l)}$, $1 = 1, \cdots, L$ taken from the device as follows [17]:

$$\hat{\mathbf{K}} = \left(\sum_{l=1}^{L} \mathbf{I}^{(l)} \cdot W(\mathbf{I}^{(l)}) \right) \cdot \left(\sum_{l=1}^{L} \mathbf{I}^{(l)} \cdot \mathbf{I}^{(l)} \right)^{-1} \tag{2}$$

In Eq. (2), all the operations are pixel-wise.

Given a test image $\mathbf{I}$, the source identification is accomplished by solving a binary hypothesis test consisting in verifying whether $\mathbf{I}$ contains the same PRNU as the camera fingerprint $\hat{\mathbf{K}}$. We will denote the null hypothesis of this test (i.e., $\mathbf{I}$ does not contain $\hat{\mathbf{K}}$) by H_0 and the alternative one (i.e., $\mathbf{I}$ contains $\hat{\mathbf{K}}$) by H_1. We verify these hypotheses using as test statistics the Peak-to-Correlation Energy ratio (PCE) defined in Eq. (3), which consists in computing the peak cross-correlation between the test image residual $W(\mathbf{I})$ and the camera fingerprint $\hat{\mathbf{K}}$ and normalizing it by an estimate of the correlation noise under H_0 [21]:

$$\text{PCE}(\hat{\mathbf{K}}, \mathbf{I}) = \frac{\text{sgn}(\rho(\hat{\mathbf{K}}, W(\mathbf{I})_{\boldsymbol{\delta}_{\text{peak}}})) \cdot \rho^2(\hat{\mathbf{K}}, W(\mathbf{I})_{\boldsymbol{\delta}_{\text{peak}}})}{\frac{1}{MN - |\mathcal{D}|} \sum_{\boldsymbol{\delta} \in \mathcal{I} \setminus \mathcal{D}} \rho^2(\hat{\mathbf{K}}, W(\mathbf{I})_{i+\delta_1, j+\delta_2})} \tag{3}$$

where $\boldsymbol{\delta} = (\delta_1, \delta_2)$ are all the possible shifts that occur between $W(\mathbf{I})$ of size $M' \times N'$ and $\hat{\mathbf{K}}$ of size $M \times N$ such that $0 \leq \delta_1 \leq M - M'$ and $0 \leq \delta_2 \leq N - N'$ and $\boldsymbol{\delta}_{\text{peak}}$ are the coordinates of the peak of correlation according to [22]. $\mathcal{I}$ defines all possible image pixels coordinates, while $\mathcal{D}$ is a cyclic exclusion neighbourhood around the peak of correlation value of size 11×11 pixels to avoid contamination of cross-correlation peaks from H_1 when estimating the cross-correlation noise under H_0 [6].

2.3. Electronic Image Stabilization

Electronic Image Stabilization (EIS) [8] is a post-processing technique used in modern devices and cameras to stabilize video sequences by compensating for temporal camera motion. To this end, a spatial transformation is applied to the acquired frames, which entails a parametric coordinate mapping followed by interpolation. We can model as follows the inverse coordinate mapping between the stabilized and the original frame pixels:

$$\begin{pmatrix} w \\ h \\ 1 \end{pmatrix} = \mathbf{T}_{\mathbf{t}} \cdot \begin{pmatrix} w' \\ h' \\ 1 \end{pmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & 1 \end{bmatrix} \cdot \begin{pmatrix} w' \\ h' \\ 1 \end{pmatrix} \tag{4}$$

where (w', h') are the coordinates of the stabilized pixels and (w, h) the original ones, while $\mathbf{t} = [t_{1,1}, t_{1,2}, ..., t_{3,2}]$ is a 8-dimensional vector of varying parameters. In particular, $t_{1,1}$ and $t_{2,2}$ are related to horizontal/vertical scaling, $t_{1,2}$ and $t_{2,1}$ to rotation, $t_{1,3}$ and $t_{2,3}$ to translation, and finally $t_{3,1}$ and $t_{3,2}$ are the projective parameters. Such model encompasses different types of EIS systems [8], operating on 3-, 5- or 6-axes, depending on the nature of the transformations. With a slight abuse of notation, if $\mathbf{Y}$ is a generic video frame, we will write $\mathbf{T}_{\mathbf{t}}(\mathbf{Y})$ to indicate the version of $\mathbf{Y}$ whose pixels underwent the grid transformation with parameters $\mathbf{t}$ followed by interpolation.

2.4. Keypoint matching and homography estimation

The homography relation between two frames can be estimated through the detection and the interframe matching of keypoints. In particular, we can associate to a pair of two generic frames $\mathbf{X}$ and $\mathbf{Y}$:

- $\mathcal{S}$, a set containing pairs $(\mathbf{s}_{\mathbf{X}}, \mathbf{s}_{\mathbf{Y}})$ of 2-D SIFT keypoints [23] that have been detected in $\mathbf{X}$ and $\mathbf{Y}$, respectively, and result as matching from the application of the DEGENSAC algorithm [24]. An example of this detection and matching process is reported in Figure 1. We employ the *Open-CV* libraries for this purpose;
- $\mathbf{H}_{\mathcal{S}}$, the estimated homography matrix between $\mathbf{X}$ and $\mathbf{Y}$, which is provided as a by-product by the DEGENSAC algorithm starting from the keypoints in $\mathcal{S}$. $\mathbf{H}_{\mathcal{S}}$ has a similar model as in Eq. (4) and, by using the same notation convention, we expect $\mathbf{X} \approx \mathbf{H}_{\mathcal{S}}(\mathbf{Y})$.

In addition, we also define $\tilde{\mathcal{S}}$, a sanitized set of matching keypoints where only those yielding an interframe Euclidean distance $||\mathbf{s}_{\mathbf{X}} - \mathbf{s}_{\mathbf{Y}}||_2$ below an empirical threshold are retained.

3. PROPOSED SOLUTION

We consider the hybrid scenario where the reference fingerprint of a device is estimated starting from flat images acquired by the same device, as in (2).

As highlighted in previous approaches, in order for it to be used for testing video frames, the image-based fingerprint needs to be properly down-scaled and cropped due to size mismatch between

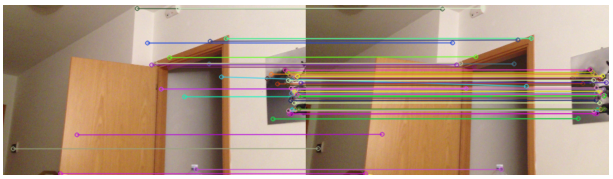

Fig. 1: Example of two frames where SIFT keypoints are detected (colored circles) and matched through the DEGENSAC algorithm (colored lines.)

image and video acquisition; we perform this operation similarly to what is done in [9], so to obtain a fingerprint $\hat{\mathbf{K}}$ to be used for testing video frames.

For a generic frame $\mathbf{I}$, the core of the identification analysis consists in searching for the parameters $\mathbf{t}$ that maximize the PCE value as in Eq. (3) between $\hat{\mathbf{K}}$ and the residual extracted from $\mathbf{T}_{\mathbf{t}}(\mathbf{I})$, the latter being as close as possible to the originally acquired frame before the stabilization.

In order to improve the efficiency and the accuracy of this process, we propose a two-phase methodology encompassing a preselection of lightly stabilized frames (described in Section 3.1) and a frame-wise inversion analysis boosted by a SIFT-based homography estimation (described in Section 3.2). Moreover, we developed a Tensorflow implementation of the overall procedure, building on the Tensorflow add-on libraries for the frame-wise inversion operations, and the PCE and the cross-correlation formulas defined in Eqs. (3) and (1), respectively.

3.1. Selection of low-stabilization frames

In this first phase, pairs of consecutive I-frames are analyzed, with the goal of locating the Group of Pictures (GOPs) where the weaker stabilization has supposedly been applied. Inspired by [25], we achieve this by computing a *camera momentum*, which expresses the global amount of motion between frames. We interpret this measure as a proxy for the strength of the stabilization operation applied to the originally acquired frame: our intuition is that less stabilized frames yield more reliable frame inversion and PRNU matching processes.

Given two consecutive I-frames $\mathbf{I}_u$ and $\mathbf{I}_{u+1}$, the set $\tilde{\mathcal{S}}_u$ containing sanitized matching keypoint pairs $(\mathbf{s}_u, \mathbf{s}_{u+1})$ is obtained as described in Section 2.4. The camera momentum between $\mathbf{I}_u$ and $\mathbf{I}_{u+1}$ is then defined as

$$\overline{\mathbf{\Delta}}_u \doteq \sum_{(\mathbf{s}_u, \mathbf{s}_{u+1}) \in \tilde{\mathcal{S}}_u} ||\mathbf{s}_u - \mathbf{s}_{u+1}||_2 \cdot |\tilde{\mathcal{S}}_u|^{-1}, \tag{5}$$

that is, the average interframe displacement between matching keypoint pairs in $\tilde{\mathcal{S}}_u$.

By iterating this operation along the video duration, we can identify the index A such that

$$A = \arg\min_u \overline{\mathbf{\Delta}}_u \tag{6}$$

The corresponding I-frame $\mathbf{I}_A$ is defined as the *anchor* and identifies the starting point of the successive frame-wise inversion analysis, which will be limited to the frames

$$\mathbf{I}_A, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{V_A} \tag{7}$$

where $\mathbf{P}_v$, $v = 1, \ldots, V_A$ are predicted frames (P or B type) except for $\mathbf{P}_{V_A} \equiv \mathbf{I}_{A+1}$, and V_A is the GOP size. When the set $\tilde{\mathcal{S}}_u$ is empty, as for flat videos, and $\overline{\mathbf{\Delta}}_u$ cannot be estimated, the index A corresponds to the first I-frame of the video.

3.2. SIFT-aided EIS Inversion

In this second phase, we aim at filling a vector $\boldsymbol{\gamma} = [\gamma_A, \gamma_1, \ldots, \gamma_{V_A}]$ containing the maximum PCE value with the reference fingerprint $\hat{\mathbf{K}}$ measured at each of the selected frame indices in Eq. (7) under a number of tested inverse transformations of the frame.

For this purpose, we define the following operators:

- CORRECTION$(\hat{\mathbf{K}}, \mathbf{I}, \mathbf{T}_{\text{init}})$: given a reference PRNU fingerprint $\hat{\mathbf{K}}$ and a frame $\mathbf{I}$, this operator applies a breadth-first search [26] over the parameters in $\mathbf{t}$ with the goal of maximizing PCE$(\hat{\mathbf{K}}, \mathbf{T}_{\mathbf{t}}(\mathbf{I}))$. The search starts from the parameters contained in the optional input matrix $\mathbf{T}_{\text{init}}$, if given.

  As in [9], simplifying assumptions are made, in particular on the scaling parameters ($t_{1,1} = t_{2,2} \doteq \lambda$) and on the rotation parameters ($t_{1,2} = -t_{2,1} \doteq \theta$). Moreover, $t_{3,1}, t_{3,2}$ are set to 0, and $t_{1,3}, t_{2,3}$ are estimated once and kept fixed.

  At each n-th iteration, the pair $(\lambda^{(n)}, \theta^{(n)})$ is determined through exhaustive search over $\mathbf{\Lambda}^{(n)} \times \mathbf{\Theta}^{(n)}$ as the one yielding the transformation of $\mathbf{I}$ with the highest PCE value.

  The sets $\mathbf{\Lambda}^{(n)}$ and $\mathbf{\Theta}^{(n)}$ are finite and progressively narrower neighborhoods of the previous estimates. In particular:

$$\begin{aligned} \mathbf{\Lambda}^{(n)} &= \{\lambda^{(n-1)} + \alpha \cdot 0.01^{-n}\}_{\alpha \in \mathbb{Z} \cap [-5,5]} \\ \mathbf{\Theta}^{(n)} &= \{\theta^{(n-1)} + \alpha \cdot 0.1^{-n}\}_{\alpha \in \mathbb{Z} \cap [-5,5]} \end{aligned} \tag{8}$$

  where α sets the size of $\mathbf{\Lambda}^{(n)}$ and $\mathbf{\Theta}^{(n)}$. At the first iteration ($n = 1$), if the matrix $\mathbf{T}_{\text{init}}$ is given as input, then a corresponding vector $\mathbf{t}_{\text{init}}$ is derived, from which $\lambda^{(0)}$ and $\theta^{(0)}$ are extracted accordingly. Otherwise, they are initialized to 1 and 0, respectively. Moreover, at the first iteration $\theta^{(1)}$ is searched on a denser grid ($\{-5, -4.9, \ldots, 4.9, 5\}$), to improve accuracy.

  If at the n-th iteration, $(\lambda^{(n)}, \theta^{(n)}) \equiv (\lambda^{(n-1)}, \theta^{(n-1)})$, the process stops. Also, in our experiments we fixed the maximum number of iterations at 3. $\mathbf{t}_{\text{max}}$ is the parameter vector for which the maximum PCE value γ is observed. γ is returned as output together with the corrected frame $\mathbf{I}^{(c)}$ obtained by transforming $\mathbf{I}$ with $\mathbf{t}_{\text{max}}$.

- COREGISTRATION$(\mathbf{I}, \mathbf{P})$: this operator returns as output the estimated homography matrix $\mathbf{H}$ between $\mathbf{I}$ and $\mathbf{P}$ computed as in Section 2.4, and the resulting co-registered frame $\mathbf{P}^{(r)} \doteq \mathbf{H}(\mathbf{P}) \approx \mathbf{I}$.

Algorithm 1 SIFT-aided EIS inversion

Inputs: anchor index A,
frames $\mathbf{I}_A, \mathbf{P}_{A+1}, \ldots, \mathbf{P}_{A+V_A}$
reference fingerprint $\hat{\mathbf{K}}$

Initialize: vector $\boldsymbol{\gamma}$, temporary frame $\mathbf{U}$

▷ *Correction of the anchor*
$(\mathbf{I}_A^{(c)}, \gamma_A) \leftarrow$ CORRECTION$(\hat{\mathbf{K}}, \mathbf{I}_A)$
$\mathbf{U} \leftarrow \mathbf{I}_A^{(c)}$

▷ *Co-registration and correction of the successive frames*
for $v = 1, \ldots, V_A$ **do**
  $(\mathbf{H}_v, \mathbf{P}_v^{(r)}) \leftarrow$ COREGISTRATION$(\mathbf{U}, \mathbf{P}_v)$
  if PCE$(\hat{\mathbf{K}}, \mathbf{P}_v^{(r)}) >$ PCE$(\hat{\mathbf{K}}, \mathbf{P}_v)$ **then**
    $(\mathbf{P}_v^{(c)}, \gamma_v) =$ CORRECTION$(\hat{\mathbf{K}}, \mathbf{P}_v, \mathbf{H}_v)$
  else
    $(\mathbf{P}_v^{(c)}, \gamma_v) =$ CORRECTION$(\hat{\mathbf{K}}, \mathbf{P}_v)$
  $\mathbf{U} \leftarrow \mathbf{P}_v^{(c)}$

return vector $\boldsymbol{\gamma}$

	D02		D05		D06		D10		D14		D15		D18		D19		D20		D25		D29		D34	
	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS	TPR	ETPS
Ours $\tau_{0.05} = 19.5$	**1**	12.27	**1**	11.72	0.72	9.85	**1**	16.56	0.92	19.77	**1**	23.29	**1**	13.34	**0.91**	11.30	**1**	11.69	0.64	10.47	**1**	4.94	**1**	8.21
M2019 $\tau_{0.05} = 36$	0.87	61.61	0.62	54.08	**0.88**	52.33	0.87	51.47	0.87	51.46	0.63	60.65	0.5	47.21	0.75	38.27	0.88	37.97	**1**	37.88	0.63	53.21	0.57	44.88
MFM $\tau_{0.05} = 34$	0.89	110.13	0.89	107.33	0.78	95.05	0.89	78.42	**1**	72.29	0.78	66.32	0.89	76.50	0.89	57.54	**1**	51.97	**1**	49.50	0.67	37.58	0.55	39.88
Ours CPU $\tau_{0.05} = 19.5$	**1**	615.19	**1**	492.42	0.72	538.82	**1**	519.15	**0.92**	554.55	**1**	611.62	**1**	566.81	**0.91**	558.43	**1**	563.09	0.636	469.04	**1**	408.81	**1**	523.74

Table 1: Results obtained by the proposed method "Ours", M2019 [9] and MFM [10] in terms of TPR and ETPS (Elaboration Time Per Second of video) for a FPR=0.05 on the different devices. In **boldface** are highlighted the best results, the time is expressed in seconds.

Those operators are combined in our proposed method as formalized in Algorithm 1. Essentially, the iterative correction procedure is applied to each frame. The overall idea is to first correct the anchor $\mathbf{I}_A$ so to obtain $\mathbf{I}_A^{(c)}$; then, the successive frames $\mathbf{P}_v$ are co-registered with respect to the previous one prior to correction, obtaining for each of them a registered version $\mathbf{P}_v^{(r)}$. If $\mathbf{P}_v^{(r)}$ yields a higher PCE value than $\mathbf{P}_v$, the correction is initialized by taking into account the homography estimation.

The final decision is taken by thresholding with a value τ the mean of the vector $\boldsymbol{\gamma}$ provided by Algorithm 1.

4. EXPERIMENTAL RESULTS

We compared our method with the works presented in [9] (denoted as M2019) and [10] (denoted as MFM), whose codes are available on git-hub We measure the method performance in terms of computational cost, True Positive Rates (TPR) for a fixed False Positive Rate FPR ≈ 0.05 and Area Under the Curve (AUC).

The dataset used for the experiments is VISION [27], from which we selected horizontal videos taken using the EIS (except videos from device D23 which have very low resolution 640×480). Metadata were checked for every video to know whether it was taken upside down and, in this case, we rotated it by 180 degrees. For each device, we composed the image-based camera fingerprint with $L = 100$ flat images. Similarly to [9], it gets down-sampled and cropped, so to obtain a new fingerprint $\hat{\mathbf{K}}$ directly comparable to the test frames. We estimated offline such down-scaling and crop parameters and report them in Table 2 for all tested devices.

We analysed between eight and twelve videos per device, for a total of 131 videos. Results were obtained on a server with the following characteristics: RAM 64GB, Processor Intel(R) Xeon(R) CPU E5-2630 v3 2.40GHz, GPU NVIDIA Tesla K40c 12 GB.

	D02	*D05*	*D06*	*D10*	*D14*	*D15*	*D18*	*D19*	*D20*	*D25*	*D29*	*D34*
λ	1.333	1.455	1.417	1.333	1.455	1.416	1.454	1.417	1.227	1.933	1.455	1.455
w_{tl}	206	103	134	206	103	134	104	133	38	182	103	103
h_{tl}	345	269	291	345	269	291	269	291	216	327	269	269
w_{br}	2242	2140	2170	2242	2140	2170	2140	2170	2074	2218	2140	2140
h_{br}	1491	1414	1437	1491	1414	1437	1414	1436	1362	1437	1414	1414

Table 2: Estimated parameters for obtaining $\hat{\mathbf{K}}$ for each device starting from the image-based fingerprint. λ is the down-scaling parameters, and $(w_{\text{tl}}, h_{\text{tl}})$ and $(w_{\text{br}}, h_{\text{br}})$ are the coordinates of the top-left and bottom-right corners of the rectangular crop.

In Figure 2 we report ROC curves for our solution and state-of-the-art methods. We computed the PCE values used for Figure 2 by matching all the 131 video with its device for H1, and with a different one for H0. The improvement with respect to M2019 [9] and MFM [10] is evident both in terms of AUC and TPR, demonstrating the effectiveness of the proposed algorithm.

A more detailed comparison at acquisition device level is shown in Table 1, where we report the TPR for a fixed FPR=0.05 and the computational time required by each solution. These results prove the strong reduction of computational cost achieved with our strategy, which shows a much smaller Elaboration Time Per Second (ETPS) of video, even if analysing up to three times more frames

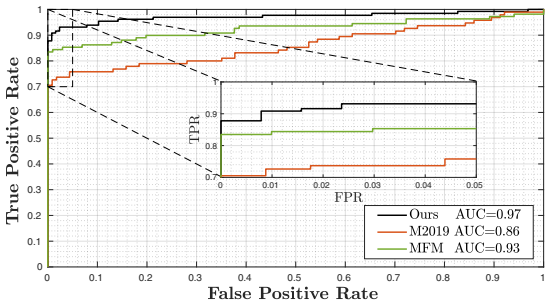

Fig. 2: ROC of the proposed method "Ours", M2019 [9] and MFM [10].

than [9, 10]. Conversely, the much higher ETPS of our method on the CPU proves the suitability of the GPU in problems with a large number of parameter combinations. While we did not have the possibility to fully match the hardware described in [9] (which might allow for a faster application of their method), the time measures in Table 1 demonstrate the significantly lower computational and hardware constraints of our algorithm. The proposed method outperforms the state-of-the-art also in terms of TPR with FPR fixed to 0.05, except for video sequences coming from the devices D06 and D25. We conjecture that their worse performance in terms of TPR is related to a noisy estimation of Eqs. (5) and (6), that we often observed on highly textured videos. However, we believe that with a more accurate keypoints selection, the identification results can be further improved and it is our purpose to investigate solutions similar to the ones proposed in [25].

5. DISCUSSION AND CONCLUSIONS

In this paper, we presented an innovative method for the identification of the source of EIS videos, where the more promising frames for this purpose are temporally localized. We did it by taking into account the inevitable temporal correlation of the EIS applied to neighbour frames and by defining a measure for the camera momentum. The results obtained so far on stabilized videos coming from VISION [27] outperform previous approaches both in terms of identification accuracy and of computational efficiency.

However, we believe there is space for improvement in different aspects. In particular, we aim at improving the model used for the estimation of the camera momentum (e.g., by exploiting strategies described in [25]) and to further optimize the use of the GPU for the EIS inversion in video frames. Another future direction could be to investigate strategies to incorporate deep networks (such as the Noiseprint [12] fingerprint extractor) in the forensics analysis, which have a high potential in improving algorithmic efficiency in testing but need to be adapted to deal with stabilization issues. Moreover, moving from the hybrid scenario where the camera fingerprint is estimated on flat images to a fully video-based one were (potentially stabilized) videos are used for getting the reference fingerprints would be of high practical relevance.

6. REFERENCES

[1] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE TIFS*, vol. 1, no. 2, pp. 205–214, 2006.

[2] D. Cozzolino and L. Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *IEEE TIFS*, vol. 15, 2020.

[3] A. E. Dirik, H. T. Sencar, and N. Memon, "Source camera identification based on sensor dust characteristics," in *2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics*. IEEE, 2007, pp. 1–6.

[4] J. Fridrich, "Digital image forensics," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26–37, 2009.

[5] M. Goljan and J. Fridrich, "Sensor-fingerprint based identification of images corrected for lens distortion," in *Media Watermarking, Security, and Forensics 2012*. International Society for Optics and Photonics, 2012, vol. 8303, p. 83030H.

[6] M. Goljan, "Digital camera identification from images–estimating false acceptance probability," in *International workshop on digital watermarking*. Springer, 2008, pp. 454–468.

[7] M. Darvish Morshedi Hosseini and M. Goljan, "Camera identification from hdr images," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2019.

[8] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*. IEEE, 1998, vol. 5.

[9] S. Mandelli, P. Bestagini, L. Verdoliva, and S. Tubaro, "Facing device attribution problem for stabilized video sequences," *IEEE TIFS*, vol. 15, pp. 14–27, 2019, Available at `https://github.com/polimi-ispl/TIFS2019-stabilized-video-attribution`.

[10] S. Mandelli, F. Argenti, P. Bestagini, M. Iuliani, A. Piva, and S. Tubaro, "A modified fourier-mellin approach for source device identification on stabilized videos," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, Available at `https://github.com/polimi-ispl/mfm-sdi-stabilized-videos`.

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

[12] D. Cozzolino, G. Poggi, and L. Verdoliva, "Extracting camera-based fingerprints for video forensics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 130–137.

[13] Fabio Bellavia, Massimo Iuliani, Marco Fanfani, Carlo Colombo, and Alessandro Piva, "Prnu pattern alignment for images and videos based on scene content," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 91–95.

[14] Fabio Bellavia, Marco Fanfani, Carlo Colombo, and Alessandro Piva, "Experiencing with electronic image stabilization and prnu through scene content image registration," *Pattern Recognition Letters*, vol. 145, pp. 8–15, 2021.

[15] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE TIFS*, vol. 3, no. 1, pp. 74–90, 2008.

[16] M. Goljan and J. Fridrich, "Estimation of lens distortion correction from single images," in *Media Watermarking, Security, and Forensics 2014*. International Society for Optics and Photonics, 2014, vol. 9028, p. 90280N.

[17] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," *Proceedings of SPIE - The International Society for Optical Engineering*, February 2009.

[18] A. Cortiana, V. Conotter, and F. GB Boato, G.and De Natale, "Performance comparison of denoising filters for source camera identification," in *Media Watermarking, Security, and Forensics III*. International Society for Optics and Photonics, 2011, vol. 7880, p. 788007.

[19] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva, "Analysis of denoising filters for photo response non uniformity noise extraction in source camera identification," in *2009 16th International Conference on Digital Signal Processing*. IEEE, 2009, pp. 1–7.

[20] 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 (SPL)*, vol. 6, pp. 300–303, 1999.

[21] X. Kang, Y. Li, Z. Qu, and J. Huang, "Enhancing source camera identification performance with a camera reference phase sensor pattern noise," *IEEE TIFS*, vol. 7, no. 2, pp. 393–402, 2012.

[22] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*. International Society for Optics and Photonics, 2008, vol. 6819, p. 68190E.

[23] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[24] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, 2005, vol. 1.

[25] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, "Sift features tracking for video stabilization," in *14th international conference on image analysis and processing (ICIAP 2007)*. IEEE, 2007.

[26] A. Bundy and L. Wallen, "Breadth-first search," in *Catalogue of artificial intelligence tools*, pp. 13–13. Springer, 1984.

[27] 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, pp. 1–16, 2017.