Blind Detection and Localization of Video Temporal Splicing Exploiting Sensor-Based Footprints

Sara Mandelli, Paolo Bestagini, Stefano Tubaro
Dipartimento di Elettronica,
Informazione e Bioingegneria
Politecnico di Milano, Milan, Italy

Davide Cozzolino, Luisa Verdoliva
Dipartimento di Ingegneria Elettrica
e Tecnologie dell'Informazione
Università Federico II di Napoli, Naples, Italy

***Abstract*—In recent years, the possibility of easily editing video sequences led to the diffusion of user generated video compilations obtained by splicing together in time different video shots. In order to perform forensic analysis on this kind of videos, it can be useful to split the whole sequence into the set of originating shots. As video shots are seldom obtained with a single device, a possible way to identify each video shot is to exploit sensor-based traces. State-of-the-art solutions for sensor attribution rely on Photo Response Non Uniformity (PRNU). Despite this approach has proved robust and efficient for images, exploiting PRNU in the video domain is still challenging.**

In this paper, we tackle the problem of blind video temporal splicing detection leveraging PRNU-based source attribution. Specifically, we consider videos composed by few-second shots coming from various sources that have been temporally combined. The focus is on blind detection and temporal localization of splicing points. The analysis is carried out on a recently released dataset composed by videos acquired with mobile devices. The method is validated on both non-stabilized and stabilized videos, thus showing the difficulty of working in the latter scenario.

I. Introduction

Manipulating visual contents has become a relatively easy task, thanks to advanced video editing software and computer graphics tools. Some forged videos are so well crafted to elude visual scrutiny even by forensic experts. Therefore, serious concerns arise about the trustworthiness of the huge number of videos flowing over the Internet. In the era of fake news, the risk of being flooded by *realistic* fake videos is very high, and definitely alarming. For this reason, there is a growing interest for automatic tools which can reliably establish video integrity.

A large number of video forensics methods have been proposed in the literature [1]. Some of them aim at detecting and localizing video copy-moves, involving the insertion or deletion of a specific video object in a sequence of frames [2], [3]. Others specialize on the manipulation of entire groups of frames [4], [5]. These methods, like many others, rely on specific prior hypotheses and hence work only for some types of manipulation. On the contrary, methods based on camera sensor noise, also known as Photo Response Non Uniformity (PRNU), are independent of the specific type of manipulation, which is why they are drawing considerable attention in both research and applications.

The PRNU pattern is caused by inhomogeneity in silicon wafers and imperfections in the sensor manufacturing process, and can be retrieved in all images or videos taken by a given camera. Originally, it was used for image forensic tasks, like source identification or image forgery detection [6], [7]. Very soon, however, it was also exploited for video source attribution [8], and also to help identifying duplicate and modified video copies [9], [10].

The extension of PRNU-based methods to video, however, is not straightforward, and several peculiar issues need to be properly addressed. PRNU estimation is a much harder task for videos than for images, since videos are usually compressed with relatively low quality, compromising the sensor footprints. Gaining robustness against compression is one of the main focus of current research. For example, to face the effects of strong compression, in [11] it is proposed to reorder and weigh video frames according to their reliability (I-frames turn out to be more reliable than P-frames for PRNU estimation). Another major problem is video stabilization, that causes misalignments of individual pixels across frames. This is a serious issue since the video fingerprint cannot be estimated by misaligned frames. In addition, even if available, it may not correlate with the noise residual extracted from a given stabilized frame [12]. Since many modern smart-phone cameras adopt video stabilization, PRNU-based methods are not effective anymore [13] in the absence of suitable countermeasures.

Even neglecting the above problems, to estimate reliably the PRNU pattern, a large number of videos should be available. Unfortunately, this is not always the case. Quite often, one is required to work in a *blind* setting, analyzing a single video of unknown origin downloaded from the net. In this situation, one can use some of the video frames to estimate the PRNU, but the quality will significantly impair, not only for the limited number of available frames but also because of their content correlation. A possible approach is to work on noise residuals estimated from the single video and extract as much information as possible from them. In [14] some initial video frames

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

ISBN 978-90-827970-1-5 © EURASIP 2018

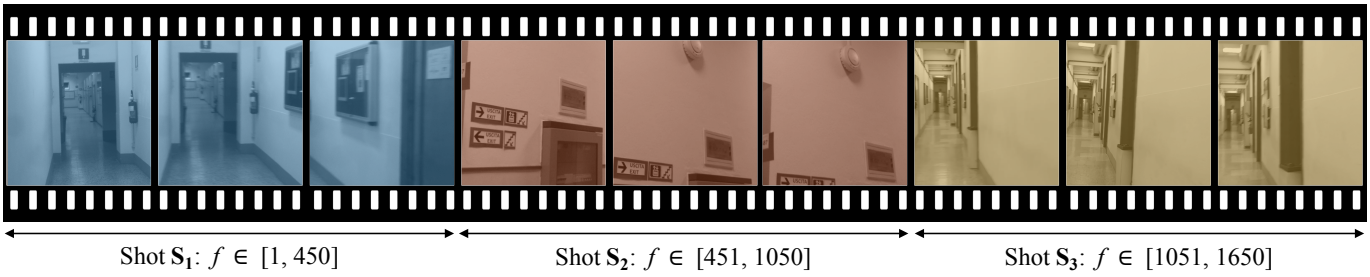

Fig. 1: Video sequence composed of 3 shots coming from different devices.

are used to estimate a reference pattern and check for video authenticity. In [15] the temporal correlation of noise residuals is analyzed through a Gaussian mixture model, while in [16] the inconsistencies of the photon shot noise characteristics are used for forgery detection. In [17] the noise residuals of a few pristine frames are used to extract reference features characterizing the video source. When features extracted from new frames depart from the reference an anomaly is revealed, which can be used for forgery localization.

Here, we address the detection of video temporal splicing, which arises when two or more video shots are used to compose a new video. As in [14], the noise residuals of the initial video frames are used to extract a reference pattern (a coarse PRNU estimate), which is used in turn to detect the presence and position of a possible splicing. The process is then iterated, with the aim to detect eventually the precise combination of different shots and their temporal composition. To the best of our knowledge, this is the first time this manipulation is considered in the literature.

II. Problem Formulation

The PRNU is a noise fingerprint characteristic of any image and video acquisition device. Specifically, PRNU is introduced in all acquired images and video frames as a multiplicative zero-mean noise pattern [6], [7]. Narrowing down the research field to non-stabilized videos, PRNU can be estimated from a set of J frames $\mathbf{I}_j$, $j \in [1, J]$ coming from the same device [7] as

$$\mathbf{K} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{I}_j \Big/ \sum_{j=1}^{J} \mathbf{I}_j^2 \,, \tag{1}$$

where $\mathbf{W}_j$ is the noise residual extracted from $\mathbf{I}_j$, and all operations are performed pixel-wise. Precisely, $\mathbf{W}_j = \mathbf{I}_j - \dot{\mathbf{I}}_j$, being $\dot{\mathbf{I}}_j$ a denoised version of $\mathbf{I}_j$ computed as suggested in [7]. Given a frame $\mathbf{I}$, we can infer whether it has been captured by a certain camera computing the Normalized Cross-Correlation (NCC) between $\mathbf{W}$ and the camera PRNU pixel-wise scaled by $\mathbf{I}$, denoted as $\mathrm{NCC} = \rho(\mathbf{W}, \mathbf{KI})$. If NCC is higher than a confidence threshold, $\mathbf{I}$ is attributed to that camera [6], [7].

The goal of this paper is the blind detection and localization of video temporal splicing, leveraging PRNU-based source attribution for splitting the video into the set of originating shots. Formally, let us consider a video $\mathbf{V}$ modeled as the temporal concatenation of an unknown number N_s of shots $\mathbf{S}_n$, $n \in [1, N_s]$, i.e., $\mathbf{V} = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{N_s}\}$. Each shot $\mathbf{S}_n$ is composed by an unknown number of frames recorded from a single device. Devices are assumed to be unknown. Fig. 1 shows an example of video compilation composed by three shots. Our goal is to estimate the amount of shots N_s, and segment the video $\mathbf{V}$ into its originating shots $\mathbf{S}_n$.

In principle, if each shot corresponds to a single acquisition device whose PRNU is known, it could be possible to aggregate frames sharing significant correlation with each PRNU, thus detecting and localizing the various splices. However, we consider the challenging scenario in which shots' PRNUs are not available, as we do not know the camera models exploited for generating the video compilation under analysis. To overcome this problem, we propose an algorithm to estimate the various camera fingerprints directly from video frames, in an iterative fashion. This allows to blindly identify how many shots generate the compilation, and localize the splicing portions. In the next section, we report a detailed description of the pipeline.

III. Proposed Method

As previously stated, every analyzed video $\mathbf{V}$ is the combination of various shots with distinct characteristics. More specifically, we are completely unaware of the number of involved devices, related camera models, and number of frames of each splicing shot. In this section, we show the rationale driving the proposed method through an example, followed by an exhaustive description of the algorithm.

Let us suppose we randomly select from the whole sequence a reference frame $\mathbf{I}_r$ and extract its noise residual $\mathbf{W}_r$ [7]. This frame belongs to a random shot, thus its noise $\mathbf{W}_r$ is supposed to correlate only with noise residuals extracted from other frames of the same shot. By scanning all video frames $\mathbf{I}_f$, $f \in [1, N_f]$ and extracting the relative noise $\mathbf{W}_f$, we define the cumulative sample mean noise as

$$\overline{\mathbf{W}}(f) = \sum_{i=1}^{f} \mathbf{W}_i \Big/ f \,. \tag{2}$$

This cumulative noise contains information about noises extracted from all frames until the f-th one. If $\mathbf{V}$ is generated from a single shot, $\overline{\mathbf{W}}(f)$ is directly related to the PRNU of the recording camera as defined in (1). In case of multiple shots, $\overline{\mathbf{W}}(f)$ contains averaged information about different shots' fingerprints, depending on f.

ISBN 978-90-827970-1-5 © EURASIP 2018

Taking into account these considerations, we can solve the camera-attribution problem between the available fingerprint estimate $\overline{\mathbf{W}}(f)$ and the reference frame $\mathbf{I}_r$. It is worth noticing that, following the theory in Sect. II and computing the frame-variant NCC denoted as $c(f) = \rho(\mathbf{W}_r, \overline{\mathbf{W}}(f)\mathbf{I}_r)$, we can observe this behavior:

- If $f < r$ and the considered f frames do not belong to the same shot of $\mathbf{I}_r$, $c(f)$ is low and more or less constant. As a matter of fact, $\overline{\mathbf{W}}(f)$ is a completely wrong estimate of the fingerprint related to the reference shot and does not correlate with $\mathbf{W}_r$.
- At a given $f \leq r$, $\overline{\mathbf{W}}(f)$ starts being built exploiting noise residuals from frames belonging to the very same device of $\mathbf{I}_r$. Hence, $\overline{\mathbf{W}}(f)$ starts matching $\mathbf{W}_r$, and $c(f)$ begins to increase.
- After all frames of the reference device have been scanned (i.e., the f-th and r-th frames come from different devices), $c(f)$ starts dropping, since $\overline{\mathbf{W}}(f)$ begins containing contributions from noises not correlating anymore with $\mathbf{W}_r$.

For the sake of clarity, we report in Fig. 2 an example of $c(f)$ behavior over a video composed by three splicing portions (shown in Fig. 1):

- $\mathbf{S}_1$, composed by frames $\mathbf{I}_f, f \in [1, 450]$;
- $\mathbf{S}_2$, composed by frames $\mathbf{I}_f, f \in [451, 1050]$;
- $\mathbf{S}_3$, composed by frames $\mathbf{I}_f, f \in [1051, 1650]$.

If the reference frame is $\mathbf{I}_{100}$ (i.e., belonging to $\mathbf{S}_1$), $c(f)$ increases up to $f = 450$, then it starts dropping as frames after $\mathbf{I}_{450}$ do not belong to $\mathbf{S}_1$ anymore. If the reference frame is $\mathbf{I}_{700}$ (i.e., belonging to $\mathbf{S}_2$), $c(f)$ is almost flat for $f \leq 450$ (i.e., frames belonging to $\mathbf{S}_1$), shows an increasing behavior for $450 < f \leq 1050$ (i.e., frames belonging to $\mathbf{S}_2$), then it drops again for $f > 1050$ (i.e., frames belonging to $\mathbf{S}_3$). A coherent behavior can be observed if we consider reference frame $\mathbf{I}_{1300}$.

Bearing this in mind, the proposed pipeline for blind detection and localization of temporal splicing consists of the following steps: (i) *selecting the reference frame* – randomly select one reference frame from the video and compute $c(f), f \in [1, N_f]$; (ii) *clustering frames* – group together frames for which $c(f)$ locally increases and delete the selected group from the entire video; iterate steps (i) and (ii) until almost all video frames have been clustered in different groups; (iii) *clustering shots* – to counteract the problem of over-estimating the number of splicing shots, cluster the groups of frames with higher inter-correlation; (iv) *assigning left-out frames* – assign the remaining frames to the best-matching shot. It follows an exhaustive description of each step.

A. Selecting the Reference Frame

Since information about temporal segmentation is not available, the only way for selecting the reference frame is to pick it up randomly. Actually, interpretation of $c(f)$ is not always straightforward like in Fig. 2. As a matter of fact, correlation $c(f)$ can exhibit an increasing behavior even for frames not belonging to the same shot of $\mathbf{I}_r$, as well as multiple local maxima (e.g., due to correlated frame content). Therefore, to increase the algorithm's robustness, we perform multiple experiments, picking up a pool of different reference frames.

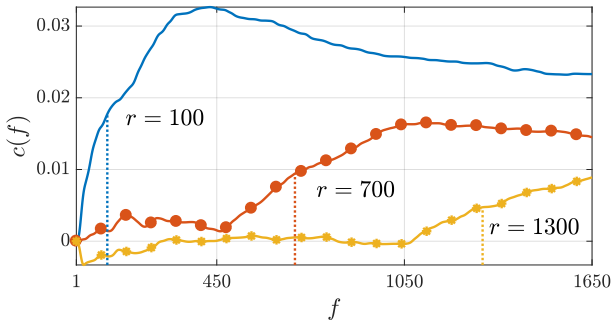

Fig. 2: Example of $c(f)$ behavior over the video shown in Fig. 1.

The algorithm extracts R possible $\mathbf{I}_r$ frames, and computes $c_r(f)$ for each realization $r \in [1, R]$. We define three quantities useful to evaluate $c_r(f)$ goodness:

- The maximum of $c_r(f)$, defined as $\mathcal{M}_r = \max_f (c_r(f))$.
- The frame index related to the maximum $c_r(f)$ value, defined as $m_r = \arg\max_f (c_r(f))$.
- The largest set of frame indexes for which $c_r(f)$ shows a monotonically increasing behavior, defined as Δ_r.

The best reference $\tilde{r}$ out of the R ones is selected as the realization with highest $\mathcal{M}_r$, given that $m_r \in \Delta_r$. This ensures that frames whose index lies in Δ_r belong to a single device. In our experiments, we chose $R = 10$ as a good trade-off between algorithm's robustness and efficiency.

B. Clustering Frames

Once the best realization $\tilde{r}$ has been selected, we average noise residuals of frames belonging to $\Delta_{\tilde{r}}$, in order to estimate a fingerprint $\hat{\mathbf{K}}_n$ which will be related to a new shot $\hat{\mathbf{S}}_n$.

To cluster frames together, we follow the standard PRNU-based source attribution pipeline: being $\hat{\mathbf{K}}_n$ the estimated fingerprint, noises from all video frames are correlated with $\hat{\mathbf{K}}_n$. We assign to the new shot $\hat{\mathbf{S}}_n$ all frames for which NCC is above a predefined threshold.

Next operation consists in removing the estimated group of frames from the video sequence, and iterate steps (i) and (ii) until remaining frames are less than a default value (e.g., 100 in our experiments).

C. Clustering Shots

Estimation of true fingerprint from a small subset of frames is far from being an easy task. For this reason, it sometimes happens that frames belonging to the same original shot are not clustered together, due to low correlation values. Therefore, we usually end up with an estimated compilation $\hat{\mathbf{V}} = \{\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, ..., \hat{\mathbf{S}}_{M_s}\}$, whose number of shots M_s is higher than the true one (i.e., $M_s > N_s$).

Some control on over-estimation is thus necessary. On the other hand, it is still better over-segmenting the compilation

ISBN 978-90-827970-1-5 © EURASIP 2018

than clustering shots of different sources. To this purpose, we propose a clustering strategy for blindly grouping shots wrongly split:

- We compute the reference noise pattern $\hat{\mathbf{K}}_n$ for each estimated shot in $\hat{\mathbf{V}}$.
- We correlate through NCC all pairs of reference noise patterns.
- We cluster different shots if and only if each shot of the cluster has pairwise NCC with *all* other shots greater than a threshold Γ, and the cluster is composed by temporally adjacent shots.
- The estimated video sequence $\hat{\mathbf{V}}$ now includes a reduced set of shots, whose fingerprints are the average of noise patterns inside the same cluster.

This procedure is iterated manifolds, until no more shots are aggregated.

D. Assigning Left-Out Frames

At this step, the compilation $\hat{\mathbf{V}} = \{\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, ..., \hat{\mathbf{S}}_{L_s}\}$ includes the reduced set of L_s shots, while at most 100 remaining frames have not been assigned to any contribution. The easiest way for labeling them is to apply the standard PRNU-based source attribution pipeline. We assign each singleton frame to the shot whose estimated fingerprint better correlates with the frame noise residual.

IV. Experiments and Validation

In this section we first introduce the datasets used for experiments, then we report the achieved results.

A. Datasets

Splicing portions have been collected from a recently released dataset, acquired with more than 30 mobile devices [13]. More specifically, we created two distinct datasets for non-stabilized and stabilized compilations. For the sake of brevity, from now on we describe the generation process of non-stabilized compilations, but procedure still remains the same for both cases.

From dataset [13], we select non-stabilized devices with minimum Video Resolution set to HD-Ready (VR $\geq 720p$). Then, 5 videos per device are collected, randomly picking from indoor/outdoor scenarios and considering only move/panrot acquisition modes. This choice comes from the idea of generating plausible results, since combinations of flat or static videos are actually less likely to be found.

For each device, we cut the 5 selected videos at frame index $150, 300, 450, 600, 750$, respectively, in order to generate splicing portions of different lengths. The splicings are then cropped to common resolution of 720×720 pixels and gray-scale converted. Since the available non-stabilized devices are 19, we end up with a pool of 95 distinct splices.

The final video compilation is obtained as the temporal concatenation of $N_s \in [3, 6]$ splicing portions, randomly extracted from the pool. Following this pipeline we generated two datasets, covering 150 non-stabilized videos and just as many stabilized.

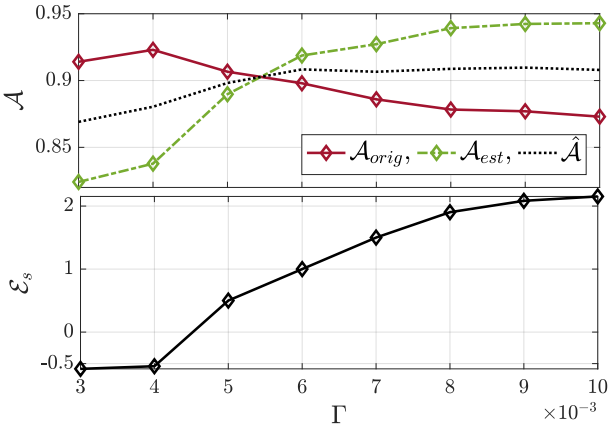

Fig. 3: Results for non-stabilized video compilations.

Note that datasets also include challenging situations, e.g., one compilation can contain: (i) 2 or more splicing portions with different scene content but belonging to the very same device; (ii) content-similar portions coming from different sources. Since we aim at estimating the temporal splicing due to device change, not to scene, in case (i) contributions of the same source are labeled as a single unique splice.

B. Evaluation Metrics

We developed two kinds of accuracy measures for inferring the quality of proposed method in splicing localization. Specifically, $\mathcal{A}_{orig}$ and $\mathcal{A}_{est}$ are defined as:

- $\mathcal{A}_{orig}$: for each *original* shot, $\mathcal{A}_{orig}$ is the percentage of frames belonging to that shot which actually have been labeled as a unique cluster in the estimation process. This measure detects presence of over-segmentation.
- $\mathcal{A}_{est}$: for each *estimated* shot, $\mathcal{A}_{est}$ is the percentage of frames belonging to that shot, which effectively belongs to a unique original splice. It decreases in case of under-segmentation.

Concerning quality evaluation in identifying the number of shots in the compilation, it is paramount to take into account previous considerations made in Sect. III-C. Our goal is to reduce as much as possible the error in the amount of estimated shots, still favoring over-segmentation in order not to mix various devices together. We define $\mathcal{E}_s$ as the error in estimating the number of shots which generates the video.

Accuracies and $\mathcal{E}_s$ are averaged over the total amount of contributions in a single compilation.

C. Results

We show results in terms of mean $\mathcal{A}_{orig}, \mathcal{A}_{est}, \mathcal{E}_s$ over the two datasets of non-stabilized and stabilized videos. More specifically, we evaluate these measures for different values of threshold Γ exploited for clustering splices.

Fig. 3 depicts outcomes for non-stabilized compilations. The more Γ increases, the less splices are clustered. Hence, while accuracy $\mathcal{A}_{orig}$ decreases for over-segmentation, $\mathcal{A}_{est}$ gets approximately to 0.95. Note that we must necessarily

ISBN 978-90-827970-1-5 © EURASIP 2018

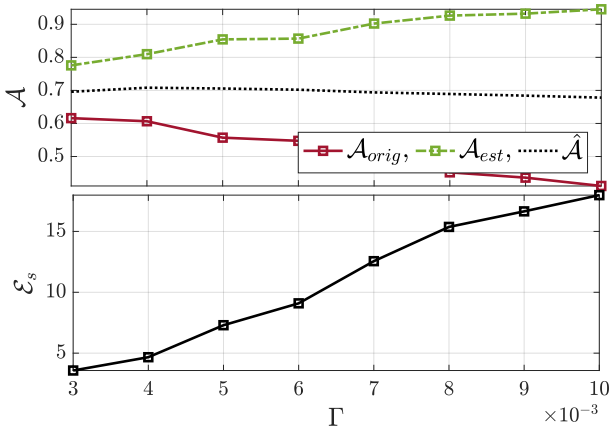

Fig. 4: Results for stabilized video compilations.

investigate accuracy behavior together with $\mathcal{E}_s$, otherwise we could fall into several interpretation mistakes. Higher values for $\mathcal{A}_{est}$ are feasible only as long as $\mathcal{E}_s$ does not excessively grow. For this reason we introduce a new accuracy measure, averaging $\mathcal{A}_{orig}$ and $\mathcal{A}_{est}$ versus Γ, ending up with $\hat{\mathcal{A}}$.

We think of $\hat{\mathcal{A}}$ representing a good measure for the selection of best Γ for clustering. Indeed, $\hat{\mathcal{A}}$ takes into account both over-segmentation risk (highlighted by $\mathcal{A}_{orig}$) and under-segmentation risk (stressed by $\mathcal{A}_{est}$). In light of this, we note that the best threshold values are $\Gamma = \{5, 6\} \times 10^{-3}$, which guarantee $\hat{\mathcal{A}}$ around 0.9, and segmentation error $\mathcal{E}_s$ below $+1$ over-estimated splices on average.

As far as stabilized compilations are concerned, results are shown in Fig. 4. Note that, in this situation, the PRNU-based source attribution approach is severely hindered. As a matter of fact, $\mathcal{A}_{orig}$ is always well below 0.7. On the other hand, $\mathcal{A}_{est}$ achieves very good measures, reaching scores about 0.94. We must beware of this result: the behavior of $\mathcal{E}_s$ is far from being acceptable, as $\mathcal{E}_s > 4$ for all thresholds. This means that we are actually over-segmenting shots very often. However, as PRNU estimation is known to be a challenging task for stabilized videos, these results are expected.

V. Conclusions

In this paper we considered the problem of blind detection of video splicing exploiting PRNU-related traces. In particular, we considered the analysis of a video compilation composed by an unknown amount of shots coming from an unknown amount of different devices. The proposed algorithm estimates the number of shots, as well as their starting and ending points in time.

The proposed method is an iterative algorithm leveraging the idea that noise traces (related to PRNU) extracted from different frames within a sequence should correlate only if frames have been acquired with the same device. It is therefore possible to iteratively group frames generated from the same device, eventually estimating all frame clusters, i.e., different shots.

Validation is carried out on a dataset of videos acquired with modern smart-phone devices in order to simulate a real-world scenario. Despite the promising results obtained on non motion-stabilized video sequences, this study confirms that video stabilization is a highly-corruptive operation in terms of PRNU-based detectors. As a matter of fact, our method's accuracy drops if considered sequences are stabilized.

In light of this, future work will be devoted to the estimation of robust noise camera fingerprints from stabilized videos. Indeed, this technology is rapidly spreading among many device vendors, thus making video device attribution an even more challenging task in the near future.

References

[1] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, Dec. 2012.

[2] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Local tampering detection in video sequences," in *IEEE International Workshop on Multimedia Signal Processing*, Oct. 2013, pp. 488–493.

[3] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A PatchMatch-based dense-field algorithm for video copy-move detection and localization," *IEEE Transactions on Circuits and Systems for Video Technology, in press*, 2018.

[4] M.C. Stamm, W.S. Lin, and K.J. Ray Liu, "Temporal Forensics and Anti-Forensics for Motion Compensated Video," *IEEE Transactionso on Information Forensics and Security*, vol. 7, no. 4, pp. 1315–1329, Aug. 2012.

[5] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detection frame deletion and insertion," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2014, pp. 6226–6230.

[6] J. Lukàš, 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.

[7] M. Chen, J. Fridrich, M. Goljan, and J. Lukàš, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, pp. 74–90, 2008.

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

[9] S. Bayram, H.T. Sencar, and N. Memon, "Video copy detection based on source device characteristics: a complementary approach to content-based methods," in *1st ACM international conference on Multimedia Information Retrieval*, 2008, pp. 435–442.

[10] S. Lameri, L. Bondi, P. Bestagini, and S. Tubaro, "Near-duplicate video detection exploiting noise residual traces," in *IEEE Internatioonal Conference on Image Processing*, 2017.

[11] W.-H. Chuang, H. Su, and M. Wu, "Exploring compression effects for improved source camera identification using strongly compressed video," in *IEEE Internatioonal Conference on Image Processing*, 2011, pp. 1993–1996.

[12] S. Taspinar, M. Mohanty, and N. Memon, "Source camera attribution using stabilized video," in *IEEE Workshop on Information Forensics and Security*, 2016.

[13] 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*, pp. 1–16, 2017.

[14] N. Mondaini, R. Caldelli, A. Piva, M. Barni, and V. Cappellini, "Detection of malevolent changes in digital video for forensic applications," in *Proc. of SPIE Conference on Security, Steganography and Watermarking of Multimedia*, 2007, vol. 6505.

[15] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu, "Video forgery detection using correlation of noise residue," in *IEEE 10th Workshop on Multimedia Signal Processing*, 2008, pp. 170–174.

[16] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting forgery from static-scene video based on inconsistency in noise level functions," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 883–892, Dec. 2010.

[17] P. Mullan, D. Cozzolino, L. Verdoliva, and C. Riess, "Residual-based forensic comparison of video sequences," in *IEEE Internatioonal Conference on Image Processing*, 2017.

ISBN 978-90-827970-1-5 © EURASIP 2018