RESIDUAL-BASED FORENSIC COMPARISON OF VIDEO SEQUENCES

Patrick Mullan[†] *, Davide Cozzolino*[‡]*, Luisa Verdoliva*[‡]*, Christian Riess*[†]

[†] Computer Science 1, University of Erlangen-Nuremberg
Martensstr. 3, 91058 Erlangen, Germany
[‡] DIETI, University Federico II of Naples
Via Claudio 21, 80125 Napoli, Italy

ABSTRACT

Video content can be acquired with off-the-shelf hardware, and is thus increasingly used to record events. With the growing role of video data for communicating to a large audience, we need tools to ensure the authenticity of video content. However, until now, only few methods exist to forensically analyze videos.

In this work, we propose a method for statistically comparing two video sequences. Per sequence, intra- and inter-frame residuals are computed. Optical flow is used to compensate for motion artifacts on inter-frame residuals. We use one sequence to build a statistical model, and compare it to the second sequence. From a forensic perspective, the proposed method enables two applications. First, manipulations can be accurately localized if both sequences are subsequences of the same video. Second, source cameras can be distinguished if both sequences stem from different videos. The proposed method is evaluated on collected smartphone data and green-screen splices. Further, it is quantitatively compared to both a recent PRNU-based approach and a technique based on autoencoders.

Index Terms— Video Forensics, Chroma Keying, Noise Residuals, SPAM Features, Optical Flow, Video Splicing

1. INTRODUCTION

The widespread of sophisticated video processing tools allows for a fast and effective manipulation of videos. Visual content may be changed for amusement, but also for malicious purposes, for example to modify evidence in court, make propaganda, or blackmailing[1].

Video forensics is an emerging research direction with the goal of exposing video manipulations and to attribute videos to source cameras [1]. Source identification has been investigated using sensor noise [2], and in particular photo-response non-uniformities (PRNU) [3, 4]. Manipulation detection has been addressed in several variants, e.g., for detecting insertion or deletion of frame groups [5–8], or copy-move forgeries [9,10]. There exists also methods that exploit codec-specific properties, e.g., in MPEG [11–13].

In this work, we propose a method to statistically compare two video sequences. We see two useful applications for the proposed scheme. First, it can be used for weak source identification, i.e., to check whether two videos are created with the same processing stack. Second, it can be used for manipulation localization: if a subsequence of a video is suspected to contain a manipulation, the method can assess the statistical consistency of this subsequence with an unsuspicious subsequence of the same video.

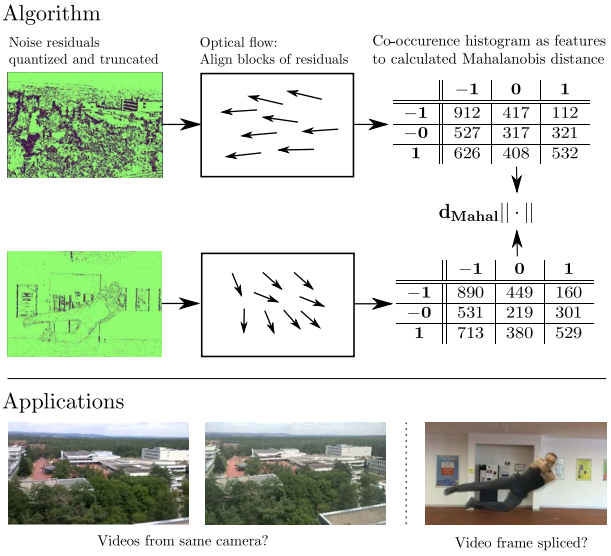

Fig. 1. Top: Simplified illustration outlining the presented algorithm. Bottom: Examples of possible usage.

Related methods include the work on photo-response non-uniformity [14, 15], which is a powerful training-based approach. However, for video, the approach is challenged by automated processing such as video stabilization [16]. Xu *et al.* propose a method that detects video compression inconsistencies on manually segmented foreground and background regions [17]. This method makes use of the DCT, and as such performs well on MPEG encoded videos. However, it is unknown how well the algorithm performs on other video codecs such as H.264. Su *et al.* detect splicing via inconsistencies in the camera's Bayer pattern interpolation along object edges [18]. Nonetheless, detecting the relatively weak Bayer pattern along relatively few edge pixels can be challenging, particularly when the video is recoded or automatically stabilized. Hsu *et al.* exploit the statistical properties of noise residuals [19]. Temporal correlation is modeled as a Gaussian mixture per block. One limitation of this approach is that it considers only two consecutive frames, and hence can only discover the first and last frame of a

This work was partially funded by the Research Training Group 1773 "Heterogeneous Image Systems, funded by the German Research Foundation (DFG). This material is based on research sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

[1] http://indianexpress.com/article/sports/cricket/someone-trying-to-blackmail-me-with-tampered-video-sharjeel-khan-2797493/

manipulation. Features extracted from noise residuals are also used in [20]. However, they worked with a very high-dimensional feature vector proposed in [21]. This requires a large database for training, and is computationally very expensive. Recently, D'Avino *et al.* used autoencoders and recurrent neuronal networks to detect splicings in videos [22]. This technique shows good results, but it lacks the capability to track objects in the video. Tracking is beneficial to differentiate between foreground and background objects to ultimately avoid merging pristine and spliced parts to one entity.

The biggest advantage of the proposed approach is that it performs a black-box comparison of the videos. It statistically relates noise residuals of two video sequences. If the software settings or the processing chain of the video sequences differ, the statistics differ. If applied within the same video, local manipulations can be detected — assuming that a consistent video has only been subject to global processing like recoding. It operates directly on the provided video sequences, and does not require additional training data.

2. RESIDUAL-BASED VIDEO MANIPULATION DETECTION

The core idea of residual-based media forensics is to compute a statistical descriptor from noise. To this end, the image is first high-pass filtered to obtain residuals. Descriptors are computed from blocks of residuals, and are used to build a statistical model. A schematic overview of the proposed method is shown on top of Fig. 1. First, residuals are computed. Then, residuals are temporally aligned using optical flow vectors. For the first video, a model is computed from spatial and temporal histograms of residual co-occurences. The Mahalanobis distance between the model and co-occurence histograms of the second video allows to compare two video sequences.

2.1. Computation of the Statistical Descriptor

We use a 1-D high-pass filter $h(u)$ with coefficients $[1, -3, 3, -1]$ along rows, columns, and temporal direction to compute the residuals. Among other filters suggested in [21], this turns out to be a good tradeoff between filtering complexity and residual quality [23].

To suppress strong edges from image content, residuals are truncated and quantized [21], such that the residual vector $I_R(c)$ at position c, is

$$I_R(c) = \min\left(t,\ \max\left(-t,\ \left\lfloor \frac{(I * h)(c)}{q} \right\rceil\right)\right) \ , \qquad (1)$$

where I denotes a 1-D slice of the video, h the high-pass filter from above, t the truncation threshold and q the quantization factor. When the intensities are represented as values between 0 and 255, it is reasonable to select small positive integer values for t and q. In our experiments, we set $t = 2$ and $q = 3$. This leads to $2t + 1$ possible values. Thus, for n neighboring residuals, there exist $(2t + 1)^n$ different combinations of values, which is also referred to as co-occurrence. We chose $n = 4$ in accordance with other recent forensic works that are based on co-occurrence features [22,24]. Mirrored co-occurrences like $[0, 2, 0, 1]$ and $[1, 0, 2, 0]$ are considered identical. Thus, the number of individual co-occurences reduces to 169. We compute histograms of co-occurences along rows, columns and frames of the video. These histograms are concatenated to form a $3 \cdot 169 = 507$ dimensional statistical block descriptor, on windows of size 128×128 pixels, with an step size of 8 pixels.

However, it may be suboptimal to directly accumulate co-occurences from the same pixel position in temporal direction. In the case of inserted objects, statistics of foreground and background

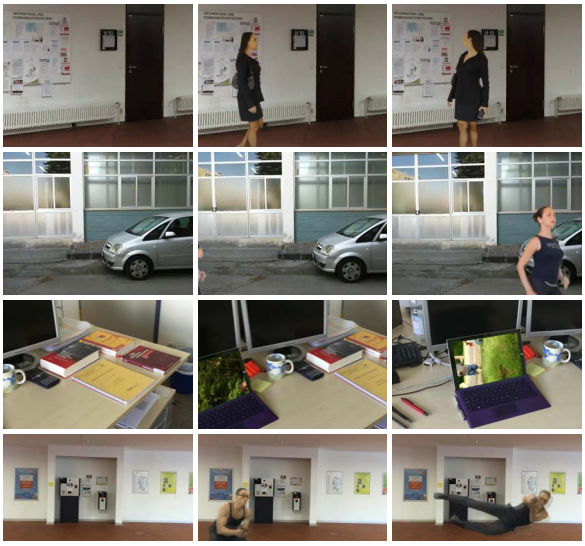

Fig. 2. Example frames from our chroma-key spliced videos. Top to bottom: `woman`, `runner`, `laptop`, and `vanDamme`.

mix if motion is not taken into account. We use optical flow to compensate for such motion effects, and to increase the likelihood that all feature vectors are either computed on background or on the overlay. More specifically, we estimate per pixel a dense motion field that tracks motion from one frame $z - 1$ to the next frame z by computing a translation vector $(v_x, v_y)^{\mathrm{T}}$ such that

$$I(x, y, z) = I(x + v_x, y + v_y, z - 1) \ . \qquad (2)$$

To estimate such a motion field, we relied on the publicly available implementation by Ce Liu [25].

Using these motion estimates, the co-occurence histogram in temporal direction is built by following the estimated motion trajectory. More precisely, we compute a seven-frame block of 3-D features, where pixels from consecutive frames follow the optical flow translation $(v_x, v_y)^{\mathrm{T}}$.

2.2. Comparison of Two Video Sequences

To compare descriptors from two video subsequences, we select all descriptors from one sequence, and compute their mean $\vec{\mu}$ and covariance Σ, after passing the feature vectors through a square-root non-linearity. Then, each descriptor of the other sequence is evaluated on these statistics by computing the Mahalanobis distance [24].

Thus, let $\vec{f}_i$ denote the i-th descriptor from the first video sequence. The model $(\vec{\mu}, \Sigma)$ is constructed by computing

$$\vec{\mu} = \frac{1}{N}\sum_{i=1}^{N} \vec{f}_i \ , \quad \Sigma = \frac{1}{N}\sum_{i=1}^{N} (\vec{f}_i - \vec{\mu})(\vec{f}_i - \vec{\mu})^{\mathrm{T}} \ . \qquad (3)$$

Let furthermore $\vec{g}$ denote a descriptor from the second video. Then, the Mahalanobis distance d_{Mahal} is

$$d_{\mathrm{Mahal}}(\vec{g}; \vec{\mu}, \Sigma) = \sqrt{(\vec{g} - \vec{\mu})^{\mathrm{T}} \Sigma^{-1} (\vec{g} - \vec{\mu})} \ . \qquad (4)$$

The larger the mismatch between $\vec{g}$ and the model, the larger the Mahalanobis distance. For a perfect match, i.e., if $\vec{g} \equiv \vec{\mu}$, $d_{\mathrm{Mahal}}(\vec{g}; \vec{\mu}, \Sigma)$ equals zero.

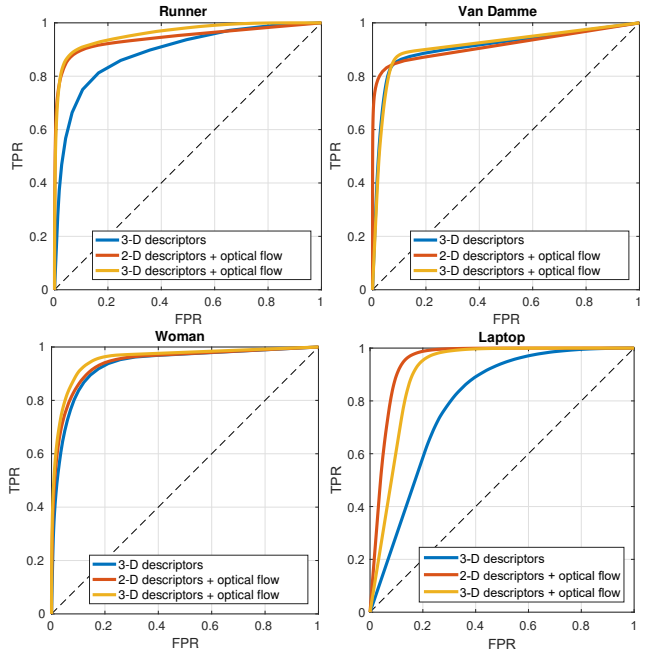

Fig. 3. Detection performance on the spliced videos without secondary compression.

3. DATASETS

3.1. Spliced Video Dataset

The spliced video dataset is compiled using green screen technique. For three videos, we collected samples for video overlays from YouTube and used smartphones to capture suitable backgrounds. A fourth example was created by recording a sequence containing a green notebook screen, in which we inserted a video. Three example frames from each of these videos are shown in Fig. 2. From top to bottom, we denote these videos as `woman`, `runner`, `laptop`, and `vanDamme`. The resolution of these videos is between 640×480 pixels and 1920×1080 pixels. All videos have between 150 and 170 frames.

3.2. Smartphone Dataset

We collected short video sequences from nine smartphones. The camera models were two Motorola G1LTE (1,2), Motorola X Play (3), HTC One M8 (4), iPhone 5s (5), iPhone 6 (6), Microsoft Lumia 820 (7), One Plus 2 (8), and Xiaomi Redmi Note 3 Pro (9). Smartphone capturing settings were not adapted in any way. All videos contain roughly the same content. A landscape as seen from a high rise while performing a camera sweep. Two example frames are shown in Fig. 1.

From each video, we arbitrarily cropped 660×330 central pixels from 120 frames. The first 50 frames of each video are reserved for building the model. Data after frame 62 was used to compute descriptors for testing against other smartphone models. No resampling, frame interpolation or other processing was applied to the input videos to ensure that only video artifacts are compared.

4. EVALUATION

For comparison, we evaluate three variants of the proposed method. First, we use 2-D features, computed only along x and y direction, with temporal alignment via optical flow. Second and third, we use 3-D features (computed in x, y and temporal direction) without and with temporal alignment via optical flow. As metric we report receiver operating characteristic curves (ROCs). Since the features are extracted with a stride width of 8 pixels, we upsample our result by that factor for pixelwise comparison with the ground truth.

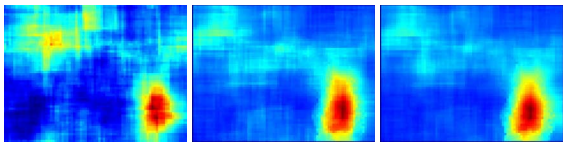

Fig. 4. Mahalanobis distances on `runner`, corresponding to the rightmost image in the second row of Fig. 2. From left to right: 3-D, 2-D with optical flow and 3-D with optical flow.

4.1. Manipulation Localization

As a baseline result, Fig. 3 shows the detection performance on the spliced videos without secondary recompression. The difficulty of these four videos greatly varies: the easiest case is `vanDamme` (top right), the hardest case is `laptop` (bottom right). The performance variation stems from several effects. First, camera egomotion (i. e., a globally moving background) makes it more difficult to build the model. Second, very dark or very bright pixels in the scene also complicate to obtain meaningful statistics. Both of these effects occur in `laptop`, which is why we consider this a particularly difficult case. Conversely, strong compression artifacts in the spliced foreground of `vanDamme` help the algorithm to distinguish background and overlay. The two remaining videos `runner` and `woman` range in-between, with a detection performance that is overall quite reliable.

In `vanDamme` and `laptop`, 2-D descriptors with optical flow perform best, in `runner` and `woman`, 3-D descriptors with optical flow perform best. In every case, the inclusion of optical flow considerably improves the results. Fig. 4 shows a qualitative example for the resulting Mahalanobis distances for frame 150 of the video `runner`. The leftmost image shows 3-D descriptors. The other two images present results with optical flow, for 2-D (middle) and 3-D (right). For the remainder of the experiments, we use 3-D descriptors with optical flow as the proposed method.

4.1.1. Number of Frames for Computing the Model

We investigate how the number of frames used for building the model affects the performance of forgery localization. We report the ROC curves averaged over all four videos using the proposed 3-D features with optical flow. The results are shown in Fig. 5 (left). Performance gently decays when reducing the number of frames for building the model. This provides some degree of flexibility to the analyst if less than 50 genuine frames are available. Good performance is achieved already for as few as fifteen frames.

4.1.2. Evaluation of Re-Compression

We evaluated the effect of secondary compression on the spliced videos. The videos were recompressed with `ffmpeg`[2]. Analogously to the previous Section, we report averaged ROC curves for the proposed method. Fig. 5 (right) shows that localization performance is

[2]Example recompression command: `ffmpeg -i input.avi -c:v mpeg4 -vtag xvid -qscale:v 7 output.avi`

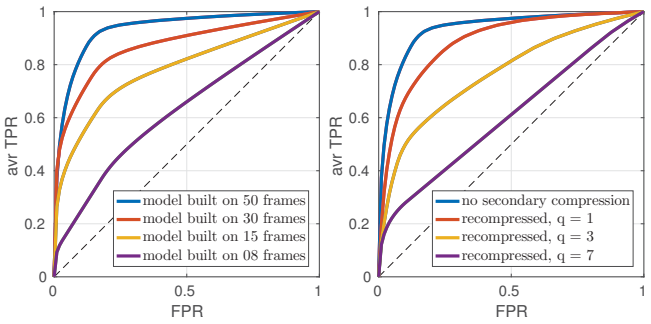

Fig. 5. Average performance on spliced videos. Left: varying number of frames for model building. Right: varying recompression.

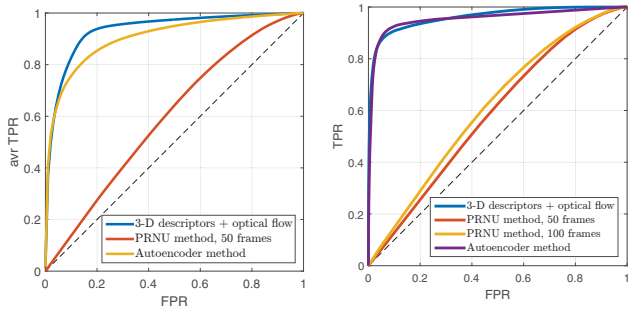

Fig. 6. Comparison of the proposed method with a PRNU-based and an autoencoder-based approach. Left: averaged results. Right: results on uncompressed `runner` video.

high under slight recompression. At MPEG-4 quality 7, localization accuracy deteriorates.

4.1.3. Comparison with other Forgery Detection Methods

We compared the proposed method to forgery detection that exploits photo-response non-nuniformity (PRNU) [26] and a procedure based on autoencoders with recurrent neuronal networks [22]. We also experimented with the methods presented in [18] and [19], but were not able to obtain meaningful results on our data. Results for the two tested reference algorithms averaged over all spliced videos are shown in Fig. 6 (left). We trained the PRNU-based method on 50 frames as done for the proposed method. PRNU performance is considerably lower compared to the proposed method. Some of the videos have relatively static background, which makes it more difficult for a PRNU-based estimator to extract a good fingerprint. To create a best-case scenario for PRNU, we also report results only on the video `runner` where we used several and different videos extracted from the same camera to obtain a better PRNU estimation. Results for this experiment are shown in Fig. 6 (right) by considering a variable number of frames. However, even when using 100 frames for PRNU estimation, performance is not competitive. Also, on average, we observe better results than the autoencoder-approach, which does not consider objects' motion. We presume that this weakens the performance of this method slightly.

4.2. Differentiation of Camera Processing Stacks

Table 1 shows results for differentiating video sequences from the smartphone dataset (Sec. 3.2). The smartphone along one row is used to build the model, the smartphone in the respective column is tested. We report the areas under the curve (AUCs) for the ROC curves over the Mahalanobis distances.

Table 1. AUCs for distinguishing video processing stacks. Rows: source for the model, Columns: source for testing the descriptors.

ID	1	2	3	4	5	6	7	8	9
1	-	0.48	0.88	0.72	0.94	0.95	0.96	0.92	0.56
2	0.99	-	1.00	0.92	1.00	1.00	1.00	0.99	0.93
3	1.00	1.00	-	0.96	0.99	1.00	0.99	1.00	0.91
4	1.00	0.99	0.99	-	0.98	0.97	0.99	1.00	0.96
5	1.00	1.00	1.00	0.99	-	1.00	1.00	1.00	1.00
6	1.00	0.98	1.00	0.95	0.99	-	1.00	1.00	1.00
7	0.99	0.99	0.99	0.96	0.96	0.99	-	0.95	0.95
8	0.99	0.95	1.00	0.98	1.00	0.99	0.96	-	0.96
9	0.99	0.99	1.00	0.98	1.00	1.00	1.00	1.00	-

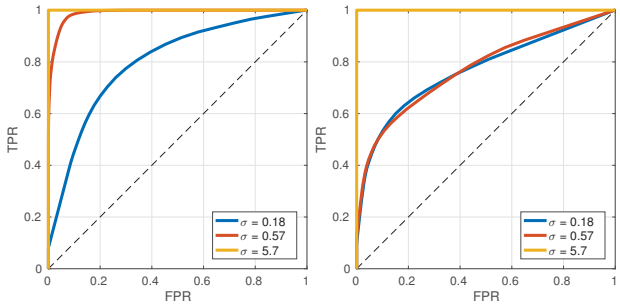

Fig. 7. Performance for distinguishing smartphones under added white Gaussian noise. Left: no secondary compression, right: recompressed with MPEG-4 quality 3.

The method provides a high detection rates, with a mean value of 0.96 ± 0.085. Three notable outliers with worse performance can be observed for the case when smartphone (1) is used for training. Upon visual inspection of the data, we noticed that the image content in (1) greatly varies between subsequences used to build the model and subsequences used to test against the model. However, the overall strong results indicate that the method is able to distinguish videos from different sources.

In a second experiment, to illustrate more decisive that differences in the noise residuals are traceable and in which order of magnitude they have to be, we add Gaussian noise to the tested descriptors of one of the smartphones. All other evaluation settings are identical to the previous experiments. Fig. 7 shows that Gaussian noise is reliably recognized down to a very low noise standard deviation of $\sigma = 0.18$. Under compression with MPEG-4 quality 3 (on the right), results are somewhat weaker, but still discriminative.

5. CONCLUSIONS

We presented a residual-based method for forensic comparison of two video sequences. We compute residual co-occurences in spatial and temporal domain, compensating video motion via optical flow, to build a statistical model. Descriptors from the second sequence are matched to that model. The method can be used either for accurate forgery localization, or for reliable discrimination of smartphones. It is less sensitive to scene content than PRNU, detects also small statistical disturbances such as mildly added Gaussian noise, and performs well under moderate recompression. In future work, we will expand this investigation towards the methods robustness under stronger recompression, noise pre-filtering or globally added noise.

6. 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, pp. 1–18, Dec. 2012.

[2] K. Kurosawa, K. Kuroki, and N. Saitoh, "CCD Fingerprint Method — Identification of a Video Camera from Videotaped Images," in *IEEE International Conference on Image Processing*, Oct. 1999, pp. 537–540.

[3] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Source digital camcorder identification using sensor photo response nonuniformity," in *Proc. of SPIE Security, Steganography, and Watermarking of Multimedia Contents IX*, Feb. 2007.

[4] W. van Houten and Z. Geradts, "Source video camera identification for multiply compressed videos originating from YouTube," *Digital Investigation*, vol. 6, no. 1-2, pp. 48–60, Sept. 2009.

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

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

[7] R. D. Singh and N. Aggarwal, "Optical flow and prediction residual based hybrid forensic system for inter-frame tampering detection," *Journal of Circuits, Systems and Computers*, vol. 26, no. 07, pp. 1–37, 2017.

[8] Y. Wu, X. Jiang, T. Sun, and W. Wang, "Exposing video inter-frame forgery based on velocity field consistency," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2674–2678.

[9] 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.

[10] L. D'Amiano, D. Cozzolino, G.Poggi, and L. Verdoliva, "Video forgery detection and localization based on 3D PatchMatch," in *IEEE International Conference on Multimedia and Expo Workshops*, 2015, pp. 1–6.

[11] W. Wang and H. Farid, "Exposing Digital Forgeries in Video by Detecting Double Quantization," in *ACM Workshop on Multimedia and Security*, 2009, pp. 39–48.

[12] W. Wang and H. Farid, "Exposing Digital Forgeries in Interlaced and Deinterlaced Video," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 438–449, Sept. 2007.

[13] D. Labartino, T. Bianchi, A. De Rosa, M. Fontani, D. Vázquez-Padín, A. Piva, and M. Barni, "Localization of forgeries in MPEG-2 video through GOP size and DQ analysis," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Sept 2013, pp. 494–499.

[14] 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, Mar. 2008.

[15] 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.

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

[17] J. Xu, Y. Yu, Y. Su, B. Dong, and X. You, "Detection of blue screen special effects in videos," *Physics Procedia*, vol. 33, pp. 1316–1322, 2012.

[18] Y. Su, Y. Han, and C. Zhang, "Detection of blue screen based on edge features," in *IEEE Joint International Information Technology and Artificial Intelligence Conference*, 2011, vol. 2, pp. 469–472.

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

[20] S. Chen, S. Tan, B. Li, and J. Huang, "Automatic Detection of Object-based Forgery in Advanced Video," *IEEE Transactions on Circuits and Systems for Video Technology*, in press 2015.

[21] J. Fridrich and J. Kodovský, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, June 2012.

[22] D. D'Avino, D. Cozzolino, G. Poggi, and L. Verdoliva, "Autoencoder with recurrent neural networks for video forgery detection," in *IS&T Electronic Imaging: Media Watermarking, Security, and Forensics*, Feb. 2017.

[23] L. Verdoliva, D. Cozzolino, and G. Poggi, "A feature-based approach for image tampering detection and localization," in *IEEE International Workshop on Information Forensics and Security*, 2014, pp. 149–154.

[24] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: a new blind image splicing detector," in *IEEE International Workshop on Information Forensics and Security*, 2015, pp. 1–6.

[25] C. Liu, *Beyond pixels: exploring new representations and applications for motion analysis*, Ph.D. thesis, Massachusetts Institute of Technology, 2009.

[26] J. Lukáš, J. Fridrich, and M. Goljan, "Digital Camera Identification From Sensor Pattern Noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.