An overview on video forensics

S. Milani[1], M. Fontani[2,4], P. Bestagini[1], M. Barni[2,4], A. Piva[3,4], M. Tagliasacchi[1], S. Tubaro[1]

1 Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milano, Italy

2 University of Siena, Dept. of Information Engineering, Siena, Italy

3 University of Florence, Dept. of Electronics and Telecommunications, Florence, Italy

4 National Inter-University Consortium for Telecommunications (CNIT), Florence, Italy

e-mail: milani@elet.polimi.it,marco.fontani@unisi.it,bestagini@elet.polimi.it,barni@dii.unisi.it
alessandro.piva@unifi.it,marco.tagliasacchi@polimi.it,stefano.tubaro@polimi.it

Abstract

The broad availability of tools for the acquisition and processing of multimedia signals has recently led to the concern that images and videos cannot be considered a trustworthy evidence, since they can be altered rather easily. This possibility raises the need to verify whether a multimedia content, which can be downloaded from the internet, acquired by a video surveillance system, or received by a digital TV broadcaster, is original or not. To cope with these issues, signal processing experts have been investigating effective video forensic strategies aimed at reconstructing the processing history of the video data under investigation and validating their origins. The key assumption of these techniques is that most alterations are not reversible and leave in the reconstructed signal some "footprints", which can be analyzed in order to identify the previous processing steps.

This paper presents an overview of the video forensic techniques that have been proposed in the literature, focusing on the acquisition, compression, and editing operations, trying to highlight strengths and weaknesses of each solution. It also provides a review of simple processing chains that combine different operations. Anti-forensic techniques are also considered to outline the current limitations and highlight the open research issues.

Index Terms

video forensics, image forensics, forgery detection, double compression, processing history estimation.

I. Introduction

In the recent years the availability of inexpensive, portable, and highly-usable digital multimedia devices (such as cameras, mobile-phones, digital recorders, etc.) has increased the possibility of generating digital audiovisual data without any time, location, and network-related constraints. In addition, the versatility of the digital support allows copying, editing, and distributing the multimedia data with little effort. As a consequence, the authentication and validation of a given content have become more and more difficult, due to the possible diverse origins and the potential alterations that could have been operated. This difficulty has severe implications when the digital content is used to support legal evidences. Digital videos and photographs can be no longer considered "proof of evidence/occurrence" since their origin and integrity cannot be trusted [1]. Moreover, the detection of copyright infringements and the validation of the legal property of multimedia data may be difficult since there is no way to identify the original owner.

From these premises, a significant research effort has been recently devoted to the forensic analysis of multimedia data. A large part of the research activities in this field are devoted to the analysis of still images, since digital photographs are largely used to provide objective evidence in legal, medical, and surveillance applications [2]. In particular several approaches target the possibility of validating, detecting alterations, and recovering the chain of processing steps operated on digital images. As a result, nowadays digital image forensic techniques enable to determine: whether an image is original or artificially created via cut and paste operations from different photos; which source generated an image (camera model, vendors); whether the whole image or parts of it have been artificially modified and how; what was the processing history of an image. These solutions rely on the consideration that many processing steps are not reversible and leave some traces in the resulting signal (hereby called "footprints"). Detecting and analyzing these footprints allow the reconstruction of the chain of processing steps. In other words, the detection of these footprints allows a sort of reverse engineering of digital content, in order to identify the type and order of the processing steps that a digital content has undergone, from its first generation to its actual form.

Despite the significant available literature on digital image forensics, video forensics still presents many unexplored research issues, because of the peculiarities of video signals with respect to images and the wider range of possible alterations that can be applied on this type of digital content. In fact, all the potential modifications concerning digital images can be operated both on the single frames of a video sequence and along the temporal dimension. This might be aimed at hiding or erasing details from the

recorded scene, concealing the originating source, redistributing the original signal without the owner's permission or pretending on its characteristics (e.g., low-quality contents re-encoded at high quality) [3], [4]. Moreover, forensic analysis of video content proves to be harder with respect to the analysis of still images since video data is practically always available in compressed formats and several times a high compression factor is used to store it. Strong compression ratios may cancel or fatally compromise the existing footprints so that the processing history is, entirely or in part, no longer recoverable.

On top of that, forensic analysts must now face the problem of anti-forensic techniques, which consist in modifying the forging process in order to make the unauthorized alterations transparent to forgery detection algorithms. Since each of these techniques is usually targeted to erase one specific trace left during the manipulation, anti-forensic methods are very heterogeneous. Nevertheless, all of them should satisfy two basic principles: do not hinder significantly the quality of the forged content that is produced; do not introduce artifacts that are easily detectable, so that anti-forensic techniques could be countered by the content owner. Although most of the anti-forensic strategies presented in literature have been developed for still images only, there are some techniques concerning video data.

The original contribution of this paper relies in providing an overview of the main forensic techniques that have been designed so far in the video content analysis. Previous overview papers in the literature mainly address image forensics and just a few details are provided about video content analysis. We believe that video forensic analysis has been maturely developed so that a review of the proposed techniques is widely justified.

In the following, we outline the structure of the paper. Section II provides the necessary background on digital image forensics, as it provides the foundations for analogous techniques targeting video content. The remaining sections deal with various aspects related to video forensics. We start addressing video acquisition in Section III, presenting several strategies to identify the device that captured a given video content. Then, in Section IV we consider the traces left by video coding, which are used to determine, e.g., the coding parameters, the coding standard or the number of multiple compression steps. Video doctoring is addressed in Section V, which presents forensic analysis methods based on detecting inconsistencies in acquisition and coding based footprints, as well as methods that reveal traces left by the forgery itself. Section VI concludes the survey, indicating open issues in the field of video forensics that might be tackled by future research efforts.

II. A Quick Overview of the State-of-the-Art in Image forensics

As mentioned in the previous section, image forensic tools have been widely studied in the past years due to the many applications of digital images that require some kind of validation. Many of them can be applied to video signals as well by considering each frame as single images, while others can be extended including the temporal dimension as well.

For this reason, a preliminary review of the state-of-the-art on image forensics is necessary in order to outline the baseline scenario from where video forensics departs. Many detailed overviews can be found in literature on digital image forensics (for example, see [5] and [6]). Here, we just outline some of the most important works that offered a sort of common background for the current and future video forensic techniques. In particular, we will discuss methods dealing with camera artifacts, compression footprints, and geometric inconsistencies.

The methods that follow enable to perform image authentication and, in some cases, tampering localization, without resorting to additional side information explicitly computed by the content owner. This is in contrast with other approaches based on, e.g., digital watermarking [7][8] or multimedia hashing [9][10][11][12][13], or a combination of both [14].

A. Camera Artifacts

Studies on camera artifacts that are left during the acquisition pipeline have laid the basis for image forensics. The far more studied artifact is the multiplicative noise introduced by CCD/CMOS sensors, named Photo Response Non Uniformity (PRNU) noise. PRNU has been exploited both for digital camera identification [15] and for image integrity verification [16], and it proves to be a reliable trace also when an image is compressed using the JPEG codec.

Since common digital cameras are equipped with just one sensor, color images are obtained by overlaying a Color Filter Array (CFA) to it, and using a demosaicing algorithm for interpolating missing values. The specific correlation pattern introduced during this phase allows to perform device model identification and tampering detection [17], provided that images are not (or very little) compressed.

The last artifact that we mention is chromatic aberration, that is due to the camera lens shape; inconsistencies in this effect can be searched on to identify tampered regions in the image, as explained in [18] and [19].

B. Image compression

A significant investigation activity has been carried on image coding forensics since the lossy nature of many compression strategies leaves peculiar traces on the resulting images. These footprints allow the forensic analyst to infer whether an image has been compressed, which encoder and which parameters have been used, and if the image has undergone multiple compression steps [20]. In order to understand if an image has been compressed, in [21] the authors show how to exploit a statistic model called Benford's law. Alternatively, in [22], the authors focus on identifying if an image has been block-wise processed, also estimating the horizontal and vertical block dimensions. If the image has been compressed, in [23] the authors propose a method capable of identifying the used encoder, which is useful, for example, to differentiate between DCT- and DWT-based coding architectures. A method to infer the quantization step used for a JPEG compressed image is shown in [24] and [25]. Finally, [26] [27] [28] and [29] [30] propose some methods to expose double JPEG compression based on the analysis of the histograms of DCT coefficients.

C. Geometric/Physics inconsistencies

Since human brain is notoriously not good in calculating projections and perspectives, most forged images contain inconsistencies at the "scene" level (e.g. in lighting, shadows, perspective, etc.). Although being very difficult to perform in a fully automatic fashion, this kind of analysis is a powerful instrument for image integrity verification. One of the main advantages of this approach is that, being fairly independent on low-level characteristics of images, it is well suited also for strongly compressed or low-quality images.

Johnson and Farid proposed a technique allowing to detect inconsistencies in scene illumination [31] and another one which reveals inconsistencies in spotlight reflection in human eyes [32]. Zhang et al. introduced methods for revealing anomalous behavior of shadows geometry and color [33]. Also, inconsistencies in the perspective of an image have been exploited, for example, in the work from Conotter et al. [34], which detects anomalies in the perspective of signs and billboards writings.

III. Forensic tools for video acquisition analysis

The analysis of image acquisition is one of the earliest problems that emerged in multimedia forensics, being very similar to the "classical" forensic technique of ballistic fingerprinting. Its basic goal is to understand the very first steps of the history of a content, namely identifying the originating device. The source identification problem has been approached from several standpoints. We may be interested

in understanding: i) which *kind* of device/technique generated the content (e.g., camera, scanner, photo realistic computer graphics, etc.), ii) which *model* of a device was used or, more specifically, iii) *which device* generated the content.

Different techniques address each of these problems in image forensics, and some of them have naturally laid the basis for the corresponding video forensic approaches. However, Section III-A will show that source identification has not yet reached a mature state in the case of videos.

Another interesting application that recently emerged in the field of video forensics is the detection of illegal reproductions, noticeably bootlegs videos and captured screeenshots. This problem will be separately discussed in Section III-B.

Before deepening the discussion, we introduce in Figure 1 a simplified model of the acquisition chain, when a standard camcorder is adopted. First, the sensed scene is distorted by optical lenses and then mosaiced by an RGB Color Filter Array (CFA). Pixel values are stored on the internal CCD/CMOS array, and then further processed by the in-camera software. The last step usually consists in lossy encoding the resulting frames, typically using MPEG-x or H.26x codecs for cameras and 3GP codecs for mobile phones (see Section IV). The captured images are then either displayed/projected on screen or printed, and can be potentially recaptured with another camera.

A. Identification of acquisition device

In the field of image forensics, many approaches have been developed to investigate each of the aforementioned questions about the acquisition process. Conversely, the works on video forensics assume that the content has been recorded using a camcorder, or a modern cell phone. To the best of our knowledge, no video-specific approaches have been developed to distinguish between computer graphics and real scenes. Instead, all the works in this field focus on identifying the specific device that originated a given content.

Kurosawa et al. [35] were the first to introduce the problem of camcorder fingerprinting. They proposed a method to identify individual video cameras or video camera models by analyzing videotaped images. They observed that dark-current noise of CCD chips, that is determined during the manufacturing process, creates a fixed pattern noise, which is practically unique for each device, and they also proposed a way to estimate this fixed pattern. Due to very strong hypotheses on the pattern extraction procedure (hundreds of frames recording a black screen were needed) this work did not allow to understand if a *given* video came from a specific camera. Nevertheless, it can be considered as one of the pioneering works in video forensics. Later, research in image forensics demonstrated that Photo Response Non Uniformity (PRNU)

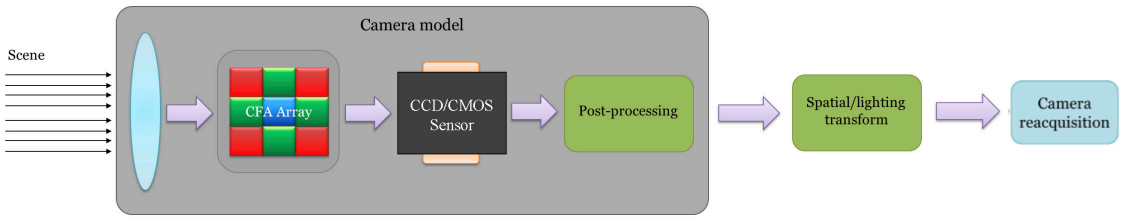

Fig. 1: Typical acquisition pipeline: light enters the camera through the lens, is filtered by the Color Filter Array and converted to a digital signal by the sensor. Usually, this is followed by some in-camera post processing and compression. In some cases, the video can be projected/displayed and re-acquired with another camera, usually undergoing lighting and spatial distortions.

noise could provide a much more strong and reliable fingerprint of a CCD array and, consequently, more recent works targeting source identification for video are based on this kind of feature.

1) PRNU based source identification: Many source identification techniques in image forensics exploit the PRNU noise introduced by the sensor. Although not being the only kind of sensor noise [36], PRNU has proven to be the most robust feature. Indeed, being a multiplicative noise, it is difficult for device manufacturers to remove it. First, we describe how this method works in the case of images. Then, we discuss its extension to videos, highlighting the challenging issues that arise.

Given a noise free image I_0, the image I acquired by the sensor is modeled as:

$$I = I_0 + \gamma I_0 K + N, \tag{1}$$

where γ is a multiplicative factor, K is the PRNU noise and N models all the other additive noise sources (see [36] for details). Note that all operations are intended element-wise.

If we could perfectly separate I from I_0, it would be easy to compute a good estimate of K from a single image. Unfortunately, this cannot be done in general: separating content from noise is a challenging task, as demonstrated by several works on image denoising. Consequently, the common approach is to estimate K from a *group* of authentic images I_j, $j = 1, \dots, N$. Each image I_j is first denoised using an appropriate filter. Then, the denoised version $\bar{I}_j$ is subtracted from I_j, yielding:

$$W_j = I_j - \bar{I}_j, \tag{2}$$

where W_j is the residual noise for the j-th image. The PRNU is then estimated as

$$K = \frac{\sum_{j=1}^{n} W_j I_j}{\sum_{j=1}^{n} I_j^2}. \tag{3}$$

From a technical point of view, two factors are of primary importance to obtain a good estimate of K:

1) using a group of flat, well illuminated images, e.g. pictures of a wall, of the sky, etc. Few tens of images usually suffice;
2) choosing an appropriate denoising filter (see [37]).

Once K is obtained for a device, checking if a query image S has been generated from that device reduces to evaluating the correlation between the noise component of the query image and the reference noise of the device. Formally, S is denoised with the same filter and subtracted from itself, yielding W_S. Then, the correlation between the query image and the PRNU mask is obtained as:

$$\rho = SK \otimes W_S, \tag{4}$$

where the operator $\otimes$ denotes normalized correlation. The value of ρ is usually low (e.g., $\rho \simeq 0.2$) even for images that were actually acquired with the device that originated the mask. However, ρ is sufficiently discriminative, since correlation values with extraneous images is smaller by two or three orders of magnitude. Furthermore, experiments demonstrated that this kind of analysis is robust to JPEG compression at large quality factors (e.g. $> 80\%$).

Having provided the background for PRNU-based source identification in the case of still images, we move the scope of the discussion to the case of videos. At a first glance, it may seem that estimating the PRNU of a camcorder from a video sequence should be easier, due to the usually large amount of frames available. However, this is not true for two main reasons. First, typical spatial resolution of videos is much lower than that of images. Second, frames usually undergo strong quantization and aggressive coding that introduce more artifacts than those affecting JPEG-compressed images.

The first work about camcorder identification was proposed by Chen et al. [38]. They rely on the method described above for extracting the PRNU mask. However, a significant effort is devoted to the proper choice the denoising filter, which led to the selection of a wavelet-based filter designed to remove Gaussian noise [39]. In addition, a pre-processing step is included to mitigate quantization artifacts introduced by lossy coding. More specifically, the authors observe that blocking artifacts and ringing artifacts at frame boundaries (introduced to adjust the size of the frame to a multiple of the block size) introduce a noise pattern that strongly depends on the compression algorithm rather than on the acquisition hardware. They propose a method to identify the frequencies of the DFT transform where such noise contribution is located and suppress them, thus increasing noticeably the performance of the estimation. The experiments in [38] showed that a tradeoff exists between video quality (in terms of bitrate) and length to achieve successful detection. If the video is compressed at high quality (e.g. 4-6 Mb/s), then

a relatively short sequence (40 sec.) suffices for a good estimation of the mask. Conversely, for low quality videos (e.g. 150 Kb/s) the length of the training sequence must be doubled to obtain comparable performance.

The challenging problem of video source identification from low quality videos has been deeply explored by van Houten et al. in several works [40] [41] [42]. The authors recorded videos using several different cameras, with various resolutions and bitrates. Then, they uploaded these videos on YouTube and downloaded them. Since YouTube re-encodes video during uploading, frames underwent at least *double* compression. After a large set of experiments, the authors came to the final conclusion that PRNU based source identification is still possible for very low quality videos, provided that the forensic analyst can extract the PRNU mask from a flat field video and that the aspect ratio of the video is not (automatically) changed during uploading.

In all the aforementioned works, video compression is considered to be a factor significantly hindering the identification of the PRNU-related footprints. However, digital video content mainly exists in compressed format, and the first compression step is operated by the camera itself using a proprietary codec. Therefore, the identification of the acquisition device could also be based on the identification the codec, leveraging the techniques described in Section IV

B. Detection of (illegal) reproduction of videos

An important problem in copyright protection is the proliferation of bootleg videos: many illegal copies of movies are made available on the Internet even before their official release. A great deal of these fake copies are produced by recording films with camcorders in cinemas (the last steps reported in Figure 1). Video forensics contributes to facing these problems by: i) detecting re-projected videos, as described in Section III-B1; ii) providing video retrieval technique based on device fingerprinting described in Section III-B2.

1) Detection of Re-acquisition : Re-acquisition occurs when a video sequence that is reproduced on a display or projected on a screen is recaptured. In the literature, some approaches were proposed based on active watermarking to perform both the identification of bootleg video [43] and to locate pirate's position in cinemas [44]. Recently, blind techniques are also emerging. Wang et al. [3] developed the most significant work in this field, exploiting the principles of multiple view geometry. They observed that re-acquisition captures a scene that is constrained to belong to a planar surface (e.g., the screen), whereas the original acquisition of the video was taken projecting objects from the real world to the camera plane. The authors show both mathematically and experimentally that re-projection usually causes non-

zero skew[1] in the intrinsic matrix of the global projection. Assuming that the skew of the camera used for the first acquisition was zero, significant deviations of this parameter in the estimated intrinsic matrix can be used as evidence that a video has been re-projected. Although very promising, this approach suffers from some limitations. Specifically, the original acquisition is modeled under several simplifying hypotheses, and skew estimation on real world video is difficult to perform without supervision. In [3], many experiments are conducted in a synthetic setting, yielding good performance (re-projected videos are detected with 88% accuracy and with 0.4% false alarm probability). However, only one experiment is based on real-world video content, presumably because of the complexity of skew estimation in this setting.

Lee et al. [45] addressed the problem of detecting if an image might be a screenshot re-captured from an interlaced video. In an interlaced video, half of the lines are recorded at time t in the *field* $f(x, y, t)$, and the other half are recorded at time $t + 1$ in the field $f(x, y, t + 1)$. There are several possible ways to obtain the full (spatial) resolution frame, i.e., $F(x, y, t)$, and one of the simplest is to weave fields together, as in Figure 2. Therefore, lines of the full resolution frame are acquired at *different*, though very near, time instants. If the video contains rapidly moving objects (or, equivalently, the camera is moving rapidly), this will introduce artifacts that are referred to as "combing". In [45], the authors exploit the directional property of combing artifacts to devise six discriminative features. These features are extracted from wavelet transform subbands (since combing artifacts are most evident near edges) and from vertical and horizontal differential histograms (which will expose strong differences in presence of such artifacts). Experimental results show an average accuracy higher than 97%.

2) Detection of Copying: The most common approach in video copy detection is to extract salient features from visual content that do not depend on the device used to capture the video. However, in [46], Bayram et al. pointed out that robust content-based signatures may hinder the capability of distinguishing between videos which are similar, although they are not copies of each other. This issue might arise, e.g., in the case of videos taken by two different users of the same scene. For this reason, they proposed to use source device characteristics extracted from videos to construct a copy detection technique. In [46], a video signature is obtained by estimating the PRNU fingerprints of camcorders involved in the generation of the video. The authors suggest to compute the PRNU fingerprint in the classical way. In the case of professional content, video is usually acquired using more than one device. As a consequence, this automatically yields a weighted mean of the different PRNU patterns, in which more frames taken with

[1]Camera skew accounts for the inclination of pixels: if pixels are assumed to be rectangular, camera skew is zero.

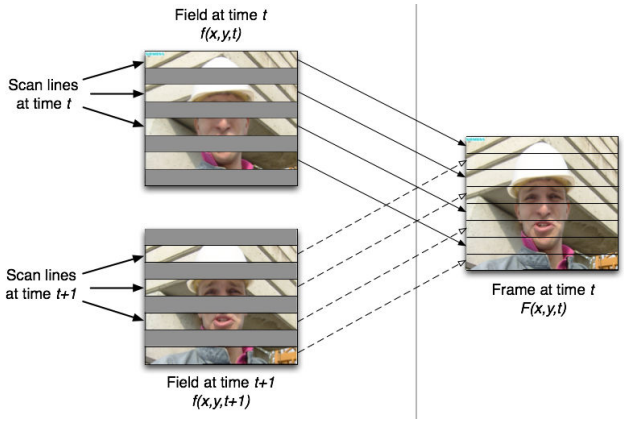

Fig. 2: A simple field weaving algorithm for video de-interlacing. This scheme uses T fields to produce a de-interlaced video of $T/2$ frames.

the same camera will result in a stronger weight assigned to it. Furthermore, it was observed that PRNU signatures are not totally insensible to the underlying frame content. Therefore, the weighted mean will also implicitly carry information about the content of the video. Notice that this method aims at obtaining a fingerprint for the content rather than for the device. Although it reuses PRNU fingerprinting techniques described in Section III-A1, it does so with a completely different objective. The authors also show that the fingerprint is robust against a set of common processing operations, i.e., contrast enhancement, blurring, frame dropping, subtitles, brightness adjustment, compression. Experiments performed on video downloaded from YouTube show a 96% detection rate for a 5% false alarm probability. However, slight rotation or resizing, not mentioned in [46], are likely to completely destroy the fingerprint.

IV. Forensic tools for video compression

Video content is typically available in a lossy compression format due to the large bit rate that is necessary to represent motion pictures either in an uncompressed or lossless format. Lossy compression leaves characteristic footprints, which might be detected by the forensic analyst. At the same time, the study of effective forensic tools dealing with compressed videos is a challenging task since coding operations have the potential effect of erasing the footprints left by previous manipulations. In this way, the processing history cannot be recovered anymore. Moreover, the wide set of video coding architectures that have been standardized during the last two decades introduces several degrees of freedom in the way different compression steps can be composed. As such, the codec adopted to compress a video sequence represents a distinctive connotative element. Therefore, if detected, it can be useful for the identification of the acquisition device, as well as for revealing possible manipulations.

Most of the existing video coding architectures build on top of coding tools originally designed for images. The JPEG standard is, by far, the most widely adopted coding technique for still images and many of its principles are reused for the compression of video signals [47]. A JPEG codec converts color images into a suitable color space (e.g. *YCbCr*), and processes each color component independently. The encoder operates according to three main steps:

- The image is divided into non-overlapping 8×8 pixel blocks $\mathbf{X} = [X(i,j)]$, $i,j = 0,\ldots,7$, which are transformed using a Discrete Cosine Transform (DCT) into coefficients $Y(i,j)$ (grouped into 8×8 blocks $\mathbf{Y}$).
- The DCT coefficients $Y(i,j)$ are uniformly quantized into levels $Y_q(i,j)$ with quantization steps $\Delta(i,j)$, which depend on the desired distortion and the spatial frequency (i,j), i.e.:

$$Y_q(i,j) = \text{sign}(Y(i,j))\text{round}\left(\frac{|Y(i,j)|}{\Delta(i,j)}\right). \tag{5}$$

  At the decoder, the reconstructed DCT coefficients $Y_r(i,j)$ are obtained by multiplying the quantization levels, i.e., $Y_r(i,j) = Y_q(i,j) \cdot \Delta(i,j)$.
- The quantization levels $Y_q(i,j)$ are lossless coded into a binary bitstream by means of Huffman coding tables.

Video coding architectures are more complex than those adopted for still images. Most of the widely-used coding standards (e.g. those of MPEG-x or H.26x families) inherit the use of block-wise transform coding from the JPEG standard. However, the architecture is complicated by several additional coding tools, e.g., spatial and temporal prediction, in-loop filtering, image interpolation, etc. Moreover, different transforms might be adopted within the same coding standard.

Fig. 3 illustrates a simplified block diagram representing the main steps in a conventional video coding architecture. First, the encoder splits the video sequence into frames, and each frame is divided into blocks of pixels $\mathbf{X}$. Each block is subtracted to a prediction generated by P exploiting either spatial and/or temporal correlation. Then, the prediction residual is encoded following a sequence of steps similar to those adopted by the JPEG standard. In this case, though, the values of the quantization steps and the characteristics of transform might change according to the specific standard.

Quantization is a non-invertible operation and it is the main source for information loss. Thus, it leaves characteristic footprints, which depend on the chosen quantization steps and quantization strategy. Therefore, the analysis of coding-based footprints might be leveraged to: i) infer details about the encoder (e.g. coding standard, coding parameters, non-normative tools); ii) assess the quality of a sequence in a no-reference framework; or iii) study the characteristics of the channel used to transmit the sequence.

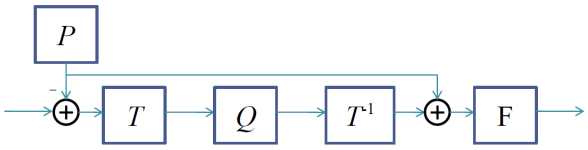

Fig. 3: Simplified block diagram of a conventional video codec. P computes the prediction, T the orthonormal transform, Q is the quantizer, and F is responsible of rounding and in-loop filtering.

In addition, block-wise processing introduces an artificial partition of the coded frame, which is further enhanced by the following processing steps. Unlike JPEG, the actual partitioning strategy is not fixed, as it depends on the specifications of coding standard and on the adopted rate-distortion optimization policy. Therefore, blockiness artifacts can be used to infer information about the adopted codec.

Finally, different codec implementations may adopt diverse spatial or temporal prediction strategies, according to rate-distortion requirements and computational constraints. The identification of the adopted motion vectors and coding modes provides relevant footprints that can be exploited by the forensic analyst, e.g. to validate the originating devices.

When each frame is considered as a single image, it is possible to apply image-based forensic analysis techniques. However, to enable a more thorough analysis, it is necessary to consider coding operations along the temporal dimension. In the following, we provide a survey of forensic tools aimed at reconstructing the coding history of video content. Whenever applicable, we start by briefly illustrating the techniques adopted for still images. Then, we show how they can be modified, extended and generalized to the case of video.

A. Video coding parameter identification

In image and video coding architectures, the choice of the coding parameters is driven by non-normative tools, which depend on the specific implementation of the codec and on the characteristics of the coded signal. In JPEG compression, the user-defined coding parameters are limited to the selection of the quantization matrices, which are adopted to improve the coding efficiency based on the psycho-visual analysis of human perception. Conversely, in the case of video compression, the number of coding parameters that can be adjusted is significantly wider. As a consequence, the forensic analyst needs to take into account a larger number of degrees of freedom when detecting the codec identity. This piece of information might enable the identification of vendor-dependent implementations of video codecs. As such, it could be potentially used to: i) verify intellectual property infringements; ii) identify the codec that

(a) original (b) compressed

Fig. 4: Original (a) and compressed (b) frames of a standard video sequence. The high compression rate is responsible for blocking artifacts.

generated the video content; iii) estimate the quality of the reconstructed video without the availability of the original source. In the literature, the methods aiming at estimating different coding parameters and syntax elements characterizing the adopted codec can be grouped into three main categories, which are further described below: i) approaches detecting block boundaries; ii) approaches estimating the quantization parameters, and; iii) approaches estimating the motion vectors.

1) Block detection: Most video coding architectures encode frames on a block-by-block basis. For this reason, artifacts at block boundaries can be exploited to reveal traces of previous compression steps. Typical blocking artifacts are shown in Fig 4. Identifying block boundaries allows also estimating the block size. It is possible to detect block-wise coding operations by checking local pixel consistency, as shown in [24], [25]. There, the authors evaluate whether the statistics of pixel differences across blocks differ from those of pixels within the same block. In this case, the image is supposed to be the result of block-wise compression.

In [48], the block size in a compressed video sequence is estimated by analyzing the reconstructed picture in the frequency domain and detecting those peaks that are related to discontinuities at block boundaries, rather than intrinsic features of the underlying image.

However, some modern video coding architectures (including, e.g., H.264/AVC as well as the recent HEVC standard under development) enable to use a deblocking filter to smooth artifacts at block boundaries, in addition to variable block sizes (also with non-square blocks). In these situations, traditional block detection methods fail, leaving this as an open issue for further investigations.

2) Quantization step detection: Scalar quantization in the transform domain leaves a very common footprint in the histogram of transform coefficients. Indeed, the histogram of each coefficient $Y_r(i,j)$

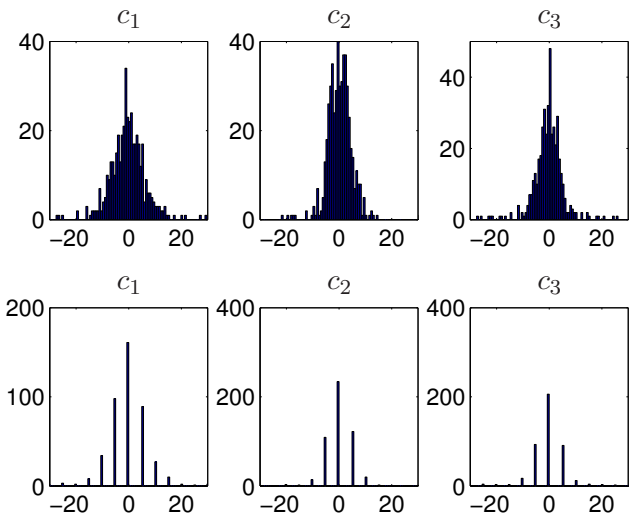

Fig. 5: Histograms of DCT coefficients (c_1, c_2, c_3) before (first row) and after (second row) quantization. The quantization step $\Delta(i,j)$ can be estimated by the gaps between consecutive peaks.

shows a typical comb-like distribution, in which the peaks are spaced apart by $\Delta(i,j)$, instead of a continuous distribution (Fig. 5). Ideally, the distribution can be expressed as follows:

$$p(Y_r;\Delta) = \sum_k w_k \delta(Y_r - k\Delta), \tag{6}$$

where δ is the Dirac delta function and w_k are weights that depend on the original distribution (note that indexes (i,j) are omitted for the sake of clarity). For this reason, the quantization step $\Delta(i,j)$ can be recovered by studying the distance between peaks of these histograms.

To this end, the work in [24] and [25] proposes to exploit this footprint to estimate the quality factor of JPEG compression. Specifically, the envelope of the comb-shaped histogram is approximated by means of a Gaussian distribution for DC coefficients, and a Laplacian distribution for AC coefficients. Then, the quality factor is estimated with a maximum likelihood (ML) approach, where the quantized coefficients are used as observations, and data coming from uniform and saturated blocks is discarded to make the estimation more robust.

In [49] the authors propose a method for estimating the elements of the whole quantization table. Separate histograms are computed for each DCT coefficient subband (i,j). Analyzing the periodicity of the power spectrum, it is possible to extract the quantization step $\Delta(i,j)$ for each subband. Periodicity is detected with a method based on the second order derivative applied to the histograms.

In [23], another method based on the histograms of DCT coefficients is proposed. There, the authors estimate the quantization table as a linear combination of existing quantization tables. A first estimate of the quantization step size for each DCT band is obtained from the distance between adjacent peaks of the histogram of transformed coefficients. However, in most cases, high-frequency coefficients do not contain enough information. For this reason some elements of the quantization matrix cannot be reconstructed, and they are estimated as a linear combination (preserving the already obtained quantization steps) of other existing quantization tables collected into a database.

A similar argument can be used to estimate the quantization parameter in video coding, when the same quantization matrix is used for all blocks in a frame. In [50] and [51], the authors consider the case of MPEG-2 and H.264/AVC coded video, respectively. There, the histograms are computed from DCT coefficients of prediction residuals. To this end, motion estimation is performed at the decoder side to recover an approximation of the motion-compensated prediction residuals available at the encoder.

Based on the proposed method for quantization step estimation a possible future line of investigation could be the inference of the rate-control algorithm applied at the encoder side, by tracking how quantization parameters vary over time. This could be an important hint to identify vendor-specific codec implementations.

3) Identification of Motion vectors: A significant difference between image and video coding is the use of predictors exploiting temporal correlation between consecutive frames. The idea is that of reducing temporal redundancy by exploiting similarities among neighboring video frames. This is achieved constructing a predictor of the current video frame by means of motion estimation and compensation. In most video coding architectures, a block-based motion model is adopted. Therefore, for each block, a motion vector (MV) is estimated, in such a way to generate a motion-compensated predictor. In [52], it is shown how to estimate, at the decoder, the motion vectors originally adopted by the encoder, also when the bitstream is missing. The key tenet is to perform motion estimation by maximizing, for each block, an objective function that measures the comb-like shape of the resulting prediction residuals in the DCT domain.

Although the estimation of coding parameters has been investigated, mainly focusing on block detection and quantization parameter estimation, there are still many unexplored areas due to the wide variety of coding options that can be enabled and the presence of a significant number of non-normative aspects in the standard definition (i.e., rate distortion optimization, motion estimation algorithm, etc.). These coding tools offer a significant amount of degrees of freedom to the video codec designer, who can implement in different ways an encoder producing a bitstream compliant with the target coding standard. On the

other hand, the task of forensic analyst becomes more and more difficult, when it comes to characterize and detect the different footprints left by each operation.

B. Video re-encoding

Every time a video sequence that has already been compressed is edited (e.g., scaling, cropping, brightness/contrast enhancement, local manipulation, etc.), it has to be re-compressed. Studying processing chains consisting of multiple compression steps is useful, e.g., for tampering detection or to identify the original encoder being used. This is a typical situation that arises, e.g., when video content is downloaded from video-sharing websites.

Of course, it is straightforward to obtain the parameters used in the last compression stage, as they can be read directly from the bitstream. However, it is much more challenging to extract information about the previous coding steps. For this reason, some authors have studied the footprints left by double video compression. The solutions proposed so far in the literature are mainly focused on MPEG video, and they exploit the same ideas originally used for JPEG double-compression.

1) Double compression: Double JPEG compression can be approximated by double quantization of transform coefficients $Y(i,j)$, such that

$$Y_{Q_1,Q_2} = \Delta_2 \cdot \mathrm{sign}(Y) \cdot \mathrm{round}\left(\frac{\Delta_1}{\Delta_2}\mathrm{round}\left(\frac{|Y|}{\Delta_1}\right)\right), \tag{7}$$

where indexes (i,j) have been omitted for the sake of clarity. Re-quantizing already quantized coefficients with different quantization step sizes affects the histogram of DCT coefficients. For this reason, most solutions are based on the statistical footprints extracted from such histograms.

In [26], Lukáš and Fridrich show how double compression introduces characteristic peaks in the histogram, which alter the original statistics and assume different configurations according to the relationship between the quantization step sizes of consecutive compression operations, i.e., respectively, Δ_1 and Δ_2. More precisely, the authors highlight how peaks can be more or less evident depending on the relationship between the two step sizes, and propose a strategy to identify double compression. Special attention is paid to the presence of double peaks and missing centroids (i.e., those peaks with very small probability) in the DCT coefficient histograms, as they are identified to be robust features providing information about the primary quantization. Their approach relies on cropping the reconstructed image (in order to disrupt the structure of JPEG blocks) and compressing it with a set of candidate quantization tables. The image is then compressed using $\Delta_2(i,j)$ and the histogram of DCT coefficients is computed. The proposed method chooses the quantization table such that the resulting histogram is as close as possible

to that obtained from the reconstructed image. This method is further explored in [53], providing a way to automatically detect and locate regions that have gone through a second JPEG compression stage. A similar solution is proposed in [54], which considers only the histograms related to the nine most significant DCT subbands, which are not quantized to zero. The corresponding quantization steps, i.e. those employed in the first compression stage, are computed via a Support Vector Machine (SVM) classifier. The remaining quantization steps are computed via a Maximum Likelihood estimator.

A widely-adopted strategy for the detection of double compression relies on the so-called Benford's law or first digit law [21]. In a nutshell, it relies on the analysis of the distribution of the most significant decimal digit m (also called "first digit") of the absolute value of quantized transformed coefficients. Indeed, in the case of an original uncompressed image, the distribution is closely related to the Benford's equation or its generalized version, i.e.,

$$p(m) = N \log_{10}\left(1 + \frac{1}{m}\right) \text{ or } p(m) = N \log_{10}\left(1 + \frac{1}{\alpha + m^{\beta}}\right), \tag{8}$$

respectively (where N is a normalizing constant). Whenever the empirical distribution deviates significantly from the interpolated logarithmic curve, it is possible to infer that the image was compressed twice. Then, it is also possible to estimate the compression parameters of the first coding stage. Many double compression detection approaches based on Benford's law have been designed focusing on still images [21], giving detection accuracy higher than 90%. These solutions have also been extended to the case of video signals, but the prediction units (spatial or temporal) that are part of the compression scheme reduce the efficiency of the detector, leading to an accuracy higher than 70%. More recently, this approach has also been extended to the case of multiple JPEG compression steps since in many practical cases images and videos are compressed more than twice [20].

In [4], the authors address the problem of estimating the traces of double compression of an MPEG coded video. Two scenarios are considered, depending on whether the Group of Pictures (GOP) structure used in the first compression is preserved or not. In the former situation, every frame is re-encoded in a frame of the same kind, so that I,B, or P frames remain, respectively, I,B, or P. Since encoding I-frames is not dissimilar from JPEG compression, when an I-frame is re-encoded at a different bitrate, DCT coefficients are subject to two levels of quantization. Therefore, the histograms of DCT coefficients assume a characteristic shape that deviates from the original distribution. In particular, when the quantization step size decreases from the first to the second compression, some bins in the histogram are left empty. Conversely, when the step size increases, the histogram is affected in a characteristic way. Instead, the latter situation typically arises in the case of frame removal or insertion attacks. Since the GOP structure

is changed, I-frames can be re-encoded into another kind of frame. However, this gives rise to larger prediction residuals after motion-compensation. The authors show that by looking at the Fourier transform of the energy of the displaced frame difference over time, the presence of spikes reveals a change in the GOP structure, which is a clue of double-compression.

In [55], the authors propose another method for detecting MPEG double compression based on blocking artifacts. A metric for computing the Block Artifact Strength (BAS) for each frame is defined. This score is inspired to the method in [25] and relies on the difference of pixel values across a grid. The mean BAS is computed for sequences obtained removing from one to eleven frames, obtaining a feature vector of BAS values. If the sequence has been previously tampered with by frame removal and re-compression, the feature vector presents a characteristic behavior.

In [56], MPEG double quantization detection is addressed on a macroblock-by-macroblock basis. In particular, a probability distribution model for DCT coefficients of a macroblock in an I-frame is discussed. With an Estimation-Maximization (EM) technique, the probability distribution that would arise if a macroblock were double-quantized is estimated. Then, such distribution is compared with the actual distribution of the coefficients. From this comparison, the authors extract the probability that a block has been double-compressed. These solutions can be extended to enable the detection of double video compression even in a realistic scenario in which different codecs are employed in each compression stage.

The approach in [57] presents an effective codec identification strategy that allows to determine the codec used in the first compression stage in the case of double video compression (note that the codec used in the second compression stage is known since the bitstream is usually available). The proposed algorithm relies on the assumption that quantization is an idempotent operator, i.e., whenever a quantizer is applied to a value that has already been previously quantized and reconstructed by the same quantizer, the output value is highly correlated with the input value. As a matter of fact, it is possible to identify the adopted codec and its configuration by re-encoding the analyzed sequence a third time, with different codecs and parameter settings. Whenever the output sequence presents the highest correlation with the input video, it is possible to infer that the adopted coding set-up corresponds to that of the first compression.

Although the detection of double compression for images is a widely-investigated issue, double video compression still proves to be an open research problem, because of the complexity and diversity of video coding architectures. Whenever two different codecs are involved with similar parameters, the detection of double video compression becomes significantly more difficult [57]. Moreover, multiple compression is a current and poorly-explored topic despite the fact that multimedia content available on the internet

has been often coded more than twice [20].

C. Network footprints identification

Video transmission over a noisy channel leaves characteristic footprints in the reconstructed video content. Indeed, packet losses and errors might affect the received bitstream. As a consequence, some of the coded data will be missing or corrupted. Error concealment is designed to take care of this, trying to recover the correct information and mitigate the channel-induced distortion. However, this operation introduces some artifacts in the reconstructed video, which can be detected to infer the underlying loss (or error) pattern. The specific loss pattern permits the identification of the characteristics of the channel that was employed during the transmission of the coded video. More precisely, it is possible to analyze the loss (error) probability, the burstiness, and other statistics related to the distribution of errors in order to identify, e.g., the transmission protocol or the streaming infrastructure.

Most of the approaches targeting the identification of network footprints are intended for no-reference quality monitoring, i.e., the estimation of the quality of the video sequence without having access to the original source as a reference signal. These solutions are designed to provide network devices and client terminals with effective tools that measure the Quality-of-Experience (QoE) offered to the end user. The proposed approaches can be divided into two main groups.

The first class of network footprint identification algorithms takes into consideration transmission statistics to estimate the channel distortion on the reconstructed sequence. In [58], the authors present an algorithm based on several quality assessment metrics to estimate the packet loss impairment in the reconstructed video. However, the proposed solution adopts full-reference quality metrics that require the availability of the original uncompressed video stream. A different approach is presented in [59], where the channel distortion affecting the received video sequence is computed according to three different strategies. A first solution computes the final video quality from the network statistics; a second solution employs the packet loss statistics and evaluates the spatial and temporal impact of losses on the final sequence; the third one evaluates the effects of error propagation on the sequence. These solutions target control systems employed by network service providers, which need to monitor the quality of the final video sequences without having access to the original signal. Another no-reference PSNR estimation strategy is proposed in [60]. The proposed solution evaluates the effects of temporal and spatial error concealment without having access to the original video sequence, and the output values present a good correlation with MOS scores. As a matter of fact, it is possible to consider this approach as a hybrid solution, in that it exploits both the received bitstream and the reconstructed pixel values.

A second class of strategies assumes that the transmitted video sequence has been decoded and that only the reconstructed pixels are available. This situation is representative of all those cases in which the video analyst does not have access to the bitstream. The solution proposed in [61] builds on top of the metrics proposed in [60], but no-reference quality estimation is carried out without considering the availability of the bitstream. Therefore, the proposed solution processes only pixel values, identifying which video slices were lost, and producing as output a quality value that presents good correlation with the MSE value obtained in full reference fashion. The method assumes that slices correspond to rows of macroblocks. However, modern video coding standard enable more flexible slicing schemes. Hence, the method has been recently extended in [62], in which a maximum a-posteriori approach is devised to take into account a spatial prior on the distribution of lost slices.

D. Video compression anti-forensics

The design of novel forensic strategies aimed at characterizing image and video compression is paralleled by the investigation of corresponding anti-forensic methods. That is, a malicious adversary might tamper with video content in such a way to disguise its traces.

An anti-forensic approach for JPEG compression has been recently proposed in [63]. There, the traces of compression are hidden by adding a dithering noise signal. Dithering is devised to reshape the histogram of DCT coefficients in such a way that the original Laplacian distribution is restored. In a following work by the same authors [64], a similar strategy is proposed to erase the traces of tampering from an image and hide double JPEG compression. This is achieved by a combined strategy, i.e., removing blocking artifacts by means of median filtering and restoring the original distribution of DCT coefficients with the same method as in [63]. In this way, the forensic analyst is not able to identify the tampered region by inspecting the distribution of DCT coefficients. However, it has been recently shown that anti-forensic methods are prone to leave their own footprints. In [65], [66], the authors study the distortion which is inevitably introduced by the anti-forensic method in [63] and propose an effective algorithm to counter it.

The aforementioned anti-forensic methods might be potentially applied to videos on a frame-by-frame basis. To the authors' knowledge, the only work that addresses an anti-forensic method specifically tailored to video compression is [67]. There, the authors propose a method to fool the state-of-the-art frame deletion and detection technique in [4], which is based on the analysis of the motion-compensated prediction error sequence. However, this is achieved by paying a cost in terms of coding efficiency, since some of the frames of the video sequence need to be re-encoded at a bitrate higher than the one originally

used. However, this research field is quite recent and just a few works can be found on the subject.

V. Forensic tools for video doctoring detection

Although being more complicated than for images, creating a forged video is now easier than before, due to the availability of video editing suites. At the same time, videos are extensively used for surveillance, and they are usually considered a much stronger proof than a single shot. There are many different ways of tampering with a video, and some of them are not complicated at all: one may be interested in replacing or removing some frames (e.g., from a video-surveillance recording), replicating a set of frames, introducing, duplicating or removing some objects from the scene.

It is possible to classify both video forgery and video forensic techniques as intra-frame (attack/analysis is performed frame-wise, considering one frame at a time), or inter-frame (relationships between adjacent frames are considered). Although it would be possible to analyze the integrity of a video by simply applying image forensic tools to each separate frame, this approach is considered unpractical, mainly for these reasons:

- complexity: tools for detecting forgeries in images are usually computationally demanding;
- reliability: replication/deletion of frames would not be detected by any image forensic tools;
- convenience: creating doctored videos that are temporally consistent is very difficult, so these kinds of inter-frame relationships are a valuable asset for forgery identification.

In the following subsections we survey existing techniques for video doctoring detection. We group them according to the type of analysis they rely on. Section V-A covers camera-based techniques. Section V-B covers coding-based techniques and Section V-C discusses some pioneering works that exploit geometrical/physical inconsistencies to detect tampering. In Section V-D, we analyze the problem of identifying frames, or portion of frames, copy-move forgeries. In Section V-E, we discuss anti-forensic strategies. Finally, in Section V-F we present a solution to the problem of understanding the relationships between objects in large multimedia collections (phylogeny).

A. Camera based editing detection

As discussed in Section III, camcorders usually leave a characteristic fingerprint in recorded videos. Although these kinds of artifacts are usually exploited just for device identification, some works leverage on them also for tampering detection. The main contributions in this field are from Mondaini et al. [68], Hsu et al. [69] and Kobayashi et al. [70].

Mondaini et al. [68] proposed a direct application of the PRNU fingerprinting technique (see Section III-A1) to video sequences: the characteristic pattern of the camcorder is estimated on the first frames of the video, and is used to detect several kinds of attacks. Specifically, authors evaluate three correlations coefficient (see Equation 4): i) the one between each frame noise and the reference noise, ii) the one between the noise of two consecutive frames, iii) the one between frames (without noise extraction). Each of these correlation coefficients is thresholded to obtain a binary event, and different combinations of events allow to detect different kind of doctoring, among which: frame insertion, object insertion within a frame (cut-and-paste attack), frame replication. Experiments are carried both on uncompressed and on MPEG compressed videos: results show that the method is reliable (only some case-studies are reported, not averaged values) on uncompressed videos, while MPEG encoding afflicts performances significantly.

Hsu et al. [69] adopt a technique based on temporal correlation of noise residues, where the "noise residue" of a frame is defined as what remains after subtracting from the frame its denoised version (the filtering technique proposed in [39] is used). Each frame is divided into blocks, and the correlation between the noise residue of temporally neighboring blocks (i.e. blocks in the same position belonging to two adjacent frames) are evaluated. When a region is forged, the correlation value between temporal noise residues will be radically changed: it will be decreased if pixels of the blocks are pasted from another frame/region (or automatically generated through inpainting), while it will be raised to 1 if a frame replication occurs. Authors propose a two-step detection approach to lower the complexity of the scheme: first a rough threshold decision is applied to correlations and, if the frame contains a significant number of suspect blocks, a more deep statistical analysis is performed, modeling the behavior of noise residue correlation through a Gaussian mixture and estimating its parameters. Performances are far from ideal: when working on copy-paste attacked videos, on average only 55% of forged blocks are detected (false positive rate being 3.3%); when working on synthetically inpainted frames, detection raises to 74% but also false positive rate increases to 7% on average. Furthermore, when the video is lossy encoded, performances drop rapidly with the quantization strength. Nevertheless, despite authors do not provide experiments in this direction, this method should be effective for detecting frame replication, which is an important attack in the video-surveillance scenario. It is worth noting that, although exploiting camera characteristics, this work does not target the fingerprinting of the device at all.

Another camera-based approach is the one from Kobayashi et al. [70]: they propose to detect suspicious regions in video recorded from a *static scene* by using noise characteristics of the acquisition device.

Specifically, photon shot noise[2] is exploited, which mainly depends on irradiance through a function named Noise Level Function (NLF). The method computes the probability of forgery for each pixel by checking the consistency of the NLFs in forged regions and unforged regions. Since it is not known a-priori which pixels belong to which region, the Expectation-Maximization [71] algorithm is employed to simultaneously estimate the NLF for each video source and the probability of forgery for each pixel. The core of the technique resides in correctly estimating the function from temporal fluctuations of pixel values, and this estimate is thoroughly discussed from a theoretical point of view. On the other hand, from a practical point of view, the estimate can be performed only for pixels whose temporal variation results entirely from noise and not from motion of objects or camera. This limits the applicability of the approach to stationary videos, like those acquired by steady surveillance cameras. When this assumption is respected, and the video is not compressed, this method yields very good performances (97% of forged pixels are located with 2.5% of false alarm); also, the perfect resolution of the produced forgery map (each pixel is assigned a probability) should be appreciated. Unfortunately, since videos usually undergo some kind of noise reduction during encoding, performances drop dramatically when the video is compressed using conventional codecs like MPEG-2 or H.264, and this further limits the practical applicability of this work.

Going back to a global view, it can be stated that camera based methods are effective on uncompressed videos. However, videos are typically stored in compressed format in most practical applications. This motivates the investigation of camera footprints that are more robust to aggressive coding.

B. Detection based on coding artifacts

From what emerged in the previous section, video encoding strongly hinders the performances of camera based detection techniques. On the other hand, however, coding itself introduces artifacts that can be leveraged to investigate the integrity of the content. Since video codecs are designed to achieve strong compression ratios, they usually introduce rather strong artifacts in the content (as seen in Section IV). In the last years, some forensic researchers investigated the presence or the inconsistencies of these artifacts to asses the integrity of a video, and to localize which regions are not original.

The first approach in this direction was from Wang and Farid [4], focusing on MPEG compressed videos, where two phenomena are explored, one static (inter-frame) and one temporal (intra-frame). The static phenomena, which has been discussed in Section IV-B, relies on the fact that a forged MPEG

[2]This noise originates from the temporal fluctuations of the number of photons that fall onto a CCD element.

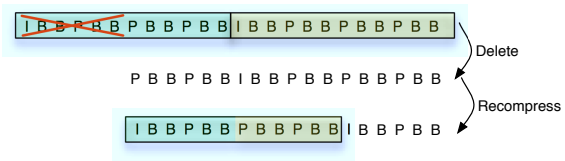

Fig. 6: In this example, the first six frames of the original MPEG compressed video (first row) are deleted, thus obtaining a new sequence (second row). When this sequence is re-compressed using MPEG, each GOP will contain frames that belonged to different GOPs in the original video (frames highlighted in yellow in the third row).

video will almost surely undergo two compressions the first being performed when the video is created, and the second when video is re-saved after being doctored. The temporal phenomena is based on the GOP (Group of Pictures) structure of MPEG files. As shown in Figure 6, when a video is re-compressed after removing or adding a group of frames, a desynchronization will occur in the GOP pattern. Due to the predictive nature of MPEG compression, all the P frames in a GOP are correlated to the initial I frame. In the re-compressed sequence, some of the frames are likely to move from one GOP to another (last row of Figure 6), so their correlation with the I frame of the new GOP will be smaller, resulting in larger prediction errors. If a single set of frames is deleted, the shift of P frames will be the same throughout all the video sequence, and the variability of prediction error in P frames along time will exhibit a periodic behavior. That is, smaller error values will result for frames that remained in the same GOP as the original video, and larger error for those that changed GOP.

This periodicity can be revealed via a Fourier analysis of the frame-wise average values of motion error. Authors show the effectiveness of this approach on several examples, although they do not allow us to give a value for precision-recall or overall accuracy of the method.

Another work from the same authors [56] provides a more accurate description of double compression in MPEG videos, which allows them to detect doubly compressed macro-blocks (16x16 pixels) instead of frames. Consequently, this approach allows to detect if only *part* of the frame has been compressed twice, which usually happen when the common digital effect of green-screening is applied (that is, a subject is recorded over a uniform background then it is cut and pasted into the target video). Performances of this technique depend on the ratio between the two compression quality factors: for ratios over 1.7 the method is almost ideal (99.4% detection rate) while for ratios less then 1.3 detection drops to 2.5%.

Quantization artifacts are not the only effect that have been exploited for video doctoring detection: Wang and Farid proposed another approach [72] for detecting tampering in interlaced and de-interlaced video (see Section III-B1 for a brief explanation of what an interlaced video is). For de-interlaced video,

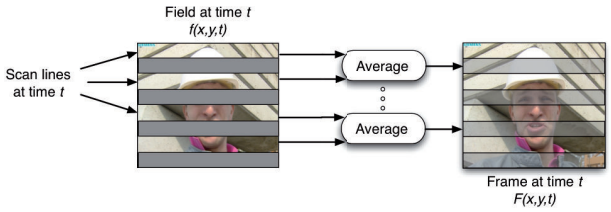

Fig. 7: Video interpolation based on line averaging, which is a field extension scheme. Compared to the method in Figure 2, this one has the advantage of producing a final video with T frames instead of $T/2$, without showing the combing artifact. On the other hand, vertical resolution is halved.

the authors consider how the missing rows of the frame are generated (see Figure 7 for an example): if they are not tampered with, they should be a combination of fields that are adjacent in time and/or space. Instead, if a region is forged, this relationship should not hold, thus exposing the doctoring. However, in practice, both the adopted interpolation method and the possibly doctored region are not known in advance. The authors propose to exploit the Expectation Maximization algorithm [71] to simultaneously estimate the parameters of the filter and assign pixels to original or tampered regions. To give a simple example, let us consider the odd rows $F_o(x, y, t)$ of an interlaced frame $F(x, y, t)$. Pixels that are not tampered with are said to belong to the model M_1, and should satisfy the following constraint:

$$F_o(x,y,t) = \sum_{i\in\{-3,-1,1,3\}} \alpha_i F(x, y+i, t) + \sum_{i\in\{-2,0,2\}} \beta_i F(x, y+i, t+1) + n(x,y),$$

where α_i and β_i are the coefficients of the interpolation filter and $n(x, y)$ is i.i.d. Gaussian noise. On the other hand, pixels in tampered regions belong to another model, M_2, for which a uniform distribution is assumed. With these settings, the EM algorithm iteratively determines the probability of each pixel of $F_o(x, y, t)$ to belong to M_1 (Expectation step). Then, it uses these assignments to refine the model M_1, by minimizing a cost function expressed in terms of α_i and β_i (Maximization step). Notice that the final result is a pixel-resolution probability map of tampering, and this is an important contribution in that tampering localization is always more difficult than tampering detection.

For interlaced video, in which frames are created by simply weaving together the odd and even fields, the presence of rapidly moving objects introduces the combing artifact, already mentioned in Section III-B1. Since the magnitude of this effect depends on the amount of motion between fields, authors use incoherence between inter-field and inter-frame motion to reveal tampering. Both techniques in [56] allow

the localization of tampering in time (frame) as well in space (region of the frame). Furthermore, both algorithms can be adapted to detect frame rate conversion. Since compression partially removes inter-pixel correlations this approach is mostly suited for medium/high quality video. For interlaced video, instead, compression does not seem to hinder performance.

We argue that much has still to be discovered in coding-based doctoring detection for videos. As a matter of fact, video coding algorithms are much more complex than JPEG compression. This makes detection of introduced artifacts more difficult, since mathematical models are not easy to derive. However, this should also motivate researchers to look for traces left by such video coding schemes, which are likely to be much stronger compared to the case of images, due to the aggressive compression that it is typically performed.

C. Detection based on inconsistencies in content

As already stated in Section II, it is very difficult to understand whether the geometry or the physical/lighting properties of a scene are consistent. In particular, it is very hard to do so unless some assistance from the analyst is provided. If this effort from the analyst may be affordable when a single image is to be checked, it would be prohibitive to check geometric consistencies in video on a frame-by-frame basis. Existing works usually exploit phenomena connected to motion in order to detect editing. So far, two approaches have been proposed: i) the one in [73], based on artifacts introduced by video inpainting, ii) the one in [74], that reveals inconsistencies in the motion of objects in free-flight.

Going into details, Zhang et al. [73] propose a method to detect video inpainting, which is a technique that automatically replaces some missing content in a frame by reproducing surrounding textures. Though originally developed for still images, this technique is also applicable frame-by-frame to video signals introducing annoying artifacts, known as “ghost shadows”, due to temporal discontinuity of the inpainted area. Authors observe that these artifacts are well exposed in the Accumulative Difference Image (ADI). This is obtained by comparing a reference image with every subsequent frame and using each pixel as a counter, which is incremented if the current frame differs significantly from the reference image. Unfortunately, ADI would also detect any moving object. Therefore, the authors propose a method to automatically detect the presence of these artifacts, provided that the removed object was a moving object. The authors point out that only detection of forgery is possible, and no localization is provided. Experiments, performed on just a few real world video sequences, show that the method is robust against strong MPEG compression.

Before moving to the work in [74], a remark must be made: if detecting geometrical inconsistencies

in an inter-frame fashion is difficult, it is perhaps more difficult to detect physical inconsistencies, since this requires to mix together tracking techniques and complex physical models to detect unexpected phenomena. Nevertheless, restricting the analysis to some specific scenarios, it is possible to develop ad-hoc techniques capable of such a task. This is what has been done by Conotter et al. in [74]: an algorithm is proposed to detect physically implausible trajectories of objects in video sequences. The key idea is to explicitly model the three-dimensional parabolic trajectory of objects in free-flight (e.g. a ball flying towards the basket) and the corresponding two-dimensional projection into the image plane. The flying object is extracted from video, compensating camera motion if needed, then the motion in the 3D space is estimated from 2D frames and compared to a plausible trajectory. If the deviation between observed and expected trajectories is large, the object is classified as tampered. Although analyzing a very specific scenario, the method inherits all the advantages that characterize forensic techniques based on physical and geometrical aspects; for example, performance does not depend on compression and video quality.

D. Copy-move detection in videos

Copy and copy-move attacks on images have been considered in order to prevent the illegal duplication or reusing of images. More precisely, these approaches check for similarities between pairs of images that are not supposed to be related (since they have been taken in different time/places or different origins are claimed). However, it is possible to verify that different images are copies of the same visual content checking the similarity between their features [75]. Many approaches for copy detection in images are based on SIFT, which allows detecting the presence of the same objects in the acquired scene [76].

Copy-move attacks are defined for video both as intra and inter-frame techniques. An intra-frame copy-move attack is conceptually identical to the one for still images, and consists in replicating a portion of the frame in the frame itself (the goal is usually to hide or replicate some object). An inter-frame copy-move, instead, consists in replacing some frames with a copy of previous ones, usually to hide something that entered the scene in the original video. To this end, partial inter-frame attacks can be defined, in which only a portion of a group of frames is substituted with the same part coming from a selected frame. To the best of our knowledge, there is only one work authored by Wang and Farid [77] that targets copy-move detection directly in video. The method uses a kind of divide-and-conquer approach: the whole video is split in subparts, and different kinds of correlation coefficients are computed in order to highlight similarities between different parts of the sequence. In the same work, a method for detecting region duplication, both for the inter-frame and intra-frame case, is defined. Results are good

(accuracy above 90%) for a stationary camera, and still interesting for a moving camera setting (approx. accuracy 80%). MPEG compression does not hinder performance.

E. Anti-forensic strategies

For what concerns video, only a work has been proposed by Stamm et al. [78] to fool one of the forensic techniques described in [4] (see Section V-B), specifically the one based on GOP desynchronization. Authors of [78] observe that the simplest way to make the forgery undetectable is to raise prediction errors of all frames to the values assumed in the spikes, so that peaks in the error due to desynchronization will be no longer distinguishable. In order to raise prediction errors, they alter the encoder so that a certain number of motion vectors will be set to zero even if they were not null. The quality of the video will not be reduced, since the error is stored during encoding and compensated before reproduction; furthermore, authors select which vector will be set to zero starting from those that are already small, so that the introduced error is spread on many vectors, and introduced modification is harder to detect. Authors also point out that the other detection technique proposed by Wang et al. in the same work [4] can be attacked using counter forensic methods designed for still images, in particular those that hide JPEG quantization effects [79].

For what concerns camera-artifacts based methods, there is a straightforward counter forensic method, which also applies to images: it simply consists in scaling the doctored video (even by a very low factor) and then re-encode it. Since rescaling requires an interpolation step, noise artifacts will be practically erased; furthermore, the correlation operator used in Equation 4 is performed element-wise, so frames having different sizes cannot be even compared directly.

F. Video Phylogeny

Two videos are termed "near-duplicate" if they share the same content but they show differences in resolution, size, colors and so on. If we have a set of near duplicate videos, like the one in Figure 8, it would be interesting to understand if one of them has been used to generate the others, and draw a graph of causal relationships between all these contents. This problem, which was firstly posed for images under the name "image phylogeny" [80] or "image dependencies" [81], is being studied on video under the name of "video phylogeny". The first (and by now the only) work on video phylogeny is the one by Dias et al. [82].

Given two near-duplicate and frame-synchronized videos V_A and V_B, given a fixed set $T_{\vec{\beta}}$ of possible

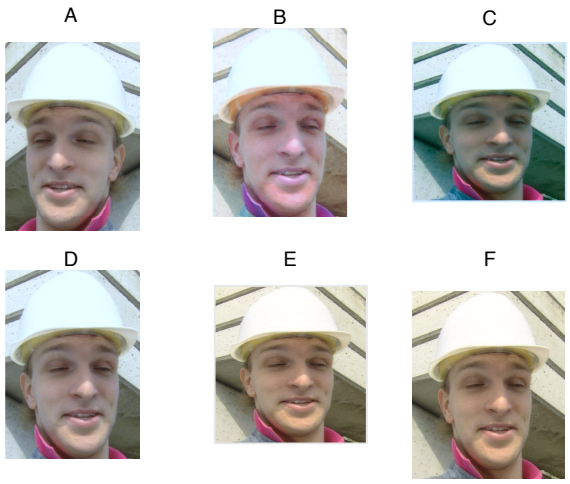

Fig. 8: An example of near-duplicate frames of the a video.

video transformations parameterized by $\vec{\beta}$, the dissimilarity between V_A and V_B is defined as

$$d_{V_A,V_B} = \min_{\vec{\beta}} \left| V_B - T_{\vec{\beta}}(V_A) \right|_L$$

where L is a comparison method. The best array of parameter $\vec{\beta}$ is searched by choosing a set of analogous frames from V_A and V_B, extracting robust interest points from frames and finding the affine warping between these points. Using this definition of dissimilarity, and for a chosen number f of frames taken from N near-duplicate videos, authors build f dissimilarity matrices, and each of them give the dissimilarity between all couples of videos evaluated on that frame. Instead of directly deriving the video phylogeny tree from these matrices, authors found more convenient to use the image phylogeny approach [80] to build f phylogeny trees, one for each set of frames, and then use a *tree reconciliation* algorithm that fuses information coming from these trees into the final video phylogeny tree (in our example, the phylogeny tree resulting from Figure 8 would be as in Figure 9). Experiments carried by authors

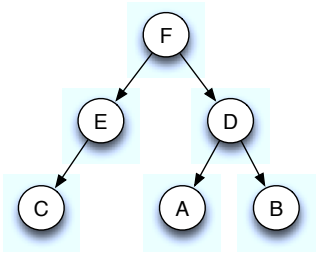

Fig. 9: The ground-truth phylogeny tree for the near-duplicate set in Figure 8.

show that the method is good (accuracy $\sim 90\%$) in finding the root of the tree (corresponding to the video originating the whole set) and also correctly classifies leafs 77.7% of the times, but the overall performances in terms of correctly positioned edges are still poor ($\sim 65.8\%$).

VI. Conclusions and Future Works

As it has been shown in the previous sections, video forensics is nowadays a hot research issue in the signal processing world opening new problems and investigation threads.

Despite several techniques have been mutuated from image forensics, video signals pose new challenges in the forensic application world because of the amount and the complexity of data to be processed and the wide employment of compression techniques, which may alter or erase footprints left by previous signal modifications.

This paper presented an overview of the state-of-the-art in video forensic techniques, underlying the future trends in this research field. More precisely, it is possible to divide video forensic techniques into three macro-areas concerning the acquisition, the compression, and the editing of the video signals. These three operations can be combined with different orders and iterated multiple times in the generation of the final multimedia signal. Current results show that it is possible to reconstruct simple processing chains (i.e., acquisition followed by compression, double compression, etc...) under the assumption that each processing step does not introduce an excessive amount of distortion on the signal. This proves to be reasonable since a severe deterioration of the quality of the signal would make it useless.

The investigation activity on video forensics is still an ongoing process since the complexity of video editing possibilities requires additional research efforts to make these techniques more robust.

Future research has still to investigate more complex processing chains where each operation on the signal may be iterated multiple times. These scenarios prove to be more realistic since the possibility of transmitting and distributing video content over the internet favors the diffusion of copies of the same multimedia content which has been edited multiple times.

Moreover, anti-forensic and counter-antiforensic strategies prove to be an interesting issue in order to identify those techniques that could be enacted by a malicious user in order to hide alterations on the signal and how to prevent them.

Future applications will include forensics strategies into existing multimedia applications in order to, e.g., provide the devices with built-in validating functionalities.

ACKNOWLEDGEMENT

The present work has been developed within the activity of the EU project REWIND (REVerse engineering of audio-VIsual coNtent Data) supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 268478.

REFERENCES

[1] H. Farid, "Exposing digital forgeries in scientific images," in *Proceedings of the 8th workshop on Multimedia and security (MM&Sec 2006)*, Sep. 26–27, 2006, pp. 29–36.

[2] D. Venkatraman and A. Makur, "A compressive sensing approach to object-based surveillance video coding," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2009)*, Taipei, Taiwan, Apr. 19–24, 2009, pp. 3513–3516.

[3] W. Wang and H. Farid, "Detecting re-projected video," in *Information Hiding*, ser. Lecture Notes in Computer Science, K. Solanki, K. Sullivan, and U. Madhow, Eds., vol. 5284. Springer, 2008, pp. 72–86.

[4] ——, "Exposing digital forgeries in video by detecting double MPEG compression," in *MM&Sec*, S. Voloshynovskiy, J. Dittmann, and J. J. Fridrich, Eds. ACM, 2006, pp. 37–47.

[5] H. T. Sencar and N. Memon, *Overview of State-of-the-art in Digital Image Forensics, Part of Indian Statistical Institute Platinum Jubilee Monograph series titled 'Statistical Science and Interdisciplinary Research,'*. World Scientific Press, 2008.

[6] R. Poisel and S. Tjoa, "Forensics investigations of multimedia data: A review of the state-of-the-art," in *IT Security Incident Management and IT Forensics (IMF), 2011 Sixth International Conference on*, Stuttgart, Germany, May 10 – 12, 2011, pp. 48 –61.

[7] J. Fridrich, "Image watermarking for tamper detection," in *ICIP (2)*, 1998, pp. 404–408.

[8] J. Eggers and B. Girod, "Blind watermarking applied to image authentication," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 3, 2001, pp. 1977 –1980 vol.3.

[9] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *ICIP*, 2000.

[10] S. Roy and Q. Sun, "Robust hash for detecting and localizing image tampering," in *ICIP (6)*. IEEE, 2007, pp. 117–120.

[11] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Hash-based identification of sparse image tampering," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2491–2504, 2009.

[12] M. Cossalter, M. Tagliasacchi, and G. Valenzise, "Privacy-enabled object tracking in video sequences using compressive sensing," in *Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS 2009)*, Genova, Italy, Sep. 2009, pp. 436–441.

[13] Y.-C. Lin, D. P. Varodayan, and B. Girod, "Image authentication using distributed source coding," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 273–283, 2012.

[14] G. Valenzise, M. Tagliasacchi, S. Tubaro, G. Cancelli, and M. Barni, "A compressive-sensing based watermarking scheme for sparse image tampering identification," in *Proc. of the 16th IEEE International Conference on Image Processing (ICIP 2009)*. Cairo, Egypt: IEEE, Nov. 7–10, 2009, pp. 1265 – 1268.

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

[16] M. Chen, J. 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.

[17] A. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *Signal Processing, IEEE Transactions on*, vol. 53, no. 10, pp. 3948 – 3959, oct. 2005.

[18] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *MM&Sec*, S. Voloshynovskiy, J. Dittmann, and J. J. Fridrich, Eds. ACM, 2006, pp. 48–55.

[19] I. Yerushalmy and H. Hel-Or, "Digital image forgery detection based on lens and sensor aberration," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 71–91, 2011.

[20] S. Milani, M. Tagliasacchi, and M. Tubaro, "Discriminating multiple jpeg compression using first digit features," in *to appear on Proc. of the 37^{th} International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, Mar. 25 – 30, 2012, pp. 2253–2256.

[21] D. Fu, Y. Q. Shi, and W. Su, "A generalized benfords law for jpeg coefficients and its applications in image forensics," in *Proceedings of SPIE, Volume 6505, Security, Steganography and Watermarking of Multimedia Contents IX*, vol. 6505, Jan. 28 – Feb. 1, 2009, pp. 39–48.

[22] H. Liu and I. Heynderickx, "A no-reference perceptual blockiness metric," in *ICASSP*. IEEE, 2008, pp. 865–868.

[23] W. S. Lin, S. K. Tjoa, H. V. Zhao, and K. J. R. Liu, "Digital image source coder forensics via intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 460–475, 2009.

[24] Z. Fan and R. L. de Queiroz, "Maximum likelihood estimation of JPEG quantization table in the identification of bitmap compression history," in *ICIP*, 2000.

[25] ——, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, 2003.

[26] J. Lukás and J. Fridrich, "Estimation of primary quantization matrix in double compressed jpeg images," in *Proc. of DFRWS*, 2003.

[27] Z. C. Lin, J. F. He, X. Tang, and C. K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, nov 2009.

[28] T. Bianchi and A. Piva, "Detection of non-aligned double JPEG compression with estimation of primary compression parameters," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, sept. 2011, pp. 1929 –1932.

[29] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *ICASSP*. IEEE, 2011, pp. 2444–2447.

[30] T. Bianchi and A. Piva, "Detection of nonaligned double jpeg compression based on integer periodicity maps," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 2, pp. 842 –848, april 2012.

[31] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3-1, pp. 450–461, 2007.

[32] ——, "Exposing digital forgeries through specular highlights on the eye," in *Information Hiding*, ser. Lecture Notes in Computer Science, T. Furon, F. Cayre, G. J. Doërr, and P. Bas, Eds., vol. 4567. Springer, 2007, pp. 311–325.

[33] W. Zhang, X. Cao, J. Zhang, J. Zhu, and P. Wang, "Detecting photographic composites using shadows," in *ICME*. IEEE, 2009, pp. 1042–1045.

[34] V. Conotter, G. Boato, and H. Farid, "Detecting photo manipulation on signs and billboards," in *ICIP*. IEEE, 2010, pp. 1741–1744.

[35] K. Kurosawa, K. Kuroki, and N. Saitoh, "Ccd fingerprint method-identification of a video camera from videotaped images," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 3, 1999, pp. 537 –540 vol.3.

[36] G. C. Holst, *CCD Arrays, Cameras, and Displays, 2nd edition*. CD Publishing & SPIE Pres, 1998.

[37] 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 *Digital Signal Processing, 2009 16th International Conference on*, july 2009, pp. 1 –7.

[38] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Source digital camcorder identification using sensor photo response non-uniformity," in *Proceedings of SPIE*, 2007.

[39] M. Kivanc Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 6, mar 1999, pp. 3253 –3256 vol.6.

[40] W. van Houten, Z. J. M. H. Geradts, K. Franke, and C. J. Veenman, "Verification of video source camera competition (camcom 2010)," in *ICPR Contests*, ser. Lecture Notes in Computer Science, D. Ünay, Z. Çataltepe, and S. Aksoy, Eds., vol. 6388. Springer, 2010, pp. 22–28.

[41] W. van Houten and Z. J. M. H. Geradts, "Using sensor noise to identify low resolution compressed videos from youtube," in *IWCF*, ser. Lecture Notes in Computer Science, Z. J. M. H. Geradts, K. Franke, and C. J. Veenman, Eds., vol. 5718. Springer, 2009, pp. 104–115.

[42] ——, "Source video camera identification for multiply compressed videos originating from youtube," *Digital Investigation*, vol. 6, no. 1-2, pp. 48–60, 2009.

[43] M.-J. Lee, K.-S. Kim, H.-Y. Lee, T.-W. Oh, Y.-H. Suh, and H.-K. Lee, "Robust watermark detection against d-a/a-d conversion for digital cinema using local auto-correlation function," in *ICIP*. IEEE, 2008, pp. 425–428.

[44] M.-J. Lee, K.-S. Kim, and H.-K. Lee, "Digital cinema watermarking for estimating the position of the pirate," *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 605–621, 2010.

[45] J.-W. Lee, M.-J. Lee, T.-W. Oh, S.-J. Ryu, and H.-K. Lee, "Screenshot identification using combing artifact from interlaced video," in *Proceedings of the 12th ACM workshop on Multimedia and security*, ser. MM&Sec '10. New York, NY, USA: ACM, 2010, pp. 49–54. [Online]. Available: http://doi.acm.org/10.1145/1854229.1854240

[46] S. Bayram, H. T. Sencar, and N. D. Memon, "Video copy detection based on source device characteristics: a complementary approach to content-based methods," in *Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30-31, 2008*, M. S. Lew, A. D. Bimbo, and E. M. Bakker, Eds. ACM, 2008, pp. 435–442. [Online]. Available: http://doi.acm.org/10.1145/1460096.1460167

[47] G. Wallace, "The jpeg still picture compression standard," *Consumer Electronics, IEEE Transactions on*, vol. 38, no. 1, pp. xviii –xxxiv, feb 1992.

[48] H. Li and S. Forchhammer, "MPEG2 video parameter and no reference PSNR estimation," in *Picture Coding Symposium, 2009. PCS 2009*, may 2009, pp. 1 –4.

[49] S. Ye, Q. Sun, and E.-C. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *ICME*. IEEE, 2007, pp. 12–15.

[50] Y. Chen, K. S. Challapali, and M. Balakrishnan, "Extracting coding parameters from pre-coded MPEG-2 video," in *ICIP (2)*, 1998, pp. 360–364.

[51] M. Tagliasacchi and S. Tubaro, "Blind estimation of the QP parameter in H.264/AVC decoded video," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, april 2010, pp. 1 –4.

[52] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Estimating QP and motion vectors in H.264/AVC video from decoded pixels," in *Proceedings of the 2nd ACM workshop on Multimedia in forensics, security and intelligence*, ser. MiFor '10. New York, NY, USA: ACM, 2010, pp. 89–92. [Online]. Available: http://doi.acm.org/10.1145/1877972.1877995

[53] J. He, Z. Lin, L. Wang, and X. Tang, "Detecting doctored JPEG images via DCT coefficient analysis," in *Lecture Notes in Computer Science*. Springer, 2006, pp. 423–435.

[54] T. Pevny and J. Fridrich, "Estimation of primary quantization matrix for steganalysis of double-compressed JPEG images," *Proceedings of SPIE*, vol. 6819, pp. 681 911–681 911–13, 2008. [Online]. Available: http://link.aip.org/link/PSISDG/v6819/i1/p681911/s1&Agg=doi

[55] W. Luo, M. Wu, and J. Huang, "MPEG recompression detection based on block artifacts," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, vol. 6819, Mar. 2008.

[56] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *Proceedings of the 11th ACM workshop on Multimedia and security*, ser. MM&Sec '09. New York, NY, USA: ACM, 2009, pp. 39–48. [Online]. Available: http://doi.acm.org/10.1145/1597817.1597826

[57] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification," in *to appear on Proc. of the 37^{th} International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, Mar. 25 – 30, 2012, pp. 2257–2260.

[58] A. R. Reibman and D. Poole, "Characterizing packet-loss impairments in compressed video," in *ICIP (5)*. IEEE, 2007, pp. 77–80.

[59] A. R. Reibman, V. A. Vaishampayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 327–334, 2004.

[60] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-reference video quality monitoring for H.264/AVC coded video," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 932–946, 2009.

[61] G. Valenzise, S. Magni, M. Tagliasacchi, and S. Tubaro, "Estimating channel-induced distortion in H.264/AVC video without bitstream information," in *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, june 2010, pp. 100 –105.

[62] ——, "No-reference pixel video quality monitoring of channel-induced distortion," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[63] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-forensics of JPEG compression," in *ICASSP*. IEEE, 2010, pp. 1694–1697.

[64] ——, "Undetectable image tampering through JPEG compression anti-forensics," in *ICIP*. IEEE, 2010, pp. 2109–2112.

[65] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "The cost of jpeg compression anti-forensics," in *ICASSP*. IEEE, 2011, pp. 1884–1887.

[66] G. Valenzise, V. Nobile, M. Tagliasacchi, and S. Tubaro, "Countering jpeg anti-forensics," in *ICIP*, B. Macq and P. Schelkens, Eds. IEEE, 2011, pp. 1949–1952.

[67] M. C. Stamm and K. J. R. Liu, "Anti-forensics for frame deletion/addition in mpeg video," in *ICASSP*. IEEE, 2011, pp. 1876–1879.

[68] N. Mondaini, R. Caldelli, A. Piva, M. Barni, and V. Cappellini, "Detection of malevolent changes in digital video for

forensic applications," in *Proceedings of SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX*, E. J. D. III and P. W. Wong, Eds., vol. 6505, no. 1. SPIE, 2007, p. 65050T.

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

[70] 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, 2010.

[71] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[72] W. Wang and H. Farid, "Exposing digital forgeries in interlaced and deinterlaced video," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3-1, pp. 438–449, 2007.

[73] J. Zhang, Y. Su, and M. Zhang, "Exposing digital video forgery by ghost shadow artifact," in *Proceedings of the First ACM workshop on Multimedia in forensics*, ser. MiFor '09. New York, NY, USA: ACM, 2009, pp. 49–54. [Online]. Available: http://doi.acm.org/10.1145/1631081.1631093

[74] V. Conotter, J. O'Brien, and H. Farid, "Exposing digital forgeries in ballistic motion," *Information Forensics and Security, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[75] L.-W. Kang, C.-Y. Hsu, H.-W. Chen, C.-S. Lu, C.-Y. Lin, and S.-C. Pei, "Feature-based sparse representation for image similarity assessment," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1019–1030, Oct. 2011.

[76] L.-W. Kang, C.-Y. Hsu, H.-W. Chen, and C.-S. Lu, "Secure sift-based sparse representation for image copy detection and recognition," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME 2010)*. Singapore: IEEE, Jul. 2010, pp. 1248–1253.

[77] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *MM&Sec*, D. Kundur, B. Prabhakaran, J. Dittmann, and J. J. Fridrich, Eds. ACM, 2007, pp. 35–42.

[78] M. Stamm and K. Liu, "Anti-forensics for frame deletion/addition in mpeg video," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 1876 –1879.

[79] M. Stamm, S. Tjoa, W. Lin, and K. Liu, "Undetectable image tampering through jpeg compression anti-forensics," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, sept. 2010, pp. 2109 –2112.

[80] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on*, dec. 2010, pp. 1 –6.

[81] A. De Rosa, F. Uccheddu, A. Costanzo, A. Piva, and M. Barni, "Exploring image dependencies: a new challenge in image forensics," in *Proceeding of SPIE*, 2010, pp. X1 – X12.

[82] Z. Dias, A. Rocha, and S. Goldenstein, "Video phylogeny: Recovering near-duplicate video relationships," in *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*, dec. 2011.

[83] *Proceedings of the International Conference on Image Processing, ICIP 2007, September 16-19, 2007, San Antonio, Texas, USA*. IEEE, 2007.

[84] *Proceedings of the International Conference on Image Processing, ICIP 2010, September 26-29, Hong Kong, China*. IEEE, 2010.

[85] S. Voloshynovskiy, J. Dittmann, and J. J. Fridrich, Eds., *Proceedings of the 8th workshop on Multimedia & Security, MM&Sec 2006, Geneva, Switzerland, September 26-27, 2006*. ACM, 2006.

[86] *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*. IEEE, 2011.

LIST OF FIGURES

1 Typical acquisition pipeline 7
2 A simple field weaving algorithm for video de-interlacing. 11
3 Simplified block diagram of a conventional video codec. 13
4 Original and compressed frames of a standard video sequence. 14
5 Histograms of DCT coefficients before and after quantization. 15
6 Example of coding artifacts for editing detection 25
7 Video interpolation based on line averaging. 26
8 An example of near-duplicate frames of the a video. 30
9 The ground-truth phylogeny tree for the near-duplicate set in Figure 8. 30