Video Integrity Verification and GOP Size Estimation via Generalized Variation of Prediction Footprint

David Vázquez-Padín, *Member, IEEE,* Marco Fontani, Dasara Shullani, Fernando Pérez-González, *Fellow, IEEE,* Alessandro Piva, *Fellow, IEEE,* and Mauro Barni, *Fellow, IEEE*

***Abstract*—The Variation of Prediction Footprint (VPF), formerly used in video forensics for double compression detection and GOP size estimation, is comprehensively investigated to improve its acquisition capabilities and extend its use to video sequences that contain bi-directional frames (B-frames). By relying on a universal rate-distortion analysis applied to a generic double compression scheme, we first explain the rationale behind the presence of the VPF in double compressed videos and then justify the need of exploiting a new source of information such as the motion vectors, to enhance the VPF acquisition process. Finally, we describe the shifted VPF induced by the presence of B-frames and detail how to compensate the shift to avoid misguided GOP size estimations. The experimental results show that the proposed Generalized VPF (G-VPF) technique outperforms the state of the art, not only in terms of double compression detection and GOP size estimation, but also in reducing computational time.**

***Index Terms*—Double compression detection, GOP size estimation, video forensics, B-frames, rate distortion optimization.**

I. Introduction

When compared with the proliferation of digital image forensic techniques, video forensic research has evolved at a much slower pace. This is probably caused by the challenging nature of video processing, which includes many different, highly configurable coding algorithms and deals with enormous amounts of data compared to images, making it difficult to create comprehensive standard datasets for assessing the validity of the developed tools. Yet, deciding about the integrity and authenticity of video sequences is of the utmost importance in several applications, even because of the alleged trustworthiness of videos due to the (ever more erroneous) belief that videos are more difficult to tamper with than images. According to the guidelines issued by the Scientific Working Group on Imaging Technologies, video integrity verification consists in assessing whether the video is complete and unaltered since the time of acquisition [1]. This definition directly implies that re-encoding a video breaks its integrity. Video authentication, in turn, is the process of substantiating that the content is an accurate representation of what it purports to be [2]. Since the vast majority of video editing operations are carried out on the decompressed version of the video, video tampering typically undergoes two compression stages: the first during acquisition, and the second after processing. Therefore, double compression detection, that is, the task of blindly assessing whether a video was compressed either once or (at least) twice, has gained great attention as a direct indication of integrity violation and as an indirect indication of tampering.

In this paper, we target the problem of video integrity verification following a standard practice required in forensic science [3], i.e., we start with a deep study of the traces left behind in a double video coding process and then we design a forensic solution based on such analysis. In doing so, we ensure that the derived algorithm is **explainable**, which is an essential characteristic in legal contexts (e.g., when a forensic analyst must provide objective judgments on some observed data in a court of law). In addition, the acquired knowledge from the theoretical characterization of the problem allows us to precisely establish the limits of the proposed algorithm, in constrast to techniques based on training complex machines whose applicability in the forensic field is becoming questioned [4], because: i) generalization issues may occur, given that the output results mostly depend on the training data, and ii) there is still a lack of theoretical understanding about the decision making process leading to the final result (especially for deep learning), which usually reflects in hard explanability of the outputs [5].

A. Literature review

The first works dealing with video integrity verification through the detection of double video compression date back to 2006, when Wang and Farid proposed a method to detect double encoding of MPEG-2 video sequences [6] exploiting double quantization traces. In the same work, the authors observed that when one or more video frames are removed before re-encoding the video, a periodic increase in the prediction error is observed since some frames are re-encoded

This work was supported in part by the Agencia Estatal de Investigación (Spain) and the European Regional Development Fund (ERDF) under the project WINTER (TEC2016-76409-C2-2-R), in part by the Xunta de Galicia and the European Union (ERDF) under projects Agrupación Estratéxica Consolidada de Galicia accreditation 2016-2019 and Grupo de Referencia ED431C2017/53, and in part by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under Grants number FA8750-16-2-0188 and FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

D. Vázquez-Padín and F. Pérez-González are with the Department of Signal Theory and Communications, University of Vigo, 36310 Vigo, Spain (e-mail: dvazquez@gts.uvigo.es; fperez@gts.uvigo.es).

M. Fontani is with Amped Software, Loc. Padriciano 99, 34149, Trieste, Italy (e-mail: marco.fontani@ampedsoftware.com).

D. Shullani, and A. Piva are with the Department of Information Engineering, University of Florence, 50139 Florence, Italy (e-mail: dasara.shullani@unifi.it; alessandro.piva@unifi.it).

M. Barni is with the Department of Information Engineering and Mathematics, University of Siena, 53100 Siena, Italy (e-mail: barni@dii.unisi.it).

in a Group Of Pictures (GOP) different from the one of the first compression. Later on, Stamm *et al.* [7] expanded this idea proposing an automatic way for detecting such a periodic artifact. While the above works considered a setting wherein some frames are removed between the two compression steps, it turns out that even simply re-encoding the video with a different GOP structure leads to similar artifacts. Indeed, when the GOP structure changes between the first and second compression steps, some frames that were coded as I-frames in the first video stream are re-encoded as predicted P- or B-frames; these frames are sometimes referred to as "relocated I-frames" in the literature [8], [9]. This fact has an impact on the prediction residual and on the distribution of different types of MacroBlocks (MBs). Such an observation was exploited in [10] by showing that the anomalous use of certain MB types in double compressed videos, i.e., the so-called *Variation of Prediction Footprint* (VPF), can be used to detect whether a given video is compressed twice and, in case it is, to estimate the size of the GOP used in the first compression.

The main limitations of the method proposed in [10] are that: i) the GOP size of the first compressed stream is assumed to be constant, and ii) the use of B-frames is not allowed in the second compression. More recently, Chen *et al.* [11] proposed to improve the VPF acquisition by computing the distribution of the average prediction residual in each frame and analyzing to which extent such a prediction varies between adjacent frames; the periodicity of such a variation is then studied in a way that closely resembles [10]. While [11] slightly improves the performance of the original VPF analysis, it also shares the main limitations of [10] since it does not support B-frames, and it assumes a constant GOP in the first encoding. Moreover, the method was tested only on H.264 videos encoded with Constant BitRate (CBR). He *et al.* [9] proposed to analyze the behavior of motion vectors in P-frames, limiting the analysis to the static background of videos. In those regions, for I-frames that were re-encoded as P-frames, predicted macroblocks behave differently in terms of motion vector magnitude and energy of prediction residual. A problem with the method in [9] is that it works only for static videos, and it has been proposed and tested using MPEG-4 only for the last compression. In addition, it does not account for the possible presence of B-frames. More recently, the same authors proposed a method that improves the robustness of the VPF for videos with rapidly changing content [12]. The extension consists in measuring the strength of the blocking artifacts in each frame, and combining this information with the VPF before running the periodicity analysis. The method is defined and tested on MPEG-4 videos only, having a constant GOP structure and without B-frames. Moreover, measuring the strength of blocking artifacts requires decoding and analyzing all the frames, thus increasing significantly the computational complexity. Still leveraging on the fact that in a double compressed video the time correlation is weak for I-frames that are re-encoded as P-frames, Yao *et al.* [13] observed that these frames require a larger number of bits in the bitstream. Thus, the bit size of each frame is considered as the main feature in [13], allowing a fast and accurate detection of double encoding. As all previous schemes, the method proposed in [13] cannot cope with B-frames; in addition it has been tested by using H.264 only for the last compression and CBR as coding strategy. Finally, a different approach to the problem was proposed in [8], where authors employ a deep learning approach to distinguish, in a frame-wise fashion, relocated I-frames from other frames (which means: single encoded frames or double encoded frames that maintained the original type). In contrast to all previously mentioned approaches, the latter method allows a fine-grained classification as opposite to a video-wise classification. The method, however, has been designed and tested on videos encoded with H.264, in CBR coding mode only, and without the use of B-frames.

Bestagini *et al.* introduced a completely different approach to the analysis of double compressed video sequences: they propose to re-encode the video multiple times trying to match the compression settings of the first encoding [14]. Indeed, thanks to the (partial) idempotency property of video compression, and under the hypothesis that the second encoding does not alter the video significantly, when the first compression settings are matched the re-compressed video will not change much. One of the main advantages of this method is that it reveals many details about the first compression, including the employed codec, the quality factor, and the GOP size. The method was tested on MPEG-2, MPEG-4 and H.264 videos, and also GOP structures with B-frames were considered. The main limitation is related to the huge computational effort required to re-encode the input video tens or even hundreds of times, depending on the number of tested combinations. Moreover, by definition, this method is expected to work when the first compression is carried out under a Variable BitRate (VBR) control mode, since idempotency hardly holds for the CBR case. Finally, [14] considers only the estimation of the parameters of the first encoding step for double compressed videos, but it does not define any rule for double compression detection.

Contrarily to all the methods mentioned so far, Jiang *et al.* [15] focused on double compression detection in the case where the same coding parameters are used in the first and second compressions. The authors introduced a pool of features that capture the quality degradation due to recompression, the features are then used to train an SVM classifier. Different feature extraction algorithms are proposed for MPEG-2/4 and H.264 videos. The method is defined and tested for MPEG-2, MPEG-4, and H.264, using CBR, VBR, and even the more advanced Constant Rate Factor (CRF) bitrate control mode. The main drawbacks of this scheme are the necessity of re-encoding the video as part of the feature extraction procedure and also the fact that B-frames are not considered.

Finally, some methods have been recently proposed for double compression detection in HEVC videos. Costanzo and Barni [16] focused on the case of H.264 videos re-compressed with higher quality to HEVC, and observed that the first compression influences the decision strategy of the latter encoding rearding the motion prediction modes in bi-directionally predicted frames. Liang *et al.* proposed in [17] to feed histograms of Prediction Unit (PU) partition types to a support vector machine to uncover double compression, limiting to P-frames and targeting the case where the last bitrate is higher than the former (the so-called "fake quality scenario"); interestingly for us, the PU partition types are a generalized version of the MB types to cover larger block sizes and other prediction types in HEVC, thus anticipating the generalization of the theory and algorithms presented in this paper to the emerging HEVC standard. Li *et al.* [18], instead, observed that double encoding has an impact both on the transform unit size and on the statistics of DCT coefficients (double quantization effect); they used such information to train a classifier capable of detecting recompressed videos.

B. Contributions

From the above discussion, it is evident that most of the existing schemes for double encoding detection roughly leverage on the same idea, that is, studying the artifacts introduced in P-frames that were previously encoded as I-frames. While being rather easy to justify intuitively, the theoretical reasons underlying these artifacts have never been investigated with the depth that a trustworthy use in forensic scenarios demands. Moreover, most of the existing methods cannot deal with the presence of B-frames in the second compression step, which severely limits their applicability in real cases. The only method providing support for B-frames, namely [14], becomes computationally intractable for modern high-resolution videos. Most methods have been tested on a limited set of codecs and bitrate control modes. With the aim of solving these issues, this paper offers a four-fold contribution:

1) **An in-depth theoretical analysis of the artifacts that explain the presence of the VPF in double compressed videos:** in contrast to [10] where only few shallow hints are provided, here we offer a unifying view of all the video forensic techniques which are directly or indirectly related to the VPF and which permits to understand why such a footprint appears and to approximately predict its strength, contributing to the explainability of this branch of video forensics.
2) **An improvement of the VPF acquisition process:** by exploiting the insights provided by the theoretical analysis, information from the motion vectors, which was not considered in [10], is used to better capture the VPF with sizable improvements on performance.
3) **A novel algorithm to compute the VPF under the presence of B-frames in the second compression step:** this represents a major contribution since, to the best of our knowledge, none of the schemes proposed so far is able to work in the presence of B-frames, with a reasonable complexity.
4) **An extensive experimental campaign:** by testing the effectiveness of the derived method under a wide variety of settings we conduct a far more exhaustive validation than in [10], encompassing some of the latest and most advanced video coding standards.

In the rest of this paper, we will refer to the newly proposed approach for capturing the VPF, featuring contributions at points 2) and 3), with the name Generalized VPF (G-VPF).

The paper is structured as follows: in Section II we recall the video coding concepts necessary to understand the subsequent sections. Section III defines the problem addressed by the paper and introduces the notation used afterwards. In Section IV we analytically explain the presence of the VPF and its shifted version induced by B-frames, while in Section V we detail the proposed G-VPF technique for better capturing the VPF and performing double compression detection and GOP size estimation. Finally, Section VI provides a thorough experimental validation and comparison with existing methods, while Section VII concludes the paper.

II. Basics on Video Coding

Throughout the paper, we consider three major video coding standards, namely MPEG-2 [19], MPEG-4 (Part 2) [20], and H.264 [21]. Although each standard has its own coding characteristics, they are designed over a common block-based hybrid video coding architecture and, consequently, the three standards share several syntax features.

According to the block-based structure, each frame of a captured video sequence is divided into MBs of size 16×16 samples, that are encoded with a suitable coding mode from each particular standard.[1] Different types of frames are defined depending on the prediction process carried out during the encoding. The three standards share the definition of intra-coded frames (or I-frames), where each MB is encoded without any reference to other frames within the video sequence,[2] and inter-coded frames, where the MBs can additionally be predicted from already coded and reconstructed frames (i.e., reference frames), which leads to two possible types of frames: P-frames and B-frames. The MBs in P-frames can only be predicted by referring to previously encoded frames, while those in B-frames can be predicted from past and/or future reference frames.

The different types of frames can be grouped into sequences, creating a GOP which can be closed or open. A closed GOP is an encoding of successive frames that can be completely decoded without any reference to other GOPs [22]. On the other hand, an open GOP can only be entirely decoded by referring to other GOPs. Here, we will only consider closed GOPs that are composed of a single I-frame that indicates the beginning of the group and some combinations of P- and B-frames.

Each video coding standard defines a number of coding modes for each type of frame, with the final goal of increasing the coding efficiency. Table I shows some of these coding modes arranged according to the standard and the type of prediction carried out in each case.[3] As noted above, the MBs of an I-frame can only be encoded by means of intra coding modes (i.e., I-MBs) with or without prediction depending on the standard, while the MBs of a P-frame can be further encoded through the inter coding modes that use reference

[1] Only progressive-scan videos and full-frame encodings are considered in this work, thus, for the sake of clarity, the term "frame" is used to represent a picture or a slice independently of the standard.

[2] In contrast to MPEG-2/4 standards, in H.264 an intra prediction is carried out in the spatial domain by referring to neighboring samples of already coded blocks (always within the same frame).

[3] Note that even if the same name is used, the functionality of each mode could be different from one standard to another. Other existing modes and sub-modes are not presented for the sake of brevity.

TABLE I
CODING MODES FOR EACH STANDARD

Coding Mode	Intra coding		Inter coding			
Prediction / Standard	No prediction	Intra Prediction	No residue	Past	Future	Bipredictive
MPEG-2	INTRA-16×16	×	SKIP	INTER-16×16		
MPEG-4	INTRA-16×16	×	SKIP	INTER-16×16 INTER-8×8		
H.264	×	INTRA-16×16 INTRA-4×4	SKIP DIRECT	INTER-16×16 INTER-16×8 INTER-8×16 INTER-8×8		
MB type	I-MB		S-MB	P-MB	F-MB	B-MB

I-frames

P-frames

B-frames

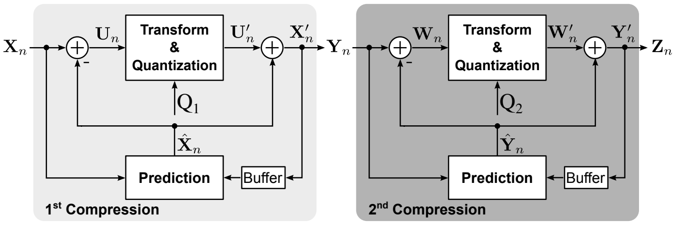


Fig. 1. Double compression scheme: the left block diagram shows the first compression stage and, correspondingly, the right block depicts the structure of the second compression stage.

frames from the past, either with residual data (i.e., P-MBs) or without any residue nor motion vector (i.e., S-MBs). For the case of B-frames, the MBs can additionally be encoded using future reference frames (i.e., F-MBs) and bipredictive coding modes (i.e., B-MBs), which build the prediction through a weighted average of past and future frames.

The last row of Table I shows the five MB types we will consider along the paper, which are grouped in line with the type of frame that can use each of them. Accordingly, an encoded I-frame only admits I-MBs, whereas a P-frame can be composed of I-MBs, P-MBs, and S-MBs, and finally a B-frame can contain I-MBs, P-MBs, S-MBs, F-MBs, and B-MBs. The procedure that each encoder follows to select which of these MB types is the most suitable in terms of coding efficiency is described in Sect. III-A.

III. PROBLEM FORMULATION AND MODELING

The goal of this work is twofold: to detect video double compression, and to estimate the size of the GOP employed during the first encoding so that a more elaborate video forensic analysis can be accomplished (see for instance [23], [24]). To that end, we consider a nowadays common video double encoding scenario where the first compression is carried out by an acquisition device, typically a smartphone camera, and the second compression is either performed by a video editing tool or by a video storage service provider such as YouTube, Vimeo, etc.

In this scenario, we assume that a sequence of N frames with time indices $n = 0, \dots, N-1$ is first compressed with a constant closed GOP of length G_1 which is solely composed of I- and P-frames. We discard the use of B-frames because in practice most of the smartphone cameras do not usually meet the computational resources to encode B-frames in real-time.[4] Regarding the bitrate control, we consider that any of the VBR, CBR, or CRF modes can be applied during the first compression, however, to keep the model analytically tractable, we assume that the quality of the compressed video is controlled by a fixed quantization parameter Q_1 that is proportional to the compression strength, i.e., the larger the value of Q_1 the stronger the compression. On the other hand, the second compression is conducted with a closed GOP of length G_2 not necessarily constant, but different from any integer multiple or submultiple of G_1. We assume that no temporal shift nor any inter- or intra-frame forgery is introduced between both compressions. In this case, apart from I- and P-frames, the GOP of the second encoding is assumed to contain B-frames, since video editing tools or video service providers are not subject to strict time constraints and so they allow the use of B-frames for improving the coding efficiency. To identify the type of encoding a frame has undergone at time index n, we define the sets I, P, and B which respectively contain the time indices of I-, P-, and B-frames (the subindex 1 or 2 will be added to explicitly refer to the first or second encoding). As for the first compression, we assume that the quantization is directed by a single parameter Q_2.

Concerning the video encoders, we assume that any of the three contemplated standards can be indistinctly applied at each compression stage. However, we limit the applicability of the extensions made available by H.264, such as the multi-frame motion-compensation [25] and the use of hierarchical B-frames [26]. In particular, this means that for encoding P-frames, we assume that only the last encoded I/P-frame is used as a reference frame, while for encoding B-frames, only two reference frames are considered: the I/P-frame that precedes the B-frame and the I/P-frame that succeeds it. [5]

Since we are interested in analyzing the double compression processing chain independently of the standard employed during each compression, we make use of the general framework depicted in Fig. 1. The left block diagram models the coding scheme and variables that are used under the first compression, while the right block diagram shows the second compression counterpart. In particular, the left scheme in Fig. 1 models how a given MB from the originally captured scene at a particular time index n, denoted by $\mathbf{X}_n$, is predicted based on a set of previously coded and reconstructed samples at different time and spatial locations stored in a buffer. Depending on the type of frame, $\mathbf{X}_n$ is predicted according to the available coding modes shown in Table I, obtaining the prediction $\hat{\mathbf{X}}_n$.[6] After the prediction, a residual signal is obtained as $\mathbf{U}_n = \mathbf{X}_n - \hat{\mathbf{X}}_n$, whose samples are transformed applying the Discrete Cosine Transform (DCT) on an 8×8 block-basis for MPEG-2/4 or on a 4×4 block-basis in the case of H.264. In the DCT domain, each coefficient is quantized by a scalar quantizer with a particular step size Δ_1, which is

[4] Note, however, that the effect of B-frames in the first encoding is later empirically evaluated in Sect. VI-E.

[5] This constrained scenario simplifies the upcoming analysis, but it does not prevent the use of our approach on encoded videos with the mentioned H.264 extensions. These extensions will only affect the method's performance.

[6] In the particular case of MPEG-2/4, no prediction is computed for the intra coding modes, thus having $\hat{\mathbf{X}}_n = 0$.

controlled by the aforementioned quantization parameter Q_1. Finally, the reconstructed samples $\mathbf{X}'_n$ are recovered by adding back the de-quantized and inverse transformed samples $\mathbf{U}'_n$ to the prediction $\hat{\mathbf{X}}_n$, such that $\mathbf{X}'_n = \mathbf{U}'_n + \hat{\mathbf{X}}_n$.

The above description straightforwardly extends to the second compression block on the right of Fig. 1: the source and predicted samples are denoted by $\mathbf{Y}_n$ and $\hat{\mathbf{Y}}_n$, respectively, the residual signal by $\mathbf{W}_n$ and its reconstructed version by $\mathbf{W}'_n$; in this case, the quantization parameter is Q_2 and the recovered samples are indicated by $\mathbf{Y}'_n$.

A. Description of video coding strategies

The main goal of an encoder is to efficiently represent an input sequence of pictures using the available syntax elements of a particular video coding standard. To achieve this, the encoder minimizes under certain constraints the distortion between the original sequence and its reconstruction after encoding. Here, we only use a maximum rate r as a constraint, because no error-prone transmissions are taken into account. Therefore, the general problem of finding the most suitable syntax elements for encoding a set of samples $\mathcal{X}$ stemming from a captured scene can be formulated as

$$\begin{aligned} \text{minimize} \quad & D(\mathcal{X}, \mathcal{X}') \\ \text{subject to} \quad & R(\mathcal{X}') \leq r, \end{aligned} \tag{1}$$

where $D(\mathcal{X}, \mathcal{X}')$ measures the distortion between the original source samples $\mathcal{X}$ and their reconstructed version $\mathcal{X}'$, while $R(\mathcal{X}')$ denotes the number of bits needed to encode the syntax elements that allow the reconstruction of the samples $\mathcal{X}'$.

There are multiple ways of addressing the above minimization problem. In practice the overall problem is split into different parts, such that specific rules can be applied for MB coding-mode decision, motion estimation, and quantization [27]. In this case, since the footprint we rely on for double compression detection is directly related to the methodology adopted for MB coding-mode decision, we focus on the strategies employed to solve this problem. One of the first (and also less sophisticated) encoding strategies that were proposed by the ITU-T Video Coding Experts Group is the one described in Test Model Number 9 (TMN-9) [28], which is fundamentally based on the use of thresholds to compare the different distortion values achieved by each coding mode. As noted in [27], this strategy is very convenient because it is computationally cheap, however, the coding performances are much lower than those achieved by novel encoding strategies that are based on Rate-Distortion (R-D) theory. In that sense, in [27] the optimization problem in (1), is solved by using Lagrange multipliers, resulting in the following unconstrained problem

$$\text{minimize} \quad J(\mathcal{X}, \mathcal{X}') = D(\mathcal{X}, \mathcal{X}') + \lambda R(\mathcal{X}'),$$

where λ represents the Lagrange multiplier associated to the optimum solution of (1) for a particular value of r. When this Lagrangian-based encoding strategy is applied to the selection of the MB coding-mode, the minimization of the Lagrangian functional for each coding mode c applied to the source macroblock $\mathbf{X}_n$ at time index n (i.e., $J_{\mathrm{c}}(\mathbf{X}_n, \mathbf{X}'_n)$), yields the following minimization problem

$$\text{MB type} = \arg\min_{\mathrm{c}\in\mathcal{C}} D(\mathbf{X}_n, \mathbf{X}'_n) + \lambda_{\mathrm{c}} R(\mathbf{X}'_n), \tag{2}$$

where c denotes any of the available coding modes from Table I, thus having $\mathcal{C} \triangleq \{\text{I-MB}, \text{P-MB}, \text{S-MB}, \text{F-MB}, \text{B-MB}\}$, and λ_{c} is the corresponding Lagrange multiplier for the selected coding mode (whose value is obtained as a function of the quantization parameter, as described in [29]). The distortion measure $D(\mathbf{X}_n, \mathbf{X}'_n)$ is commonly taken as the Sum of Squared Differences (SSD) between the reconstructed block $\mathbf{X}'_n$ and the source block $\mathbf{X}_n$, such that $D(\mathbf{X}_n, \mathbf{X}'_n) = \|\mathbf{X}_n - \mathbf{X}'_n\|_2^2$, while the rate measure $R(\mathbf{X}'_n)$ is generally an estimate of the number of bits required for encoding $\mathbf{X}'_n$ with the coding mode c.

The above Lagrangian-based strategy is nowadays established as the de-facto paradigm in video coding because it significantly improves the coding efficiency of a video sequence independently of the standard employed. So, assuming the use of such encoding strategy in our coding scheme, in the next section we will analyze how the consecutive application of two compressions affects the decisions taken by the encoder in each compression stage, which ultimately leads to the anomalous variation of prediction footprint, i.e., the VPF, first identified in [10]. Since the derivation of the R-D function of the double-compression processing chain could be overwhelmingly complicated due to the different types of frames and coding modes involved, we rather perform the analysis of the VPF by relying on known properties of R-D functions and on the particular characteristics of the double compression scheme shown in Fig. 1.

IV. Analysis of the VPF through R-D curves

Here, we provide a more rigorous analysis of the reasons behind the appearance of the VPF, showing that this footprint is directly connected to the technique employed for the selection of the MB coding-mode. With the aim of explaining the origin of the VPF, we first describe the expected behavior of the encoder when a single compression is carried out (Sect. IV-A), then we focus on how the application of a subsequent compression biases the decisions of the encoder so that the VPF shows up in P-frames that were originally encoded as I-frames (Sect. IV-B). This part of the study will validate the use of motion vectors by our novel G-VPF approach to enhance the VPF acquisition process with respect to [10]. Finally, we also characterize the VPF in GOPs that contain both P- and B-frames (Sect. IV-C), thus laying the basis for the adoption of the G-VPF technique also in the case where B-frames are present in the second encoding.

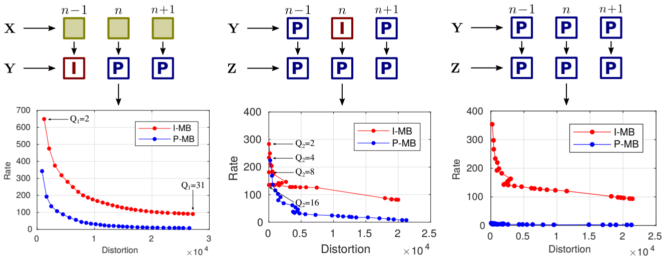
Before starting the analysis, we recall two important properties of the R-D function $R(D)$ for the SSD distortion measure described in Sect. III-A: i) the rate distortion function $R(D)$ is a non-increasing and convex function of D, and ii) there exists a value $D_{\max}$, so that $\forall D \geq D_{\max}$, $R(D) = 0$, where $D_{\max}$ is proportional to the variance of the source. In the following, the conclusions drawn from the R-D characterization of the VPF are always supported by R-D curves obtained empirically from the average calculation of the distortion and rate of the set of video sequences from [30] that are described in Sect. VI-A. With regard to the encoder, we only use the implementation of the MPEG-2 standard available in the FFmpeg library [31], because its code can be conveniently adapted to extract the values of distortion and rate from each MB type during the encoding. Still, the specific characteristics of the other two standards are also discussed, especially for H.264, which introduces different prediction structures.

A. Single compression of P- and B-frames

The expected behavior of a video encoder when the coding-mode decision is guided by a Lagrangian-based strategy is examined here under a single compression: initially for P-frames and then for B-frames. To guide the discussion, we show R-D curves stemming from each of the MB types collected in Table I, except for the S-MBs because the use of S-MBs is decided afterwards, i.e., once the Lagrangian functional achieves a minimum using any of the remaining inter-coding modes: P-MB, F-MB, or B-MB, with the peculiarity of having a zero-motion vector and null residual data. Specifically, the empirical R-D curves are obtained after computing the average distortion and rate for all MBs from an uncompressed frame at time index n and for all possible values of the quantization parameter Q_1, which ranges from 2 to 31, thus obtaining a data-driven version of the R-D function.

1) P-frames: In this case, only I-MBs, P-MBs, and S-MBs are available. In a predictive coding scheme such as the one shown on the left of Fig. 1, the R-D function depends on the variance of the residue $\mathrm{Var}\left(\mathbf{U}_n\right)$ since the distortion of the source can be expressed in terms of the prediction residue $\mathbf{U}_n$, i.e., $D\left(\mathbf{X}_n, \mathbf{X}'_n\right) = \|\mathbf{X}_n - \mathbf{X}'_n\|_2^2 = \|\mathbf{U}_n + \hat{\mathbf{X}}_n - (\mathbf{U}'_n + \hat{\mathbf{X}}_n)\|_2^2 = D\left(\mathbf{U}_n, \mathbf{U}'_n\right)$. For typical video contents with a certain amount of temporal redundancy, the intra-coding modes are the less efficient option given that no temporal prediction is carried out. Indeed, for the particular standards MPEG-2 and MPEG-4, the available intra-coding modes do not apply any intra-prediction process, so $\hat{\mathbf{X}}_n = 0$ and $\mathbf{U}_n = \mathbf{X}_n$, implying that the variance of the residue equals the variance of the source, which is generally large. For the H.264 standard, even if a spatial prediction is performed, the resulting variance of the prediction residue is supposed to be larger than the one obtained through an inter-prediction (or motion-compensation) process, thus in general we can assume that $\mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=I-MB}} \gg \mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=P-MB}}$. As a consequence, the R-D curve for I-MBs is expected to be well above the one obtained for P-MBs, as it is perfectly reflected in the empirical R-D curves shown in Fig. 2(a).

Given the gap between the curves, when applying the Lagrangian optimization procedure, the encoder will regularly opt for the use of P-MBs instead of I-MBs. In fact, the only case in which the above condition on the variance of the residues is not expected to be satisfied is when the motion compensation is ineffective (e.g., at a change of scene or when a certain portion of the scene is suddenly uncovered), thus leading to residual signals with variance of the same magnitude, and the probability of using I-MBs would marginally increase. On the other hand, the amount of S-MBs will depend on the video content characteristics, since their use is only effective in static regions. In general, the smaller the variance of the residue, the larger the probability of using S-MBs given that a low variability in the residual signal means a high temporal redundancy. Also typically, large values of Q_1 favor the use of S-MBs, since the probability of having a null residual increases at higher compression rates.



(a) (b) (c)

Fig. 2. Single compression R-D curves obtained on average when encoding P-frames (a). Double compression R-D curves obtained on average at $n \in \mathsf{I}_1$ (b) and $n \in \mathsf{P}_1$ (c).

2) B-frames: For this type of frames, the encoder can use all the MB types shown in Table I. The aforementioned selection process of I-MBs is equally applicable to this case, so we mainly focus on how the P-MBs, F-MBs, and B-MBs are selected. To simplify the discussion, we introduce the notion of *sub-gop* which denotes the set of adjacent B-frames that are delimited by the reference frames (I or P) on both sides as illustrated in Fig. 3.

Considering again a video sequence with a certain amount of temporal redundancy, at the beginning of the sub-gop we expect the variance of the residue obtained by the P-MBs to be the smallest one, i.e., $\mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=P-MB}} < \min\left(\mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=F-MB}}, \mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=B-MB}}\right)$, due to the proximity in time to the reference frame. This variance will follow an upward trend towards the end of the sub-gop. Conversely, the same reasoning applies to the F-MBs, showing a downward trend from the beginning to the end of the sub-gop. In the intermediate positions of the sub-gop, the smallest variance should be achieved by the B-MBs, while for P- and F-MBs the variance of the residue should have a similar magnitude.

All these effects are well reflected in the R-D curves shown in Fig. 3, where it can be observed how $\mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=P-MB}}$ gradually increases on the way to the end of the sub-gop (conversely, $\mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=F-MB}}$ decreases), and also how $\mathrm{Var}\left(\mathbf{U}_n\right)|_{\text{c=B-MB}}$ keeps constant across the sub-gop. Consequently, regarding the number of each type of MB along the sub-gop, we expect P-MBs to prevail at the beginning of the sub-gop, whereas F-MBs will be dominant at the end. The number of B-MBs will moderately increase in the intermediate positions of the sub-gop.

B. Double compression: VPF on P-frames

From the previous analysis, we know that the selection of the most efficient MB type is driven by the relation between the variance of the prediction residues. Let us focus now on the right block diagram of Fig. 1 where, depending on the type of frame employed during the first compression (either I or P), the encoder will behave differently with respect to the single compression case. This time, to obtain the empirical R-D curves, all the video sequences are first compressed at a medium quality (i.e., $\mathsf{Q}_1 = 16$) and then we compute the average distortion and rate values achieved during the second encoding, covering the values of Q_2 from 2 to 31. In the following, we specialize the analysis for a given frame at time index n, when $n \in \mathsf{I}_1$ and when $n \in \mathsf{P}_1$.

1) Case $n \in \mathsf{I}_1$: With respect to the single compression case, the main differences introduced by the re-encoding of an I-frame with I-MBs are:

- $\mathsf{Var}\left(\mathbf{W}_n\right)|_{\text{c=I-MB}} < \mathsf{Var}\left(\mathbf{U}_n\right)|_{\text{c=I-MB}}$: after the first compression, part of the spatial details of the scene are removed, yielding a smaller variance of $\mathbf{Y}_n$ and $\mathbf{W}_n$.
- $D\left(\mathbf{Y}_n, \mathbf{Y}'_n\right) = D\left(\mathbf{W}_n, \mathbf{W}'_n\right) = 0$ for those values of Q_2 that divide Q_1: the reason is that the distribution of DCT coefficients of $\mathbf{W}_n$ is discrete (due to the quantization applied in the first compression) with a probability mass function denoted by $P_W(w)$. Therefore, the expression for the above distortion can be written as

$$D(\mathbf{W}_n, \mathbf{W}'_n) = \sum_{k=-\infty}^{\infty} \sum_{j=\left\lceil (k+\frac{1}{2})\frac{\Delta_2}{\Delta_1} \right\rceil}^{\left\lfloor (k-\frac{1}{2})\frac{\Delta_2}{\Delta_1} \right\rfloor} (j\Delta_1 - k\Delta_2)^2 P_W(j\Delta_1),$$

where a mid-riser quantizer is assumed without loss of generality. Δ_1 and Δ_2 represent the quantization step sizes used for a given DCT coefficient during the first and second compression, respectively. Due to the previous quantization, the above distortion is zero whenever Δ_1 is an integer multiple of Δ_2, i.e., whenever Q_2 divides Q_1.

Both effects become apparent when comparing the red curves of Figs. 2(a) and 2(b). In addition, the main characteristic that arises from the re-encoding of an I-frame with P-MBs is:

- $\mathsf{Var}\left(\mathbf{W}_n\right)|_{\text{c=P-MB}} \approx \mathsf{Var}\left(\mathbf{U}_n\right)|_{\text{c=P-MB}}$: surprisingly, even if the input frame is the result of a previous compression and hence one would expect a smaller prediction variance (as for I-MBs), in this case, what is actually rising up the variance of the residue is not a sudden lack of temporal redundancy (since the original scene remains unchanged between compressions), but the quantization noise added when encoding the I-frame in the first compression, which is not of the same nature as that arising from the reference frame of the second compression.

The mentioned differences imply that the gap between the R-D curves decreases (see Fig. 2(b)), and accordingly the following variation of coding modes takes place in the second compression when encoding an original I-frame as a P-frame:

a) The number of I-MBs increases: the decrease in $\mathsf{Var}\left(\mathbf{W}_n\right)|_{\text{c=I-MB}}$ together with the zero distortion values in $D(\mathbf{W}_n, \mathbf{W}'_n)$ bias the decision toward I-MBs.

b) The number of S-MBs decreases: even if the real motion of the scene leads to a null motion vector, the different nature of the residual noise prevents the use of S-MBs.

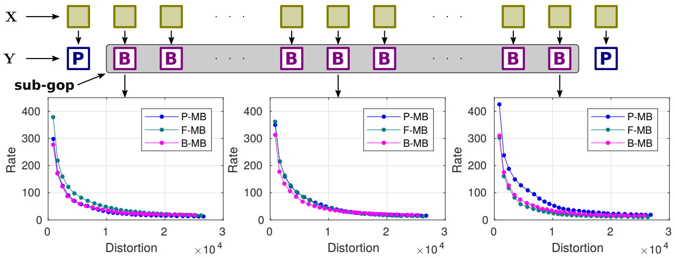


Fig. 3. Single compression R-D curves obtained on average when encoding B-frames.

c) The number of P-MBs with null motion vector increases: this is a consequence of the above point.

In [10], only the first two effects were pointed out, however, the last finding definitely contributes to improving the acquisition of the VPF, as we will later confirm in Sect. VI.

2) Case $n \in \mathsf{P}_1$: The differences that emerge with respect to the single compression case when a previously encoded P-frame is predicted using I-MBs are:

- $\mathsf{Var}\left(\mathbf{W}_n\right)|_{\text{c=I-MB}} < \mathsf{Var}\left(\mathbf{U}_n\right)|_{\text{c=I-MB}}$: the reason is the same as the one pointed out when $n \in \mathsf{I}_1$.
- $D\left(\mathbf{W}_n, \mathbf{W}'_n\right) \neq 0$: it is still possible to find distortion values equal to zero for some MBs when Q_2 divides Q_1, but it is not the prevailing case as in $n \in \mathsf{I}_1$. In fact, only null distortion values can be attained in MBs that are predicted with S-MBs in the first compression and whose content is directly taken from an I-frame.

On the other hand, the differences introduced when a P-frame is encoded with P-MBs are:

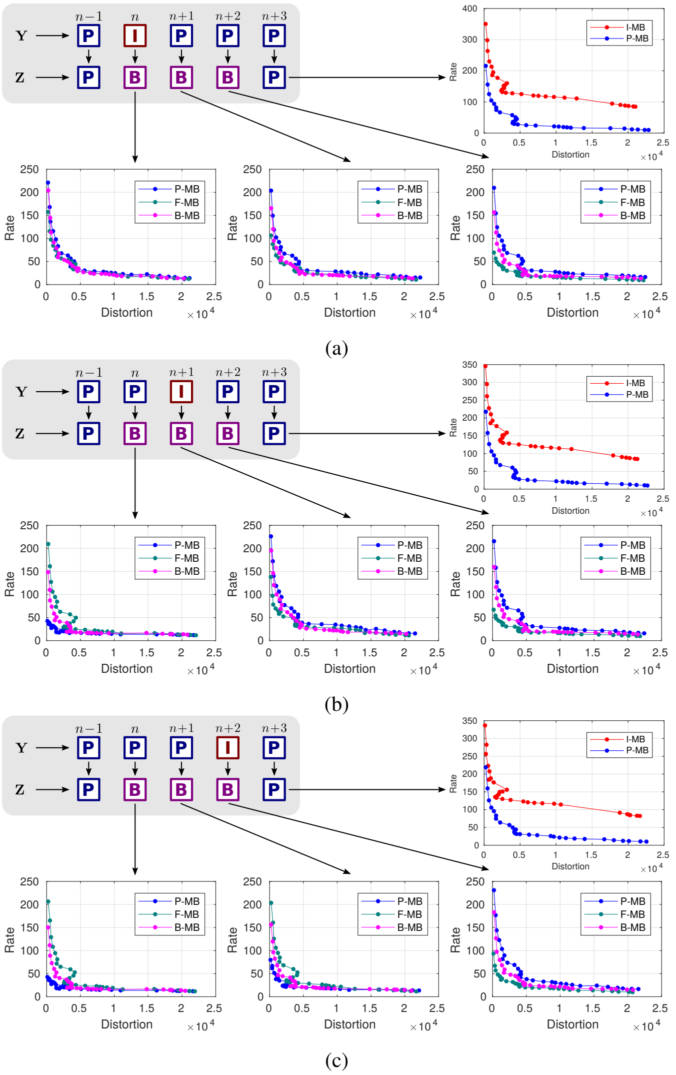
- $\mathsf{Var}\left(\mathbf{W}_n\right)|_{\text{c=P-MB}} \ll \mathsf{Var}\left(\mathbf{U}_n\right)|_{\text{c=P-MB}}$: As opposed to the case $n \in \mathsf{I}_1$, here the variance of the residue is notably smaller. This is consistent with what we expect from the application of a compression in a previous stage. The reason of such reduction is that in this case the quantization noise added during the encoding of the P-frame in the first compression shares the same nature of the one coming from the reference frame.

Fig. 2(c) shows the corresponding empirical R-D curves for this case, where it can be observed that the gap between the R-D curves has grown with respect to the one in Fig. 2(b), so that the behavior of the encoder is nearly aligned with that of a single compression. This "return to normality" ensures that the VPF can be captured, as later described in Sect. V. For further details on how different coding parameters, such as the quantizer deadzones and step sizes used in MPEG-2, affect the strength of the VPF, the reader is referred to the technical report that complements this work in [32].

C. Double compression: VPF on B-frames

We now focus on the two effects that a previously encoded I-frame produces on the second compression when it is re-encoded as part of a sub-gop of B-frames: i) a shifted VPF is induced, and ii) an abrupt change in the number of P-MBs and F-MBs shows up within the sub-gop.

Regarding the first effect, as long as the reference frames that delimit the sub-gop are P-frames and the size of the



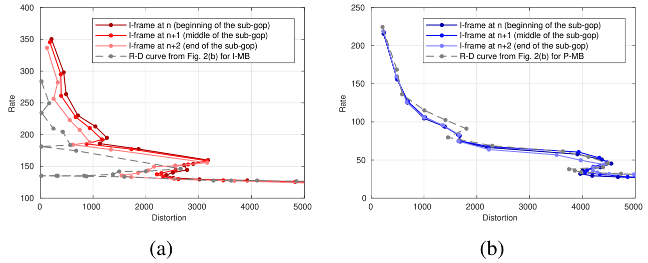
(a)

(b)

(c)

Fig. 4. Double compression R-D curves obtained on average when encoding P- and B-frames from previously encoded I-frames located at the beginning (a), middle (b), and end (c) of the sub-gop.

sub-gop is short (e.g., not larger than 5 frames, which is reasonable in practice), a shifted version of the VPF arises during the encoding of the reference frame that follows the sub-gop. The encoding of this reference frame shares several similarities with the case discussed in Sect. IV-B1 with the subtle difference that here the source of the reference frame is not the I-frame from the first compression, but a subsequent P-frame that results from a concatenation of predictions that starts from the I-frame. Consequently, the similarity between the R-D curves of the I-MB and P-MB coding modes shown across the top-right corners of Figs. 4(a)-(c) with those shown in Fig. 2(b) becomes more evident as the I-frame approaches the end of the sub-gop. This effect can be better observed in Fig. 5, where the R-D curves at the top-right corners of Figs. 4(a)-(c) are plotted together (zoomed in and separated according to the MB type) and compared with the R-D curves from Fig. 2(b) (dashed lines). The set of chained predictions is the reason why the VPF appears at a shifted point with respect to the real position of the I-frame. The proposed approach to correct such shift is further described in Sect. V-B.



(a)

(b)

Fig. 5. Zoomed version of the R-D curves shown at the top-right corners of Figs. 4(a)-(c), splitted according to the MB type: (a) I-MB, (b) P-MB. For comparison, the respective R-D curve from Fig. 2(b) is added (dashed line).

Concerning the second effect, the explanation below analyzes the abrupt changes in the number of P-MBs and F-MBs that occur under three different scenarios depending on the position of the I-frame: at the beginning, in the middle, or at the end of the sub-gop.

1) I-frame located at the beginning of the sub-gop: This scenario is depicted in the top-left corner of Fig. 4(a). As noted in Sect. IV-A2, when encoding the first B-frame of a sub-gop, the P-MB is expected to be the most efficient coding mode due to the proximity to the first reference frame. However, in this particular scenario, with an I-frame located at the beginning of the sub-gop, the temporal redundancy is broken because the I-frame adds a new quantization noise that cannot be predicted by a motion translation, thus increasing the variance of the prediction residue for P-MBs.

In addition, from the above analysis, it is clear that the reference frame that comes after the sub-gop is the result of a series of predictions that start from the I-frame, which converts this future reference frame into a more suitable source for predicting the first frame of the sub-gop, and so the variance of the residue for the F-MB coding mode decreases. This observation is consistent with the R-D curves shown in Fig. 4(a), from which we can deduce that the use of F-MBs will prevail in the second encoding with respect to B-MBs and P-MBs.

This anomalous prevalence of F-MBs at the origin of a sub-gop will be used as a feature to measure the VPF in Sect. V-A.

2) I-frame located in the middle of the sub-gop: Considering the scenario shown in the top-left corner of Fig. 4(b), we observe that the behavior of the encoder prior to the compression of the I-frame follows the path described in Sect. IV-A2. Hence, in the first position of the sub-gop, the use of P-MBs is more frequent. Then, once the I-frame comes into play, we return to the same situation discussed in the previous scenario, with a sudden drop of P-MBs in favor of an abrupt increase on the number of F-MBs. Once more the R-D curves depicted in Fig. 4(b) support these findings.

The main difference with respect to the previous scenario is that the abrupt change in the number of P-MBs and F-MBs reveals the position of the I-frame, which will also be used as a feature to measure the VPF in Sect. V-A.

3) I-frame located at the end of the sub-gop: In this case, no new feature arises, but the R-D curves collected in Fig. 4(c) reinforce the observation of the abrupt change in the coding modes to the end of the sub-gop, where the I-frame is located.

V. Description of the G-VPF technique

In this section we describe a new improved strategy to detect whether a given video sequence has been doubly compressed and, if so, to estimate the size of the GOP employed during the first compression. The proposed G-VPF technique consists of three phases: first, the strength of the VPF is quantified (Sect. V-A), then, if needed, the shifted VPF is corrected (Sect. V-B), and finally a periodicity analysis is carried out to detect double compression and to estimate the GOP size of the first compression (Sect. V-C).

A. Enhanced acquisition of the VPF

For a given video sequence of N frames, we first need to extract the location of each type of frame to build up the sets I, P, and B. Then, for each of these frames, the position of the different MB types must be accounted for to measure the strength of the VPF. Assuming that each frame contains a total of N_{MB} macroblocks, we define the set of MB indices coded as I-MBs at time index n, as follows

$$\mathcal{I}_n \triangleq \left\{ j \in \{0,\dots N_{\text{MB}}-1\} : \text{MB\_type}\left(\mathbf{X}_n^{(j)}\right) = \text{I-MB} \right\},$$

where MB_type $(\cdot)$ is a function that returns the MB type of a given MB, and $\mathbf{X}_n^{(j)}$ denotes the j-th MB from the n-th frame of the given video sequence. Similarly, we define the sets $\mathcal{S}_n$, $\mathcal{P}_n$, and $\mathcal{F}_n$, which respectively contain the indices of the S-MBs, P-MBs, and F-MBs. From the analysis in Sect. IV-C, it is clear that the evolution of the B-MBs does not present significant changes in case of double compression, so their use in the measure of the VPF is disregarded. On the other hand, since the P-MBs with null motion vector can positively contribute to VPF detection, we define the set $\mathcal{M}_n$ that contains the indices of the MBs with null motion vector:

$$\mathcal{M}_n \triangleq \left\{ j \in \{0,\dots N_{\text{MB}}-1\} : \text{MV}\left(\mathbf{X}_n^{(j)}\right) = (0,0) \right\},$$

where MV $(\cdot)$ is a function that returns the motion vector of a given MB. The set containing the indices of P-MBs with null motion vector is then defined as $\tilde{\mathcal{P}}_n \triangleq \mathcal{P}_n \cap \mathcal{M}_n$. Note that this type of MBs is the new feature that results from the R-D analysis in Sect. IV-B1 to improve the VPF acquisition process, which was not previously contemplated in [10].

The quantification of the VPF is related to the number of MBs of each type and their evolution over time. Therefore, we build vectors of N samples that contain the number of MBs of each type. In the particular case of I-MBs, we build the vector $\mathbf{i}$, whose i_n-th component is:

$$i_n \triangleq \begin{cases} |\mathcal{I}_n|, & \text{if } n \in \mathsf{P} \cup \mathsf{B} \\ |\mathcal{I}_{n-1}|, & \text{if } n \in \mathsf{I} \\ 0, & \text{if } n = 0 \end{cases}, \quad \text{for } n = 0,\dots,N-1,$$

where $|\cdot|$ indicates the cardinality of a set. Notice that in the above equation, the first case corresponds to counting the number of I-MBs in P- and B-frames, while the remaining cases are taken into account to avoid strong peaks in the vector $\mathbf{i}$ that are not related to the VPF, but to the presence of an I-frame in the second encoding. For the other MB types, we define the vectors $\mathbf{s}$, $\mathbf{p}$, $\tilde{\mathbf{p}}$, and $\mathbf{f}$, whose components are respectively obtained as $s_n \triangleq |\mathcal{S}_n|$, $p_n \triangleq |\mathcal{P}_n|$, $\tilde{p}_n \triangleq |\tilde{\mathcal{P}}_n|$, and $f_n \triangleq |\mathcal{F}_n|$, for $n = 0,\dots,N-1$.

Once all the information for each MB type has been collected, we have to measure the strength of the VPF. Here, we do not only gather information from P-frames, but also from B-frames, in contrast to [10]. Specifically, to capture the abrupt change in the number of P-MBs and F-MBs discussed in Sect. IV-C, we define the function $h_{\mathbf{p},\mathbf{f}}(n)$ that analyzes, at each index value n, if there is a sudden change in the number of P-MBs and F-MBs. If so, we measure the magnitude of such an effect by taking the product of the corresponding slopes:

$$h_{\mathbf{p},\mathbf{f}}(n) \triangleq \begin{cases} |(p_n - p_{n-1})(f_n - f_{n-1})|, & \text{if } p_{n-1} > f_{n-1} \text{ and } p_n < f_n \\ 0, & \text{otherwise} \end{cases}.$$

On the other hand, to quantify the 3 prediction variations pointed out in Sect. IV-B1, we define a generic function $g_{\mathbf{a}}(\cdot)$ applied over an arbitrary vector $\mathbf{a}$ that computes the amplitude difference between a rising peak and its immediate neighbors

$$g_{\mathbf{a}}(n,k) \triangleq \begin{cases} |a_n - a_{n-k}|, & \text{if } a_n > \max(a_{n-1}, a_{n+1}) \\ 1, & \text{otherwise} \end{cases},$$

where $k \in \{-1, 1\}$ in such a way that the magnitude of the rise level is measured for $k = 1$ and the fall level for $k = -1$. Note that the above function is equal to 1 when there is no peak at time n. This function is applied to process the evolution of the number of I-MBs, S-MBs, and P-MBs with null motion vector, thus resulting in the functions $g_{\mathbf{i}}(n,k)$, $g_{-\mathbf{s}}(n,k)$, and $g_{\tilde{\mathbf{p}}}(n,k)$, where we take the negative version of $\mathbf{s}$ to seize the decreasing peaks. Considering these functions, we define the vector $\mathbf{v}$ that quantifies the VPF, whose components are

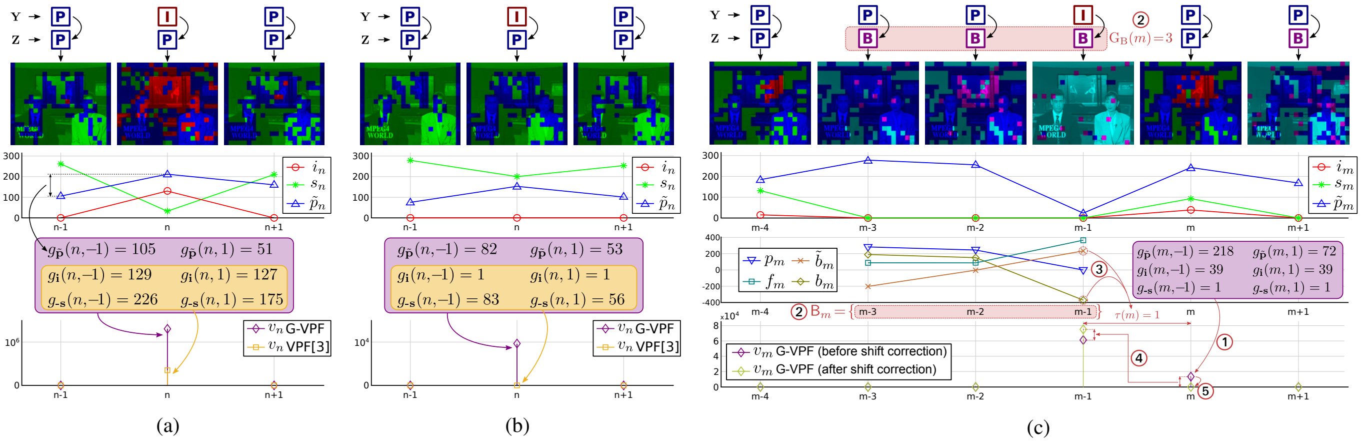
$$v_n \triangleq \begin{cases} h_{\mathbf{p},\mathbf{f}}(n), & \text{if } g_{\mathbf{i}}(n,1) = g_{\text{-}\mathbf{s}}(n,1) = g_{\tilde{\mathbf{p}}}(n,1) = 1 \\ \sum_{k\in\{-1,1\}} g_{\mathbf{i}}(n,k) g_{\text{-}\mathbf{s}}(n,k) g_{\tilde{\mathbf{p}}}(n,k), & \text{otherwise} \end{cases},$$

where the first and second cases represent the contribution of the VPF from B- and P-frames, respectively. In Fig. 6(a) we graphically explain how to compute v_n for P-frames and compare the resulting VPF strength with that from the method in [10]. Fig. 6(b) shows a particular example where the method in [10] fails detecting the VPF, whereas our new approach relying on the P-MBs with null motion vector is able to quantify it. Still, following this new VPF acquisition process, the vector $\mathbf{v}$ can present shifted peaks caused by the presence of B-frames that are corrected as described next.

B. Correction of the shifted VPF induced by B-frames

As detailed in Sect. IV-C, a shifted version of the VPF occurs in the P-frame that follows a sub-gop of B-frames. This can be seen in point ① from the illustrative example shown in Fig. 6(c). Therefore, to correct the position of a shifted peak in vector $\mathbf{v}$, we must separately analyze all the components for which $v_m > 0$ and $m \in \mathsf{P}$. Then, we need to identify the sub-gop preceding the P-frame at each time index m. We do so by calculating the set of indices B_m:

$$\mathsf{B}_m \triangleq \{k \in \mathsf{B} : m - \mathrm{G}_{\mathsf{B}}(m) \le k \le m-1,\ m-1 \in \mathsf{B}\},$$



(a) (b) (c)

Fig. 6. Graphical examples showing how to obtain v_n with the proposed G-VPF technique vs. the method in [10] on P-frames from *news*: (a) $Q_1=16,\ Q_2=2$, (b) $Q_1=16,\ Q_2=24$. In (c), from points ① to ⑤, graphical illustration of the shift correction procedure under B-frames with the proposed G-VPF technique ($Q_1=12, Q_2=6$). Note that in the above frames from *news*, we superimpose the colors defined in Table I to represent each type of MB.

where $G_B(m) \triangleq \min_{l\in \mathsf{I}\cup\mathsf{P}, l<m}(m-1-l)$ represents the size of the sub-gop (see points ② in Fig. 6(c)). Note that if $\mathsf{B}_m = \emptyset$, this means that there is no sub-gop of B-frames at time index m and so the value of v_m does not need to be corrected. Now, since the effect that reveals the actual position of the I-frame in the first encoding (and thus the correction to apply) is the sudden change in the number of P-MBs and F-MBs between adjacent frames, we need to capture that particular event. To do so, we first define the vector $\mathbf{b}$ which gathers the difference between the number of P-MBs and F-MBs at each time index n as $\mathbf{b} \triangleq \mathbf{p} - \mathbf{f}$, with $b_n = p_n - f_n$ and $n = 0, \dots, N-1$. In addition, to identify the exact instant of the abrupt change, we take into account the number of co-located MBs that suddenly change from P-MB to F-MB, which we measure through the cardinality of the set that results from the intersection: $\mathcal{P}_{n-1} \cap \mathcal{F}_n$. Since this exchange of MBs can only be measured when at least two consecutive B-frames are present in the sub-gop, in case of a single B-frame, the difference $f_n - p_n$ is taken instead. Accordingly, we define a vector $\tilde{\mathbf{b}}$ whose components are given by

$$\tilde{b}_n \triangleq \begin{cases} |\mathcal{P}_{n-1} \cap \mathcal{F}_n|, & \text{if } (n-1) \in \mathsf{B} \text{ and } n \in \mathsf{B} \\ f_n - p_n, & \text{if } (n-1) \notin \mathsf{B} \text{ and } n \in \mathsf{B} \\ 0, & \text{otherwise} \end{cases}.$$

With this information, the correction of the shift for the vector $\mathbf{v}$ at time index m, i.e., $\tau(m)$, is given by

$$\tau(m) \triangleq \begin{cases} m - \arg\max_{k\in \mathsf{B}_m} \tilde{b}_k, & \text{if } \min_{k\in\mathsf{B}_m} b_k < 0 \\ 0, & \text{otherwise} \end{cases}.$$

From the analysis in Sect. IV-C, we consider in the above definition that the correction of the shift must only be captured when at some point within the sub-gop the number of F-MBs exceeds that of P-MBs (see point ③ in Fig. 6(c)), otherwise $\tau(m)$ is set to zero. Finally, provided that $\tau(m) \neq 0$, the measure at the actual position of the VPF, i.e., $v_{m-\tau(m)}$, is updated by increasing its value by the contribution of the shifted peak v_m (see point ④ in Fig. 6(c)), then the shifted peak is removed by setting $v_m = 0$ (see point ⑤ in Fig. 6(c)).

C. Periodicity analysis of the measured VPF

After correcting all the shifted peaks in the vector $\mathbf{v}$, it is possible to use it for frame-wise relocated I-frames detection, video double encoding detection, and GOP-size estimation. The former application is simply obtained by classifying all elements of $\mathbf{v}$ exceeding a selected threshold as relocated I-frames. For the other two applications, the periodicity analysis from our work in [10] can be applied, which is summarized hereafter. First, a set of candidate GOPs to match the real one applied during the first compression, i.e., G_1, is determined from the vector $\mathbf{v}$ that quantifies the VPF. Since we look for an artifact that is periodically repeated across the vector, we restrict the search to the set of the greatest common divisors (gcd) between all possible couples of non-zero elements from the vector $\mathbf{v}$. Hence, we define the set $\mathcal{G}$ of candidate GOPs as

$$\mathcal{G} \triangleq \{\gcd(n_1, n_2) : v_{n_1} > 0, v_{n_2} > 0, \forall n_1, n_2 \in \{0, \dots, N-1\}\}.$$

Then, each candidate value $g \in \mathcal{G}$ is associated with a fitness value $\phi : \mathcal{G} \to \mathbb{R}$, that measures how well the choice of g models the periodicity of the signal captured by $\mathbf{v}$. In particular, the fitness function $\phi(g)$ is built upon the combination of the three following measures:

1) The energy of the peaks that are located at multiples of g, i.e., $\phi_1(g) = \sum_{k=0}^{K} v_{k\cdot g}$, where $K \triangleq \left\lfloor \frac{N-1}{g} \right\rfloor$.
2) The absence of peaks that would be expected in multiples of g, quantified as $\phi_2(g) = \beta|\mathcal{A}_g|$, with $\mathcal{A}_g \triangleq \{k \cdot g : v_{k\cdot g} = 0, k = 0, \dots, K\}$, where β penalizes peak missing and is taken as $\beta \triangleq 0.1 \max_n v_n$.
3) The maximum energy of the peaks that are not aligned with the period g: $\phi_3(g) = \max_{i=1,\dots,g-1} \sum_{k=0}^{K-1} v_{k\cdot g+i}$.

The combination of these measures is taken as $\phi(g) = \phi_1(g) - \phi_2(g) - \phi_3(g)$, where it is evident that ϕ_2 and ϕ_3 penalize the candidate g. Once the fitness of every candidate in $\mathcal{G}$ has been evaluated for a given video sequence, the video can be classified as single or double compressed depending on the test statistic ρ, which is defined as $\rho \triangleq \max_{g\in\mathcal{G}} \phi(g)$. After setting a predefined threshold T_ϕ, video sequences with $\rho > T_\phi$ are classified as double compressed, and otherwise

TABLE II
COMPRESSION SETTINGS.

Settings	1st encoding			2nd enconding		
	Q_1 (H.264/MPEG)	R_1	C_1	Q_2 (H.264/MPEG)	R_2	C_2
VBR-VBR	[20/2, 26/5, 32/10, 42/20]	-	-	[10/1, 27/6, 31/9, 38/18]	-	-
VBR-CRF	[20/2, 26/5, 32/10, 42/20]	-	-	-	-	[5, 15, 30]
CRF-CRF	-	-	[10, 18, 26]	-	-	[5, 15, 30]
CBR-CBR	-	[100, 500, 900]	-	-	[100, 500,900]	-
ENC_1/ENC_2 algorithms	[H.264, MPEG-2, MPEG-4]		[H.264]	[H.264, MPEG-2, MPEG-4]		[H.264]
B_1/B_2 frames	[0] and [2, 3, 5] in Sect. VI-E			[0, 2, 3, 5]		
G_1/G_2	[14, 30]			[9, 25, 120]		

as single compressed. Finally, whenever a video sequence is classified as double compressed, the estimate of G_1 is $\hat{G}_1 = \arg\max_{g \in \mathcal{G}} \phi(g)$.

VI. EXPERIMENTAL RESULTS

We evaluated an implementation of our method[7] on a wide variety of configurations, some of which are challenging for the proposed algorithm itself. The experimental dataset is based on well known YUV videos at CIF resolution compressed with three different encoders (MPEG-2, MPEG-4, and H.264) using a vast range of configurations in terms of rate control modes, encoding quality, and GOP size/structure. We compare our method against its previous version (VPF [10]), and against three recently proposed algorithms, whose implementations were kindly provided by the respective authors. The first one is the algorithm described in [14], which performs GOP size estimation even on videos compressed with B-frames, but does not perform double encoding detection. The second is the one developed by Chen *et al.* [11], which performs double encoding detection but does not support B-frames. The third one is the algorithm proposed by He *et al.* in [8] which performs frame-wise "relocated I-frame" detection, a special case of double compression detection.

The experimental validation is organized as follows: Sect. VI-A describes the dataset used for the experiments and the compression settings; Sect. VI-B presents the performance of the proposed method comparing it against [10], [11], and [14] with regard to the estimation of the first GOP size; Sect. VI-C evaluates our technique with regard to double compression detection; Sect. VI-D studies the impact of video content on the G-VPF performance; Sect VI-E analyzes how the introduction of B-frames in the first encoding affects the proposed method; Sect. VI-F evaluates the "relocated I-frame" detection performance of the G-VPF with respect to [8]; finally Sect. VI-G provides some insights on the computational cost of the considered algorithms.

A. Dataset and settings description

Experimental validation was carried out by collecting 31 uncompressed YUV videos with CIF resolution [30], which are either static (low-motion scenes captured by a fixed camera) or dynamic (scenes with motion caused by a focal length change or a moving camera).[8]

Each uncompressed video is encoded[9] by means of the three encoders: MPEG-2, MPEG-4 and H.264, and rate control modes: CBR, VBR and CRF. In CBR, the number of bits per second is fixed throughout the encoding process, since the perceived quality is less important than the file size. VBR dynamically changes the bitrate of the media content depending on the specified quality factor but takes longer than CBR to produce a higher quality video. CRF is a *constant quality* encoding mode which dynamically changes the quantization parameters so as to maintain a prescribed level of perceived quality in the encoded video, thus simultaneously taking into account both quality and bitrate.

Interestingly, we noticed that the results of all the tested methods are strongly influenced by the rate control mode employed in the first or the second encoding, thus we tested 4 double compression scenarios: VBR-VBR, CBR-CBR, VBR-CRF and CRF-CRF. In detail, VBR-VBR and CBR-CBR are the most common scenarios tested in the state of the art, however we deem that VBR-CRF is the most realistic case. Indeed, the first encoding is usually performed by the acquisition device, which has limited processing power and has to operate in real-time, thus making the computationally intensive CRF mode hard to use. On the other hand, the second compression can take advantage of more processing power and time, being usually performed off-line, so using the more advanced CRF mode will produce a video with the best balance between quality and file size.

Table II summarizes the first/second compression parameters used for our validation in terms of: bitrate value (R_1/R_2), quantization value depending on the encoding algorithm (Q_1/Q_2), CRF value (C_1/C_2)[10], group of pictures size (G_1/G_2) and number of consecutive B-frames (B_1/B_2). The combination of parameters in Table II produced an overall dataset of 10044 single compressed videos and 200880 double compressed videos. We point out that the values reported in Table II with respect to the VBR scenario consider different quantization values depending on the encoding algorithm, namely H.264 and MPEG-2/MPEG-4 (e.g., $Q_1 = 20/2$ indicates H.264 with $Q_1 = 20$ and MPEG-2/MPEG-4 with $Q_1 = 2$). Indeed, the range of quantization values used by H.264 is wider ($[0, 51]$) than the one used by MPEG-2/MPEG-4 ($[0, 31]$); to allow for a meaningful comparison, we chose four quantization parameters in H.264 ranging from high to poor visual quality and we identified the corresponding quanti-

[7] Available online: https://github.com/IAPP-Group/GVPF

[8] Static: *akiyo, bridge-close, bridge-far, bowing, container, deadline, galleon, hall, mother-daughter, news, pamphlet, paris, sign-irene, silent, students, vtc1nw, washdc*. Dynamic: *city, coastguard, crew, flower, football, foreman, harbour, highway, husky, intros, mobile, soccer, tempete, waterfall.*

[9] We used FFmpeg v3.0.1 software [31] to encode with MPEG-2 and MPEG-4 codecs. x264 v0.148.x software [33] to perform H.264 compression.

[10] The CRF mode is available only with the H.264 codec.

TABLE III
G_1 MATCH ACCURACY WITH $B_1 = B_2 = 0$ (MARGINALIZED OVER THE CODEC-PAIR (ENC_1, ENC_2)).

G_1 **match accuracy**		VBR-VBR			VBR-CRF			CRF-CRF			CBR-CBR		
ENC_1	ENC_2	G-VPF	VPF [10]	[14]	G-VPF	VPF [10]	[14]	G-VPF	VPF [10]	[14]	G-VPF	VPF [10]	[11]
H.264	H.264	**81.7**	80.5	18.5	**93.2**	92.4	30.1	**75.4**	71.6	20.1	91.3	**91.8**	71.6
H.264	MPEG-4	**79.7**	56.8	9.5	-	-	-	-	-	-	**64.5**	54.1	-
H.264	MPEG-2	**80.2**	53.8	8.7	-	-	-	-	-	-	**64.2**	47.6	-
MPEG-4	H.264	**82.2**	81.4	29.0	**90.8**	90.4	35.3	-	-	-	95.5	**96.5**	63.7
MPEG-4	MPEG-4	**82.9**	66.7	25.4	-	-	-	-	-	-	**70.0**	65.2	-
MPEG-4	MPEG-2	**85.6**	63.6	26.1	-	-	-	-	-	-	**72.8**	55.7	-
MPEG-2	H.264	**73.2**	71.7	35.2	**88.2**	84.1	38.3	-	-	-	95.8	**97.3**	72.3
MPEG-2	MPEG-4	**75.6**	60.4	31.2	-	-	-	-	-	-	**68.9**	66.8	-
MPEG-2	MPEG-2	**79.8**	62.6	29.7	-	-	-	-	-	-	**71.9**	65.0	-

TABLE IV
G_1 MATCH ACCURACY IN H.264 CBR-CBR SCENARIO WITH $B_1 = B_2 = 0$.

G_1 **match accuracy**		CBR-CBR		
R_1	R_2	G-VPF	VPF [10]	[11]
100	100	99.1	**99.6**	53.6
100	500	98.9	**100.0**	74.2
100	900	97.5	**100.0**	70.3
500	100	**90.9**	88.9	45.9
500	500	97.3	**99.8**	89.1
500	900	96.4	**100.0**	93.0
900	100	**78.3**	73.8	31.7
900	500	94.3	**96.1**	74.0
900	900	95.2	**98.6**	91.0

zation values for codecs MPEG-2/MPEG-4 by comparing the PSNR of YUV videos after compression with both codecs. Considering that B-frames are rarely employed by capturing devices (e.g., in videos of the recent VISION dataset [34] only one smartphone out of 35 uses them), we chose to limit the main experimental validation to consider only I/P frames in the first encoding, whereas the second encoding uses I/P and $2, 3$, or 5 consecutive B-frames. Nevertheless, we dedicate Sect. VI-E to analyze the sensitivity of the proposed method to the presence of B-frames in the first encoding. Since the periodicity analysis of the measured VPF only works when the first compression GOP size is static, we decided to fix this parameter in both compression stages. While this could seem a strong limitation, we observe that many modern smartphones[11] acquire videos by means of a static GOP (see again the VISION dataset [34]).

B. GOP size estimation

The evaluation regarding the first GOP size estimation is presented in terms of percentage of exact match, here on referred as "G_1 match accuracy". Results are presented separately, first for the case without B-frames and then with B-frames in the second encoding. In order to perform a fair comparison with the algorithms presented in [14] and [11], we tested the former under fixed quantization scenarios only, and the latter under a fixed bitrate scenario only, according to each method capabilities. In the following tables we show the G_1 match accuracy marginalized over one coding parameter at a time, averaging with respect to all the other parameters and all the tested videos.

1) Results without B-frames: Table III reports the performance of the proposed method, without B-frames, compared with those achieved by the algorithms by Bestagini *et al.* [14], by Chen *et al.* [11] and by the original VPF [10]. We describe the G_1 match accuracy for each encoding mode marginalized over the codec-pair.

TABLE V
G_1 MATCH ACCURACY IN VBR-VBR SCENARIO WITH $B_1 = B_2 = 0$.

G_1 **match accuracy**		VBR-VBR		
Q_1	Q_2	G-VPF	VPF [10]	[14]
20/2	10/1	**83.9**	61.9	39.4
20/2	27/6	**54.8**	29.0	2.6
20/2	31/9	**43.1**	15.5	0.7
20/2	38/18	**21.0**	7.2	0.3
26/5	10/1	**92.1**	88.0	55.1
26/5	27/6	**87.6**	74.7	12.7
26/5	31/9	**78.7**	50.3	2.8
26/5	38/18	**53.4**	21.5	0.6
32/10	10/1	**96.4**	95.3	58.0
32/10	27/6	**96.5**	94.4	32.9
32/10	31/9	**94.7**	86.9	15.8
32/10	38/18	**83.3**	46.8	1.6
42/20	10/1	**99.3**	99.2	55.9
42/20	27/6	98.5	**99.9**	48.2
42/20	31/9	99.3	**99.7**	38.9
42/20	38/18	**99.0**	88.9	13.7

We see that the proposed algorithm outperforms almost always all the other algorithms. Noticeably, the new method is also the only one that can be used under all considered codecs and rate control modes. Indeed the best performance, 95.8% of first GOP estimation accuracy is obtained with the pair $(\text{MPEG-2}, \text{ H.264})$ in the CBR-CBR scenario, whereas only 64.2% of accuracy is obtained with the opposite encoding pair, namely $(\text{H.264}, \text{ MPEG-2})$. This is likely due to the stronger compression artifacts introduced by MPEG-2 in the second compression stage, resulting in a significant loss of information. The limited performance obtained with [14] is probably due to the fact that our experimental setting focuses on exploring several combinations of coding modes and GOP sizes rather than different codec implementations.

Table IV shows how the G-VPF accuracy varies for different bitrate pairs, focusing on the H.264 codec (CBR-CBR setting). In this specific configuration, the proposed algorithm outperforms Chen *et al.* but it is slightly worse than the original VPF (5% of accuracy loss in the worst case). However, it is worth observing that G-VPF improves the VPF accuracy when the second encoding bitrate is lower that the first one ($+5\%$ in the best case). Table V reports the percentage of first GOP accuracy in a VBR-VBR scenario comparing the proposed method to the original VPF and Bestagini *et al.*'s method. The quantization parameters in Q_1 and Q_2 describe the corresponding values used for the H.264 encoder and the

[11] Examples include: *Samsung Galaxy S5, Huawei P9/P9Lite, Xiaomi Redmi Note3, Apple iPhone 6s, etc.*

TABLE VI
G_1 MATCH ACCURACY WITH $B_1=0$, $B_2 \in \{0,2,3,5\}$ (MARGINALIZED OVER THE NUMBER OF B-FRAMES IN THE SECOND ENCODING B_2).

G_1 **match accuracy**	VBR-VBR			VBR-CRF			CRF-CRF			CBR-CBR		
B_2	G-VPF	VPF [10]	[14]	G-VPF	VPF [10]	[14]	G-VPF	VPF [10]	[14]	G-VPF	VPF [10]	[11]
0	**80.1**	66.4	23.7	**90.7**	89.0	34.5	**75.4**	71.6	20.1	**77.2**	71.1	69.2
2	**38.0**	-	20.3	**66.5**	-	28.0	**47.2**	-	15.9	45.8	-	-
3	**42.0**	-	19.7	**65.5**	-	24.6	**48.1**	-	12.8	51.0	-	-
5	**36.1**	-	20.3	**44.9**	-	25.9	**31.6**	-	12.8	43.1	-	-

MPEG-2/MPEG-4 encoders (e.g., $Q_1 = 20/2$ indicates H.264 with $Q_1 = 20$ and MPEG-2/MPEG-4 with $Q_1 = 2$). It is interesting to note that the accuracy tends to decrease as the second quantization parameter increases, since in this situation a great part of the traces left by the first compression are erased. The G-VPF clearly outperforms the state-of-the-art algorithms, with the exception of $(Q_1 = 42, Q_2 \in \{10, 27, 31\})$ where the original VPF performs equally or slightly better. The hardest scenario for all the algorithms presented in Table V is $(Q_1 = 20,\ Q_2 \in \{27, 31, 38\})$: G-VPF performs best, but it obtains an accuracy of only 21.0% in the worst case.

2) Results with B-frames: Table VI represents the accuracy of the first GOP estimation of each algorithm when B-frames are allowed in the second compression. To facilitate the performance comparison with the previous case (Sect. VI-B1), Table VI also shows the aggregated results in absence of B-frames: in that case the G-VPF is compared against the original VPF, Bestagini *et al.* and Chen *et al.*, whereas when B-frames are present the G-VPF algorithm can be compared only against Bestagini *et al.*'s method. Our results show that the best performance is obtained when B-frames are not used, indeed G-VPF accuracy is greater than 75% in all encoding scenarios. Clearly, the decreased accuracy is linked to the increment of the number of consecutive B-frames. Interestingly, a 66.5% accuracy is obtained by G-VPF in a VBR-CRF scenario with two consecutive B-frames, that is $+38.5\%$ compared to Bestagini *et al.*

Table VII presents the first GOP estimation accuracy aggregating the cases where 2, 3 and 5 B-frames are used, comparing the results of G-VPF with Bestagini *et al.* whenever possible; the results are split based on the encoding algorithm used in the first and second compressions. The best performance is obtained in the CBR-CBR case when the first compression algorithm is MPEG-2 and the second is H.264. Table VII confirms once more that the VPF effect is more evident when the first encoding is stronger than the second one (MPEG-2/MPEG-4 encoding algorithms are undoubtedly rougher than H.264). Overall, the task is challenging and calls for further research, nevertheless the G-VPF is one of the first video forensic algorithms dealing with B-frames and outperforms existing schemes.

Table VIII analyzes G-VPF and Bestagini *et al.* performance in the VBR-VBR and VBR-CRF scenarios, focusing on the first and second encoding quantizer scale. Interestingly, both methods obtain the highest accuracy in the VBR-CRF scenario, with the proposed algorithm outperforming Bestagini *et al.*'s approach in all tested configurations. Once more we observe that performance drops significantly when the second compression is more aggressive than the former, for the same reasons provided in Sect. VI-B1.

TABLE VII
G_1 MATCH ACCURACY WITH $B_1=0$, $B_2 \in \{2,3,5\}$ (MARGINALIZED OVER THE CODEC-PAIR (ENC_1, ENC_2)).

G_1 **match accuracy**		VBR-VBR		VBR-CRF		CRF-CRF		CBR-CBR
ENC_1	ENC_2	G-VPF	[14]	G-VPF	[14]	G-VPF	[14]	G-VPF
H.264	H.264	**47.6**	10.3	**63.7**	20.8	**42.3**	13.8	62.2
H.264	MPEG-4	**38.8**	7.5	-	-	-	-	33.5
H.264	MPEG-2	**35.0**	6.2	-	-	-	-	33.1
MPEG-4	H.264	**43.2**	25.2	**58.2**	27.4	-	-	63.3
MPEG-4	MPEG-4	**40.5**	23.1	-	-	-	-	41.9
MPEG-4	MPEG-2	**33.5**	22.3	-	-	-	-	39.8
MPEG-2	H.264	**41.9**	32.8	**54.9**	30.4	-	-	65.4
MPEG-2	MPEG-4	**36.6**	28.1	-	-	-	-	42.4
MPEG-2	MPEG-2	**31.5**	25.6	-	-	-	-	37.9

TABLE VIII
G_1 MATCH ACCURACY IN VBR-VBR AND VBR-CRF SCENARIOS WITH $B_1=0$, $B_2 \in \{2,3,5\}$.

G_1 **match accuracy**	VBR-VBR			VBR-CRF		
Q_1	Q_2	G-VPF	[14]	C_2	G-VPF	[14]
20/2	10/1	**45.2**	32.9	5	**52.6**	46.3
20/2	27/6	**4.4**	2.3	15	**55.7**	6.8
20/2	31/9	**1.3**	0.6	30	**3.5**	0.2
20/2	38/18	**0.7**	0.4	-	-	-
26/5	10/1	**55.7**	52.0	5	**62.3**	51.3
26/5	27/6	**43.2**	8.3	15	**67.5**	24.2
26/5	31/9	**12.1**	2.1	30	**30.0**	0.5
26/5	38/18	**1.8**	0.6	-	-	-
32/10	10/1	**64.3**	54.7	5	**66.1**	53.1
32/10	27/6	**65.0**	24.8	15	**72.7**	33.0
32/10	31/9	**56.8**	9.8	30	**71.9**	3.2
32/10	38/18	**10.6**	1.7	-	-	-
42/20	10/1	**68.4**	53.0	5	**70.3**	52.6
42/20	27/6	**65.7**	40.0	15	**78.1**	33.9
42/20	31/9	**66.3**	30.5	30	**76.9**	9.3
42/20	38/18	**58.1**	8.3	-	-	-

C. Double compression detection

Double compression detection performance is evaluated using the G-VPF as a detector: we use a set of training videos to build the ROC curve, compute its AUC and then determine a threshold targeting a False Positive Rate (FPR) of 3%. Finally, we use such threshold to classify videos in the test set. The detector performance is evaluated by means of: True Positive Rate (TPR), True Negative Rate (TNR) and balanced accuracy $B{-}Acc$ = (TPR + TNR) / 2.

Single compressed videos (the negative class) are generated by using the same encoding parameters employed for the second compression stage of double compressed videos. Therefore, for each scenario, the single compression parameters used are those described in Table II in the *second encoding* column, whereas the double compressed videos use the same settings of Table II. The detector performance is evaluated by averaging the results of 5 random train-test folds. Each train-test fold is made of 14 training videos and 14 testing videos. For both the train and test set we select randomly 7 motion and 7 static videos from the 31 YUV sequences. In Table IX we report the double compression detection performance without B-frames by comparing the proposed method against the VPF [10] and

TABLE IX
DOUBLE COMPRESSION DETECTION WITH $\mathrm{B}_1 = \mathrm{B}_2 = 0$ (BY MEANS OF A TARGET FPR $= 3\%$ DURING TRAINING).

Double Compression Detection		G-VPF				VPF [10]				[11]			
Mode 1	Mode 2	Train AUC	B−Acc	TPR	TNR	Train AUC	B−Acc	TPR	TNR	Train AUC	B−Acc	TPR	TNR
CBR	CBR	0.89	**0.85**	0.73	0.96	0.85	0.81	0.65	0.96	0.84	0.76	0.56	0.96
VBR	VBR	0.90	**0.86**	0.75	0.97	0.85	0.79	0.62	0.96	-	-	-	-
VBR	CRF	0.95	**0.91**	0.85	0.97	0.95	0.90	0.84	0.96	-	-	-	-
CRF	CRF	0.88	**0.82**	0.68	0.97	0.88	0.80	0.63	0.96	-	-	-	-

TABLE X
DOUBLE COMPRESSION DETECTION FOR G-VPF WITH $\mathrm{B}_1 = 0$,
$\mathrm{B}_2 \in \{2, 3, 5\}$ (BY MEANS OF A TARGET FPR $= 3\%$ DURING TRAINING).

Mode 1	**Mode 2**	Train AUC	B−Acc	TPR	TNR
CBR	CBR	0.73	0.69	0.41	0.97
VBR	VBR	0.71	0.67	0.37	0.97
VBR	CRF	0.81	0.76	0.55	0.96
CRF	CRF	0.74	0.68	0.39	0.96

Chen *et al.*'s [11] algorithms; and finally in Table X we present the G-VPF double compression detection results in case of B-frames. Note that we cannot compare with Bestagini *et al.*'s approach, since their method estimates first compression parameters *assuming* that the video is double compressed.

The best performance is obtained in a VBR-CRF scenario without B-frames, where G-VPF yields $\mathrm{B-Acc} = 0.91$ for $\mathrm{TNR} = 0.97$. However, encouraging results are also obtained in presence of B-frames, indeed G-VPF B−Acc ranges from 0.67, in the worst case, to 0.76 in VBR-CRF mode that remains the best scenario even for videos with B-frames.

D. Sensitivity to video content

The proposed method analyzes MB types to perform GOP size estimation and double compression detection. Clearly, the MB type selection is influenced by the compression parameters but also by the video content. We have therefore studied the G-VPF performance separately for videos with static and dynamic contents (as listed at the beginning of Sect. VI-A).

Tables XI and XII report, respectively, the performance of G_1 match accuracy and double compression detection marginalized over the video content. Table XI shows that overall G-VPF performs the best with static video sequences, reaching an exact estimation 95.5% of the times without B-frames, and 73.6% with B-frames in the best scenario. Nevertheless, even in the case of dynamic videos, G-VPF still reaches 84.9% without B-frames in the VBR-CRF scenario.

Table XII reports the average G-VPF performance for double compression detection using 5 random train-test folds. Similarly to Sect. VI-C, for each split and each motion category, 7 video sequences are randomly chosen for training and 7 are randomly chosen for testing, in such a way that every video appears at least once in both training and testing. As in the previous case, G-VPF performs best for static video contents, indeed the accuracy without B-frames in the VBR-CRF scenario decreases by 9% from static to dynamic contents, whereas in case of B-frames in the second compression, the drop is more significant, with a 17% accuracy loss.

E. Sensitivity to B-frames in the first encoding

To complete the experimental validation of our G-VPF method we also analyze its sensitivity to the presence of B-frames during the first compression stage. For the sake of conciseness, we limit the analysis to the best and worst performing scenarios for the G-VPF, namely VBR-CRF and CRF-CRF, using the settings described in Table II. To highlight how the presence of B-frames affects the G-VPF performance, we compare the cases where no B-frames are used in the first encoding (i.e., $\mathrm{B}_1 = 0$), with those where sub-gops of either 2, 3, or 5 B-frames are employed (i.e., $\mathrm{B}_1 \in \{2, 3, 5\}$).

TABLE XI
G_1 MATCH ACCURACY FOR G-VPF WITH $\mathrm{B}_1 = 0$ AND COMPARING $\mathrm{B}_2 = 0$ WITH $\mathrm{B}_2 \in \{2, 3, 5\}$ (MARGINALIZED OVER THE VIDEO CONTENT).

Video Content	VBR-VBR	VBR-CRF	CRF-CRF	CBR-CBR
$\mathrm{B}_2 = 0$ (without B-frames)				
Dynamic	68.9	84.9	63.4	78.0
Static	90.1	95.5	85.2	77.8
$\mathrm{B}_2 \in \{2, 3, 5\}$ (with B-frames)				
Dynamic	30.5	41.7	31.5	38.9
Static	45.5	73.6	51.9	53.4

Table XIII presents the G_1 match accuracy of the G-VPF method as a function of the number of B-frames used in the first and second compression phases. In general, these results reveal that the use of B-frames in the first compression reduces the accuracy of our method, achieving a loss in performance of 1.1% in the best case and almost a 14% in the worst. The reason behind this loss in performance is the rise of noisy contributions in the G-VPF signal at transitions between sub-gops of B-frames, which impair the periodicity analysis described in Sect. V-C. Although a more robust analysis could be conceived to filter out noisy components, further research is required to understand the rise of such artifacts in the VPF acquisition process when $\mathrm{B}_1 \neq 0$.

Table XIV reports the average double compression detection capabilities of the proposed method over 5 train-test splits. Interestingly, the decrease in detectability with respect to $\mathrm{B}_1 = 0$ is at worst 3% in terms of B−Acc, thus indicating that the referred noisy contributions do not strongly affect the detection of double compression traces. Still, as mentioned above, a better characterization of these emerging artifacts is further needed to minimize their effect on the acquisition of the VPF, which will be the subject of future work.

F. Relocated I-frames detection

Using the G-VPF for frame-wise relocated I-frames detection is as simple as thresholding the G-VPF signal instead of conducting the periodicity analysis. Since the method proposed by He *et al.* [8] is designed to work on H.264 videos with CBR coding mode and without the presence of B-frames, we used a subset of our experimental settings (Table II) that respects these constraints for a fair comparison. Similarly to what we did for video-wise double encoding detection, and

TABLE XII
Double compression detection for G-VPF with $\mathrm{B}_1 = 0$ and comparing $\mathrm{B}_2 = 0$ with $\mathrm{B}_2 \in \{2, 3, 5\}$ (by means of a target FPR = 3% during training and marginalized over the video content).

Video Content	VBR-VBR			VBR-CRF			CRF-CRF			CBR-CBR		
	$\mathrm{B}_2 = 0$ (without B-frames)											
	B−Acc	TPR	TNR	B−Acc	TPR	TNR	B−Acc	TPR	TNR	B−Acc	TPR	TNR
Dynamic	0.80	0.63	0.97	0.86	0.73	0.98	0.75	0.51	0.99	0.84	0.70	0.98
Static	0.93	0.88	0.97	0.95	0.94	0.95	0.89	0.81	0.96	0.86	0.74	0.98
	$\mathrm{B}_2 \in \{2, 3, 5\}$ (with B-frames)											
	B−Acc	TPR	TNR	B−Acc	TPR	TNR	B−Acc	TPR	TNR	B−Acc	TPR	TNR
Dynamic	0.63	0.28	0.97	0.67	0.36	0.98	0.63	0.27	0.98	0.66	0.36	0.96
Static	0.71	0.44	0.98	0.84	0.71	0.97	0.73	0.50	0.97	0.73	0.49	0.97

TABLE XIII
G_1 match accuracy for G-VPF comparing $\mathrm{B}_1 = 0$ with $\mathrm{B}_1 \in \{2, 3, 5\}$ (marginalized over B_2).

B-frames	VBR-CRF		CRF-CRF	
B_1 / B_2	0	[2, 3, 5]	0	[2, 3, 5]
0	90.7	78.4	75.4	65.7
2	66.5	57.3	47.2	45.3
3	65.5	51.6	48.1	36.9
5	44.9	40.0	31.6	30.5

TABLE XIV
Double compression detection for G-VPF comparing $\mathrm{B}_1 = 0$ with $\mathrm{B}_1 \in \{2, 3, 5\}$ (by means of a target FPR = 3% during training).

Mode 1	**Mode 2**	$\mathrm{B}_1 = 0$, $\mathrm{B}_2 \in \{2, 3, 5\}$			$\mathrm{B}_1, \mathrm{B}_2 \in \{2, 3, 5\}$		
		B−Acc	TPR	TNR	B−Acc	TPR	TNR
VBR	CRF	0.76	0.55	0.96	0.73	0.49	0.96
CRF	CRF	0.68	0.39	0.96	0.66	0.35	0.96

TABLE XV
Relocated I-frames detection performance obtained by the G-VPF and the method from [8] in our experimental setting.

Measure	G-VPF			[8]
	T_1 : [8]'s TNR	T_2 : [8]'s TPR	T_3 : Perfcurve	
B−Acc	0.88	0.84	0.91	0.75
TPR	0.94	0.68	0.89	0.68
TNR	0.83	0.99	0.93	0.83

consistently with [8], we use a train-test approach to this task, with the same dataset construction strategy explained in Sect. VI-C. For training, we build the positive class by collecting the values of the G-VPF vector $\mathbf{v}$ (as defined in Sect. V-A), at those positions of the re-located I-frames stemming from double compressed train videos. On the other hand, the negative class is built by gathering an equivalent number of values of $\mathbf{v}$ randomly picked at the positions of double compressed P-frames and single compressed P-frames from double and single compressed train videos, respectively. For testing, both positive and negative classes are built in the same way, but using test videos. The whole train-test procedure is repeated five times and results are averaged for better statistical significance.

We first trained He *et al.*'s own implementation of the deep network described in [8], and used the resulting network to classify all frames in the test set. Performance is reported in Table XV, last column. We then built the ROC for our detector and selected three different thresholds: T_1, yielding the same TNR achieved by [8]; T_2, yielding the same TPR achieved by [8]; and T_3, yielding our detector's best operating point as computed by Matlab's `perfcurve` function; for each case we then computed the performance of our threshold-based classifier on the test set, as reported in Table XV. We observe that the G-VPF detector outperforms He *et al.*'s method at every selected operating point: when matching the same TNR obtained by the deep learning classifier, the G-VPF achieves +26% on the TPR; when matching the same TPR obtained by the classifier, the G-VPF achieves +16% on the TNR. When setting both classifiers to work at their best best operating point, the G-VPF achieves a B−Acc gain of 16%.

We acknowledge that the performance achieved by the method proposed by He *et al.* in our experimental setting is significantly different than the one reported in the original paper, where the balanced accuracy reaches 97%. We thus swept through the hyper-parameters regulating the training phase (learning rate, number of epochs), without any advantage. To eliminate any remaining doubts on the comparative results, we first tested He *et al.*'s own implementation *on the same* experimental setting described in [8] (achieving very close results to those reported in their paper) and then proceeded to evaluate the G-VPF on the same scenario. Table XVI collects the performance obtained by our method next to the results reported by He *et al.* in their original work, showing that our detector even slightly improves the already excellent performance of their classifier. This comparative study reinforces the importance of designing forensic solutions rooted on theoretical grounds, as our G-VPF approach is, because they are not only simpler, faster and less data-depedent, but also generalize much better than learning-based methods which may often fail for reasons that are hard to fathom.

G. Computational cost summary

We compared the computational cost of the GOP size estimation task for each tested algorithm. We used a workstation with an `Intel(R) Core(TM) i7-3770 CPU @3.40GHz` with `Matlab 2018b`, and we averaged the execution time obtained for 10 videos, each 10 seconds long. The obtained values are: 2.7s for the G-VPF, 1.8s for the VPF [10], 5.4s for Chen *et al.*'s algorithm, and 217.2s for Bestagini *et al.*'s algorithm. We can conclude that the VPF is the fastest solution, closely followed by the G-VPF and by Chen *et al.*'s approach. As expected, Bestagini *et al.*'s method takes much longer to run.

As to relocated I-frames detection, we had to use a workstation equipped with an appropriate GPU, namely: `Intel(R) Core(TM) i9-7940X CPU @ 3.10GHz` with an `NVIDIA Quadro P6000` with 24GB of RAM.

TABLE XVI
RELOCATED I-FRAMES DETECTION PERFORMANCE OBTAINED BY THE G-VPF AND [8] UNDER THE EXPERIMENTAL SETTING FROM [8].

Measure	G-VPF	Performance reported in [8]
B−Acc	0.98	0.97
TPR	0.99	0.96
TNR	0.97	0.97

The average G-VPF computation time for a single video is 2.1 seconds whereas He *et al.*'s approach takes 2.5 seconds.

VII. CONCLUSIONS

This paper brings a contribution to the field of digital video forensics, presenting a tool for double encoding detection and previous GOP size estimation. Compared to the existing literature and to the original work introducing the VPF [10], the main contributions of this paper are: a deeper theoretical investigation of the reasons behind the VPF, reasons that are implicitly shared by several other works covering the same topic; the introduction of the G-VPF technique, which builds on an enhanced VPF acquisition process and allows the analysis of videos containing B-frames (which are neglected by most works in the state of the art); and a wide experimental validation covering different encoders, quality factors, and GOP sizes. Noticeably, the proposed method can be implemented efficiently, without even the need of fully decoding the video, allowing tractable computation times, which is also an important contribution compared to the state of the art.

New research questions will be addressed in the future, starting from the extension of our VPF-related theory to the recent HEVC standard, which presumably will become the most commonly adopted video coding standard in the coming years. Furthermore, we will also investigate the noisy artifacts that emerge in the G-VPF signal when B-frames are employed during the first video compression, since the use of more complex coding profiles in modern acquisition devices (such as smartphones, camcorders, etc.) will probably increase in the near future.

REFERENCES

[1] Scientific Working Group on Imaging Technologies, "Best practices for maintaining the integrity of digital images and digital video, v. 1.1," January 2012. [Online]. Available: www.swgit.org

[2] ——, "Best practices for image authentication, v. 1.1," January 2013. [Online]. Available: www.swgit.org

[3] M. Pollitt, E. Casey, D.-O. Jaquet-Chiffelle, and P. Gladyshev, "A framework for harmonizing forensic science practices and digital/multimedia evidence," 2018. [Online]. Available: https://www.nist.gov/sites/default/files/documents/2018/01/10/osac_ts_0002.pdf

[4] M. Jerian, "Can you use filtered images in court?" *Lawyer Monthly*, vol. 93, no. 18, pp. 36–37, 2018.

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[6] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *Proc. of the 8th Workshop on Multimedia and Security (MM&Sec)*. NY, USA: ACM, 2006, pp. 37–47.

[7] M. C. Stamm, W. S. Lin, and K. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 4, pp. 1315–1329, August 2012.

[8] P. He, X. Jiang, T. Sun, S. Wang, B. Li, and Y. Dong, "Frame-wise detection of relocated i-frames in double compressed H. 264 videos based on convolutional neural network," *Journal of Visual Communication and Image Representation*, vol. 48, pp. 149–158, 2017.

[9] P. He, X. Jiang, T. Sun, and S. Wang, "Double compression detection based on local motion vector field analysis in static-background videos," *Journal of Visual Communication and Image Representation*, vol. 35, pp. 55–66, 2016.

[10] D. Vázquez-Padín, M. Fontani, T. Bianchi, P. Comesaña, A. Piva, and M. Barni, "Detection of video double encoding with GOP size estimation," in *Proc. of the 4th IEEE Intl. Workshop on Information Forensics and Security (WIFS)*, December 2012, pp. 151–156.

[11] S. Chen, T. Sun, X. Jiang, P. He, S. Wang, and Y. Q. Shi, "Detecting double H.264 compression based on analyzing prediction residual distribution," in *Proc. of the Intl. Workshop on Digital Watermarking (IWDW)*. Springer, 2016, pp. 61–74.

[12] P. He, X. Jiang, T. Sun, and S. Wang, "Detection of double compression in MPEG-4 videos based on block artifact measurement," *Neurocomputing*, vol. 228, pp. 84–96, 2017.

[13] H. Yao, S. Song, C. Qin, Z. Tang, and X. Liu, "Detection of double-compressed H. 264/AVC video incorporating the features of the string of data bits and skip macroblocks," *Symmetry*, vol. 9, no. 12, 2017.

[14] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Codec and GOP identification in double compressed videos," *IEEE Trans. on Image Processing*, vol. 25, no. 5, pp. 2298–2310, 2016.

[15] X. Jiang, P. He, T. Sun, F. Xie, and S. Wang, "Detection of double compression with the same coding parameters based on quality degradation mechanism analysis," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 1, pp. 170–185, 2018.

[16] A. Costanzo and M. Barni, "Detection of double AVC/HEVC encoding," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 2245–2249.

[17] X. Liang, Z. Li, Y. Yang, Z. Zhang, and Y. Zhang, "Detection of double compression for HEVC videos with fake bitrate," *IEEE Access*, vol. 6, pp. 53 243–53 253, 2018.

[18] Q. Li, R. Wang, and D. Xu, "Detection of double compression in HEVC videos based on TU size and quantised DCT coefficients," *IET Information Security*, vol. 13, no. 1, pp. 1–6, 2018.

[19] ITU-T and ISO/IEC JTC1, *Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video*. ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), 1994.

[20] ——, *Coding of Audio-Visual ObjectsPart 2: Visual*. ISO/IEC 14496-2 (MPEG-4 Visual), 1999.

[21] ——, *Advanced Video Coding for Generic Audiovisual Services*. ITU-T Recommendation H.264 and ISO/IEC 14496-10 (AVC), 2003.

[22] ——, *Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video*. ISO/IEC 11172-2 (MPEG-1 Video), 1993.

[23] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in *Proc. of the 39th IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6226–6230.

[24] D. Labartino, T. Bianchi, A. D. Rosa, M. Fontani, D. Vázquez-Padín, A. Piva, and M. Barni, "Localization of forgeries in MPEG-2 video through GOP size and DQ analysis," in *Proc. of the IEEE 15th Intl. Workshop on Multimedia Signal Processing*, Sept 2013, pp. 494–499.

[25] T. Wiegand and B. Girod, *Multi-Frame Motion-Compensated Prediction for Video Transmission*. Norwell, MA, USA: Kluwer Academia Publishers, 2001.

[26] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B-pictures and MCTF," in *Proc. of the IEEE Intl. Conference on Multimedia and Expo (ICME)*, July 2006, pp. 1929–1932.

[27] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 74–90, 1998.

[28] ITU-T SG16/Q15, *Video codec test model number 9 (TMN-9)*. ITU-T SG16/Q15 document Q15-D-13, Apr. 1998.

[29] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. of the IEEE Intl. Conference on Image Processing (ICIP)*, vol. 3, 2001, pp. 542–545.

[30] [Online]. Available: https://media.xiph.org/video/derf/

[31] [Online]. Available: http://ffmpeg.org/

[32] D. Vázquez-Padín and F. Pérez-González, "Prediction residue analysis in MPEG-2 double compressed video sequences," in *2019 27th European Signal Processing Conference (EUSIPCO)*, Sep 2019, pp. 1109–1113.

[33] [Online]. Available: https://www.videolan.org/developers/x264.html

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