Detection of video double encoding with GOP size estimation

D. Vázquez-Padín[#1], M. Fontani[*§2], T. Bianchi[†§3], P. Comesaña[#4], A. Piva[†§5], M. Barni[*§6]

[#] *Signal Theory and Communications Department, University of Vigo - 36310 Vigo, Spain*
[1] dvazquez@gts.uvigo.es, [4] pcomesan@gts.uvigo.es

[*] *Dept. of Information Engineering, Università degli Studi di Siena - Via Roma 56, Siena, Italy*
[2] marco.fontani@unisi.it, [6] barni@dii.unisi.it

[†] *Dept. of Electronics and Telecommunications, Università degli Studi di Firenze - Via di Santa Marta 3, 50139 Firenze, Italy*
[3] tiziano.bianchi@unifi.it, [5] alessandro.piva@unifi.it

[§] *National Inter-University Consortium for Telecommunications - Via di Santa Marta 3, 50139 Firenze, Italy*

Abstract—**Video forensics is an emerging discipline, that aims at inferring information about the processing history undergone by a digital video in a blind fashion. In this work we introduce a new forensic footprint and, based on it, propose a method for detecting whether a video has been encoded twice; if this is the case, we also estimate the size of the Group Of Pictures (GOP) employed during the first encoding. As shown in the experiments, the footprint proves to be very robust even in realistic settings (i.e., when encoding is carried out using typical compression rates), that are rarely addressed by existing techniques.**

I. Introduction

Recent advances in video compression have made possible the adoption of digital video technologies in many different fields, such as digital television braodcasting, videotelephony or Internet video streaming, among others. As it first happened with digital images, today we can easily find powerful and considerably cheap video editing software that facilitates the modification of video sequences. Consequently, in the last years, the creation of forensic tools that analyze the authenticity and integrity of digital videos has become an important field of research.

Generally, existing video editing tools do not work directly on the compressed domain, but in the reconstructed spatial domain. Therefore, the process of editing a video sequence is composed of three main steps: first, the decoding of the input sequence; second, the actual video editing; and finally, the re-encoding of the edited video (possibly with a distinct codec or different coding parameters). Several techniques have been published in the last few years trying to identify some footprints left by this double encoding process [1].

By relying on the resulting double quantization, authors in [2] propose a method to identify tampered regions on MPEG-2 video sequences with only I-frames, i.e., in a similar way as with digital images. The same authors propose in [3] to consider the information about the motion error when P-frames are used, in order to detect deletion or addition of frames.

Although these two techniques make the localization of tampered regions possible (either in the spatial or temporal domain), they do not allow to acquire knowledge about the origin of a given video stream. In this sense, some works have been developed trying to retrieve information about the processing history of a compressed video. For instance, estimation of video coding parameters has been addressed in [4], providing a method to estimate MPEG-2 settings from the decoded video stream. Valenzise et al., in [5], later extend this work to H.264 video, estimating the quantization parameter and motion vectors from decoded frames.

Concerning the first steps in the processing history of a digital video, a new approach has been proposed in [6] for the identification of the first codec applied (out of three possible ones) to a video sequence that has been doubly encoded. This method works by recompressing the video under analysis with the three possible codecs and computing a similarity measure between the two sequences. Based on the same approach, Luo et al. [7] propose a method for detecting double encoding in MPEG-2 compressed videos, by recompressing a given sequence with different sizes for the Group of Pictures (GOP) and then performing an analysis of blocking artifacts. The main drawback of all the proposed techniques for detection of double encoding is the way they are affected by the second encoding, since their performance drops very rapidly as the strength of the last compression increases.

Motivated by these shortcomings and with the aim of generalizing the double encoding detection to a scenario with several codecs, different GOP sizes and distinct target bitrates, we propose to use a robust and very distinctive footprint based on the variation of the macroblock prediction types in the re-encoded P-frames. An advantage of this Variation of Prediction Footprint (VPF) is its presence in the twice encoded video without the need of re-compression. Furthermore, given that the VPF becomes apparent only in P-frames that were intra-

WIFS'2012, December, 2-5, 2012, Tenerife, Spain.
978-1-4673-2287-4/12/$31.00 © 2012 IEEE.

coded in the first encoding, we also describe a method to estimate the size of the GOP used in the first compression. Estimation of the GOP size is not only an important step toward assessing the processing history of a digital video, but can also act as a catalyst for further forensic analysis, e.g., tampering detection. For this reason, although in this work we are not targeting video doctoring detection, we believe the proposed approach could enable further development in this direction.

In the next section, we introduce the considered scenario for double encoding detection, analyzing why the VPF appears. In Section III, we explain how this particular footprint can be measured and the method for the estimation of the first compression GOP size. Section IV presents the experimental results for validating the detection accuracy and the performance of the estimator. Finally, Section V concludes the paper discussing future research steps.

II. Preliminaries and problem statement

In the past few years, different video compression standards have emerged, being MPEG-2, MPEG-4 Visual Part 2 (we will refer to this one as MPEG-4) and H.264 the three most commonly used currently. The first one, that is also the oldest, is still widely used for video content storage in DVD and for broadcast television. The second one has been adopted in many video surveillance systems and employed for video content sharing over the Internet. The third and most recent one, is considered as the state-of-the-art in video compression, and is gradually replacing all its predecessors in almost all the mentioned applications, because it gives better performance than any of the preceding standards [8].

Since the footprint we introduce in this paper relies on principles that are valid for all the mentioned compression standards, the analysis that follows will not focus on a particular one. However, we point out that, for simplicity, in this work we do not consider the use of B-frames; to this end, we constrain the compression to be performed according to the baseline profile for H.264 and to the equivalent simple profile for MPEG-2 and MPEG-4. These profiles support only I-frames and P-frames, along with the three main types of macroblocks: intra-coded macroblocks (I-MB), inter-coded macroblocks (P-MB) and skipped macroblocks (S-MB). More types and subtypes can be considered depending on the particular standard, but these three are common to the considered codecs. Besides, even if the algorithm for intra or inter coding differs from one codec to another, the functionality is the same. In fact, for any of the three standards, an encoded I-frame can only contain macroblocks I-MB, while encoded P-frames may contain any of the mentioned macroblocks, i.e., I-MB, P-MB or S-MB. We will assume that the GOP structure is fixed for each video sequence and for the extraction of the VPF we will only process the luma component.

Let us consider the following scenario. In the first place, during the capture of a scene, a first compression is performed with an arbitrary GOP size, denoted by G_1, and a fixed constant bitrate, represented by B_1. Then, after the reconstruction of the video sequence in a RAW uncompressed video format, a second compression (temporally aligned with the first one) is carried out on the uncompressed sequence, but with a different GOP size, i.e., G_2 such that $\mathrm{G}_2 \neq \mathrm{G}_1$, and a fixed constant bitrate, i.e., B_2, that can be equal or different from the one used in the first compression. Considering this double encoding framework, a specific variation of the number of macroblocks coded as I-MB and S-MB shows up in the P-frames previously encoded as I-frames in the first compression.

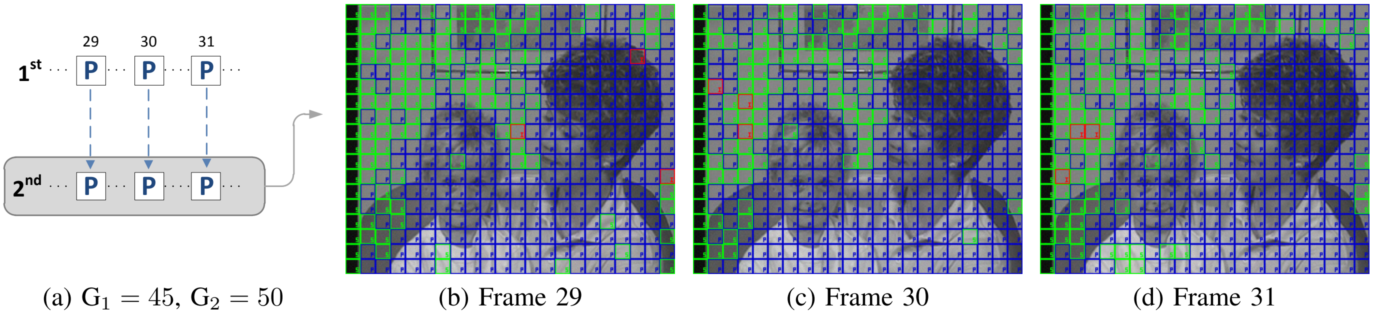
To get a better understanding on this change of macroblock types, we first consider an example where this variation does not take place and then, we analyze the opposite situation. Fig. 1 refers to the first case where a double encoding with $\mathrm{G}_1 = 45$ and $\mathrm{G}_2 = 50$ is considered. The conversion between the types of frames for the indices 29, 30 and 31[1] is illustrated in Fig. 1(a), and as we can see, each P-frame in the first compression is encoded again as a P-frame. From Figs. 1(b)-(d), where the macroblock types for the doubly compressed P-frames are overlaid, we cannot notice a clear variation of the number of each type of macroblock between the 3 considered frames.

Nevertheless, if we just change the GOP size in the first compression to $\mathrm{G}_1 = 30$ and we repeat the same double encoding, we get the results shown in Fig. 2. In this case, as it is depicted in Fig. 2(a), the frame with index 30 is converted from an I-frame to a P-frame in the second compression. Checking the corresponding macroblock types for the frame 30 in Fig. 2(c), we can easily appreciate a noticeable increase of I-MB and a considerable reduction of S-MB. Hence, the VPF takes place in the frame number 30. Since until this frame nothing changes from the previous case, we get exactly the same macroblock types for frame 29, and, as it can be observed, the number of each macroblock type in the frame 31 returns to its normal value, even if the underlying grid has changed.

The explanation of this effect is based on the different way an I-frame is encoded with respect to a P-frame. Generally, the quantization matrix or the quality factor for encoding an I-frame differs from the one considered for a P-frame because I-frames are used directly or indirectly as a reference for encoding several future frames. Besides, the following effects are observed:

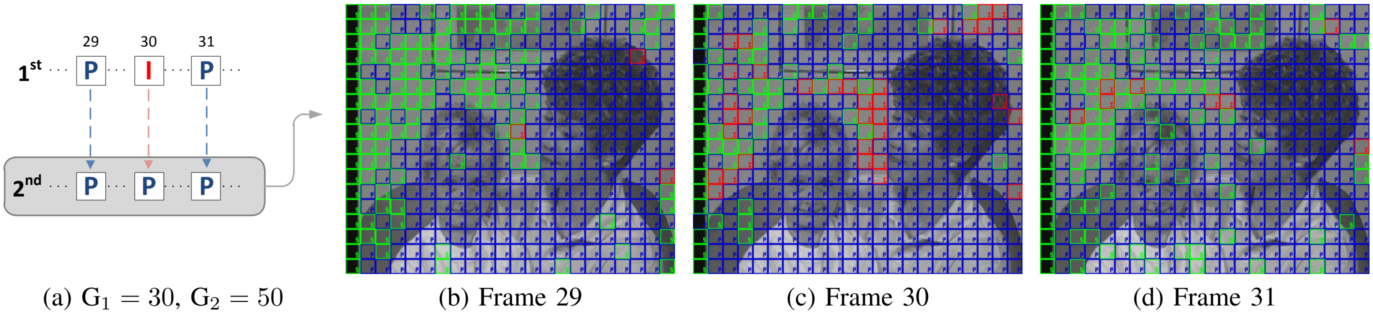
- Change of a P-MB or S-MB in homogeneous regions into an I-MB. In general, the use of a macroblock I-MB in a P-frame is intended for encoding more efficiently a region where there is not a good match in previous reference frames, like a new uncovered region.
  In this case, the compression of a reconstructed I-frame with a P-frame (whose reference frame will probably not be so correlated with this uncompressed frame) will lead to a less efficient encoding in general. However, if the changes introduced by the I-frame are small in homogeneous regions (for instance, like a change in the DC component of a whole block), then those blocks will

[1] Note that we consider that the frame indices start counting from 0.



(a) $G_1 = 45$, $G_2 = 50$ (b) Frame 29 (c) Frame 30 (d) Frame 31

Fig. 1. Example where the VPF is not present. Leftmost picture shows the types of frames with indices 29, 30 and 31 for both compressions. The remaining three pictures represent the macroblock types for each frame. Red color is used for I-MB, blue color for P-MB and green color for S-MB. Both first and second encodings are carried out using the `x264` library, with a QP fixed to 20.



(a) $G_1 = 30$, $G_2 = 50$ (b) Frame 29 (c) Frame 30 (d) Frame 31

Fig. 2. Example where the VPF becomes apparent for frame 30.

be more efficiently coded as I-MB than macroblocks P-MB where at least a motion vector should be considered and more bits would be needed. That is the main reason why the I-MB appear in smooth regions.

- Change of S-MB in static regions into a P-MB. The use of skipped macroblocks is very likely for any encoder given that neither residual information nor motion vector are needed and a lot of bits are saved.
  Nevertheless, in the case we are studying, when a reconstructed I-frame comes into play during the encoding of a P-frame, small variations are introduced in static regions with respect to its reference frame and, thus, the use of a S-MB is no longer possible. Consequently, a P-MB must be used instead for satisfying the perceptual requirements.

As we stated earlier, even if each standard performs prediction and quantization in a different way, the common characteristics shared by the codecs make them agree with the behavior described above. Of course, the presence of VPF will also depend on the particular implementation of each codec, but since the main objective of any implementation is to reduce the bitrate according to a predefined quality, the observed behavior should also be consistent with any specific implementation.

As a conclusion, if we can detect those variations in the number of prediction types I-MB and S-MB, then we will be able to detect if a double encoding of the same sequence has been carried out and, if this is the case, we have a way to estimate the size of the first GOP from those variations.

III. Measuring the VPF

In this section we show how the VPF can be used to detect double encoding and to estimate the GOP size of the first compression. The method we introduce is essentially based on two steps: first, the frames showing the VPF are located, and the strength of the footprint is measured; secondly, since the obtained signal should show relevant peaks where I-frames of the first compression were located, a periodicity analysis is carried out.

In the rest of this section, the following notation is used: for a given video sequence $\underline{\boldsymbol{x}}(n)$, with $n = 0, \ldots, N-1$, being N the total number of frames, we denote with $i(n)$ and $s(n)$ respectively the number of macroblocks I-MB and S-MB that are present in the n-th frame. We also recall that G_1 and G_2 are the GOP sizes used for the first and the second compression, respectively.

a) Peak extraction: In this phase, we jointly consider the two signals $i(n)$ and $s(n)$. From Section II, we know that in correspondence of those P-frames of the video that were encoded as intra in the first compression, the number of I-MB will increase while the number of S-MB will decrease. Obviously, we cannot consider directly $i(n)$, since all the macroblocks are I-MB where the I-frames of the second encoding are located, resulting in very strong peaks that are not related to the first encoding. However, since G_2 is known, we can simply remove those peaks located at frames $k\mathrm{G}_2$, where $k = 0, \dots, \lfloor N/\mathrm{G}_2 \rfloor$. To do so, we substitute those elements of the array with the average value obtained from the previous and the following ones: $i(k\mathrm{G}_2) = (i(k\mathrm{G}_2+1)+i(k\mathrm{G}_2-1))/2$.

For the sake of clarity, we will denote by $\mathcal{P}$ the set of frames where the effect described in Section II is present, where

$$\begin{aligned}\mathcal{P} = \{n \in \{0, \dots, N-1\} : i(n-1) < i(n) \ \wedge \\ i(n) > i(n+1) \ \wedge \ s(n-1) > s(n) \ \wedge \ s(n) < s(n+1)\}.\end{aligned}$$

Based on that, we define a new vector that quantifies the strength of the effect for every $n \in \mathcal{P}$ as follows

$$v(n) = \begin{cases} E(n), & \text{if } n \in \mathcal{P} \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where $E(n)$ measures the energy of the effect in the n-th frame, being defined as

$$\begin{aligned}E(n) = |(i(n) - i(n-1))(s(n) - s(n-1))| + \\ + |(i(n+1) - i(n))(s(n+1) - s(n))| .\end{aligned}$$

Indeed what we expect is, first, an increase in the number of I-MB together with a decrease in the number of S-MB and, then, a decrease in the number of I-MB together with an increase in the number of S-MB. Therefore, by taking the product of the variations of $i(\cdot)$ and $s(\cdot)$ we measure the magnitude of the effect we are considering.

b) Analysis of periodicity: The second step of the proposed scheme consists in investigating the periodicity of the extracted feature. If no periodic behavior is detected we can classify the video as singly encoded; conversely, if a periodicity is present, then it will allow us to estimate G_1.

Usually, the periodicity of a signal is well-exposed using its frequency representation, e.g. taking its Fourier transform; however, this approach is well-suited for cases where many periods of the signal are available, otherwise the resulting representation is noisy and periodicity estimation is inaccurate. On the other hand, we want our method to work also with a limited number of frames, so the frequency representation is not the best tool for our task.

For these reasons, we propose a simple yet effective approach for estimating the periodicity of peaks in $v(n)$, that is based on two steps: candidate GOP selection, and candidate evaluation.

The candidate GOP selection aims at determining a set of possible values for G_1. Since we are searching, in a sequence of integers, an element generating a subsequence of multiples of itself, it makes sense to restrict the search to the set of the Greatest Common Divisors (GCD) between all possible couples of elements of the sequence. Therefore, we define the set $\mathcal{C}$ of candidate GOPs as

$$\mathcal{C} = \{c \in \{2, \dots, N\} : \exists \ n_1, n_2 \in \mathcal{P}, \mathrm{GCD}(n_1, n_2) = c\}.$$

Notice that evaluating $\mathcal{C}$ requires at most N^2 runs of the GCD algorithm, whose complexity is quadratic in the number of base-10 digits of its argument ($\lceil \log_{10} N \rceil$ at most, in our case). However, since the signal $v(n)$ is typically sparse (in the experiments presented in Section IV, $\sim 90\%$ components are null on average), the practical computational effort is surely affordable.

In the GOP estimation stage, each candidate value $c \in \mathcal{C}$ is associated with a fitness value $\phi : \mathcal{C} \to \mathbb{R}$, that measures how well the choice of c models the periodicity of the signal $v(n)$. Before giving the formal definition of $\phi(c)$, we briefly give the intuition behind this measure. Due to content related issues, like sudden changes of scene or strongly textured regions, the signal $v(n)$ could contain some noisy components, or could be missing some expected peaks in multiples of G_1. With this in mind, it is essential to define a fitness measure that takes into account, for each candidate value c, the following aspects:

1) The energy of peaks that are located in multiples of c, given by
$$\phi_1(c) = \sum_{i=kc} v(i) \ , \quad \text{with } i \in \mathcal{P}, k \in \left[0, \left\lfloor \frac{N}{c} \right\rfloor\right] .$$
2) The absence of peaks that would be expected in multiples of c, quantified as
$$\phi_2(c) = \sum_{i=kc} \beta \ , \quad \text{with } i \notin \mathcal{P}, k \in \left[0, \left\lfloor \frac{N}{c} \right\rfloor\right] ,$$
where β is a penalization factor for missing peaks, that can be taken as $\beta = 0.1 \times \max_n\{v(n)\}$.
3) The energy of the most relevant periodic component with a period smaller than c, defined as
$$\phi_3(c) = \max_{z \in [1, c-1]} \left\{ \sum_{k=0}^{\lfloor N/z \rfloor} v(kz) \right\} .$$

Then, we combine these three measures to define the function $\phi(c)$ as

$$\phi(c) = \phi_1(c) - \phi_2(c) - \phi_3(c), \tag{2}$$

where it is evident that ϕ_2 and ϕ_3 act as a penalization for the candidate c. Once the fitness of every candidate in $\mathcal{C}$ has been evaluated, we can classify the video as singly or doubly encoded and, in the latter case, provide the estimate for G_1. The video $\underline{\boldsymbol{x}}(n)$ is assigned to a class with the following rule:

$$C(\underline{\boldsymbol{x}}) = \begin{cases} 1, & \text{if } \max_{c \in \mathcal{C}} \phi(c) > T_\phi \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where T_ϕ is a threshold, $C(\underline{\boldsymbol{x}}) = 1$ for videos classified as doubly encoded, and $C(\underline{\boldsymbol{x}}) = 0$ for videos classified as singly encoded. Whenever a video is classified as doubly encoded, the estimate of G_1 is

$$\hat{\mathrm{G}}_1 = \arg\max_{c \in \mathcal{C}} \phi(c). \tag{4}$$

IV. Experimental results and discussion

The performance of the proposed approach for double encoding detection and for GOP size estimation is evaluated in this section. To this end, a realistic setting is considered, which is often challenging for video forensics. We build the datasets for our experiments using 14 video sequences[2] with CIF resolution, i.e., 352×288 pixels, that are available in YUV-uncompressed format. Given that these sequences have different lengths, we always limit ourselves to consider only their first 250 frames (that is, 10 seconds of video at 25 fps), in order to investigate the reliability of the proposed approach in presence of short clips. Furthermore, in all the experiments, video encoding is performed specifying a target constant bitrate (CBR) and not by fixing the quantization parameters, since this is the typical encoding setting in a realistic scenario. As it was mentioned in Section II, adaptive GOP structures are not considered in this work. For all the tests, we have used the `libavcodec` and `x264` libraries (through `FFmpeg`) to encode/decode all the videos.

Because we propose to use the VPF both for double encoding detection and GOP size estimation, we split the experiments in two parts; this choice also accounts for the different nature of these tasks, since detection and estimation methods need different evaluation criteria.

A. Double encoding detection

To test the discrimination capability of the proposed approach, we use the mentioned 14 raw sequences to create a dataset consisting of:

- 672 singly encoded videos, by using all combinations of encoders and parameters in the right column of Table I;
- 672 doubly encoded videos, randomly sampling for each sequence 48 joint configurations for first and second encoding from those allowed by Table I.

Since the proposed detection method relies on a threshold-based rule (see eq. (3)), we use Receiver Operating Characteristic (ROC) curves to evaluate its performance: we report in Fig. 3 both the ROC of the proposed method on the whole dataset, and the performances obtained separately, differentiating the encoder employed for the second compression (that, of course, is known to the analyst). It is worth noting that when the second encoding is carried out using H.264 (as we have seen, the most commonly used nowadays), the detector yields its best performance (94% detection rate for a false positive rate of 5%). In fact, while the VPF will rarely appear, independently from the codec being used, in singly encoded sequences, it cannot be taken for granted that it will show up clearly in a doubly encoded video: when the quality of the second compression is very low (e.g. 100kb/s) the footprint could be hidden by spurious effects. This explains the behavior shown in Fig. 3: since H.264 is known to provide better quality with respect to MPEG-x codecs for a fixed bitrate, it facilitates the detection of the VPF and, consequently, the correct classification of the video. That said, the proposed method retains considerable accuracy also when MPEG-x codecs are used, and yields on average a detection rate of 80% when only 5% of false positives are allowed.

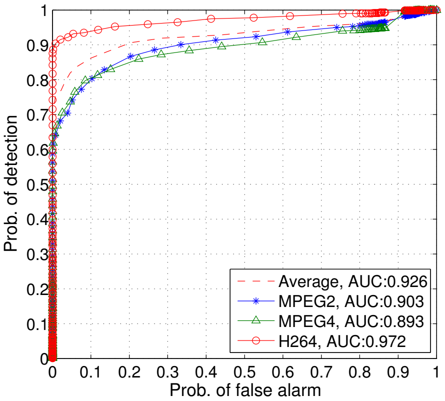


Fig. 3. ROC curve for the proposed double encoding detector.

TABLE I
PARAMETERS FOR CREATING DOUBLY ENCODED SEQUENCES

Parameters	1st encoding	2nd encoding
Encoder	{MPEG-2, MPEG-4, H.264}	{MPEG-2, MPEG-4, H.264}
Bitrate (kb/s)	{100, 300, 500, 700}	{100, 300, 500, 700}
GOP size	{10, 15, 30, 40}	{9, 16, 33, 50 }

B. First GOP size estimation

For studying the performance of the proposed GOP size estimation technique, we create a dataset of 32,256 doubly encoded videos, by compressing each of the 14 available sequences with all combinations of settings given in Table I. Each sequence is analyzed in about 1.4 seconds on a desktop computer[3], but the actual analysis, that starts when types of macroblocks have been extracted, takes only 0.025 seconds.

We investigate the results of the estimation method from different points of view: as a function of 1st and 2nd bitrate, as a function of the 1st and 2nd encoder, and as a function of the 1st and 2nd GOP size. Each time we investigate a parameter, all the other settings are marginalized out, i.e., results are averaged over them. Each estimate is classified as exact (that is, $\hat{G}_1$ from eq. (4) actually matches G_1) or wrong, since we believe that having just an approximation of G_1 is not meaningful from a forensic point of view. Finally, since we are considering 14 different source sequences, for each experiment we report: i) average performance; ii) performance for the video sequence yielding best results (*paris* in all the experiments); iii) performance for the video yielding worst results (*waterfall* in all the experiments).

In Fig. 4 we report the performance as a function of B_1 - B_2 combination of bitrates. We see that lower bitrates for the first encoding result in higher performance, in agreement to what is said in Section II: since low bitrates require strong quantization, acting like a lowpass filter, the number of blocks that will be more conveniently encoded as I-MB will increase. This will be especially true for videos where uniform regions are available, like the *paris* sequence (which yields the best results), while textured content is against this phenomenon,

[2] Freely available at this website: http://trace.eas.asu.edu/yuv/
Chosen sequences are: *akiyo*, *bridge-close*, *bridge-far*, *coastguard*, *container*, *foreman*, *hall*, *highway*, *mobile*, *news*, *paris*, *silent*, *tempete*, *waterfall*.

[3] Intel Core2Duo @3.4GHz, 8GB RAM, running Ubuntu 10.04.

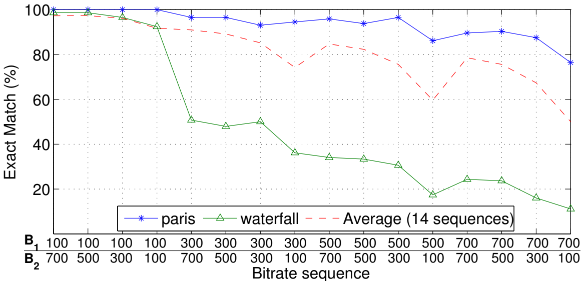


Fig. 4. Performance of the method as a function of the B_1-B_2 bitrate.

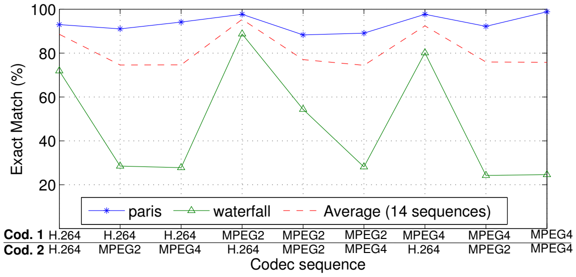


Fig. 5. Performance of the method as a function of the codec combination.

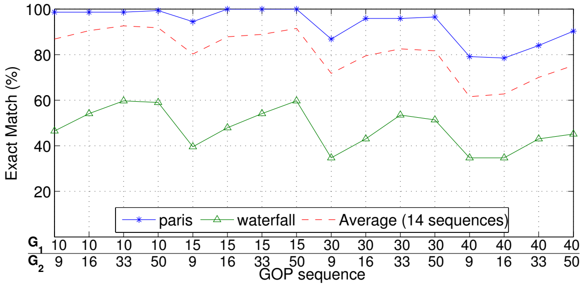


Fig. 6. Performance of the method as a function of the G_1 - G_2 combination.

as confirmed by the *waterfall* sequence (which is rich of textures) being the worst. From the second compression point of view, it is confirmed that low bitrates negatively affect the performance, since they reduce the possible choices for the encoder when assigning macroblock types; nevertheless, even in the worst conditions, the proposed footprint is able to correctly estimate G_1 half of the times.

Fig. 5 shows the performance for different combinations of codecs. We see that reliability increases when the second encoding is carried out with H.264, in agreement to what we observed in Section IV-A about the presence of VPF in doubly encoded videos.

Finally, we evaluate (Fig. 6) the performance for different combinations of G_1 and G_2. Results show an intuitive fact: as G_1 increases, the accuracy of the method drops. The simplest justification for this phenomenon is that, since we are using a fixed number of frames for the estimation, the higher G_1, the less number of periods we are able to observe. This, as expected, results in noisier estimates. Another interesting fact is that results improve as G_2 increases: in fact, this reduces the number of spurious effects induced by the GOP structure of the second compression.

V. Conclusions

Video forensics is an emerging field, targeting the investigation of the processing history of a digital video. To this extent, detecting whether a video has been compressed once or twice is an interesting task, especially if an estimation of some of the 1st encoding parameters can be provided. In this work, we have introduced both a new kind of footprint based on the variation of the macroblock prediction types in the re-encoded P-frames (VPF), and a method to exploit this footprint to detect video double encoding and estimate the size of the GOP used in the first compression.

Experiments show that, being based on a simple principle, the VPF is a very robust footprint: detection of double encoding and estimation of GOP size remain possible (although with some impact on performance) even when the second compression is stronger than the first one, while this configuration is prohibitive for most of the existing forensic methods. This work opens to several future developments: firstly, the proposed approach to periodicity estimation can be extended to address the case where the video is re-encoded after that some frames have been removed. Secondly, since the VPF yields good results on relatively short sequences, when a long video has to be analyzed we could both expand the number of considered frames or use a decision fusion method to improve the reliability. Finally, we will also consider the analysis of videos that are compressed using encoding profiles of MPEG-x and H-264 that introduce B-frames in the GOP structure.

Acknowledgment

Research supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission under project REWIND (FET-Open grant number: 268478), the European Regional Development Fund (ERDF) and the Galician Regional Government under projects Consolidation of Research Units 2010/85 and SCALLOPS (10PXIB322231PR), and by the Spanish Government under project COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010) and DYNACS (TEC2010-21245-C02-02/TCM).

References

[1] S. Milani *et al.*, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, 2012.

[2] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *Proceedings of the 8th workshop on Multimedia and security*, ser. MM&Sec '06. NY, USA: ACM, 2006, pp. 37–47.

[3] ——, "Exposing digital forgeries in video by detecting double quantization," in *Proceedings of the 11th ACM workshop on Multimedia and security*, ser. MM&Sec '09. NY, USA: ACM, 2009, pp. 39–48.

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

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

[6] P. Bestagini *et al.*, "Video codec identification," in *2012 37th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, 2012.

[7] W. Luo, M. Wu, and J. Huang, "MPEG recompression detection based on block artifacts," in *SPIE Conference on Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, 2008.

[8] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. John Wiley & Sons, 2010.