

Received October 27, 2021, accepted December 14, 2021, date of publication January 6, 2022, date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140588

Statistical H.264 Double Compression Detection Method Based on DCT Coefficients

GAËL MAHFOUDI¹, FLORENT RETRAINT¹, FRÉDÉRIC MORAIN-NICOLIER²,
AND MARC MICHEL PIC³

¹Institute Charles-Delaunay, University of Troyes, 10300 Troyes, France

²CRéSTIC EA 3804, Université de Reims Champagne Ardenne, 51097 Reims, France

³Digital Laboratory, SURYS, 77607 Bussy-Saint-Georges, France

Corresponding author: Gaël Mahfoudi (gael.mahfoudi@utt.fr)

This work was supported by the French National Research Agency (ANR) Project DEFACTO under Project ANR-16-DEFA-0002.

ABSTRACT With the 2019 Coronavirus pandemic, we have seen an increasing use of remote technologies such as remote identity verification. The authentication of the user identity is often performed through a biometric matching of a selfie and a video of an official identity document. In such a scenario, it is essential to verify the integrity of both the selfie and the video. In this article, we propose a method to detect double video compression in order to verify the video integrity. We will focus on the H.264 compression which is one of the mandatory video codecs in the WebRTC Requests For Comments. H.264 uses an integer approximation of the Discrete Cosine Transform (DCT). Our method focuses on the DCT coefficients to detect a double compression. The coefficients roughly follow a Laplacian distribution, we will show that the distribution parameters vary with respect to the quantisation parameter used to compress the video. We thus propose a statistical hypothesis test to determine whether or not a video has been compressed twice.

INDEX TERMS Video forensics, double compression, DCT, H.264, hypothesis testing.

I. INTRODUCTION

With the 2019 Coronavirus pandemic, we have seen an increasing use of remote technologies such as remote identity verification. In a remote identity verification system, a video acquisition of both the Identity Document and the person seems like an obvious choice.

In fact, the person and the ID are not static by nature and thus require many frames to be authenticated. Video has been commonly used for some time to perform liveness verification of an individual and is being used more and more to authenticate security elements such as holograms or variable ink on identity documents. Another great advantage of video stream against simple images is the added complexity for a counterfeiter to tamper such a stream.

In fact, with a video stream the counterfeit needs to develop complex tampering algorithms that work in real time. To tamper a text field, a simple copy-move would be enough for an image. For video, the counterfeit would have to detect and precisely track the identity document by using methodology such as shape-from-template. We understand intuitively how challenging the tampering process become in comparison to

a simple image tampering. Recently, those arguments were acknowledged and lead to new regulations such as the French requirement rule set for remote identity verification service providers [1] enforcing the use of video in the context of remote identity verification.

The challenging aspect of video tampering must not induce a blind confidence in such media. Remote identity verification is heavily based on face biometry we thus expect attacks on either the live person acquisition or on the identity document picture. If the detection and tracking of the full document are not particularly well study. Face detection and tracking, on the other hand, has been extensively studied for quite some time now. The research in this field is in fact so advanced that it is even possible to detect and track as much as 468 3D face landmarks in realtime in a web browser using open-source frameworks [2], [3]. Assuming that the counterfeit will not be able to tamper the video stream in realtime or inject a prepared video is thus unreasonable.

We see that before any biometric matching between a person and the identity document, it is necessary to first authenticate the video media. While liveness detection methods are well studied and allow to reasonably reject the hypothesis of an injected stream when combined with random challenges such as eye blinking, smiling, etc. Those are not

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

enough to authenticate the video, as it could be tampered in realtime.

In this article, we suppose that a counterfeit will tamper a video in realtime. We assume that the acquisition device is controlled and safe, and that the counterfeit will intercept the stream before being sent to the server. In order to tamper the video, the counterfeit must first decompress the stream then perform the tampering and finally recompress it before sending it back to the server. Detecting the double compression of the video is thus a first step toward authenticating the media. We will focus on the H.264 compression which is, along with VP8, the only codec imposed by the WebRTC RFC [4].

A. STATE OF THE ART

The first H.264 encoder has been officially approved in 2003. It was proposed to have an extension to the previous encoder i.e. H.263 and aimed at providing a good visual quality while lowering the bitrate as much as possible. This led to a few major differences from previous encoders. Even though H.264 has been around since 2003, many research [5]–[8] kept focusing on older versions. This made sense as older encoders were still extensively used at that time and H.264 was still rapidly evolving. Nowadays, H.264 has become one of the most used video encoders in particular for video content on the internet as it is one of the two mandatory video codecs used in the WebRTC protocol.

This extensive use soon encouraged researchers to move their attention to H.264 instead of older encoders. In its core principles, H.264 is similar to the older standards. In particular, it is mainly composed of two stages. A first prediction stage aiming at reducing the amount of information and a second stage which further compress that information using a DCT transformation and quantification. Unlike previous standards, H.264 introduced a new integer approximation of the DCT transform and also introduced a variable size prediction algorithm.

As most video encoding algorithms, H.264 takes advantage of the temporal redundancy in video to reduce the information needed to encode multiple frames. H.264 groups many frames into a Group Of Pictures (GOP) where an I-frame usually serves as a reference and the next frames (P or B-frames) are predicted based on this I-frame and other B or P-frame of the same GOP. When a video is compressed twice, some I-frame might be recompressed as P or B-frame and vice versa. This is often called frame relocation. Many research focuses on frame relocation to detect double video compression. In [9]–[11], authors trained deep neural networks on the frame residual to detect relocated frames. In [12], authors trained a One-Class classifier on the reconstructed frame residual to detect the double compression. In [13], the authors directly study the bit size of each encoded frame. They showed that relocated I-frame requires more bits than typical P or B-frame and can thus be detected. This allows them to estimate the primary GOP size in case of a double compression. Similarly [14]–[20] also try to estimate the primary GOP size as an evidence of double H.264 compression.

One advantage of those methods is that they are applicable to other video encoder as the principle of GOP is present in many video compression algorithms.

Other approaches such as [21]–[23] focus on recompression using the same quantification parameters. They showed that for H.264 the frames converge to a particular state when compressed multiple times using the same quantification parameters. This property can be exposed through an analysis of the DCT coefficient or using the frame noise residual.

Finally, some methods [24], [25] try to expose the double H.264 by studying the DCT coefficient distribution. They trained different classifier on the DCT coefficient to detect if a video is compressed twice.

B. ORGANISATION OF THE PAPER

The paper will be organised as follows. A brief overview of the main step of the H.264 compression will first be introduced. After, the motivation behind the choice of the analysis of the DCT coefficient to expose a double compression will be explained. Then we will present how those coefficients are sampled and modelled prior to the analysis.

We will then derive two hypothesis tests to detect a double video compression. First, a simple ratio test will be presented when all parameters are known in advance. Then a generalised likelihood ratio test will be introduced to take into account the lack of knowledge regarding some parameters.

Then, a few numerical experimentation will be performed. We will first validate the theoretical model and evaluate the performances on a set of simulated frames. Then the method will be evaluated on a set of real video.

Finally, we will conclude with a few remarks and perspectives regarding the presented method.

II. H.264 INTRA-FRAME COMPRESSION

In this section, we will give a brief overview of the main steps of the H.264 compression. We will skip through many aspects of the compression as they are not relevant in our analysis. We encourage the reader to read [26] to get a more in-depth presentation of the complete H.264 encoding process.

We will only focus on the intra-frame compression and on the luma component in the rest of the chapter. Intra-frames, and the luma component, of H.264 stream contains the most of the information.

For those frame, the compression is mostly divided into two major steps. The prediction step and the transformation and quantification step. We will first briefly explain the objective of the prediction step and then explain the transformation and quantification process. Finally, we will briefly introduce the mechanism of the rate control which is a relevant part of the encoding process for our method.

A. PREDICTION

At the prediction stage the H.264 aim at producing an estimate of the frame using the least amount of information as possible. To do so, the frame is first split into Macroblocks (MB) of size 16×16 . Each MB is then predicted

only by extrapolating information from neighbouring MBs. For intra-frames the MB can be predicted at three different sizes i.e. 16×16 , 8×8 and 4×4 . In each case, the MB is subdivided into smaller sub-blocks that are predicted using information from already decoded sub-blocks or neighbouring MB. For each sub-block, the encoder find the best approximation (in terms of sum of absolute error) by choosing one of the available prediction modes for a given sub-block size. In the rest of the article, we will use the notation PredX with X the size of the prediction used to refer to a MB subdivided into sub-blocks of size X. The prediction PredX dictates which transformation will be used in the following stage, so we will always treat MB with different prediction mode separately.

Once the prediction is made, it is subtracted to the current frame to obtain a residual. This residual is mostly null and can thus be compressed efficiently.

B. TRANSFORMATION AND QUANTIFICATION

The residual is compressed using a process similar to JPEG. It first transformed into the frequency domain using a DCT transform and then compressed by removing higher frequencies.

The DCT transformation is an approximation of the integer DCT. In H.264 there exist two main transformations. A 4×4 DCT transformation for MB predicted with Pred4 and Pred16. And an 8×8 transformation for Pred8. It is worth noticing that the 8×8 prediction and transformation are only available in the High compression profile of H.264. In theory, this profile is not mandatory in the WebRTC RFC [4]. In practice, this profile has been included in H.264 version 3 in 2005 and is nowadays the most commonly used profile. Both transformation follows the same principle. First the residual is transformed, then it is scaled and quantised:

$$\mathbf{C} = \lfloor (\text{DCT}(\mathbf{R}) \circ \mathbf{Q}) \cdot s \rfloor \quad (1)$$

with \circ the Hadamard product, \mathbf{R} the residual sub-block, \mathbf{Q} the quantification matrix and s a scaling scalar.

The quantification matrix \mathbf{Q} and the scaling scalar s depends on the quantisation parameter QP. This quantisation parameter can vary between MBs. In H.264 QP can vary from 0 to 51 with 0 being almost lossless, 23 considered as visually lossless and 51 the strongest compression.

When Pred16 is used, an additional transformation, called the DC transform, can be applied. This transformation is applied to every DC component just before quantification. We decided to ignore MBs predicted with Pred16 for simplicity. For the rest of the article, we will only consider MBs predicted either with *Pred8* or *Pred4*.

C. RATE CONTROL

As we mentioned the quantisation parameter QP can vary for each Macroblocks within the same frame. This depends on the rate control used by the H.264 encoder. There exists multiple modes that can be chosen for the rate control.

There are mainly two objectives that one might want to achieve when compressing with H.264. He will either want to archive the file or stream the file. For archiving, the typical rate controls used are the Constant QP which maintain a fixed QP for each frame or the Constant Rate Factor (CRF) which will try to maintain a constant visual quality given a target QP. When streaming, rate controls that try to maintain a given bitrate is usually preferred such as the Average Bitrate mode or the Constant Bitrate mode.

Apart from the constant QP rate control, every mode allows the encoder to vary the QP per Macroblock. This implies that the choice of QP for each Macroblock cannot be controlled exactly unless one chooses the constant QP mode. While it is possible to implement a H.264 encoder for which we can control the QP at the Macroblock level, we argue that it is not trivial and we will consider that the counterfeit will use a standard encoder a will thus not have full control over the QP.

D. IMPACT OF A DOUBLE H.264 COMPRESSION

We briefly introduced the I-Frame compression in the earlier section. We showed that a frame is first segmented into many Macroblocks of size 16×16 . Every Macroblock is then predicted in order to extract a residual. That residual is finally transformed using an integer approximation of the DCT and quantised. One particularity of an H.264 encoder is that it can change the algorithm used to perform the prediction, the type of DCT and the quantisation parameter at the Macroblock level. All that information can be retrieved for each Macroblock while decoding the H.264 stream. But when compressing a video using a standard H.264 encoder, those parameters cannot be predicted in advance. As a result, when for a Macroblock predicted using PredX and a quantisation parameter QP_1 we expect to observe things in case of a double compression:

- 1) The MB will be predicted by PredY with $Y \neq X$
- 2) The MB will be quantised using $QP_2 \neq QP_1$

Of course we could have $Y = X$, $QP_2 = QP_1$ in which case the recompression will have no impact on the MB. Nevertheless, it is reasonable to assume that a non-negligible number of MB will be recompressed with either $Y \neq X$ or $QP_2 \neq QP_1$ or both.

We thus propose to study the distribution of the DCT coefficient to detect a double compression. In particular, we will see that the coefficients of MB predicted using PredX and a quantisation parameter QP_1 have a characteristic distribution and that the recompression have an impact on that distribution.

E. SAMPLING BY QUANTISATION PARAMETER AND PREDICTION MODE

As previously exposed, the prediction and compression are performed at the level of Macroblocks. While processing a video, it is thus proposed to first partitioned all Macroblocks according to their prediction mode i.e. *Pred4* and *Pred8*.

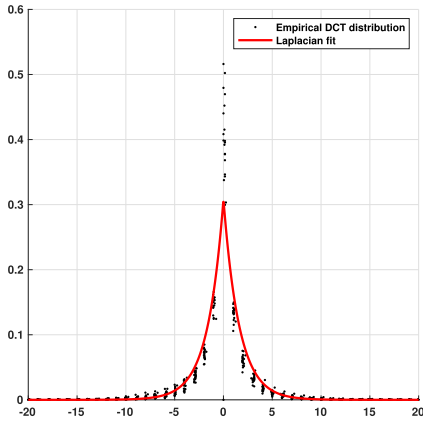


FIGURE 1. Empirical distribution of the DC coefficient for 20 videos fitted with a Laplacian distribution.

This partitioning is necessary as the prediction mode also dictates which transformation is applied before the quantification. Then the Macroblocks are further partitioned according to the quality factor QP used. With $\mathbf{B}^{x,q}$ denoting all the sub-blocks predicted with $Pred_x$ and quantified at QP, we thus have a set of vectors denoted $\mathbf{C}_{i,j}^{x,q}$ containing all coefficients at the location (i, j) of each sub-block $\mathbf{B}^{x,q}$.

F. MODELLING OF THE COEFFICIENT

In this article, we propose to study the DCT coefficient. In particular, we propose to study if the DCT coefficient at a specific quantification level can be characterised. The distribution of DCT coefficients for images has been extensively studied. Firstly, supposed to be normally distributed [27]. It was, then showed that the Laplacian distribution [28] was a better modelling for AC coefficients. Since then, the Laplacian modelling has been a predominant choice because of its simplicity and good overall accuracy. Another model has been proposed such as Cauchy [29], Gaussian mixture [30] etc. More recently the authors of [31] proposed a doubly stochastic model of AC coefficients and showed that it was more accurate than other models. For H.264, the Laplacian and Cauchy distribution remain the preferred choice [32].

We will consider that the DCT coefficients $\mathbf{C}_{i,j}^{x,q}$ follow a Laplacian distribution:

$$\mathbf{C}_{i,j}^{x,q} \sim \text{Laplace}(0, b_{i,j}^{x,q}). \tag{2}$$

In Fig. 1 it can be seen that the Laplacian distribution is indeed a good approximation.

In Fig. 2 it can be seen that for a given QP the parameter $b_{i,j}^{x,q}$ seems stable across multiple videos. To the best of our knowledge, this stability was first pointed in [32]. We will thus consider a single scale parameter b for each tuple (x, q, i, j) .

As shown in [33], it is not possible to assume the coefficient of a DCT transformation independent and identically distributed (i.i.d) when directly applied to the image content. In H.264, the prediction tries to approximate each pixel value. This prediction can be seen as an estimator for

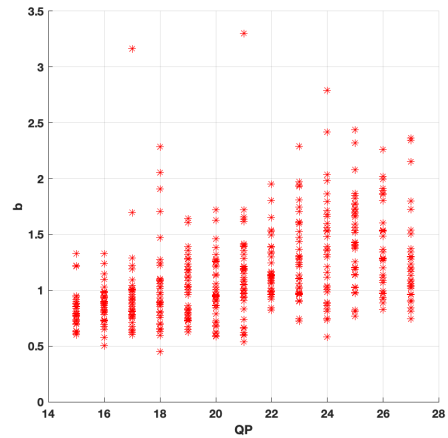


FIGURE 2. Distribution of $b_{1,1}^{4,q}$ for various QP for 40 video.

each pixel mean. The DCT transformation is finally applied on the residual of the initial frame to which the prediction is subtracted. This allows us to consider $\mathbf{C}_{i,j}^{x,q}$ i.i.d.

In the following sections, we will omit the tuple (x, q, i, j) to improve readability. The coefficients $\mathbf{C}_{i,j}^{x,q}$ for a given tuple (x, q, i, j) will simply be denoted as $\mathbf{C} = \{c_1, c_2, \dots, c_N\}$ with N the number of coefficients. In the same manner, $b_{i,j}^{x,q}$ will be denoted as b . Finally, all the coefficients $c_i, i \in [1; N]$ will be considered i.i.d.

The probability density function for a given coefficient c_i is thus given by

$$f(c_i|b) = \frac{1}{2b} \exp\left(\frac{-|c_i|}{b}\right). \tag{3}$$

III. STATISTICAL TEST DESIGN

We consider that \mathbf{C} follows Laplacian distribution with zero mean and with scale b . We expect b to be affected by the double compression process. In the following section, we will first introduce the first statistical test when every parameter is known (i.e. the value of b for the first and second compression). Then we will derive a more practical test where only the first compression parameter is known.

A. LIKELIHOOD RATIO TEST FOR TWO SIMPLE HYPOTHESES

We saw that for a given tuple (x, q, i, j) , the scale parameter b seems to approach a fixed value. We will thus assume in the rest of the article that for a video compressed with H.264 once. The coefficients \mathbf{C} follows a zero mean Laplacian distribution of scale b_0 .

To verify if a video has been compressed twice we then propose to define the following hypothesis test.

$$\begin{cases} \mathcal{H}_0 : \mathbf{C} \sim \text{Laplace}(0, b_0) \\ \mathcal{H}_1 : \mathbf{C} \sim \text{Laplace}(0, b_1), b_1 \neq b_0. \end{cases} \tag{4}$$

If the video has gone through a single compression then it should follow a Laplacian distribution of scale b_0 . Else, it will follow a Laplacian distribution of scale b_1 .

We can define the likelihood ratio as

$$\Lambda(\mathbf{C}) = \frac{\mathcal{L}_1(\mathbf{C})}{\mathcal{L}_0(\mathbf{C})}. \quad (5)$$

Because the coefficients $c_i, i = \{1, 2, \dots, N\}$ are i.i.d, we can rewrite the likelihood ratio as

$$\Lambda(\mathbf{C}) = \prod_{i=0}^N \Lambda(c_i). \quad (6)$$

The log-likelihood ratio is then obtained by combining (35) and (6)

$$\begin{aligned} \Lambda(\mathbf{C}) &= \log \prod_{i=0}^N \Lambda(c_i) \\ &= \sum_{i=0}^N \log \Lambda(c_i) \\ &= N \log \frac{b_0}{b_1} + \frac{b_1 - b_0}{b_0 b_1} \sum_{i=0}^N |c_i|. \end{aligned} \quad (7)$$

With $N \rightarrow \infty$ the Central Limit Theorem (CLT) gives us

$$\frac{1}{N} \sum_{i=0}^N |c_i| \sim \mathcal{N}(\mu, \frac{\sigma}{\sqrt{N}}), \quad \mu = \mathbf{E}|C|, \quad \sigma = \mathbf{Var}|C|. \quad (8)$$

Under the hypotheses $\mathcal{H}_h, h \in \{0, 1\}$ we have that $|C| \sim \text{Exponential}(b_h^{-1})$ which lead to

$$\sum_{i=0}^N |c_i| \sim \mathcal{N}(N b_h, |N| \frac{b_h}{\sqrt{N}}). \quad (9)$$

By combining (7), (9) we have that under $\mathcal{H}_h, h \in \{0, 1\}$:

$$\Lambda_h(\mathbf{C}) \sim \mathcal{N}(\mu_h, \sigma_h), \quad (10)$$

with

$$\mu_h = N \log \frac{b_0}{b_1} + N b_h \frac{b_1 - b_0}{b_0 b_1} \quad (11)$$

$$\sigma_h = |N \frac{b_1 - b_0}{b_0 b_1}| \frac{b_h}{\sqrt{N}}. \quad (12)$$

Let define

$$\Lambda^*(\mathbf{C}) = \frac{\Lambda_h(\mathbf{C}) - \mu_0}{\sigma_0} \sim \mathcal{N}(\frac{\mu_h - \mu_0}{\sigma_0}, \frac{\sigma_h}{\sigma_0}). \quad (13)$$

The statistic $\Lambda^*(\mathbf{C})$ thus follows a standard normal distribution under \mathcal{H}_0 .

In virtue of the Neyman-Pearson lemma, the most powerful test δ for the problem (4) is the likelihood ratio test:

$$\delta(\mathbf{C}) = \begin{cases} \mathcal{H}_0 & \text{if } \frac{\mathcal{L}_1(\mathbf{C})}{\mathcal{L}_0(\mathbf{C})} < \tau \\ \mathcal{H}_1 & \text{if } \frac{\mathcal{L}_1(\mathbf{C})}{\mathcal{L}_0(\mathbf{C})} \geq \tau. \end{cases} \quad (14)$$

We can define the test δ^*

$$\delta^*(\mathbf{C}) = \begin{cases} \mathcal{H}_0 & \text{if } \Lambda^*(\mathbf{C}) < \tau^* \\ \mathcal{H}_1 & \text{if } \Lambda^*(\mathbf{C}) \geq \tau^*. \end{cases} \quad (15)$$

Which is equivalent as the logarithm is monotonic and the transformation (13) is linear.

One advantage of hypothesis testing is to allow us to guaranty a prescribed false alarm rate α_0 . It is also possible to define the theoretical power of the test as a function of the false alarm rate.

The power β of a test δ is given by the probability α of rejecting the null hypothesis \mathcal{H}_0 under \mathcal{H}_1 :

$$\beta(\delta) \triangleq \mathbb{P}_{\mathcal{H}_1}[\delta(\mathbf{C}) = \mathcal{H}_1] = 1 - \alpha. \quad (16)$$

For our test δ^* the threshold τ^* with respect to the false alarm rate α_0 can be deduced by solving

$$\mathbb{P}_{\mathcal{H}_0}[\Lambda^*(\mathbf{C}) \geq \tau^*] = \alpha_0, \quad (17)$$

then, the power of the test is simply given by

$$\beta(\delta^*) = \mathbb{P}_{\mathcal{H}_1}[\Lambda^*(\mathbf{C}) \geq \tau^*]. \quad (18)$$

B. GENERALISED LIKELIHOOD RATIO TEST

For the test δ^* define in (15), both the parameter b_0 and b_1 are supposed to be known in advance.

If we assume b to mostly depend on the quantisation parameter QP, then b_1 cannot be known in advance. In fact, even though all the coefficients of \mathbf{C} come from macroblocks quantised using the same known quantisation parameter QP₂. The value of the previous quality factor QP₁ is unknown and may even vary for each coefficient.

In practice, in case of a double compression the coefficients \mathbf{C} will not exactly follow a Laplacian distribution as shown by the authors of [24]. We can thus expect b_1 to differ from the expected value of a quantisation parameter QP₂.

In case of a simple compression, we expect \mathbf{C} to follow a Laplacian distribution of scale b_0 . So to verify if a frame is double compressed, we propose to test if the coefficient \mathbf{C} does follow a Laplacian distribution of scale b_0 which depend on the quantisation parameter QP₂ or if it follows a Laplacian distribution of scale $b_1 \neq b_0$ and with b_1 unknown.

This is equivalent to the test proposed in (4) but with the parameter b_1 replaced by the maximum likelihood estimate (38).

We thus have the log-likelihood ratio given by

$$\begin{aligned} \Lambda(\mathbf{C}) &= N \log\left(\frac{b_0}{\hat{b}}\right) + \frac{\hat{b} - b_0}{b_0 \hat{b}} \sum_{i=0}^N |c_i| \\ &= \frac{1}{b_0} \sum_{i=0}^N |c_i| - N \log\left(\frac{1}{N} \sum_{i=0}^N |c_i|\right) + N(\log(b_0) - b_0) \\ &= \frac{N\bar{C}}{b_0} - N \log(\bar{C}) + N(\log(b_0) - b_0) \end{aligned} \quad (19)$$

with

$$\bar{C} = \frac{1}{N} \sum_{i=0}^N |c_i|. \quad (20)$$

Under \mathcal{H}_h we have that

$$\bar{C} \sim \mathcal{N}(b_h, \frac{b_h}{\sqrt{N}}). \quad (21)$$

Let

$$\mathbf{C}^* = \frac{\bar{\mathbf{C}} - b_h}{b_h} \sqrt{N} \sim \mathcal{N}(0, 1). \quad (22)$$

We then have

$$\begin{aligned} \Lambda(\mathbf{C}) &= \frac{N}{b_0} \left(\frac{b_h}{\sqrt{N}} \mathbf{C}^* + b_h \right. \\ &\quad \left. - b_0 \log(b_h) - b_0 \log\left(\frac{1}{\sqrt{N}} \mathbf{C}^* + 1\right) \right) \\ &\quad + N(\log(b_0) - b_0). \end{aligned} \quad (23)$$

The Taylor expansion gives us that

$$\log\left(\frac{1}{\sqrt{N}} \mathbf{C}^* + 1\right) \simeq \frac{1}{\sqrt{N}} \mathbf{C}^* - \frac{1}{2N} (\mathbf{C}^*)^2. \quad (24)$$

Finally, by combining (23) and (24) we have that

$$\begin{aligned} \Lambda(\mathbf{C}) &= \frac{(\mathbf{C}^*)^2}{2} + \sqrt{N} \frac{b_h - b_0}{b_0} \mathbf{C}^* + N \frac{b_h - b_0 \log(b_h)}{b_0} \\ &\quad + N(\log(b_0) - b_0) \\ &= \frac{1}{2} (\mathbf{C}^* + d_h)^2 + a_h \end{aligned} \quad (25)$$

with

$$d_h = \sqrt{N} \frac{b_h - b_0}{b_0} \quad (26)$$

$$a_h = N(\log(b_0) - b_0) + N \frac{b_h - b_0 \log(b_h)}{b_0} - \frac{1}{2} d_h^2. \quad (27)$$

In particular, under \mathcal{H}_0 we will have $d_0 = 0$ and

$$a_0 = N(1 - b_0). \quad (28)$$

Finally

$$\hat{\Lambda}(\mathbf{C}) = 2(\Lambda(\mathbf{C}) - a_0) \sim \chi^2(1). \quad (29)$$

In virtue of the Neyman-Pearson lemma, the most powerful test is the generalised likelihood ratio

$$\hat{\delta}(\mathbf{C}) = \begin{cases} \mathcal{H}_0 & \text{if } \hat{\Lambda}(\mathbf{C}) < \hat{\tau} \\ \mathcal{H}_1 & \text{if } \hat{\Lambda}(\mathbf{C}) \geq \hat{\tau}. \end{cases} \quad (30)$$

As for the test δ , the threshold $\hat{\tau}$ can be deduced by solving

$$\mathbb{P}_{\mathcal{H}_0}[\hat{\Lambda}(\mathbf{C}) \geq \hat{\tau}] = \alpha_0. \quad (31)$$

Finally, the power $\beta(\hat{\delta})$ is given by

$$\beta(\hat{\delta}) = \mathbb{P}_{\mathcal{H}_1}[\hat{\Lambda}(\mathbf{C}) \geq \hat{\tau}]. \quad (32)$$

IV. NUMERICAL EXPERIMENTATION

A. MODEL VALIDATION

To verify the validity of the proposed test (15), we performed a Monte Carlo simulation. We generated 2000 random vectors \mathbf{C} of 1000 elements c , those 2000 vectors were split in half with 1000 vectors following the hypotheses \mathcal{H}_0 and 1000 vectors following the hypotheses \mathcal{H}_1 . We fixed the value of the parameters to $b_0 = 0.8$ and $b_1 = 0.9$.

In Fig. 3, a comparison between the theoretical and the empirical distribution is given under \mathcal{H}_0 and \mathcal{H}_1 . One can see

how the empirical distributions match the theoretical model given in (13).

In Fig. 3, the theoretical and the empirical power $\beta(\hat{\delta})$ of the test are shown. Once again the empirical simulation matches the theoretical model.

We performed the same simulation for the test (29). On Fig. 4 one can see that the empirical distribution once again match with the theoretical model. This is also true for the theoretical and empirical power as one can see in Fig. 4.

The power of the two tests mostly depends on the difference between b_0 and b_1 i.e. $|b_0 - b_1|$. On Fig. 5 we evaluate the theoretical power of the test $\hat{\delta}(\mathbf{C})$ for a fixed false alarm rate $\alpha_0 = 0.05$ and with varying b_0 and b_1 .

We can observe that when we increase $|b_0 - b_1|$, the power increase. This is not surprising as we show in (8) that the maximum likelihood estimation of b becomes normally distributed for a sufficient number of samples. Then naturally if $|b_0 - b_1|$ is much greater than the variance of the maximum likelihood estimators then (19) tends to become perfectly separable between \mathcal{H}_0 and \mathcal{H}_1 .

It is also important to note that as b_0 and b_1 increase, the difference $|b_0 - b_1|$ must increase to maintain the test power. This is also explained by the distribution given in (8). As b increase the variance of the maximum likelihood estimator increase and thus the distance $|b_0 - b_1|$ must also increase to overcome this loss of precision. We will see that this phenomenon affects the performances when the quantisation parameter QP is high.

B. PERFORMANCES ON SIMULATED FRAMES

The test (30) is first evaluated on simulated H.264 frame. This allows us to precisely control both the prediction mode and the quality factor use for each macroblock.

To do so, we randomly selected 500 images from the RAISE [34] dataset. For each image, only a central portion of size 504×504 is kept. Those images have then been converted to grayscale, before being compressed. We reimplemented the H.264 compression as described in [35].

We first compressed every image with a prediction and transformation of size 4 at various QP₁. We then repeat this process for a prediction and transformation of size 8.

Then each of these compressed images is recompressed with both prediction mode and various QP₂. For a given image predicted with *PredX* and compressed with a quality factor QP₁ we thus have two scenarios of interest after the recompression

- 1) The frame is predicted with *PredY* and $Y \neq X$
- 2) The frame is compressed at QP₂ \neq QP₁

We will first focus on the case where *PredY* = *PredX* to evaluate the impact of the quantisation parameter on the detection performances.

Then we will study the case where *PredY* \neq *PredX* and various QP to evaluate the impact of the prediction mode on the prediction.

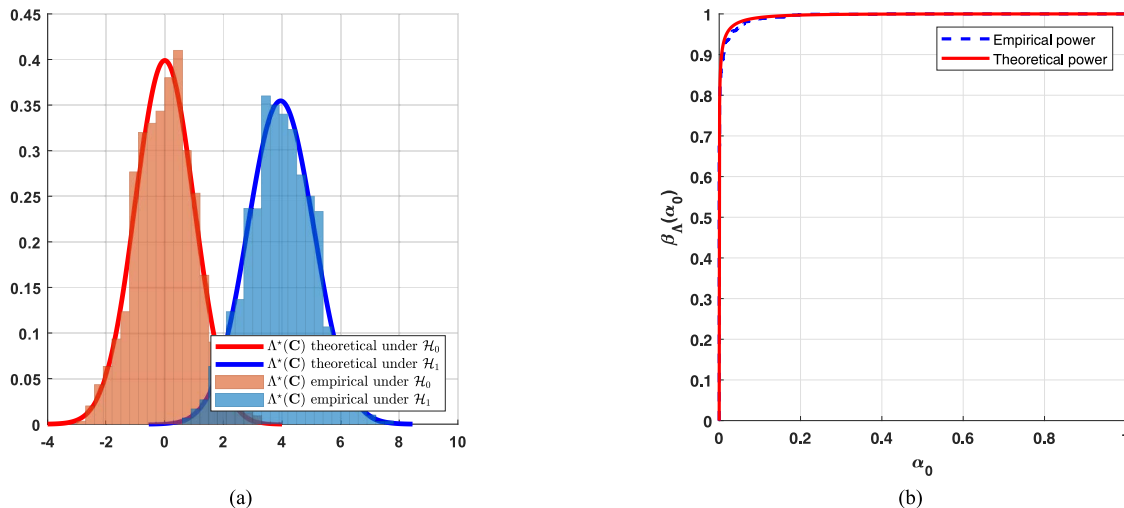


FIGURE 3. From left to right: Theoretical and empirical distribution under \mathcal{H}_0 and \mathcal{H}_1 , theoretical and empirical power under \mathcal{H}_0 and \mathcal{H}_1 with $b_0 = 0.8$ and $b_1 = 0.9$.

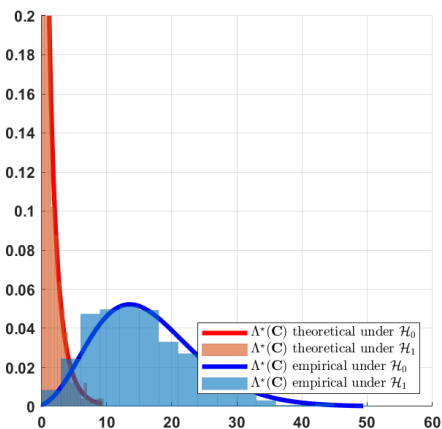


FIGURE 4. Theoretical and empirical distribution under \mathcal{H}_0 and \mathcal{H}_1 with $b_0 = 0.8$ and $b_1 = 0.9$.

In each case, the parameter b_0 is estimated as the median of all the maximum likelihood estimations \hat{b} observed for images simply compressed by a quantisation parameter QP_2 .

1) RECOMPRESSION WITH THE SAME PREDICTION MODE

The generalised log-likelihood ratio given in (29) is calculated for each image compressed at QP_2 and images first compressed at QP_1 and then recompressed at QP_2 .

The empirical Area Under the Curve (AUC) was computed in order to obtain an overview of the detection performance for various QP_1 and QP_2 . The results are also given for different coefficient i.e. $C_{1,1}$, $C_{1,4}$ and $C_{4,4}$ In Fig. 6 the first and second predictions were made using Pred4.

Whatever the coefficient used, the first observation to be made is that for $QP_1 = QP_2$ the detection is completely random (i.e. an AUC of 0.5). This is expected as a recompression at the same quantisation parameter in H.264 has no impact on the DCT coefficients. In a practical scenario where the rate control mechanism is not constant, some methods [21]–[23]

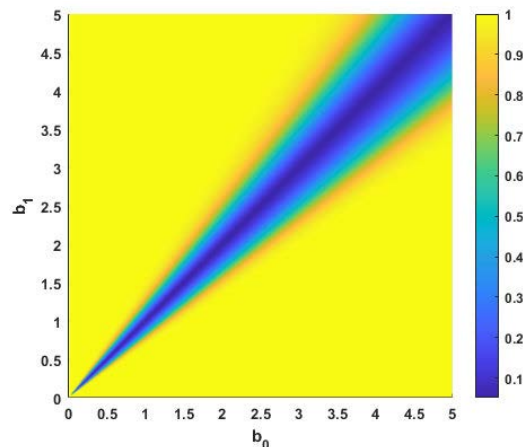


FIGURE 5. Theoretical power with $\alpha_0 = 0.05$ for varying b_0 and b_1 .

have been able to detect the double compression when the targeted quantisation parameters were equal. This suggests that the rate control mechanism introduces enough variation such that $QP_1 = QP_2$ is unlikely.

It is also important to remark that the detection is not possible for $QP_2 > QP_1$. In Fig. 6, this corresponds to the upper-left part. With $QP_2 > QP_1$ the second compression is stronger than the first compression and thus erase any traces of the first compression.

The detection is possible only for $QP_2 < QP_1$. In particular, the performance increase with $|QP_2 - QP_1|$. We also notice that for every coefficient the detection performances are satisfactory for $|QP_2 - QP_1| > 10$.

Finally, the choice of the coefficient has a strong influence on the detection performance. We can see how the performances for lower values of QP_2 are worst for the DC coefficient $C_{1,1}$ than for the other two. The performances also increase between $C_{1,4}$ and $C_{4,4}$.

To understand this phenomenon, it is important to recall two things. First, the value of b_0 depends mostly on the quantisation parameter. And secondly, the compression becomes increasingly stronger for coefficients further away from the DC coefficient. This implies that b_0 decrease as QP_2 increase. But also that for a fixed value of QP_2 , b_0 also decrease as the studied coefficient gets farther from the DC coefficient. As shown in Fig. 5, the performances increase when b_0 and b_1 are lower.

For lower value of QP_2 , it is then natural to observe better performance for coefficients farther away from the DC coefficient. But this is only true as long as there exists a sufficiently large number of non-zero coefficients. In fact, one can notice that for $QP_2 > 35$ the detection becomes random for the coefficient $C_{4,4}$ whereas for the DC coefficient we still observe an AUC of about 0.8.

On Fig. 7, the same simulation has been performed but with a first and second prediction using Pred8. It can be seen that the results are mostly similar. For the 8×8 transform, the results are slightly worse than the 4×4 transform when both QP_1 and QP_2 are lower.

2) RECOMPRESSION WITH A DIFFERENT PREDICTION MODE

In the previous section, we evaluated the performances in the case where the first and second predictions were the same. As we mentioned, it is also possible to observe Macroblocks for which the first and second prediction will not be the same.

On Fig. 8, we can see the result of a first prediction with Pred8 and a second prediction with Pred4. In this case, b_0 is estimated from simply compressed images with Pred4 and QP_2 . We can see that the performances are lower but overall similar. The double compression can only be detected for $QP_1 > QP_2$.

On Fig. 9, we observe similar result when the first prediction is Pred4 followed by Pred8. Interestingly, we can see that the detection is somewhat possible with $QP_1 \ll QP_2$ for the coefficient $C_{1,1}$ but the performances are really low.

We can observe that the performance drop is more important in the case of Pred4 followed by Pred8. This can be explained by the fact that Pred8 is less accurate than Pred4, we will thus have a residual that might not be affected by the first compression. In fact, we can see that unless the QP_1 was extremely high (i.e. really strong compression), the detection is pretty much impossible.

For Pred8 followed by Pred4 the performances are slightly better. This time the second prediction is more accurate than the first one. One block of size 8×8 is now predicted using 4 blocks of size 4×4 . Because of the first compression, every lower right 4×4 block will appear as if it was more compressed than every upper left 4×4 block. This will create a discrepancy between the Pred4 block which affects the estimation of \hat{b} .

Overall, the performances decrease in this scenario. As we explained, H.264 apply the transformation to the residual. When the first and second prediction match, it is likely that the H.264 will choose the same prediction mode. This leads

to the same residual data compressed twice. When the prediction size mismatch, this does not hold. The block will be predicted on a different scale and thus the residual will not be the same. The performances are better when the second prediction is more accurate than the first one.

C. PERFORMANCES ON SMARTPHONE VIDEOS

In this section we evaluate the performances on a dataset of real videos. The dataset contains 45 videos taken with 4 different smartphones. All videos are in full HD i.e. 1920×1080 pixels. All videos were compressed by the various smartphones H.264 encoders using the high profile. Each video thus contains both 4×4 and 8×8 macroblocks. The videos are then recompressed using the x264 encoder. We recompressed the video using the CRF rate control with different quality factors. On Fig. 10, the distribution of the original quantisation parameters for every video is given. The average QP across all videos is around 20. To recall, a quantisation parameter of 23 is considered as visually lossless. We can reasonably consider that the videos were originally compressed with a rate control aiming at maintaining the QP around 23.

We will evaluate three different scenarios. In the first scenario we will set $QP = 15$, so that macroblocks will tend to be recompressed at lower quality factor than the original. In the second scenario, QP is set two 20 so that the second compression is close to the first one. Finally, we evaluate the performances for $QP = 25$ and $QP = 30$ for which macroblocks will tend to be recompressed at a higher quantisation parameter.

Unlike the previous evaluation on simulated frames, we cannot predict the primary prediction mode nor the primary quantisation parameter. We expect the performance to be worst when the second compression is set to $QP = 20$ and $QP = 25$ as it is then less likely that a macroblock will be recompressed at a lower quantisation parameter. In every scenario, b_0 is estimated as the median of the observed \hat{b} for the original video. For the theoretical results b_1 is also estimated as the median of the observed \hat{b} for the recompressed videos.

On Fig. 11, the results are given for the coefficients $C_{1,1}^{4,20}$. Both the empirical power and the theoretical power are given. Firstly, we can see that the recompression does affect the value \hat{b} . The difference between b_0 and b_1 is big enough so that the theoretical power is almost perfect. In practice we can observe a significant loss in power. As observed in Fig. 2, even though the values \hat{b} seem to vary around some b_0 . It is obvious that the assumption that $C \sim \text{Laplace}(0, b_0)$ does not fully reflect the real world and that b_0 is not only defined by the quantisation parameter and the coefficient position. This variance around the hypothetical value b_0 translates into a loss of power in practice. Nonetheless, we observe good detection performances in that scenario which validate the approach to real-world examples.

On Fig. 12, the results are given for $C_{1,1}^{4,20}$. In that scenario, the second compression approximately matches the

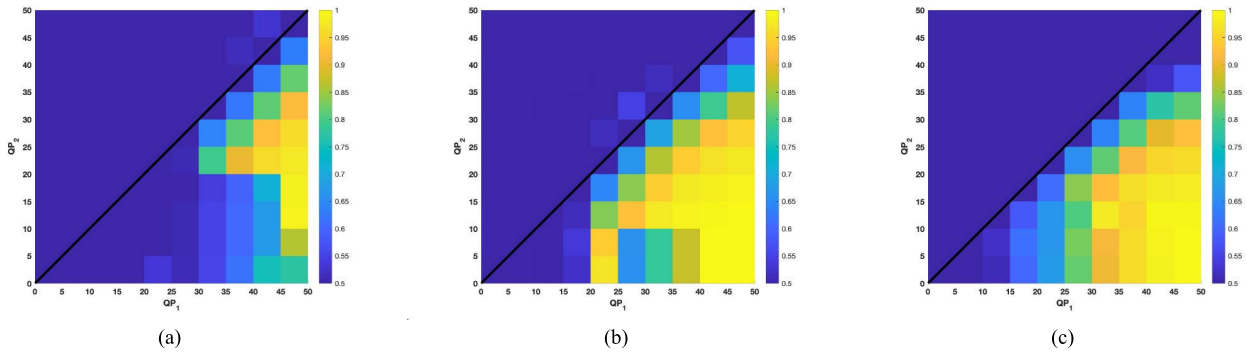


FIGURE 6. Empirical AUC for the coefficient $C_{1,1}$, $C_{1,4}$ and $C_{4,4}$ predicted with Pred4 and recompressed with Pred4 with respect to $|QP_2 - QP_1|$.

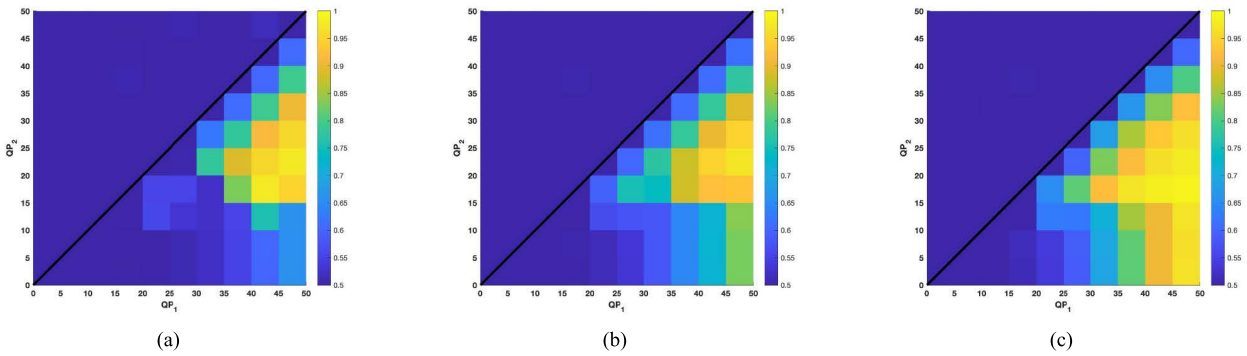


FIGURE 7. Empirical AUC for the coefficient $C_{1,1}$, $C_{1,4}$ and $C_{4,4}$ predicted with Pred8 and recompressed with Pred8 with respect to $|QP_2 - QP_1|$.

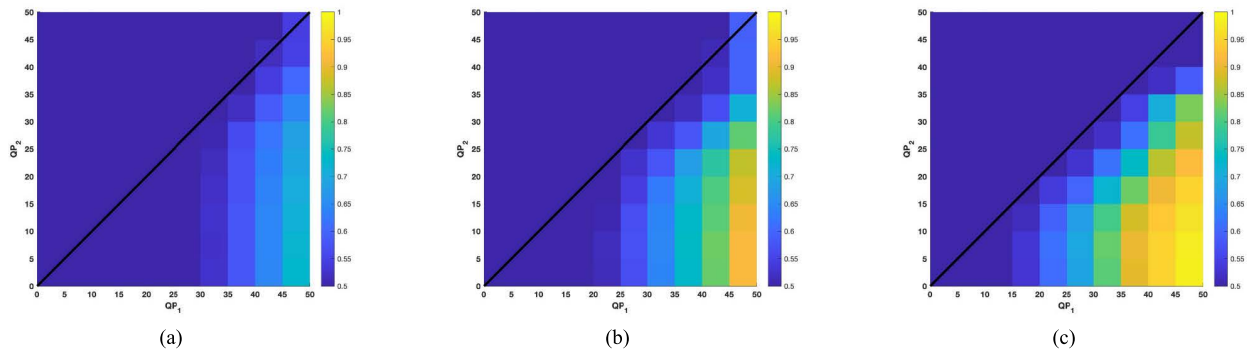


FIGURE 8. Empirical AUC for the coefficient $C_{1,1}$, $C_{1,4}$ and $C_{4,4}$ predicted with Pred8 and recompressed with Pred4 with respect to $|QP_2 - QP_1|$.

first compression. As a result, it is more likely that a macroblock will be recompressed at the same quantisation parameter or above as the distribution of QP overlaps. We indeed observe both lower theoretical and empirical performances as b_0 and b_1 are closer. Once again we observe a loss in power between the theoretical model and the empirical evaluation.

Finally, on Fig. 13 the results are given for $C_{1,1}^{4,23}$. This time the second compression is set to $QP = 25$. In this scenario, it is more likely that a macroblock will be recompressed at a higher quantisation parameter so we expect the performances to be lower. We can see in Fig. 13 that the performances are

indeed slightly lower than for the first scenario (i.e. 11) but are still reasonably good.

Those results are really encouraging as they show that even though it is not possible to detect a double compression at the same or higher quantisation parameter. The mechanism of rate control in H.264 introduce enough perturbation to obtain good detection performances. It is important to recall that in Fig. 11, 12 and 13 only a single QP and a single DCT sub-band are used to perform the detection. In practice the test (30) can be performed for each QP and each sub-band of a given video. We expect that lower values of QP will yield the better performances as they have more chances of being

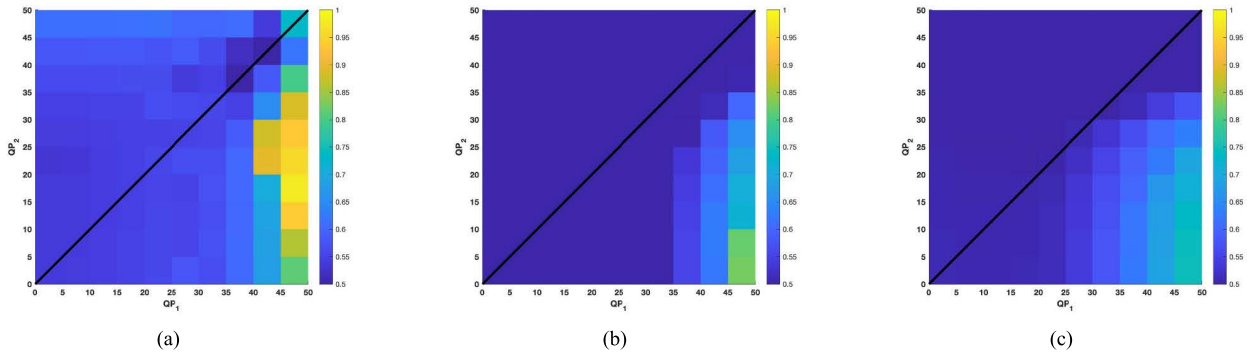


FIGURE 9. Empirical AUC for the coefficient $C_{1,1}$, $C_{1,4}$ and $C_{4,4}$ predicted with Pred4 and recompressed with Pred8 with respect to $|QP_2 - QP_1|$.

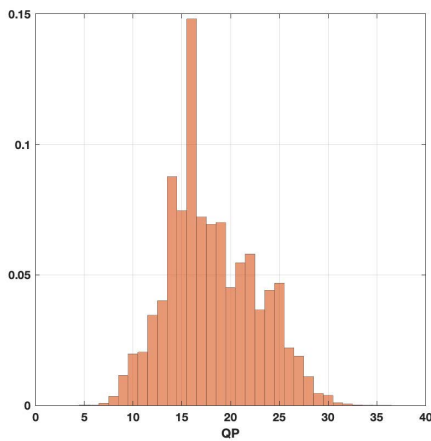


FIGURE 10. Distribution of QP across the 45 videos.

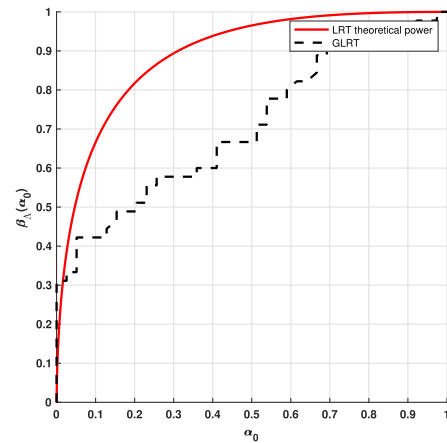


FIGURE 12. Empirical and theoretical power for the coefficients $C_{1,1}^{4,20}$ with $QP_2 = 20$.

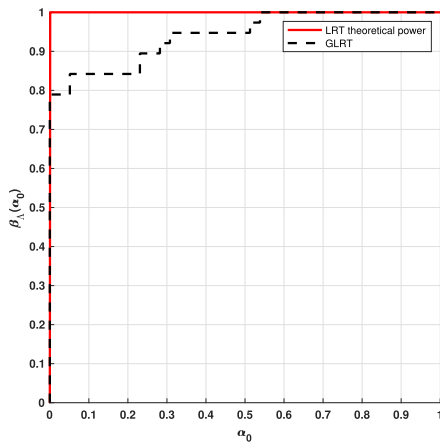


FIGURE 11. Empirical and theoretical power for the coefficients $C_{1,1}^{4,20}$ with $QP_2 = 15$.

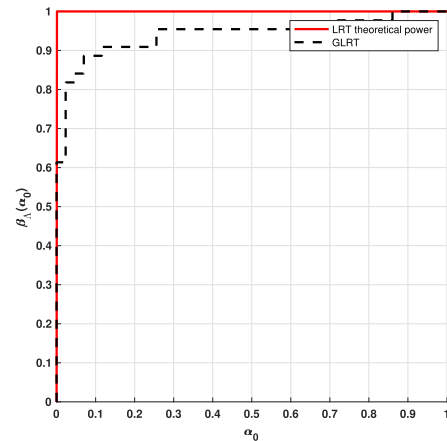


FIGURE 13. Empirical and theoretical power for the coefficients $C_{1,1}^{4,23}$ with $QP_2 = 25$.

recompressed at a lower quantisation parameter. In Table. 1, we performed a naive combination of the subbands $C_{1,1}^4$ and $C_{1,1}^8$ by taking the average value of the test (30) for each QP present in the video. We can see how this simple fusion greatly improves the performances.

V. COMPARISON TO STATE-OF-THE-ART METHODS

Finally, we evaluate our method against two state-of-the-art methods. For the first method, we implemented the algorithm described in [24] which is based on the DCT coefficients like our approach. They propose to extract non zero coefficients of every I-frame. They then extract all the

TABLE 1. AUC obtained on the smartphone dataset using the naive score fusion for various QP_2 .

QP_2	15	20	25	30
AUC	0.9995	0.9274	0.9723	0.9990

TABLE 2. Comparison to state-of-the-art methods.

Quantisation parameter		Method		
QP_1	QP_2	Proposed	SVM-DCT [24]	G-VPF [16]
25	18	1	0.9781	0.9580
25	20	1	0.7503	0.9170
25	23	0.9512	0.6629	0.8926
23	18	0.9990	0.8241	0.9561
23	20	0.9863	0.7575	0.9111
20	18	0.9570	0.5078	0.9268
23	25	0.4453	0.4724	0.8936
25	30	0.3028	0.3632	0.7432

coefficients in the range $[-10; 10]$ excluding 0. Finally, they compute the empirical probability of a coefficient being equal to $-10, -9, \dots, 9, 10$ to create a feature vector of dimension 20. A SVM is then used to perform the classification. For the second method, we used the available implementation of [16]. They study the distribution of macroblocks types to both estimate the GOP size of the first compression and to detect a possible double compression.

Because the method [24] requires a training dataset and the method [16] requires the first compression GOP size to be fixed, we constructed two datasets. A first dataset of 11 HD videos from [36] which we used to train the method [24] and also to get an estimate of the parameter b_0 for each QP for our method. And a second dataset of 31 CIF videos from [36].

For both dataset we compressed the video using ffmpeg and the x264 encoder with the following compression parameters. We fixed the GOP size to 9 for the first compression and a GOP size of 25 for the second compression. We used the CRF for the rate control mechanism with $QP \in \{18, 20, 23, 25, 30\}$ for both compressions. We kept QP around 23 which correspond to a visually lossless compression and which is a common value for this parameter. Finally, we did not specify any parameters regarding the use of B-Frames.

In the previous section, we used a single DCT subband and a single QP to perform the detection. Here we perform a naive combination of the subbands $C_{1,1}^4$ and $C_{1,1}^8$ by taking the average value of the test (30) for each QP present in the video.

The detection results are given in terms of Area Under the Curve (AUC) in Table 2 for various QP_1 and QP_2 . In the first part of the Table, we can see the results for $QP_2 < QP_1$. In such case, we see that our method outperform the state-of-the-art algorithms. In the second part of the Table, we show two examples where $QP_2 > QP_1$. We know that for a fixed QP the detection is theoretically not possible for our method based on the DCT coefficients. But for our dataset on smartphone video we saw that the rate control introduced enough perturbation to perform the detection. Here we see that the

perturbation does not overcome this limitation which could be explained by the implementation of the H.264 encoder. If the variance around the targeted QP value is lower, then it is more likely that we will have $QP_2 > QP_1$ for an individual macroblock. Similarly the method [24] fail in that scenario as it is based on the DCT coefficients. In contrary, G-VPF [16] suffer less in that scenario. The authors of [16] also notice that the performances eventually collapse when $QP_1 \ll QP_2$. Here we see that for $QP_1 = 25$ and $QP_2 = 30$, the AUC of G-VPF drops to 0.7432.

VI. CONCLUSION

In this article we proposed a method to detect a double H.264 video compression detection algorithm based on an analysis of the DCT coefficient. We showed that the DCT coefficients can be roughly approximated by a zero mean Laplacian distribution and that the scale parameter is dependent on the quantisation parameter. We thus proposed a statistical test to determine whether or not the observed coefficients follow a Laplacian distribution with a scale parameter b_0 based on the observed QP.

We showed that the detection was only possible when the second quantisation parameter was lower than the first one. Even though this seems like a strong limitation, we showed on real example that in practice this might not be as problematic thanks to the rate control mechanism of H.264. Indeed, in H.264 a single frame can be encoded using many different quantisation parameters. Our experimental evaluation showed that this behaviour introduces enough variation in the difference between the first and second quantisation parameters to make the detection possible.

In future works, many points could be addressed to improve the results of the proposed method. In [33], it was shown that the DCT coefficients for JPEG images could only be assumed i.i.d after suppressing the image content (i.e. the image expectation). Unlike JPEG images, H.264 compression includes a prediction stage prior to the DCT transformation and quantification. In this article, we considered this prediction as a rough estimation of the image expectation and thus considered the DCT coefficients to be i.i.d and following a Laplacian distribution. But we can see in Fig. 2 that the estimated scale b has a non-negligible variance and on Fig. 1 is not perfectly accurate in particular around zero. This suggests that the H.264 prediction may not be considered as a good approximation of the image prediction. In fact, it is not designed to estimate the expectation but rather to estimate the exact pixel values (noise included).

A first perspective to improve the results of the proposed method would then be to proceed as in [33] by first decoding the H.264 stream in order to compute the expectation and remove it prior to the estimation of the scale parameter.

Another perspective would be to propose a more elaborate model of the DCT coefficients as in [31] by adding the impact of the prediction stage prior to the transformation and quantification.

In this article, we proposed a statistical test for a single DCT coefficient at certain quantisation parameters. In practice, it would be interesting to design a method using every coefficient at every QP to maximise the detection performance. A last perspective is to study the application of our method to other video compression algorithms. Here we focused on H.264 compression only but video compression algorithms are often quite similar. For instance, the successor of H.264 (namely H.265) mainly follows the same compression scheme. Similarly VP9 and its successor AV1 also uses a DCT transformation on residual blocks. Moreover, the two latest encoders (i.e. AV1 and H.265) can both be used to perform image compression. This convergence of technologies is a great opportunity to develop forensic algorithms for both images and videos.

APPENDIX MAXIMUM LIKELIHOOD ESTIMATOR

We suppose that $\mathbf{C} \sim \text{Laplace}(0, b)$. We can then define the likelihood function of a given parameter b as

$$\mathcal{L}_b(c_i) = \frac{1}{2b} \exp\left(\frac{-|c_i|}{b}\right). \quad (33)$$

For \mathbf{C} we then have

$$\begin{aligned} \mathcal{L}_b(\mathbf{C}) &= \prod_{i=0}^N \mathcal{L}_b(c_i) \\ &= \frac{1}{(2b)^N} \exp\left(\frac{-\sum_{i=0}^N |c_i|}{b}\right). \end{aligned} \quad (34)$$

The log-likelihood function for \mathbf{C} is finally given by

$$\begin{aligned} \ell_b(\mathbf{C}) &= \log(\mathcal{L}_b(\mathbf{C})) \\ &= -N \log(2b) + \frac{-\sum_{i=0}^N |c_i|}{b}. \end{aligned} \quad (35)$$

The maximum likelihood estimate is thus give for

$$\frac{\partial \ell_b(\mathbf{C})}{\partial b} = 0 \quad (36)$$

with

$$\frac{\partial \ell_b(\mathbf{C})}{\partial b} = \frac{-N}{b} + \frac{\sum_{i=0}^N |c_i|}{b^2}. \quad (37)$$

We finally derive the maximum likelihood estimator \hat{b} as

$$\hat{b} = \frac{\sum_{i=0}^N |c_i|}{N}. \quad (38)$$

REFERENCES

- [1] ANSSI. (Mar. 2021). *Publication du Référentiel d'Exigences Applicables aux Prestataires de Vérification d'Identité à Distance (PVID)*. Accessed: Sep. 27, 2020. [Online]. Available: https://www.ssi.gouv.fr/uploads/2021/08/anssi-requirements_rule_set-pvid-v%1.1.pdf
- [2] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile GPUs," 2019, *arXiv:1907.06724*.
- [3] Google. (2020). *Face Mesh Mediapipe*. Accessed: Sep. 27, 2020. [Online]. Available: https://google.github.io/mediapipe/solutions/face_mesh.html
- [4] A. Roach, *WebRTC Video Processing and Codec Requirements*, document RFC 7742, Mar. 2016.
- [5] W. Chen and Y. Q. Shi, "Detection of double MPEG compression based on first digit statistics," in *Proc. Int. Workshop Digit. Watermarking*, Berlin, Germany: Springer, 2008, pp. 16–30.
- [6] T. Sun, W. Wang, and X. Jiang, "Exposing video forgeries by detecting MPEG double compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 1389–1392.
- [7] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *Proc. 8th workshop Multimedia Secur. (MM&Sec)*, 2006, pp. 37–47.
- [8] Y. Su and J. Xu, "Detection of double-compression in MPEG-2 videos," in *Proc. 2nd Int. Workshop Intell. Syst. Appl.*, May 2010, pp. 1–4.
- [9] 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," *J. Vis. Commun. Image Represent.*, vol. 48, pp. 149–158, Oct. 2017.
- [10] Q. Xu, X. Jiang, T. Sun, P. He, S. Wang, and B. Li, "Relocated I-frames detection in H.264 double compressed videos based on genetic-CNN," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2018, pp. 710–716.
- [11] S.-H. Nam, J. Park, D. Kim, I.-J. Yu, T.-Y. Kim, and H.-K. Lee, "Two-stream network for detecting double compression of H.264 videos," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 111–115.
- [12] Q. Li, S. Chen, S. Tan, B. Li, and J. Huang, "One-class double compression detection of advanced videos based on simple Gaussian distribution model," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 29, 2021, doi: [10.1109/TCSVT.2021.3069254](https://doi.org/10.1109/TCSVT.2021.3069254).
- [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, p. 313, Dec. 2017.
- [14] H. Yao, R. Ni, and Y. Zhao, "Double compression detection for H.264 videos with adaptive GOP structure," *Multimedia Tools Appl.*, vol. 79, nos. 9–10, pp. 5789–5806, Mar. 2020.
- [15] D. Vazquez-Padin, M. Fontani, T. Bianchi, P. Comesana, A. Piva, and M. Barni, "Detection of video double encoding with GOP size estimation," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2012, pp. 151–156.
- [16] D. Vazquez-Padin, M. Fontani, D. Shullani, F. Perez-Gonzalez, A. Piva, and M. Barni, "Video integrity verification and GOP size estimation via generalized variation of prediction footprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1815–1830, 2020.
- [17] 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. Int. Workshop Digit. Watermarking*. Springer, 2016, pp. 61–74.
- [18] J. Zheng, T. Sun, X. Jiang, and P. He, "Double H.264 compression detection scheme based on prediction residual of background regions," in *Proc. Int. Conf. Intell. Comput.* Springer, 2017, pp. 471–482.
- [19] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Codec and GOP identification in double compressed videos," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2298–2310, May 2016.
- [20] P. He, X. Jiang, T. Sun, and S. Wang, "Double compression detection based on local motion vector field analysis in static-background videos," *J. Vis. Commun. Image Represent.*, vol. 35, pp. 55–66, Feb. 2016.
- [21] Z. Zhang, J. Hou, Y. Zhang, J. Ye, and Y. Shi, "Detecting multiple H.264/AVC compressions with the same quantisation parameters," *IET Inf. Secur.*, vol. 11, no. 3, pp. 152–158, Aug. 2016.
- [22] 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. Inf. Forensics Security*, vol. 13, no. 1, pp. 170–185, Jan. 2018.
- [23] S. Ting, S. Tanfeng, and J. Xinghao, "Detection of double compression with the same quantization parameter based on dual encoding parameter model," *J. Shanghai Jiaotong Univ.*, vol. 53, no. 3, p. 334, 2019.
- [24] D. Liao, R. Yang, H. Liu, J. Li, and J. Huang, "Double H.264/AVC compression detection using quantized nonzero ac coefficients," *Proc. SPIE*, vol. 7880, Feb. 2011, Art. no. 78800Q.
- [25] X. Jiang, W. Wang, T. Sun, Y. Q. Shi, and S. Wang, "Detection of double compression in MPEG-4 videos based on Markov statistics," *IEEE Signal Process. Lett.*, vol. 20, no. 5, pp. 447–450, May 2013.
- [26] I. E. Richardson, *The H. 264 Advanced Video Compression Standard*. Hoboken, NJ, USA: Wiley, 2011.
- [27] W. K. Pratt, *Digital Image Processing*. New York, NY, USA: Wiley, 1991.

- [28] R. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, no. 6, pp. 835–839, Jun. 1983.
- [29] J. D. Eggerton and M. D. Srinath, "Statistical distributions of image DCT coefficients," *Comput. Electr. Eng.*, vol. 12, nos. 3–4, pp. 137–145, Jan. 1986.
- [30] T. Eude, R. Grisel, H. Cherifi, and R. Debrie, "On the distribution of the DCT coefficients," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 1994, p. V/365.
- [31] T. H. Thai, R. Cogranne, and F. Reiraint, "Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1980–1993, May 2014.
- [32] Y. Altunbasak and N. Kamaci, "An analysis of the DCT coefficient distribution with the H.264 video coder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, p. 177.
- [33] T. Qiao, F. Reiraint, R. Cogranne, and C. Zitzmann, "Steganalysis of JSteg algorithm using hypothesis testing theory," *EURASIP J. Inf. Secur.*, vol. 2015, no. 1, pp. 1–16, 2015.
- [34] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proc. 6th ACM Multimedia Syst. Conf.*, Mar. 2015, pp. 219–224.
- [35] *Information Technology—Coding of Audio-Visual Objects—Part 10: Advanced Video Coding*, Standard ISO 14496-10:2020, International Organization for Standardization, Geneva, Switzerland, 2020.
- [36] *Derf's Collection*. Accessed: Aug. 17, 2021. [Online]. Available: <https://media.xiph.org/video/derf/>



GAËL MAHFOUDI received the double Engineering Diploma/M.S. degree in information system security from the University of Technology of Troyes, Troyes, France, in 2018, where he is currently pursuing the Ph.D. degree in image processing. His research interests include image processing, hypothesis testing theory, and digital image forensics.



FLORENT REIRAINT received the engineering degree in computer science from the University of Technology of Compiègne (UTC) in 1993, the M.Sc. degree in applied mathematics from the National School of Computer Science and Applied Mathematics of Grenoble (ENSIMAG) in 1994, and the Ph.D. degree in applied mathematics from the National Institute of Applied Sciences (INSA), Lyon, France, in 1998. He is currently a Full Professor at the University of Technology of Troyes.

His research interests include image modeling, statistical image processing, hypothesis testing theory, and anomaly detection and localization, with a main application to digital image forensics.



FRÉDÉRIC MORAIN-NICOLIER received the M.Sc. degree in applied physics and the Ph.D. degree in image processing from the University of Bourgogne, France, in 2000. He is currently a Full Professor at the University of Reims Champagne-Ardenne, where he is integrated at the CREsTIC Laboratory. His research interests include image processing and analysis, similarity measures and dissimilarity detection, and biomedical imaging.



MARC MICHEL PIC received the engineering degree from Centrale Marseille, in 1991, and the Ph.D. degree in computer science from the University of Sorbonne, in 1996. He was a Researcher at CEA for ten years, founded a company to recognize audiovisual content, in 2002, then sold it to SURYS/IN-Groupe, in 2009, where he is currently managing Digital Solutions. He has taught at Ecole Polytechnique and Paris Dauphine. He has 20 patents and numerous publications. He is an Administrator of think-tank Cyberlex.

• • •