Accepted in IEEE International Conference on Signal Processing and Communication Systems (ICSPCS 2018), 2018

The Impact of H.264 Non-desynchronizing Bits on Visual Quality and its Application to Robust Video Decoding

Firouzeh Golaghazadeh
Department of Software and IT Engineering
École de technologie supérieure
Université du Québec
Montreal, Canada
Firouzeh.Golaghazadeh.1@etsmtl.net

Stéphane Coulombe
Department of Software and IT Engineering
École de technologie supérieure
Université du Québec
Montreal, Canada
Stephane.Coulombe@etsmtl.ca

François-Xavier Coudoux
UPHF, CNRS, Univ. Lille, YNCREA
Centrale Lille, UMR 8520 IEMN
DOAE, F-59313 Valenciennes, France
Francois-Xavier.Coudoux@uphf.fr

Patrick Corlay
UPHF, CNRS, Univ. Lille, YNCREA,
Centrale Lille, UMR 8520 IEMN
DOAE, F-59313 Valenciennes, France
Patrick.Corlay@uphf.fr

***Abstract*—In this paper, the concept of non-desynchronizing bits (NDBs) is defined in the context of H.264 video as a bit whose inversion does not cause desynchronization at the bitstream level or change the number of decoded macroblocks. We established that, on the whole, NDBs make up about a third (about 30%) of a bitstream, and that their flipping effect on visual quality is mostly insignificant. In most cases (90%), the PSNR value obtained when modifying an NDB is very close to the intact value. The performance of the proposed non-desync-based decoding framework, which retains a corrupted packet, under the condition of not causing desynchronization, has been compared to the JM-FC and a state-of-the-art concealment approach using the STBMA approach, and on average, respectively, provides 3.5 dB and 1.42 dB gain over them.**

***Index Terms*—video transmission, H.264, syntax elements, non-desynchronizing bit, concealment**

I. Introduction

Although high efficiency video coding (HEVC) [1] provides better coding efficiency, H.264 [2] is still the most widely used standard because it provides a good compromise between coding efficiency and computational complexity and is widely deployed in decoders [3]. Because of the high compression performance of video coding standards (such as H.264/MPEG advanced video coding (AVC)), the compressed stream is more vulnerable to transmission errors, and an error can propagate from one frame to consecutive ones and lead to persistent visual artifacts [4]. Different error concealment (EC) approaches have been proposed in the literature [5], which try to conceal lost areas by exploiting the inherent correlations between spatially [6] or temporally [7] adjacent pixels, or both [8], [9].

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant.

Packets partially damaged due to transmission errors may contain valuable information that can be used to enhance the visual quality of the reconstructed video [10], [11]. A standardized transport protocol, lightweight user datagram protocol (UDP-Lite), allows partially damaged packets to be delivered to the application layer instead of having them discarded upon reception [12]. However, the application layer should be responsible for deciding whether to retain the corrupted packets or throw them away.

In this work, we are more focused on utilizing the corrupted packets to demonstrate when it is most beneficial to keep them instead of discarding them and propose a new robust decoding approach. In the literature, much of the effort on the syntax analysis has been dedicated to error detection capability of the syntaxes. In [10], [13], the authors illustrate the exploitation of syntax elements in corrupted packets to detect non-valid syntaxes or semantic errors and eventually save unharmed macroblocks (MBs) in the corrupted packet and apply EC on the rest of the MBs. But as we know, the error detection location is not always the same as the error occurrence. This distance, as presented in [10] for low resolution QCIF sequences, is on average 15 MBs for an inter frame. Due to the predictive nature of video coding, the wrongly decoded MBs will drastically degrade the visual quality of the subsequent frames.

In other works such as [14] and [15], the authors examined the effect of an isolated error on visual quality by applying the error randomly on different parts of the coded packet. For instance, the impact of a random bit error, in the payload and

Paper presented at 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), Cairns, Australia, 17-19 Dec. 2018
https://doi.org/10.1109/ICSPCS.2018.8631785

header, on the visual quality is investigated separately in [14]. In [15], the authors examined the effect of an isolated error in each syntax element of H.264 on visual quality. By simulation on only low resolution QCIF sequences, they presented which syntaxes are less sensitive to an isolated error. They also showed that if an error hits these syntax elements, decoding those corrupted packets leads to better quality than using slice level concealment. But as known, the error may create another valid (but wrong) codeword and the effect of such error may appear in the following syntaxes or MBs. Therefore, this is not realistic since, in an erroneous packet, it is more likely to not detect the error in the exact position. The sensitivity of a syntax to errors depends on how the syntax has been coded and how the following syntaxes depend on it. For example, one isolated bit error on a specific part of an Exponential Golomb Code (EGC) syntax could cause direct desynchronization, which makes it highly sensitive to errors, while the other bits of the same syntax do not cause any desynchronization effect. Unlike [15], we look for the least sensitive bits of each syntax element, the bits that errors on them will not cause any desynchronization. In [16], the authors identified some of the bits for the ciphering in their encoding stage, which has little to no influence on the decoding, for generating different replacement codewords to perform encipherment within the compression.

Our contributions in this work are as follows: we identify and analyze the most common non-desynchronizing bits (NDBs) of each syntax element of H.264 context-adaptive variable-length coding (CAVLC) sequences. We present the percentage of NDBs in typical video bitstreams; we examine the effect of individual errors on the NDBs to confirm that they have an insignificant impact on visual quality. With all the new knowledge we acquire on NDBs, we propose a robust decoding approach that retains the corrupted packets for which only NDBs are erroneous. This new complementary approach can work in conjunction with EC approaches (especially at low error rates), and it could be useful in broadcast and low-latency applications, such as the Internet of Things (IoT), transport and remote control of devices [17]–[20].

This paper is organized as follows: Section II describes the NDBs of H.264 coded sequences. The frequency of the NDBs and their effect on visual quality are presented in Section III. The proposed framework based on keeping corrupted packets, as well as the simulation results, are presented in Section IV and Section V, respectively, followed by concluding remarks in Section VI.

II. Syntax Element Analysis

In this section, we look at the syntax elements in H.264 baseline profile to identify their NDBs. A bit is identified as an NDB if its inversion satisfies two conditions:

- It does not cause desynchronization of the bitstream when decoding.
- It does not change the number of decoded MBs.

Note that we assume that each slice normally contains one row of MBs.

TABLE I: Examples of EGC mapping values to UGC and SGC. Bold bits are the possible NDBs of EGC.

Bit Pattern	UGC	SGC	Bit Pattern	UGC	SGC
1	0	0	001**00**	3	2
01**0**	1	1	001**01**	4	-2
01**1**	2	-1	001**10**	5	3

A. NDBs in H.264 syntax elements

The main syntax elements in H.264 are EGCs, which start with $N(\geq 0)$ zero bits as zero-prefix followed by a bit 1 (called here the middle bit), and then N bits of information as the INFO part. The complete codeword has a length of $2N+1$ bits as shown:

$$\underbrace{0\ 0\ ...\ 0}_{\text{zero-prefix}}1\ \underbrace{\mathrm{X}_1\ \mathrm{X}_2\ ...\ \mathrm{X}_N}_{\text{INFO}},\quad \mathrm{X}_i \in \{0,1\}, \forall i \in [1,N] \tag{1}$$

The value of an EGC, based on its type being signed-EGC (SGC) or unsigned-EGC (UGC), is decoded as follows [2]:

1) Count the number of zeros (N) until the first bit 1.
2) Read N bits after the first bit 1 (INFO).
3) $\text{UGC} = 2^N + \text{INFO} - 1$; $\quad \text{SGC} = (-1)^{UGC+1} \times \lfloor \frac{\text{UGC}}{2} \rfloor$

It is obvious that if an error occurs in the zero-prefix part of an EGC or changes the middle bit "1" into 0, it will directly desynchronize the bitstream from that point forward because of incorrect parsing of the following syntax elements. However, an error in the INFO part (shown with X_i in Eq. (1)) does not have a direct desynchronization effect on the bitstream. Thus, all the INFO bits are possible NDBs. Some examples of EGC are presented in Table I.

The INFO bits of an EGC are not always categorized as NDBs. For instance, in the *mb_type* case, the "00100" pattern describes an inter (P)-MB, and motion syntaxes are parsed at the next step of the decoding process, while no such syntaxes exist following the "00110" pattern, which describes an intra (I)-MB. Therefore, although the bold bit zero in "001**0**0" and the bold bit one in the "001**1**0" pattern are INFO bits of EGC, they are not categorized as NDBs. Flipping an NDB should not affect the meaning of the syntax (although the value will change) and this should be studied individually for each syntax. We will now study the two most common syntax elements in H.264; *mvd_l0* and *trailing_ones_sign_flag* (T1).

***mvd_l0*:** The motion vector (MV) is a key element for exploiting temporal redundancy in a P-MB, and provides significant compression. The *mvd_l0* syntax is comprised of a pair of SGCs, one for the x, and the other for the y components, which represents the difference between the actual MV and the predicted one from the available neighbors. As mentioned previously, a bit error in the INFO part of an SGC does not cause any desynchronization at the bit level. Therefore, all the INFO part bits of x and y components of the *mvd_l0* syntax are categorized as NDBs. The only issue of an NDB associated with an MV is the propagation of an erroneous MV (without any effect on the bit level) to the following ones. This is discussed in detail in section III-B.

***trailing_ones_sign_flag* (T1):** The first parsing syntax present in the residual block is the pair value of the *coeff_token* syntax, as *TrailingOnes* and *TotalCoeff*, which signal the number of coefficients with ± 1 values and the total number of non-zero coefficients (out of 16), respectively. The sign of each *TrailingOnes* is signaled by a single bit (0/1) in the *trailing_ones_sign_flag* (T1) syntax. The zero value (bit 0) of the syntax means that the corresponding coefficient is +1, otherwise, it is -1. Since the T1 syntax does not have any direct or indirect desynchronization effect on the bitstream, its corresponding bits are categorized as NDBs.

Other NDBs: We characterize all the other NDBs by using its definition. If a bit is flipped and still satisfies the two conditions in the definition, which imply that 1) the corrupted packet is decodable, and 2) the number of decoded MBs in the packet is correct, then the bit is categorized as an NDB. This includes all the NDBs in other syntax elements, except the bits corresponding to the *mvd_l0* and T1 syntaxes.

III. Analysis of the NDBs

In this section, we will look first at the frequency of NDBs, and then at the effect of flipping each NDB on visual quality. To measure this, the first 60 frames of the following sequences- CIF (352×288) (*foreman*), 4CIF (704×576) (*crew*), 720×480 (*driving*) and 720×576 (*walk*)- are coded in IPPP… format (Intra refresh rate of 30 frames) using the H.264 Baseline profile with the Joint Model (JM) software, version 18.5 [21]. Each slice contains a single row of MBs.

A. Frequency of occurrence of the NDBs

In the first simulation, we sequentially invert all bits (flip individual bits one at a time) in each slice, and then the JM software is used to decode the corrupted packet. Each corrupted packet falls under one of the following categories:

1) It is not decodable (syntax/semantic error).
2) It is decodable, but the number of decoded MBs is not correct.
3) It is decodable, and the number of decoded MBs is correct.

Note that only the last category contains all the NDBs since it satisfies the two conditions in the definition of the NDB.

Fig. 1 shows the percentage of each category for the *crew* sequence at quantization parameter (QP) 32. As can be seen, flipping a single bit in the packet results in it being non-decodable in more than half of the cases (average 57.5%). On the other hand, on average, around 32.5% of the bits belong to the third category, which means that flipping those individual bits will not cause any desynchronization at the bit level, and the number of decoded MBs is also correct. Almost the same percentage is observed for the other QPs.

More detailed percentage values of the third category are presented in Table II. As shown, the predominant NDBs are different for low and high QP values. For instance, at low QP values, they originate mostly from the residual syntax (T1), while at higher QP values the *mvd_l0* syntax constitutes the majority of the NDBs. By averaging over different QP values for the *crew* sequence, we see that 32.9% of the bitstreams contain NDBs. We conducted this simulation for several other

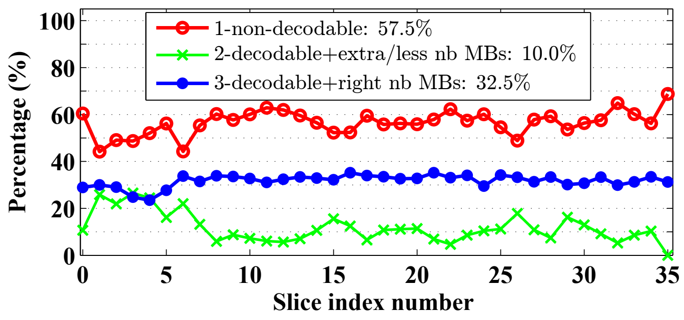

Fig. 1: Percentage of the three different categories of a corrupted slice on frame index 44 of the *crew* sequence at QP=32.

TABLE II: Frequency of occurrence of NDBs on frame index 44 for different sequences and different QPs. All the values inside the table presented in [%].

sequence name		**QP**			
		22	**27**	**32**	**37**
crew	T1	11.3	10.3	7.9	5.1
	mvd_l0	2.6	6.2	11.8	17.0
	nonsign, sign	1.5, 1.1	3.8, 2.4	7.5, 4.3	11.4, 5.6
	Other NDBs	21.1	16.3	12.8	9.1
	All NDBs	**35.0**	**32.8**	**32.5**	**31.2**
		32.9			
Known NDBs = NDBs of T1 and *mvd_l0*					
foreman (352×288)		14.0	17.0	21.1	22.7
crew (704×576)		13.9	16.5	19.7	22.1
driving (720×480)		11.5	13.7	15.8	18.3
walk (720×576)		7.9	10.2	13.2	15.7
Average		11.8	14.4	17.5	19.7

video sequences, and the percentage of “known NDBs” (NDBs of T1 and *mvd_l0*) are also presented in Table II.

B. Visual quality measurements of the NDBs

Although the NDBs do not have any effect on the bitstream level, they may cause some context modification. In this subsection, to examine the effect of the NDBs on the visual quality, each is flipped individually, and the peak signal-to-noise ratio (PSNR) is calculated on the corrupted frame.

Fig. 2 depicts the percentage of PSNR degradation for each NDB (the difference between the corrupted and the intact frames) in the *crew* sequence. For QP values of 22, 27, 32 and 37, respectively, around 96%, 92%, 88% and 90% of the error events still lead to PSNR values very close to the intact value (with less than a 0.05dB difference). As we can see, there are more cases with higher PSNR drops (>0.05dB) for higher QP values. From different simulations, we observed that the NDBs of *mvd_l0* are the more sensitive than those of other NDBs in terms of PSNR degradation. This is because a wrong MV does not only affect the current MB but propagates to the following MVs due to the coding of the residual MV. The propagation can stop when there is an I-MB or when the contribution of the wrong MV disappears from the calculation of a subsequent MV.

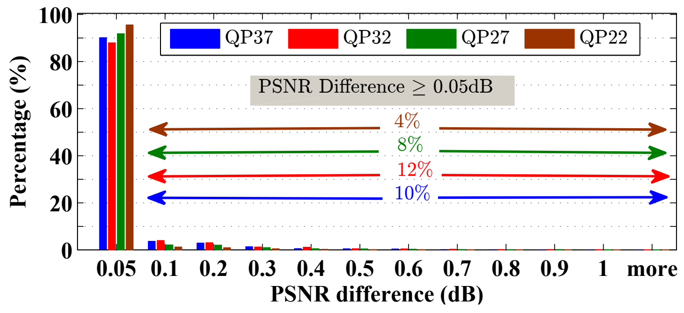

Fig. 2: Percentage of PSNR difference of all NDBs against the intact case on frame index 44 of the *crew* sequence.

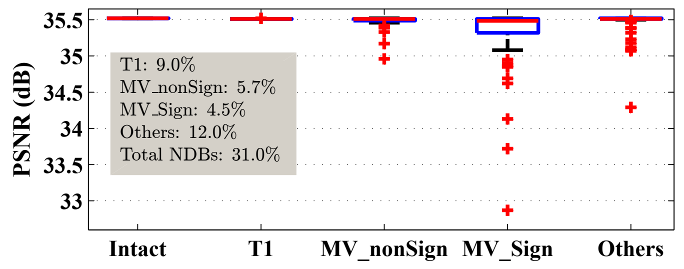

Fig. 3: PSNR of all NDBs on slice index 11 and frame index 44 of the *crew* sequence at QP 32 using box plots [22].

Thus, an erroneous NDB of an MV, without having any desynchronization effect on the bitstream, may still significantly affect the visual quality. This is demonstrated in Fig. 3, which shows the PSNR distributions of a slice in the *crew* sequence. We divided the NDBs of the MV syntax into the sign bit (least significant bit (LSB)) and Non-Sign bits. This division reveals that there are more outliers (as shown with '+' red symbol) in the case of sign bits, and the quality drops more severely with them as compared to the other cases.

However, it is worth mentioning that the percentage of MV sign bits is very small versus the whole bitstream, standing at about 5%, even in higher QPs (see Table II). Furthermore, not all of them can result in a significant PSNR degradation. As shown in Fig. 3, the median value (red line in the middle of the box) of each box is very close to the intact one; as well, the lower and higher bands of boxes (25–75 percentile of the data) confirm that the effect of flipping NDBs on the visual quality is insignificant and probably restricted to a very small area. Similar results are obtained with other video sequences.

IV. Proposed non-desync-based decoding framework

With all the new knowledge we have acquired on NDBs in the previous section, we now propose a robust decoding framework that retains the corrupted packets if the following two conditions are met:

1) the received corrupted packet should be decodable which means there is no syntax or semantic error, otherwise the decoder crashes, and
2) the number of MBs in the corrupted slices should be correct.

When those conditions are satisfied, we decode and render the received corrupted packet (i.e. consider it as the best candidate). Otherwise, we discard it and perform EC.

The proposed approach is illustrated in Fig. 4 as ② and ④ for two well-known EC approaches: (i) frame copy (FC) concealment by JM, (ii) a state-of-the-art concealment approach using the spatiotemporal boundary matching algorithm (STBMA) [23]. The first approach is the common frame copy concealment by JM in which the corrupted slice is ignored and replaced by the same slice from the previous decoded frame. The other approach is a state-of-the-art concealment approach using the STBMA, which is a superior but complex method of MB level error concealment [23].

To investigate the performance of the proposed approach, we propose to add a primary processing step before each EC approaches. First, the received corrupted packet is decoded if a corrupted packet satisfies the two conditions (this can be simply validated by decoding the corrupted packet whitout any crash or error status), which means that, most probably, one of the NDBs in the packet is hit by the error. Therefore, we keep the corrupted packet only in this case instead of ignoring it and performing any concealment. Otherwise, one of the EC approaches is employed to handle the corrupted packet. It is worth mentioning that the proposed approach can be combined with any other EC approach.

In Fig. 4, we present various approaches which will be compared in the next section. The JM-FC and STBMA approaches are named as approach ① and ③ respectively. Their corresponding proposed approaches for keeping the corrupted packets are also described as ② and ④ respectively. The proposed design of the approaches ② and ④ helps us to find out the improvement of the proposed approach over each EC approach separately. In other words, it will show how much gain each EC approach, individually, can get by retaining such corrupted packets. Note that when the received corrupted packet satisfies the two mentioned conditions, both proposed approaches ② and ④ are going to have the same performance.

V. Simulation results

Since in our simulations, we coded the sequences with one row of MBs in each slice, we can easily verify the second condition. Moreover, the number of MBs in the slice can be deduced from the information within other slices (the difference between the value of the *first_mb_in_slice* syntax element in the consecutive slices). Therefore, the proposed method applies even in the case where the number of MBs is not constant or known. The decoder validates the first condition by decoding the corrupted packet to find any syntax or semantic errors.

For each QP, a random error (single-bit) is considered. In fact, there are several interleaving techniques that can combat the problem of burst errors in wireless communication channels by fully randomizing the errors [24], [25]. A single frame (between the 30th frame and the 60th frame) is randomly selected for error. Then, we apply a uniform error distribution

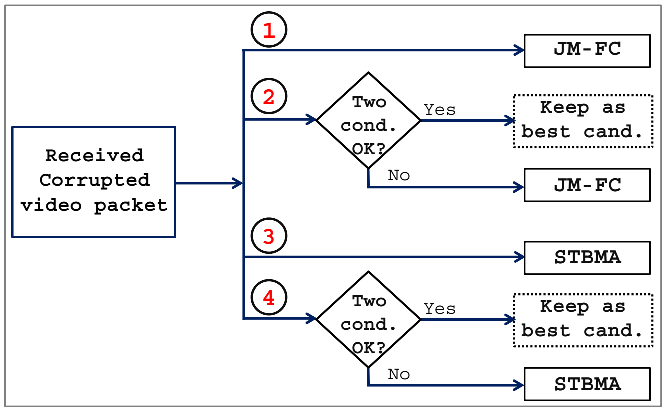

Fig. 4: Schematic of different approaches that we use in our simulations. Approach ① and ③ are concealment by JM-FC and STBMA respectively. The approach ② and ④ are the two proposed strategies for keeping a corrupted packet (under the two conditions) integrated with the JM-FC and STBMA concealment respectively.

on the bits of each packet with a channel residual bit error rate (ρ) value varying between approximately 10^{-7} for small QPs, and 10^{-6} for large QPs, to obtain one bit in error. These residual bit error rates are much higher than those observed in some broadcasting systems, such as DVB-H and DVB-SH-A, in recommended operational conditions [26]. The simulation is repeated 100 times for each QP, to ensure that the location of the erroneous bits did not bias our conclusions.

Table III presents the average PSNR values for different error handling approaches. The last column in the table shows the percentage of times that the received corrupted packet satisfies the two conditions. As it can be seen, even in randomly applied error in different frames, on average 34% of the time the received packet was decodable and the number of MBs in the decoded corrupted packet was correct. As we mentioned earlier in the Section III-B, these cases have very close PSNR to the intact one. Therefore, it is not reasonable to discard (or ignore) these good packets which occur frequently. The detailed results show that the proposed approaches ② and ④ outperform respectively ① (JM-FC) and ③ (STBMA) approaches in all cases. Note that the PSNR difference between each method and JM-FC appears in parentheses in the table.

Fig. 5 depicts the average PSNR difference of each approach from JM-FC for different QP values. We observe that keeping corrupted packets provides significant PSNR gains over JM-FC (approach ①) for all four QP values. For instance, when it is added to the STBMA approach (described as approach ④), it is more than 3.93 dB better than JM-FC at QP 22. On average, over all QPs, it offers a 1.2 dB gain improvement in approach ② and over 2.51 dB gain improvement in approach ④ over concealment in JM-FC (①).

As it can be observed from Table III, in some cases in higher QP values such as *walk* sequence at QP=22, the average

TABLE III: Comparison of the average PSNR (dB) of reconstructed corrupted frames for different approaches. The PSNR differences between each method and approach ① appear in parentheses. The last column shows the percentage of the cases that the received corrupted packet satisfied the two conditions.

Sequence	**QP**	**Average PSNR** (reconstructed corrupted frame)					2 Conds. Met
		Intact	①	②	③	④	
foreman (352×288)	22	41.35	37.60	39.00 (1.40)	39.49 (1.89)	40.21 (2.61)	36%
	27	37.82	35.79	36.32 (0.53)	36.92 (1.13)	37.04 (1.25)	34%
	32	34.67	33.70	33.93 (0.23)	34.19 (0.49)	34.31 (0.61)	30%
	37	31.92	31.39	31.55 (0.16)	31.63 (0.24)	31.69 (0.30)	34%
crew (704×576)	22	41.78	39.21	39.93 (0.72)	40.64 (1.43)	40.97 (1.76)	28%
	27	38.53	37.09	37.61 (0.52)	38.03 (0.94)	38.22 (1.13)	38%
	32	35.69	34.96	35.23 (0.27)	35.44 (0.48)	35.53 (0.57)	31%
	37	33.00	32.64	32.73 (0.09)	32.86 (0.22)	32.89 (0.25)	28%
Ice (704×576)	22	43.70	39.18	40.58 (1.40)	41.74 (2.56)	42.28 (3.10)	36%
	27	41.44	38.00	39.27 (1.27)	40.05 (2.05)	40.52 (2.52)	30%
	32	39.00	36.50	37.51 (1.01)	38.15 (1.65)	38.50 (2.00)	35%
	37	36.43	34.37	34.95 (0.58)	35.77 (1.40)	35.98 (1.61)	34%
driving (720×480)	22	41.02	34.05	36.85 (2.80)	38.08 (4.03)	39.24 (5.19)	40%
	27	37.05	32.59	34.15 (1.56)	35.59 (3.00)	36.05 (3.46)	37%
	32	33.29	30.84	31.80 (0.96)	32.64 (1.80)	32.92 (2.08)	40%
	37	30.00	28.84	29.26 (0.42)	29.72 (0.88)	29.80 (0.96)	33%
walk (720×576)	22	43.19	30.62	34.53 (3.91)	35.33 (4.71)	37.61 (6.99)	31%
	27	39.25	30.20	33.25 (3.05)	34.95 (4.75)	36.31 (6.11)	34%
	32	35.55	29.30	31.33 (2.03)	33.51 (4.21)	34.03 (4.73)	34%
	37	31.98	28.08	29.17 (1.09)	30.88 (2.80)	31.10 (3.02)	35%
Average difference from ①			—	**1.20**	**2.03**	**2.51**	**34%**

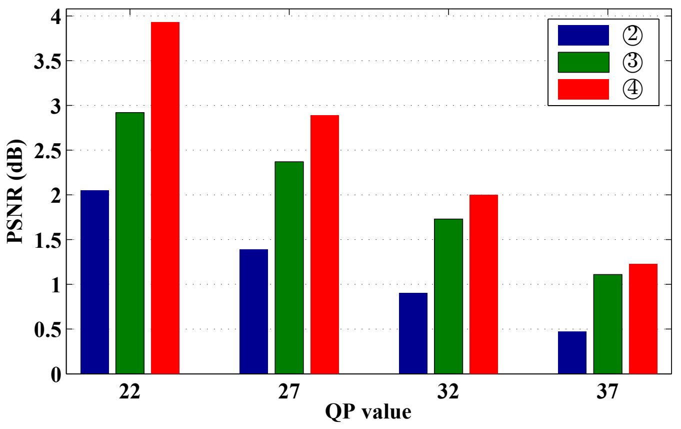

Fig. 5: Average PSNR difference of all approaches from JM-FC (approach ①) for different QP values.

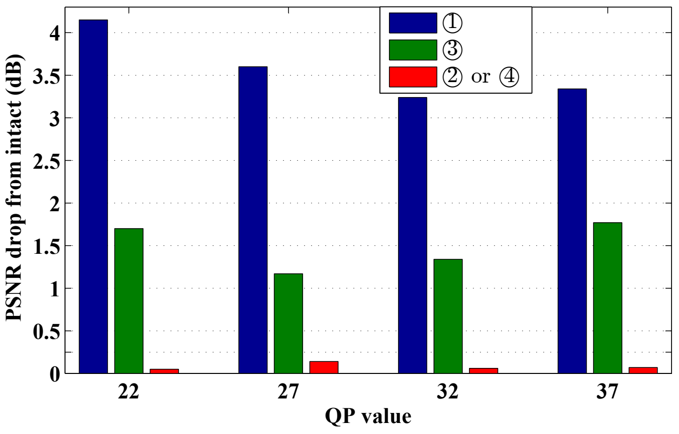

Fig. 6: Average PSNR drop from the intact frame in all approaches for different QP values only when the two conditions are met.

TABLE IV: Average PSNR (dB) improvement over each EC method (approach ① and ③) after considering to keep corrupted packets as in approach ② and ④ respectively.

QP	Two conditions Met			All	
	Percentage	PSNR ②-①	PSNR ④-③	PSNR ②-①	PSNR ④-③
22	34%	4.10	1.65	2.05	1.01
27	35%	3.46	1.03	1.39	0.52
32	34%	3.18	1.28	0.90	0.27
37	33%	3.27	1.70	0.47	0.12
Avg.	**34%**	**3.50**	**1.42**	**1.20**	**0.48**

PSNR value of the proposed approach ④ or ② is still far from the intact one. This PSNR reduction mainly comes from the other 66% of the case, i.e. from those cases that the received corrupted packet does not satisfy the two conditions and the integrated concealment approaches such as STBMA or JM-FC are used but failed to reconstruct well the lost information.

Fig. 6 depicts the average PSNR drop in each approach compared to the intact frame when the received corrupted packet meets the two conditions (on average 34% of the case with random single-bit error). Note that in this case, approaches ② and ④ retain that packet as best candidate while ① and ③ will perform the concealment task. It is clear from the figure that the PSNR drop compared to the intact case, for the proposed strategy is less than 0.1 dB while for a state-of-the-art concealment approach using the STBMA alone it is almost 1.5 dB. The reduction is even more, around 3.58 dB, as the basic simple concealment approach, JM-FC, is employed.

As separately presented in Table IV, on average on all QPs, the proposed approach brings an average gain of about 3.5 dB over JM and 1.42 dB over STBMA (in 34% of the cases).

The gain in visual quality is illustrated in Fig. 7. The difference between the reconstructed frame and the intact one is also provided in the figure. Comparing the reconstructed frames by JM and STBMA concealment approaches, it is clear that keeping the corrupted packet is a beneficial choice

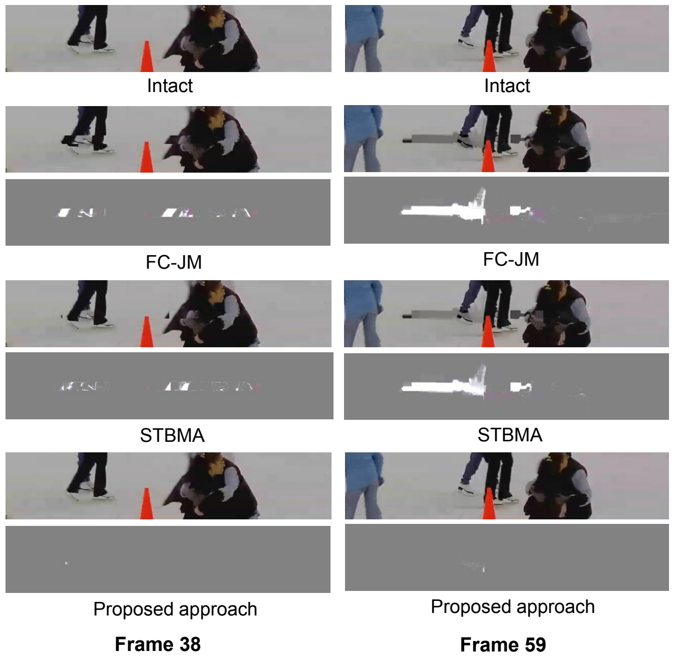

Fig. 7: Visual comparison of a reconstructed frame in *Ice* sequence at QP=37 for different methods. In this case, the one bit error occurred in frame 38, slice 15 and on the *coeff_token* syntax element. The received corrupted packet was decodable and the number of decoded MBs was correct (44 MBs). Therefore, both proposed approach ② and ④ will keep the corrupted packet as best candidate without doing any concealment. While the ① and ③ will ignore the corrupted packet and perform only the concealment. The pictures on the left side of the figure are showing the reconstructed frame 38 by different approaches. In each case, the difference from intact are also captured by stream analyzer and provided. The PSNR values for frame 38 for each approach are as follows: Intact (36.76 dB), JM-FC (32.36 dB), STBMA (35.24 dB), the proposed approach (36.75 dB). The pictures on the right side of the figure are showing the error propagation in each case. In fact, twenty frames after the corrupted frame, frame 59 is captured to demonstrate the effect of the error on the following frames. The PSNR values for frame 59 for each approach are as follows: Intact (36.26 dB), JM-FC (32.10 dB), STBMA (32.74 dB), proposed approach (36.25 dB).

and it outperforms the two EC approaches. Furthermore, the error propagation effect is also shown in the figure, which confirms that the quality degrades drastically in the following frames (even after 10 frames) for the cases with EC. It is obvious that, keeping the corrupted packet (if it satisfies the two conditions) has a huge impact on the visual quality of the reconstructed corrupted frame, and more importantly, reduces the propagation of errors to subsequent frames due to the predictive coding.

From the results of all figures and tables, it can be inferred that for around one-third of the cases, the received corrupted packet is valid (satisfies two conditions), going through extra processes in concealment does not provide better results than the received one. As the results have shown, the received corrupted but valid packet is going to have a PSNR value very close to the intact one in most cases and we can not go beyond that. So, as a result, keeping the corrupted packet provides a significantly higher PSNR value and better quality compared to EC approaches alone. This is important not only for the corrupted frame, but for the following ones, as fewer visible drifting effects will result as shown in Fig. 7.

VI. Conclusion

In this paper, we identified the NDBs of the two most common syntax elements in H.264 CAVLC coded sequences. It was observed that on average, the NDBs make up one-third of all bitstreams. The simulation results revealed that the effect of NDBs on context modification is insignificant and that the majority of them (90%) provide PSNR values that are highly comparable to the intact value. Therefore, it is beneficial to keep the corrupted packets if the NDBs are erroneous. This proposed approach can be combined with any other concealment or correction approaches.

References

[1] ISO/IEC JTC 1/SC 29/WG 11, "High Efficiency Video Coding," p. 317, 2013.

[2] International Telecommunications Union, "ITU-T recommendation H.264: Advanced video coding for generic audiovisual services," 2003.

[3] "ITU-T SG 16 standardization on visual coding – the video coding experts group (VCEG)," [Online; accessed 20-January-2018]. Available: https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/vceg.aspx.

[4] W. Tan, B. Shen, A. Patti, and G. Cheung, "Temporal propagation analysis for small errors in a single-frame in H.264 video," in *IEEE 15th Int. Conf. Image Process.*, 2008, pp. 2864–2867.

[5] M. Usman, X. He, M. Xu, and K. Lam, "Survey of error concealment techniques: Research directions and open issues," in *IEEE Picture Coding Symposium (PCS)*, 2015, pp. 233–238.

[6] J. Liu, G. Zhai, X. Yang, B. Yang, and L. Chen, "Spatial error concealment with an adaptive linear predictor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 353–366, 2015.

[7] W. Lie, C. Lee, C. Yeh, and Z. Gao, "Motion vector recovery for video error concealment by using iterative dynamic-programming optimization," *IEEE Trans. Multimedia.*, vol. 16, no. 1, pp. 216–227, 2014.

[8] T. Lin, T. Ding, N. Yang, P. Wu, K. Tung, C. Lai, and T. Chang, "Video motion vector recovery method using decoding partition information," *IEEE Journal of Display Technology*, vol. 12, no. 11, pp. 1451–1463, 2016.

[9] Z. Zhou, M. Dai., R. Zhao, B. Li, H. Zhong, and Y. Wen, "Video error concealment scheme based on tensor model," *Multimedia Tools and Applications*, pp. 1–17, 2016.

[10] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," in *Proc. MoMM*, 2006, pp. 49–58.

[11] L. Trudeau, S. Coulombe, and S. Pigeon, "Pixel domain referenceless visual degradation detection and error concealment for mobile video," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 2229–2232.

[12] L. Larzon, M. Degermark, S. Pink, L. Jonsson, and G. Fairhurst, "The lightweight user datagram protocol (UDP-Lite)," IETF, RFC 3828, Jul. 2004. [Online; accessed 20-January-2018]. Available: https://www.rfc-editor.org/rfc/rfc3828.txt.

[13] M. Barni, F. Bartolini, and P. Bianco, "On the performance of syntax-based error detection in H.263 video coding: a quantitative analysis," in *Electronic Imaging, SPIE Conf. Image and Video Communications*, vol. 3974, 2000, pp. 949–957.

[14] Hendrawan and N. I. Yusuf, "Impact analysis of bit error transmission on quality of H.264/AVC video codec," in *IEEE 7th Int. Conf. Telecommunication Systems, Services, and Applications (TSSA)*, 2012, pp. 314–317.

[15] A. Demirtas, A. Reibman, and H. Jafarkhani, "Performance of H.264 with isolated bit error: Packet decode or discard?" in *Proc. IEEE 18th Int. Conf. Image Process.*, 2011, pp. 949–952.

[16] C. Lamy-Bergot and C. Bergeron, "Video H.264 encryption preserving synchronization and compatibility of syntax," US Patent 8,160,157, Apr. 17, 2012, https://www.google.com/patents/US8160157.

[17] E. Tsimbalo, X. Fafoutis, and R. Piechocki, "CRC error correction in IoT applications," *IEEE Trans. Industrial Informatics*, vol. 13, no. 1, pp. 361–369, 2017.

[18] M. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, 2017.

[19] A. Badr, A. Khisti, W. T. Tan, and J. Apostolopoulos, "Perfecting protection for interactive multimedia: A survey of forward error correction for low-delay interactive applications," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 95–113, March 2017.

[20] K. Yu, M. Gidlund, J. Åkerberg, and M. Björkman, "Reliable and low latency transmission in industrial wireless sensor networks," *Procedia Computer Science*, vol. 5, pp. 866–873, 2011.

[21] "H.264/AVC JM Reference Software," version 18.5, [Online]. Available: http://iphome.hhi.de/suehring/tml/.

[22] "Box and Whisker Plot," [Online; accessed 20-January-2018]. Available: https://datavizcatalogue.com/methods/box_plot.html.

[23] Y. Chen, Y. Hu, O. Au, H. Li, and C. Chen, "Video error concealment using spatio-temporal boundary matching and partial differential equation," *IEEE Trans. Multimedia.*, vol. 10, no. 1, pp. 2–15, 2008.

[24] Y. Q. Shi, X. M. Zhang, Z.-C. Ni, and N. Ansari, "Interleaving for combating bursts of errors," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 29–42, 2004.

[25] K. Kang and L. Sha, "An interleaving structure for guaranteed qos in real-time broadcasting systems," *IEEE Trans. Computers*, vol. 59, no. 5, pp. 666–678, 2010.

[26] L. Polák and T. Kratochvíl, "DVB-H and DVB-SH-A performance and evaluation of transmission in fading channels," in *IEEE 34th Int. Conf. Telecommunications and Signal Processing (TSP)*, 2011, pp. 549–553.