Temporal information based GoP adaptation for linear video delivery schemes

Anthony Trioux, François-Xavier Coudoux, Patrick Corlay, M Gharbi

Version of Record: https://www.sciencedirect.com/science/article/pii/S0923596519302577
Manuscript_4e08ce506d16ef388d27fbdb385fa163

Temporal Information based GoP Adaptation for Linear Video Delivery schemes

Anthony Trioux[a,*], François-Xavier Coudoux[a], Patrick Corlay[a] and Mohamed Gharbi[a]

[a]Univ. Polytechnique Hauts-de-France, CNRS, Univ. Lille, ISEN-Yncréa, Centrale Lille, UMR 8520 - IEMN - DOAE, F-59313 Valenciennes, France

ARTICLE INFO

Keywords:
GoP-size adaptation
Temporal Information index
Linear Video Coding
SoftCast
Joint Source Channel Coding (JSCC)
Uncoded Video Transmission

Abstract

An original wireless video transmission scheme called SoftCast has been recently proposed to deal with the issues encountered in conventional wireless video broadcasting systems (e.g. cliff effect). In this paper, we evaluate and optimize the performances of the SoftCast scheme according to the transmitted video content. Precisely, an adaptive coding mechanism based on GoP-size adaptation, which takes into account the temporal information fluctuations of the video, is proposed. This extension denoted Adaptive GoP-size mechanism based on Content and Cut detection for SoftCast (AGCC-SoftCast) significantly improves the performances of the SoftCast scheme. It consists in modifying the GoP-size according to the shot changes and the spatio-temporal characteristics of the transmitted video. When hardware capacities, such as buffer or processor performances are limited, an alternative method based only on the shot changes detection (AGCut-SoftCast) is also proposed. Improvements up to 16 dB for the PSNR and up to 0.55 for the SSIM are observed with the proposed solutions at the cut boundaries. In addition, temporal visual quality fluctuations are reduced under 1dB in average, showing the effectiveness of the proposed methods.

1. Introduction

According to Cisco Visual Networking Index report [1], video traffic will represent 82% of all consumer's Internet traffic in the coming years. However, broadcasting video content to multi-users is challenging because each user's wireless channel is unreliable and different. In such applications, traditional approaches based on video codecs such as H.264/AVC [2] or HEVC [3] are not suitable since they require a permanent adaptation of the source and channel coding parameters by the transmitter. Indeed, they are adjusted to match an available bitrate that is given under predicted or assumed channel state. Due to channel heterogeneities between users, receivers whose channel conditions are degraded are subject to significant visual disturbances (e.g. freeze), while receivers experiencing a better channel quality than the estimated one cannot take full advantage of it. These two problematic issues are known as cliff effect [4] and levelling-off effect [5], respectively. The first one, refers to a sudden and brutal loss in received video quality. The second one, refers to a video quality that stays almost constant even if the Channel Signal-to-Noise Ratio (CSNR) increases.

Scalable video coding [6] may partly solve these problems but a more radical approach known as SoftCast has been recently proposed. SoftCast represents the pioneer work of linear video delivery systems also known as uncoded video transmission schemes. The pixels are processed by successive linear operations and directly transmitted without quantization or channel coding. This allows users to receive a video quality that increases linearly with channel quality (graceful degradation [7]). It operates without any feedback information [8], while avoiding the complex adaptation mechanisms of conventional schemes. In addition, a unique data stream is transmitted for all receivers. The latter can be decoded by anyone even those experiencing poor channel conditions.

In its current configuration, SoftCast uses a fixed Group of Pictures (GoP) size. Unfortunately, this method does not adapt to the specific spatio-temporal characteristics of each video content. In [9], we reported a preliminary study on the impact of the video content in a SoftCast context. This study showed that it is of paramount importance to consider the spatio-temporal specificities of the video before applying any transformation of the SoftCast scheme.

In this paper, we extend our previous work [9] and propose an original adaptive coding mechanism to significantly improve the performances of the SoftCast scheme. The proposed mechanism takes into account the temporal information fluctuations of the video to be transmitted. First, we provide additional results by considering bandwidth-limited environments and various video formats. Then, we show that SoftCast performs poorly when considering shot changes scenarios also known as cuts. It leads to annoying visual artifacts: a ghost effect phenomenon, as well as severe temporal quality fluctuations. We analyze these impairments and show that they can be annihilated (ghost effect) or greatly reduced (quality fluctuations) by properly encoding the video. Finally, we propose an Adaptive GoP-size mechanism based on Content and Cut detection for SoftCast (AGCC-SoftCast). This extension dynamically adapts the GoP-size, prior to encoding, by taking into account both shot changes and spatio-temporal characteristics of the transmitted video. When hardware capacities, such as buffer or processor performances are limited, an alternative method based on the cut detection only (AGCut-SoftCast) is also

*Corresponding author
anthonytrioux@laposte.net (A. Trioux)
ORCID(s): 0000-0003-3457-3301 (A. Trioux); 0000-0002-5817-7429 (F. Coudoux); 0000-0002-3407-8805 (P. Corlay); 0000-0003-3777-4336 (M. Gharbi)

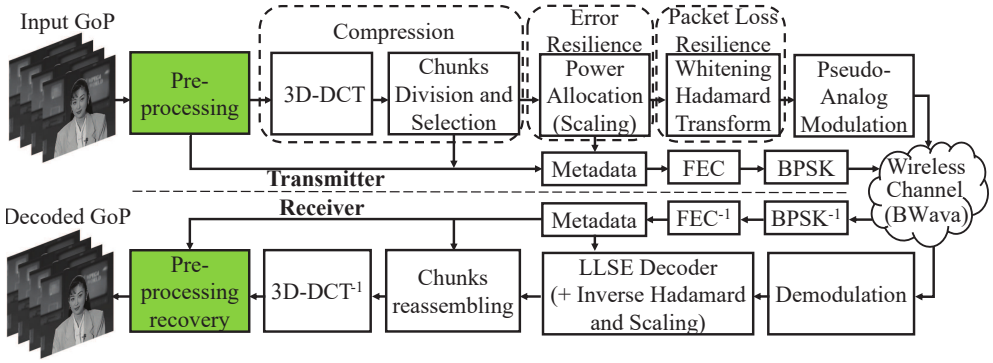


Figure 1: Block diagram of the SoftCast video transmission scheme.

proposed. Simulation results show that the proposed extension optimizes the trade-off between received video quality, available bandwidth and complexity cost. Improvements up to 16 dB for the PSNR and up to 0.55 for the SSIM are observed with the proposed methods at the cut boundaries. In addition, visual quality fluctuations are significantly reduced under 1dB in average, showing the effectiveness of the proposed methods.

The rest of the paper is organized as follows: Section 2 reviews the original SoftCast scheme and related works. Then, Section 3 describes the proposed AGGC-SoftCast solution. First, preliminary results on the SoftCast performances for different GoP-sizes and compression levels are given and analyzed in Section 3.1. In Section 3.2, the ghost effect artifact is highlighted, analyzed and a solution to annihilate it is given. Based on these studies, an extension of SoftCast, called AGCC-SoftCast, is proposed and detailed in Section 3.3. Simulation results of the proposed methods and comparisons with the classical SoftCast scheme (considering different GoP-sizes) are given in Section 4. Conclusions and discussions are given in Section 5.

2. Background

The basic scheme of *SoftCast* [10] is shown in Fig. 1. We note that the green blocks are not part of the original SoftCast scheme but have been added as additional steps. In the following, for ease of notation, we refer to this version as the classical SoftCast scheme. The green block at the transmitter consists of an energy reduction that allows a better distribution of the available power, therefore offering a greater resilience to channel disturbances. To reduce the signal energy, the spatial average is subtracted from each image. This average is transmitted as additional side information (metadata). As indicated in [11], such preprocessing method improves the received quality by up to 2.5 dB.

SoftCast operates GoP by GoP. For each GoP it first decorrelates the signal through a three-dimensional full-frame Discrete Cosine Transform (3D-DCT) as shown in Fig. 2. Then, the transformed frames are divided into small rectangular blocks called *chunks* and rearranged to form a new matrix where each row defines a *chunk*. These *chunks*

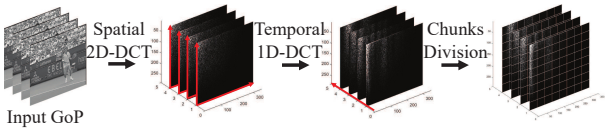


Figure 2: Compression Step in SoftCast scheme. From left to right: GoP in pixel domain, 2D-transformed frames, 3D-transformed frames, *chunks* division after 3D-DCT.

are sorted in decreasing energy order. Then, compression may be applied depending on the available bandwidth BW_{ava} and the modulation type used at the transmitter side. It consists in discarding a certain (given) amount of *chunks*. The corresponding compression ratio (CR) [12] is defined as

$$\mathrm{CR} = M/N, \tag{1}$$

where, M is the number of transmitted *chunks* and N the total number of *chunks* within a GoP. This CR ranges between 0 (no data sent) and 1 (no compression). If the available bandwidth BW_{ava} is less than the source bandwidth, the value of M is adjusted accordingly [11].

The next block called Power Allocation or Scaling is used to provide error resilience to the data. Based on a fixed power budget P, SoftCast scales the magnitude of the DCT coefficients to offer a better protection against channel noise. Hadamard Transform is then applied on the scaled coefficients to provide packet loss resilience. By mixing the data, it ensures that each packet contains approximately the same amount of information. Finally, the obtained coefficients are directly mapped in pairs (I and Q planes in the Orthogonal Frequency Division Multiplexing technology) and transmitted without any coding step in a pseudo-analog manner referred as Raw-OFDM [13]. In addition, metadata are transmitted to the receiver to be able to recover the video signal. For each GoP, they represent a small amount of three datasets:

- The mean of each *chunk*, denoted by μ_p with $p = (1, 2, \ldots, M)$;
- The variance/energy of each *chunk*, noted λ_p;
- A bitmap which indicates the positions of the discarded *chunks* in the GoP.

To ensure a correct decoding process, they are transmitted in a robust way after entropy coding (e.g. by using Binary Phase-Shift Keying modulation [14] and Huffman coding). We note that the added green block in Fig. 1 induces an additional small dataset corresponding to the average value of each frame in the GoP (8 bits per frame before Huffman coding [11]).

At the receiver side, a Linear Least Square Error (LLSE) decoder is used to get the best estimation of received values. The decoded values are then reassembled to form frames that are passed through an inverse 3D-DCT process. In case of bandwidth-constrained environments, the discarded *chunks* at the transmitter side are replaced by null values.

Following the original works [8], linear video coding has gathered a significant interest from the research community [13–17]. Among the existing works, Xiong *et al.* [13] studied the theoretical performances of SoftCast. They showed that the latter are directly related to the *data activity* of a GoP denoted by

$$H = \frac{1}{N}\sum_{d=1}^{N}\sqrt{\lambda}_d, \tag{2}$$

where $\lambda_d = E[\mathbf{X_d}^2]$ is the energy of the d^{th} *chunk* after 3D-DCT process [8]. They demonstrated that this term directly affects the reconstructed PSNR at the receiver side as follows

$$\text{PSNR} = c + \text{CSNR} - 20\log_{10}\left(H\right), \tag{3}$$

with $c = 20\log_{10}(255)$. Note the linear characteristic of the PSNR with the channel transmission conditions.

The data activity H actually represents a good way to analyze the performances of SoftCast. However, it is not well adapted to perform dynamic GoP-size adaptation. Indeed, the computation requires to transform the signal through 3D-DCT. Furthermore, due to this 3D transformation, the data activity is a GoP-based measure, meaning that a constant value is obtained for the duration of a GoP. Consequently, it is not adapted to detect shot changes scenarios inside a GoP.

Although the trade-off between GoP-size and complexity cost for SoftCast was briefly discussed in [18], no proposal was made to find an optimal GoP-size according to the transmitted content. By optimal, we mean the GoP-size that optimizes the trade-off between received quality and complexity cost. Despite, some adaptive GoP-size solutions for conventional standards (e.g., H.264/AVC, H.264/SVC) or Wyner-Ziv codec already exist [19–21] they cannot be directly applied to SoftCast. Indeed, different from conventional approaches that rely on motion estimation/compensation and entropy coding, SoftCast uses a temporal DCT to exploit inter-frame correlation and a pseudo-analog modulation for transmission. Since both are performed in a very different way than conventional approaches, adaptation of the GoP-size should be studied in SoftCast-based schemes to optimize performances.

3. The proposed AGCC-SoftCast scheme

In most of the linear video transmission related works, only a fixed GoP-size usually equal to 8 or 16 frames is used for performance assessment. Although some works have already mentioned that adapting the GoP-size may improve the received quality [13, 18], to the best of our knowledge, this is the first attempt to propose an adaptive GoP-size coding scheme for SoftCast that takes into account the characteristics of the video content. Specifically, we propose an Adaptive GoP-size mechanism based on Content and Cut detection for SoftCast (AGCC-SoftCast), which takes into account the temporal fluctuations of the video. This method significantly improves the performances of the SoftCast scheme. Preliminary analyses are first carried out to facilitate the introduction of the proposed method.

3.1. Preliminary Analysis

In this section, we examine the impact of the video content on the received quality for different GoP-sizes. To evaluate the amount of spatial and temporal information in a video sequence, we use the Spatial Information (SI) and Temporal Information (TI) indexes proposed by the ITU-T [22], which are defined as follows

$$\text{SI} = max_{time}\{std_{space}[Sobel(F_k(i,j))]\}, \tag{4}$$

$$\text{TI} = max_{time}\{std_{space}[F_k(i,j) - F_{k-1}(i,j)]\}, \tag{5}$$

where $F_k(i,j)$ represents the k^{th} frame, (i,j) the corresponding spatial coordinates and $Sobel()$ the Sobel filtering operation, respectively.

However, as mentioned in [23], due to the current definition that selects the highest value along the time axis, computing the TI index for a video with relative slow motions that contains cut(s) results in a high value. To avoid such inconsistency, we choose to average the results over the sequence. The following new definitions are considered in the rest of this paper instead of eq. (4) and eq. (5):

$$\text{SI} = mean_{time}\{std_{space}[Sobel(F_k(i,j))]\}, \tag{6}$$

$$\text{TI} = mean_{time}\{std_{space}[F_k(i,j) - F_{k-1}(i,j)]\}. \tag{7}$$

3.1.1. Simulation Setup

Video sources: Two commonly used video formats are here considered: HD720p (1280 × 720 pixels, 60fps) and CIF (352 × 288 pixels, 30fps). Only the luminance part of the videos (from the Xiph collection [24]) is considered. The process is performed GoP by GoP considering three different fixed GoP-sizes: 8, 16 and 32 frames. Each CIF frame is split into 64 *chunks* [14, 25] whereas each HD frame is split into 256 *chunks* as in [26, 27].

Wireless characteristics: Transmissions through Additive White Gaussian Noise channels (AWGN) with a CSNR in the range of [0∼25dB] are considered. In addition, four available channel bandwidth cases are used: *Full, three-quarter, half* and *quarter* bandwidth. This corresponds to CR=1, 0.75, 0.5 and 0.25, respectively.

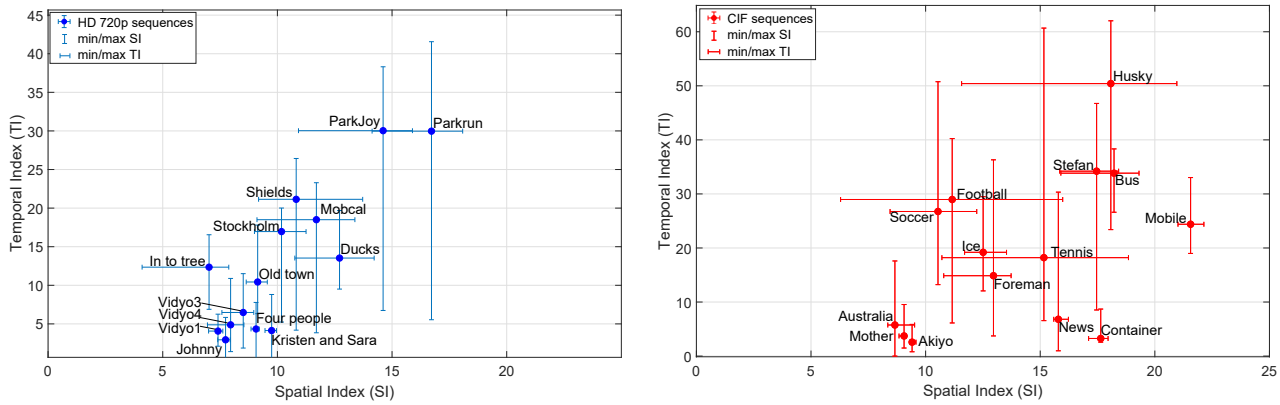


Figure 3: Illustration of the spatio-temporal index for the selected HD and CIF video sequences. Dots correspond to the average values across the video sequence. Vertical and Horizontal bars represent respectively the min/max value of the Temporal Index and the Spatial Index of the video sequences. From left to right: HD sequences, CIF sequences.

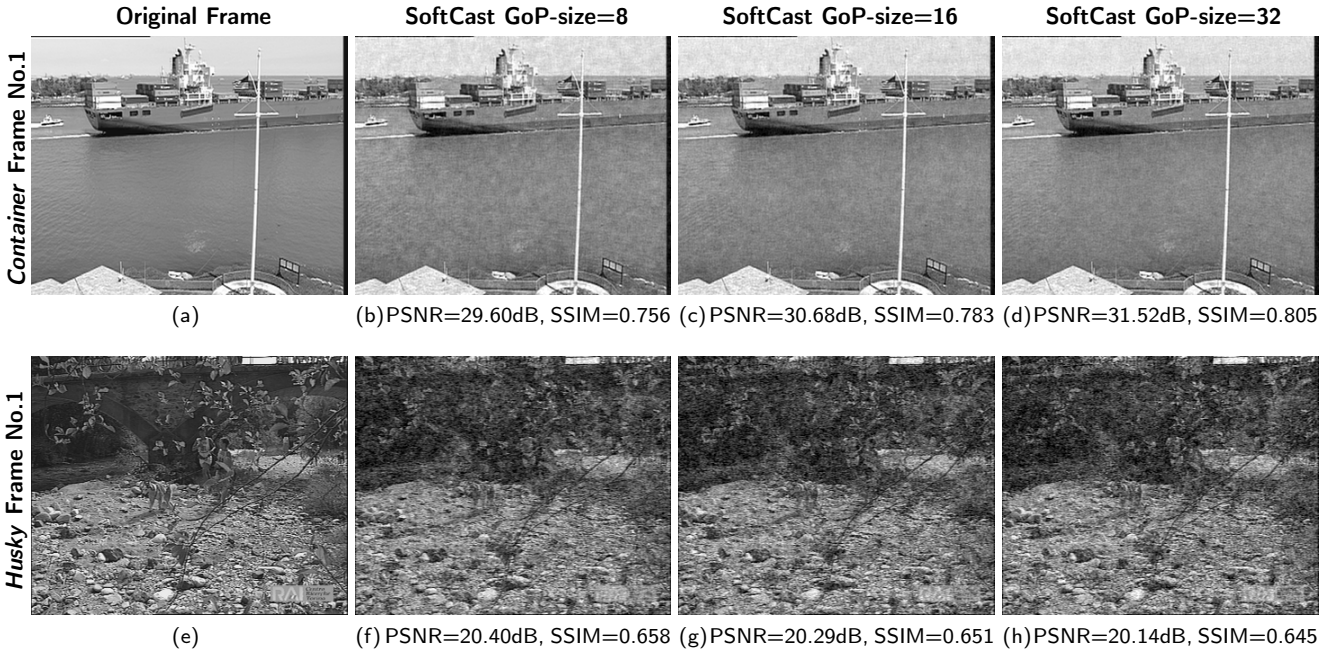


Figure 4: Visual quality comparison at a CSNR = 0dB, and with no compression applied (CR = 1). (a),(b),(c),(d): first frame of *Container*. (e),(f),(g),(h): first frame of *Husky*. (a),(e): Original frame. (b),(f): SoftCast fixed GoP-size = 8. (c),(g): SoftCast fixed GoP-size = 16. (d),(h): SoftCast fixed GoP-size = 32.

Evaluation Metrics: Two metrics widely used by the scientific community are here considered: the PSNR [28] and the SSIM. The latter [29] provides a quality index, which is more correlated with the Human Visual System (HVS) [16].

In this section, we choose to show the results for the *Akiyo*, *Husky* and *Container* CIF sequences as well as the *Johnny*, *Parkrun* and *Parkjoy* HD sequences because of their different spatio-temporal characteristics, as shown in Fig 3. Similar results are obtained with other sequences. In order to evaluate the fluctuations of the SI and TI indexes, Fig 3 also contains the minimum and maximum values of the SI, TI represented by a horizontal and vertical bar, respectively.

3.1.2. Simulation results

Table 1 gives a summary of the received quality for the selected video sequences. Results show that:

- A large GoP for video sequences with low spatio-temporal activity (e.g., *Akiyo* or *Johnny*) gives the best reconstructed quality video quality;
- In the opposite case, i.e., for content with strong spatio-temporal activity like *Husky*, there is no interest to increase the size of the GoP in terms of reconstructed quality. This is even more true as the complexity increases according to $O(Klog(K))$ with K the number of frames in a GoP [10, 30]. Using a small GoP-size allows to reduce the complexity up to 40%;

Table 1
Table of the resulting PSNR and SSIM scores for different GoP-size and different CSNR with CR = 1 (no compression) and CR=0.25 (75% of discarded coefficients).

Simulation Setup			CSNR(dB) 0		10		20	
			PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
GoP-size = 8	CR=1	*Akiyo*	36.27	0.875	45.06	0.977	53.77	0.996
		Container	30.11	0.781	39.17	0.956	48.55	0.994
		Husky	**20.56**	**0.671**	**28.62**	**0.912**	**38.26**	**0.987**
		Johnny	35.87	0.916	44.66	0.984	53.35	0.997
		Parkjoy	**24.71**	**0.672**	**32.83**	**0.903**	**42.41**	**0.986**
		Parkrun	24.53	0.738	32.83	0.935	42.42	0.992
	CR=0.25	*Akiyo*	29.12	0.764	37.46	0.938	44.36	0.988
		Container	25.66	0.628	33.93	0.883	41.42	0.977
		Husky	**18.02**	**0.476**	**21.44**	**0.726**	**22.50**	**0.804**
		Johnny	31.61	0.844	39.65	0.955	45.19	0.984
		Parkjoy	**22.11**	**0.545**	**26.54**	**0.781**	**28.16**	**0.876**
		Parkrun	21.58	0.603	26.90	0.840	29.18	0.921
GoP-size = 16	CR=1	*Akiyo*	35.13	0.901	44.05	0.982	52.84	0.997
		Container	31.57	0.821	40.61	0.967	49.84	0.996
		Husky	20.60	0.671	28.66	0.912	38.29	0.987
		Johnny	37.17	0.930	45.82	0.987	54.52	0.998
		Parkjoy	24.84	0.676	32.95	0.905	42.52	0.987
		Parkrun	**24.89**	**0.750**	**33.17**	**0.939**	**42.75**	**0.992**
	CR=0.25	*Akiyo*	30.72	0.808	38.89	0.952	45.29	0.990
		Container	27.09	0.677	35.41	0.908	42.71	0.982
		Husky	18.09	0.477	21.43	0.725	22.45	0.801
		Johnny	33.07	0.869	40.83	0.963	45.77	0.985
		Parkjoy	22.26	0.551	26.65	0.784	28.25	0.877
		Parkrun	**21.99**	**0.622**	**27.23**	**0.849**	**29.44**	**0.925**
GoP-size = 32	CR=1	*Akiyo*	**36.27**	**0.917**	**45.06**	**0.985**	**53.77**	**0.998**
		Container	**32.81**	**0.851**	**41.79**	**0.974**	**50.88**	**0.996**
		Husky	20.61	0.671	28.65	0.912	38.29	0.987
		Johnny	**38.13**	**0.939**	**46.67**	**0.988**	**55.53**	**0.998**
		Parkjoy	24.88	0.676	32.97	0.905	42.55	0.987
		Parkrun	25.01	0.753	33.26	0.939	42.84	0.992
	CR=0.25	*Akiyo*	**31.99**	**0.839**	**39.97**	**0.961**	**45.85**	**0.991**
		Container	**28.34**	**0.717**	**36.65**	**0.927**	**43.64**	**0.985**
		Husky	18.11	0.476	21.36	0.721	22.35	0.796
		Johnny	**34.21**	**0.886**	**41.70**	**0.967**	**46.11**	**0.986**
		Parkjoy	22.31	0.552	26.66	0.783	28.24	0.875
		Parkrun	22.17	0.629	27.27	0.850	29.37	0.924

- An optimal GoP-size that either maximizes the received quality scores or minimizes the complexity cost can be chosen for each sequence. In this work, we fix an informal threshold of 0.4dB to decide the optimal GoP-size as the MPEG committee considers that a difference of 0.5dB is visually noticeable [31]. The results of the selected GoP-size are indicated in bold. According to these results, we note that the TI index prevails over the SI one for the optimal GoP-size choice (e.g. the GoP-size=32 gives the best results for both *Akiyo* and *Container* video sequences, whereas strong SI differences exist between them);
- Likewise, we note that the CSNR value and CR level applied do not influence the GoP-size selection.

Fig. 4 gives a visual comparison between the reconstructed images for different GoP-size configurations and a CSNR = 0dB, hence validating the conclusion made above. The low PSNR and SSIM scores may catch the attention of the readers, however, we recall that in such channel quality (CSNR=0dB), classical standards (e.g. H.264/AVC) offer worse visual quality and may suffer glitches due to severe decoding errors. In contrast, *SoftCast* can deal with any channel quality, even unreliable ones, by delivering low but acceptable video quality [10].

Finally, a global synthesis for all the videos is shown in

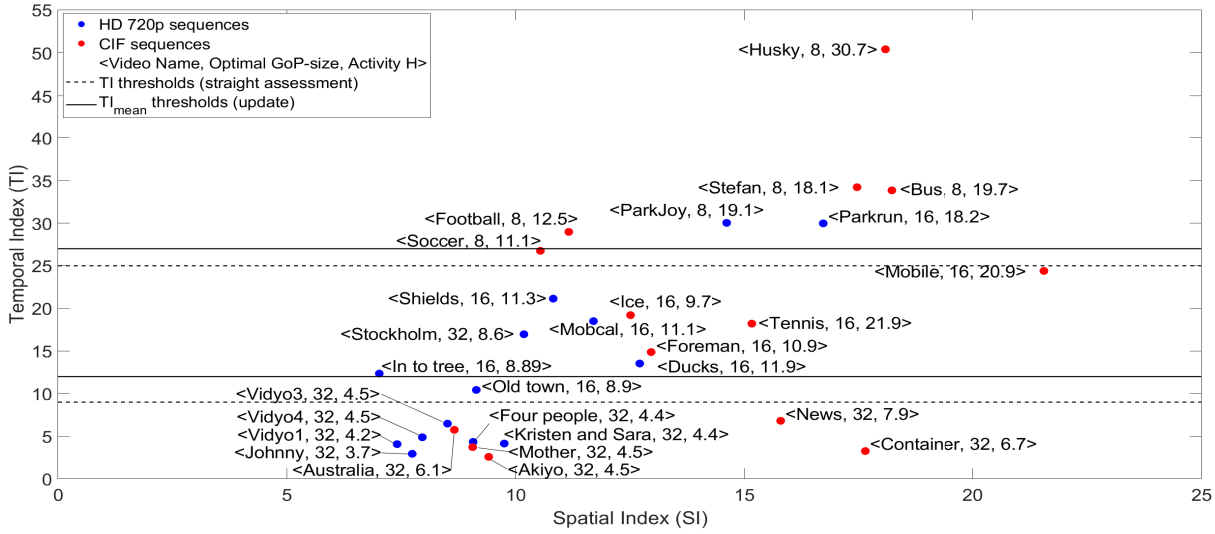


Figure 5: Illustration of the optimal GoP-size and resulting data activity H over spatio-temporal indexes for the selected CIF and HD video sequences. Red and blue dots correspond to the average values of the SI, TI indexes for the CIF and HD sequences, respectively. The label of each dot refers to the following couple data: <Video name, Optimal GoP-size, Activity H>.

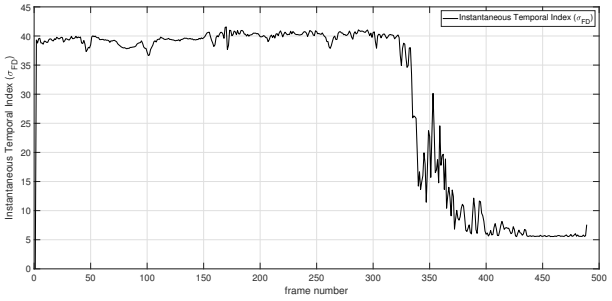


Figure 6: Illustration of the instantaneous temporal index fluctuations of the *Parkrun* video sequence.

Fig. 5 where each label indicates the video used, the optimal GoP-size as well as the resulting data activity H [13] as follows <video name, optimal GoP-size, activity H>. At a first glance, one may first establish TI thresholds (dashed lines) based on a straight assessment to classify the videos (e.g., choosing 9 and 25 as thresholds). However, mismatch for some sequences can be identified (e.g. *Parkrun*). This is due to the fact that choosing only one GoP-size for the whole sequence is insufficient due to huge temporal index fluctuations (as observed with the instantaneous TI index in Fig. 6). The instantaneous TI index is denoted by $\sigma_{\mathrm{FD}}(k)$, where $\sigma_{\mathrm{FD}}(k) = std_{space}[F_k(i,j) - F_{k-1}(i,j)]$. The index fluctuations are important and choosing only one optimal GoP-size for the whole sequence is insufficient. An optimal GoP-size = 8 can be chosen for the first 336 frames whereas a GoP-size = 32 is selected for the remaining frames to get the best reconstructed quality. To overcome this issue, we propose to locally adapt the GoP-size according to the temporal fluctuations. We define a local arithmetic mean TI_{mean} over the instantaneous TI index values that will be compared to thresholds. This local mean is evaluated each 8 frames.

Since initial thresholds are based on average results over the whole sequence, their values need to be updated and refined. In this paper, we consider empirical frame by frame analysis performed over all the sequences in Fig. 5 to fix them. These new thresholds values (12 and 27) are reported in Table 2 and visible in solid lines in Fig. 5. We note that these thresholds has been successfully tested on other video sequences (class B and C of the JCT-VC, used by the MPEG committee for the standardization of HEVC [32]) showing the validity of the proposed thresholds.

Table 2
Look-up table for the GoP-size adaptation based on threshold over the TI_{mean} indexes.

TI_{mean} threshold	Optimal GoP-size
$\mathrm{TI}_{mean} \leq 12$	32
$12 < \mathrm{TI}_{mean} < 27$	16
$\mathrm{TI}_{mean} \geq 27$	8

According to the look-up table (Table 2) and the resulting TI_{mean} value, the buffer whose size ranges between 8 and 32 frames is either emptied or filled. We choose 8 frames for each local mean computation to avoid a constant delay of 32 frames (one second for CIF sequence) before starting the encoding process. Therefore, the delay is reduced periodically to only 8 frames. For instance, if the TI_{mean} value over 8 frames is already greater than 27, we consider that these frames will not be part of a GoP of 16 or 32 frames even if the 8 next frames have a low TI_{mean} value ($\leq$12). In contrast, when the $\mathrm{TI}_{\mathrm{mean}}$ value is lower than 27, the frames remain in the analysis buffer. In such cases, The $\mathrm{TI}_{\mathrm{mean}}$ value is updated by considering 8 additional frames for the computation. The maximum number of frames that can be stored in the buffer is 32 as it is the largest selected GoP.

As stated in Section 2, we note that proposing an adaptation based on a threshold over the data activity H is not really adequate (e.g. *Mobile* and *Bus* have similar average

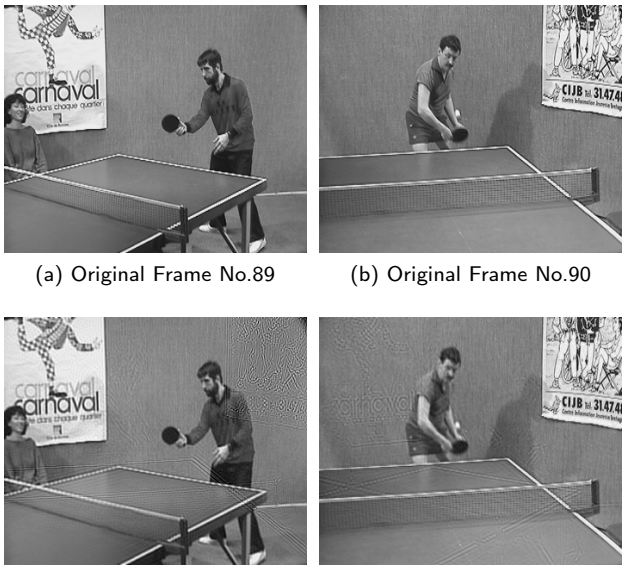
value of H, however their optimal GoP-size is respectively set to 16 and 8 frames). The data activity is actually a measure of the diversity in signal energy distribution [13] after 3D-DCT in a GoP, (i.e., after spatial and temporal decorrelation) whereas the instantaneous TI index gives a measure of the difference between two consecutive frames. Even though such adaptive GoP-size solution based on H value was proposed, it would require to first transform the video signal multiple times through 3D-DCT before finding the lowest activity. In contrast, our method, based on a frame difference measure actually gives a hint on the ability to decorrelate the signal across the temporal axis (a low σ_{FD} value means small temporal differences, therefore the signal can be well decorrelated after temporal DCT). By using the instantaneous TI values, we are able to predict the possible reduction of H, without having to calculate it. This reduction leads to an improvement of the reconstructed PSNR according to eq. (3) that is constant regardless of the CSNR value.

In this preliminary analysis, we showed that depending on the video characteristics, switching the GoP-size is an efficient way either to improve the quality at the receiver side or to decrease complexity while offering similar performances. However, we noticed that when a compression is applied over a sequence that contains shot changes or cuts (e.g. advertisement, movie/trailer and sports events), switching the GoP-size is not sufficient and may lead to suboptimal results. In the presence of shot changes and when the bandwidth is limited, SoftCast performs poorly leading to an annoying artifact: a ghost effect between the two different shots. This phenomenon is introduced and explained in the next section.

3.2. Ghost Effect Analysis

As shown in Fig. 7, the ghost effect is characterized by a superposition of the edges (high frequencies) between the frames before and after the cut. An analogous phenomenon known as *cross-fade effect* or *transparency effect* has been noticed in [33, 34] in a 3D-DCT block-based compression context. Despite some solutions to reduce the cross-fade effect were given, they are always related to a trade-off between the cross-fade effect reduction and the bitrate increase. The authors in [35] also noticed the transparency effect in a so-called accordion JPEG2000 (ACC-JPEG2000) scheme and used a change detection module based on local comparison to avoid it. However, no further details were given regarding the change detection module. Furthermore, in all of these works no theoretical clarification was introduced.

To analyze the origin of the ghost effect phenomenon, we use the linearity and separability properties of the DCT as shown in Fig. 8. Without loss of generality, we focus on a temporal DCT over 4 frames on a same spatial coordinates (i, j). For ease of notation, indexes (i, j) are omitted hereafter. Let us first denote $x = [y_1, y_2, z_3, z_4]$ the vector representing either pixels or 2D-DCT coefficients across the temporal direction at the same spatial coordinates (i, j). The coefficients $[y_1, y_2]$ and $[z_3, z_4]$ represent two different video sequences, respectively. By using the separibility we define



(a) Original Frame No.89 (b) Original Frame No.90

(c) Reconstructed Frame No.89 (d) Reconstructed Frame No.90

Figure 7: Ghost effect phenomenon, CR = 0.25, GoP-size=8. First row: Original frame No.89-90 of the *Tennis* sequence. Second row: Reconstructed frame after compression process (no transmission).

first $x \overset{\Delta}{=} y + z \overset{\Delta}{=} [y_1, y_2, 0, 0] + [0, 0, z_3, z_4]$. Likewise, let us denote by X, Y, Z, the DCT of x, y and z. Using the linearity property of the DCT, we have $X = Y + Z$.

After the compression process illustrated in red in Fig. 8 (e.g. CR=0.5), the discarded components are replaced by null values, the corresponding new vectors are denoted by $\tilde{X}, \tilde{Y}$, and $\tilde{Z}$. Using the inverse 1D-DCT process, we obtain the final vector $\tilde{x} = \mathrm{DCT}^{-1}(\tilde{X}) = \mathrm{DCT}^{-1}\left\{\tilde{Y} + \tilde{Z}\right\}$. In the same way, by considering each sequence individually, we have $\tilde{y} = \mathrm{DCT}^{-1}\left\{\tilde{Y}\right\}$ and $\tilde{z} = \mathrm{DCT}^{-1}\left\{\tilde{Z}\right\}$. Obviously, we have $\tilde{x} = \tilde{y} + \tilde{z}$ since $\tilde{X} = \tilde{Y} + \tilde{Z}$.

As demonstrated, each reconstructed pixel or 2D-DCT coefficient is actually an addition of the disturbed components of $\tilde{y}$ and $\tilde{z}$. The remaining information contained in each of the sequence after compression has been respectively spread across the four components after the inverse temporal 1D-DCT. This results in the ghost effect phenomenon when considering the whole frame.

Based on the results obtained in Section 3.1 and Section 3.2, we present in the following section, an original extension of the SoftCast scheme. First, an Adaptive GoP-size mechanism based on Content and Cut detection for SoftCast (AGCC-SoftCast) is proposed. This solution adjusts the GoP-size based on the Temporal Information index of the video content, and avoids the ghost effect phenomenon by using a scene change detection process. Nevertheless, increasing the GoP-size may lead to excessive complexity and hardware requirements (memory, processor). This may not be compatible with all applications, even though it is well known that conventional video coders are much more complex than SoftCast due to motion estimation/compensation [36]. Still, an alternative method based on the cut detec-

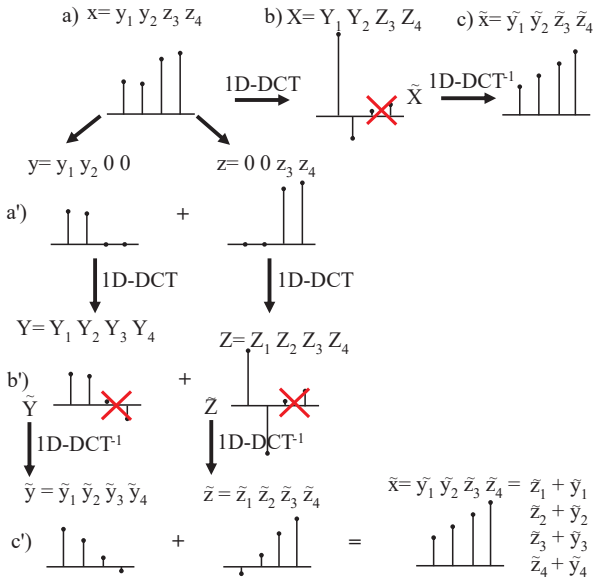


Figure 8: Illustration of the ghost effect phenomenon in the one-dimensional case considering 4 frames. a) and a') Vector of either pixels or 2D-DCT coefficients. b) and b') Resulting coefficients after temporal 1D-DCT; compression process in red. c) and c') Reconstructed coefficients after inverse 1D-DCT. The steps denoted by ') use the linearity property of the DCT.

tion only (AGCut-SoftCast) is also proposed for resources-limited applications. These two proposed methods are detailed hereafter.

3.3. Description of the proposed algorithms

As observed in Section 3.1, the TI index prevails over the SI one for the choice of the optimal GoP-size. Furthermore, the computation of the SI index requires to apply the Sobel operator for each frame, which is time and computation consuming. Consequently, the proposed methods only consider the instantaneous TI index, which is simply based on a frame difference measure.

3.3.1. Cut detection process

The first step of the AGCC-SoftCast algorithm consists in cut detection. The cut detection process is based on the instantaneous TI index $\sigma_{\mathrm{FD}}(k)$ introduced in Section 3.1.2. The instantaneous TI index takes a high value when a shot change occurs. However, high $\sigma_{\mathrm{FD}}(k)$ values may also arise due to rapid changes in a single shot (e.g., sports content). In order to avoid false detection, a moving average ($\mathrm{TI}_{mov}(k)$) is performed over the instantaneous TI values with a sliding window of 7 frames. Then, for each frame, the corresponding $\mathrm{TI}_{mov}(k)$ value is subtracted from $\sigma_{\mathrm{FD}}(k)$. The resulting signal is compared to a fixed threshold in order to detect the cuts' position. Based on extensive simulations, we set the threshold to a value of 10 to ensure a proper detection. An example of the cut detection process is given in Fig. 9. Here, the shot changes appear at frame numbers 90 and 149 and are perfectly detected by the proposed solutions.

We showed in Section 3.2 that the cut detection process is of paramount importance in a SoftCast context to avoid the ghost effect phenomenon related to shot changes. Such shot changes may appear anywhere inside the video. Therefore, it is necessary to adapt the GoP-size around the cuts' position. In the case of the AGCut solution, the GoP-size has to be adjusted only at the cut boundaries; elsewhere it is fixed and chosen equal to 8, 16 or 32 frames, depending on the available hardware resources as well as application requirements. In the case of the AGCC scheme, the GoP-size is adapted inside a shot as discussed in Section 3.1.2. This additional shot based GoP-size adaptation allows to fully benefit from the decorrelation properties of the 3D-DCT. The resulting AGCC and AGCut GoP-size decomposition is described hereafter.

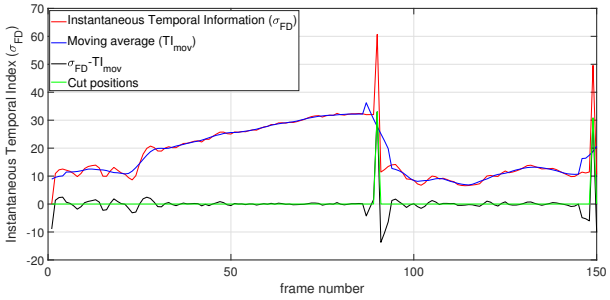


Figure 9: Example of the cut detection based on TI index for the *Tennis* sequence (cut on frames No.90, 149).

3.3.2. GoP-size decomposition

In what follows, we reasonably assume that at least 8 frames separate two consecutive cuts. Let us first consider the AGCut method, based on GoP-size of 8 frames. The resulting GoP-size after cut detection ranges between 9 and 15 frames. The AGCut method requires a reasonable buffer of 15 frames that induces a maximum resulting delay of 15 frames (e.g., half a second for 30fps video sequence). Of course, depending on the hardware capacities and the targeted delay, the GoP-size basis can be extended to 16 or 32 frames. Finally, the AGCC method allows to adapt dynamically the GoP-size inside a shot, hence offering the best trade-off between visual quality improvement and complexity cost. In that case, the GoP-size may vary between 8 and 39 frames.

Fig. 10 summarizes the different approaches described above. In this figure, a shot change happens between two sequences. For instance and without loss of generality, let us assume that the sequence A and B correspond respectively to the *Akiyo* and *Husky* sequences. With the classical SoftCast scheme, the cuts as well as the characteristics of the video are not taken into account. However, with the proposed cut detection method (AGCut-SoftCast), the last GoP-size for both sequences consists of $8 + N_1$ frames due to the cut detection. N_1 denotes the cut's position that ranges between 1 and 7 assuming that at least 8 frames separate two consecutive cuts. In addition, intra-shot local GoP-size adaptation is performed in the AGCC-SoftCast extension, resulting in a last GoP of $32 + N_1$ and $8 + N_1$ frames for the *Akiyo* and *Husky* sequences, respectively.

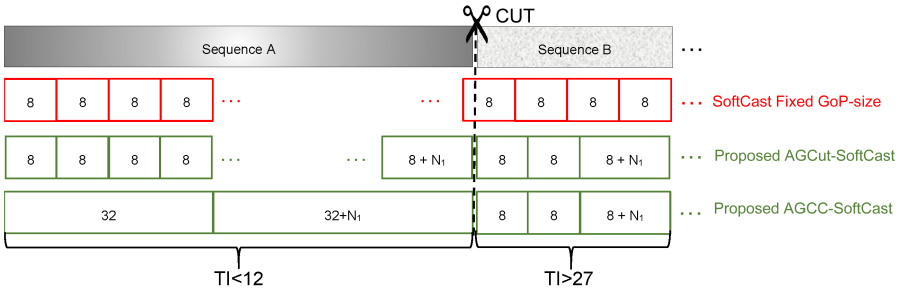


Figure 10: Illustration of the proposed adaptive GoP-size methods.

Table 3
Characteristics of the Mixed video test sequences

Subsequence	$Mixed_{CIF}$ video sequence Number of frames, (Resulting AGCC decomposition)
News	30, (30)
Husky	22, (8+14)
Mother	64, (2x32)
Stefan	40, (2x8+24)
Australia	64, (2x32)
Akiyo	70, (32+38)
Container	66, (32+34)
Mobile	31, (16+15)
Foreman	54, (32+22)
Football	39, (3x8+15)
Subsequence	$Mixed_{HD}$ video sequence Number of frames, (Resulting AGCC decomposition)
Parkjoy	75, (8x8+11)
Johnny	94, (2x32+30)
In to tree	97, (2x32+33)
Parkrun	71, (7x8+15)
Kristen and Sara	125, (3x32+29)
Shields	105, (6x16+9)
Vidyo3	88, (2x32+24)
*Parkrun**	69, (32+37)
Stockholm	76, (4x16+12)

4. Simulation results

The simulation setup remains mostly the same as the one used in Section 3.1. The changes only concern the video sequences which are described below:

Video sources: Two mixed video sequences are randomly generated in this section to evaluate the proposed extension. They are denoted by $Mixed_{CIF}$ and $Mixed_{HD}$ sequences, respectively. The sequences used as well as the resulting number of frames for each sequence are given in Table 3. They have been chosen to cover a large portion of the SI, TI map (see Fig. 3 in Section 3.1). We note that two parts of *Parkrun* have been incorporated for the $Mixed_{HD}$ sequence. This is to illustrate the possible temporal fluctuations inside a same sequence as previously shown in Fig. 6. Specifically, the first 71 frames are used as well as 69 others corresponding to the end of the video (frames No.400∼469 denoted by *Parkrun**).

Finally, the resulting AGCC GoP-size decomposition is also given in brackets in Table 3. As observed, both cuts and temporal fluctuations are perfectly detected: For each subsequence, the optimal GoP-sizes are found and the last one is adjusted to avoid the ghost effect.

The proposed methods are compared with the classical SoftCast scheme assuming three standards and fixed GoP-sizes (8, 16 and 32 frames) [8, 13, 14].

4.1. Frame by frame analysis

4.1.1. Full transmission

Fig. 11 and Fig. 12 represent the per frame evolution of the PSNR and SSIM scores considering no bandwidth restriction, i.e., CR=1. Vertical dashed lines represent the cuts' position. We choose here an intermediate CSNR (CSNR=15dB). Similar results for other CSNR values are obtained. Regardless of the metric used, results show that:

- As explained by Xiong *et al.* [13] in Section 2, under the same channel conditions, the received quality directly depends on the data activity $\boldsymbol{H}$ (see eq. (3)). Therefore, one video containing high temporal fluctuations (e.g., *Husky*) is more difficult to transmit than a video with low temporal fluctuations (e.g., *Akiyo*) resulting in a lower reconstructed quality;
- As explained in Section 3.1, for the video with low spatio-temporal activity (e.g., *Container*), a large GoP allows to obtain a better reconstructed quality at the receiver side. In the opposite case, (i.e., *Husky*), using a small GoP-size allows to reduce the complexity with a comparable received quality;
- Even without the ghost effect phenomenon (since CR=1), we observe that the PSNR may become suboptimal for fixed GoP-size solutions. This is particularly noticeable for the *News* sequence, where the optimal GoP-size should be 32 frames (blue curve). Furthermore, using fixed GoP-size method leads to drastic quality fluctuations (e.g. the blue curve for the *Mother* sequence). In contrast, the proposed methods AGCut-SoftCast and AGCC-SoftCast provide a quality that stays relatively constant for each subsequence;
- The proposed method AGCC-SoftCast mostly provides the best received quality since it benefits from

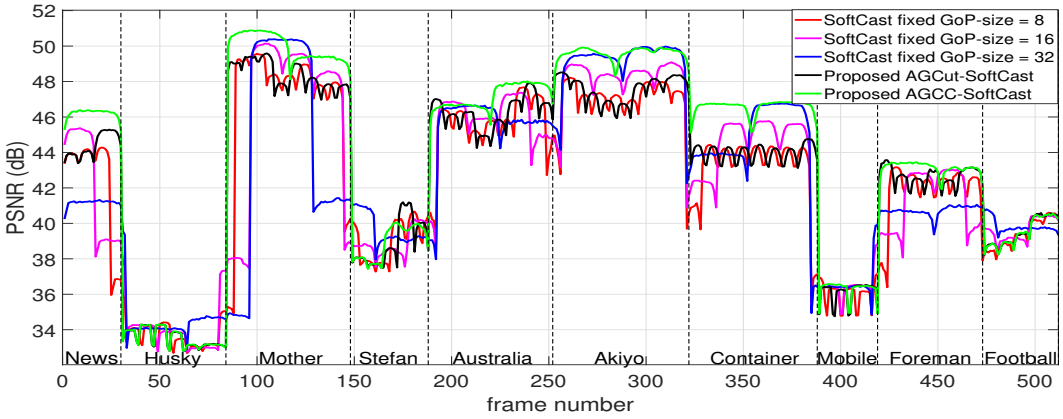


Figure 11: PSNR_{dB} per frame for the $Mixed_{CIF}$ sequence, CSNR=15dB, CR=1 (no compression applied).

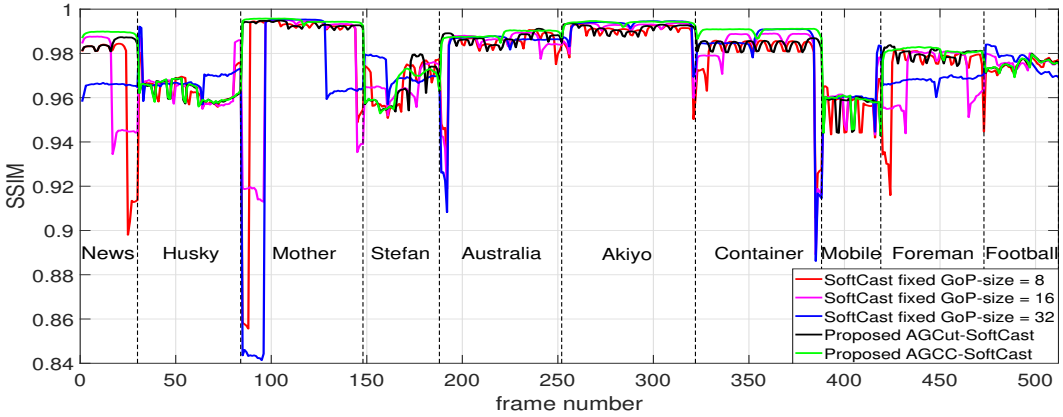


Figure 12: SSIM per frame for the $Mixed_{CIF}$ sequence, CSNR=15dB, CR=1 (no compression applied).

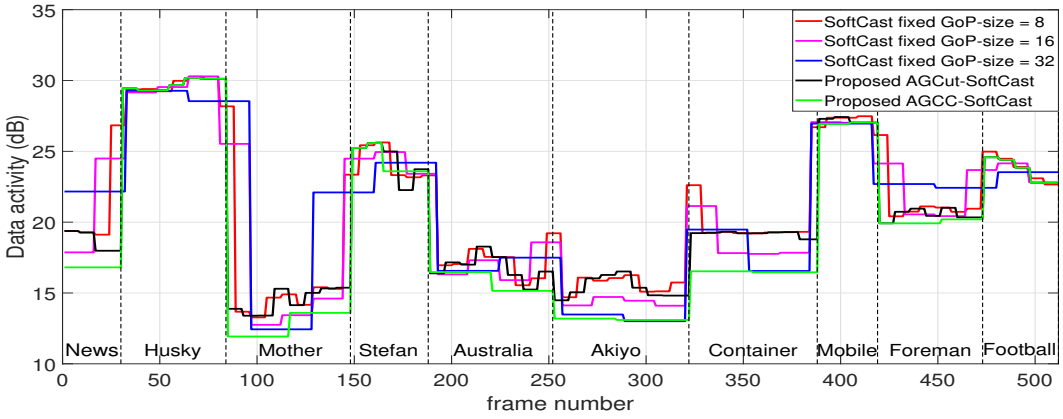


Figure 13: Data Activity H per GoP expressed in decibels for the $Mixed_{CIF}$ sequence.

both cut and content detection processes. However, readers may notice some frames where the quality given by the fixed GoP-size solutions is better.

We give clarifications about the two last items mentioned above. To do so, we use eq. (2) proposed by Xiong *et al.* Fig. 13 represents the data activity H expressed in decibels as used in eq. (3) and displayed in Fig. 13. We first recall that a low value means that the received quality (in terms of PSNR) will be higher since it is subtracted from $c + \text{CSNR}$ with c=20log$_{10}$(255). We also recall that as observed in the figure, H is constant over a GoP since it is an indicator of the received quality at the GoP level (due to 3D-DCT).

As observed, for a shot transition, when a fixed GoP is used, the resulting data activity is actually a mix of the two activities of the videos during the GoP that contains the cut. Therefore, for the fixed GoP-size = 32 frames (blue curve), the data activity of *News* (30 frames) and *Husky* sequences (2 frames) are mixed together, resulting in an intermediate value. This mix is beneficial for the 2 first frames of *Husky* as it increases the received quality as shown in Fig. 11, Fig. 12 and Fig. 13. However, it drastically reduces the visual quality of the 30 frames of the *News* sequence. Even if the fixed

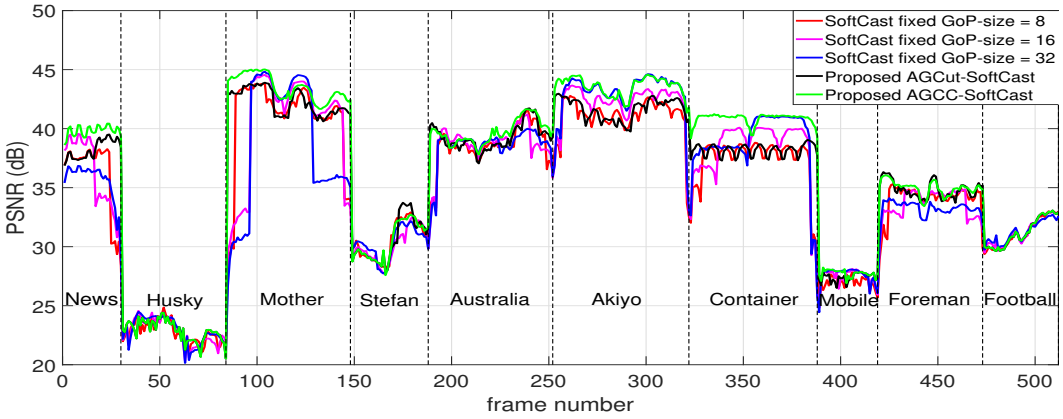


Figure 14: PSNR_{dB} per frame for the $Mixed_{CIF}$ sequence, CSNR=15dB, CR=0.25 (75% of discarded coefficients).

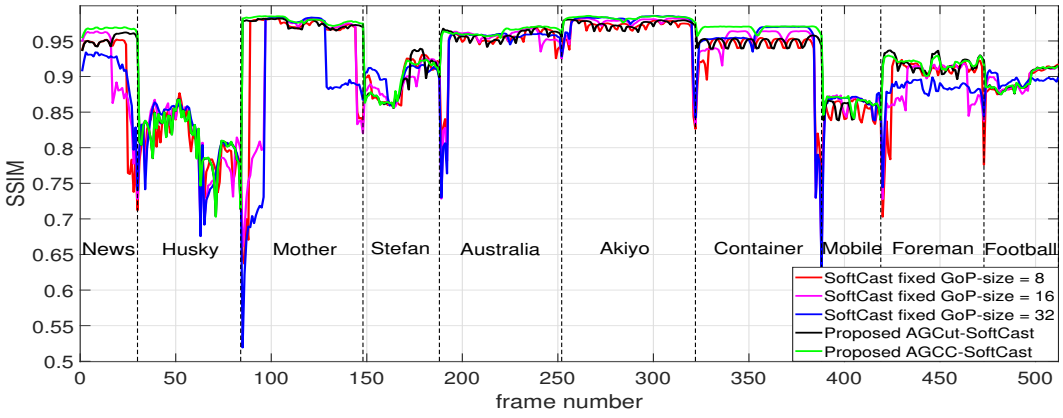


Figure 15: SSIM per frame for the $Mixed_{CIF}$ sequence, CSNR=15dB, CR=0.25 (75% of discarded coefficients).

GoP-size solution gives sometimes better results, it causes a drastic loss of quality for low TI content and works only when the GoP contains a cut.

4.1.2. Bandwidth-constrained transmission

The bandwidth-constrained environments are now considered in Fig. 14 and Fig. 15, where 75% of the coefficients are discarded (CR=0.25). Results show that:

- The occasional quality improvements obtained for the fixed GoP-size methods at the cut boundaries are greatly attenuated since the ghost effect appears;
- The ghost effect is spread along the duration of the GoP. Therefore, shot changes must be taken into account as a larger GoP increases the duration of the disturbed frames.

A synthesis of the PSNR and SSIM scores for each subsequence is available in Table 4. The two best quality scores per subsequence have been highlighted as well as the gains between the proposed AGCC-SoftCast method and the fixed GoP-size of 16 or 32 frames. In addition, the σ_{PSNR} scores are computed to evaluate the PSNR fluctuations inside each subsequence. Indeed, numerous studies [20, 37, 38] reported that the quality fluctuations are annoying for the user and should be considered as an index of the reconstructed quality as well.

The σ_{PSNR} score is defined as

$$\sigma_{\text{PSNR}} = \sqrt{\frac{1}{F}\sum_{k=1}^{F}(\text{PSNR}_k - \text{PSNR}_{avg})^2}, \tag{8}$$

where F represents the number of frames in the considered subsequence, PSNR_k and PSNR_{avg} are respectively the PSNR score of the k^{th} frame and the average PSNR score over the considered subsequence.

As shown in this table:

- The proposed solutions guarantee a better overall quality with limited PSNR fluctuations compared to the classical fixed GoP-size solutions. Regardless of the compression applied, the σ_{PSNR} scores for the AGCC-SoftCast and AGCut-SoftCast extensions always remain below 1dB in average;
- In contrast, for the fixed GoP-size solutions, large fluctuations about 2.5-3.5dB and up to 6dB are observed due to mix between sequences inside a GoP;
- It is interesting to note that the proposed AGCut-SoftCast method (based on GoP-size of 8 frames) globally performs better than the fixed GoP of 32 frames according to visual quality scores for the same reason mentioned above;

Table 4
Table of the resulting PSNR, σ_{PSNR} and SSIM scores for the $Mixed_{CIF}$ sequence, for a CSNR=15dB and considering different GoP-size configurations and compression ratios: CR = 1; CR = 0.25.

Simulation Setup			Piece of video sequence									
			News	*Husky*	*Mother*	*Stefan*	*Australia*	*Akiyo*	*Container*	*Mobile*	*Foreman*	*Football*
CSNR=15dB CR=1	PSNR(dB)	GoP-size=8	41.17	33.66	44.16	**38.89**	45.09	46.52	42.55	35.99	41.43	39.34
		GoP-size=16	41.17	**33.86**	43.18	38.84	45.18	47.59	43.05	**36.41**	41.03	**39.62**
		GoP-size=32	41.16	**34.33**	40.43	**39.53**	44.98	**48.62**	43.45	**36.57**	40.71	**39.80**
		AGCut	**44.35**	33.51	**48.31**	38.70	**46.14**	47.49	**43.95**	36.05	**42.62**	39.45
		AGCC	**46.19**	33.51	**49.87**	38.81	**47.09**	**49.58**	**46.53**	36.24	**43.12**	39.43
		Gain AGCC/16	**5.01**	**-0.35**	**6.69**	**-0.03**	**1.92**	**1.99**	**3.48**	**-0.17**	**2.08**	**-0.19**
		Gain AGCC/32	**5.03**	**-0.82**	**9.44**	**-0.73**	**2.12**	**0.96**	**3.09**	**-0.33**	**2.40**	**-0.37**
	σ_{PSNR}(dB)	GoP-size=8	2.93	0.77	3.75	1.17	1.76	1.51	1.93	0.69	1.64	0.81
		GoP-size=16	3.11	1.32	4.83	**0.77**	1.85	1.61	2.44	0.81	1.66	**0.63**
		GoP-size=32	**0.20**	1.06	6.11	**0.87**	1.85	1.59	2.57	1.02	**0.28**	**0.38**
		AGCut	0.67	**0.52**	**0.79**	1.28	**0.88**	**0.73**	**0.46**	**0.61**	0.52	0.79
		AGCC	**0.29**	**0.52**	**0.78**	0.96	**0.69**	**0.35**	**0.44**	**0.52**	**0.35**	0.77
	SSIM	GoP-size=8	0.968	0.964	0.982	**0.966**	0.983	0.990	0.979	0.957	0.973	0.975
		GoP-size=16	0.967	**0.965**	0.976	**0.966**	0.983	0.991	0.981	**0.960**	0.970	**0.976**
		GoP-size=32	0.966	**0.968**	0.957	**0.971**	0.982	**0.993**	0.983	**0.962**	0.968	**0.977**
		AGCut	0.985	0.963	**0.993**	0.965	**0.987**	0.991	**0.984**	0.956	**0.980**	0.975
		AGCC	**0.989**	0.963	**0.995**	0.965	**0.989**	**0.994**	**0.991**	0.958	**0.981**	0.975
		Gain AGCC/16	**0.023**	**-0.003**	**0.019**	**-0.001**	**0.005**	**0.003**	**0.009**	**-0.003**	**0.011**	**-0.001**
		Gain AGCC/32	**0.024**	**-0.006**	**0.038**	**-0.006**	**0.006**	**0.001**	**0.008**	**-0.004**	**0.013**	**-0.001**
CSNR=15dB CR=0.25	PSNR(dB)	GoP-size=8	34.63	22.63	38.37	30.23	38.22	40.53	36.47	**26.89**	33.73	31.01
		GoP-size=16	34.97	22.63	37.27	29.90	38.21	41.47	36.66	27.36	33.49	31.12
		GoP-size=32	35.09	22.72	35.32	30.10	38.12	**42.26**	37.00	27.49	33.18	**31.18**
		AGCut	**38.17**	**22.79**	**42.17**	**30.18**	**39.16**	41.57	**38.17**	27.39	**34.76**	31.06
		AGCC	**39.78**	**22.79**	**43.35**	**30.18**	**39.61**	**43.50**	**40.81**	**27.72**	**34.93**	**31.06**
		Gain AGCC/16	**4.81**	**0.15**	**6.07**	**0.27**	**1.40**	**2.03**	**4.16**	**0.36**	**1.44**	**-0.06**
		Gain AGCC/32	**4.69**	**0.07**	**8.02**	**0.07**	**1.48**	**1.24**	**3.82**	**0.23**	**1.75**	**-0.11**
	σ_{PSNR}(dB)	GoP-size=8	3.18	0.94	3.65	1.63	1.72	1.72	2.19	0.58	1.36	1.23
		GoP-size=16	3.12	1.11	4.66	**1.43**	1.70	1.88	2.83	0.73	1.16	**1.16**
		GoP-size=32	1.74	1.13	5.53	**1.31**	1.58	1.98	2.95	0.78	**0.65**	**0.91**
		AGCut	**0.80**	**0.88**	**1.10**	1.83	**1.08**	**0.91**	**0.48**	**0.38**	0.73	1.21
		AGCC	**0.46**	**0.89**	**1.07**	1.60	**0.99**	**0.80**	**0.48**	**0.32**	**0.59**	1.21
	SSIM	GoP-size=8	0.910	0.808	0.946	**0.896**	0.947	0.968	0.935	0.852	0.898	0.894
		GoP-size=16	0.908	0.808	0.928	0.892	0.947	0.974	0.940	0.862	0.888	**0.896**
		GoP-size=32	0.906	**0.812**	0.895	**0.900**	0.946	**0.978**	0.946	**0.864**	0.881	**0.895**
		AGCut	**0.953**	**0.812**	**0.974**	0.893	**0.958**	0.974	**0.948**	0.858	**0.918**	**0.895**
		AGCC	**0.967**	**0.812**	**0.979**	0.893	**0.963**	**0.983**	**0.969**	**0.864**	**0.919**	**0.895**
		Gain AGCC/16	**0.059**	**0.003**	**0.051**	**0.001**	**0.017**	**0.009**	**0.028**	**0.002**	**0.031**	**-0.001**
		Gain AGCC/32	**0.060**	**-0.001**	**0.084**	**-0.007**	**0.017**	**0.005**	**0.023**	**0.000**	**0.039**	**0.000**

- The obtained SSIM gains have relatively low numerical values. This is due to the limited dynamic of the metric at high quality level (for CSNR>15dB, SSIM scores are already >0.97). Nevertheless, we observe PSNR gains up to 8dB depending on the transmitted video content and the fixed GoP-size considered, showing the improvement of the received quality;
- As an additional performance indicator, we evaluate the time percentage during which the video quality obtained with the proposed AGCC-SoftCast method is better or equal than the fixed GoP-size method. We use the informal PSNR threshold of ±0.4dB [31] to decide whether the video is better, equal or lower in terms of reconstructed quality. In comparison to the fixed GoP-size of 32 frames, the proposed AGCC-SoftCast performs better for more than 82% and 72% of the time for the $Mixed_{CIF}$ and $Mixed_{HD}$ sequences, respectively. These percentages increase up to 89% as the CR decreases due to ghost effect appearance. These percentages range between 90% to 94% when considering the fixed GoP-size = 16, showing the effectiveness of the proposed AGCC extension.

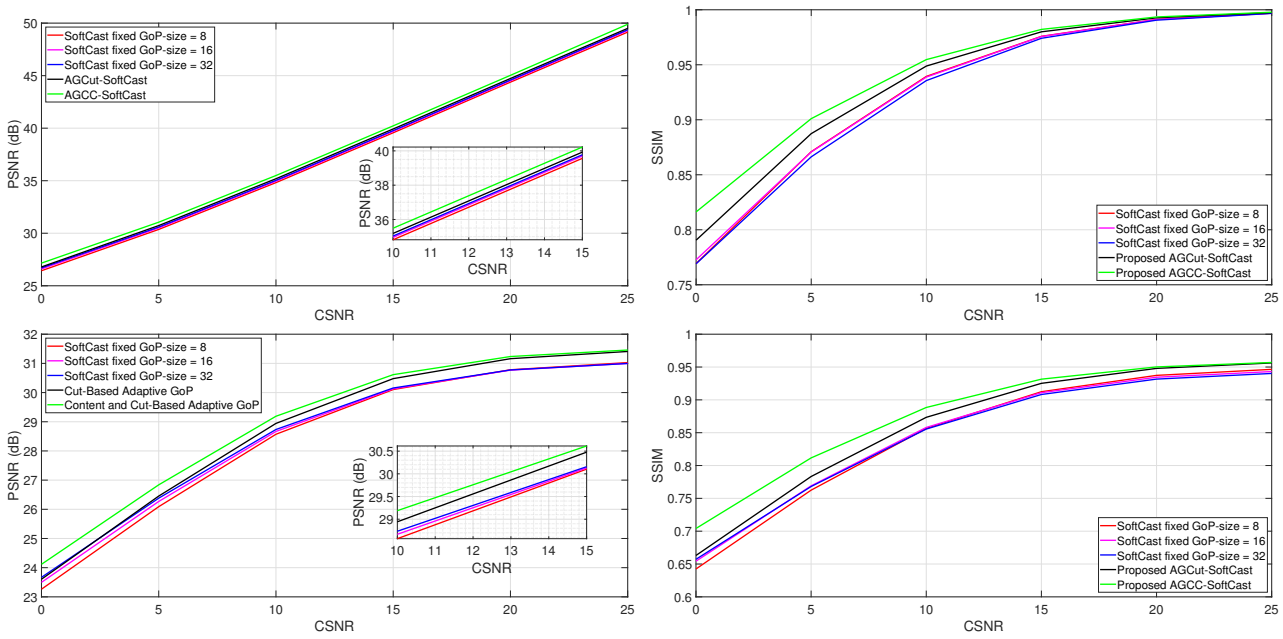


Figure 16: Average quality scores vs CSNR. First column: PSNR results. Second column: SSIM results. First row: CR=1. Second row: CR=0,25.

4.2. Global performances

To better see the contribution of the proposed approaches and the difference between the fixed GoP-size methods, the average PSNR and SSIM scores are displayed over a large range of CSNR (0∼25dB) in Fig. 16. Regardless of the compression ratio applied, results show that:

- The proposed methods always provide better average results than the classical SoftCast scheme with fixed GoP-size. The average gain is about 1dB in terms of PSNR score and 0.05 in terms of SSIM index;
- We note that the improvement of the SSIM index is higher for low CSNR values (up to 0.1). At high CSNR values, the noise becomes insignificant thus, the received video quality approaches the maximum quality that is available after compression;
- When a compression ratio is applied, as shown in the bottom left figure in Fig. 16, the linearity of SoftCast is broken due to unrecoverable loss of information. This phenomenon is known as the levelling-off effect [5];
- Since the shot changes have not been considered in the fixed GoP-size scenarios, the quality improvement behaviors between 8, 16 and 32 frames previously observed in Section 3.1 become here insignificant (<0.5dB), showing the effectiveness of the proposed methods.

4.3. Visual comparison

A visual comparison is given in Fig. 17 and Fig. 18 to evaluate the reconstructed frames for different methods. Due to space limitations, only the CIF sequences results are displayed. The fixed GoP-size of 32 frames is chosen as reference since it usually gives the best performances [13]. The GoP-size basis used for AGCut-SoftCast is 8 frames. In the present case, we choose an intermediate channel quality (CSNR = 15dB). Similar conclusions are obtained for other CSNR values and for the $Mixed_{HD}$ sequence.

4.3.1. Full transmission

We first compare the results considering no bandwidth restriction (Fig. 17). The middle rows represent one cut boundary of the $Mixed_{CIF}$ (frames No.388-389). As observed, the proposed methods give better results for the low TI content (i.e., *Container*). However, as explained before, the fixed GoP-size benefits from the *Container* sequence to improve a few frames of the *Mobile* sequence. To show that this effect only depends on the duration of the GoP containing a cut, we also display in the first and fourth rows, the results obtained for adjacent GoPs (i.e., 28^{th} frames before and after the cut). As observed, the *Container* sequence suffers from severe degradations and temporal fluctuations of quality (>10dB in terms of PSNR score) whereas the proposed methods stay relatively constant. Since the total number of frames for the *Mobile* sequence equals 31 frames (randomly generated number), there are not enough frames to encode a GoP of 32 frames without including shot changes. Therefore, the frame No.417 also benefits from the low TI content of the next video i.e., *Foreman*, explaining the fact that both frames No.389 and 417 have higher received quality than the proposed methods.

4.3.2. Bandwidth-constrained transmission

When the available bandwidth for transmission is limited (Fig. 18, CR=0.25), we can clearly observe that the classical SoftCast gives the lowest received video quality, regardless of the GoP-size considered. In contrast, the proposed extensions achieve better video quality under the same channel characteristics. The PSNR difference between the

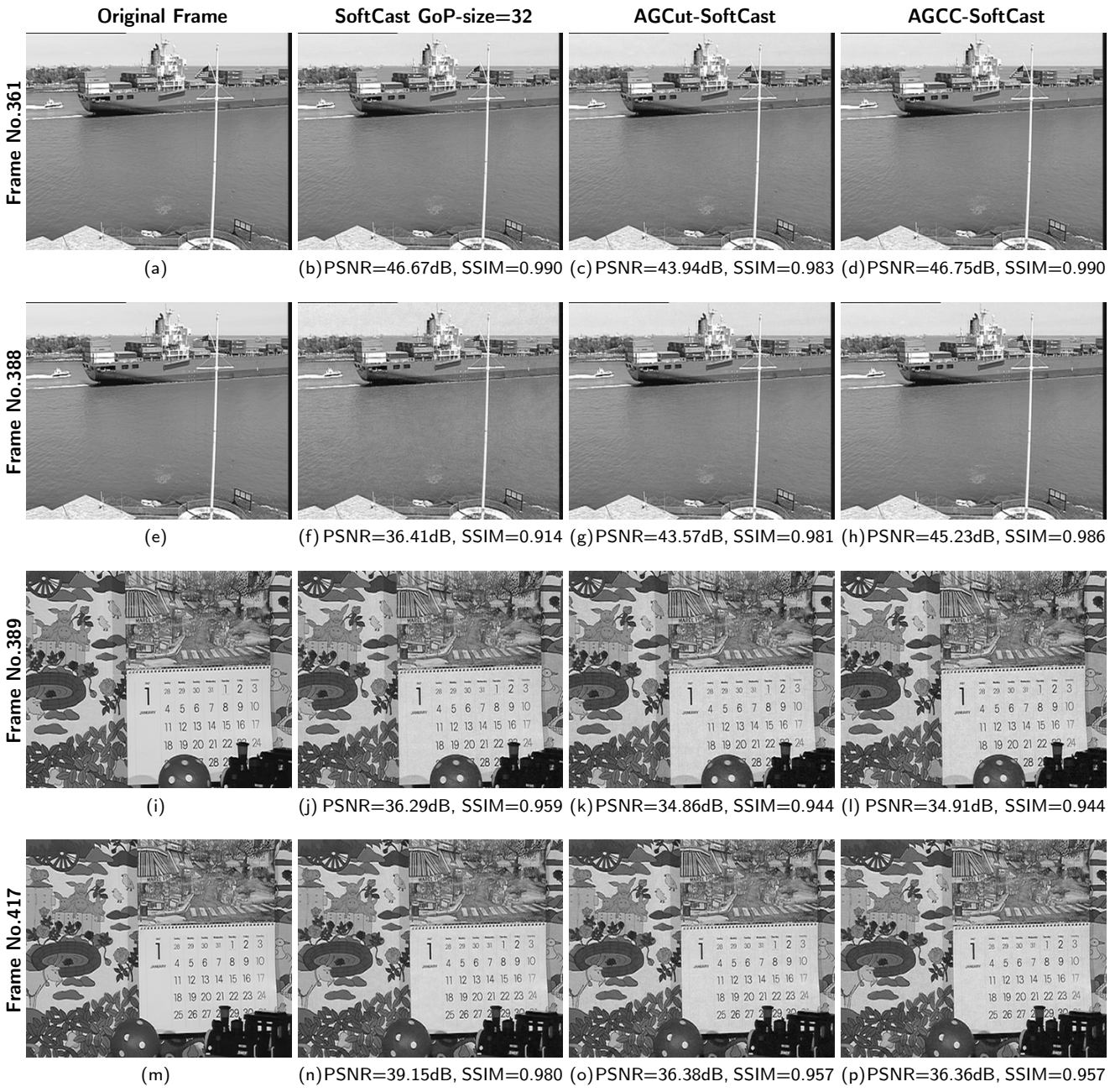


Figure 17: Visual quality comparison at a CSNR = 15dB, CR = 1 for the $Mixed_{CIF}$ sequence (Frames No.361, 388, 389, 417). (a),(b),(c),(d): Frame No.361 = *Container* frame (28 frames before cut). (e),(f),(g),(h): Frame No.388 = *Container* frame (last frame before cut). (i),(j),(k),(l): Frame No.389 = *Mobile* frame (first frame after cut). (m),(n),(o),(p): Frame No.417 = *Mobile* frame (28 frames after cut). (a),(e),(l),(m): Original frame. (b),(f),(j),(n): SoftCast fixed GoP-size of 32 frames. (c),(g),(k),(o): AGCut-SoftCast (GoP-size basis = 8 frames). (d),(h),(l),(p): AGCC-SoftCast.

classical solutions and the proposed ones is up to 16dB in terms of PSNR scores at the cut boundaries. When no cut is detected, the AGCC-SoftCast and the fixed GoP-size of 32 frames perform similar since they are all based on GoP-size of 32 frames. However, AGCC-SoftCast reduces periodically the GoP-size to reduce the complexity, leaving the hardware (processor, RAM, etc.) available for others tasks.

To evaluate the global complexity of the proposed methods we evaluate the amount of GoP per GoP-size needed to transmit the $Mixed_{CIF}$ video sequence. We verified that the proposed AGCut method represents a good trade-off between received quality and complexity cost since it mainly uses GoP-size of 8 frames (52 GoP of 8 frames, 7 others GoP-sizes ranges from 10 to 15). When hardware capacities allow to use larger GoPs, then the proposed AGCC-SoftCast extension gives the best reconstructed quality with a good trade-off regarding complexity cost (7, 1, 10 and 9 GoPs of respectively 32, 16, 8 and 14∼38 frames). We obtained similar results for the $Mixed_{HD}$ video sequence.

The results presented in this section clearly underline the

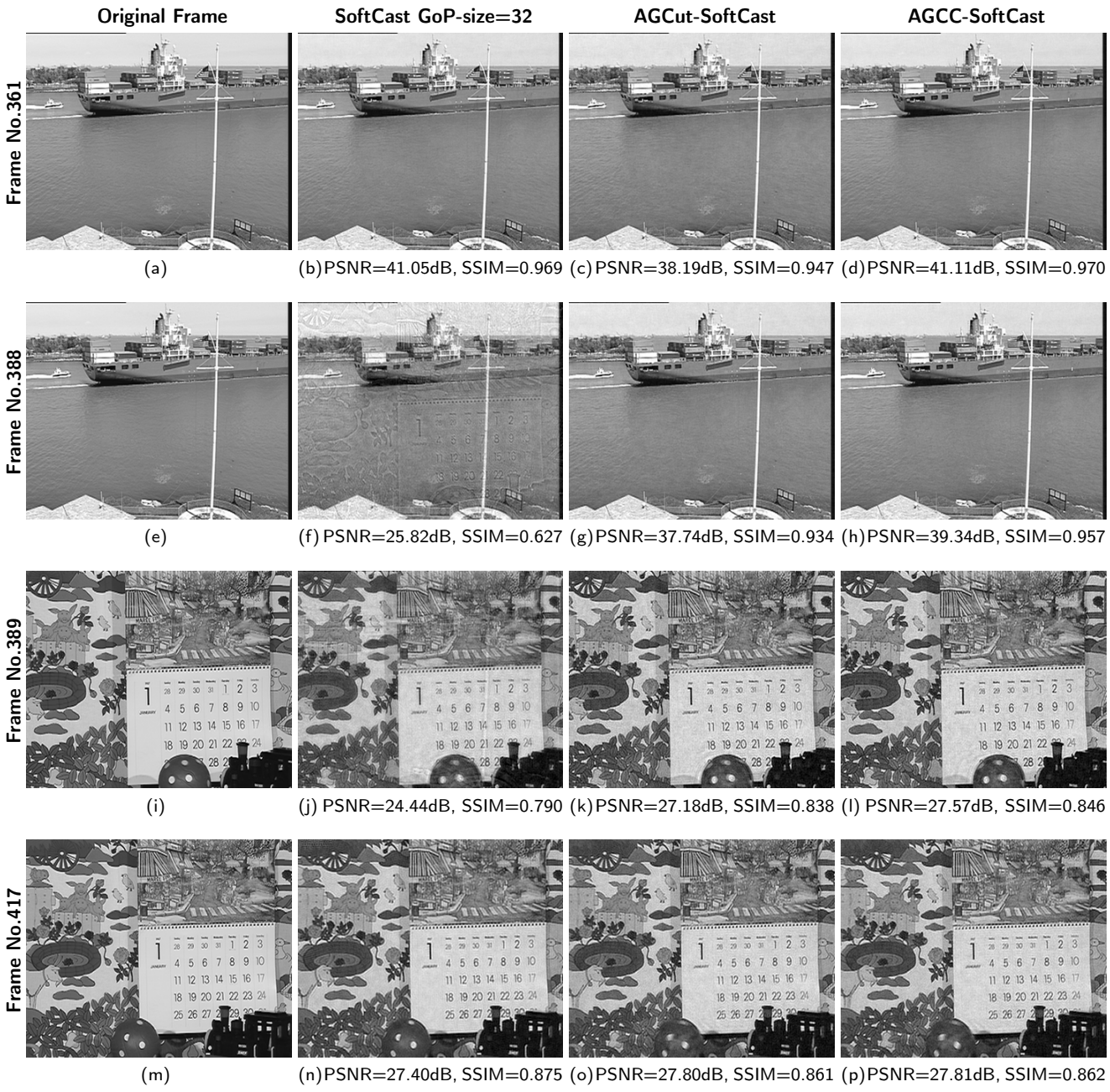


Figure 18: Visual quality comparison at a CSNR = 15dB, CR = 0.25 for the $Mixed_{CIF}$ sequence (Frames No.361, 388, 389, 417). (a),(b),(c),(d): Frame No.361 = *Container* frame (28 frames before cut). (e),(f),(g),(h): Frame No.388 = *Container* frame (last frame before cut). (i),(j),(k),(l): Frame No.389 = *Mobile* frame (first frame after cut). (m),(n),(o),(p): Frame No.417 = *Mobile* frame (28 frames after cut). (a),(e),(l),(m): Original frame. (b),(f),(j),(n): SoftCast fixed GoP-size of 32 frames. (c),(g),(k),(o): AGCut-SoftCast (GoP-size basis = 8 frames). (d),(h),(l),(p): AGCC-SoftCast.

benefits of the two proposed methods. Regardless of the channel conditions, the received quality is significantly improved while quality fluctuations are greatly reduced.

5. Conclusion

In this work, we evaluate and optimize the performances of the SoftCast scheme by analyzing the temporal fluctuations of the transmitted video content. A first analysis shows that, it is of paramount importance to take into account the spatio-temporal characteristics of the video in a SoftCast context. Depending on these characteristics, switching the GoP-size is an efficient way to either improve the quality at the receiver side or to decrease complexity while offering similar performances. We also highlight the fact that using a fixed GoP-size as classically done in a SoftCast context leads to severe visual quality fluctuations (about 2.5-3.5dB for the PSNR scores) as well as annoying ghost effect artifacts when the available bandwidth is limited. We show that these impairments can be annihilated (ghost effect) or

greatly reduced (quality fluctuations) by ensuring that no GoP contains two different shots. Based on these results, an extension of SoftCast is proposed namely, an Adaptive GoP-size mechanism based on Content and Cut detection for SoftCast (AGCC-SoftCast). For resources-limited applications, this mechanism is reduced to the cut detection only (AGCut-SoftCast). An improvement in terms of PSNR score up to 16 dB and up to 0.55 for SSIM score can be observed with the proposed methods at the cut boundaries. In addition, visual quality fluctuations are reduced under 1dB in average, showing the effectiveness of the proposed methods. The proposed AGCut-SoftCast with a GoP-size basis of 8 frames allows to maintains good performances while reducing the complexity cost as well as the needed hardware requirements. Finally, the AGCC method performs local GoP-size adaptation inside a shot, hence offering the best trade-off between visual quality improvement and complexity cost with improvements in terms of PSNR scores up to 2.6dB compared to the AGCut-SoftCast solution.

6. Acknowledgment

This work was supported by the Université Polytechnique Hauts-de-France. We would like to thank the anonymous reviewers for their valuable comments that helped to write this journal paper. We also would like to thank Dr. Matteo NACCARI for making available the Matlab code for the computation of the spatio-temporal indexes [39].

References

[1] Cisco, Cisco visual networking index: Forecast and trends, 2017-2022.

[2] I. E. G. Richardson, The H.264 advanced video compression standard, 2nd Edition, Wiley, Chichester, 2010.

[3] G. J. Sullivan, J. R. Ohm, W. J. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) Standard, IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) 1649–1668. doi:10.1109/TCSVT.2012.2221191.

[4] S. Kokalj-Filipovic, E. Soljanin, Suppressing the cliff effect in video reproduction quality, Bell Labs Technical Journal 16 (4) (2012) 171–185. doi:10.1002/bltj.20540.

[5] F. Liang, C. Luo, R. Xiong, W. Zeng, F. Wu, Superimposed Modulation for Soft Video Delivery with Hidden Resources, IEEE Trans. Circuits Systems Video Technol. 28 (9) (2018) 2345–2358.

[6] H. Schwarz, D. Marpe, T. Wiegand, Overview of the Scalable Video Coding Extension of the H.264/AVC Standard, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1103–1120.

[7] R. Xiong, J. Zhang, F. Wu, J. Xu, W. Gao, Power Distortion Optimization for Uncoded Linear Transformed Transmission of Images and Videos, IEEE Transactions on Image Processing 26 (1) (2017) 222–236.

[8] S. Jakubczak, D. Katabi, SoftCast: Clean-slate scalable wireless video, MIT Technical report.

[9] A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, Etude de l'influence du contenu vidéo dans une transmission pseudo-analogique de type Softcast, in: 20 ème édition du colloque COmpression et REprésentation des Signaux Audiovisuels (CORESA), 2018.

[10] S. Jakubczak, D. Katabi, A cross-layer design for scalable mobile video, in: Proc. of the 17th annual international conference on Mobile computing and networking (MobiCom), 2011, pp. 289–300.

[11] A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, A comparative preprocessing study for softcast video transmission, in: Proc. IEEE International Symposium on Signal Image and Video Communications (ISIVC), 2018.

[12] Z. Li, H. Lu, Y. Wu, Compressed uncoded screen content video transmission in bandwidth-constrained wireless networks, in: Proc. IEEE International Conference on Wireless Communications & Signal Processing (WCSP), 2016, pp. 1–5.

[13] R. Xiong, F. Wu, J. Xu, X. Fan, C. Luo, W. Gao, Analysis of decorrelation transform gain for uncoded wireless image and video communication, IEEE Transactions on Image Processing 25 (4) (2016) 1820–1833.

[14] T. Fujihashi, T. Koike-Akino, T. Watanabe, P. V. Orlik, High-Quality Soft Video Delivery With GMRF-Based Overhead Reduction, IEEE Transactions on Multimedia 20 (2) (2018) 473–483.

[15] S. Zheng, M. Cagnazzo, M. Kieffer, Optimal and suboptimal channel precoding and decoding matrices for linear video coding, Signal Processing: Image Communication 78 (2019) 135–151. doi:10.1016/j.image.2019.06.011.

[16] D. He, C. Luo, C. Lan, F. Wu, W. Zeng, Structure-preserving hybrid digital-analog video delivery in wireless networks, IEEE Transactions on Multimedia 17 (9) (2015) 1658–1670.

[17] J. Shen, L. Yu, L. Li, H. Li, Foveation Based Wireless Soft Image Delivery, IEEE Trans. Multimedia 20 (10) (2018) 2788–2800.

[18] D. Yang, Y. Bi, Z. Si, Z. He, K. Niu, Performance evaluation and parameter optimization of SoftCast wireless video broadcast, in: Proc. of the 8th International Conference on Mobile Multimedia Communications (MobiMedia), 2015, pp. 79–84.

[19] B. Zatt, M. Porto, J. Scharcanski, S. Bampi, GoP structure adaptive to the video content for efficient H.264/AVC encoding, in: 2010 IEEE International Conference on Image Processing, 2010, pp. 3053–3056.

[20] S.-C. Hsia, S.-H. Wang, High-performance adaptive group-of-picture rate control for H.264/AVC, Signal, Image and Video Processing 5 (2) (2011) 155–163. doi:10.1007/s11760-009-0150-3.

[21] T. N. T. Huong, H. P. Cong, T. V. Huu, X. H. Van, Artificial Intelligence Based Adaptive GOP Size Selection for Effective Wyner-Ziv Video Coding, in: 2018 International Conference on Advanced Technologies for Communications (ATC), 2018, pp. 120–124.

[22] P. ITU-T RECOMMENDATION, Subjective video quality assessment methods for multimedia applications.

[23] T. Brandao, L. Roque, M. P. Queluz, Quality assessment of H.264/AVC encoded video, in: Proc. of Conference on Telecommunications - ConfTele, Sta. Maria da Feira, Portugal, 2009, p. 5.

[24] Xiph.org :: Derf's Test Media Collection.
URL https://media.xiph.org/video/derf/

[25] B. Tan, J. Wu, H. Cui, R. Wang, J. Wu, D. Liu, A Hybrid Digital Analog Scheme for MIMO Multimedia Broadcasting, IEEE Wireless Communications Letters 6 (3) (2017) 322–325.

[26] D. He, C. Luo, C. Lan, F. Wu, W. Zeng, Structure-preserving hybrid digital-analog video delivery in wireless networks, IEEE Transactions on Multimedia 17 (9) (2015) 1658–1670.

[27] D. He, C. Lan, C. Luo, E. Chen, F. Wu, W. Zeng, Progressive Pseudo-analog Transmission for Mobile Video Streaming, IEEE Transactions on Multimedia 19 (8) (2017) 1894–1907.

[28] Q. Huynh-Thu, M. Ghanbari, Scope of validity of PSNR in image/video quality assessment, Electronics Letters 44 (13) (2008) 800–801. doi:10.1049/el:20080522.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

[30] M. Frigo, S. Johnson, The Design and Implementation of FFTW3, Proceedings of the IEEE 93 (2) (2005) 216–231.

[31] D. Salomon, G. Motta, Handbook of Data Compression, 5th Edition, Springer-Verlag, London, 2010.

[32] M. Wien, High Efficiency Video Coding: Coding Tools and Specification, Signals and Communication Technology, Springer-Verlag, Berlin Heidelberg, 2015.

[33] T. Fryza, S. Hanus, Video signals transparency in consequence of 3d-dct transform, in: Radioelektronika 2003 Conference Proceedings, 2003, pp. 127–130.

[34] T. Fryza, Improving Quality of Video Signals Encoded by 3d DCT Transform, in: Proceedings ELMAR 2006, 2006, pp. 89–93.

[35] T. Ouni, W. Ayedi, M. Abid, New Non Predictive Wavelet Based Video Coder: Performances Analysis, in: Image Analysis and Recognition, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 344–353.

[36] F. Urban, R. Poullaouec, J. Nezan, O. Deforges, A Flexible Heterogeneous Hardware/Software Solution for Real-Time HD H.264 Motion Estimation, IEEE Transactions on Circuits and Systems for Video Technology 18 (12) (2008) 1781–1785.

[37] C. Yim, A. C. Bovik, Evaluation of temporal variation of video quality in packet loss networks, Signal Processing: Image Communication 26 (1) (2011) 24–38. doi:10.1016/j.image.2010.11.002.

[38] M. Paul, Weisi Lin, Chiew-Tong Lau, Bu-Sung Lee, Explore and Model Better I-Frames for Video Coding, IEEE Transactions on Circuits and Systems for Video Technology 21 (9) (2011) 1242–1254.

[39] Spatiotemporal index, matlab code.
URL https://sites.google.com/site/matteonaccari/software