A comparative preprocessing study for softcast video transmission

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

A Comparative Preprocessing Study for SoftCast Video Transmission

Anthony Trioux, François-Xavier Coudoux *Senior Member, IEEE,* Patrick Corlay, Mohamed Gharbi
UMR 8520 - IEMN, DOAE
Univ. Polytechnique Hauts-de-France, CNRS, Univ. Lille, YNCREA, Centrale Lille
F-59313 Valenciennes, France
{anthony.trioux, francois-xavier.coudoux, patrick.corlay, mohamed.gharbi}@uphf.fr

***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 such as cliff effect. In this paper, we first review the SoftCast scheme. We then analyze and compare two simple preprocessing methods that help to increase the quality at the receiver side. Each method consists of the subtraction of either the 8-bit mean gray level 128 or the mean value of all pixels for each video frame. Preprocessing methods are compared and evaluated under different channel signal-to-noise ratios using two metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM). Simulation results clearly highlight the importance of the preprocessing block in a SoftCast wireless video transmission scheme. Depending on the video input characteristics, an improvement of the PSNR score up to 2.1dB and 2.5dB can be observed for the average gray level method and for the mean frame method, respectively.**

***Index Terms*—Wireless Video Transmission, SoftCast, Uncoded Transmission, Preprocessing Method**

I. Introduction

Broadcasting video content constitutes a challenge because each user is subject to unreliable and different wireless channel that varies over time. Traditional video coding schemes such as H.264/AVC [1] and HEVC [2] are not suitable for transmission in such environments and applications. Indeed, the parameters are adjusted to match a bitrate available that is given under predicted or assumed channel state. However, a mismatch between the actual channel state and the predicted or assumed one leads to a cliff effect [3] or levelling-off effect [4] in received video quality. The first one refers to the fact that video quality drops quickly when the Channel Signal-to-Noise Ratio (CSNR) is below a presumed value whereas the latter one refers to the fact that video quality stays almost constant even if the CSNR increases. In the last few years, the so-called *SoftCast* [5] has been proposed to deal with these issues.

Different from traditional schemes, *SoftCast* is a Joint Source–Channel Coding scheme and represents the pioneer work of the uncoded video transmission [6] also called the soft video delivery [7]. In uncoded video transmission the video pixels are processed by successive linear operations and are directly transmitted without neither quantization nor coding process such as entropy and channel coding.

A few major properties which makes *SoftCast* a good candidate for broadcasting video content over wireless channels are listed below:

- Uncoded video transmission can achieve graceful degradation [8];
- For each user, the video quality at the receiver side is a linear function of the CSNR;
- A single data stream is delivered and can be decoded by any receivers even if they are subject to bad channel conditions;
- *SoftCast* scheme works without the need of any feedback from receivers [5].

Since the original works [5], uncoded video transmission has gathered a significant interest from the research community [4], [6]–[10]. The authors in [9]–[11] propose improvements based on the characteristics of the Human Vision System (HVS) whereas [7], [8] propose efficient signal energy modeling to better allocate bandwidth resources and therefore improve the received video quality.

In this paper, we first review the original *SoftCast* scheme and then compare two simple preprocessing methods that has been proposed in the literature [12], [13]. By preprocessing, we mean the energy reduction before applying all the transformations in the *SoftCast* scheme. These methods help to increase the received video quality with none to reduce bandwidth consumption. Results underline the importance of the preprocessing block in an uncoded scheme such as *SoftCast*. The comparison between the two methods shows that depending on the application and the corresponding available bandwidth one may be preferred to the other.

The rest of this paper is organized as follows: Section II gives a review of the *SoftCast* scheme. Section III introduces the preprocessing methods evaluated in this paper. In section IV, we compare the methods against classical *SoftCast* scheme based on simulation results. Conclusions and discussions are presented in section V.

II. SoftCast Overview

In this section, the basic scheme of *SoftCast* [5] is given in Fig. 1. We note that the green blocks are not in the original *SoftCast* scheme but have been added as additional steps and are described in Section III .

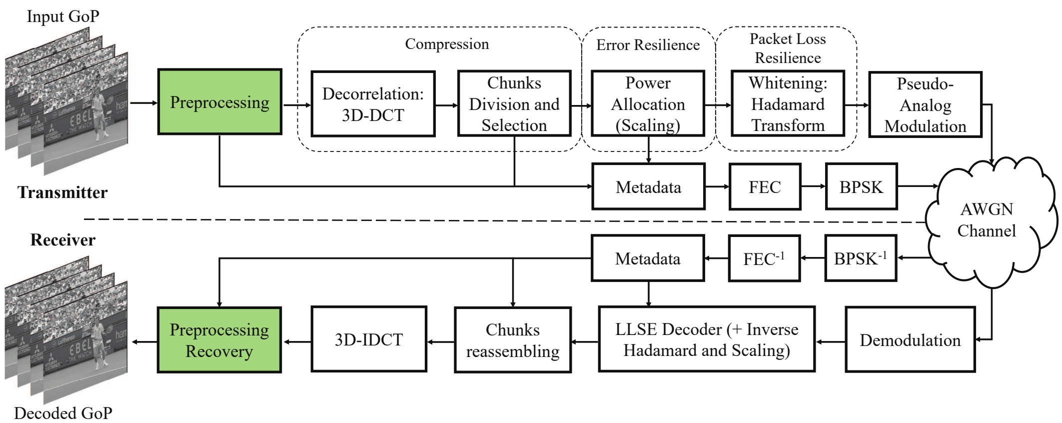


Fig. 1: Block diagram of the *SoftCast* video transmission scheme.

A. *Compression Step*

SoftCast first operates on Group of Pictures (GoP) and decorrelates the signal through a three-dimension full-frame Discrete Cosine Transform (DCT). The 3D-DCT is simply done by exploiting the separability of the DCT transform, i.e., the scheme first transforms each frame with a spatial 2D-DCT and then performs a temporal 1D-DCT over the GoP as shown in Fig. 2.

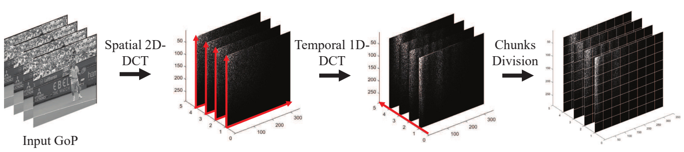


Fig. 2: Compression Step in *SoftCast* scheme. From left to right: GoP in pixel domain, 2D-transformed frames, 3D-transformed frames, Chunks division after 3D-DCT.

The 3D-DCT [14] of a GoP $f(i,j,k)$ is denoted by $\mathbf{F}(u,v,w)$ and is given by:

$$\mathbf{F}(u,v,w) = \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1}\sum_{k=0}^{R-1} f(i,j,k)\cdot C_{i,u}\cdot C_{j,v}\cdot C_{k,w} \quad (1)$$

where P, Q denote the frame size and R the GoP-size.

$$C_{p,q} = \begin{cases} \frac{1}{\sqrt{Z}}, q = 0 \\ \frac{2}{\sqrt{Z}}\cdot\cos(\frac{(2p+1)q\pi}{2Z}), otherwise \end{cases} \quad (2)$$

with Z equals to P, Q or R, depending on the selected $C_{p,q}$.

After 3D-DCT, the frames are divided into small blocks called chunks and rearranged to form a new matrix where each row contains a chunk. These chunks are ordered by energy descending order.

In *SoftCast* scheme, the compression can be done after decorrelation transform by discarding a certain amount of chunks. This amount is fixed by the available bandwidth and the modulation used at the transmitter side. The procedure to determine the number of chunks that can be transmitted is further introduced in subsection II-D.

In uncoded video transmission, the compression ratio [15] is defined by:

$$CR = K/N \quad (3)$$

where, K is the number of transmitted chunks and N the total number of chunks within a GoP. This ratio ranges between 0 (no data sent) and 1 (no compression).

The number of chunks per GoP N is defined by:

$$N = \frac{nb_R \cdot nb_C \cdot nb_F}{nb_r \cdot nb_c}, \text{[chunks/GoP]} \quad (4)$$

where nb_R, nb_C represents the frame size, nb_r, nb_c the chunk size and nb_F refers to the number of frames within a GoP.

B. *Error Resilience Step*

The next block called Power Allocation or Scaling is used to provide error resilience to the scheme. *SoftCast* scales the magnitude of the DCT coefficients to offer a better protection against noise that affects the components during transmission. Since the available power transmission P is limited and fixed, it must be distributed to all the chunks in a way that limits the Mean Square Error (MSE). This division is a typical Lagrangian problem and the solution is given by:

$$g_i = \lambda_i^{-1/4}\cdot\sqrt{\frac{P}{\sum_j\sqrt{\lambda_j}}} \quad (5)$$

where $g_i, i = 1, 2, \ldots, K$ is the scaling factor for the i^{th} chunk, and $\lambda_i = E[\mathbf{X_i}^2]$ is the energy of the i^{th} chunk [6].

The resulting scaled-chunks are defined as $\mathbf{U}_i[j] = g_i\mathbf{X}_i[j]$ where $j = 1, 2, \ldots, nb_r \cdot nb_c$ represents the j^{th} DCT coefficient in the chunk i.

We note that only one scaling factor per chunk is computed. This is the result of a trade-off between quality received, amount of metadata and computation cost. Readers may refer to [5] for further details.

C. *Packet Loss Resilience Step*

The scaled coefficients are then multiplied by a Hadamard matrix to provide packet loss resilience. The Hadamard matrix is an orthogonal transform composed of $+1$ and -1 elements.

After 3D-DCT transformation each chunk presents huge discrepancy in terms of energy. The goal of this transformation is to ensure that each packet contains approximately the same amount of information making them equally important for the reconstruction process. This transformation takes the chunks as input and outputs slices. Each slice is simply a linear combination of all scaled-chunks and is defined by $\mathbf{Y}_i[j] = H_i \cdot \mathbf{U}_i[j]$ where H_i denotes the i^{th} row of the Hadamard matrix.

D. Modulation Step

After all above listed operations process, the obtained coefficients are directly mapped in pairs (I and Q in OFDM) and transmitted without any coding step in a pseudo-analog manner referred as Raw Orthogonal Frequency Division Multiplexing (Raw-OFDM) [6].

In the Raw-OFDM, the Forward Error Correction (FEC) code is bypassed and Pseudo-Analog Modulation replaces the classical modulation part of OFDM. Therefore, instead of bitrate only symbol rate is considered hereafter. Since coefficients are sent in pairs (I and Q planes), the maximal resulting matching channel bandwidth in *SoftCast* video transmission can be described as follows:

$$BW_{max} = \frac{nb_R \cdot nb_C \cdot \vartheta}{2}, \quad \text{[symbs/s]} \tag{6}$$

where ϑ is the frame rate of the video expressed in frame per second (fps).

For instance, a CIF video format with 30fps represents a data volume of $352 \cdot 288 \cdot 30 = 3.04 \cdot 10^6$ real values per second to be transmitted [9]. The resulting matching channel symbol rate is $3.04 \cdot 10^6/2$ = 1.52Msymbols/s. If we consider a wireless bandwidth available of 1MHz per user, nearly 30% of compression is needed. Furthermore, in the case of transmission High Definition (HD) contents or new format such as 4K the wireless bandwidth needed is too large and discarding chunks in the compression step becomes unavoidable in *SoftCast* scheme. A 4K video format example with 60fps results in $601.3 \cdot 10^6$ real values per second to be transmitted corresponding to a wireless bandwidth of 300.6 MHz which is unrealistic.

The Compression Ratio (CR) in (3) can also be expressed as a bandwidth ratio as follows:

$$CR = BW_{ava}/BW_{max} \tag{7}$$

where BW_{ava} denotes the available bandwidth at the transmitter.

If the available bandwidth BW_{ava} is less than the maximal resulting bandwidth channel, the number of transmitted chunk K within a GoP is adjusted accordingly.

By using (3), (4) and (7) we get the maximum number of chunks K per GoP that can be transmitted:

$$K = \lfloor \frac{BW_{ava} \cdot N}{BW_{max}} \rfloor \tag{8}$$

$$= \lfloor \frac{2.BW_{ava} \cdot nb_F}{nb_r \cdot nb_c \cdot \vartheta} \rfloor \tag{9}$$

E. Metadata

The metadata represent the key elements in *SoftCast* scheme that are essential to recover video data signal. They represent a small amount of three datasets:

- The mean of each chunk, noted μ_i;
- The variance/energy of each chunk, noted λ_i;
- A bitmap which indicates the positions of the discarded chunks into the GoP.

To ensure a high probability of correct delivery and therefore, a correct decoding process, they are strongly protected and transmitted in a robust way (BPSK for example [7]). We note that the bandwidth required for transmitting metadata has not been considered above but typical values are given in [7].

F. LLSE Decoder

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.

In *SoftCast*, an Additive White Gaussian Noise (AWGN) with zero-mean, variance of σ^2 and noted $n_i[j]$ is assumed.

After all linear operations in the encoder, $\mathbf{Y}_i[j]$ is sent and the corresponding $\hat{\mathbf{Y}}_i[j] = \mathbf{Y}_i[j] + n_i[j]$ is received. *SoftCast* decodes and gets the best estimate [5] as:

$$\hat{\mathbf{X}}_i[j] = \frac{g_i \lambda_i}{g_i^2 \lambda_i + \sigma^2} \cdot \hat{\mathbf{Y}}_i[j] \tag{10}$$

The transmitter side can be synthesized into matrix form as follows [5]:

$$Y = HGX = CX \tag{11}$$

where X is the chunk's matrix and Y the corresponding slices transmitted over the channel. The encoding matrix C is the product of HG where H and G are the Hadamard matrix and the scaling factors matrix, respectively.

Therefore, the LLSE equation can be rewritten in matrix form as follows:

$$\hat{\mathbf{X}}_{LLSE} = \lambda_x C^T (C \lambda_x C^T + \Sigma)^{-1} \hat{Y} \tag{12}$$

where λ_x is a diagonal matrix whose diagonal elements are the energy λ_i of the i^{th} chunk, and Σ is a diagonal matrix in which the i^{th} element is the channel noise power σ_i^2 experienced by the packet carrying the i^{th} row of Y.

Assuming without loss of generality that a packet contains a single slice, the LLSE decoder can be rewritten in the presence of packet loss as [5]:

$$\hat{\mathbf{X}}_{LLSE} = \lambda_x C_{*i}^T (C_{*i} \lambda_x C_{*i}^T + \Sigma_{(*i*i)})^{-1} \hat{Y_{*i}} \tag{13}$$

where C_{*i} and Y_{*i} denote C and Y after removing the i^{th} lost row.

III. Preprocessing Methods

In this section, we study how the preprocessing methods influence the reconstructed video quality after transmission. Recently, Xiong *et al.* [6] introduced the concept of *data activity* for uncoded transmission. They showed that this term denoted by $H = \frac{1}{N}\sum_{i=1}^{N}\sqrt{E[x_i^2]}$ affect the reconstructed PSNR at receiver side as follows:

$$PSNR_{dB} = c + CSNR_{dB} - 20\log_{10}\left(H\right) \tag{14}$$

with $c = 20\log_{10}(255)$.

This formula underlines the importance of having a reduced data activity and hence emphasizes the benefit of reducing the energy before applying all the transformations in the *SoftCast* scheme.

It is well-known that the DC component after DCT transformation carries most of the energy. A simple solution would be to remove the DC component of each frame after 2D-DCT and before applying the 1D-temporal DCT. However, preprocessing method can only work if the removed information is perfectly recovered at decoder level. This is why these data should be transmitted in metadata, recalling that they are strongly protected (FEC) and transmitted in a robust way (BPSK) to ensure correct delivery. Nevertheless, it means that a larger bandwidth must be allocated to the metadata. When the available bandwidth is greater than the needed one, transmitting additional information in metadata is a feasible solution but it is not when the targeted application has bandwidth constraints.

We now show that transmitting DC component directly in metadata is not relevant. To see how, we recall (1) and (2) and focus on the 2D-DCT case. We calculate the DC component $\mathbf{F}(0,0)$ as follows:

$$\begin{aligned}\mathbf{F}(0,0) &= \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} f(i,j)\cdot C_{i,0}\cdot C_{j,0} \\ &= C_{i,0}\cdot C_{j,0}\sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} f(i,j) \\ &= \frac{1}{\sqrt{P}}\cdot\frac{1}{\sqrt{Q}}\cdot P\cdot Q\cdot\bar{f}\end{aligned} \tag{15}$$

where $C_{i,0} = \frac{1}{\sqrt{P}}$ and $C_{j,0} = \frac{1}{\sqrt{Q}}$, P, Q denote the size of the frames and $\bar{f}$ denotes the mean-pixel value of the frame.

A simple calculation for a CIF video sequence (352×288 pixels) shows that each 2D-DC component in *SoftCast* context should at least be coded on 18 bits (8 bits for the mean-pixel value of the frame, 1 sign bit and 9 bits to code the mathematical operation $\sqrt{352}\cdot\sqrt{288}$) whereas only 8 bits are needed for each frame when directly subtracting the mean value in pixel domain and before applying the 2D-DCT. Therefore, we do not consider the preprocessing applied in transformed domain but focus hereafter on pixel domain preprocessing.

The two preprocessing methods are defined as follows:

$$f_{proc}(i,j) = f(i,j) - \bar{p} \tag{16}$$

where $f_{proc}(i,j)$ denotes the pixel frame after preprocessing. The value of $\bar{p}$ is given differently in the two methods: $\bar{p} = \lfloor|mean(x)|\rceil$ in the first case and $\bar{p} = 128$ in the second. We note that $\lfloor\bullet\rceil$ denotes the rounding operation to the nearest integer.

The first solution has been recently used by Hagag *et al.* [13] whereas the alternative solution has been proposed by Cui *et al.* [12]. This alternative solution which consists of subtracting an average 8-bit mean gray level (i.e., 128) avoids to increase the allocated bandwidth for metadata transmission. Indeed, adding an offset of $+128$ for each frame after decoding process does not require any additional information to be transmitted. In the next section, we first compare both methods to the original *SoftCast* scheme in terms of reconstructed video quality and then evaluate the gap between the two methods.

IV. Performance Analysis of the Preprocessing Methods

A. Simulation Setup

Video sources: We evaluate and compare the preprocessing methods through extensive simulations. The luminance part of video CIF sequences (with a frame rate of 30 fps) from the Xiph collection [16] are used as the inputs. The process is performed GoP by GoP with a GoP-size of 16 frames and each frame is split into 64 chunks of 44×36 pixels as in [7], [8]. For each sequence, we send the first 288 frames.

In the following tests, we choose the *Australia*, *News* and *Stefan* sequences because of their different spatiotemporal characteristics. *Stefan* contains high temporal and spatial activities. In contrast, *Australia* and *News* presents slow motions and low to medium spatial contents, respectively. Results with other CIF sequences are similar.

Wireless characteristics: Transmissions through AWGN channels in the range of $[5{\sim}25dB]$ are considered. The transmission power is normalized to $P_{total} = 1$. To ensure a fair comparison, the same noise is generated and applied to all methods. Results are averaged over 10 realizations.

Evaluation Metrics: Two commonly used metrics are here considered: the PSNR based on Mean Square Error (MSE) and the SSIM. The PSNR is used as a purely objective metric whereas the SSIM provides a quality index more correlated with the Human Visual System (HVS) [9]. The SSIM calculation outputs values between 0 (worst quality) and 1 (best quality). The PSNR is given by $PSNR_{dB} = 10\log_{10}\left(\frac{(2^L-1)^2}{MSE}\right)$ and the MSE is defined as $MSE = \frac{1}{PQ}\sum_{x=1}^{P}\sum_{y=1}^{Q}\left[(I_{ori}(x,y) - I_{rec}(x,y))^2\right]$ where I_{ori} denotes the original frame and I_{rec} the reconstructed one.

B. Simulation Results

Fig. 3 shows the average video quality results in terms of PSNR and SSIM for the original *SoftCast* scheme where no preprocessing is applied and the two others where subtraction in pixel domain is done before encoding process.

We can firstly note that the PSNR follows a linear relationship. As mentioned in Section I, it is a key feature from *SoftCast* that has been studied in [6].

Then, regardless of the input video sequence, the subtraction of the pixel-mean image before encoding process gives the best results. The biggest difference in quality reconstruction between the classical *SoftCast* scheme and the preprocessing methods appears for the *Australia* sequence. This is because Australia contains low spatiotemporal activities making it easy to decorrelate. Most of the energy after 3D-DCT is concentrated in the low frequency bands and protecting almost all the information contained in DC components leads to an improvement bigger than 2dB. In contrast, *News* and *Stefan* contains higher spatiotemporal contents, i.e., the energy is spread across more components and therefore the gap becomes lower.

We can also note that the gap between the two methods is higher for the *News* sequence and almost null for the *Stefan* sequence. This is because the pixel-mean value of each frame is in average equal to 132 (close to 128) and 78 for the *News* and *Stefan* respectively. It is normal to observe better results with the preprocessing block, since the subtraction of the pixel-mean value or an average gray level helps to reduce the DC component of each 2D-DCT frames as seen in Section III. The DC component reduction leads to a reduced data activity [6] which results in a better reconstructed video quality.

Depending on the video input characteristics, the improvement in PSNR and SSIM between the two methods can be visible or almost null (i.e., when the pixel-mean value of each frame is around 128). As an illustrative example we give numerical values for the *News* and *Australia* sequence in Table I. P1 and P2 denote the preprocessing methods, i.e., the subtraction of the pixel-mean value and the subtraction of 128 respectively.

TABLE I: Evaluation of the maximal quality improvement for the *News* and *Australia* sequences

Maximum Quality Improvement			
Video Sequence	SoftCast vs P2	SoftCast vs P1	P1 vs P2
News	PSNR = 0.73dB SSIM = 0.012	PSNR = 1.4dB SSIM = 0.023	PSNR = 0.67dB SSIM = 0.011
Australia	PSNR = 2.1dB SSIM = 0.043	PSNR = 2.5dB SSIM = 0.052	PSNR = 0.41dB SSIM = 0.009

Regarding the SSIM curves, we observe that the gap between the three evaluated schemes decreases when the CSNR becomes higher. This is due to the fact that at high CSNR ($>$25dB) the perturbation of the AWGN noise becomes negligible and therefore the reconstruction of the DC component in classic *SoftCast* scheme approaches the real DC value before transmission.

Finally, a visual comparison is given in Fig. 4 where the reconstructed frames and the error images are displayed (the error images have been shifted by $+128$ for viewing purposes). We deliberately set the CSNR to 0dB in order to accentuate the noise during transmission. We can clearly observe that the classical *SoftCast* gives the lower video quality received. In contrast, the studied preprocessing methods achieve better reconstructed quality under the same channel characteristics.

V. Conclusion

In this paper, we make a comparative study of two simple preprocessing methods that can be used in a *SoftCast* context. Results show that depending on the video and its frame-mean value, the pixel-mean subtraction either gets better results (up to 0.67dB for the tested sequences) than average 8-bit gray level method or performs similar (i.e., when the image-mean is close to 128). Regardless of that aspect, the reconstructed video quality at the receiver side is always better than the classical *SoftCast* scheme. The choice between the two preprocessing method is made according to the available bandwidth. Indeed, the latter one (8-bit gray level method) does not need to allocate more bandwidth for the metadata making it a possible solution in bandwidth-constrained environments.

Acknowledgment

We would like to thank Prof. Cagnazzo and Prof. Kieffer for their help during the implementation of the *SoftCast* scheme.

References

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

[2] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] S. Kokalj-Filipovi and E. Soljanin, “Suppressing the cliff effect in video reproduction quality,” *Bell Labs Technical Journal*, vol. 16, no. 4, pp. 171–185, Mar. 2012.

[4] F. Liang, C. Luo, R. Xiong, W. Zeng, and F. Wu, “Superimposed Modulation for Soft Video Delivery with Hidden Resources,” *IEEE Trans. Circuits Systems Video Technol.*, vol. 28, no. 9, pp. 2345–2358, Sep. 2018.

[5] S. Jakubczak and D. Katabi, “SoftCast: Clean-slate scalable wireless video,” *MIT Technical report*, Feb. 2011.

[6] R. Xiong, F. Wu, J. Xu, X. Fan, and al., “Analysis of decorrelation transform gain for uncoded wireless image and video communication,” *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1820–1833, Apr. 2016.

[7] T. Fujihashi, T. Koike-Akino, T. Watanabe, and P. V. Orlik, “High-Quality Soft Video Delivery With GMRF-Based Overhead Reduction,” *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 473–483, Feb. 2018.

[8] R. Xiong, J. Zhang, F. Wu, J. Xu, and W. Gao, “Power Distortion Optimization for Uncoded Linear Transformed Transmission of Images and Videos,” *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 222–236, Jan. 2017.

[9] D. He, C. Luo, C. Lan, F. Wu, and W. Zeng, “Structure-preserving hybrid digital-analog video delivery in wireless networks,” *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1658–1670, Sep. 2015.

[10] J. Shen, L. Yu, L. Li, and H. Li, “Foveation Based Wireless Soft Image Delivery,” *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2788–2800, Oct. 2018.

[11] J. Zhao, R. Xiong, C. Luo, F. Wu, and W. Gao, “Wireless image and video soft transmission via perception-inspired power distortion optimization,” in *IEEE Visual Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

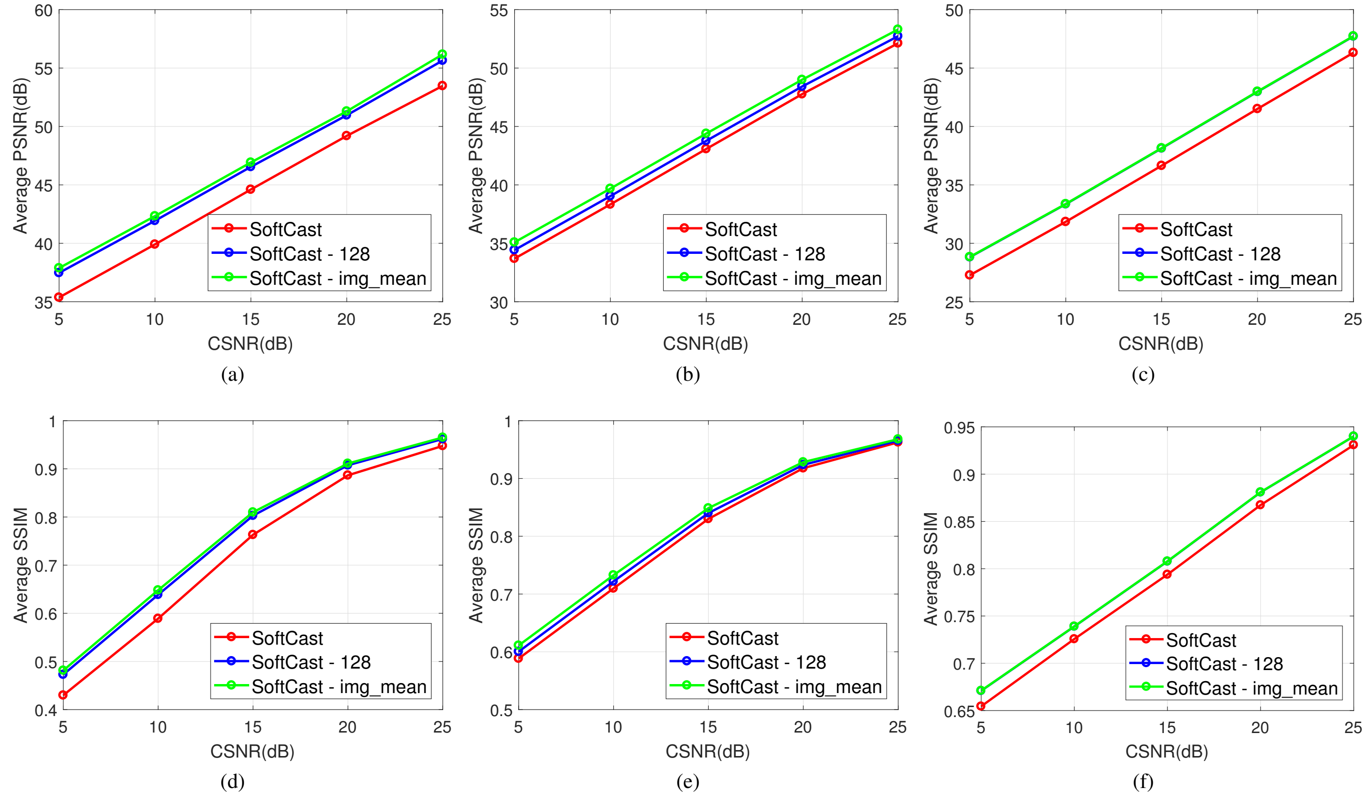
[12] H. Cui, D. Liu, Y. Han, and J. Wu, “Robust Uncoded Video Transmission under Practical Channel Estimation,” in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6.

[13] A. Hagag, X. Fan, and F. E. A. El-Samie, “Satellite Images Broadcast Based on Wireless SoftCast Scheme.” *Int. Journal Computer Science*, vol. 44, no. 1, Feb. 2017.

[14] J. S. Park and T. Ogunfunmi, “A 3D-DCT video encoder using advanced coding techniques for low power mobile device,” *Journal of Visual Commun. Image Representation*, vol. 48, pp. 122–135, Oct. 2017.

[15] Z. Li, H. Lu, and Y. Wu, “Compressed uncoded screen content video transmission in bandwidth-constrained wireless networks,” in *IEEE Int. Conf. Wireless Commun. & Signal Process. (WCSP)*, Nov. 2016, pp. 1–5.

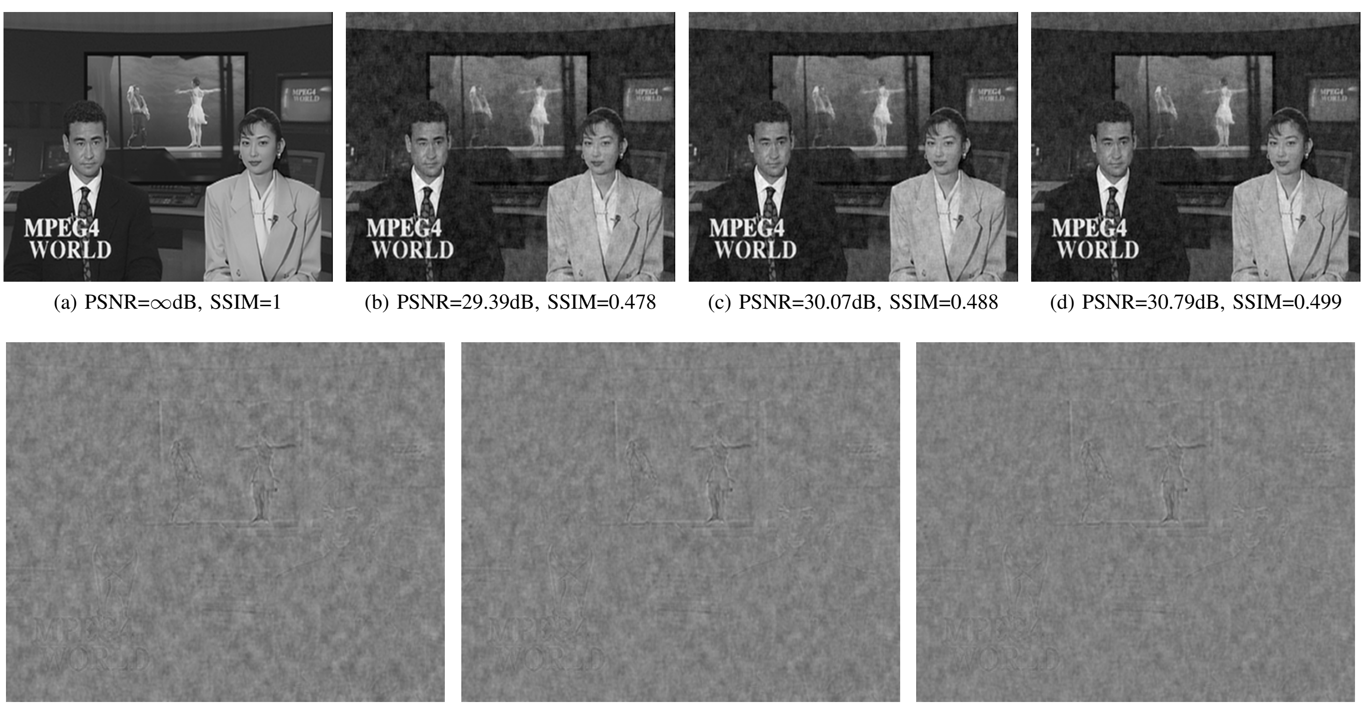
[16] “Xiph.org media.” [Online]. Available: https://media.xiph.org/video/derf/



(a) (b) (c)

(d) (e) (f)

Fig. 3: Average simulation results for classic SoftCast scheme and two added preprocessing methods: Subtraction of the 8-bit mean gray level 128 and subtraction of the mean value of all pixel for each video frame. From left to right: (a), (d) *Australia* sequence, (b), (e) *News* sequence and (c), (f) *Stefan* sequence. First row: Average PSNR results. Second row: Average SSIM results. Please enlarge the figure to observe details.



(a) PSNR=∞dB, SSIM=1 (b) PSNR=29.39dB, SSIM=0.478 (c) PSNR=30.07dB, SSIM=0.488 (d) PSNR=30.79dB, SSIM=0.499

(e) PSNR=29.39dB, SSIM=0.478 (f) PSNR=30.07dB, SSIM=0.488 (g) PSNR=30.79dB, SSIM=0.499

Fig. 4: Visual quality comparison at a CSNR equal to 0dB for *News* sequence (first frame). First row: Reconstructed frames in pixel domain. Second row: Resulting error images. From left to right: (a) Original frame, (b) and (e) Classic SoftCast, (c) and (f) SoftCast with subtraction of the 8-bit mean gray level 128, (d) and (g) SoftCast with subtraction of the mean value of all pixels for each video frame. Please enlarge the figure to observe details.