AN OBJECT-ORIENTED COGNITIVE SOURCE CODING ARCHITECTURE FOR 3D VIDEO COMMUNICATIONS

Simone Milani and Giancarlo Calvagno

Dept. of Information Engineering, University of Padova, Italy
e-mail: {simone.milani,calvagno}@dei.unipd.it

ABSTRACT

Reliable transmission of 3D video signals is nowadays an interesting research issue both for the new coding challenges that three-dimensional video signals pose and for the wide diffusion of multimedia communications over wireless networks. In order to deal effectively with packet losses over radio channels, several robust source coding schemes have been proposed. This article presents a reconfigurable and flexible architecture (named Cognitive Source Coder in analogy with Cognitive Radio systems) that implements different robust source coding solutions and adaptively adopts them according to channel conditions. The proposed approach permits improving the quality of the 3D scene reconstructed at the end terminal with respect to the corresponding non-adaptive approaches.

1. INTRODUCTION

Wireless networks are nowadays intended to provide a wide variety of multimedia applications, which are characterized by intense bandwidth requirements and high sensitivity to packet losses. The recent advent of 3D video transmission has utterly exacerbated these peculiarities since three-dimensional video streams are characterized by great amounts of data and poor resilience to delays and errors. As a matter of fact, novel protocols and transmission strategy have been designed in order to grant a certain Quality-of-Experience (QoE) to the end user by minimizing the loss of crucial data and limiting delays and jitters. Among these, cross-layer (CL) solutions are able to maximize the quality of the received multimedia content by jointly tuning the transmission parameters of the different layers in the protocol stack. In this way, it is possible to combine different protection and retransmission strategies to satisfy the requirements related to the specific application. Within the existing cross-layer solutions, a subset of the proposed approaches adapt the chosen source coder to the characteristics of the transmitted video sequence and to the network state. In this paper we will refer to these solutions with the term Cognitive Source Coding (CSC) scheme in analogy with Cognitive Radio schemes. Cognitive Radio (CR) architectures are wireless systems that can sense the transmission environment, identify which spectrum frequencies are available, and change the modulation scheme in order to be able to transmit over the available channels [1]. It is possible to notice that CSC schemes presents many features in common with CR solutions. CSC architectures implement many source coding strategy and adaptively switch from one to another depending on the channel state. In a similar way, CR schemes implement many modulation schemes and can adaptively switch from one to another depending on which portion of the radio spectrum they want to use. Moreover, CSC schemes, as well as CR solutions, must sense the transmission environment in order to understand how many transmission channels are available and what their states are. Both CSC and CR must present a high degree of flexibility and reconfigurability in order to change configuration without requiring exceeding computational complexity and hardware resources on the transmitting and receiving terminals.

The paper present a CSC video coding solution that reuses the building blocks of the H.264/AVC FRExt coder and presents a limited computational complexity. The proposed approach includes a single description standard H.264/AVC coder (SDC), a Multiple Description video coder based on a polyphase sub-sampling, and a Wyner-Ziv coder that is used to characterize the residual signal after prediction in the standard H.264/AVC and in the MDC coders. These elements can be obtained by a simple rewiring of the signals in the H.264/AVC coder. An additional FEC coder is also applied on the video RTP packets in order to protect the video packet stream against losses. The designed CSC coder is applied to a video+depth 3D signals[1] sequence optimizing both the quality of the reconstructed view and the accuracy of the received depth map. More precisely, the optimization strategy distinguishes the static elements in the sequence from the dynamic ones and chooses the most appropriate coding configuration according to the characteristics of the coded video sequence and to the channel state. In the following, Section 2 presents the adopted CSC scheme in detail by specifying its basic building blocks. Section 3 presents how the source coding strategy is optimized according to the characteristics of the video and depth signals. Experimental results in Section 4 show how it is possible to improve the 3D experience provided to the end user. Conclusions are drawn in Section 5.

2. THE PROPOSED COGNITIVE SOURCE CODING SCHEME

The adopted scheme (see Fig. 1) implements a set of different video coding strategies that can be adaptively employed to maximize the quality of the sequence reconstructed at the decoder. More precisely, the designed architectures combines a simple Multiple Description Coding with a traditional H.264/AVC scheme and a Wyner-Ziv (WZ) video coding. As a matter of fact, we adopted MDC and WZ schemes whose building blocks present many similarities with those of the H.264/AVC encoder. In this way, it is possible to reuse many functional units since each different source coding solution can be implemented by rewiring the connections between different blocks. From these premises, we adopted the MDC scheme in [3] and the WZ coding solution in [4] since they

[1]This 3D video format is also called Depth Image Based Rendering (DIBR) [2].

 ISSN 2076-1465

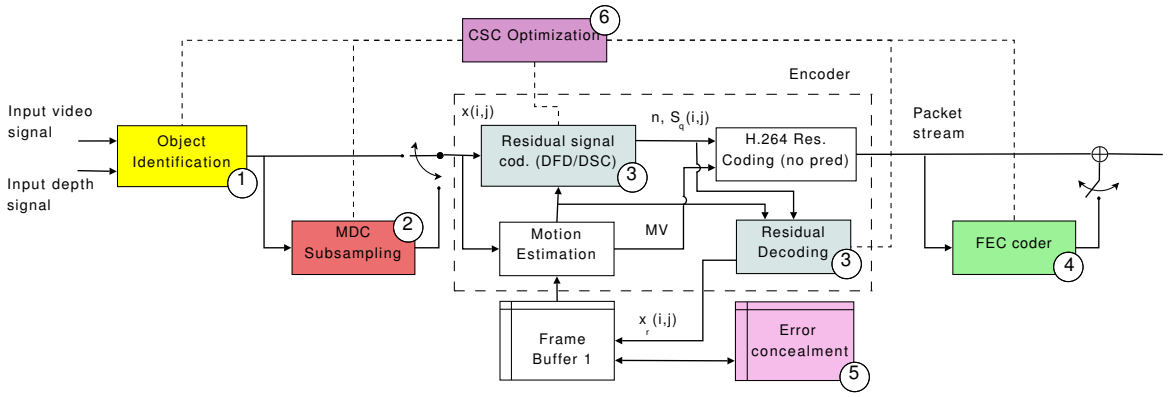


Figure 1: Block diagram for the encoder.

inherit most of the building blocks of H.264/AVC with the addition of some low-complexity functional blocks. The resulting CSC coder requires the same computational complexity of the H.264/AVC coder. The output packet stream can be protected by adding extra FEC packets generated by an erasure code as defined in the RFC2733 [5]. In the following, we will describe the different units of the CSC encoder and their possible configurations.

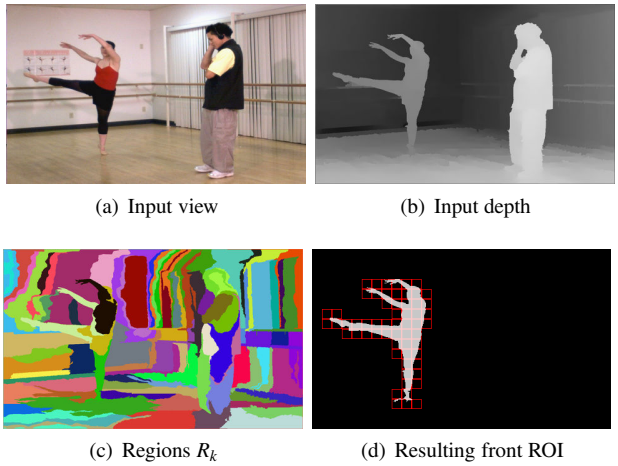
2.1 Object identification unit

The input video and depth signals are processed by an object identification unit (see the block ① in Fig. 1), which analyzes the captured scene and the different objects within the frame distinguishing fast-moving objects in the foreground from slowly-moving objects and background elements. According to the spatial-temporal characteristics of the signals associated to the different objects it is possible to vary the configuration of the coder in order to maximize the 3D visual quality experienced by the end user.

In our approach, we distinguished two separate classes of objects in the captured scene. The first class includes fast-moving elements close to the point of view, which have a stronger impact on the final visual quality and proves to be affected more significantly by packet losses. The second class comprises elements in the background and slowly moving objects in the foreground: the first have a minor impact on the final visual quality, while the latter can be easily estimated in case of packet losses with a limited channel distortion. This classification can be performed by segmenting [6] the input frames into multiple small portions according to the information provided by the depth signal. The different segments are then fused together according to the following procedure.

Let R_k, with $k = 0,\ldots,N_R-1$, be N_R regions of pixels obtained segmenting the current depth map frame via the algorithm in [6]. The object detection algorithm computes the average depth value $\overline{d}$ within each region R_k and the MSE between the pixels in R_k of the current video frame and the corresponding pixels of the previous video frame. In case the MSE is lower than the threshold T_1, the region R_k is assigned to the class of static objects and background. As for the remaining regions, the object detection routine clusters the set of regions according to the associated value $\overline{d}$ and fuses them together into a new set of regions R'_k. In the end, the average depth values $\overline{d}'$ for the regions R'_k are then quantized into two classes, and the associated regions R'_k are merged together. According to the average MSE values for the pixels in the last two regions, pixels are associated to the class of moving objects or to the class of static elements.

The results of the object identification unit are two pixel masks R_F (containing the moving objects in the foreground) and R_B (containing background and static objects). Pixel masks can be used to distinguish the image regions in an object-based video coder and characterize them separately. In the current version of the codec, we leave this possibility for the future versions of the CSC architecture, and, in order to limit the additional computational complexity related to the processing of masks both at the encoder and at the decoder, we employ the obtained pixel masks to partition the input frames into two Regions-Of-Interest (ROI). The first one is made of macroblocks M_F related to R_F (front ROI), while the second one is made of macroblocks M_B related to R_B (back ROI). As a result, four frame sequences are generated: the sequences of front macroblocks for both the video and the depth signals (named V_F and D_F respectively), and the sequences of background macroblocks (named V_B and D_B respectively). The four subsequences V_F, V_B, D_F, and D_B are then coded by different source coding strategy according to the characteristics of the processed signal and of the transmission channel.



(a) Input view (b) Input depth

(c) Regions R_k (d) Resulting front ROI

Figure 2: Video signals in the object detection algorithm.

2.2 MDC subsampling

The MDC subsampling unit permits splitting the input video or depth signal into two descriptions via a polyphase sub-

sampling (see the block ② in Fig. 1). The odd and the even pixel rows of the input ROI are separated into two fields creating a couple of subsequences that are coded independently by the following video source coder into two packet streams. In case both descriptions associated to the current ROI are correctly received and decoded, the input sequence can be reconstructed without any additional channel distortion. In case only one description is received, the vertical correlation among adjacent pixels of the even and the odd rows allows the Error Concealment unit (block ⑤ in Fig. 1) to estimate the lost description by interpolating the missing rows from the available ones.

In case both descriptions are lost, the Error Concealment unit replaces the missing information by copying the corresponding pixels of the previous frame. The following subsection describes how the generated subsequences are processed by the source coding unit.

2.3 Residual coding unit

After the object detection and the MDC subsampling units, the video or depth signal is coded into a packet stream that is transmitted to the end user. The input frame/field is partitioned into blocks $\mathbf{x}$ of 4×4 pixels which are approximated by the Motion Estimation unit that searches for a predictor block $\mathbf{x}_p$ in the previous frames/fields. According to the selected residual coding/decoding strategy (employed at block ③ in Fig. 1), the input signal $\mathbf{x}$ is then processed into different manners. Whenever the adopted residual coding strategy involves characterizing the Displaced Frame Difference (DFD), the source coder computes the prediction error block $\mathbf{d} = \mathbf{x} - \mathbf{x}_p$, which is then transformed into the block $\mathbf{D}$ via an approximated DCT transform defined within the standard H.264/AVC FREext [7]. The block $\mathbf{D}$ is then quantized into the coefficients $\mathbf{D}_q$, which are coded into a binary bit stream which is then packetized into a stream of RTP packets. The reconstructed signal can be obtained by dequantizing and inversely-transforming the block $\mathbf{D}_q$ into the decoded residual signal $\mathbf{d}_r = \mathbf{d} + \mathbf{e}_r$, which permits approximating the original block $\mathbf{x}$ with $\mathbf{x}_r = \mathbf{d}_r + \mathbf{x}_p = \mathbf{x} + \mathbf{e}_r$. Throughout these operations, the DFD coding strategy produces a H.264/AVC-compliant packet stream.

The error signal $\mathbf{e}_r$ is related to the quantization of $\mathbf{d}$ operated in the transform domain. Whenever the packet stream is affected by losses, the predictor block $\mathbf{x}'_p$ at the decoder differs from $\mathbf{x}_p$ since an additional channel distortion is present in the sequence such that $\mathbf{x}'_p = \mathbf{x}_p + \mathbf{e}_c$ and $\mathbf{x}_r = \mathbf{x} + \mathbf{e}_r + \mathbf{e}_c$. As a matter of fact, the distortion propagates throughout the sequence and degrades significantly the quality of the reconstructed sequence.

It is possible to mitigate this effect by choosing a more robust characterization of the residual signal. As it was proposed in [4], a source coding techniques that relies on the principle of Wyner-Ziv Coding (WZ) generates a set of symbols (called *syndromes*) which permits reconstructing the signal $\mathbf{x}$ from a set of different predictors. For each pixel $x(i,j)$ in $\mathbf{x}$ and its predictor $x_p(i,j)$ in $\mathbf{x}_p$, the WZ residual coding block computes the number of bits $n(i,j)$ associated to the syndrome $s(i,j)$ as

$$n(i,j) = \begin{cases} \log_2(|d(i,j)|) + 2 & \text{if } |d(i,j)| > \delta \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $|d(i,j)| = |x(i,j) - x_p(i,j)|$ and δ is a threshold value depending on the Quantization Parameter (QP) chosen for the current block (in our setting, we have set $\delta = \Delta/12$ where Δ is the quantization step associated to the current QP). Then, the coding unit computes the maximum value $n_{\max}$ of the syndrome bits $n(i,j)$ within the current block, and, in case $n_{\max}$ is higher than 0, it generates a block $\mathbf{s}$ of syndromes $s(i,j)$ via the following equation

$$s(i,j) = x(i,j) \& (2^{n_{\max}} - 1) \quad (2)$$

where the symbol & denotes a bitwise AND operators. The block $\mathbf{s}$ is then processed like the block $\mathbf{d}$ in the DFD strategy generating the block $\mathbf{S}_q$ of quantized transformed syndromes and the reconstructed syndromes $\mathbf{s}_r = \mathbf{s} + \mathbf{e}_r$. Each reconstructed syndrome $s_r(i,j)$ identifies a different quantizer $Q_{s_r(i,j)}$ with quantization step $2^{n_{\max}}$ and offset $s_r(i,j)$ such that the reconstruction levels can be expressed as $s_r(i,j) + k\,2^{n_{\max}}$, $k \in \mathbb{Z}$.

Given the predictor block $\mathbf{x}_p$, it is possible to reconstruct the coded pixel $x_r(i,j) = x(i,j) + e_r(i,j)$ by quantizing $x_p(i,j)$ using the quantizer $Q_{s_r(i,j)}$. Note that the signal $x_r(i,j)$ can be reconstructed using a different predictor $x'_p(i,j) \neq x_p(i,j)$ provided that the correlation between $\mathbf{x}$ and $\mathbf{x}'_p$ is the same or higher (see [4]).

2.4 FEC coder

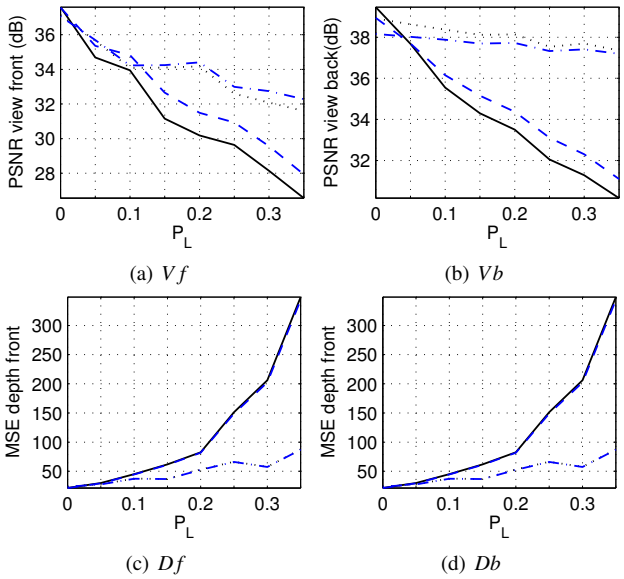
At packet level it is possible to reduce the amount of artifacts introduced by packet losses employing a protection strategy based on a cross-packet FEC code (see block ④ in Fig. 1). According to the protection strategy defined in the RFC 2733 [5], it is possible to generate in the RTP packet stream additional redundant packets which are correlated to the original packet sequence and permit recovering the lost data up to a certain number of lost packets. This protection scheme can be combined with the previous ones in order to maximize the final performance. In the following, the adopted configuration will be presented.

2.5 The adopted configurations

The presented video coder implements a hybrid highly-flexible architecture which needs to be appropriately tuned. In the following we will consider some of the possible configurations that will be adaptively employed to code the signals V_F, V_B, D_F, and D_B in order to maximize the final performance.

- **SD-DFD**: The input ROIs are coded into a single description, whose prediction residual is coded with the DFD configuration. The output packets are protected by additional FEC packets generated using block ④.
- **MD-DFD**: The input ROIs are split into two descriptions, whose prediction residual is coded with the DFD configuration. No additional FEC are generated.
- **SD-WZ**: The input ROIs are coded into a single description, whose prediction residual is coded with the WZ configuration, and additional FEC packets are added in the final packet stream.
- **MD-WZ**: The input ROIs are split into two descriptions, whose prediction residual is coded with the WZ configuration.

The CSC optimization unit chooses the most appropriate configuration according to the characteristics of the video signal and the packet loss percentages (estimated from RTCP



(a) Vf (b) Vb (c) Df (d) Db

Figure 3: Average PSNR and MSE values vs. P_L for signals Vf, Vb, Df, and Db from the sequence `breakdancers`. Lines reports the results for SD-DFD (solid), MD-DFD (dotted), SD-WZ (dashed), MD-WZ (dash-dotted).

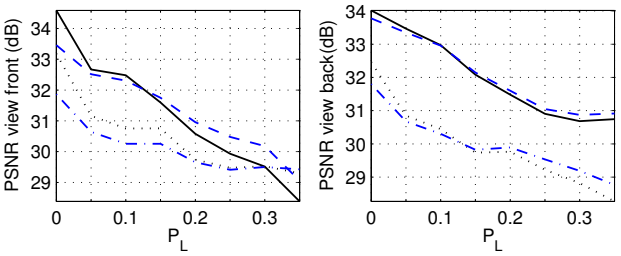
packets). Each Group-Of-Picture (GOP) in each sequence V_F, V_B, D_F, and D_B can be characterized with the array

$$\mathbf{G}_I = [S_y \; \nabla_t], \qquad I \in \mathscr{I} = \{V_F, V_B, D_F, D_B\}, \tag{3}$$

where S_y is the vertical Sobel operator averaged on the whole ROI I and ∇_t the overall temporal gradient with respect to the previous picture. In our approach, we considered the array $\overline{\mathbf{G}}_I$ that averages the arrays $\mathbf{G}_I$ for the ROIs I in the current GOP. Experimental results have shown that for sequences with different $\mathbf{G}_I$ at different packet losses the performance of each configuration significantly changes. As a matter of fact, it is necessary to design a classification strategy that identifies the most effective configuration.

3. CL OPTIMIZATION OF THE SOURCE CODER

Experimental results have shown that different configurations have different efficiencies depending on the characteristics of the video signals, on the coded object, and on the network state (see Figure 3). Under different transmission conditions the algorithm effectively adapts the coding strategy for fast moving objects in the foreground, while for the depth signal in the background the optimization strategy employs the MD-WZ configuration in most of the configurations. As a consequence, it is necessary to identify the coding setting that maximizes the quality of the reconstructed sequence for the current GOP. A first optimization approach relies on a Bayesian classification of the arrays $\overline{\mathbf{G}}_I$ conditioned on the loss probability and on the signal type. The arrays $\overline{\mathbf{G}}_I$ are partitioned into 4 classes according to which configuration C (among the settings SD-DFD, MD-DFD, SD-WZ, MD-WZ) proves to be the best one in terms of PSNR (for the video signal) or in terms of MSE (for the depth signal). More precisely, the classification strategy evaluates the probability



(a) Vf (b) Vb

Figure 4: Average PSNR vs. P_L for signals Vf and Vb from the sequence `horse`. Lines reports the results for SD-DFD (solid), MD-DFD (dotted), SD-WZ (dashed), MD-WZ (dash-dotted).

that the configuration C is the best for the signal I

$$P[C/\overline{\mathbf{G}}_I] = \frac{P[\overline{\mathbf{G}}_I/C] \cdot P[C]}{P[\overline{\mathbf{G}}_I]} \tag{4}$$

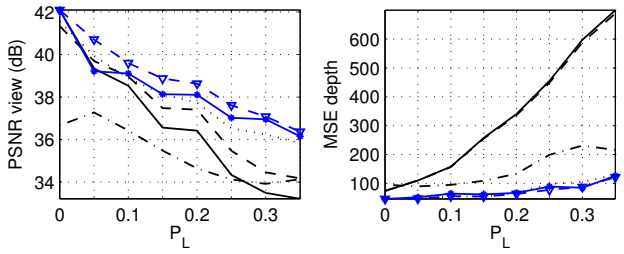
with $C \in \mathscr{C} = \{\text{SD-DFD}, \text{MD-DFD}, \text{SD-WZ}, \text{MD-WZ}\}$ and $I \in \mathscr{I}$ for different values of P_L. The conditioned probability $P[\overline{\mathbf{G}}_I/C]$ is modelled using a normal distribution $N(\overline{\mathbf{G}}_{I,C}, \sigma_{I,C})$, where parameters $\overline{\mathbf{G}}_{I,C}$ and $\sigma_{I,C}$ are estimated from a set of experimental data obtained for some training sequences. The same training set is also used to estimate the probabilities $P[C]$. Before coding the current GOP, the CSC optimization strategy computes $\overline{\mathbf{G}}_I$ for each signal I and computes the probability $P[C/\overline{\mathbf{G}}_I]$ via the equation (4). Then, the optimization strategy choose the coding configuration C_I^* such that

$$C_I^* = \arg\max_{C \in \mathscr{C}} P[C/\overline{\mathbf{G}}_I]. \tag{5}$$

$\forall I \in \mathscr{I}$. Experimental results show that this possibility permits improving significantly the quality of the resulting 3D experience.

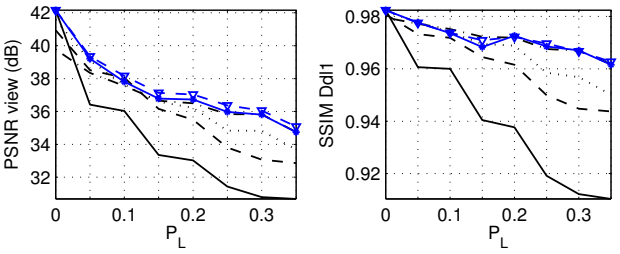
4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, we coded and simulated the transmission of several DIBR video sequences. The transmission channels have been simulated using a two-states Gilbert model with burst length $L_B = 4$ and varying loss probability P_L. In case MDC coding is adopted for the signal I (with $I \in \mathscr{I}$), two packet streams are generated and transmitted on independent channels. In the same way, the packets related to background and the packets related to front objects are transmitted on independent channel to increase the robustness of the transmission. Since the depth information is transmitted associated to the corresponding video data, the number of involved independent channels varies from 2 to 4 according to which coding configurations have been chosen. Sequences were coded using GOP with structure IPPP and constant QP equalizing the resulting overall bit rate for the different set-ups in $\mathscr{C}$. Figure 3 reports the average PSNR (for V_F and V_B signals) and the average MSE (for D_F and D_B signals) obtained with 10 channel realizations for different loss probabilities P_L. It is possible to notice that the effectiveness of the different configurations strongly depends on the signal characteristics. Experimental results on other sequences (see Figure 4) also



(a) Video (b) Depth

Figure 5: Average PSNR and MSE values vs. P_L for video and depth signals of the sequence `ballet`. Lines reports the results for SD-DFD (solid), MD-DFD (dotted), SD-WZ (dashed), MD-WZ (dash-dotted), CSC classification (solid with stars), optimal (dashed with triangles).



(a) V_F (b) video+depth

Figure 6: Average PSNR value of the video signal and SSIMDdl1 value vs. P_L for the sequence `car`. Lines reports the results for SD-DFD (solid), MD-DFD (dotted), SD-WZ (dashed), MD-WZ (dash-dotted), CSC classification (solid with stars), optimal (dashed with triangles).

show that the curves related to the different configurations C cross at different values of P_L according to the vertical and temporal correlations, which are measured by $\overline{\mathbf{G}}_I$. As a consequence, we trained the CSC optimization strategy using a set of heterogeneous sequences including `breakdancers` from [8], `horse` and `car` from [9]. The classifying strategy was tested on the sequence `ballet` to verify the accuracy of the training phase. Figures 5 and 6 compares the 3D visual quality obtained by the CSC algorithm (measured via the metrics PSNR, MSE, and SSIMDdl1[2]) with the performance obtained by each single static configuration for the sequence `ballet`. The proposed approach varies the coding mode for the different signals in order to maximize the final 3D experience. As an example, with $P_L = 0.1$ the first GOP of the sequence is transmitted with V_F coded with SD-WZ, V_B with MD-WZ, D_F with SD-WZ, and D_B with SD-DFD. The reported graphs also displays the performance associated to an optimal arrangement of the blocks which has been obtained via an off-line exhaustive optimization. It is possible to notice that the proposed CSC solution is quite close to the optimal setting both for test sequence (see 5) and one of the training sequences (see Fig. 6). Optimization proves to be easier for depth signals since they prove to be less complex to characterize with respect to video signals. As an evidence, it is possible to evaluate the distances of the CSC line from the optimal one in Fig. 5(a) for the video signal and in Fig. 5(b) for the depth information. A significant quality improvement is also evinced for the training sequences, as Fig. 6(a) shows. The PSNR values for the proposed CSC solution are greater than those of all the other configurations. As for the quality of the final 3D experience, Figure 6(b) shows that the average values of SSIMDdl1 metric for the CSC strategy is approximately equal to the best one.

5. CONCLUSION

The paper has presented a Cognitive Source Coding architecture that combines a Multiple Description Coding scheme with a traditional predictive video coder, a Wyner-Ziv video coder, and an FEC coder that introduces some additional redundant packets to protect the video packet stream from losses. An object detection unit classifies the different regions of the input frames according to their temporal and spatial characteristics. All the different configurations are optimized using a cognitive adaptive strategy given the characteristics of the signal to be coded and the network state. Experimental results show that the proposed scheme can identify the most effective solution for different signals and channel configurations.

[2]The SSIMDdl1 metric is intended to evaluate jointly the quality of both depth and texture signals (see [10] for more details).

REFERENCES

[1] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE J. Select. Areas Commun.*, vol. 23, no. 2, pp. 201 – 220, Feb. 2005, (Invited).

[2] C. Fehn, "3D-TV Using Depth-Image-Based Rendering (DIBR)," in *Proc. of PCS 2004*, San Francisco, CA, USA, Dec. 2004.

[3] R. Bernardini, M. Durigon, R. Rinaldo, P. Zontone, and A. Vitali, "Real-Time Multiple Description Video Streaming Over QoS-Based Wireless Networks," in *Proc. of ICIP 2007*, San Antonio, TX, USA, Sept. 2007, pp. 245 – 248.

[4] S. Milani and G. Calvagno, "Multiple Description Distributed Video Coding Using Redundant Slices and Lossy Syndromes," *IEEE Signal Processing Lett.*, vol. 17, no. 1, pp. 51 – 54, Jan. 2010.

[5] J. Rosenberg and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction (RFC2733)," *Internet Draft, Network Working Group*, Dec. 1999.

[6] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167 – 181, Sept. 2004.

[7] T. Wiegand, "Version 3 of H.264/AVC," in 12^{th} *JVT Meeting*, Redmond, WA, USA, July 17 – 23, 2004.

[8] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. of ACM SIGGRAPH 2004*, Los Angeles, CA, USA, Aug. 2004, pp. 600–608.

[9] "MOBILE3DTV Project: 3D Video database," http://sp.cs.tut.fi/mobile3dtv.

[10] A. Benoit, P. Le Callet, P. Campisi, and R. Cousseau, "Quality assessment of stereoscopic images," *EURASIP Journal on Image and Video Processing*, vol. 2008, Oct. 2008, ID 659024.