A Cognitive Approach for Effective Coding and Transmission of 3D Video

SIMONE MILANI and GIANCARLO CALVAGNO, University of Padova

Future multimedia applications will rely on the transmission of 3D video contents within heterogeneous fruition scenarios, and as a matter of fact, the reliable delivery of 3D video signals proves to be a crucial issue in such communications. To this purpose, multimedia communication experts have been designing cross-layer strategies to improve the quality of the perceived 3D experience. The paper presents a new cross-layer strategy, called Cognitive Source Coding (CSC), that defines a new 3D video system able to identify the different elements of the 3D scene and choose the most appropriate coding strategy.

Categories and Subject Descriptors: I.4.2 [**Image Processing and Computer Vision**]: Compression (Coding)—; H.4.3 [**Information Systems Applications**]: Communication applications—*Computer conferencing, teleconferencing, and videoconferencing*; I.2.10 [**Image Processing and Computer Vision**]: Artificial Intelligence—*3D/stereo scene analysis*

General Terms: Algorithms, Design, Measurements, Performance.

Additional Key Words and Phrases: Cross-layer optimization, source coding, joint source-channel coding, 3D video, cognitive source coding.

ACM Reference Format:

Milani, S. and Calvagno, G. 2011. A Cognitive Approach for Effective Coding and Transmission of 3D Video. ACM Trans. Multimedia Comput. Commun. Appl. 00, 00, Article 00 (June 2011), 20 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

During the last years, the widespreading of 3D video displays and applications has finally opened the way to the long-waited 3D video revolution. Movies, television broadcasting, gaming and video communication applications are adopting more and more frequently 3D video technology in order to increase the level of immersiveness and involvement of users. Novel 3D visualization schemes (such as Multiview Video, Free Viewpoint Video, etc.) permit a more realistic and interactive fruition of the three-dimensional scene, where the final user is allowed to choose the preferred viewpoint [Fehn 2004] and navigate in the displayed video scene as if it were actually present. This possibility has also been nurtured by the availability of new consumer 3D displays on the market, which enables a wider range of people with the possibility of viewing 3D video signals at different locations. Starting from these premises, wireless communications seem to provide the required versatility for an effective distribution of 3D video contents to different users thanks to the flexible topology (nodes can enter

Authors' address: Simone Milani and Giancarlo Calvagno are with the Dept. of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy; email: {simone.milani,calvagno}@dei.unipd.it.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

and leave the network more easily) and to the possibility of mobility offered to the different terminals. However, these multimedia systems pose new challenges to the designer of video and transmission architectures [Shi et al. 2009]. As a matter of fact, enabling an effective delivery of 3D video contents to a wide set of heterogeneous users under different fruition scenario is still an open issue in the ICT community.

One of the main problems that 3D video transmission systems need to face consists in the significant amount of data that characterize three-dimensional signals and have to be compressed in order to fit the constraints posed by the available transmission bandwidth. Moreover, the sensitivity to packet losses of compressed multimedia signals require robust coding and transmission solutions in order to grant a certain level of Quality-of-Experience (QoE) to the end user. As a matter of fact, the development of reliable coding algorithms and transmission protocols proves to be a crucial point for the diffusion of 3D video applications.

Research works have shown that traditional layered approaches in designing and configuring the communication protocol stacks often lead to misconfigurations and conflicting set-up. Cross-layer (CL) optimization strategies have proved to be more efficient in maximizing the quality of the received multimedia content [Katsaggelos et al. 2005] since the transmission parameters at the different layers are jointly tuned combining the effects of different protection strategies and taking full advantage of all the possibility offered by the whole transmission system. Among these, it is possible to mention the solutions proposed by Alregib *et al.* [2005] and by Balter *et al.* [2006].

However, CL solutions present some problems that preclude their effectiveness and severely limit their final performance. A first problem is related to the reliability and the stability of the estimated optimal configuration. The parameters that characterize the network state and the transmission conditions may be affected by a certain estimation error or could be obsolete since the channel conditions significantly vary during the time. These inaccuracies leads to a misconfiguration of the transmission system and to a poorer quality in the reconstructed signals. A second problem refers to the computational complexity required to compute the optimal configuration. As the number of transmission parameters increases by including additional layers in the optimization process, the set of possible joint configurations rapidly grows. Thereupon the identification of the optimal setting requires more complex solving algorithms which prove to be inappropriate for battery-powered mobile devices and prevent the interactivity in 3D multimedia communications.

Experimental results [Milani and Calvagno 2010a] have shown that these problems could be solved by adapting the source coding strategy to the characteristics of the transmitted video sequence and to the network state. Different source coding strategy, like Multiple Description Coding (MDC) or Distributed Video Coding[1] (DVC), present different levels of robustness, which usually increase as the compression efficiencies decrease. Moreover, their effectiveness strongly depends on the transmission scenario (e.g., MDC solutions prove to be more effective in a multipath scenario like in Peer-To-Peer networks). Changing the very architecture of the source coder leads to transmission set-ups which prove to be less sensitive to estimation noise in the channel state parameter and permit simpler optimization strategies.

In this paper we will refer to these solutions with the term Cognitive Source Coding (CSC) schemes in analogy to Cognitive Radio (CR) schemes [Haykin 2005] adopted for radio transmissions. However, while cognitivity in CR systems is only referred to the transmission environment, CSC systems sense and analyze both the network and the input signals adapting the coded data according to the target application and the transmitting conditions.

[1]In the paper, this coding mode is also called Wyner-Ziv coding from the name of the related theorem on which the coding strategy is based.

The paper present a new CSC coder, which is applied to a video+depth 3D sequence[2] optimizing both the quality of the reconstructed view and the accuracy of the received depth map. The proposed approach combines a Multiple Description Coder (MDC) with a Single Description Coder (SDC) and adaptively switches from a traditional predictive coding to Wyner-Ziv video coding. These coding strategies 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 stream against losses. At the same time, the proposed architecture permits reducing the computational complexity with respect to a standard H.264/AVC coder. Consequently, the CSC scheme can be effectively employed for both Video-On-Demand applications and Live 3D video transmission.

In the following, Section 2 reviews some of the CL approaches for 3D video transmission presented in literature and underlines the analogies between CSC and CR systems. Section 3 presents the adopted CSC scheme in detail by specifying its basic building blocks. Section 4 describes how the source coding strategy is optimized according to the characteristics of the video and depth signals. Experimental results in Section 5 show how it is possible to improve the 3D experience provided to the end user by adopting a Support Vector Machine (SVM)-based optimization of the different coding modes. Conclusions are drawn in Section 6.

2. STATE-OF-THE-ART FOR COGNITIVE SOURCE CODING

2.1 Related works on robust 3D video transmission

During the last years, different works have been focusing on the reliable transmission of multimedia and 3D video data over unreliable networks. The most widely-adopted solutions rely on a scalable coding of the data as they permit handling the different signals and varying the quality of the reconstructed scene more flexibly. The different signals need to be reconstructed with different accuracy according to the transmission capability of the network, their role in the rendering of the final 3D scene, the characteristics of displays, and the level of QoE for which the user is paying for. Several scalable coding solutions have been proposed in literature for stereoscopic signals ([Aksay et al. 2006] and [Karim et al. 2007]), as well as for DIBR sequences [Schierl et al. 2007].

Many approaches rely on a CL configuration of the transmission environment performing an Unequal Error Protection (UEP) of the transmitted data ([Alregib et al. 2005] and [Balter et al. 2006]).

Other solutions rely on characterizing the signal to be transmitted via multiple descriptions [Norkin et al. 2006; Karim et al. 2007].

A third set of strategies consider characterizing the 3D video signal using Wyner-Ziv coding solutions. One of these approaches can be found in [Yeo and Ramchandran 2007] where the DVC coder named PRISM [Puri and Ramchandran 2002] was adapted to the transmission of multiview video streams. Other works follow this strategy [Adikari et al. 2008] since a distributed source coder that exploits the correlation existing between different camera views permits a successful decoding of the transmitted information from any side view.

At the same time, Distributed Video Coding paradigms have been applied to traditional MDC schemes. In this case, the correlation between different descriptions permits an error-free reconstruction of the coded data using any of the available descriptions as reference. From the initial schemes in [Jagmohan and Ahuja 2003; Wu et al. 2004], Multiple Description Distributed Video Coding (referenced in literature with the acronym MDDVC or MDVC) have so far evolved into a wide range of different approaches. Some of the proposed solutions separately generate a set of descriptions and replace the traditional predictive coding for their compression with a Wyner-Ziv coder (see [Wang et al. 2006]).

[2]This 3D video format is also called Depth Image Based Rendering (DIBR) [Fehn 2004].

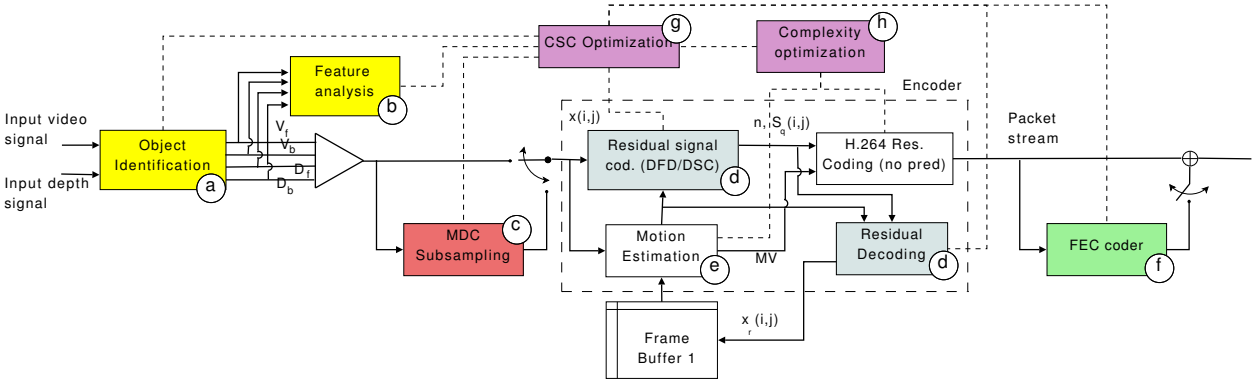


Fig. 1. Block diagram for the encoder.

Other solutions include the distributed source coding approach within the native MDC architecture, like the approaches in [Fan et al. 2003; Wang et al. 2009]. Many MDDVC solutions rely on Wyner-Ziv coding strategies employing a feedback-channel [Crave et al. 2008] like most of the previous DVC video coders [Aaron et al. 2002; Artigas et al. 2007]. Other solutions rely on a characterization of the residual signal similar to that of the PRISM coder [Puri and Ramchandran 2002] where no feedback information is needed [Milani and Calvagno 2009; 2010c].

During the last years, MDDVC approaches have been applied to the transmission of both stereoscopic and DIBR signals [Milani and Calvagno 2010b]. In fact, MDDVC significantly improves the transmission quality of DIBR video whenever the video signal presents strong correlation and the state of the network proves to be critical. From these preliminary results, it is possible to infer that performances can greatly benefit from an adaptation of the characteristics of the source coder to the features of the 3D data and of the network. This subset of cross-layer solutions is here referenced with the acronym CSC [Milani and Calvagno 2010a].

2.2 Analogies between Cognitive Source Coding and Cognitive Radio

As mentioned in the Introduction, Cognitive Source Coding (CSC) schemes have been named in analogy to Cognitive Radio (CR) schemes adopted for radio transmissions. As defined by Haykin [2005],

> "Cognitive radio is an intelligent wireless communication system that is aware of its surrounding environment (i.e., outside world), and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters (e.g., transmit-power, carrier-frequency, and modulation strategy) in real-time."

It is possible to notice that CSC schemes present many features in common with CR solutions. CSC architectures implement many source coding strategies and adaptively switch from one to another depending on the channel state. In a similar way, CR schemes implement many modulation schemes which are adaptively chosen according to the radio spectrum conditions. 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.

One additional feature of CSC schemes concerns the knowledge of the characteristics of the input signals which permit choosing the most appropriate robust coding solutions (e.g., static objects within a recorded video scene are easier to conceal with respect to fast-moving ones). This knowledge can also be employed to modulate the computational resources of the coding device according to the coding

complexity of each object to be transmitted. In this way, the cognitivity of the approach is present both at the channel side and at the source side of the source coder. In the end, both CSC and CR must provide a high degree of flexibility and reconfigurability.

As a matter of fact, CSC schemes need to be designed in appropriate way in order to satisfy specific requirements: providing robust multimedia communications anywhere and anytime while granting a certain level of Quality-of-Experience (QoE) to the end user; using effectively the available transmission capacity; limiting the required computational load, the involved hardware resources, and the complexity of the transmission architecture.

A first example of adaptive system that could be related to CSC can be found in [Reusens et al. 1996], where Reusens *et al.* introduce the concept of Dynamic Coding, i.e., a system that changes the coding strategy according to the characteristics of the input signal. However, in Dynamic Coding the robustness issues are not addressed and the whole system consists in a set of separate coders that are adaptively employed. As a result, the overall complexity required by the system increases with respect to implementing a single coding strategy. A second example can be found in many video coding standards that combine Intra with Inter coding modes (see [ITU-T and ISO/IEC JTC1 1994; ITU-T 1995]). This possibility has led to an adaptive mode switching coding strategy that permits increasing or decreasing the robustness of the coded data stream [Liao and Villasenor 2000]. Similarly, recent approaches combine DVC schemes with more traditional video coding solutions [Milani and Calvagno 2009].

However, in CSC solutions the coder is made of a set of basic building blocks which constitute the "minimum common factors" for each coding strategy supported by the scheme. It is possible to implement different coding architectures by simply reorganizing the different blocks and rewiring their connections (see Fig. 1). As a result, the computational complexity does not increase with respect to a standard coder. The details of the proposed CSC strategy will be presented in the next section.

3. THE PROPOSED CSC ARCHITECTURE

The previous section has underlined how the design of a CSC system relies on the identification of a set of coding strategies that present a common subset of features and building blocks. In this way, several elements can be reused and the overall complexity can be limited since it is easier for the video designer to keep the architecture simple. Following these guidelines, we were able to design a CSC system that switches among different coding solutions with a simple rewiring of the connections between building blocks.

Figure 1 shows a block diagram of the implemented video coder. The input data consists in a couple of video streams made of a standard video signal and its related depth information. Each frame (named V and D for the video and the depth signals, respectively) is partitioned into two subregions (block ⓐ) which distinguish the background and the static objects (grouped in the regions V_b and D_b) from the foreground and the moving objects (included in the regions V_f and D_f). As a result, the partitioning strategy, which will be described in detail in Subsection 3.1, generates four different video subsignals where frames are replaced by pseudo-frames made of the regions V_f, V_b, D_f, and D_b of the original pictures or depth maps.

Each subsignal $S \in \{V_f, V_b, D_f, D_b\}$ is then analyzed independently throughout a single GOP of frames, and a set of features characterizing the error resilience and the complexity of the object is computed (block ⓑ). This information is used by the optimization blocks ⓖ and ⓗ in order to find the most appropriate CSC configuration for a given state of the channel.

For each subsignal, the CSC coder may choose whether processing the subsignal as it is or splitting the original data stream into two descriptions S^1 and S^2. The first option is called Single Description Coding (SDC), while the second is referred as Multiple Description Coding (MDC). In case the MDC

option is enabled (block ⓒ), the original subsignal is processed by a polyphase sampler that separates odd and even lines of pixels into two subsequences ("descriptions"). More details about the MDC option will be presented in Subsection 3.2.

After the optional MDC splitting, either the single description S or the two descriptions S^1, S^2 are coded independently. More precisely, each frame is predicted by a Motion Estimation unit (block ⓔ). The prediction error can be compensated either coding the residual signal like in a traditional video coder (see [Wiegand 2004]) or employing a Wyner-Ziv coder that permits a more robust characterization of the signal (see the blocks ⓓ). For more details about these two coding options, see Subsection 3.3. The processed signals are then converted into a binary stream, which is handled by a network adaptation unit that encapsulate the data and adapt its format in order to make possible its transmission across several types of networks. In our implementation we considered the possibility of transmitting the coded information via RTP packets carried on a UDP/IP network (where no retransmission are allowed because of the constraints on time delays implied by multimedia transmission), but other transmission formats and protocols can be considered as well. Since each subsignal or description is coded independently, after the source coding operation the number of independently-decodable packet stream varies from a minimum of 4 (all the subsignals S are coded with a single description) to a maximum of 8 (MDC is chosen for all S).

In the end, it is possible to include some redundant packets generated via an FEC channel coder (block ⓕ). This additional information permits recovering the lost packets whenever their number is lower than a given threshold. However, since the MDC option already introduces extra redundancy in the stream, the FEC option is enabled only in the case of single description coding. More details can be found in Subsection 3.4.

Both video and FEC packets are then transmitted to the decoder via a simulated channel affected by packet losses. After the FEC decoder has recovered the lost data (whenever it is possible), the packet stream is decoded, and the 3D video sequence is reconstructed. In the following we will describe each functional block in detail.

3.1 Segmentation of the input frames into subregions

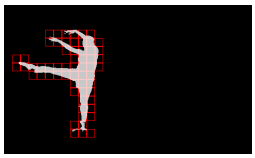
The first unit of the proposed CSC scheme partitions the input video V and depth D frames into two subregions generating the signals V_f, V_b, D_f, and D_b, where the subscripted letters f and b separate signals including foreground and fast-moving elements (associated to the pixel set $\mathcal{F}$) from static and background elements (associated to the pixel set $\mathcal{B}$).

In the design of the CSC strategy, we relied on the assumption that elements in $\mathcal{F}$ have a stronger impact on the final visual quality and their data can not be easily estimated by the error concealment unit whenever they get lost. This presumption is motivated by the fact that foreground objects usually drag the attention of the viewer watching the scene. As a consequence, many saliency maps for 3D images identify the most salient points of the acquired 3D scene salient in points close to the camera [Bremond et al. 2010]. Moreover, motion information is difficult to recover after data losses and therefore, the related protection level has to be increased. Elements in $\mathcal{B}$ instead have a minor impact on the final visual quality and can be easily approximated with small distortion whenever they get lost. The classification routine partitions the input frame into N_R small regions R_k (also called "superpixels" [Saxena et al. 2009]) via an oversegmentation of the depth signal [Felzenszwalb and Huttenlocher 2004] (see Fig. 2(c) as an example). The parameters of the segmentation routine are set in such a way that around 100 segments per image are created. For each superpixel $R_k \subset \mathbb{Z}^2$, $k = 0, \ldots, N_R - 1$, the

(a) Input view

(b) Input depth

(c) Regions R_k



(d) Resulting front ROI

Fig. 2. Video signals in the object detection algorithm.

object identification unit ⓐ computes the features

$$act_k = \frac{1}{|R_k|} \sum_{(x,y)\in R_k} \|V_t(x,y) - V_{t-1}(x,y)\| \qquad \overline{d}_k = \frac{1}{|R_k|} \sum_{(x,y)\in R_k} D_t(x,y) \tag{1}$$

where $V_t(x,y)$ and $D_t(x,y)$ are the pixels at position (x,y) of the frame at time t from the texture and depth signals, respectively. Parameter $|R_k|$ is the cardinality of R_k. In case the parameter act_k is lower than a given threshold T_1, R_k is assigned to the set $\mathcal{B}$.[3]

The values d_k related to the remaining R_k outside $\mathcal{B}$ are then clustered into 10 different classes. The regions of the same cluster are merged together creating a new set of areas R'_m and the average depth values d'_m are recomputed on the new regions. All the regions R'_m for which d'_m is higher than a given threshold are assigned to $\mathcal{F}$, while the remaining are included in $\mathcal{B}$.

The two sets $\mathcal{F}$ and $\mathcal{B}$ distinguish two different classes of objects in the scene and it is possible to signal them to the decoder using a mask coding algorithm like in object-based video coding architecture [ISO/IEC JTC1 2001]. In our implementation, we leave this possibility for future implementations, and, in order to limit the computational complexity, we partition the input frames into two Regions-Of-Interest (ROI). The first one is made of the set of macroblocks M_f including $\mathcal{F}$ (front ROI), while the second one is made of macroblocks M_b related to $\mathcal{B}$ (back ROI) (see Fig. 2(d) as an example). From this assignment of the macroblocks of the frame, four subsignals V_f, V_b, D_f, and D_b are generated including macroblocks M_f and M_b of texture and depth signals respectively. Each subsignal is then compressed using a separate source coding strategy.

3.2 Characterization of the signal via multiple description

After the partitioning into foreground and background subsignals operated by block ⓐ, the control unit decides whether each subsignal has to be coded as is (SDC) or needs to be split into multiple correlated descriptions (MDC) by unit ⓒ in Fig. 1. MDC schemes split a single information source into multiple correlated subsignals that are independently coded and transmitted to the receiver [Goyal 2001]. Whenever some data get lost, it is possible to estimate the missing information from the available descriptions exploiting the existing correlation.

In the scheme of Fig. 1, block ⓒ implements an MDC scheme based on a vertical polyphase subsampling of the pixel rows into two descriptions. Whenever the MDC option is enabled, signals V_f, V_b, D_f,

[3] In our implementation, T_1 is set to 4.

and D_b are split into the subsequences V_f^m, V_b^m, D_f^m, and D_b^m, $m = 1, 2$, including the odd and the even rows of the input signals, respectively.

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 (functional block ⓔ in Fig. 1) to estimate the lost description by interpolating the missing rows from the available ones. In case both descriptions are lost, the missing fields are approximated by copying the corresponding pixels of the previous frame into the missing ones. This solution is also adopted whenever the input signal is coded using a single description (the MDC option is disabled).

The following subsection describes how the generated subsequences are processed by the source coding unit.

3.3 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 16×16 pixels (macroblocks) which are approximated by the Motion Estimation procedure (see the unit ⓔ in Fig. 1) that searches for a predictor block $\mathbf{x}_p$ in the previous frames/fields. According to the selected residual coding/decoding strategy (employed at units ⓓ in Fig. 1), the input signal $\mathbf{x}$ is then processed differently according to the chosen coding mode.

Whenever the adopted residual coding strategy involves characterizing the Displaced Frame Difference (DFD), the source coder computes the prediction error block $\mathbf{d} = \mathbf{x} - \mathbf{x}_p$. The block $\mathbf{d}$ is then processed according to the standard H.264/AVC FREext residual coding strategy [Wiegand 2004]. The block is transformed and quantized into the block $\mathbf{D}_q$ of coefficients, which are included into a stream of RTP packets, together with motion vectors and header data. The reconstructed signal can be obtained by dequantizing and inversely-transforming the block $\mathbf{D}_q$ into the decoded residual signal $\mathbf{d}_r = \mathbf{d} + \mathbf{e}_r$, where the additional component $\mathbf{e}_r$ is related to the quantization of $\mathbf{d}$ operated in the transform domain. The reconstructed prediction residual $\mathbf{d}_r$ permits approximating the original block $\mathbf{x}$ with $\mathbf{x}_r = \mathbf{d}_r + \mathbf{x}_p = \mathbf{x} + \mathbf{e}_r$.

Whenever the packet stream is affected by losses, the predictor block $\mathbf{x}'_p = \mathbf{x}_p + \mathbf{e}_c$ at the decoder differs from $\mathbf{x}_p$ since an additional channel distortion $\mathbf{e}_c$ (related to the error concealment strategy) degrades significantly the quality of the reconstructed sequence [Färber et al. 1999].

It is possible to mitigate this effect by choosing a source coding technique that relies on the principle of Wyner-Ziv Coding (WZ) and 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 a syndrome $s(i, j)$ of $n_{\max}$ bits via the same procedure described in [Milani and Calvagno 2010c] and in [Milani and Calvagno 2009]. More precisely, $s(i, j)$ corresponds to the $n_{\max}$ least significant bits of $x(i, j)$

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

where $\&$ denotes a bitwise AND operator, and $n_{\max}$ has been computed from the correlation between the pixels of block $\mathbf{x}$ and the pixels of block $\mathbf{x}_p$ (see [Milani and Calvagno 2010c] for more details). In our implementation, the correlation is measured via the prediction error $\mathbf{d}$ leading to

$$n_{\max} = \max_{i,j=0,\ldots,3} \{n(i, j)\} \tag{3}$$

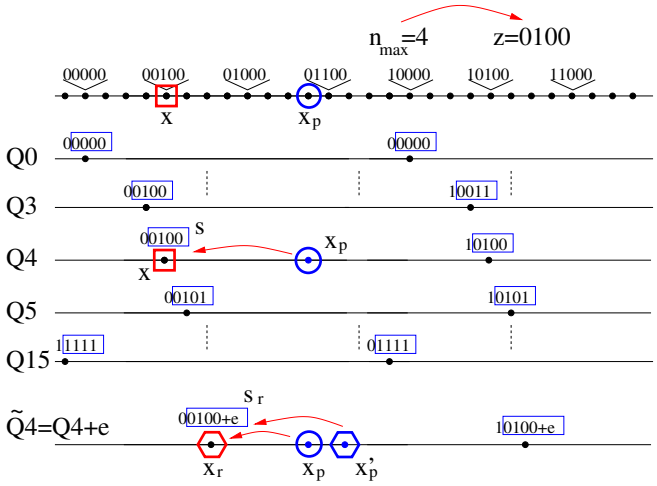


Fig. 3. Computation of the syndrome $s(i,j)$ in the WZ coder and its decoding. For the sake of clarity, indexes (i,j) have been omitted.

where

$$n(i,j) = \begin{cases} \lfloor \log_2 (|d(i,j)|) \rfloor + 2 & \text{if } |d(i,j)| > \delta \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

The parameter δ is a threshold value depending on the Quantization Parameter (QP) chosen for the current block (in our implementation, we have set $\delta = \Delta/12$ where Δ is the quantization step associated to the current QP).

The syndrome generation procedure inherits the nested scalar quantization approach of previous Wyner-Ziv coding schemes [Puri and Ramchandran 2002] but operates in the pixel domain on the original signal. The value $s(i,j)$ permits reconstructing $x(i,j)$ from $x_p(i,j)$ by identifying a quantizer characteristics $Q_{s(i,j)}$ (with quantization step $2^{n_{\max}}$ and offset $s(i,j)$) and selecting the closest output level to $x_p(i,j)$. Fig. 3 shows an example of syndrome generation and decoding, where the syndrome $s(i,j) = 0100 = 4$ obtained with $n_{\max} = 4$ selects the quantization characteristics Q_4. The output level of Q_4 that is closer to x_p corresponds to the original pixel $x(i,j)$. In order to reduce the number of $n(i,j)$ to be signalled in the bit stream, the WZ coder employs the maximum value $n_{\max}$ for all the pixels in the block since a correct reconstruction is possible whenever the chosen number of bits is greater than $n(i,j)$.

As a result, the WZ coder produces a block of syndromes s from the original pixel block x, which is then processed like the block d in the DFD strategy generating the block $\mathbf{S}_q$ of quantized transformed syndromes and the reconstructed syndromes $\mathbf{s}_r = \mathbf{s} + \mathbf{e}_r$. In this way, the quantizer characteristics $Q_{s_r(i,j)}$ selected by $s_r(i,j)$ presents its output values slightly shifted of $e_r(i,j)$ and reconstructs the coded pixel $x_r(i,j) = x(i,j) + e_r(i,j)$ with additional source coding distortion.

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 x and $\mathbf{x}'_p$ is the same or higher (see [Milani and Calvagno 2010c]).

3.4 FEC coder

The previous subsections have presented some strategies that permit mitigating the channel distortion via robust source coding schemes. 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 ⓕ

in Fig. 1). According to the protection strategy defined in the RFC 2733 [Rosenberg and Schulzrinne 1999], 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.

4. OPTIMIZATION

Previous sections have presented the different functional blocks of the CSC scheme in Fig. 1 that can be rewired and reorganized in order to implement different coding strategies. For the sake of conciseness, we identify here some of the possible configurations.

—**SD-DFD**: The input signal is coded into a single description (SDC option), and prediction residuals are coded with the DFD configuration (standard H.264/AVC coding). Additional FEC packets are generated by block ⓕ to increase the protection level.

—**MD-DFD**: MDC option is enabled, and the prediction residuals of descriptions are coded with the DFD configuration. No FEC packets are added.

—**SD-WZ**: The input signal is coded into a single description (SDC option), band the prediction residual is coded with the WZ configuration. Output RTP packets are then processed by the FEC coder to add redundant information that protects the data stream.

—**MD-WZ**: The input signal is split into two descriptions like in the MD-DFD configuration, whose prediction residuals are coded with the WZ configuration. The FEC coder is disabled.

Previous works [Milani and Calvagno 2010a] have shown that the effectiveness of each configuration varies according to the channel characteristics, the features of the signal, and the information it carries (either geometry or texture). More precisely, the different objects in the scene present heterogeneous characteristics, and as a matter of fact, the error concealment unit performs quite differently. Fast moving elements are difficult to estimate from the previously-decoded frames, and therefore, their loss introduces a significant amount of distortion in the reconstructed sequence. Moreover, the attention of viewers focuses on foreground and close objects. As a matter of fact, preserving the quality of these elements and reducing the loss probability for the packets that code them affect significantly the 3D experience offered to the end users.

This performance disparity related to the features of the signals also affects the required hardware resources in the coding process. For background and slowly-moving elements, the motion estimation unit do not need a wide search window in order to compute motion vectors and a simple block partitioning proves to be enough. On the other hand, the motion of foreground and quickly-moving objects spans larger regions with respect to the previous frame and requires more complex block partitioning that need to be tested. Since an accurate characterization of such coding elements is crucial, more computational resources have to be dedicated to this second class of objects.

In the proposed approach, a set of features is extracted from the input subsequences (block ⓑ in Fig. 1) and processed by a machine learning algorithm (block ⓖ in Fig. 1) that identifies the best coding configuration and the computational setting. Since the performance of both error concealment and motion search is related to spatial and temporal correlation, unit ⓑ computes the array of features

$$\mathbf{v}_S = [\Delta_y(S)\ \Delta_t(S)] \qquad S = V_f, V_b, D_f, D_b \tag{5}$$

which groups spatial and temporal gradient measures. The parameter $\Delta_y(\cdot)$ is the average value of the vertical Sobel operator (which measures the vertical correlation affecting the efficiency of the MDC

approach) and $\Delta_t(\cdot)$ is the temporal gradient between adjacent frames (which measures the temporal correlation affecting the error concealment and the efficiency of the WZ approach with respect to its DFD counterpart). A separate gradient is computed for each signal characterizing the 3D video sequence (V_f, V_b, D_f, and D_b), and each $\mathbf{v}_S$ is averaged for all the frames in a Group-Of-Picture (GOP) creating the value $\overline{\mathbf{v}}_S$. In this way, it is possible to adapt the transmission parameters to the varying features of the signals both limiting the effects of outlying data on the signal statistics and modifying the configuration of the system with a sufficient frequency.

The parameter space covered by the values of $\overline{\mathbf{v}}_S$ is then partitioned into 4 regions (labelled by the parameter $C \in \mathcal{C}$) including the feature values that characterize subsignals for which the setting C is the best CSC configuration among the set $\mathcal{C}$. The partitioning has been performed by a non-linear Support Vector Machine (SVM) classifier [Boser et al. 1992] that selects the best setting in the set $\mathcal{C} = \{\text{SD-FEC, MD-FEC, SD-WZ, MD-WZ}\}$.

Each region associated to the configuration $C \in \mathcal{C}$ can be described by a subset of support vectors $\mathbf{z}^C_{S,h}$, $h = 0, \ldots, N^C_S - 1$. Given the array $\overline{\mathbf{v}}_S$, the normal vector

$$\mathbf{w}^C_S = \sum_{h=0}^{N^C_S-1} \alpha^C_{S,h}\, \mathbf{z}^C_{S,h} \tag{6}$$

and the offset q^C_S, the classifier computes the discriminant function

$$c(\overline{\mathbf{v}}_S, \mathbf{w}^C_S) = <\overline{\mathbf{v}}_S, \mathbf{w}^C_S> - q^C_S = \sum_{h=0}^{N^C_S-1} \alpha^C_{S,h} <\overline{\mathbf{v}}_S, \mathbf{z}^C_{S,h}> - q^C_S \tag{7}$$

where the product $< \mathbf{v}_S, \mathbf{z}^C_{S,h} >$ is defined by the kernel function $K(\mathbf{v}_S, \mathbf{z}^C_{S,h})$.

Final class C^*_S is chosen computing

$$C^*_S = \arg\max_{C \in \mathcal{C}} c(\overline{\mathbf{v}}_S, \mathbf{w}^C_S) \tag{8}$$

for each subsequence S. In our experiments, we tested different kernel functions [Boser et al. 1992; Boughorbel et al. 2005] in order to find which $K(\cdot,\cdot)$ maximizes the number of correct classifications and the quality of the reconstructed 3D sequence. The tested functions are the following:

$$\begin{aligned}
&\text{Polynomial:} && K(\mathbf{v}, \mathbf{v}') = (\mathbf{v} \cdot \mathbf{v}')^{\alpha} \\
&\text{Gaussian:} && K(\mathbf{v}, \mathbf{v}') = \exp\left(-\frac{\|\mathbf{v} - \mathbf{v}'\|^2}{2\,\sigma^2}\right) \\
&\text{Logarithmic:} && K(\mathbf{v}, \mathbf{v}') = -\log\left(1 + \|\mathbf{v} - \mathbf{v}'\|^{\alpha}\right)
\end{aligned} \tag{9}$$

where the parameters α, σ are found during the training phase by minimizing the misclassification probability.

Previous works have shown that the optimal configuration depends on the packet loss probability as well. Whenever loss probability is low, the classifier must choose the configurations that permit obtaining high compression ratios since robustness is a less crucial issues. On the other hand, high packet loss rates require loss resilient configurations that present a lower coding gain. As a matter of fact, different sets of support vectors must be identified as the loss probability changes. The classification routine partitions the possible loss percentage values into 5 intervals, and for each range of values a different set of classifiers (one per each subsignal S) is computed in order to adapt the choice of the best mode to the state of the channel.

Since both video and FEC packets are transmitted using the RTP protocol, the loss probability P_L can be estimated from the incoming RTCP packets which are defined within the RTP specifications and contain additional information about transmission statistics (lost packets, throughput). According to the draft specification [Schulzrinne et al. 1996], RTCP information should not be transmitted more often than every 5 s (usually 5% of the employed transport bandwidth can be dedicated to RTCP). In our simulations, we adopted an RTCP packet frequency equal to 5 s. Every time an RTCP packet arrives, the CSC optimization unit ⓖ in Fig. 1 uses RTCP information to update the value of parameter P_L and to select the corresponding set of 4 SVM classifiers for the subsignals V_f, V_b, D_f, and D_b.

Before every classification, a whole GOP of frames is buffered and the optimization unit computes the array $\overline{\mathbf{v}}_S$ from the stored frame. The array $\overline{\mathbf{v}}_S$ is classified using the selected SVM mapper (one for each subsignal) identifying the most appropriate coding choice. These operations are iterated at every GOP.

The identified classes also permits varying the computational set-up for motion estimation and block partitioning units. Block ⓗ in Fig. 1 varies the size of the search window and the number of possible macroblock partitions. In the literature, several fast motion search algorithms adapting the size of the search window have been proposed (see [S. Goel and Bayoumi 2005] as an example). In these strategies, the most appropriate window size are learned "on-the-run" from the coding results of previously-coded blocks. As a matter of fact, the learning phase requires a transient period to converge that imply some misconfigurations of the coders and the relative poorer performances in terms of compression gain. However, in the CSC approach it is possible to take advantage of the object identification performed for error resiliency purposes in order to tune the motion search parameters from the start. Experimental results show that this configuration improves coding time saving with respect to learning algorithms (equalizing the rate-distortion performance). More precisely, the dimensions of the window search are reduced to one fourth for the V_b and the D_b with respect to the window size for the V_f and D_f since background and static objects present limited motion vectors. In addition, the only macroblock partition for the background signals is 16×16 since the motion is not as complex as that of foreground objects and occlusions are not present (moving objects and surrounding pixels are included in the signals V_f and D_f).

On the contrary, the foreground elements need to be coded efficiently since they present complex motion and structure which require a higher amount of bits (with respect to the number of coded pixels). As a matter of fact, no reductions in the size of search window and in the number of partitioning mode is performed. This simplification leads to a significant reduction of the computational complexity in the encoding phase, with respect to a standard H.264/AVC coder as it is underlined in Section 5.

Moreover, since the structure of the CSC coder has been derived from the H.264/AVC architecture, the implementation of the proposed CSC scheme inherits a huge set of off-the-shelf solutions and optimizations which have been designed and tested for the previous video coders (e.g., fast motion search modules, optimized arithmetic coder, transform and quantization unit).

4.1 A real-time implementation of the CSC coder

The proposed CSC approach has been adapted and optimized in order to run in real-time on a low-profile hardware using Microsoft Xbox Kinect sensor to acquire both color and geometry information from the scene. The system has been implemented on a netbook with 1.7 GHz CPU speed and 2GB RAM. The whole architecture implements a warping routine in order to map data from depth camera to color information, a full CSC encoder, a simple channel simulator, and the CSC decoder. Segmentation is replaced by a macroblock classification routine that chooses whether the MB belongs to $\mathcal{F}$ or $\mathcal{B}$ according to the average activity of the macroblock (see eq. (1)) and the maximum depth value in the MB. Motion estimation is performed on the luminance component only, and the motion search window

is adapted according to the distance from the camera and the amount of motion of the objects. An evaluation of the performance of this implementation and further information can be found consulting the additional material published with this paper and reported at [Milani 2011].

In the following, experimental results will show how this classification permits increasing the visual quality of the reconstructed 3D sequence and reducing the computational complexity with respect to a standard H.264/AVC coder.

5. SIMULATION RESULTS

The proposed solution has been tested simulating the transmission of a wide set of DIBR sequences over a set of erasure channels affected by varying loss conditions. Each channel simulates the loss of an RTP packet via a two-state Gilbert model with average burst length $L_B = 4$ and loss probability P_L, which can vary in the range $[0.05, 0.25]$ with steps of 0.05. The RTP packets related to the subsequences V_f, V_b, D_f, and D_b are transmitted over four independent channel realizations. In case the MDC option is enabled, the odd and even packets are sent to independent channels, i.e., odd packets of foreground objects are sent over the same channel of the even packets including the background elements and viceversa. The transmission performance has been simulated computing the quality of the reconstructed signals after 10 channel realizations and averaging the values of several quality metrics.

Quality evaluation for video communications has also been a discussed task whenever related to measuring the liking and level of comfort of people watching the reconstructed video sequences. As a matter of fact, different quality metrics have been recently proposed in literature with the purpose of calculating numerical values correlated with the human perception of visual quality. Together with the traditional PSNR value, it is possible to compute the Structural Similarity Index Metrics (SSIM) [Wang et al. 2004] in order to compare the Quality-of-Experience provided to the end user by a video transmission scheme. 3D video signals have introduced the need for characterizing the quality of 3D experience offered to the end user. As for the geometry signals, Mean Square Error (MSE) is widely employed to evaluate the accuracy of the reconstructed depth maps since it permits measuring the average discrepancy of the transmitted volumes with the original ones in the uncompressed depth map.

Recent works have been focusing on evaluating the quality of 3D video signal. Some research have been done in the past for static 3D images, but measuring the quality of 3D video is a brand new field. To this purpose, some papers in literature have proposed combined metrics that take into account both the reconstructed depths and views. In this work, we adopt the SSIM Ddl1 metric in [Benoit et al. 2008]. Note that PSNR, SSIM, and SSIM Ddl1 are quality metrics, i.e., their values increase as the visual quality of the reconstructed 3D scene improves, while MSE is a distortion measures, i.e., the reconstructed volumes appears to be closer to the original ones as the MSE value decreases.

The video source coding engine has been derived from the structure of the H.264/AVC coder reusing many of the functional units already present in the coder (like the Motion Estimation engine, the processing unit for the residual signal, and the arithmetic binary coder CABAC). The Motion Estimation unit adopts a simple Inter-type predictive coding (no B frames are used), which is replaced by an Intra coded picture at the beginning of each GOP (GOP IPPP of 15 frames). The input signals have been coded with fixed Quantization Parameter (QP) and equalizing the resulting bit rates for the different configurations C varying the value of the fixed QP. The channel coding rate of the FEC coder has been set to 0.2 since in this way it is possible to equalize the additional redundancy introduced by MDC coding.

Initial tests were devoted to compare the robustness of the different configurations and train the SVM classifier. Sequences were downloaded from the Microsoft Research [Microsoft Research 2011], the HHI and the Mobile3D [Mobile3DTV project 2011] repositories, and their formats have been equal-

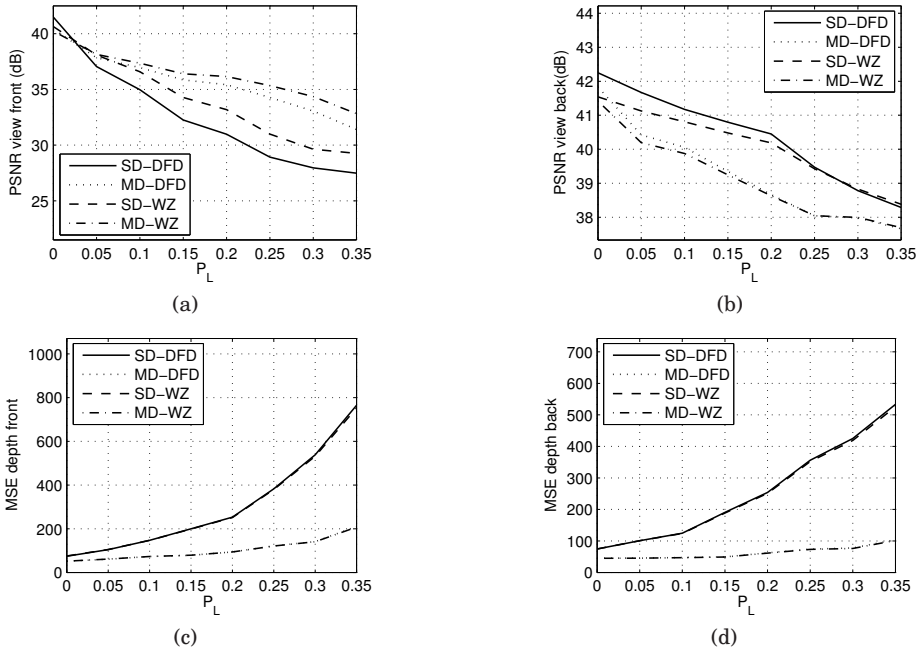


Fig. 4. Average PSNR and MSE values vs. P_L of the subsequences V_f, V_b, D_f, and D_b of the sequence `ballet` for the CSC configurations SD-DFD, MD-DFD, SD-WZ, and MD-WZ. a) PSNR for V_f b) PSNR for V_b c) MSE for D_f d) MSE for D_b.

ized in order to process video sequences with the same resolution. Additional sequences generated via a standard video camera and a lateral Time-of-Flight range cameras that produce a 640×384 color signal with an additional depth map have been included. Depth maps are estimated via a structured light system with 12 bits of precision. These data have been rescaled to 8 bits in order to be compliant with the other sequences. Similarly, we acquired an additional video+depth sequence using a Microsoft Kinect sensor, which adopts a depth estimation approach based on structured light. This system permits obtaining a DIBR sequence with 11-bits depth samples, which are equalized to 8 bits also in this case. The sequences are available at [Milani 2010].

Sequences `breakdancers`, `ballet`, `car`, `horse` have been used in the training of the classifier, i.e., finding the kernel parameters that permits the classifier to minimize the rate of misclassifications and reliably identify the best configuration C in terms of visual quality for the reconstructed sequence. In order to validate this classification, sequences `interview` and `orbi` by HHI [Fraunhofer HHI 2011] and `umbrella` have been used as validation sequences.

Figure 4 reports the average PSNR and MSE values vs. the loss probability P_L for the sequence `ballet`. It is possible to notice that at low loss rates the SD-DFD configuration provides the best performance, while at high loss rates MD-WZ permits obtaining higher average PSNR values (see Figure 4(a)). Note that this behavior is strictly dependent on the characteristics of the processed signal S since, for a given loss probability P_L, the best configuration for the different subsequences V_f, V_b, D_f, and D_b may change. As an example, from Fig. 4 it is possible to infer that the best configuration at $P_L = 0.05$ is SD-WZ for V_f, SD-DFD for V_b, and MD-WZ for D_f and D_b. The most effective CSC configuration is also dependent on the characteristics of the coded sequence. Figure 5 reports the average PSNR values vs. P_L for the signal V_f of the sequences `breakdancers` and `horse`. It is possible to notice that at packet loss $P_L = 0.1$ the configuration SD-DFD still proves to be convenient for the sequence

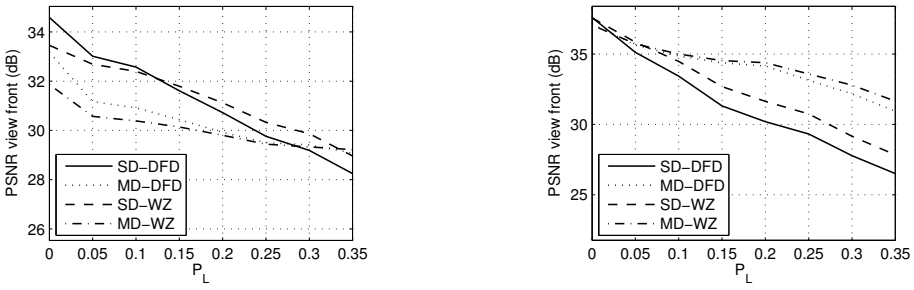


Fig. 5. Average PSNR and MSE values vs. P_L of the signal V_f for the sequences `horse` and `breakdancers` related to the different CSC configurations. a) `horse` b) `breakdancers`.

`horse` (see Fig. 5(a)), while it obtains a significantly poorer performance with respect to the configuration MD-WZ for the sequence `breakdancers` (see Fig. 5(b)). An explanation for this behavior has to be found on the temporal correlation of the two sequences. The sequence `horse` is quite static and presents a strong temporal correlation which permits the error concealment strategy to estimate the lost information effectively. As a matter of fact, there is not an urgent need for a robust coding strategy and it is possible to adopt a configuration that permits obtaining high coding gains. On the contrary, the frames of the sequence `breakdancers` present a lower temporal correlation level, and therefore, an error-resilient setting is required. Final tests were devoted to compare the performance of different SVM classifiers on both training and test sequences. Figure 6 reports the average values of different metrics obtained with different optimization algorithms for the sequence `breakdancers`. The graphs also show an envelope function E_M

$$\begin{aligned} E_M &= \max\{M_{\text{SD-DFD}}, M_{\text{MD-DFD}}, M_{\text{SD-WZ}}, M_{\text{MD-WZ}}\} \qquad \text{for } M = \{\text{PSNR}, \text{SSIM}, \text{SSIMDdl1}\} \\ E_M &= \min\{M_{\text{SD-DFD}}, M_{\text{MD-DFD}}, M_{\text{SD-WZ}}, M_{\text{MD-WZ}}\} \qquad \text{for } M = \text{MSE} \end{aligned} \tag{10}$$

where M_S is the average value of the metric M obtained with the configuration S. Moreover, plots also report the results obtained via an off-line optimization that selects the best coding strategy for each GOP maximizing the visual quality related to each metric. It is possible to notice that the proposed SVM-based optimization achieves equal or better results with respect to the envelopes of the metrics (see Fig. 6(a)), and the related curves proves to be quite close to those reporting the results for the off-line optimization. The same behavior can be noticed via different metrics (see Fig. 6(b) and Fig. 6(d)) and for the depth signal too (see Fig. 6(c)). As a matter of fact, it is possible to infer that the designed SVM classifier permits identifying the optimal setting at different transmission conditions for each subsignal. From the reported plots, it is possible to notice that the logarithmic and the polynomial kernels prove to be the most effective, while the Gaussian kernels provides a lower quality of the reconstructed sequence. Figure 7 reports the simulation results obtained on the sequences `ballet` (training) and `orbi` (test). The proposed approach improves the average PSNR value of 1 dB at $P_L = 0.2$ for the texture signal of the sequence `ballet` with respect to the envelope E_{PSNR} (see Fig. 7(a)). As for the sequence `orbi`, the performance of the SVM classifier is close to the performance of the envelope values. It is possible to visually verify the improvement of the proposed CSC approach with respect to a standard configuration (like the SD-DFD) checking the sample images in Fig. 8. Here we report two views of the scene obtained from the point cloud models associated to the video and depth signals reconstructed at the encoder. Fig. 8(a) shows the image related to the SD-DFD configuration, while Fig. 8(b) shows the same image obtained via the CSC coder. It is possible to notice that distortion in

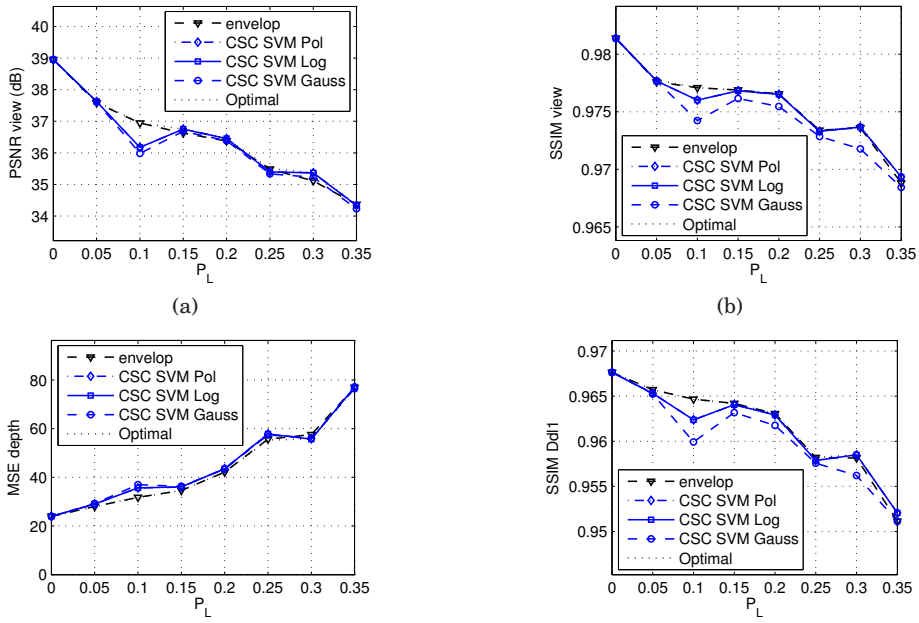


Fig. 6. Average PSNR, SSIM, MSE, SSIM Ddl1 values vs. P_L of the signals V and D of the sequence `breakdancers` obtained via different optimization algorithms. The plot labelled "envelop" reports the value of the metric E_M, while the plot labelled "optimal" reports the value obtained via an optimal off-line configuration. The subfigures show a) PSNR b) SSIM for V, c) MSE for D, and d) SSIM Ddl1.

the geometry signal D_b produces some artifacts for the SD-DFD configuration (tiles detaching from the background), while the image related to the CSC coder proves to have a better quality.

Final graphs (see Fig. 9) show the performance of the approach for the sequences `umbrella` and `door`, which have been acquired using a standard camera with a lateral ToF sensors and a Microsoft Xbox Kinect sensor, respectively. In these cases the amount of noise in the acquired signal (both depth and texture) is high and this significantly affects the error resilience and the compression ratio. Noise level depends on the IR radiation that is acquired by the sensors from the environmental light and on the approximations adopted in the depth estimation. It is worth noticing that the characteristics of noise for the two sensors are extremely different. In the case of ToF camera, noise is derived from external illumination-related IR radiation, while in the case of MS Xbox Kinect sensor noise is also correlated with the projected IR pattern for depth estimation. Moreover, ToF cameras signal processing chipset permits refining depth estimation and reducing the noise level with respect to Kinect sensor. Experimental results show that the proposed CSC scheme performs well also in these cases reducing the average distortion value (measured with MSE) of depth for both sequences despite the fact that estimation noise reduces the gain in the rate-distortion performance.

As for the computational complexity of the approach, Table I reports the relative coding time variation δT of different solutions with respect to the configuration SD-FEC. Results have been obtained using a PC with Intel Dual Core CPU 6400 @ 2.13 GHz and 2 GB RAM. The complexity required by segmentation is included in a separate row, since it is possible to omit the computational load of seg-

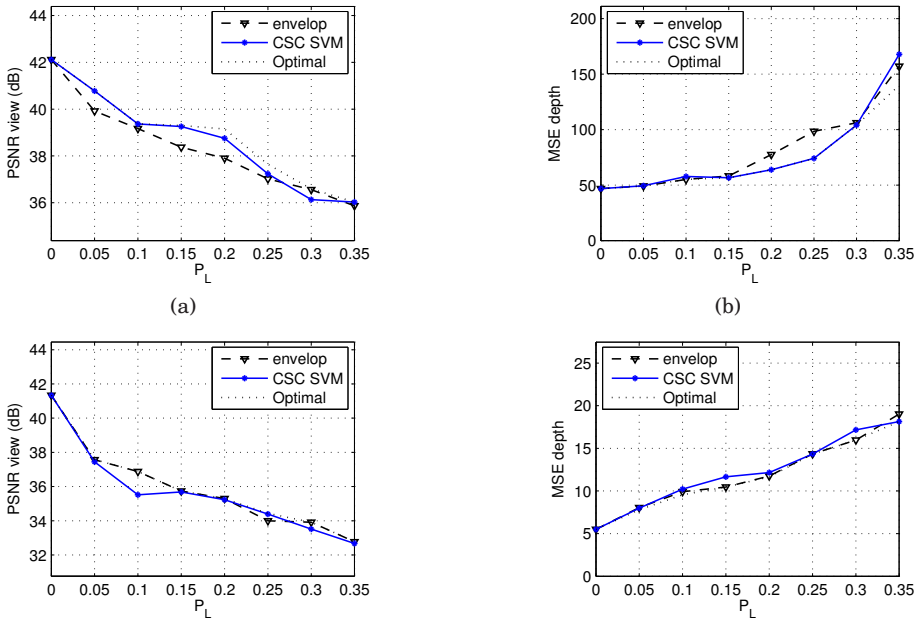


Fig. 7. Average PSNR and MSE vs. P_L of the signals V and D of different sequences. The plot "envelop" reports E_M, while the plot "optimal" reports the optimal off-line optimal. a) PSNR of V and b) MSE of D for `ballet` c) PSNR of V d) MSE of D for `orbi`.

Fig. 8. Frame 4 for the sequence `ballet` from point cloud models created with the received geometry and texture signals. The loss probability of the channels is $P_L = 0.15$. a) SD-DFD b) CSC coder.

mentation whenever object masks are already available at the encoder.[4] In addition, it is possible to mitigate the impact of segmentation on the final performance by adopting a less complex segmentation routine that requires a lower amount of operations. Experimental data show a significant saving in the coding time obtained by both reducing the search window size and the number of macroblock partitioning. The first permits limiting the number of candidate predictor blocks in the motion estimation process, the latter decreases the computational complexity required in the rate-distortion optimization. The overall complexity is reduced of about one third with respect to the complexity of the configura-

[4]This assumption relates to the fact that many depth estimation algorithms rely on the extraction of object silhouettes from the scene; therefore, object masks could be already available at the encoder.

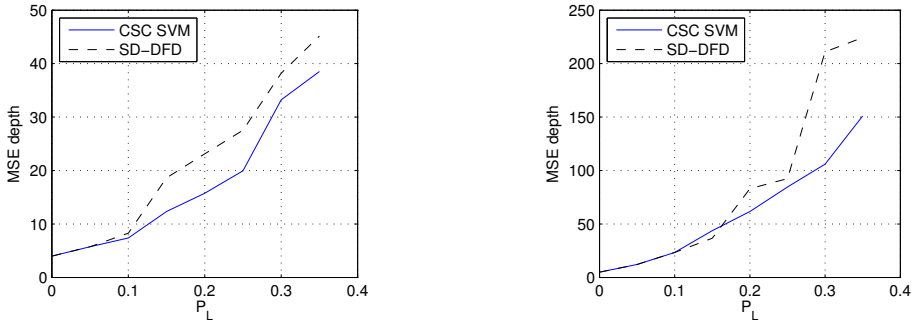


Fig. 9. Average MSE vs. P_L for different configurations coding the signal D of sequences `door` (obtained with ToF camera) and `umbrella` (obtained with Microsoft Kinect). Graphs compare SD-DFD with the CSC approach. a) `door` b) `umbrella`.

Table I. Coding time increment of different configurations for `breakdancers` and `ballet`.

Sequence	**Conf.**	δT **(%)**	**Sequence**	**Conf.**	δT **(%)**
`breakdancers`	SD-DFD	0.00	`ballet`	SD-DFD	0.00
	MD-DFD	$+0.52$		MD-DFD	-0.31
	SD-WZ	$+0.99$		SD-WZ	$+0.35$
	MD-WZ	$+1.57$		MD-WZ	$+0.37$
	SVM Log. w/out seg.	-36.04		SVM Log. w/out seg.	-36.14
	SVM Log. with seg.	-31.04		SVM Log. with seg.	-31.14

tion SD-DFD (corresponding to the complexity of the standard H.264/AVC encoder). It is also possible to notice that the overall complexity is about 31% lower including the segmentation routine too (see Table I). This reduction is possible since most of the computational complexity is related to motion search operations and rate-distortion analysis, and therefore, the reduction of the search window size, together with the reduction of the possible partitioning modes, for large regions of each frame permits compressing the captured signals with a lower amount of calculation. As a matter of fact, it is possible to conclude that the object-oriented solution proposed in the paper is also competitive from the computational point-of-view since it permits adapting easily the hardware resources to the characteristics of the processed signal.

Future works will be focused on the adoption of scalable CSC coding systems where the generated data stream can be transmitted to an heterogeneous set of users varying the quality of the reconstructed 3D scene according to the displaying device or the level of QoE for which the user is paying for. This possibility has proved to be extremely effective whenever applied to depth maps [Schierl et al. 2007]. As an example, stereoscopic displays do not require an accurately-reconstructed depth information with respect to autostereoscopic screens, where the number of views to be rendered is significantly larger. This scalable extension will be easily implemented since the adopted DVC strategy operates on a generic residual signals (which can be obtained with temporal and spatial prediction or refining the quantization operated on the signals.

A second development issue concerns testing the proposed approach on a real network where feedback information about the state of the network can be available at different frequencies and with different accuracies. These first tests were performed on a Gilbert-Elliot model since it permits an adequate level of generality for the reported results.

6. CONCLUSIONS

The paper has presented a flexible and reconfigurable architecture for robust video transmission of 3D video signals. The proposed scheme 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 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 an SVM-based cognitive 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.

Future work will be devoted to improve the coding results by testing the proposed coding scheme on a real network and combining the proposed cognitive approach with augmented reality systems.

REFERENCES

AARON, A., ZHANG, R., AND GIROD, B. 2002. Wyner-ziv coding for motion video. In *Proceedings of Asilomar Conference on Signals, Systems and Computers 2002*. Vol. 1. Pacific Grove, CA, USA, 240 – 244.

ADIKARI, A. B. B., FERNANDO, W. A. C., WEERAKKODY, W. A. R. J., KONDOZ, A., MARTÍNEZ, J. L., AND CUENCA, P. 2008. DVC Based Stereoscopic Video Transmission in a Mobile Communication System. In *Proc. of IEEE FMN 2008 (co-located with NGMAST2008)*. Cardiff, Wales, GB, 439 – 443.

AKSAY, A., BILEN, C., KURUTEPE, E., OZCELEBI, T., AKAR, G. B., CIVANLAR, R., AND TEKALP, M. 2006. Temporal and spatial scaling for stereoscopic video compression. In *Proc. of EUSIPCO 2006*. Firenze, Italy.

ALREGIB, G., ALTUNBASAK, Y., AND ROSSIGNAC, J. 2005. Error-resilient transmission of 3d models. *ACM Trans. Graph. 24,* 2, 182–208.

ARTIGAS, X., ASCENSO, J., DALAI, M., KLOMP, S., KUBASOV, D., AND OUARET, M. 2007. The DISCOVER codec: Architecture, Techniques and Evaluation. In *Proc. of PCS 2007*. Lisbon, Portugal.

BALTER, R., GIOIA, P., AND MORIN, L. 2006. Scalable and efficient coding using 3d modeling. *IEEE Trans. Multimedia 8,* 6, 1147–1155.

BENOIT, A., CALLET, P. L., CAMPISI, P., AND COUSSEAU, R. 2008. Quality assessment of stereoscopic images. *EURASIP Journal on Image and Video Processing 2008*. ID 659024.

BOSER, B. E., GUYON, I. M., AND VAPNIK, V. N. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proc. of the* 5^{th} *Annual ACM Workshop on COLT 1992*. Pittsburgh, PA, USA, 144 – 152.

BOUGHORBEL, S., TAREL, J. P., AND BOUJEMAA, N. 2005. Conditionally Positive Definite Kernels for SVM Based Image Recognition. In *Proc. of ICME 2005*. Vol. 0. Amsterdam, The Netherlands, 113 – 116.

BREMOND, R., PETIT, J., AND TAREL, J.-P. 2010. Saliency maps of high dynamic range images. In *Media Retargeting Workshop in conjunction with ECCV'10*. Heraklion, Crete, Greece. http://perso.lcpc.fr/tarel.jean-philippe/publis/weccv10.html.

CRAVE, O., GUILLEMOT, C., PESQUET-POPESCU, B., AND TILLIER, C. 2008. Multiple Description Source Coding with Side Information. In *Proc. of EUSIPCO 2008*. Lausanne, Switzerland.

FAN, Y., WANG, J., SUN, J., WANG, P., AND YU, S. 2003. A Novel Multiple Description Video Codec Based on Slepian-Wolf Coding. In *Proc. of DCC 2008*. Snowbird, UT, USA, 515.

FÄRBER, N., STUHLMULLER, K., AND GIROD, B. 1999. Analysis of error propagation in hybrid video coding with application to error resilience. In *Proc. of International Conference on Image Processing, ICIP 1999*. Thessaloniki, Greece, 550–554.

FEHN, C. 2004. 3D-TV Using Depth-Image-Based Rendering (DIBR). In *Proc. of PCS 2004*. San Francisco, CA, USA.

FELZENSZWALB, P. F. AND HUTTENLOCHER, D. P. 2004. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision 59,* 2, 167 – 181.

FRAUNHOFER HHI. 2011. Repository of FHG HHI on 3DTV NoE. https://www.3dtv-research.org/3dav/3DAV_Demos/FHG_HHI/Sequences/.

GOYAL, V. K. 2001. Multiple Description Coding: Compression Meets The Network. *IEEE Signal Process. Mag. 8,* 5, 74–93.

HAYKIN, S. 2005. Cognitive Radio: Brain-Empowered Wireless Communications. *IEEE J. Sel. Areas Commun. 23,* 2, 201 – 220. (Invited).

ISO/IEC JTC1. 2001. Coding of Audio-Visual Objects - Part 2: Visual. ISO/IEC 14 496-2 (MPEG-4 Visual version 1), Apr. 1999; Amendment 1 (version 2), Feb. 2000; Amendment 4 (streaming profile), Jan. 2001.

ITU-T. 1995. Video Coding for Low Bitrate Communications, Version 1. ITU-T Recommendation H.263.

ITU-T AND ISO/IEC JTC1. 1994. Generic Coding of Moving Pictures and Associated Audio Information-Part 2: Video. ITU-T Recommendation H.262-ISO/IEC 13 818-2 (MPEG-2).

JAGMOHAN, A. AND AHUJA, N. 2003. Wyner-Ziv Encoded Predictive Multiple Descriptions. In *Proc. of DCC 2003*. Snowbird, UT, USA, 213 – 222.

KARIM, H. A., HEWAGE, C. T. E. R., YU, A. C., WORRAL, S., DOGAN, S., AND KONDOZ, A. M. 2007. Scalable Multiple Description 3D Video Coding Based on Even and Odd Frame. In *Proc. of PCS 2007*. Lisbon, Portugal.

KATSAGGELOS, A. K., EISENBERG, Y., ZHAI, F., BERRY, R., AND PAPPAS, T. N. 2005. Advances in Efficient Resource Allocation for Packet-Based Real-Time Video Transmission. *Proc. IEEE 93,* 1, 135–147.

LIAO, J. AND VILLASENOR, J. 2000. Adaptive intra block update for robust transmission of H.263. *IEEE Trans. Circuits Syst. Video Technol. 10,* 1, 30–35.

MICROSOFT RESEARCH. 2011. MSR 3D Video download. http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload.

MILANI, S. 2010. Simone Milani's Home Page: Download. http://www.dei.unipd.it/∼sim1mil/downloads.html.

MILANI, S. 2011. Simone Milani's Home Page: Download. http://www.dei.unipd.it/∼sim1mil/publications.html♯CSCDemo.

MILANI, S. AND CALVAGNO, G. 2009. A Distributed Video Coding Approach for Multiple Description Video Transmission over Lossy Channels. In *Proc. of EUSIPCO 2009*. Glasgow, UK, 1824–1828.

MILANI, S. AND CALVAGNO, G. 2010a. A Cognitive Approach for Effective Coding and Transmission of 3D Video. In *Proc. ACM Multimedia 2010*. Florence, Italy.

MILANI, S. AND CALVAGNO, G. 2010b. A Cognitive Source Coding Scheme for Multiple Description 3DTV Transmission. In *Proc. of the 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2010)*. Desenzano del Garda, Brescia, Italy.

MILANI, S. AND CALVAGNO, G. 2010c. Multiple Description Distributed Video Coding Using Redundant Slices and Lossy Syndromes. *IEEE Signal Process. Lett. 17,* 1, 51 – 54.

MOBILE3DTV PROJECT. 2011. 3D Video database. http://sp.cs.tut.fi/mobile3dtv/stereo-video/.

NORKIN, A., AKSAY, A., BILEN, C., AKAR, G. B., GOTCHEV, A., AND ASTOLA, J. 2006. Schemes for Multiple Description Coding of Stereoscopic 3D. *Lecture Notes in Computer Science 4105/2006*, 730 – 737.

PURI, R. AND RAMCHANDRAN, K. 2002. PRISM: A new robust video coding architecture based on distributed compression principles. In *Proc. of the Allerton Conference 2002*. Allerton, IL, USA, 402–408.

REUSENS, E., CASTAGNO, R., BUHAN, C. L., PIRON, L., EBRAHIMI, T., AND KUNT, M. 1996. Dynamic video coding - an overview. In *Proc. of IEEE ICIP 1996*. Vol. III. Lausanne, Switzerland, 377–380.

ROSENBERG, J. AND SCHULZRINNE, H. 1999. An RTP Payload Format for Generic Forward Error Correction (RFC2733). *Internet Draft, Network Working Group*.

S. GOEL, Y. I. AND BAYOUMI, M. 2005. Adaptive search window size algorithm for fast motion estimation in h.264/avc standard. In *Proc. of MWSCAS 2005*. Cincinnati, OH, USA, 1557 – 1560.

SAXENA, A., SUN, M., AND NG, A. Y. 2009. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Trans. Pattern Anal. Mach. Intell. 30,* 5, 824 – 840.

SCHIERL, T., STOCKHAMMER, T., AND WIEGAND, T. 2007. Compression of multiple depth maps for DIBR. *IEEE Trans. Circuits Syst. Video Technol. 17,* 9, 1204 – 1217.

SCHULZRINNE, H., CASNER, S., FREDERICK, R., AND JACOBSON, V. 1996. RTP: A Transport Protocol for Real-Time Applications (RFC1889). In *Network Working Group*.

SHI, S., JEON, W., NAHRSTED, K., AND CAMPBELL, R. 2009. M-TEEVE: Real-Time 3D Video Interaction and Broadcasting Framework for Mobile Devices. In *Proc. of IMMERSCOM '09*. Berkeley, CA, USA.

WANG, A., ZHAO, Y., AND BAI, H. 2009. Robust multiple description distributed video coding using optimized zero-padding. *Sci China Ser F-Inf Sci 52,* 2, 206 – 214.

WANG, J., WU, X., YU, S., AND SUN, J. 2006. Multiple Descriptions in the Wyner-Ziv Setting. In *Proc. of IEEE ISIT 2006*. Seattle, WA, USA, 1584 – 1588.

WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process. 13,* 4, 600 – 612.

WIEGAND, T. 2004. Version 3 of H.264/AVC. In 12^{th} *JVT Meeting*. Redmond, WA, USA.

WU, M., VETRO, A., AND CHEN, C. W. 2004. Multiple Description Image Coding with Distributed Source Coding and Side Information. In *Proceedings of SPIE: Multimedia Systems and Applications VII*. Vol. 5600. Philadelphia, PA, USA, 120 – 127.

YEO, C. AND RAMCHANDRAN, K. 2007. Robust distributed multiview video compression for wireless camera networks. In *Proc. of VCIP 2007*. Vol. 6508. San Jose, CA, USA, 65080P–1 – 65080P–9.