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

Simone Milani
Dept. of Information Engineering
University of Padova
via Gradenigo 6/B, 35131 Padova - Italy
simone.milani@dei.unipd.it

Giancarlo Calvagno
Dept. of Information Engineering
University of Padova
via Gradenigo 6/B, 35131 Padova - Italy
calvagno@dei.unipd.it

ABSTRACT

Reliable delivery of 3D video contents to a wide set of users is expected to be the next big revolution in multimedia applications provided that it is possible to grant a certain level of Quality-of-Experience (QoE) to the end user. During the last years, several cross-layer solutions have proved to be extremely effective in tuning the transmission parameters at the different layers of the protocol stack and in maximizing the perceptual quality of the reconstructed 3D scene. Among these, Cognitive Source Coding (CSC) schemes (defined in analogy with Cognitive Radio systems) make possible to improve the quality of the 3D QoE at the receiver by adapting the source coding strategy according to the state of the transmission channel and to the characteristics of the coded signal. This knowledge also permits an optimization of the computational complexity required at the encoder. The paper presents a CSC architecture that analyzes the 3D scene, identifies the different elements, and chooses the most appropriate coding strategy via a classification of the features of each element based on Support Vector Machine theory. Experimental results show that the proposed approach permits improving the quality of the received 3D signal with respect to traditional cross-layer techniques and reducing the computational complexity of coding operation.

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.

MM'10, October 25–29, 2010, Firenze, Italy.

Keywords

Cross-layer optimization, source coding, joint source-channel coding, 3D video, cognitive source coding.

1. INTRODUCTION

The delivery of 3D video contents to a wide set of users is expected to be the next big revolution in multimedia applications. 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 [1]. In addition, the appearing of consumer 3D displays on the market enables a wider set of people with the possibility of viewing 3D video signals at different locations. To this purpose, the employment of wireless networks permits connecting and distributing 3D multimedia contents to a wider set of terminals with respect to traditional wired communications. This fact is mainly due to the flexible topology of wireless systems (nodes can enter and leave the network more easily) and to the possibility of mobility offered to the different terminals. However, appropriate solutions and strategies need to be adopted to enable a reliable and satisfying 3D video transmission [2]. In fact, the intense bandwidth requirements and high sensitivity to packet losses of multimedia signals reveal some of the crucial limits of wireless networks in delivering audio/visual contents. The time-varying nature of radio channels, the presence of external noise, and the competition among different users in accessing the network reduce the transmission performance producing a dramatic decrement in the Quality-of-Experience (QoE) of the end user. As a matter of fact, a reliable and effective transmission of 3D video signals proves to be an essential and interesting research issue for the development and the diffusion of 3D video applications.

Novel protocols and transmission strategies have been designed in order to minimize the loss of crucial data and limit 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 [3]. In this way, it is possible to combine different protection and retransmission strategies to satisfy the requirements related to the specific application. Among CL strategies for transmission of 3D signals it is possible to mention the solution proposed by Alregib *et al.* [4], which adopts a scalable compression for 3D models and applies an Unequal Error Protection (UEP) on the different coding layers in order to decrease the loss probability as the signifi-

cance of the data in the decoding process increases. Another approach has been proposed by Balter *et al.* in [5], where the compression of the different signals is organized in order to maximize the quality of the final 3D scene rendered by the end user.

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) schemes in analogy to Cognitive Radio (CR) schemes [6] adopted for radio transmissions. As defined by Haykin in [7], "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 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.

In addition to a partial knowledge of the channel states and the transmission conditions, the CSC coder can also be aware of the characteristics of the signals that have to be transmitted. In fact, error concealment algorithms perform quite differently according to the characteristics of the transmitted video signals. Static objects within a recorded video scene are easier to conceal with respect to fast-moving ones. As a matter of fact, CSC can vary the adopted source coding techniques in order to suit the characteristics of the different elements. 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.

Both CSC and CR must provide 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.

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.

Like for Cognitive Radio systems, reconfigurability is one of the key elements that permit satisfying these requirements. The possibility of orchestrating the different functional blocks of the coding architecture permits improving the effectiveness of the transmission in terms of perceptual quality experienced by the end user. In this paper we present a reconfigurable coding scheme for 3DTV signals that 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. The adopted solutions can be enabled or disabled according to the characteristics of the coded objects within the scene and according to the states of the channel. Moreover, the computational load of coding operations is varied according to the modellization complexity of each element in the scene.

The paper present a new CSC coder, which is applied to a video+depth 3D sequence[1] optimizing both the quality of the reconstructed view and the accuracy of the received depth map. One of the major innovation introduced with respect to previous architectures like those in [3, 4, 5, 2] relies on the fact that the presented approach is able to change the type of source coding according to the characteristics of the channel. Moreover, the proposed approach differentiates the compression choices for the different objects within the scene in order to tailor the robustness of coding to their characteristics. 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 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 by adopting an Support Vector Machine (SVM)-based optimization of the different coding modes. Conclusions are drawn in Section 5.

2. THE PROPOSED CSC ARCHITECTURE

According to the premises about Cognitive Source Coding presented in the previous section, effective coding and transmission of 3D video signals imply the design of a flexible scheme that permits implementing different coding schemes and requires limited hardware resources. As a matter of fact, the development of a CSC architecture considers video coding solutions that present common features and building blocks so that many parts can be reused in implementing the different coding schemes. In our approach, we have included a Multiple Description Scheme and a Wyner-Ziv coder within a traditional H.264/AVC codec such that it is possible to switch from one coding solution to another by simply rewiring the connections between different 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 separate the background and the static objects (grouped in the regions V_b and D_b) from the

[1]This 3D video format is also called Depth Image Based Rendering (DIBR) [8].

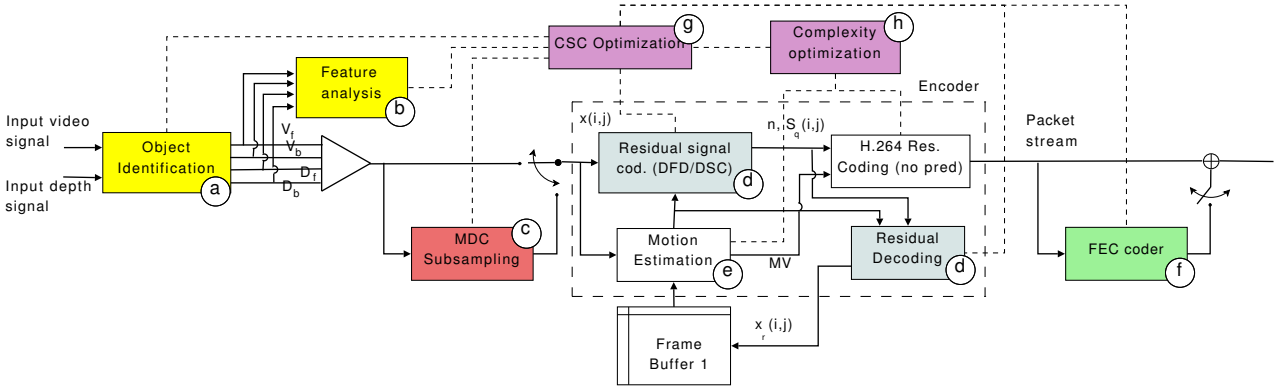


Figure 1: Block diagram for the encoder.

foreground and the moving objects (included in the regions V_f and D_f). Each subregion is then analyzed, and a set of features is extracted (block ⓑ) to describe the characteristics of the signal in terms of error resiliency. 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.

In case the MDC option is enabled (block ⓒ), the current sequence of objects is processed by a polyphase sampler that separates odd and even lines of pixels into two subsequences ("descriptions"). The output frames are then predicted by a Motion Estimation unit (block ⓔ), and the prediction error can be compensated either coding the residual signal like in a traditional video coder (see [9]) or employing a Wyner-Ziv coder that permits a more robust characterization of the signal (see the blocks ⓓ). The processed signals are then converted into a binary stream, which is encapsulated in a set of RTP packets.

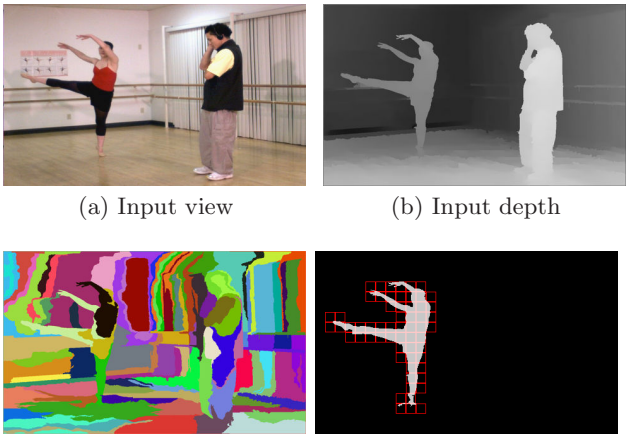
In the end, it is possible to include some redundant packets generated via FEC channel coder (block ⓕ). This additional information permits recovering the lost packets whenever their number is lower than a given threshold.

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.

2.1 Segmentation of the input frames into subregions

The first unit of the proposed CSC scheme partitions the input video and depth frames into two subregions generating the signals V_f, V_b, D_f, and D_b. The segmentation unit analyzes the captured scene distinguishing regions in the foreground with significant amount of motion (associated to the pixel set $\mathcal{F}$) from slowly-moving and background elements (represented by the pixels in the set $\mathcal{B}$). 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.

The first class $\mathcal{F}$ includes fast-moving objects close to the point of view, which have a stronger impact on the final visual quality and can not be easily estimated by the error concealment unit whenever they get lost. The second class



(a) Input view (b) Input depth

(c) Regions R_k (d) Resulting front ROI

Figure 2: Video signals in the object detection algorithm.

$\mathcal{B}$ 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 approximated using the previously-decoded data with a limited channel distortion. The classification routine partitions the input frame into N_R small regions R_k (also called "superpixels" [10]) via an oversegmentation of the depth signal [11] (see Fig. 2(c) as an example). Superpixels are then merged together according to their spatio-temporal characteristics and the values of the included texture and geometry pixels. More precisely, for each superpixel $R_k \subset \mathbb{Z}^2$, $k = 0, \ldots, N_R - 1$, the object identification unit ⓐ computes the activity

$$act_k = \sum_{(x,y)\in R_k} \|V_t(x,y) - V_{t-1}(x,y)\| \qquad (1)$$

where $V_t(x,y)$ is the pixel at position (x,y) of the frame at time t from the texture signal V. In case the parameter act_k is lower than a given threshold T_1, the superpixel R_k is assigned to the set $\mathcal{B}$. In our implementation, T_1 is set 4. For the remaining R_k regions, the algorithm computes the average depth pixel values

$$\overline{d}_k = \frac{1}{|R_k|} \sum_{(x,y)\in R_k} D_t(x,y) \qquad (2)$$

where $|R_k|$ is the cardinality of the set R_k. The average

values d_k are then grouped together into 10 different classes using a clustering algorithm. As a result, it is possible to merge together the regions of the same cluster creating the regions R'_m. According to the average depth values of pixels within R'_m, the object identification algorithm assigns the foreground regions R'_m to the pixel set $\mathcal{F}$ and the background regions R'_m to the pixel set $\mathcal{B}$.

The two pixel masks $\mathcal{F}$ and $\mathcal{B}$ are then used to distinguish two image regions in each video sequence and characterize them differently. It is possible to code the masks using an object-based video coding architecture. However, 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 $\mathcal{F}$ (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). 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 strategies according to the characteristics of the processed signal and of the transmission channel.

2.2 Characterization of the signal via multiple description

After splitting the input depth and texture signals into four subsequences, the CSC architecture decides whether to code the subsequences as they are or to divide them into multiple correlated descriptions that are coded separately (see the functional unit ⓒ in Fig. 1). In this way, each subsignal can be decoded at the receiver independently from whether the other descriptions have been correctly received or not. Moreover, the existing correlation permits approximating the lost descriptions from the available ones (see [12]), and this leads to a quality-scalable characterization of the input signal depending on the number of received descriptions.

In our implementation, we adopted an optional MDC scheme based on a vertical polyphase subsampling of the pixel rows into two descriptions. Whenever the MDC option is enabled, the subsequences V_f, V_b, D_f, and D_b are converted in the subsequences V_f^m, V_b^m, D_f^m, and D_b^m, $m = 1, 2$, which contains 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.

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 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 FREXt residual coding strategy [9]. 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 $\mathbf{x}'_p$ at the decoder differs from $\mathbf{x}_p$ since an additional channel distortion can be present in the sequence such that $\mathbf{x}'_p = \mathbf{x}_p + \mathbf{e}_c$. This additional component is related to the fact that the signal has been concealed by the unit ⓔ after some packet losses, and as a matter of fact, the distortion propagates throughout the sequence and degrades significantly the quality of the reconstructed sequence [13].

It is possible to mitigate this effect by choosing a more robust characterization of the residual signal. In our solution we adopted the approach proposed in [14], where the 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 a syndrome $s(i,j)$ of $n_{\max}$ bits via the same procedure described in [14] and in [15]. More precisely, $s(i,j)$ corresponds to the $n_{\max}$ least significant bits of $x(i,j)$, where $n_{\max}$ has been computed from the correlation between the pixels of block $\mathbf{x}$ and the pixels of block $\mathbf{x}_p$ (see [14] for more details). In our implementation, the correlation is measured via the prediction error $\mathbf{d}$ leading to the $n_{\max}$ value

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

where

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

The difference $|d(i,j)|$ is the prediction error $|x(i,j) - x_p(i,j)|$, while the parameter δ is a threshold value depending on the Quantization Parameter (QP) chosen for the current block (in our setting, we have set $\delta = \Delta/12$

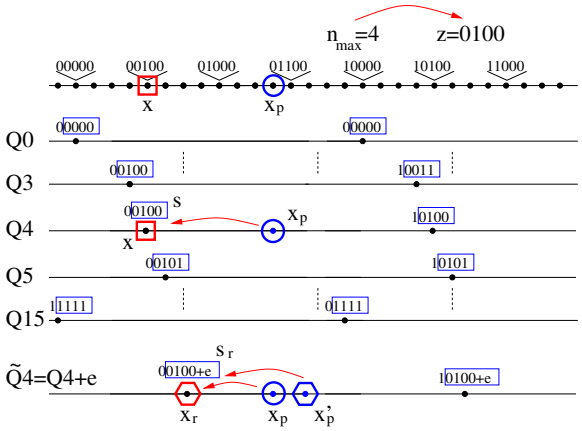


Figure 3: Computation of the syndrome $s(i,j)$ in the WZ coder and its decoding. For the sake of clarity, indexes (i,j) have been omitted.

where 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 [16] 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 $x(i,j)$. It is possible to use a different $n(i,j)$ for each pixel $x(i,j)$. However, this implies that all the $n(i,j)$ have to be specified in the bit stream leading to a significant increase of the amount of information that has to be transmitted. As a matter of fact, only the maximum value $n_{\max}$ is specified since a correct reconstruction of $x(i,j)$ is allowed whenever the chosen $n_{\max}$ value is greater than $n(i,j)$.

As a result, the WZ coder produce a block of syndromes $\mathbf{s}$ from the original pixel block $\mathbf{x}$, which 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 $Q_{s_r(i,j)}$ such that the reconstruction levels can be expressed as $s_r(i,j) + k\ 2^{n_{\max}}$, $k \in \mathbb{Z}$ (see Fig. 3). 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 [14]). As Fig. 3 depicts, the quantization of $\mathbf{x}'_p$ via the characteristics $Q_{s_r(i,j)}$ leads to the same reconstructed value $x_r(i,j)$.

2.4 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 [17], 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.

3. OPTIMIZATION

The previous section has presented the different functional units of the proposed CSC architecture. The designed scheme permits implementing several coding schemes by rewiring the connections between the different units. For the sake of conciseness, we identify here some of the possible configurations which have proved to be the most competitive with respect to other solutions under different channel states.

- **SD-DFD**: The input signal is coded into a single description (with the MDC option disabled), and the prediction residuals are coded with the DFD configuration (standard H.264/AVC coding). The output packets are protected by additional FEC packets generated using block ⓕ.
- **MD-DFD**: The input signal is split into two descriptions (with the MDC option enabled), whose prediction residual is coded with the DFD configuration. No additional FEC packets are included in the stream.
- **SD-WZ**: The input signal is coded into a single description like in the SD-DFD configuration, but the prediction residual is coded with the WZ configuration. Additional FEC packets increase the robustness of the final packet stream.
- **MD-WZ**: The input signal is split into two descriptions like in the MD-DFD configuration, whose prediction residual is coded with the WZ configuration. The FEC coder is disabled.

Previous works have shown that the effectiveness of each configuration varies according to the channel characteristics and the feature of the video signal (see [18, 15]). In the analyzed case, the effectiveness depends on the signal type (texture or geometry) and on the characteristics of the different objects in the scene. This variability can also be applied to the hardware resources that are involved in the coding process. Background and slowly-moving elements do not need a strong computational effort in motion estimation with respect to foreground quickly-moving objects that require a wider motion-search window and a more complex block partitioning structure. As a matter of fact, an accurate classification of the elements in the scene is crucial. 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

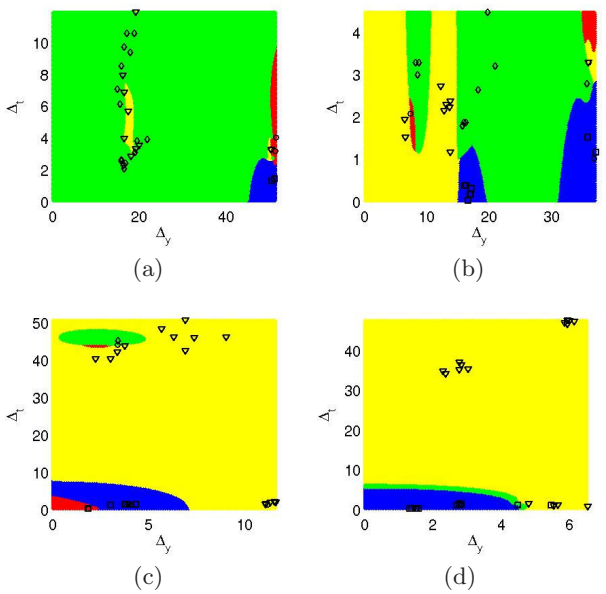


Figure 4: SVM classifiers with logarithmic kernels for different subsequences and $P_L = 0.2$. Blue points are coded with SD-DFD, yellow ones with MD-DFD, red ones with SD-WZ, green ones with MD-WZ. a) V_f b) V_b c) D_f d) D_b.

identifies the characteristics of the signal in terms of error resiliency and permits choosing the best coding configuration. This classification is also employed by a complexity optimization unit that appropriately modifies some of the coding parameters closely related to the final computational load.

In our approach, subsequences V_f, V_b, D_f, and D_b are analyzed via the feature extraction unit ⓑ, which computes the array of average values

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

where $\Delta_y(\cdot)$ is the 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 measure the temporal correlation affecting the error concealment and the efficiency of WZ approach with respect to its DFD counterpart). Since the proposed CSC strategy aims at tailoring the coding choices on the characteristics of the 3D video sequence, these values are averaged for all the frames in a Group-Of-Picture (GOP) and computed for each subsequence 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 array $\mathbf{v}_S$ is then processed by a non-linear Support Vector Machine (SVM) classifier [19], which partitions the parameter space covered by $\mathbf{v}_S$ into 4 regions associated to the 4 configurations in the set $\mathcal{C} = \{$SD-FEC, MD-FEC, SD-WZ, MD-WZ$\}$ (see Fig. 4). 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 $\mathbf{v}_S$, the

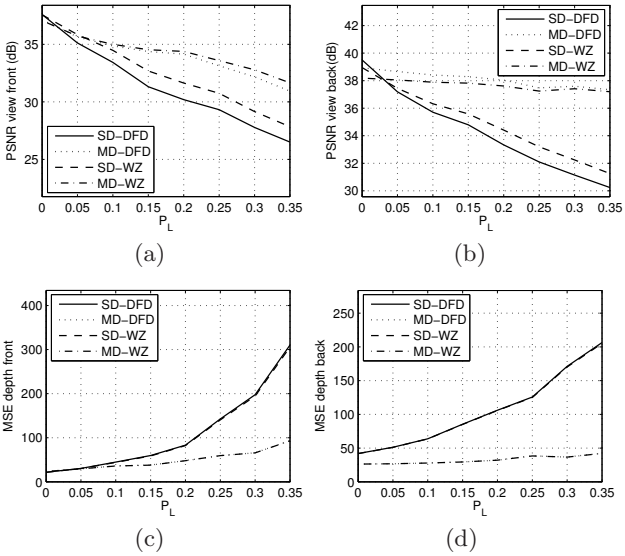


Figure 5: Average PSNR and MSE values vs. P_L of the subsequences V_f, V_b, D_f, and D_b of the sequence `breakdancers` for the CSC configurations SD-DFD, MD-DFD, SD-WZ, and MD-WZ. a) PSNR for V_f b) PSNR for V_b c) MSE for D_f d) MSE for D_b.

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

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

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

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

for each subsequence S. The employed SVM classifier has been designed using different non-linear kernel functions [19, 20]

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

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

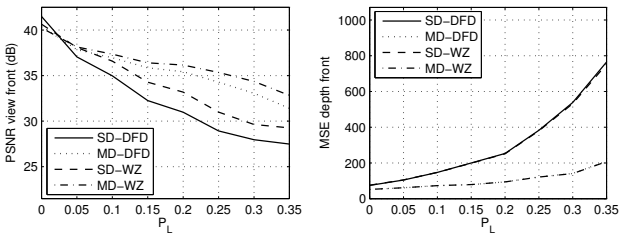
The classification routine partitions the possible loss percentage values into 5 intervals, and for each range of values a different classifier is computed in order to adapt the choice of the best mode to the state of the channel. As an example, Figure 4 depicts the SVM classifier with Logarithmic kernel for different subsignals at $P_L = 0.2$. 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). The RTCP information is periodically available from the network; according to the draft specification [21], RTCP information should not be transmitted 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 estimate the parameter P_L and, according to the obtained value, selects the most appropriate SVM classifier among the 5 possible choices. Until a new RTCP packet is received, the selected classifier will be used in identifying the most appropriate coding choice.

At the beginning of each GOP, the array $\mathbf{v}_S$ is computed by the CSC optimization unit in order to characterize the features of each subsignal S within the current GOP. The array v_S is classified using the selected SVM mapper (one for each subsignal) identifying the most appropriate coding choice for the current GOP. These operation are iterated at every GOP.

As for the computational complexity, the unit ⓗ 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 [22] as an example). 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 and reduce the computational complexity. 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, search window size is not reduced in this case both because the foreground regions present a stronger amount of motion and because they are closer to the capturing camera. 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 4. As a matter of fact, the presented solution can be effectively employed for Live 3D video streaming applications and 3D video conferencing since it permits a low-complexity robust compression of video+depth signals and an easy implementation on mobile device. Moreover, since its structure 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). In the following, experimental results will show how this classification permits increasing



(a) (b)

Figure 6: Average PSNR and MSE values vs. P_L of the subsequences V_f and D_f of the sequence `ballet` for the CSC configurations SD-DFD, MD-DFD, SD-WZ, and MD-WZ. a) PSNR for V_f b) MSE for D_f.

the visual quality of the reconstructed 3D sequence and reducing the computational complexity with respect to a standard H.264/AVC coder.

4. SIMULATION RESULTS

The proposed solution has been tested simulating the transmission of a wide set of DIBR sequences over lossy channels, which are simulated by independent two-state Gilbert models with average burst length $L_B = 4$ and varying loss probability P_L. The RTP packets related to the subsequences V_f, V_b, D_f, and D_b are transmitted over four independent 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.

Different quality metrics have been recently proposed in literature with the purpose of calculating numerical values correlated with the human perception. Together with the traditional PSNR value, it is possible to compute the Structural Similarity Index Metrics (SSIM) [23] in order to compare the Quality-of-Experience provided to the end user by a video transmission scheme. As for the geometry signals, Mean Square Error (MSE) is widely employed to evaluate the accuracy of the reconstructed depth maps. The quality of the final 3D signal can be evaluated using combined metrics that takes into account both the reconstructed depths and views, like the SSIM Ddl1 metric in [24]. Note that the visual quality of the signals increases as the values of PSNR, SSIM, SSIM Ddl1 increase and as the values of MSE decrease.

The video source coding engine has been derived from the structure of the H.264/AVC coder reusing 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 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 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 [25], the HHI and the Mobile3D [26] repositories, and their formats have been equalized in order to process video sequences with the same resolution. Sequences `breakdancers`, `ballet`, `car`, `horse` have been used in the training of the SVM classifier, while the sequences `interview` and `orbi` has been used as test sequences.

Figure 5 reports the average PSNR and MSE values vs. the loss probability P_L for the sequence `breakdancers`. It is possible to notice that at low loss rates the SD-DFD configuration provides the best performance, while at high loss rates MD-DVC permits obtaining higher average PSNR values (see Figure 5(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. 5 it is possible to infer that the best configuration at $P_L = 0.1$ is SD-WZ for V_f, MD-DFD for V_b, and MD-WZ for D_f and D_f. The most effective CSC configuration is also dependent on the characteristics of the coded sequence. Figure 6 reports the average PSNR values vs. P_L for the sequence `ballet`. It is possible to notice that the configuration MD-WZ proves to be competitive at low loss rates (event with $P_L = 0.1$) for the signal V_f with respect to the other configurations, while the best configuration for V_f from `breakdancers` at $P_L = 0.1$ is SD-WZ.

Final tests were devoted to compare the performance of different SVM classifiers on both training and test sequences. Figure 7 reports the average values of different metrics obtained with different optimization algorithms for the sequence `breakdancers`. The graphs also show an envelop function E_M

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

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. 7(a)). The proposed solution is effectively able to find the best coding mode. Moreover, the metric values for SVM-based algorithms are close to the optimal values, which have been obtained via an off-line optimization of the coding modes in $\mathcal{C}$. The same behavior can be noticed via different metrics (see Fig. 7(b) and Fig. 7(d)) and for the depth signal too (see Fig. 7(c)). 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 8 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

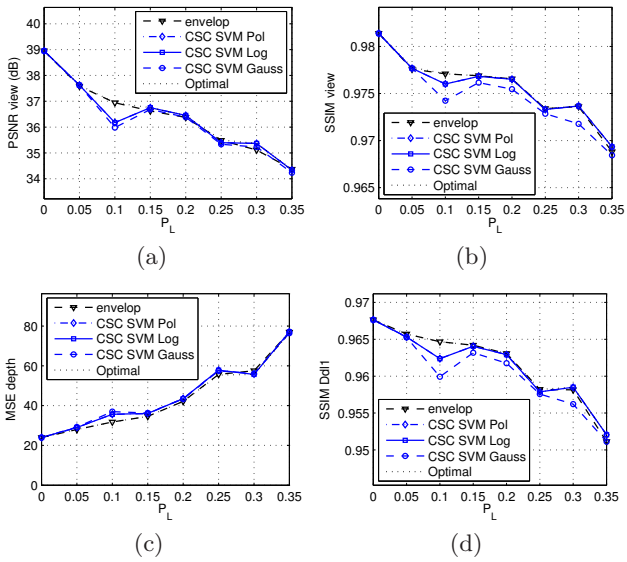


Figure 7: Average PSNR, SSIM, MSE, SSIM Ddl1 values vs. P_L of the signals V and D of the sequence `breakdancers` obtained via different optimization algorithms. The plot labelled "envelop" reports the value of the metric E_M, while the plot labelled "optimal" reports the value obtained via an optimal off-line configuration. The subfigures show a) PSNR b) SSIM for V, c) MSE for D, and d) SSIM Ddl1.

envelop E_M (see Fig. 8(a)). As for the sequence `orbi`, the performance of the SVM classifier is close to the performance of the envelop values.

As for the computational complexity of the approach, Table 1 reports the coding time increment δT of different solutions with respect to the configuration SD-FEC, which has been computed via the equation

$$\delta T = \frac{T_C - T_{SD-DFD}}{T_{SD-DFD}} \times 100 \tag{11}$$

where T_C is the coding time required by the setting C (C can be a fixed setting in $\mathcal{C}$ or the SVM Log. optimization), and T_{SD-DFD} is the coding time for configuration $SD-DFD$. 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 segmentation whenever object masks are already available at the encoder.[2] In addition, it is possible to mitigate the impact on 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

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

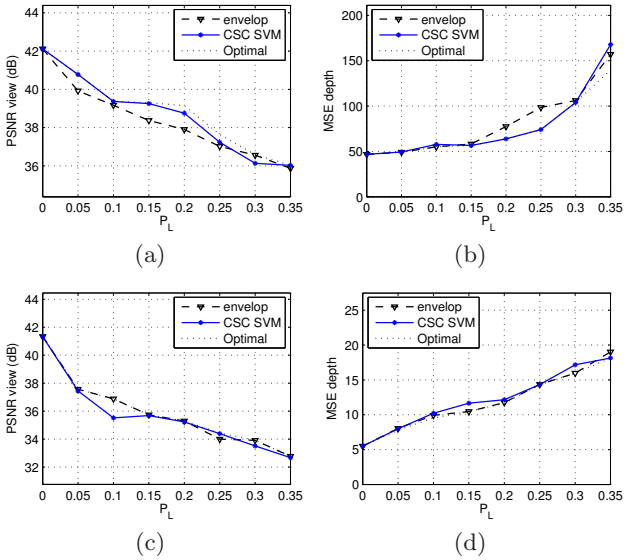


Figure 8: Average PSNR and MSE vs. P_L of the signals V and D of different sequences. The plot "envelop" reports E_M, while the plot "optimal" reports the optimal off-line optimal. a) PSNR of V and b) MSE of D for `ballet` c) PSNR of V d) MSE of D for `orbi`.

third with respect to the complexity of the configuration $SD - DFD$ (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 1). 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.

5. 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. The average PSNR values or the reconstructed sequence are always close or higher than the best values obtained with a fixed configuration. Moreover, computational complexity can be reduced of about 31% with respect to a standard H.264/AVC coder.

Future work will be devoted to improve the coding results by adopting a more effective object-coding strategy and a different classification strategy. The first activity will be focused on coding effectively object masks in order to characterize more accurately the objects in the scene. The second will be focused on improving the quality of the reconstructed sequence by identifying more precisely the best coding configuration for the current GOP.

Table 1: Coding time increment of different configurations for the sequences `breakdancers` and `ballet`.

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

6. REFERENCES

[1] C. Fehn, "A 3D-TV System Based On Video Plus Depth Information," in *Proc. of 37th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Nov. 2003.

[2] S. Shi, W. Jeon, K. Nahrsted, and R. Campbell, "M-TEEVE: Real-Time 3D Video Interaction and Broadcasting Framework for Mobile Devices," in *Proc. of IMMERSCOM '09*, Berkeley, CA, USA, May 25 – 27, 2009.

[3] A. K. Katsaggelos, Y. Eisenberg, F. Zhai, R. Berry, and T. N. Pappas, "Advances in Efficient Resource Allocation for Packet-Based Real-Time Video Transmission," *Proc. IEEE*, vol. 93, no. 1, pp. 135–147, Jan. 2005.

[4] G. Alregib, Y. Altunbasak, and J. Rossignac, "Error-resilient transmission of 3d models," *ACM Trans. Graph.*, vol. 24, no. 2, pp. 182–208, 2005.

[5] R. Balter, P. Gioia, and L. Morin, "Scalable and efficient coding using 3d modeling," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1147–1155, dec 2006.

[6] J. Mitola and G. Q. M. Jr., "Cognitive Radio: Making Software Radios More Personal," *IEEE Personal Commun. Mag.*, vol. 6, no. 6, pp. 13 – 18, Aug. 1999.

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

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

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

[10] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D Scene Structure from a Single Still Image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 824 – 840, May 2009.

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

[12] V. K. Goyal, "Multiple Description Coding: Compression Meets The Network," *IEEE Signal Process. Mag.*, vol. 8, no. 5, pp. 74–93, Sep. 2001.

[13] N. Färber, K. Stuhlmuller, and B. Girod, "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, Oct. 1999, pp. 550–554.

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

[15] ——, "A Distributed Video Coding Approach for Multiple Description Video Transmission over Lossy Channels," in *Proc. of EUSIPCO 2009*, Glasgow, UK, Aug. 24–28, 2009, pp. 1824–1828.

[16] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. of the Allerton Conference 2002*, Allerton, IL, USA, Oct. 2002, pp. 402–408.

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

[18] S. Milani, G. Calvagno, R. Bernardini, and P. Zontone, "Cross-Layer Joint Optimization of FEC Channel Codes and Multiple Description Coding for Video Delivery over IEEE 802.11e Links," in *Proc. of IEEE FMN 2008*, Cardiff, Wales, GB, Sep.17 – 18, 2008, pp. 472 – 478.

[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proc. of the* 5^{th} *Annual ACM Workshop on COLT 1992*, Pittsburgh, PA, USA, Jul. 27 – 29, 1992, pp. 144 – 152.

[20] S. Boughorbel, J. P. Tarel, and N. Boujemaa, "Conditionally Positive Definite Kernels for SVM Based Image Recognition," in *Proc. of ICME 2005*, vol. 0, Amsterdam, The Netherlands, Jul. 6 – 9, 2005, pp. 113 – 116.

[21] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications (RFC1889)," in *Network Working Group*, Jan. 1996.

[22] Y. I. S. Goel and M. Bayoumi, "Adaptive search window size algorithm for fast motion estimation in h.264/avc standard," in *Proc. of MWSCAS 2005*, Cincinnati, OH, USA, Aug. 7 – 10, 2005, pp. 1557 – 1560.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600 – 612, Apr. 2004.

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

[25] MSR 3D Video download. [Online]. Available: http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload

[26] Repository of Mobile3DTV project: 3D Video database. [Online]. Available: http://sp.cs.tut.fi/mobile3dtv/stereo-video/