A DISTRIBUTED CODING-BASED CONTENT-AWARE MULTI-VIEW VIDEO SYSTEM

M. Morbée[1], L. Tessens[1], H. Quang Luong[1], J. Prades-Nebot[2], A. Pižurica[1] and W. Philips[1]

[1] TELIN-IPI-IBBT
Ghent University
Ghent, Belgium
marleen.morbee@telin.ugent.be

[2] GTS-ITEAM
Universidad Politécnica de Valencia
Valencia, Spain
jprades@dcom.upv.es

ABSTRACT

Compared to traditional mono-view systems, stereo or in general multi-view systems provide interesting additional information about a captured scene, which can significantly facilitate content extraction. This property makes them very useful for many emerging applications, such as 3D TV and video surveillance. However, the use of such systems has been limited so far because of the processing time and bandwidth requirements for multi-view data. These major drawbacks can only be relieved by the development of dedicated algorithms. In this paper, we present an efficient, flexible and content-aware coding method for a multi-view video system. The framework consists of a central processor and camera, completed by a flexible number of smart Wyner-Ziv cameras. The latter ones provide a content-aware representation of their viewpoint, thus greatly reducing the amount of data to be sent to the central processor. By employing Distributed Video (DV) coding, i.e. joint decoding of the independently encoded frames of the different cameras, we achieve good coding efficiency without inter-camera communication.

Index Terms— Distributed video coding, Wyner-Ziv coding, multi-view, stereo, content-aware, region of interest.

1. INTRODUCTION

Multi-view video systems have recently become very popular, since they provide interesting additional information about the captured scene, compared to the traditional mono-view systems. For instance, the multi-view setup can facilitate the extraction of 3D information and the interpretation of the scene, which can be useful for many emerging applications, e.g. surveillance, 3D TV or virtual reality. However, the main drawback of these systems is that the total amount of video data increases drastically and hence, the high processing and bandwidth requirements for multi-view data exceed the capabilities of most current systems [1]. A critical component of such a system is the coding engine that compresses the multi-view video data into a rate-distortion efficient and content-aware representation.

In this paper, we propose an efficient, flexible and content-aware coding framework for a multi-view video system. This framework consists of a central processor and a key camera, completed with a flexible number of smart Wyner-Ziv (WZ) cameras that provide a content-aware video representation of their viewpoint. In order to obtain a flexible and energy-efficient system, the video signals of the key and WZ cameras are encoded independently of each other. High coding efficiency is achieved by applying Distributed Video (DV) coding principles.

Content-awareness is a first objective of our system. It is an important issue, since in applications such as surveillance, not the quality of the whole video is important but especially the quality of specific regions in the frames. In this work, we assume that the parts of the frames that correspond to objects in motion contain more critical information than the still regions. In the following, the set of pixels in a frame that move with respect to the background will be called the *Region Of Interest* (ROI) of the frame. Then, the coding algorithms used should guarantee that the quality of the ROI is higher than the quality of the remaining parts. In this paper, the WZ cameras achieve this goal by successive extraction and distributed coding of the ROI.

In distributed coding, correlated sources are encoded independently and decoded jointly. In [2] and [3] it is shown that with this approach similar coding efficiency can be achieved as in the conventional case where sources are jointly encoded *and* decoded, e.g. [4]. This efficiency cannot be reached by stereo or multi-view systems in which the signals of the different cameras are encoded and decoded independently of each other. As opposed to a set-up where stereo video streams are jointly encoded using traditional coding schemes, the system we propose does not require communication between the cameras. Indeed, only during the joint decoding phase the information of the other streams is used, which leads to a flexible and energy-efficient system.

DV coding is a recent research area and the application of

This work has been supported by the Spanish Ministry of Education and Science and the European Commission (FEDER) under grant TEC2005-07751-C02-01. A. Pižurica is a postdoctoral research fellow and L. Tessens is a research assistant of FWO Flanders.

1-4244-1354-0/07/$25.00 ©2007 IEEE

DV coding to the low-complexity encoding of 2D video has been thoroughly investigated in, amongst others, [5–8]. Unlike conventional video coding, low-complexity encoding is achieved by *intra*-frame encoding and *inter*-frame decoding. As the DV decoders (and not the *en*coders) perform motion estimation and motion compensated interpolation, most of the computational burden is moved from the encoder to the decoder. Several extensions of these 2D DV coding schemes for the coding of multi-view video have been developed [9–11]. However, the use of DV coding in a smart camera, *content-aware*, flexible, multi-view environment has to the best of the authors' knowledge not been investigated yet. A higher coding efficiency for the ROI is achieved than in our previous work, the non-content-aware multi-view coding system of [11].

The remainder of this paper is organized as follows. In Section 2, we introduce in general terms the structure of our proposed multi-view coding framework. Subsequently, we study in more detail the two main components of this framework, i.e., the smart WZ encoders (in Section 2.1) and the central decoding unit (in Section 2.2). In Section 3, we present the experimental coding efficiency results of the proposed framework and assess the gain in rate-distortion performance due to content-awareness. Finally, the conclusions are presented in Section 4.

2. THE PROPOSED FRAMEWORK

The main idea of this paper is to develop a flexible, content-aware and efficient multi-view video coding system. To this aim we propose a general framework that consists of 3 major parts: a central camera (later referred to as key camera), smart Wyner-Ziv (WZ) cameras and a central decoding unit (see Figure 1). The key camera encodes the video data with a conventional block-based inter-frame coding technique, e.g. H.264/AVC. This central camera is completed with the smart WZ cameras to form a multi-view system. These cameras are low-complexity encoders since no motion estimation is performed between the different frames (instead this computational burden is moved to the central decoding unit, see Section 2.2). For each frame, the ROI is detected and accordingly a content-aware and efficient representation of the scene is transmitted to the central processor. In the following, the frames coming from the key camera will be called key frames and the frames from the WZ cameras will be called WZ frames.

Notice that in order to obtain a flexible system, no communication between the cameras is assumed, so each of the cameras (both key and WZ) encodes its video signal independently of the other cameras. To still achieve a high coding efficiency, we rely on the above-described distributed coding principles, i.e. we decode all the signals jointly in a central decoding unit. In this central decoding unit, we exploit the temporal correlation within the video streams from the

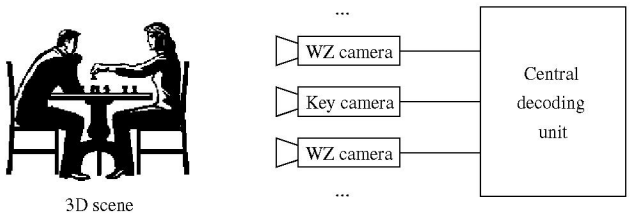

Fig. 1. General framework of the proposed multi-view video coding system

low-complexity WZ encoders and the spatial correlation between the different view-correlated sequences. Moreover, the content-awareness of the smart WZ cameras allows us to achieve higher coding efficiency for the ROIs than in our previous work [11].

In the following two subsections we will explain the main components of the described system in more detail. In Section 2.1 we will show how the smart WZ encoders are built up and in Section 2.2 we will explore the central decoding unit and we will explain how motion within the sequences and disparity between the sequences are estimated and exploited.

2.1. Smart Wyner-Ziv encoders

The smart WZ coding scheme provides a content-aware, distributed and efficient representation of the WZ frames. To this aim, the encoding process of a frame consists of 2 main parts: the ROI extraction and the WZ coding of the extracted ROI. The total computational burden of the encoding process is low since no motion estimation is performed between consecutive frames and since only the ROI of the frame is considered (the remaining part of the frame is generated at the central decoder without information from the WZ cameras, as will be explained in Section 2.2.2). A block diagram of the smart WZ encoder is depicted in Figure 2.

2.1.1. Extraction of the ROI

The Region of Interest (ROI) is the significant part of the frame. For instance, this ROI could be a moving object or even multiple moving objects. In this work, the background of the captured scene, i.e., those objects that remain still along the video sequence, is assumed to be static. This background is acquired by a background extraction initialization step. Equivalently, the background could be obtained by a more powerful and flexible background extraction algorithm, thus allowing for changes in the background. We leave this for future work. Then, for each frame the ROI extraction is a low-complexity algorithm that is based on the idea that the ROI of the frame differs from the background [12]. Hence, in order to obtain the ROI, first the difference between the frame to be encoded and the background is taken. Subsequently, the obtained difference image is processed as follows. First, a binary image is created by thresholding the difference between the considered frame and the background. The optimal

threshold value t_1 is empirically determined. Then, we apply the morphological additive operators 'opening' and 'closing'. The 'opening' operator (with a square structuring element) is used to remove isolated noisy pixels that erroneously take part in the ROI. Indeed, the result of the opening operation is an elimination of details smaller than the structuring element. On the other hand, the 'closing' filter (also with a square structuring element) joins the ROI to a more coherent unit. The optimum size of the structuring element depends on the type of video sequence and will be specified in Section 3. After these steps, we know for each pixel whether it belongs to the ROI or not. However, for the encoding, we prefer a ROI that is made up of blocks (of block size $L \times L$) instead of pixels. To this end, we use a decimate-filter (with decimating factor L), which determines for each block of $L \times L$ pixels a single value that reflects the number of pixels belonging to the ROI. Previously to this filter a low-pass filter is used to avoid aliasing. The obtained decimated image is thresholded (with threshold value t_2) to yield a block-based binary mask that shows for each block whether it belongs to the ROI (value 1) or not (value 0).

2.1.2. DV coding of the extracted ROI

A particular type of DV coding is pixel-domain DV coding with side information. The theoretical basics of coding with side information are described in [2, 3]. Applying these principles to the particular case of DV coding of stereo- or multi-view sequences, we propose to encode the extracted ROIs of the frames of the WZ camera separately, but to decode them with the knowledge of certain side information about them. This ROI side information is essentially an image (of the size of the ROI) that is correlated with the original ROI. It is generated at the decoder, since at the decoder information about the motion and the disparity of the sequences is estimated, and this information is used to generate the side information. In Section 2.2 we will explain in more detail how the ROI side information is generated.

The DV coding unit of this system is based on the one developed in our previous work about multi-view coding [11]. First, the frames are split up in GOPs (Group Of Pictures) of g frames. Each first frame of the GOP is a WZ Intra(I)-frame, the other $g-1$ frames are the WZ Predicted(P)-frames. The ROIs of the WZ I-frames W_i are encoded with a conventional intra-frame encoder, e.g. JPEG-2000. For the WZ P-frames, the pixel values of the ROI (and not of the entire frame as in [11]) of W_{i+j} ($j = 1, \ldots, g-1$) are first quantized with a uniform fixed-rate quantizer Q of 2^M levels. Subsequently, bit planes (BPs) are extracted from the quantization indices q_{i+j}. Then, the m most significant BPs $b_{i+j,k}$ ($1 \leq k \leq m$, $0 \leq m \leq M$) are independently encoded by a Slepian-Wolf (SW) coder [2]. The transmission and decoding of BPs is done in order of significance (the most significant BPs are transmitted and decoded first). The SW coding is implemented with efficient channel codes that yield the parity bits of the bit planes $b_{i+j,k}$, which are transmitted. The number of parity bits is determined by rate allocation through a feedback channel (see [8] for more details). Notice that the described DV coding operations are computationally very light.

2.2. Central decoding unit

The DV decoding unit described in this section is similar to the one from [11], the main difference being that the WZ P-frames decoder operates on ROIs instead of on entire frames. Consequently, the central decoding unit reconstructs the WZ frames in two stages. First, the ROI is obtained from the transmitted WZ parity bits together with the ROI side information generated at the central decoding unit (see Section 2.2.1). Secondly, the parts of the WZ frame that do not belong to the ROI (and which in this work are assumed to be unimportant) are recovered by an adequate perspective transformation from the corresponding pixels of the key camera (see Section 2.2.2). In order to accomplish these stages, the central decoding unit gathers all the encoded sequences from all the cameras (the key camera and the WZ cameras). The block diagram of the central decoding unit is depicted in Figure 3.

The compressed key frames coming from the central camera are decompressed in a conventional way, depending on the chosen inter-frame compression standard, e.g. H264/AVC. For the other video streams, coming from the smart WZ encoders, we distinguish between the two categories of frames within the GOP of length g: WZ I-frames and WZ P-frames. The ROIs from the WZ I-frames W_i are decoded using a conventional intra-frame decoder, e.g. JPEG-2000. For the WZ P-frames W_{i+j} ($j = 1, \ldots, g-1$), ROI side information needs to be generated for the decoding of the ROI. An adequate perspective transformation of the pixels of the key frame is applied to reconstruct the remaining pixels of the WZ frame.

2.2.1. Decoding of the ROI

If we consider the GOP with WZ I-frame W_i and GOP size g, we obtain the ROI side information for each WZ P-frame W_{i+j}, $j = 1, \ldots, g-1$ (i.e. S_{i+j}, $j = 1, \ldots, g-1$) through the following 4 steps. The decoding is done sequentially, or in other words, for the decoding of frame W_{i+j} we need the decoded frames $\hat{W}_{i+j-1}$, $\hat{K}_{i+j-1}$ and $\hat{K}_{i+j}$. For the decoding of the first WZ P-frame of the GOP W_{i+1}, this scheme allows for an obvious initialization from the decoded WZ I-frame $\hat{W}_i$ and the decoded key frames $\hat{K}_i$ and $\hat{K}_{i+1}$.

Step 1: Block-based disparity estimation
Disparity refers to the difference in image coordinates of objects and regions that are visible in both views of a pair of stereo sensors (or more general of an array of view-correlated sensors). In this step, we estimate the disparity between the

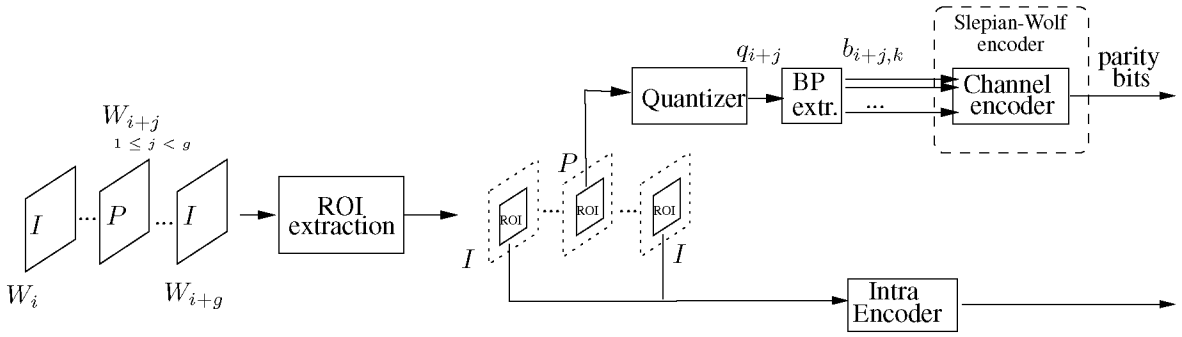

Fig. 2. Block diagram of the smart Wyner-Ziv encoders.

decoded ROI of the WZ frame $\hat{W}_{i+j-1}$ and the best corresponding part in the key frame $\hat{K}_{i+j-1}$ (i.e. the decoded ROI in the key frame). As explained in Section 2.1.1, the ROI of the WZ frame $\hat{W}_{i+j-1}$ consists of blocks of size $L \times L$. Each block is characterized by its top-left coordinates (k_1, k_2). For each block of this ROI, the disparity is characterized by the disparity vector $(d_1(k_1, k_2), d_2(k_1, k_2))$. This disparity vector is obtained by a block matching algorithm with as matching criterion the Minimum Mean Square Deviation (MMSD). More specifically, for each block of the ROI the disparity vector $(d_1(k_1, k_2), d_2(k_1, k_2))$ is the $(d_1, d_2) \in [-S_1, S_1] \times [-S_1, S_1]$ that minimizes

$$\frac{1}{L^2} \sum_{l_1=0}^{L} \sum_{l_2=0}^{L} \Big(\hat{K}_{i+j-1}(k_1 + l_1 + d_1, k_2 + l_2 + d_2) - \hat{W}_{i+j-1}(k_1 + l_1, k_2 + l_2) \Big)^2. \quad (1)$$

The parameters that characterize this disparity estimation technique are the search area $[-S_1, S_1] \times [-S_1, S_1]$ and the block size $L \times L$.

Step 2: Block-based motion estimation

We assume that the trajectory of the motion in the WZ cameras is similar to that of the key cameras, and therefore the motion of the ROI of WZ frame W_{i+j-1} (and from that its position in the WZ frame W_{i+j}) is estimated based on the motion between the decoded key frames $\hat{K}_{i+j-1}$ and $\hat{K}_{i+j}$. More concretely, we characterize the motion for each block of the ROI of $\hat{W}_{i+j-1}$ by the motion vector $(m_1(k_1, k_2), m_2(k_1, k_2))$ (with (k_1, k_2) the top-left coordinates of the considered block) and obtain this vector by applying block matching with the MMSD criterion between the key frames $\hat{K}_{i+j-1}$ and $\hat{K}_{i+j}$ on block positions determined by the disparity between the ROI in $\hat{W}_{i+j-1}$ and the best corresponding blocks in $\hat{K}_{i+j-1}$. This disparity is characterized by the disparity vector $(d_1(k_1, k_2), d_2(k_1, k_2))$ as obtained in Step 1. Then, for each block of the ROI the motion vector $(m_1(k_1, k_2), m_2(k_1, k_2))$ is the $(m_1, m_2) \in [-S_2, S_2] \times [-S_2, S_2]$ that minimizes

$$\frac{1}{L^2} \sum_{l_1=0}^{L} \sum_{l_2=0}^{L} \Big(\hat{K}_{i+j}(k_1 + l_1 + d_1 + m_1, k_2 + l_2 + d_2 + m_2) - \hat{K}_{i+j-1}(k_1 + l_1 + d_1, k_2 + l_2 + d_2) \Big)^2. \quad (2)$$

The parameters that characterize this disparity estimation technique are the search area $[-S_2, S_2] \times [-S_2, S_2]$ and the block size $L \times L$.

Step 3: Generating the side information

To obtain the pixel values of the ROI side information S_{i+j} (with size, shape and position equal to the ROI extracted at encoding) for the decoding of the ROI of frame W_{i+j}, we rely on the disparity and motion estimation of Step 1 and Step 2. First, we perform motion compensation on the ROI of $\hat{W}_{i+j-1}$. More specifically, for a block of the ROI of $\hat{W}_{i+j-1}$ with top-left coordinates (k_1, k_2), the pixel values of this block are copied to a block in the ROI side information S_{i+j} with top-left coordinates $(k_1 + m_1(k_1, k_2), k_2 + m_2(k_1, k_2))$. Notice that possibly parts of the copied blocks will fall outside the ROI of frame W_{i+j}; these pixels are discarded. Moreover, due to occlusion, part of the pixels of the side information ROI S_{i+j} remain undetermined after the motion compensation. For these pixels, we perform disparity compensation on $\hat{K}_{i+j}$: for a pixel of S_{i+j} with coordinates (p_1, p_2) and that is part of the block with top-left coordinates (k_1, k_2), we copy the pixel value on position $(p_1 + d_1(k_1, k_2), p_2 + d_2(k_1, k_2))$ of key frame $\hat{K}_{i+j}$. The obtained frame S_{i+j} will be the ROI side information used to conditionally decode the ROI of W_{i+j}.

Step 4: Decoding with side information

From the ROI parity bits sent by the WZ encoder (see Section 2.1), the corresponding BP $b'_{i+j,k}$ extracted from the available ROI side information S_{i+j} (as generated in Step 3), and the previously decoded BPs $b_{i+j,l}$ ($l = 1, \ldots, k-1$), the SW decoder [2] obtains $b_{i+j,k}$ (with $1 \leq k \leq m$, $0 \leq m \leq M$). Note that $b'_{i+j,k}$ can be considered the result of transmitting $b_{i+j,k}$ through a noisy *virtual channel*. The SW decoder is a channel decoder that recovers $b_{i+j,k}$ from its noisy version $b'_{i+j,k}$. Finally, the decoder reconstructs each

pixel of the ROI of frame $\hat{W}_{i+j}$ using the ROI side information S_{i+j} and the decoded BPs $b_{i+j,k}$ $(k = 1, \ldots, m)$ through

$$\hat{W}_{i+j}(p_1, p_2) = \begin{cases} w_{\mathrm{L}}, & S_{i+j}(p_1, p_2) < w_{\mathrm{L}} \\ S_{i+j}(p_1, p_2), & w_{\mathrm{L}} \leq S_{i+j}(p_1, p_2) \leq w_{\mathrm{R}} \\ w_{\mathrm{R}}, & S_{i+j}(p_1, p_2) > w_{\mathrm{R}} \end{cases} \tag{3}$$

with $w_{\mathrm{L}} = \sum_{k=1}^{m} b_{i+j,k}(p_1, p_2)\, 2^{8-k}$ and $w_{\mathrm{R}} = w_{\mathrm{L}} + 2^{8-m} - 1$ and (p_1, p_2) the pixel position of a pixel of the ROI W_{i+j}. w_{L} and w_{R} are respectively the left and right border of the interval to which the original pixel $W_{i+j}(p_1, p_2)$ belongs according to the decoded bits $b_{i+j,k}(p_1, p_2)$ $(k = 1, \ldots, m)$.

2.2.2. Background generation

To recover the parts of the WZ frames that fall outside of the ROI, the key camera view is mapped to the views of the WZ cameras. This method ensures us that no changes in the background are missed (dynamic background). Ouaret *et al.* have used a similar technique in [13] to obtain multi-view homography-based side information. In our case, however, the mapped view will not be used as side information (indeed, no WZ bits have been sent for these parts of the frames) but as the background generation of the WZ frames.

The mapping of the key camera view to the WZ views is implemented as a global image transformation, which means that all corresponding points visible in two camera views are assumed to be linked by the same transformation. We model this relationship by a perspective transformation, which is a good approximation for most natural scenes. It is only correct if all points visible in the two views lie within a plane in 3D-space. However, if the observed scene is at a sufficiently large distance from the cameras, the depth-relief of the background is very small and all points belonging to the background can be considered lying in a plane in 3D-space.

In homogeneous coordinates, this perspective transformation can be expressed by a 3×3 matrix relating the points in the key view to the corresponding points in the WZ views:

$$\lambda \begin{bmatrix} x_{\mathrm{WZ}} \\ y_{\mathrm{WZ}} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{bmatrix} x_{\mathrm{key}} \\ y_{\mathrm{key}} \\ 1 \end{bmatrix} \tag{4}$$

where λ is a scaling factor. The 8 parameters of the perspective transformation between the key view and each of the WZ views are determined by minimizing a cost function that is a measure of dissimilarity between the WZ view and the warped key view. This calculation needs to be done for the first frame only, as the parameters of the transformation remain the same as long as the position and orientation of the cameras with respect to each other and to the background do not change.

We choose a robust function as the cost function in order to limit the bias introduced by outliers. In order to estimate the transformation parameters, we minimize the sum of the Lorentzian cost function of all pixels using the gradient descent optimization scheme:

$$\sum_i \log(1 + \frac{1}{2} e_i^2(a_{11}, \ldots, a_{32})) \tag{5}$$

where e_i denotes the error between the WZ view and the warped key view at the pixel location i. To handle large transformations, we minimize the cost function iteratively in a coarse-to-fine multi-resolution framework based on Gaussian pyramids. In each gradient descent iteration, we resample the warped key view using the bilinear interpolation method. The optimum is reached within 30 iterations per scale.

The transformation parameters describe then the perspective transformation of the dominant background plane between the key and WZ views in 3D-space since all points that deviate much from it, i.e. outliers, contribute less to the minimization of the cost function.

3. EXPERIMENTAL RESULTS AND DISCUSSION

In this section we will present the performance of the proposed *content-aware* multi-view video system and more specifically the performance of the smart WZ coders. First, we will evaluate the rate-distortion performance of these WZ coders, by showing experimental results of the obtained quality of the ROIs in function of the bits spent on the WZ P-frames. We will compare these results with the *non-content-aware* method described in [11], both in terms of rate-distortion gain due to content-awareness and in terms of performance of the turbo codes. For a comparison that shows how much rate-distortion gain is derived from the joint multi-view decoding, we refer to our previous work [11]. Subsequently, we will evaluate the quality of the background generation algorithm.

For the experiments, we consider the specific case of a stereo video system that consists of a central key camera and one smart WZ camera. For each WZ P-frame, the smart WZ coder first extracts the ROI (see Section 2.1.1). The threshold parameters t_1 and t_2 of the ROI extraction are set to 15 and 20 respectively. The size of the square structuring elements of the morphological operators is 3×3. Subsequently, the extracted ROI is DV coded. Therefore, the ROI is decomposed into its 8 BPs. Then, the m most significant BPs are separately encoded by a channel coder; the other BPs are discarded. The higher m, the higher the encoding bit rate. In our experiments, $m = 1, \ldots, 4$. The channel coder used is a turbo coder composed of two identical constituent convolutional encoders of rate 1/2 with generator polynomials $(1, 33/31)$ in octal form [14]. As in [6–8, 11, 15], the turbo coder assumes a Laplacian residual distribution between the (ROI of the) WZ P-frame and the (ROI) side information. The key frames and the ROIs of the WZ I-frames are assumed to be losslessly transmitted. Side information is generated at the

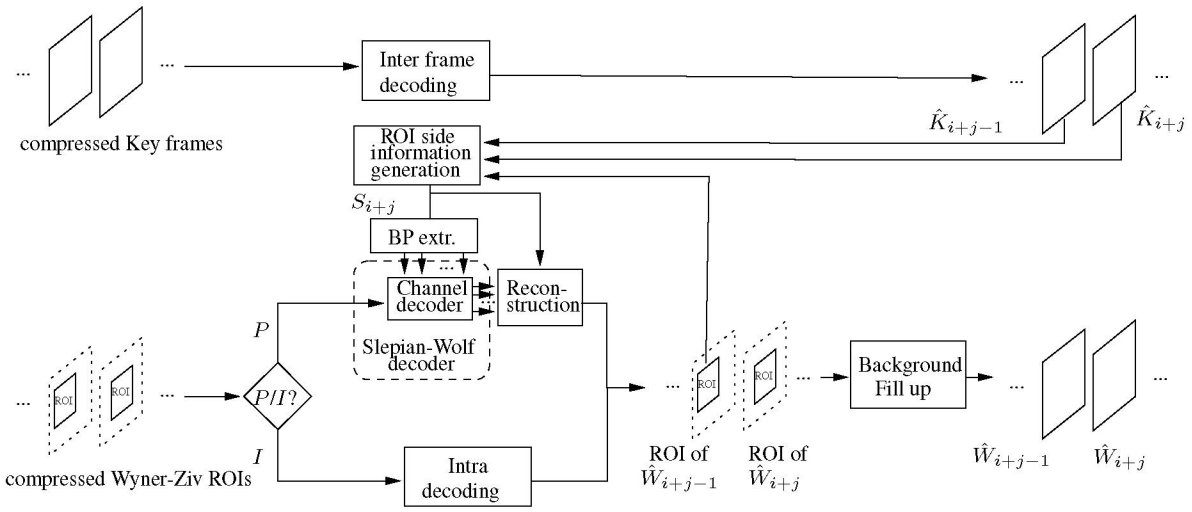

Fig. 3. Block diagram of the central decoding unit.

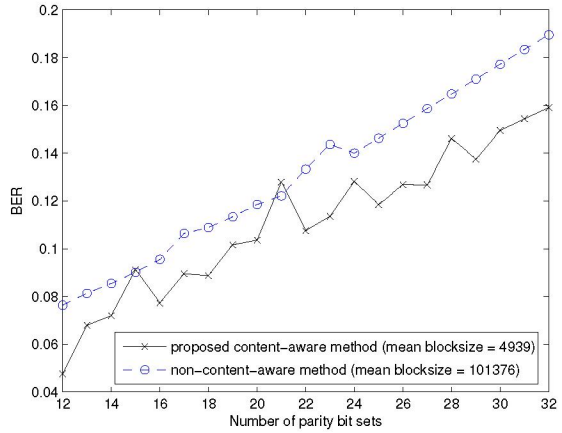

Fig. 5. Turbo code performance: BER vs. number of parity bit sets for the proposed method (mean block size = 4939) and the non-content-ware method (mean block size = 101376).

central decoding unit as described in Section 2.2. The GOP size is chosen to be 5. For the disparity and motion estimation, the block size is 16×16 and the search ranges $[-S_1, S_1]$ and $[-S_2, S_2]$ are set to $[-8, 8]$. We encoded 2 stereo test sequences: one synthetic sequence *Basement* and one natural stereo sequence *Courtyard*. The sequences have a CIF resolution (352×288 pixels/frame) and are coded at 30 frames/s. For the encoding, only the luminance component is taken into account.

Figure 4 shows for the 2 test sequences the average PSNR of the ROIs over the bit rate for the WZ P-frames. For comparison, the rate-distortion of the WZ P-frames when coded with the *non-content-aware* method described in [11] is also shown. As could be expected, we observe that for the 2 sequences a much better rate-distortion performance for the ROIs is achieved with the proposed content-aware method, since in this case no bits are spent on the coding of the background (which in our case is assumed to be unimportant, since we focus on the ROIs).

It is shown in literature that the performance of the turbo codes depends on the block size of the coded information blocks [16]. From a block size of 10000 bits on, the performance of the turbo codes is optimal and remains constant. This is an important issue in our case, since the size of the ROI varies and the ROI can be smaller than the lower bound of 10000 pixels. For instance, for the case of *Basement*, the mean block size of the ROI is 3904 and for the sequence *Courtyard*, the mean block size of the ROI is 7008. To assess the loss in performance caused by this smaller block size, we show in Figure 5 the performance of the turbo codes for the proposed method and compare it to the results for the case of larger blocks [15]. In this figure, the Bit Error Ratio (BER) is plotted as a function of the number of transmitted parity bit sets K. The total rate R corresponding to this number of parity bit sets is a function of the puncturing period of the turbo code T_{punc} [14], the number of coded information bits N, and the frame rate r: $R = rKN/T_{\text{punc}}$. We observe that the mean loss in BER is approximately 0.02. This loss is very well compensated for by the large gain in rate-distortion due to content-awareness, as shown in Figure 4.

Figure 6 shows for the 2 test sequences 3 images that allow us to evaluate the quality of the background as generated through the perspective transformation described in Section 2.2.2. The left image is the generated background, the middle image is the ground truth background, and the right image is a decoded frame composed of a WZ decoded ROI and the generated background. The ROI is delineated with a red line. We observe that the PSNR quality of the generated backgrounds is quite low (between 20 and 23 dB). However, the visual quality is, apart from some smoothness artifacts, quite good. These low PSNR values can be attributed to two

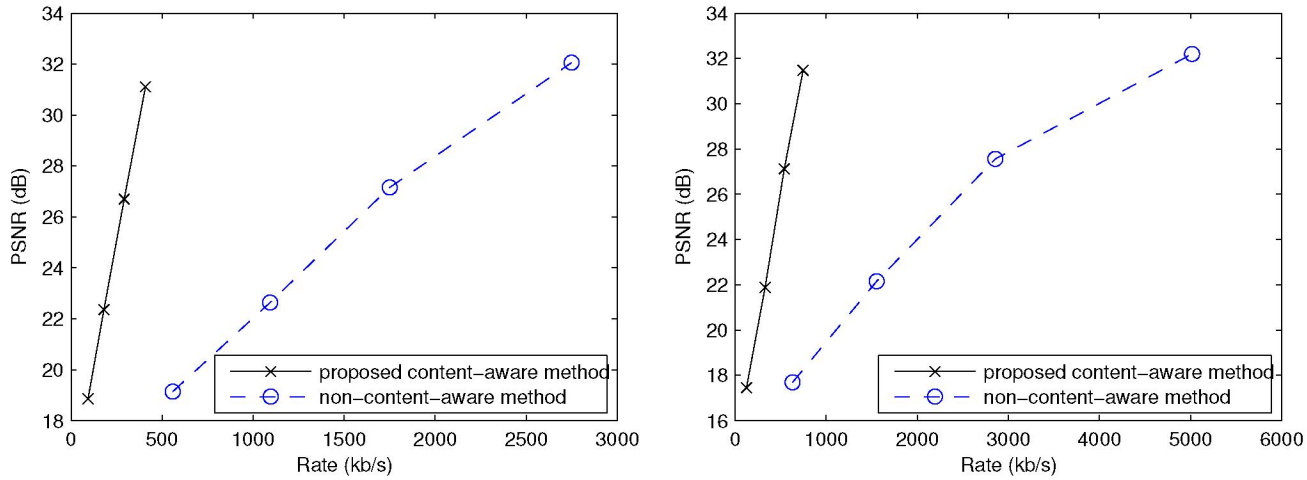

Fig. 4. Average PSNR for the ROIs of the WZ P-frames vs. bit rate of the WZ P-frames for our proposed *content-aware* method. The results are shown for the synthetic stereo sequence (a) *Basement* and for the natural stereo sequence (b) *Courtyard*. Compared is the rate-distortion of the WZ P-frames when decoded with the similar, but *non-content-aware* method described in [11].

mechanisms. First of all, the perspective transformation introduces some geometric distortions. These distortions are not visibly disturbing, but they lead to large differences between pixel values on corresponding positions in the ground truth background and the generated background, and hence to low PSNR values. Second, the pixel values as recorded by the left and right cameras might differ for the same objects due to hardware differences in the cameras. This also leads to a degradation of the PSNR. The black parts of the generated background images arise from the fact that some parts of the WZ view are not visible in the key view and hence cannot be reconstructed through a perspective transformation. Those parts are not taken into account for the calculation of the PSNR.

4. CONCLUSIONS

In this paper, we presented an efficient, flexible and content-aware multi-view video coding system. The proposed framework consists of a central processor and camera, completed by a freely adjustable number of smart Wyner-Ziv cameras. In order to obtain a flexible and energy-efficient framework, the cameras do not communicate between each other. However, good coding efficiency is still achieved by relying on Distributed Video coding principles, i.e. joint decoding of the independently encoded frames of the multi-view cameras, and because the smart WZ cameras provide a *content-aware* representation of their viewpoint. Experimental results show that our method achieves a much better rate-distortion performance than a similar non-content-aware system.

5. ACKNOWLEDGEMENTS

We thank Stefaan Lippens for the synthetic test sequences.

6. REFERENCES

[1] A. Vetro, W. Matusik, H. Pfister, and J. Xin, "Coding approaches for end-to-end 3D TV systems," in *Proc. of the PCS*, San Francisco, CA, Dec. 2004.

[2] J. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, July 1973.

[3] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[4] J.N. Ellinas and M.S. Sangriotis, "Stereo video coding based on interpolated motion and disparity estimation," in *Proc. of ISPA*, Sept. 2003, vol. 1, pp. 301 – 306.

[5] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conference on Communication, Control, and Computing*, Allerton, IL, USA, Oct. 2002.

[6] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Proc. of SPIE VCIP*, San Jose, CA, USA, January 2004.

[7] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP Conference*, Slovack, Republic, June 2005.

[8] M. Morbée, J. Prades-Nebot, A. Pižurica, and W. Philips, "Rate allocation algorithm for pixel-domain

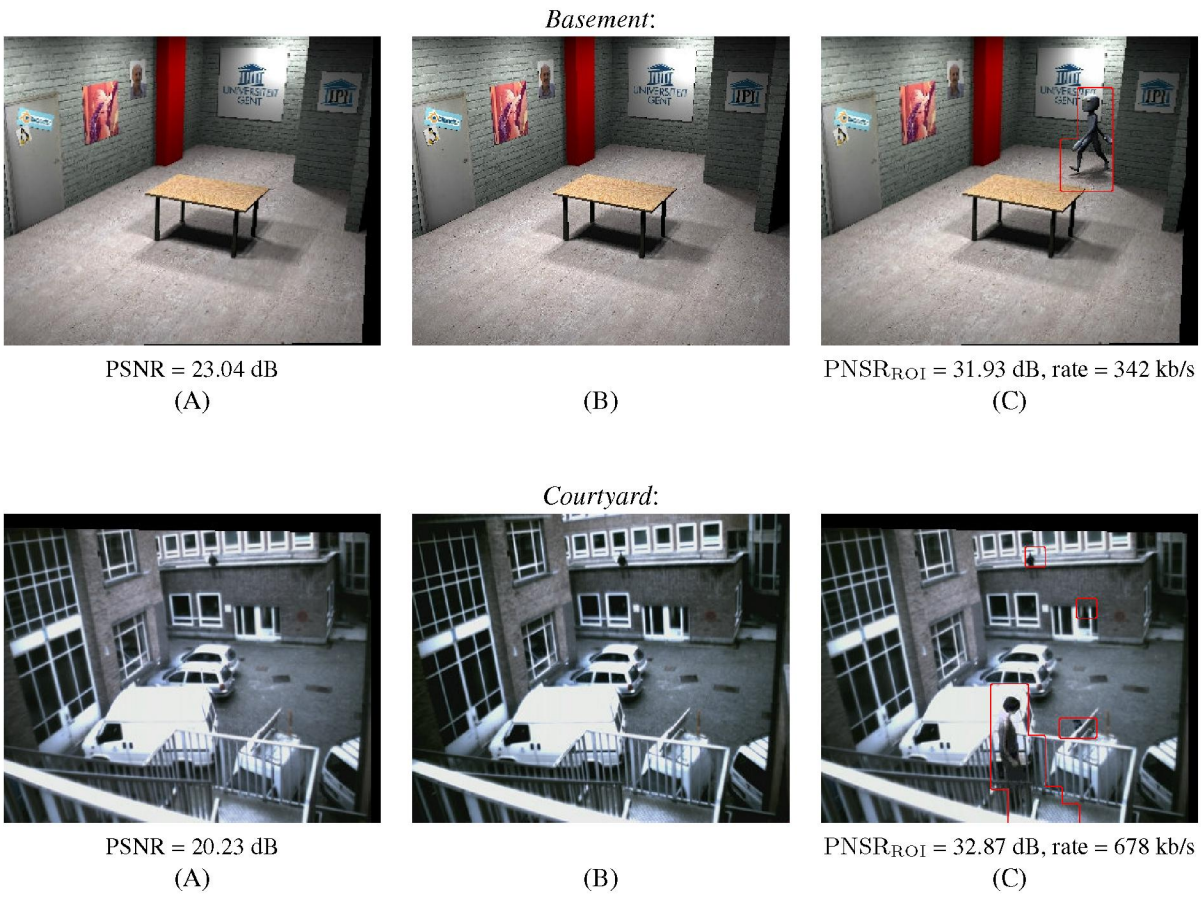

Fig. 6. Background generation for the sequences (a) *Basement* (frame 4) and (b) *Courtyard* (frame 26). (A) is the generated background, (B) is the ground truth background and (C) is a decoded frame composed of a WZ decoded ROI and the generated background (the ROI is delineated with a red line).

distributed video coding without feedback channel," in *ICASSP*, Honolulu, Hawaii, USA, April 2007, pp. 521–524.

[9] M. Flierl and B. Girod, "Coding of multi-view image sequences with video sensors," in *Proc. of ICIP*, Atlanta, GA, USA, Oct. 2006.

[10] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed multi-view video coding," in *Proc. of SPIE VCIP*, San Jose, CA, USA, Jan. 2006, vol. 6077, pp. 290–297.

[11] M. Morbée, Linda Tessens, J. Prades-Nebot, A. Pižurica, and W. Philips, "A distributed coding-based extension of a mono-view to a multi-view video system," in *3D TV Conference*, Kos, Greece, May 2007.

[12] M. Morbée, J. Prades-Nebot, A. Pizurica, and Philips. W., "Content-based mpeg-4 fgs video coding for video surveillance," in *Proc. of SPS-DARTS 2006 (the second annual IEEE Benelux/DSP Valley Signal Processing Symposium*, March 2006, pp. 135–138.

[13] Mourad Ouaret, Frederic Dufaux, and Touradj Ebrahimi, "Fusion-based multiview distributed video coding," in *Proc. of VSSN*, New York, NY, USA, 2006, pp. 139–144, ACM Press.

[14] D. Rowitch and L. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 948–959, June 2000.

[15] M. Morbée, J. Prades-Nebot, A. Pižurica, and W. Philips, "Improved pixel-based rate allocation for pixel-domain distributed video coders without feedback channel," in *ACIVS*, Delft, the Netherlands, August 2007, Lecture Notes in Computer Science, pp. 663–674, Springer-Verlag.

[16] S. Dolinar, D. Divsalar, and F. Pollara, "Turbo code performance as a function of code block size," in *Proc. of ISIT*, August 1998, p. 32.