# DISTRIBUTED SOURCE CODING APPLICATION TO LOW-DELAY FREE VIEWPOINT SWITCHING IN MULTIVIEW VIDEO COMPRESSION

*Ngai-Man Cheung and Antonio Ortega*

Signal and Image Processing Institute, Univ. of Southern California, Los Angeles, CA

## ABSTRACT

Multiview video coding (MVC) exploits the temporal and spatial redundancy between neighboring frames of the same view or that of adjacent views to achieve compression. Free viewpoint switching, however, poses challenges to MVC, as when a user is able to choose different playback paths it would become unclear to encoder which previously reconstructed frame would be available for decoding the current frame. Therefore, to support free viewpoint switching in MVC, encoder would need to operate under uncertainty on the decoder predictor status. Extending our previous work on video compression with decoder predictor uncertainty, this paper proposes a MVC algorithm based on distributed source coding (DSC) to tackle the free viewpoint switching problem, where the encoder has access to several predictor candidates but there is uncertainty as to which one will be available at decoder to serve as predictor for the current frame. Since cross-view correlation could be much less significant than temporal correlation, a main challenge of the present DSC application is to achieve competitive compression efficiency. Some of the novelties of the proposed MVC algorithm are that it incorporates different macroblock modes and significance coding within the DSC framework, so that competitive coding performance can be achieved. Experiments demonstrate the proposed algorithm can outperform solutions based on intra or closed-loop predictive (CLP) coding in terms of compression efficiency. In addition, the proposed method incurs a negligible amount of drifting, making it an attractive solution to facilitate low-delay, free viewpoint switching.

***Index Terms***— Distributed source coding, Slepian-Wolf, multiview video, free viewpoint switching

## 1. INTRODUCTION

Multiview video consists of multiple video sequences capturing different views of the same scene (Figure 1) [1]. Multiview video can potentially allow users to play back and switch between different views interactively. In addition, virtual intermediate views could be conveniently rendered from the video data based on users' requests in real-time [2]. With these features, multiview video has become the core technology in a number of emerging applications such as free viewpoint TV [1]. Multiview video coding (MVC) usually exploits the correlation between neighboring frames of the same view or that of adjacent views [2]. Free viewpoint switching, however, poses challenges to MVC. Essentially, with free viewpoint switching, at encoding time it is unclear which previously decoded frames will be available at the decoder to serve as predictor for decoding the current frame. This is illustrated in Figure 1. At the decoder, depending on whether the user is staying in the same view as in Figure 1.a, or switching views as in Figure 1.b, either the previous reconstructed frame of the same view ($Y_0$) *or* that of another view (one of $\{Y_{\pm1}, Y_{\pm2}, ...\}$) may be available as predictor for decoding the current frame $X$, respectively. However, while the encoder has access to all these candidate predictors $\{Y_k\}$, crucially, it would not know exactly which one will be available at the decoder for decoding $X$, since this depends on the playback path chosen by the user. Therefore, to support free viewpoint switching in MVC, the encoder would need to operate under *uncertainty* about which predictors will be available at decoder. Figure 1.c depicts the case when user renders a virtual view using $X$ and $X'$. In this case, it would need to recover both $X$ and $X'$ using any one of the $Y_k$ available at decoder.

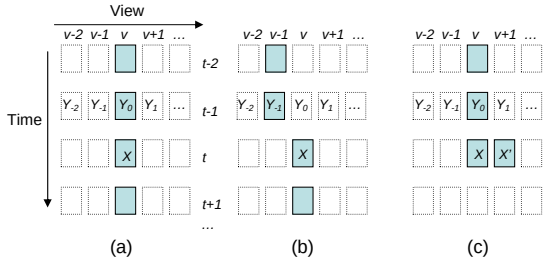


**Fig. 1**. Free viewpoint switching in multiview video playback: (a) User plays back the $v$-th view. $Y_0$ is available as predictor for decoding $X$. (b) User plays back $(v-1)$-th view and switches to the $v$-th view at time $t$. In this case, $Y_{-1}$ is the only predictor available for decoding $X$. (c) User plays back $v$-th view and renders a virtual view between $v$-th and $(v+1)$-th views at time $t$. $Y_0$ can serve as predictor for decoding $X$ and $X'$, from which a virtual view can be interpolated.

Note that our work focuses on low-delay, interactive applications, where feedback is not suitable since it may cause unacceptable delay during switching, and therefore it is not possible for decoder to inform the predictor status to encoder. Moreover, the problem formulation is also applicable to the situations when multiview video data are compressed offline. In these cases, encoder would not have any information about the playback path to be chosen by the end-users and therefore does not know which predictor $Y_k$ would be available for recovering $X$.

Conventional MVC uses closed-loop predictive (CLP) coding to exploit the correlation between neighboring frames, and motion-compensated prediction (MCP) residues or disparity-compensated prediction (DCP) residues are encoded along with the motion or disparity vectors information [2]. In order to support free viewpoint switching with CLP, encoder may send all the possible MCP/DCP residues $\{Z_i; i = 0, \pm1, ...\}$ to the decoder, where $Z_i = X - Y_i$ (following the notations in Figure 1), so that $X$ can be recovered no matter which $Y_i$ is available at the decoder. Each $Z_i$ would correspond to a P-frame in MPEG/H.26X video coding standards. There are two main problems, however, with CLP approach. First, coding performance is degraded because multiple prediction residues are included in the bitstream. Second, CLP may cause drifting. This is because,

This work was supported in part by NASA-JPL under the DRDF program.

in practical CLP systems, quantized versions of $Z_i$, $\hat{Z}_i$, are sent to the decoder. Therefore, the reconstructed sources $\hat{X}_i = \hat{Z}_i + Y_i$ are not identical when different $Y_i$ are used as predictors. Drifting may occur when $\hat{X}_i$ is used as reference for decoding future frames. Note that although some CLP algorithms employing advanced prediction-loop such as that in the H.264 SP-frame can overcome the drifting problem [3], the coding performance of these algorithms is comparable to that of the basic CLP we just discussed, as different residue $Z_i$ each corresponding to a different $Y_i$ would need to be generated and sent to the decoder[1]. In short, a fundamental shortcoming of CLP approach is that the prediction residue would be "tied" to a specific predictor. Therefore, under predictor uncertainty conditions, multiple residues each corresponding to a different predictor candidate would be required to be communicated to the decoder, so that severe mismatch and drifting can be avoided.

In this paper, we propose a MVC algorithm based on distributed source coding (DSC) [4, 5] that inherently supports free viewpoint switching. Specifically, we extend our previously proposed DSC-based video encoding algorithm [6, 7] to address viewpoint switching scenarios, where the encoder has access to the various (spatial or temporal) predictors, $Y_k$, which will play the role of side information (SI) at the decoder, but there is uncertainty as to which one will be used for decoding. Since spatially neighboring frames could be much less correlated compared to temporally consecutive frames, a main challenge of the present DSC application is to achieve competitive compression efficiency. In this paper, we demonstrate the proposed DSC-based MVC algorithm can achieve competitive performance and outperform solutions based on CLP or intra approach. In addition, the proposed algorithm incurs only a negligible amount of drifting: all DSC-coded macroblocks are identically reconstructed no matter which predictor is available at decoder.

Previous work on DSC-based multiview image or video coding mainly focuses on achieving distributed, independent encoding of different views at individual spatially-separated camera or video sensor, e.g., [8, 9]. The present work, however, targets an entirely different setting where different views are compressed in a centralized encoder similar to the assumptions in the current MPEG-MVC standardization work [1], while our main objective is to support free viewpoint switching. DSC has also been proposed to address random access and viewpoint switching in the compression of image-based rendering data/light fields and multiview video [10–12]. However, this prior work assumes encoder has knowledge of predictor status at the decoder, notably through using feedback, while in our case the encoder needs to operate with unknown predictor status. DSC application to address robust video transmission has recently been proposed in [13], which shares the same general philosophy as our problem. However, different assumptions are made. In particular, this work assumes that the encoder knows the probability that each predictor will be used, as determined by the packet erasure probability (whereas we assume all predictors are equally-likely to be used). This information is exploited to reduce the coding rate. In addition, the specific tools used are different from those proposed here. Our recent work [7] proposes DSC-based algorithms to tackle the general problem of encoding with uncertainty on decoder predictor status. The current work extends [7] to consider the specific structure in MVC, and demonstrates the performance of the DSC based solution in the case of more than two predictor candidates. Random access/viewpoint switching based on H.264 SP-frame and reference frame warping have also been proposed in [14] and [15] respectively for systems using feedback. However, in the present problem, where feedback is not available, SP-frame based solution would need to send multiple residues (each corresponds to a different predictor candidate), and therefore it is expected the coding performance of SP-frame based solution is comparable to that of CLP, and would be worse than that of our proposed algorithm.

[1]In the case of H.264 SP-frame, $Z_i$ would correspond to a primary or secondary SP-frame. And in general the coding efficiency of SP-frame is worse that that of P-frame due to the additional identical reconstruction requirement [3].

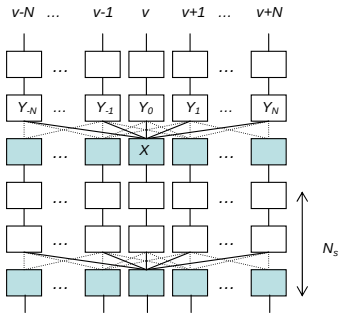


**Fig. 2**. Multiview coding structure with free viewpoint switching supported at every $N_s$ frames ($N_s = 3$ in the example depicted in this figure).

This paper is organized as follows. In Section 2 we present the proposed DSC-based MVC algorithm. Section 3 presents the experimental results and Section 4 concludes the work.

## 2. DSC-BASED MULTIVIEW VIDEO COMPRESSION

Figure 2 depicts an example of multiview coding structure that is of interest to the current work. In this case, view switching between $N$ adjacent views is supported at every $N_s$ frames. The video frames at the switching points (shaded ones in Figure 2) can be encoded using (i) intra coding; (ii) CLP; following the discussion in Section 1, $2N + 1$ residues $Z_i = X - Y_i$ would be sent for each frame at the switching points; (iii) DSC; details of the coding procedure to be discussed in the following sections. The rest of the frames can be encoded as standard P-frames, or as standard I-frames for those at the beginning of a GOP.

### 2.1. DSC-based MVC: intuition

We first review the intuition behind the proposed DSC-based MVC system [7]. In conventional CLP coding, the encoder computes a prediction residual $Z = X - Y$, between source $X$ and predictor $Y$, and sends it to decoder (Figure 3.a). DSC approaches the same compression problem taking a "virtual communication channel" viewpoint [16]. Specifically, $X$ is viewed as an input to a channel with *correlation noise* $Z$, and $Y$ is the output of the channel (Figure 3.b). Therefore, to recover $X$ from $Y$, encoder would send parity information to the decoder. Note that this parity information does not depend on a specific $Y$ being observed, and the encoder does not need $Y$ to generate the parity information - only the statistics of $Z$ are required for determining the amount of parity information to be sent. The decoder will be able to recover $X$ as long as a sufficient amount of parity information has been received.

To understand how DSC can be used for our problem, following the set-up in Figure 2, consider $2N + 1$ virtual channels each corresponding to a predictor candidate $Y_i$ (Figure 3.c). Each channel is characterized by the correlation noise $Z_i = X - Y_i$. To recover $X$ from any of these channels, the encoder could send an amount of parity information corresponding to the *worst* $Z_i$. Doing so, $X$ can be recovered no matter which $Y_i$ is available at the decoder. Note that encoder only needs to know the statistics of all the $Z_i$ to determine the amount of parity information, and this is feasible since $X$

and all $Y_i$ are accessible at encoder. In particular, the encoder does not need to know which $Y_i$ is actually present at decoder. Comparing with the CLP approach where the overhead increases with $N$, in the DSC approach, the overhead depends mainly on the worst-case $Z_i$.

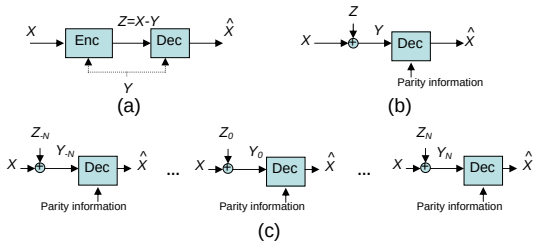


**Fig. 3**. Compression of input source $X$: (a) CLP coding; (b) DSC from the virtual channel viewpoint; (c) DSC approach to the free viewpoint switching.

## 2.2. Proposed algorithms

We present an overview of the proposed DSC-based MVC algorithms in this section (see Figure 4). Details can be found in [7].

### *2.2.1. Motion/disparity estimation and macroblock classification*

Each macroblock (MB) $M$ in current frame first undergoes block-based motion or disparity estimation w.r.t. each candidate reference frame $f_i$, and the corresponding motion or disparity vectors information (one per reference frame, $f_i$) are included in the bitstream. Denote $A_i$ the best motion/disparity-compensated predictor for $M$ obtained in $f_i$. If the difference between $M$ and $A_i$ is sufficiently small $M$ may be classified to be in a skip mode w.r.t. $f_i$. In such cases, the overhead in including multiple prediction residues could be small, and $M$ would be encoded using conventional CLP coding (similar to standard H.26X algorithms) w.r.t. the candidate reference frames which do not have skipping. However, for the majority of the macroblocks, there would be no skipping w.r.t. all $f_i$, and they would be encoded using DSC. Note that advanced rate-distortion (RD) based mode selection (as in H.264) can be used to decide between CLP and DSC to improve coding efficiency.

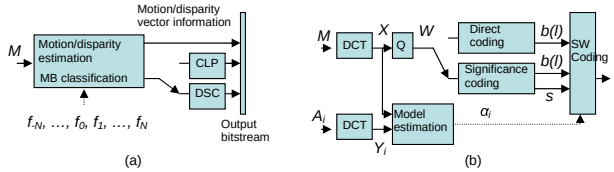


**Fig. 4**. (a) Proposed encoding algorithms; (b) Encoding macroblock $M$ using DSC. Here "Q" denotes scalar quantization.

### *2.2.2. Coefficient coding*

For those macroblocks $M$ to be encoded with DSC, we first apply standard $8 \times 8$ DCT to the pixel data to obtain the vector of transform coefficients $X$, and we then quantize $X$ to obtain the quantization indices $W$ (Figure 4.b). This is similar to intra-frame coding in standard H.26X algorithms. Denote $Y_i$ the DCT coefficient in $A_i$ corresponding to $W$ (recall $A_i$ is the best motion-compensated predictor from each $f_i$). We compress $W$ by exploiting its correlation with the worst case $Y_i$, so that it can be recovered with any $Y_i$ that may be present at the decoder.

The quantized values of the $K$ lowest frequency DCT coefficients (along a zig-zag scan order) are encoded with *direct* coefficient coding (DCC), and for the rest we use *significant* coefficient coding (SCC). In DCC, we form the $k$-th frequency coefficient vector by grouping together the $k$th $(0 \leq k \leq K-1)$ frequency coefficients from all $8 \times 8$ blocks in a frame (except those in skip modes). Then each of these vectors is converted into a bit-plane representation, and the bit-planes are passed to a Slepian-Wolf (SW) coder, where inter-frame/cross-view correlation is exploited to compress the bit-planes losslessly.

The quantized values of the $k$-th highest frequency coefficients, $k \geq K$, are encoded using SCC. Specifically, we first use a *significance bit* $s$ to signal if the quantized value of a coefficient is zero or not, so that only the value of a non-zero coefficient needs to be sent to the decoder. The significance bits for the $k$th frequency coefficients from all the $8 \times 8$ blocks in a frame (except those in skip modes) are grouped together to form a significance bit-plane to be compressed by the SW coder. On the other hand, the non-zero coefficients are grouped together to form coefficient vectors where all the DCT frequencies are combined, as we found that the correlation statistics of non-zero coefficients are similar at different frequencies. In the experiment, we use $K = 3$ following the discussion in [7].

### *2.2.3. Bit-plane compression and model estimation*

Bitplanes extracted from the $K$ coefficient vectors produced in DCC along with those produced in SCC are compressed by a SW coder, starting from the most significant bit-planes. The bit-plane at the $l$-th significance level, denoted $b(l)$, is to be compressed using a low density parity check (LDPC) based SW encoder [17], with $Y_i$ and decoded (more significant) bits $b(l+1), b(l+2), ...$ served as decoder side information. The SW decoder also needs the conditional probability $p(b(l)|Y_i, b(l+1), b(l+2), ...)$ to aid recovering $b(l)$. Therefore, the encoder also estimates the conditional p.d.f. $f_{X|Y_i}(x|y_i)$ for each coefficient vector and for each candidate predictor, and sends to the decoder all the model parameters, from which the conditional probability can be derived for any $Y_i$ available at decoder. In particular, we model $f_{X|Y_i}(x|y_i)$ using Laplacian distributions, and send all the variance information to the decoder [7].

Note that each $Y_i$ exhibits different levels of correlation with respect to $b(l)$. To ensure that $b(l)$ can be recovered with any of the predictor candidates $Y_i$, the encoder sends $R$ parity bits to the decoder, where $R = \max\ R_i$, and $R_i$ is number of parity bits required to recover $b(l)$ when $Y_i$ is used as predictor. By doing so, each bit-plane can be exactly recovered no matter which $Y_i$ is available at the decoder, and therefore $W$ can be losslessly recovered and $X$ reconstructed to the same value when any of the $Y_i$ is used as predictor. This eliminates drifting in DSC-coded macroblock.

Note that the proposed algorithm requires more computations in encoding and decoding compared with intra and CLP solutions. In particular, LDPC decoding is performed at the encoder to ensure the estimated rate using conditional entropy can lead to error-free decoding. If this is not necessary (e.g., if decoding error can be allowed in some bit-planes and this can be done with minimal impact to overall PSNR and drifting) then the encoder's complexity of the proposed algorithm would be similar to that of CLP. While this is a subject for further investigation, encoder complexity may not, in general, be a primary issue in the applications we describe since encoding is likely to be performed offline.

## 3. EXPERIMENTAL RESULTS AND DISCUSSION

Following the MVC structure depicted in Figure 2, we compare the coding performance of the following systems when compressing video frames at the switching points: (i) intra coding using H.263

I-frames; (ii) CLP approach with multiple residues each encoded using H.263 P-frames; (iii) proposed DSC-based MVC scheme implemented based on H.263 coding tools. Since all systems use the same (H.263) coding tools (e.g., half-pixel accuracy motion estimation) the comparison is fair. The rests of the frames are encoded using standard H.263 P-frame, or H.263 I-frame for those at the beginning of a GOP. We test the systems with MVC sequences Akko&Kayo and Ballroom, which are in $320 \times 240$ and encoded at 30fps and 25fps respectively. Figure 5 shows the comparison results, with $N = 1, N_s = 2$, i.e., switching between views $(v \pm 1)$ and $v$ is supported in every other frame, and either $Y_{\pm 1}$ or $Y_0$ could be present at decoder (therefore, the coding structure is $IPSPS...$, where $S$ are the switching points). As shown in the figure, the proposed algorithm outperforms CLP and intra coding. We also compare the approaches in terms of drifting with the following experiment: playback is switched from $(v-1)$-th view to $v$-th view at frame number 2. Figure 6 compares the PSNR of the reconstructed frames within the GOP with that of the non-switching case, where $v$-th view is being played back throughout the GOP. As shown in the figure, the proposed algorithm is almost drift-free.

Figure 7 illustrates how the coding performance of the proposed system scales with the number of predictor candidates. In this experiment, the temporally/spatially adjacent reconstructed frames are used as predictor candidates following the order depicted in Figure 7.a. As shown in Figure 7.b, the bit-rate of DSC-based solution increases at a much slower rate compared with that of the CLP counterpart. With the DSC-based solution, an additional predictor candidate would cause a bit-rate increase (when coding a bit-plane) only if it has the worst correlation among all the predictor candidates (w.r.t. that bit-plane). Note that in this case the DSC-based solution outperforms intra coding even with up to eight predictor candidates.

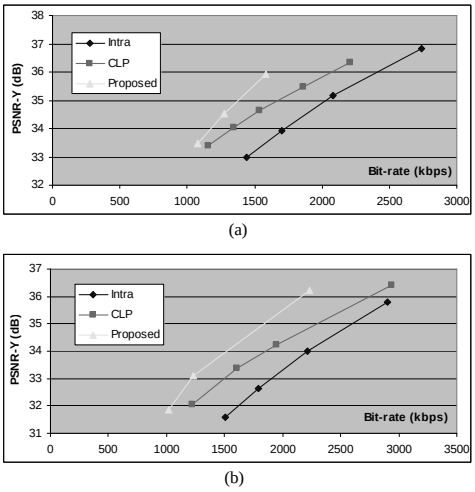


**Fig. 5**. Simulation results: (a) Akko&Kayo (30fps, GOP=30); (b) Ballroom (25fps, GOP=25). The results are the average of the first 30 frames (at the switching points, i.e., $S$) from 3 different views (Akko&Kayo: views 27th-29th; Ballroom: views 3rd-5th) arbitrarily chosen from all the available views.

## 4. CONCLUSIONS

We extended our previous work on video compression with uncertainty on decoder predictor to tackle the free viewpoint switching problem in MVC, based on DSC. The proposed encoding algorithm integrates macroblock mode and significance coding in the DSC framework to improve coding efficiency. We demonstrated the proposed algorithm can achieve better coding performance compared with other methods based on intra coding or CLP. In addition, the proposed algorithm suffers only a small amount of drifting, as all the DSC-coded macroblocks would be identically reconstructed.

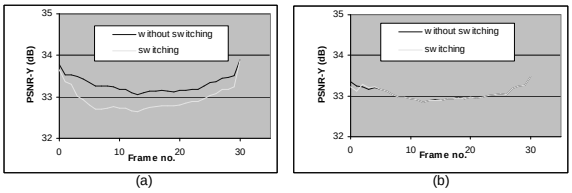


**Fig. 6**. Drifting experiment using Akko&Kayo view 28th: (a) CLP; (b) DSC. GOP size is 30 frames. Note that with DSC, the PSNR are almost the same in the switching and non-switching cases.

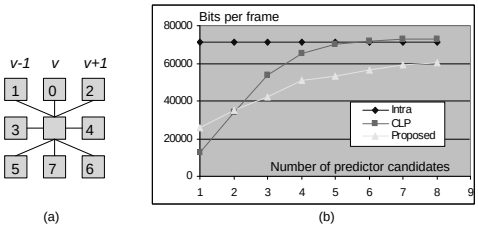


**Fig. 7**. Scaling experiment using Akko&Kayo view 29th. The PSNR of the different schemes are comparable – Intra: 35.07dB, CLP: 34.78dB, DSC:34.79dB.

## 5. REFERENCES


[1] ISO/IEC JTC1/SC29/WG11, "Introduction to multiview video coding," *MPEG document N7328*, July 2005.

[2] R.-S. Wang and Y. Wang, "Multiview video sequence analysis, compression, and virtual viewpoint synthesis," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 3, Apr. 2000.

[3] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, July 2003.

[4] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Information Theory*, vol. 19, pp. 471–480, July 1973.

[5] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. Data Compression Conference (DCC)*, 1999.

[6] N.-M. Cheung, H. Wang, and A. Ortega, "Video compression with flexible playback order based on distributed source coding," in *Proc. Visual Communications and Image Processing (VCIP)*, 2006.

[7] N.-M. Cheung and A. Ortega, "Flexible video decoding: A distributed source coding approach," in *Proc. Workshop on Multimedia Signal Processing (MMSP)*, 2007.

[8] X. Zhu, A. Aaron, and B. Girod, "Distributed compression for large camera arrays," in *Proc. Workshop on Statistical Signal Processing*, 2003.

[9] G. Toffetti, M. Tagliasacchi, M. Marcon, A. Sarti, S. Tubaro, and K. Ramchandran, "Image compression in a multi-camera system based on a distributed source coding approach," in *Proc. European Signal Processing Conference*, 2005.

[10] A. Jagmohan, A. Sehgal, and N. Ahuja, "Compressed of lightfield rendered images using coset codes," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2003.

[11] A. Aaron, P. Ramanathan, and B. Girod, "Wyner-Ziv coding of light fields for random access," in *Proc. Workshop on Multimedia Signal Processing (MMSP)*, 2004.

[12] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Free viewpoint switching in multi-view video streaming using Wyner-Ziv video coding," in *Proc. Visual Communications and Image Processing (VCIP)*, 2006.

[13] J. Wang, V. Prabhakaran, and K. Ramchandran, "Syndrome-based robust video transmission over networks with bursty losses," in *Proc. Int'l Conf. Image Processing (ICIP)*, 2006.

[14] P. Ramanathan and B. Girod, "Random access for compressed light fields using multiple representations," in *Proc. Workshop on Multimedia Signal Processing (MMSP)*, 2004.

[15] X. Guo, Y. Lu, W. Gao, and Q. Huang, "Viewpoint switching in multiview video streaming," in *Proc. Int'l Symp. Circuits and Systems*, 2005.

[16] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[17] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, Oct. 2002.