DISTRIBUTED SOURCE CODING TECHNIQUES FOR INTERACTIVE MULTIVIEW VIDEO STREAMING

Ngai-Man Cheung, Antonio Ortega

University of Southern California

Gene Cheung

Hewlett-Packard Laboratories Japan

ABSTRACT

We investigate coding tools for interactive multiview streaming (IMVS), where clients interactively request desired views for successive video frames, and in response the server sends the appropriate pre-compressed video data to the clients. Solution based on using only I-frames to support view switching would incur high transmission cost, while for that based on using only P-frames to encode every possible traversal, although it can minimize transmission cost, prohibitive server's storage may be required. Therefore, efficient solutions for IMVS need to consider the trade-off between transmission and storage cost. In this paper, we study the potential use of distributed source coding (DSC) in IMVS. Specifically, we propose two DSC constructions that could achieve good transmission-storage trade-offs. Central to these constructions is a method that can efficiently encode the least significant bits (LSB) of a frame to be decoded, leading to competitive storage and transmission requirements. Experiment results demonstrate these constructions compare favorably to existing tools, and could be valuable for interactive multiview streaming.

Index Terms— Multiview Video Streaming, Distributed Source Coding, Slepian-Wolf, Wyner-Ziv, SP-frames

1. INTRODUCTION

We consider the problem of *interactive multiview video streaming* (IMVS) [1, 2]: after one compressed representation of a multiview sequence is chosen at the server, streaming clients *interactively* request desired views for successive video frames in time. The encoding is done once at the server for a possibly large group of clients, and each client is allowed to switch to another view in every N_s frames. Thus there could be different traversals of views across time for different clients.

To provide intuition on IMVS, we first consider several structures that support interactive streaming with two views and $N_s = 1$, i.e., clients can switch view in every frame (see Figure 1(a), where a video picture $F_{i,j}$ at time instant i of view j is represented by a square)[1]. One approach encodes all pictures of all views as I-frames, so that the server can simply send the requested I-frame with no concern for inter-frame dependencies (Figure 1(b))[2]. A drawback is that a high bit-rate I-frame is sent in each request, incurring high transmission cost. Another approach uses a starting I-frame plus successive P-frames to encode every possible frame traversal in time by the clients (Figure 1(c)). Note that with such approach, each picture of each view is encoded into multiple frames (*replicas*) to avoid coding drift (assuming only I- and P-frames are used), and we denote P-frame replicas by $P_{i,j}^{(k)}$. While this results in minimum transmission cost, the storage required can be prohibitive, as the number of P-frame replicas increases exponentially with the number of pictures.

Clearly, we can achieve more practical representations between these two extremes by optimally trading off transmission and storage costs under a given set of constraints. For example, one may insert I-frames in the previously discussed P-frames only solution in order to reduce overall storage (Figure 1(d)), at the expense of some increase in the transmission cost [1, 2]. Essentially, this reduces storage because any request for the corresponding picture can be satisfied with this I-frame regardless of the decoding path that was followed, so there is no need to have multiple replicas for different decoding paths. Alternatively, H.264 SP-frames [3] could be used together with P-frames to construct solution which has practical storage cost. This works because SP-frames eliminate mismatches in multiple decoding paths and allow identical reconstructions from different traversals (see Figure 1(e), where an encoded SP-frame of $F_{i,j}$, $S_{i,j}^{(k)}$, is denoted by a triangle). Therefore, there is no need to have different P-frame replicas for different SP reconstructions, resulting in storage cost reduction (compare Figures 1(c) and 1(e) at time instant 4). Compared to I-frames, SP-frames are more efficient in transmission as each SP-frame is smaller. However, SP-frames would require more storage than I-frames, since multiple SP-frames would need to be stored, with each SP predicted from reconstruction corresponding to one of the decoding paths. In summary, both I-frames and SP-frames can be used to address mismatches in multiple decoding paths and construct practical IMVS solutions, but with different trade-offs between transmission and storage.

In this paper, we consider the potential use of distributed source coding (DSC) [4] for interactive multiview streaming. Specifically, we propose two DSC constructions that can be used as alternatives to I-frames or SP-frames to eliminate mismatches in multiple decoding paths, leading to practical solutions for IMVS. In particular, these constructions have different operating trade-offs: one (DSC1) is competitive in terms of transmission cost, while the other (DSC2) is superior in terms of storage cost. As will be discussed in detail, DSC1 and DSC2 compare favorably to SP-frames in terms of storage requirement. Therefore, both of them can be valuable tools under different server storage constraints. Moreover, DSC2 can outperform I-frames w.r.t. both transmission and storage costs, and thus can be used to replace I-frames in the IMVS problem.

Central to our proposed constructions is a method that can convey efficiently the least significant bits (LSB) of a frame to be decoded, based on DSC, so that mismatches can be eliminated and identical constructions can be achieved starting from different "side information", i.e., different decoding paths [5]. Consider the SP-frames solution discussed previously (Figure 1(e)). In this case, one would use a primary SP-frame for one decoding path and *multiple*

[1] As will be clear our proposed tools are applicable for any switching period N_s. However, the overall improvement tends to be significant when switching points are frequent.

[2] We use the term *picture* to denote the original image and *frame* the encoded version of a picture.

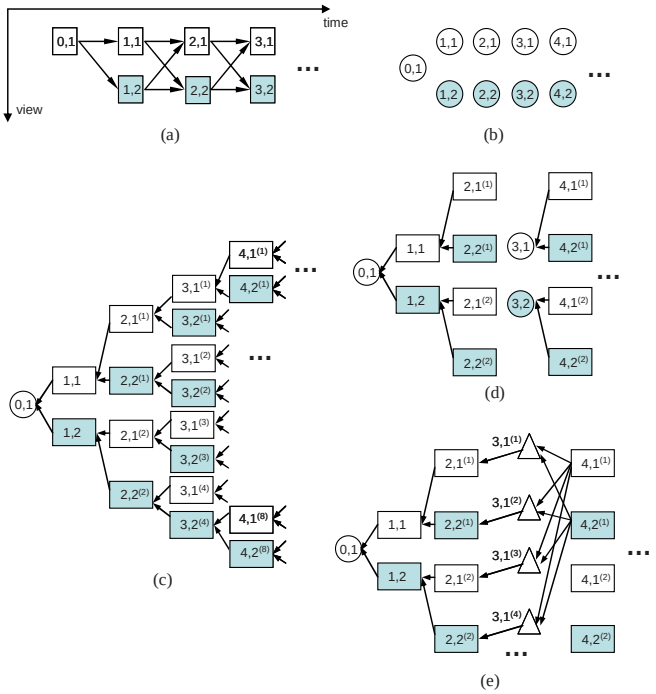

Fig. 1. (a) Interactive multiview streaming example: a client may switch between views 1 and 2. (b) Solution using all I-frames (each represented by a circle). (c) Solution with an initial I-frame and all P-frames (rectangles). Here an original picture $F_{i,j}$ may be encoded into multiple replicas in order to avoid drifting. We denote P-frame replicas by $P_{i,j}^{(k)}$. E.g., $P_{3,1}^{(1)}, ..., P_{3,1}^{(4)}$ are P-frame replicas of $F_{3,1}$ in the figure. (d) Solution using P-frames and I-frames. (e) Solution including SP-frames (triangles). We denote an encoded SP-frame of $F_{i,j}$ by $S_{i,j}^{(k)}$. One of the SP-frames would be a primary SP (say $S_{3,1}^{(1)}$) and the others would be secondary SP-frames ($S_{3,1}^{(2)}, ..., S_{3,1}^{(4)}$). Since the reconstructions from $S_{3,1}^{(1)}, ..., S_{3,1}^{(4)}$ are identical, any of them can be used as reference for $P_{4,1}^{(1)}$ here.

secondary SP-frames, one for each of the other decoding paths. In order for the secondary SP reconstructions to match the primary SP reconstruction exactly, each secondary SP-frame would need to losslessly encode the LSBs of the identical reconstruction[3]. In other words, the SP-frames solution would need to losslessly encode the expensive LSB *multiple times*, requiring considerable storage. In contrast, based on our previously proposed method [5] with DSC these LSB would need to be encoded *only once*, resulting in considerable improvement in storage requirement as compared to SP-frames.

Previous research on multiview video focuses mainly on designing advanced motion/disparity-compensated coding techniques to encode all frames in a rate-distortion optimal manner, e.g., [6, 7]. Interactive multiview streaming based on I- and P-frames was studied in [1, 2]. Previous work on DSC-based multiview image/video coding mainly focuses on achieving distributed, independent encoding of different views at individual spatially-separated camera or video sensor, e.g., [8]. Application of DSC to enable identical reconstruction in lightfield compression was studied in [9] and [10]. As will be discussed in Section 2, the work in [9] does not address offline compression, while the work in [10] requires storing multiple sets of parity for encoding LSB resulting in a storage cost comparable to SP-frame. Our previous work [5] applies DSC to address view switching for applications where feedback from users to encoders is not available. The current work extends [5] to address interactive multiview streaming. In particular, we demonstrate DSC can be an efficient tool to eliminate mismatches, and propose two configurations to achieve different transmission-storage trade-offs.

This paper is organized as follows. In Section 2, we briefly review a method to eliminate mismatches in multiple decoding paths, and compare the work in [9, 10]. Section 3 presents the two proposed DSC configurations. Section 4 presents the experimental results and Section 5 concludes the work.

2. DSC-BASED METHOD TO ELIMINATE MISMATCH

We briefly review a method to use DSC in order to achieve identical reconstruction from multiple predictor candidates [5]. In a conventional closed-loop predictive (CLP) system (e.g. those based on H.264), the encoder computes a prediction residual $Z = X - Y$, between source X and predictor Y, and communicates Z to the decoder. DSC approaches the same compression problem by viewing X as an input to a virtual channel with *correlation noise* Z, and Y as the output of the channel. Therefore, to recover X from Y, encoder would send *parity information* to the decoder. Significantly, this parity information, which is computed entirely from X taking into account the statistics of Z, is independent of a specific Y being observed, and X can be *exactly* reconstructed as long as a sufficient amount of parity information has been communicated.

The framework can be extended to address the mismatches in multiple decoding paths as follows. We consider N virtual channels, each corresponding to a predictor candidate Y_n obtained from one of the possible decoding paths. Each channel is characterized by the correlation noise $Z_n = X - Y_n$. In order to recover X exactly from any of these channels, the encoder would need to send an amount of parity sufficient for *all* the channels. That is, the encoder would transmit enough parity information to allow decoding of the *worst-case* Z_n. By doing so, X can be exactly recovered from any of the Y_n. Note that in our problem formulation since the encoder has access to X and all Y_n, the statistics of Z_n and hence the amount of parity information can be readily determined [5].

We now compare this approach with those proposed by [9] and [10] applied to enable identical reconstruction in lightfield compression. [9] proposed to eliminate mismatches by having the client continuously requesting parity during the streaming session and server generating these parity on-line. The amount of requested parity depends on which Y_n is available at the client. This approach is intended for dealing with a large number of different Y_n (in their application Y_n are images rendered from different combinations of adjacent reconstructed images). While this could be modified for pre-compression it would require storing multiple sets of parity for each possible Y_n incurring considerable storage cost. In [10] multiple sets of parity information were generated ahead of time. During the streaming session, the server communicates the appropriate set to the client depending on the available Y_n so that identical reconstruction can always be achieved from any Y_n. This is done so that reconstructions are independent of the viewer's trajectory of the lightfield data. A drawback of this scheme is that this would require storing multiple sets of parity bits. In particular, since the parity bits encode the LSB of X, the storage cost of multiple sets of parity could be non-trivial (this scheme can be considered as the counter-part of SP-frames). In contrast, in our approach we store a single set of parity capable to recover X from any of the Y_n, leading to a better storage cost, but incurring some inefficiency in transmission when, for

[3]For this reason secondary SP-frames tend to be considerably larger than P-frames [3], e.g., more that $3\times$ in this setting as suggested by our results.

some Y_n, it would be possible to recover X without transmitting the amount of parity information required in the worst case. However, such inefficiency could be small in our problem since, as will be discussed, some Y_n are indeed different replicas of the *same* frame at similar quality. Therefore, the degree of correlation between X and any of these Y_n is similar.

3. PROPOSED CODING CONSTRUCTIONS

3.1. DSC1

In this section we discuss two constructions that can be used to support IMVS. In DSC1, we encode $F_{i,j}$ into multiple *intermediate* P-frames, $P_{i,j}^{(k)}$, each predicted from the reconstruction of a different decoding path. In addition, we generate a *common* DSC coded frame $W_{i,j}$ which can be used to reconstruct $F_{i,j}$ using *any* of the $P_{i,j}^{(k)}$ as side information (Figure 2(a)). Note that in general $P_{i,j}^{(k)}$ leads to different reconstruction of picture $F_{i,j}$ (decoding path mismatch), and thus $W_{i,j}$ ensures that mismatch is eliminated by leading to an identical target reconstructed frame $\widehat{F}_{i,j}$ regardless of the decoding path. Following the discussion in Section 2, $W_{i,j}$ and $P_{i,j}^{(k)}$ correspond to the parity bits and side information Y_n, respectively, in the DSC framework, and $W_{i,j}$ would be generated according to the worst case correlation between $P_{i,j}^{(k)}$ and $\widehat{F}_{i,j}$. Assume N is the number of decoding paths leading to $F_{i,j}$. With this approach N intermediate P-frames $P_{i,j}^{(k)}$ and a single $W_{i,j}$ would need to be stored in the server. During the streaming session, only one of the N intermediate P-frames (depending on the decoding path taken by the client) and $W_{i,j}$ would be sent to the client. As $\widehat{F}_{i,j}$ could be identically reconstructed from any of the $P_{i,j}^{(k)}$, mismatch can be eliminated at instant i, and $\widehat{F}_{i,j}$ would serve as the merging point of multiple decoding paths. Note that it is possible to encode the intermediate P-frames at different qualities. In the experiment we choose to encode $P_{i,j}^{(k)}$ at a quality similar to $\widehat{F}_{i,j}$, which we found empirically would lead to the best performance. As $P_{i,j}^{(k)}$ and $\widehat{F}_{i,j}$ are very similar ($P_{i,j}^{(k)}$ and $\widehat{F}_{i,j}$ are different reconstructions of $F_{i,j}$ with similar quality), the bitrate of $W_{i,j}$ tends to be small.

Similar to SP-frames, DSC1 is efficient in transmission but expensive in storage. Specifically, in DSC1, an intermediate P-frame and a small $W_{i,j}$ would be sent to the client upon request, and experiment results suggest that such bitrate could be comparable to a SP-frame. On the other hand, the storage cost for $F_{i,j}$ using DSC1, B_{DSC1}, is[4]:

$$B_{DSC1} = H(W_{i,j}) + \sum_{k=1}^{N} H(P_{i,j}^{(k)}). \quad (1)$$

Therefore, DSC1 has a storage cost that scales with the number of decoding paths, which is similar to SP-frames. Nevertheless, DSC1 could compare favorably to SP-frames in terms of storage cost. This is because SP-frames approach encodes the LSB of $\widehat{F}_{i,j}$ $N-1$ times, while DSC1 encodes these LSB only once in $W_{i,j}$ with the N intermediate P-frames generated by efficient lossy coding.

3.2. DSC2

In DSC2 (Figure 2(b)), we encode $F_{i,j}$ directly as $W_{i,j}$ based on DSC techniques, using reconstructed frames at instant $i-1$ as side information. Specifically, $W_{i,j}$ would include an amount of parity information that can accommodate the worst case correlation between $\widehat{F}_{i,j}$ and $P_{i-1,j'}^{(k)}$ for any view j' that has non-zero transition probability to view j at instant i. Therefore, $P_{i-1,j'}^{(k)}$ corresponds to Y_n in Section 2. By sending $W_{i,j}$, $\widehat{F}_{i,j}$ can be identically reconstructed from any of these $P_{i-1,j'}^{(k)}$.

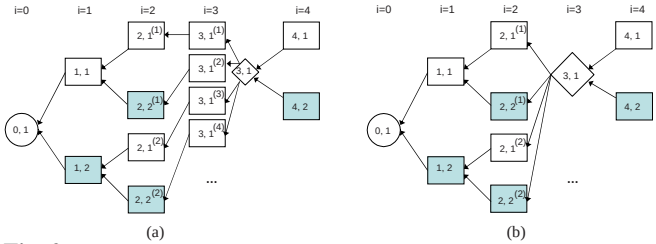

Fig. 2. DSC Configurations for IMVS: (a) DSC1; (b) DSC2. Here a diamond represents parity information $W_{i,j}$.

Note that with DSC2 a single $W_{i,j}$ would be stored in the server regardless of the number of decoding paths, and the same $W_{i,j}$ would be sent when a client requests $F_{i,j}$ from any $P_{i-1,j'}^{(k)}$. In general, DSC2 would compare favorably to SP frames in terms of storage cost, for the following arguments. The storage cost of $F_{i,j}$ using SP-frames, B_{SP}, would depend on the number of replicas, since multiple SP-frames would need to be stored, one for each of the possible replicas, $P_{i-1,j'}^{(k)}$. Specifically, $B_{SP} = \sum_{j'} \sum_k H(\widehat{F}_{i,j}|P_{i-1,j'}^{(k)})$. In contrast, the frame size of $W_{i,j}$ would depend on the worst case correlation between $\widehat{F}_{i,j}$ and $P_{i-1,j'}^{(k)}$:

$$B_{DSC2} = \max_{j',k} H(\widehat{F}_{i,j}|P_{i-1,j'}^{(k)}) \approx \max_{j'} H(\widehat{F}_{i,j}|P_{i-1,j'}^{(1)}). \quad (2)$$

The approximation holds since different replicas of $F_{i-1,j'}$ are of similar quality. (2) suggests that B_{DSC2} depends on only inter-view correlation and is somewhat independent of the number of replicas. Comparing B_{DSC2} with B_{SP}, one may conclude that DSC2 would tend to have a smaller storage cost in scenarios with a moderate number of replicas.

DSC2 is similar to I-frames in that in multiview streaming a single I-frame $I_{i,j}$ would be kept in the server regardless of the number of decoding paths, and the same $I_{i,j}$ would be sent to the client for any request of $F_{i,j}$. However, since $W_{i,j}$ exploits inter-view correlation and would require only a rate close to the conditional entropy (given by (2)), it would compare favorably to $I_{i,j}$ in both storage and transmission cost, since $I_{i,j}$ would need a rate close to $H(\widehat{F}_{i,j})$. Such gain would depend on inter-view correlation. In particular, if switching is only possible between *adjacent* views (which are highly correlated) DSC2 can outperform I-frames by a significant margin, as demonstrated in the experiments[5].

4. EXPERIMENTAL RESULTS

We assume, at each switching point, with probability $1-2\alpha$ clients would remain in the same view (i.e., transition from $F_{i,j}$ to $F_{i+1,j}$), and with probability α switch to the adjacent views (i.e., transition from $F_{i,j}$ to $F_{i+1,j-1}$ or $F_{i+1,j+1}$). We consider IMVS solutions where in the P-frames only structure the following types of frames are inserted periodically: [i] I-frame (Figure 1(d)), [ii] DSC1 (Figure 2(a)), [iii] DSC2 (Figure 2(b)), [iv] SP (Figure 1(e)), and [v] SP+SI. In [iv] and [v] the (smaller) primary SP is used for the most probable request. We assume clients may switch in every frame, i.e., $N_s = 1$. Since we compare only the bitrates at the switching points

[4]We use $H(X)$ to denote the rate to convey X and $H(X|Y)$ the rate to convey X with Y as predictor.

[5]Essentially, I-frame allows switching between any pair of views. If such functionality is not necessary (e.g., only switchings between neighboring views are required), then DSC2 can be a better solution.

the results are mostly independent of N_s. In these experiments we use the DSC encoding algorithm proposed in [5] to generate $W_{i,j}$ in DSC1 and DSC2. The algorithm in [5] is modified such that the same set of motion vectors is used for decoding from all replicas of a given frame. This can be justified as different replicas of $F_{i-1,j'}$ would have similar motion/disparity w.r.t. current frame $F_{i,j}$. Moreover, since the DSC system uses H.263 coding tools (e.g., half-pixel motion estimation (ME)) we compare the DSC results with H.263 encoded I-frames and P-frames. For SP or SP+SI system (which is based on H.264) we use only 16×16 motion compensation so that a more fair comparison with DSC1/DSC2 can be made. However, the SP or SP+SI system is still using some advanced coding tools (e.g. quarter-pixel ME, CAVLC) compared to DSC1/DSC2 systems. We use MVC sequences Akko&Kayo and Ballroom in the experiment, which are in 320×240 and encoded at 30fps and 25fps respectively. Figure 3 compare the storage costs of different solutions. As discussed and illustrated in (2), B_{DSC2} would be somewhat independent of the number of decoding paths. This can be clearly seen in the figure. Figure 3 also suggests that DSC2 can outperform other solutions with moderate number of decoding paths. In addition, DSC1 can outperform SP-frames, since with each additional decoding path, B_{DSC1} would increase by an amount equivalent to the size of P-frame, while B_{SP} would increase by the size of secondary SP-frame.

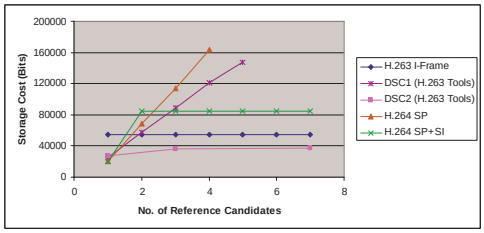

Fig. 3. Storage costs comparison (Akko&Kayo). The x-axis represents the number of decoding paths, and the y-axis represents the average number of bits to encode a picture $F_{i,j}$, e.g., B_{DSC1} for DSC1.

Figure 4 compares the average transmission cost of DSC1 and SP (it is clear that other solutions would have worse transmission costs)[6]. Figure 4 shows that, with $\alpha = 0.2$, DSC1 and SP are comparable w.r.t. transmission cost. Note that sum bitrate of an intermediate P-frame and parity bits in DSC1 is larger than that of a primary SP-frame, but less than that of a secondary SP. Therefore, DSC1 could outperform SP w.r.t. transmission cost when view switching is more likely (i.e., when α is large), and vice versa when α is small. Figure 5 compares the Lagrangian costs ($C + \lambda B$, where C is the average transmission cost, B the storage cost, and λ the Lagrange multiplier; thus we consider the problem of minimizing the average transmission cost given some storage constraint). As DSC1 is excellent for transmission while DSC2 is efficient for storage, we expect DSC1 to outperform DSC2 when the storage constraint is loose (small λ), and the opposite when storage constraint is tight (large λ). Therefore, both DSC1 and DSC2 can be used in a streaming system depending on different storage constraints. Figure 5 also suggests that for a sufficiently large α DSC solutions would outperform all other schemes. In particular, DSC1 can be a better choice over SP. It is because while the required transmission costs could be comparable, DSC1 would require less server's storage. Our recent results [11] further demonstrate DSC1 and DSC2 can be combined with existing tools (I- and P- frames) to achieve considerable overall improvement.

[6]The average (expected) transmission cost is the weighted sum of the frame sizes in the structure, with the weighting factors being the probability of sending those frames in response to clients' requests. E.g., in Figure 1(e) server may send one of the $S_{3,1}^{(1)}, ..., S_{3,1}^{(4)}$ to respond to a request of $F_{3,1}$, with different probability, since clients may arrive at $P_{2,1}^{(1)}, ..., P_{2,2}^{(2)}$ with different probability (depending on α).

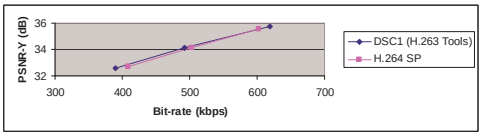

Fig. 4. Transmission costs comparison (Ballroom), with $\alpha = 0.2$.

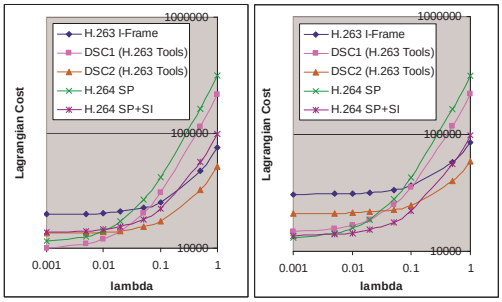

Fig. 5. Lagrangian costs comparison (Akko&Kayo): (a) $\alpha = 0.2$; (b) $\alpha = 0.1$. The smaller is the Lagrangian cost the better is the trade-off.

5. CONCLUSIONS

In this paper we proposed two DSC configurations DSC1 and DSC2 to facilitate interactive multiview streaming. As DSC1 is excellent for transmission while DSC2 is superior for storage, both of them can be used depending on different storage constraints. Experimental results suggest for sufficiently large switching probability DSC solutions can outperform all other schemes. Therefore, DSC1 and DSC2 could be valuable tools in IMVS and in other streaming system [12].

6. REFERENCES

[1] G. Cheung, A. Ortega, and T. Sakamoto, "Coding structure optimization for interactive multiview streaming in virtual world observation," in *IEEE International Workshop on Multimedia Signal Processing*, October 2008.

[2] G. Cheung, A. Ortega, and N.-M. Cheung, "Generation of redundant coding structure for interactive multiview streaming," in *Proc. Int'l Packet Video Workshop*, 2009.

[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.K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Information Theory*, vol. 19, pp. 471–480, July 1973.

[5] N.-M. Cheung and A. Ortega, "Distributed source coding application to low-delay free viewpoint switching in multiview video compression," in *Proc. of Picture Coding Symposium, PCS'07*, Lisbon, Portugal, Nov. 2007.

[6] E. Martinian, A. Behrens, Jun Xin, A. Vetro, and Huifang Sun, "Extensions of H.264/AVC for multiview video compression," in *Proc. Int'l Conf. Image Processing (ICIP)*, 2006.

[7] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," in *IEEE Transactions on Circuits and Systems for Video Technology*, November 2007, vol. 17, no.11, pp. 1474–1484.

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

[9] 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.

[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] G. Cheung, N.-M. Cheung, and A. Ortega, "Optimized frame structure using distributed source coding for interactive multiview video streaming," submitted to *IEEE ICIP*, Feb 2009.

[12] T. Fujii and M. Tanimoto, "Free viewpoint TV system based on ray-space representation," in *Proceedings of SPIE*. SPIE, 2002, vol. 4864, p. 175.