OPTIMIZED FRAME STRUCTURE USING DISTRIBUTED SOURCE CODING FOR INTERACTIVE MULTIVIEW VIDEO STREAMING

Gene Cheung

Hewlett-Packard Laboratories Japan
3-8-13, Higashi-Takaido, Suginami-ku
Tokyo, Japan 168-0072

Ngai-Man Cheung, Antonio Ortega

University of Southern California
Signal and Image Processing Institute
Los Angeles, CA 90089-2564

ABSTRACT

While multiview video coding typically focuses on the rate-distortion performance of compressing all frames of all views, we address the problem of designing a pre-encoded frame structure for a streaming server to enable a new functionality—interactive multiview switching, where a streaming client can send requests periodically to a server to switch to different views while continuing uninterrupted temporal playback of streaming video. We observe that providing bandwidth-efficient interactive view switching usually comes at the price of additional overall storage. Thus, our goal is to find a frame structure that minimizes the expected transmission rate during interactive multiview streaming, subject to a storage constraint. Noting that standard tools for random access (i.e., I-frame insertion) can be bandwidth-inefficient for this functionality, we propose to automatically generate a structure, combining I-frames, redundant P-frames and Distributed Source Coded (DSC) frames, in a near-optimal fashion to facilitate view switching. We present three new DSC techniques for view switching and discuss how these techniques can be integrated into an optimization framework. We show experimentally that near-optimal coding structures using DSC frames, in addition to I- and P-frames, reduce transmission cost over structures using I-frames only for view switching by up to 28%, and over structures using I- and P-frames only by up to 20% for the same storage cost.

1. INTRODUCTION

Multiview video consists of sequences of spatially related pictures captured simultaneously and periodically by multiple closely spaced cameras. Much of previous research on multiview video focuses on compression, where the goal is to design novel motion- and disparity-compensated coding techniques to encode all frames of all views in a sequence in a rate-distortion optimal manner [1, 2].

In this paper, we focus instead on the problem of *interactive multiview video streaming* (IMVS): designing a pre-encoded frame representation of a multiview sequence for a streaming server, so that streaming clients can periodically request desired views for successive video frames in time. More precisely, each client can watch and request one single view at a time out of possibly many available views, while continuing uninterrupted temporal playback of streaming video. We call this new media interaction functionality *interactive multiview switching*. Thus our work can be seen as addressing practical coding issues that may be encountered in designing a streaming FTV system [3] using standard video coding tools.

Note that the requested data corresponds to only a small subset out of a large set of available multiview data at the server. Each client of a possibly large group can navigate the content by playing it back (in time) while switching views, thus resulting in different traversals of views across time for each client. Our goal is to provide a desired level of view interactivity with minimum expected transmission bandwidth cost. The extent of view interactivity is determined by the *view switching period* M, i.e., view switching can only take place at frames whose time indices are multiples of M.

A natural approach to enable this kind of interactive view switching is to make use of standard random access tools, e.g., making every M-th frame (in all views) an I-frame. The key observation in our work is that *random access and view switching are fundamentally different functionalities*, and thus efficient tools for one problem may not provide the best solution for the other. For random access to a frame, one can make no assumptions about which frames are available at the decoder, and hence independently coded I-frames are well suited for this purpose. View switching, on the other hand, arises when temporal playback is not interrupted; i.e., successive frames are displayed, but one wishes to switch point of view. Therefore, the decoder has access to some of the frames immediately preceding the requested frame in time (albeit from a different view). Since consecutive frames in different views tend to be correlated, using an independently coded I-frame for switching is sub-optimal in terms of bandwidth usage.

The main focus of our work is then to study alternatives for view switching that are more bandwidth-efficient than simple I-frame insertions. Note that our proposed tools *do not* support random access, and thus we are not advocating using these tools *instead* of random access tools such as I-frames. Rather, we argue that view switching and random access are fundamentally two different functionalities, supported with different tools. It will be up to the system designer to select the appropriate setting for a given application: one may select a parameter M for view switching and separately allow random access at every M'-th frame, where typically $M \ll M'$.

For view switching, we know that only one of a *few* previous frames could have been decoded. Hence it is possible to use differential coding tools like P-frames to exploit inter-frame correlation. In our previous work [4, 5], we found that doing so means involving a tradeoff between the expected transmission rate and the storage space required to store the multiview representation that enables interactive view switching. For intuition, consider the $M = 1$ case, i.e., enabling view switching any time. We will henceforth use "*frame*" to denote a specific coded version of a picture, and use "*picture*" for the corresponding original frame. One extreme case would be to encode all pictures of all views as I-frames, so that the server can simply send the corresponding I-frame for each requested picture with no concern for inter-frame dependencies. This leads to high bandwidth usage in a IMVS session due to aforementioned coding inefficiency of I-frames.

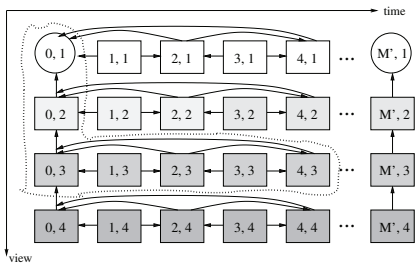


Fig. 1. A 4-view, 5-instant Example of Multiview Frame Structure using Inter-view Prediction for Key Frames in [1].

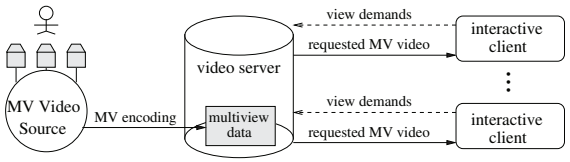


Fig. 2. Interactive Multiview Video Streaming System

As an alternative, if we allow *redundant P-frames*—multiple P-frames representing the same picture, each differentially coded using a different predicted frame—then we can use a starting I-frame plus successive P-frames to encode every possible frame traversal in time by the client. (Encoding different versions of a picture is needed to eliminate decoding drift.) While this results in minimum transmission cost, the storage required is prohibitive.

Clearly, more practical multiview representations compose a mixture of I- and P-frames that lie between these two extremes and optimally trade off transmission and storage costs. In particular, an I-frame (large transmission cost, small storage cost) can situationally replace a set of corresponding P-frames representing the same picture in a frame structure, if the P-frame set is sufficiently large (small transmission cost, large storage cost). Our previous work [4, 5] leveraged on this observation to optimize frame structures by selecting I- and redundant P-frames appropriately.

In this paper, we extend our frame structure optimization to include frames encoded based on distributed source coding (DSC). Applying DSC to facilitate view switching was proposed in our previous work [6]. In particular, the work proposed efficient DSC-based coding algorithms that can lead to identical reconstruction from references of different decoding paths, each playing the role of "side information". The DSC frames thus provide similar functionality in multiview video as H.264 SP-frames [7]. [6] demonstrated these DSC coded frames compare favorably to existing coding tools like SP-frames on a frame-by-frame basis in terms of RD performance. The focus of this paper, however, is to investigate how these DSC coded frames can be appropriately combined with I-frames and P-frames to create efficient overall frame structures that minimize the average transmission cost of every possible traversal, subject to an overall storage constraint. We derive the Lagrangian costs of the DSC constructions, and discuss how DSC frames can complement I- and P-frames. Moreover, we demonstrate experimentally the usefulness of DSC in a practical multiview streaming scenario.

The outline of the paper is as follows. We first discuss related work in Section 2. We discuss our IMVS optimization framework in Section 3. We then propose three techniques to use DSC in IMVS in Section 4. We present our experimental results in Section 5.

2. RELATED WORK

As mentioned, much of the previous research in multiview video has focused on efficient compression of all frames and all views using motion- and disparity-compensated techniques [1, 2]. As in single-view video, I-frames can be periodically inserted, say one for every M'-frame interval, to permit some desired level of temporal random access. Consider as an example the frame structure proposed in [1] and shown in Fig. 1, where every M'-th frame becomes a "key frame", providing temporal random access.

In order to facilitate interactive view switching for a desired M frame period, the simplest strategy is to allow more random access I-frame insertions by setting $M' = M$. Clearly, for small M this leads to high bandwidth usage, which is not desirable.

An alternative strategy is to select a compression-optimized frame structure (e.g., Fig. 1) with $M' \gg M$, and send to the decoder all necessary frames for a specific view switching request (all frames needed to reconstruct the requested frame). For example, in Fig. 1, in order to switch from frame (3,3) to frame (4,2), a server would send frames (1,2) through (4,2) to the decoder, but only frame (4,2) would be displayed. We call this strategy *rerouting*[1]. Besides the increased client complexity to decode multiple frames just to display one frame, rerouting causes a spike in transmission rate during a view switch which may also be undesirable.

Thus, in structures like Fig. 1, coding efficiency (total storage) is the main design criterion but this leads to bandwidth inefficiencies in view switching situations. Our IMVS formulation differs in that we explicitly consider target bandwidth needs for view interactivity, at the expense of a modest and controlled increase in storage.

Study of the conflicting requirements of interactivity and compression is not new, and solutions have been proposed in the context of light fields [8, 9, 10] using DSC, SP-frames, and aforementioned rerouting, respectively. Our IMVS work differs in that we construct a varying number of decoded versions of a single original picture (at the expense of increased in storage), so that transmission rate can be gracefully reduced if more storage becomes available.

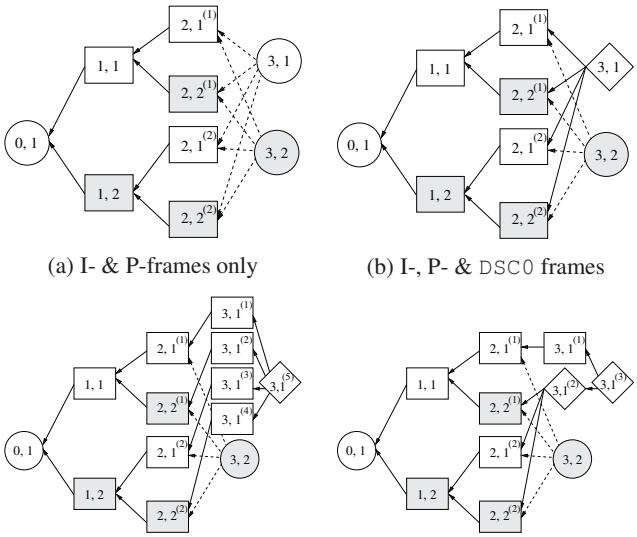
We formally posed the IMVS problem as a combinatorial optimization and proved its NP-hardness in [4]; subsequently we derived a near-optimal optimization algorithm to find good coding structures for IMVS in [5] using only I- and P-frames. In this paper we integrate DSC techniques we have previously proposed [6] for view switching into our optimization framework, and we show experimentally DSC's merits in the IMVS context.

3. INTERACTIVE MULTIVIEW VIDEO STREAMING

3.1. System Model

The system model we consider for IMVS is shown in Fig. 2. A *Multiview Video Source* simultaneously captures multiple pictures of different views at regular intervals. A *Video Server* sequentially grabs the captured uncompressed pictures in windows of M' pictures at a time from MV Video Source and pre-encodes each window into an optimized frame structure $\mathcal{T}$. Using a single (albeit redundant) frame structure $\mathcal{T}$, the server can serve multiple streaming clients without further encoding or transcoding. An alternative approach of live encoding a path traversal tailor-made for each streaming client's interactivity is computationally prohibitive for large number of clients.

[1] We studied the design of frame structures when limited rerouting is permitted in [5]. In this paper, however, we consider only the no-rerouting case.



(a) I- & P-frames only

(b) I-, P- & DSC0 frames

(c) I-, P- & DSC1 frames

(d) I-, P-, DSC0+1 frames

Fig. 3. IMVS structures for two views (two shades). I-, P- and DSC frames are denoted as circles, squares, and diamonds respectively.

3.2. View Interaction Model

We assume a view interaction model where, upon watching a decoded version of picture $F^o_{i,j}$ of time instant i and view j, a client will request a version of picture $F^o_{i+1,k}$ of view k and instant $i+1$, where $j-1 \le k \le j+1$, with known *view transition probability* $\alpha_{i,j}(k)$. Recall that our system has redundant storage; i.e., there are multiple frames representing a given picture. Any decodeable frame $F_{i+1,k}$ can satisfy a request for picture $F^o_{i+1,k}$.

We assume that clients will only play back video content forward in time while switching views interactively; we call this interactivity *forward view switching*[2]. Another possible interactivity for multiview video is to freeze video in time and switch view (*static view switching*); we conjecture that this interactivity can be efficiently supported by novel usage of DSC and is left for future work.

If view switching at every M-th frame is desired, it can be easily modeled with our interaction model, where a frame node $F_{i,j}$ in a graph like Fig. 3(a) abstractly represents M consecutive actual encoded frames of the same view j (a carefully chosen I-, P- or DSC frame followed by $M-1$ consecutive P-frames of the same view).

3.3. IMVS Optimization

We first describe how a server uses a frame structure $\mathcal{T}$ of I- and P-frames during IMVS. We represent a structure as a graph; an example is shown in Fig. 3(a), where a P-frame $F_{i,j}$ points (solid edges) to a previous frame $F_{i-1,k}$ using which it is differentially coded[3]. A frame-to-frame schedule dictates which frame $F_{i+1,k}$ (out of possibly many encoded versions) server should send to the client if view k is selected by client after $F_{i,j}$. We assume no rerouting is permitted in this paper, in which case frame scheduling is straightforward: $F_{i,j}$ switches to a P-frame $P_{i+1,k}$ differentially coded using $F_{i,j}$ if it exists, or to an I-frame $I_{i+1,k}$ otherwise (dotted edge in Fig. 3(a)).

[2]All video streaming systems today offer forward playback of video, hence forward view switching is a natural extension.

[3]Our graphical model is sufficiently general to represent any differential coding scheme used, including motion- and/or disparity-compensation.

A client displays a frame $F_{i,j}$ with *display probability* $q(F_{i,j})$; we assume starting frame F_{0,v^o} of view v^o has probability $q(F_{0,v^o}) = 1$. Given the view interaction model, a P-frame $P_{i+1,k}$ differentially coded from $F_{i,j}$ has display probability $q(F_{i,j})\alpha_{i,j}(k)$. On the other hand, an I-frame can be reached from multiple frames in previous instant. Thus, the display probability of an I-frame is the sum of all transition probabilities $q(F_{i,j})\alpha_{i,j}(k)$'s from these frames.

We now summarize the optimization in [5], which generates a near-optimal frame structure using I- and P-frames only. The Lagrangian cost of a structure $\mathcal{T}$ is the expected transmission cost plus multiplier λ times the storage cost. We can find the optimal structure front-to-back recursively: given a structure $\mathcal{T}_{i-1}$ built up to instant $i-1$, the minimum Lagrangian cost $L_i(\mathcal{T}_{i-1})$ for instant i onwards is the sum of *local Lagrangian costs* at instant i of all views j's, $l_{i,j}(\mathcal{T}_{i-1}, \mathbf{t}_{i,j})$'s, plus future recursive cost $L_{i+1}(\mathcal{T}_i)$:

$$L_i(\mathcal{T}_{i-1}) = \min_{\mathbf{t}_{i,j}} \left\{ \sum_{j=1}^{K} l_{i,j}\left(\mathcal{T}_{i-1}, \mathbf{t}_{i,j}\right) + L_{i+1}(\mathcal{T}_i) \right\} \quad (1)$$

where $\mathbf{t}_{i,j}$ is the structure for instant i and view j. $\mathcal{T}_i$ is simply the combined structure of $\mathcal{T}_{i-1}$ and $\mathbf{t}_{i,j}$'s for all j's. $\mathbf{t}_{i,j}$'s determine the local Lagrangian costs $l_{i,j}$'s. Hence we aim to select the optimal structure $\mathbf{t}_{i,j}$'s in order to minimize Lagrangian cost; [5] discussed computation-efficient methods to find near-optimal structures $\mathbf{t}_{i,j}$'s.

For each view j, the local Lagrangian cost $l_{i,j}$ is the sum of *local I-frame cost* $l^I_{i,j}$ and *local P-frame cost* $l^P_{i,j}$:

$$l_{i,j}(\mathcal{T}_{i-1}, \mathbf{t}_{i,j}) = l^I_{i,j}(\mathcal{T}_{i-1}, \mathbf{t}^I_{i,j}) + l^P_{i,j}(\mathcal{T}_{i-1}, \mathbf{t}^P_{i,j}) \quad (2)$$

I-frame cost $l^I_{i,j}$ is the sum of all transition probabilities into the I-frame plus λ times the size of the single I-frame $r^I_{i,j}$. Assuming there is at least one transition from $F_{i-1,k}$ to I-frame $I_{i,j}$, then:

$$l^I_{i,j}(\mathcal{T}_{i-1}, \mathbf{t}^I_{i,j}) = \left[\sum_{F_{i-1,k} \leftarrow \mathbf{t}^I_{i,j}} q(F_{i-1,k})\alpha_{i-1,k}(j) + \lambda \right] r^I_{i,j} \quad (3)$$

P-frame cost $l^P_{i,j}$, on the other hand, is the sum of individual transition probability plus λ times size of P-frame $r^P_{i,j}(k)$:

$$l^P_{i,j}(\mathcal{T}_{i-1}, \mathbf{t}^P_{i,j}) = \sum_{F_{i-1,k} \leftarrow \mathbf{t}^P_{i,j}} \left[q(F_{i-1,k})\alpha_{i-1,k}(j) + \lambda \right] r^P_{i,j}(k) \quad (4)$$

As discussed earlier, using I-frame for view switching is nevertheless inefficient. Hence we consider DSC as an alternative next.

4. DISTRIBUTED SOURCE CODING IN IMVS

We now discuss three novel usages of DSC that can be easily integrated into the previous IMVS optimization. Each usage represents a different tradeoff between transmission and storage cost. An extension of method in [5] finds near-optimal structures $\mathbf{t}_{i,j}$'s for instant i, selecting among redundant P-frames, and various DSC usages, for the best tradeoff between transmission rate and storage given λ.

4.1. DSC Usage 0 for IMVS

The first DSC usage for IMVS (DSC0) is straightforward: construct a single DSC frame $W_{i,j}$ for all possible transitions into view j of instant i from frames $F_{i-1,k}$'s of previous instant. Target DSC frame $W_{i,j}$ uses transiting frames $F_{i-1,k}$'s as predictors. The size of DSC frame $W_{i,j}$ is modeled as $r^{W0}_{i,j}(d)$, where d is the maximum view index difference between the target and a predictor; size of a DSC frame in general is proportional to the amount of correlation between the target and the weakest correlated predictor [6]. An example of DSC0 is shown in Fig. 3(b).

Expression for Lagrangian cost of DSC0, $l_{i,j}^{W0}$, is the same as (3) with DSC frame size $r_{i,j}^{W0}(d)$ replacing I-frame size $r_{i,j}^{I}$. Since $r_{i,j}^{W0}(d) \leq r_{i,j}^{I}$—any correlation between the weakest predictor and the target can be exploited for coding gain over intra-coding, Lagrangian cost of DSC0 is no worse than an I-frame.

4.2. DSC Usage 1 for IMVS

The second DSC usage (DSC1) is the following: first construct multiple P-frames $P_{i,j}$'s corresponding to all possible transitions from frames $F_{i-1,k}$'s of previous instant, then construct a DSC frame of the *same view* j and *same instant* i from the constructed P-frames. See Fig. 3(c) for an example of DSC1.

With the additional P-frames, storage cost of DSC1 is obviously larger than DSC0. However, the target DSC frame and all predictors are of the same instant and view, meaning a large correlation exists between the target and the weakest predictor, resulting in a small DSC frame. That means that highly likely and small same-view-transition P-frames (assuming users are more likely to remain in the same view) plus a small DSC frame can lead to a smaller overall expected transmission cost than DSC1. More precisely, the local Lagrangian cost of DSC1, $l_{i,j}^{W1}$, can be written as:

$$l_{i,j}^{W1}(\mathcal{T}_{i-1}, \mathbf{t}_{i,j}^{W1}) = l_{i,j}^{P}(\mathcal{T}_{i-1}, \mathbf{t}_{i,j}^{W1}) + \left[\sum_{F_{i-1,k} \leftarrow \mathbf{t}_{i,j}^{W1}} q(F_{i-1,k}) \alpha_{i-1,k}(j) + \lambda \right] r_{i,j}^{W1} \quad (5)$$

where $r_{i,j}^{W1}$ denotes the size of the DSC frame in DSC1.

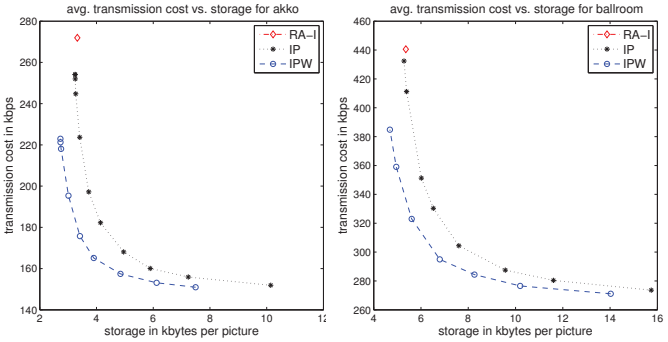
4.3. Combination of DSC Usage 0 & 1 for IMVS

We can combine the two DSC usages into a hybrid one (DSC0+1): construct multiple P-frames and DSC frame as done in DSC1, then replace unlikely transitioned P-frames with a DSC frame. See Fig. 3(d) for an example where three P-frames $P_{3,1}$'s in Fig. 3(c) are replaced by DSC frame $W_{3,1}$. The size of a DSC frame $W_{i,j}$ with multiple predictors is larger than a P-frame, hence the transmission cost of each replaced decoding path is larger. On the other hand, the combined storage costs of multiple replaced P-frames is likely larger than a single DSC frame, hence DSC0+1 offers a tradeoff of transmission and storage that is in between DSC0 and DSC1.

5. RESULTS

In the experiments we used three neighboring views from sequences akko&kayo and ballroom at 320 × 240 resolution and 30fps and 25fps, respectively. To generate DSC frames, we used the algorithm in [6], which is based on H.263 tools (e.g., half-pel motion estimation). Thus we also used H.263 to generate I-/P-frames for fair comparison. We selected QP such that I-, P- and DSC-frames were reconstructed to the same quality (around 34 dB). We assume view switching period of $M = 3$ and random access period of $M' = 30$, which correspond to roughly 100ms and 1s in time, respectively. We assume client switches to neighboring view(s) with probability 0.1.

We compare our proposed IMVS scheme using combinations of I-, P- and DSC frames (IPW) with two other schemes: scheme in [5] using only I- and P-frames (IP), a scheme that uses I-frames for all view switches (RA−I). In Figure 4(a) and 4(b), we see the tradeoffs between expected transmission rate and storage per picture for akko&kayo and ballroom, respectively, when multiplier λ was varied while optimizing (1). The results suggest followings. First, we notice that RA−I had only one point; placing I-frames at all view switching points offered no flexibility to trade off transmission rate with storage even if more storage was available. IP and IPW, on the other hand, offered a range of tradeoff points.



(a) akko&kayo Sequence (b) ballroom Sequence

Fig. 4. Tradeoffs between Expected Transmission Rate and Storage per Picture for Various IMVS Schemes

Second, we see that IP and IPW presented operating points that were lower and to the left of RA−I; i.e., using our optimization we can actually generate frame structures that are more efficient in transmission rate *and* in storage than simple I-frame insertion.

Third, we see that IPW operated at a lower convex hull than IP, indicating that DSC frame usages offer better tradeoffs than using I- and P-frames only. In particular, for the same storage as RA−I, IPW outperformed RA−I in bandwidth-efficiency by up to 28.1%, and IP by up to 20.2%. The difference between IPW and IP is more pronounced at small storage (large λ), where DSC frames for IPW (and I-frames for IP) were selected more frequently in optimized structures. Moreover, DSC frames were selected for IPW slightly more frequently than I-frames were selected for IP for the same λ, accentuating DSC frames' benefit over I-frames. At twice the storage of RA−I, IPW outperformed RA−I by up to 43.7%, showing that with larger storage, transmission rate can be dramatically reduced.

6. REFERENCES

[1] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," in *IEEE Transactions on Circuits and Systems for Video Technology*, November 2007, vol. 17, no.11, pp. 1461–1473.

[2] 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.

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

[4] 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*, Cairns, Queensland, Australia, October 2008.

[5] G. Cheung, A. Ortega, and N.-M. Cheung, "Generation of redundant frame structure for interactive multiview streaming," in *Seventeenth International Packet Video Workshop*, Seattle, WA, May 2009.

[6] N.-M. Cheung, A. Ortega, and G. Cheung, "Distributed source coding techniques for interactive multiview video streaming," in *27th Picture Coding Symposium*, Chicago, IL, May 2009.

[7] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003, vol. 13, no.7.

[8] A. Jagmohan, A. Sehgal, and N. Ahuja, "Compression of lightfield rendered images using coset codes," in *Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, November 2003, vol. 1, pp. 830–834.

[9] Prashant Ramanathan and Bernd Girod, "Random access for compressed light fields using multiple representations," in *IEEE International Workshop on Multimedia Signal Processing*, Siena, Italy, September 2004.

[10] I. Bauermann and E. Steinbach, "RDTC optimized compression of image-based scene representation (part I): Modeling and theoretical analysis," in *IEEE Transactions on Image Processing*, May 2008, vol. 17, no.5, pp. 709–723.