FRAME STRUCTURE OPTIMIZATION FOR INTERACTIVE MULTIVIEW VIDEO STREAMING WITH BOUNDED NETWORK DELAY

Xiaoyu Xiu[#], Gene Cheung[o], and Jie Liang[#]

[#] Simon Fraser University, [o] National Institute of Informatics

ABSTRACT

Interactive multiview video streaming (IMVS) is an application that streams to a client one out of N available video views for observation, but client can periodically request switches to neighboring views as the video is played back uninterrupted in time. Previous IMVS works focused on the design of a frame structure at encoding time, trading off expected transmission rate with storage, without knowing the exact view trajectory a client may select at stream time. None of the existing IMVS schemes, however, explicitly addressed the network delay problem, and so a client will suffer a round trip time (RTT) delay for each requested view-switch. In this paper, we optimize frame structure for a bounded RTT, so that a client can switch to neighboring views as the video is played back without view-switching delay. The key idea is to send additional views likely to be requested by a client within one RTT beyond the current requested view. Each required set of contiguous views (corresponding to a given current requested single view) are pre-encoded using frames of previously transmitted set of views as predictors to lower transmission rate. Using I-, P- and distributed source coding (DSC) frames, we first formulate the structure design problem as a Lagrangian minimization for a desired bandwidth/storage tradeoff. We then develop a low-complexity greedy algorithm to automatically generate a good structure. Experimental results show that for the same storage cost, the transmission rate of the proposed structure can be 42% lower than that of I-frame-only structure, and 8% lower than that of the structure without DSC frames.

Index Terms— multiview video coding, video streaming

1. INTRODUCTION

Multiview video consists of video sequences captured simultaneously by multiple closely spaced cameras. Much of previous research on multiview video focuses on multiview video coding (MVC) [1, 2], *i.e.*, how to efficiently compress *all* captured videos in a rate-distortion (RD) optimal manner by exploiting inherent correlation among neighboring views. However, MVC frame structures are not suitable for applications such as *interactive multiview video streaming* (IMVS) [3, 4, 5], where a client watches only one *single* view among N captured views at a time, but can periodically request switches to adjacent views, as the single-view video is streamed and played back in time. This is because typical MVC structures create interdependency among cross-view frames of the same time instant via inter-view prediction, and hence a server must send video data of multiple views just to correctly decode one single view requested by the client, wasting precise bandwidth.

Recently, frame structure optimizations [4, 5, 6] for IMVS have been studied, where the goal is to design frame structures at encoding time to trade off expected transmission rate and storage required to store the structure, *without* knowing the exact view trajectory a client will take at stream time. As an example, in [6] a distributed source coding (DSC) method is proposed for IMVS, where a pre-coded DSC frame can be correctly decoded at stream time, no matter which one of a known set of frames a client has available in its buffer for prediction. DSC frame is used for a targeted view in IMVS, so that multiple frames of a set of adjacent views in previous time instant can switch to the decodeable DSC frame without using a bandwidth-expensive I-frame. Though SP-frame in H.264 [7] also possesses this property of correctly decoding from multiple decoding paths, DSC frame can achieve demonstratably better storage/transmission tradeoff due to its efficient exploitation of correlation between the target frame and multiple frames of previous time instant (side information) for coding gain.

Previous IMVS works, however, do not explicitly address the problem of network delay, and hence the resulting view-switch due to a client's request will suffer a round trip time (RTT) delay. In an IMVS scenario, view-switches can occur often (e.g., on the order of several frames for view selection driven by client's head movements [8]), and hence a view-switching delay of typical RTT in the Internet (up to 300ms) can be detrimental to the interactive media experience. In this paper, we focus on the problem of IMVS with bounded network delay (IMVS-ND), where a client can play back the video in time without interruption and *perceive no view-switching delay*, even when RTT is non-negligible. The key idea is to send additional views likely to be requested by a client within one RTT beyond the current requested view. Each required set of contiguous views (corresponding to a given requested view) are pre-encoded using frames of previously transmitted set of views as predictors to lower transmission rate.

More specifically, using I-, P- and DSC frames as building blocks, we formulate a Lagrangian optimization to find the optimal frame structure that enables zero-delay view-switching in IMVS-ND. The crux of the optimization lies in finding the right mixture of different prediction types to encode video frames, each offering different tradeoff between storage and transmission rate. For a given RTT, different tradeoffs between expected transmission rate and storage are possible by varying the Lagrange multiplier value. Experimental results show that structure using appropriate I-, P- and DSC frames can lower expected transmission rate of IMVS-ND over I-frames-only structure by up to 42%, and over the structure without DSC frames by up to 8% for the same storage cost.

The outline of the paper is as follows. We overview the proposed IMVS-ND system in Sec. 2. In Sec. 3, we formulate the problem of generating the optimal frame structure for IMVS-ND. A greedy algorithm is then developed in Sec. 4 to generate the structure. Experimental results and conclusion are given in Sec. 5 and Sec. 6, respectively.

2. SYSTEM OVERVIEW

In our proposed IMVS-ND framework, videos from N closely spaced cameras capture a scene of interest, which are then encoded

offline by a server into a frame structure with I-, P- and DSC frames. A client can watch one of N available views at a given time, and can switch to adjacent viewpoints every Δ frames (*view-switching period*). In addition, the server inserts an I-frame for each viewpoint every Δ' frames to permit a required level of random access.

In the sequel, we denote *frame* to be a particular coded version of an original captured image. We use $F_{i,j}$ to denote a frame at *view-switching instant* i and view j; *i.e.*, given view-switching period Δ, $F_{i,j}$ represents a frame at frame number $i\Delta$. We assume a view switching model where, after observing frame $F_{i,j}$ of view j at view-switching instant i, the client can switch to frame $F_{i+1,k}$ of adjacent view k for the next view-switching instant $i+1$, where $\max(1, j-1) \leq k \leq \min(N, j+1)$. We define *view transition probability* $\alpha_{i,j}(k)$ as the probability that upon watching view j at view-switching instant i, the client requests view k at the next view-switching instant $i+1$. Client remains in the same view j between frame $F_{i,j}$ of view-switching instant i and frame $F_{i+1,k}$ of instant $i+1$. Thus, conventional P-frames can be used to code $\Delta - 1$ original images of the same view j after $F_{i,j}$. We focus instead on the coding structure for original images at view-switching instants.

2.1. Timing Events in Server-Client Communication

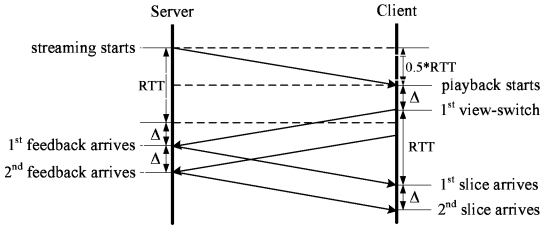


Fig. 1. Timing diagram showing communication between streaming server and client during session start-up.

We first discuss timing events during server-client communication in our system, shown in Fig. 1. At time 0, server first transmits an *initial chunk* of coded multiview data to the client, arriving at the client $\frac{1}{2}RTT$ frame time[1] later. The client starts playback at time $\frac{1}{2}RTT$, and makes her first view-switch Δ frame time later. Her first view-switch decision (feedback) is tranmitted immediately after the view-switch, and arrives at server at time $RTT + \Delta$. Responding to the client's first feedback, server immediately sends a *structure slice*, arriving at the client $\frac{1}{2}RTT$ frame time later, or RTT frame time after the client transmitted her feedback. More generally then, the client sends feedbacks in interval of Δ frame time, and in response, server sends a structure slice for each received feedback every Δ frame time. We assume there are no packet losses, and the round-trip delay between server and client remains constant.

Notice that from the time the client started playback to the time the first structure slice is received, $\Delta + RTT$ frame time has elapsed. That means the initial chunk must contain enough data to enable $\delta = \lfloor(\Delta + RTT)/\Delta\rfloor = 1 + \lfloor RTT/\Delta\rfloor$ view-switches before the first structure slice arrives. In other words, given initial view v^o at the start of the IMVS-ND streaming session and each view-switching instant can alter view position by ± 1, initial chunk must contain frames spanning contiguous views $v^o - \delta$ to $v^o + \delta$. Because subsequent structure slices arrive every Δ frame time, each structure slice only needs to enable one more view-switch of one instant for the client to play back video uninterrupted and select view without RTT delay. The view span of each structure slice, like the initial chunk, is also $2\delta + 1$.

[1] We express time in number of frames for fixed video playback speed.

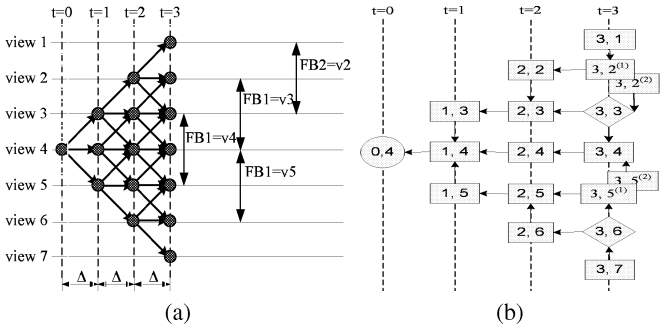


Fig. 2. Two examples for 7 total views, initial view $v^o = 4$, and $RTT = \Delta - \epsilon$: (a) progressive view-switches; (b) redundant frame structure where I-, P- and DSC frames are represented by circles, squares and diamonds, respectively. (i, j) denotes a frame at view-switching instant i of view j. A solid edge from $F_{i',j'}$ to $F_{i,j}$ means $F_{i',j'}$ is predictively coded using reference frame $F_{i,j}$.

2.2. View Switching Example

Fig. 2(a) illustrates a concrete example, where the number of views is $N = 7$, initial view is $v^o = 4$, and round trip time is $RTT = \Delta - \epsilon$ for some small positive constant $\epsilon > 0$. Therefore, the initial chunk contains only enough multiview data to enable $\delta = 1$ view-switch, spanning view $v^o - \delta = 3$ to $v^o + \delta = 5$. If the first view-switch is view 3, then the first structure slice must provide multiview data at instant 2 for possible view-switches to view 2, 3 and 4. If the first view-switch is instead view 5, then the next structure slice must contain data for possible view-switches to view 4, 5 and 6.

3. PROBLEM FORMULATION

Given the description of IMVS-ND system in previous section, the problem is to find optimal frame structure so that the expected server transmission rate is minimized while providing zero-delay view-switching. We first present necessary definitions in Section 3.1. We then formally define the IMVS-ND optimization problem in Section 3.2. Note that, for simplicity, we will use the term "instant" to mean "view-switching instant" in the sequel.

3.1. Definitions

3.1.1. Redundant Frame Structure

Given a fixed number of views N, one can construct a *redundant frame structure* $\mathcal{T}$, comprised of I-, P- and DSC frames, denoted as $I_{i,j}$'s, $P_{i,j}$'s and $M_{i,j}$'s respectively, to enable neighboring view-switches for IMVS-ND. By "redundant", we mean a given original image $F^o_{i,j}$ of instant i and view j can be represented by more than one frame $F_{i,j}$. Fig. 2(b) shows an example of a redundant frame structure for seven captured views. We see that original image $F^o_{3,2}$ is represented by two P-frames, $P^{(1)}_{3,2}$ and $P^{(2)}_{3,2}$, each encoded using a different predictor, $P_{2,2}$ and $M_{3,3}$ respectively (indicated by edges). Corresponding to which predictor frame is available at the decoder buffer, different coded frames $F_{i,j}$'s can be transmitted to enable correct (albeit slightly different) rendering of original image $F^o_{i,j}$. This is done to lower transmission cost by exploiting correlation between requested images and frames in the decoder buffer, and to avoid coding drift [4].

As an alternative to the described redundant P-frames, a single DSC frame $M_{i,j}$ (diamond in Fig. 2(b)) can achieve an identical ren-

dering of image $F^o_{i,j}$ no matter which one of several potential predictor frames is available at decoder [6]. However, unlike an I-frame (circle), by exploiting correlation between predictor frames (side information) and target frame, the size of a DSC frame is much smaller than that of I-frame. Further, DSC frames have been shown to offer better transmission/storage tradeoffs than H.264 SP-frames [7]. A DSC frame is nevertheless large than a single P-frame. Given redundant P-frames and DSC frames offer different transmission rate / storage tradeoffs, the crux of a structure optimization is to judiciously limit the use of redundant P-frames that offer saving in transmission rate over DSC frames but incur larger storage cost.

3.1.2. Structure Slice

Depending on the view-switch decision made by the client at instant $i-\delta$, different sets of coded frames will be transmitted for decoding at instant i. We define *structure slice* Ξ_i, with *center view* $v(\Xi_i)$, as a set of coded frames for decoding at instant i. Given center view $v(\Xi_i)$, the set of contiguous views covered by slice Ξ_i is $\{\max(1, v(\Xi_i)-\delta), \ldots, \min(N, v(\Xi_i)+\delta)\}$. Center view $v(\Xi_i)$ is the view selected by client at instant $i-\delta$. For example, in Fig. 2(b), initial chunk contains frames $I_{0,4}, P_{1,3}, P_{1,4}, P_{1,5}$ to cover view-switches to view 3, 4 and 5 at instant 1. If client's view selection at instant 1 is 3, then the structure slice that needs to be transmitted is $\Xi^{(1)}_2 = \{P_{2,2}, P_{2,3}, P_{2,4}\}$ with $v(\Xi^{(1)}_2) = 3$. Instead, if client remains in view 4 at instant 1, then structure slice $\Xi^{(2)}_2 = \{P_{2,3}, P_{2,4}, P_{2,5}\}$ will be sent to decoder with $v(\Xi^{(2)}_2) = 4$. Notice that different slices ($\Xi^{(1)}_2$ and $\Xi^{(2)}_2$ in this example) can contain the same frame(s) ($P_{2,3}$ and $P_{2,4}$).

3.1.3. Transmission Schedule

As shown in the above example, given a frame structure $\mathcal{T}$, the structure slice Ξ_i of instant i to be transmitted depends on the structure slice Ξ_{i-1} of previous instant available in decoder, and client's view selection h at instant $i-\delta$. The center view $v(\Xi_i)$ of Ξ_i will necessarily be h, where $h \in \{v(\Xi_{i-1}), v(\Xi_{i-1}) \pm 1\}$. We can formalize associations among Ξ_{i-1}, h and Ξ_i for all i given frame structure $\mathcal{T}$ via a *transmission schedule* G. More precisely, G dictates which structure slice Ξ_i will be transmitted for decoding at instant i, given previous slice Ξ_{i-1} is available at decoder and client selects view h at instant $i-\delta$:

$$G : (\Xi_{i-1}, h) \Rightarrow \Xi_i, \quad h \in \{v(\Xi_{i-1}), v(\Xi_{i-1}) \pm 1\} \tag{1}$$

where center view of Ξ_i is $v(\Xi_i) = h$. In what follows, we denote a scheduled transmission from slice Ξ_{i-1} to slice Ξ_i, with client selected view h at instant $i-\delta$, as: $(\Xi_{i-1}, h) \overset{G}{\Rightarrow} \Xi_i$.

3.2. Optimization Problem

Having defined a structure slice Ξ_i and transmission schedule G for a given frame structure $\mathcal{T}$, we can now formalize the design of an optimal redundant frame structure as an optimization. We first present some necessary definitions, then formally define the problem.

3.2.1. Definitions

A). Structure Slice Probability: $q(\Xi_i)$ is the probability that structure slice Ξ_i for decoding at instant i is transmitted. Considering the initial chunk Ξ_0 is always sent to the client, this probability could be computed recursively using view transition probability $\alpha_{i,j}(k)$:

$$\begin{aligned} q(\Xi_0) &= 1 \qquad (2)\\ q(\Xi_{i+1}) &= \sum_{\Xi_i | (\Xi_i, v(\Xi_{i+1})) \overset{G}{\Rightarrow} \Xi_{i+1}} q(\Xi_i)\alpha_{i-\delta, v(\Xi_i)}(v(\Xi_{i+1})) \end{aligned}$$

In words, (2) states that $q(\Xi_{i+1})$ is the sum of probabilities of each slice Ξ_i tarnsitioning to slice Ξ_{i+1}, scaled by the slice probability of Ξ_i itself, $q(\Xi_i)$, given schedule G dictates slice transmission in structure $\mathcal{T}$.

Further, we define *frame transmission probability* $q_c(F_{i,j})$ as the probability that a frame $F_{i,j}$ is transmitted from server to decoder, which can be calculated using the defined structure slice probability (2):

$$q_c(F_{i,j}) = \sum_{\Xi_i | F_{i,j} \in \Xi_i} q(\Xi_i) \tag{3}$$

In words, the transmission probability of a frame $F_{i,j}$ is the sum of probabilities of slices Ξ_i's that include $F_{i,j}$.

B). Storage Cost: For a given frame structure $\mathcal{T}$, we can define the corresponding *storage cost* by simply adding up the sizes of all the coded frames $F_{i,j}$'s in $\mathcal{T}$, i.e.,

$$B(\mathcal{T}) = \sum_{F_{i,j} \in \mathcal{T}} |F_{i,j}| \tag{4}$$

C). Transmission Cost: Given a frame structure $\mathcal{T}$ and associated schedule G, *transmission cost* is defined as the sum of the sizes of all the frames $F_{i,j}$'s in $\mathcal{T}$, scaled by the corresponding frame transmission probabilities $q_c(F_{i,j})$'s:

$$C(\mathcal{T}) = \sum_{F_{i,j} \in \mathcal{T}} q_c(F_{i,j}) |F_{i,j}| \tag{5}$$

3.2.2. Optimization Problem Definition

We can now define the design of redundant frame structure for IMVS-ND as an optimization problem: given a fixed number of viewpoints, how to find a structure $\mathcal{T}^*$ and associated schedule G^*, using a combination of I-, P- and DSC frames, that minimizes the transmission cost $C(\mathcal{T})$ while a storage constraint $\overline{B}$ is observed:

$$\arg\min_{\mathcal{T}} C(\mathcal{T}) \quad s.t.\ B(\mathcal{T}) \le \overline{B} \tag{6}$$

Instead of the constrained problem in (6), we solve the corresponding unconstrained Lagrangian problem; *i.e.*,

$$\min_{\mathcal{T}} J(\mathcal{T}) = C(\mathcal{T}) + \lambda B(\mathcal{T}) = \sum_{F_{i,j} \in \mathcal{T}} (q_c(F_{i,j}) + \lambda)|F_{i,j}| \tag{7}$$

where λ is the Lagrangian multiplier. From (7), we can see that a captured image can be represented by redundant P-frames, each having a comparatively small transmission cost $q(P^{(h)}_{i,j})|P^{(h)}_{i,j}|$, but all together comprising a large storage $\sum_h |P^{(h)}_{i,j}|$. When λ is small, the penalty on large storage is negligible and redundant P-frames are attractive. However, when λ is large, the penalty on large storage cost becomes expensive and one single representation of the picture as I- or DSC frame with large transmission cost but small storage is more preferable.

4. ALGORITHM DEVELOPMENT

We now derive a greedy optimization algorithm to generate good frame structures, based on the optimization problem defined in Sec. 3.2. In a nutshell, we iteratively build one "layer" t_i (composed of structure slices Ξ_i's) at each instant from front to back; *i.e.*, given structure from initial chunk Ξ_0 up to layer t_l of instant l, we construct layer t_{l+1} and corresponding local schedule g_{l+1} at instant $l+1$, then layer t_{l+2} and local schedule g_{l+2} at instant $l+2$, etc. At each instant i, the key question is: given the scheduled structure

$\mathcal{T}_{i-1}(G_{i-1})$ constructed up to instant $i-1$, how to optimally construct layer t_i and its local schedule g_i at instant i to minimize (7) for a given λ.

To construct locally optimal structure layer t_i at instant i, we initialize layer t_i where one DSC-frame is used to represent each view. This layer has no redundant representation (one frame per captured image), and thus it has minimum storage while large sizes of DSC-frames will lead to a large transmission cost. Next, to methodically reduce transmission cost, we can incrementally add the most beneficial redundant P-frames one at a time, thereby increasing storage. We terminate when no more beneficial redundant P-frames can be added to further lower local Lagrangian cost.

In details, we describe the algorithm as follows. First, as initial structure for layer t_i, we construct one DSC frame for each view j at instant i, where all viable view-switches to view j from coded frames $F_{i-1,k}$'s in t_{i-1} can transition. We then determine the corresponding schedule g_i and compute the local Lagrangian cost in (7). Given the initial layer, we improve layer t_i by iteratively making augmentations: selecting one candidate from a set of structure augmentations that offers the largest decrease in local Lagrangian cost. The augmentations include:

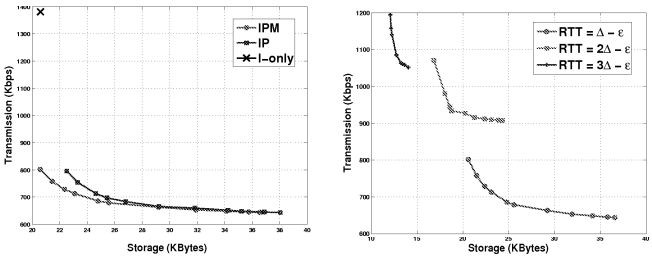
- adding a new P-frame $P_{i,j}$ to t_i, predicted from an existing frame $F_{i,k}$ of neighboring view k of same instant i.
- adding a new P-frame $P_{i,j}$ to t_i, predicted from an existing frame $F_{i-1,j}$ in Ξ_{i-1} of the same view of the previous instant $i-1$.
- selecting a different predictor $F_{i,k}$ of the same instant i for an already constructed P-frame $P_{i,j}$ in t_i.

The above process repeats to find the most locally beneficial augmentation at each iteration, update the corresponding schedule and compute local Lagrangian cost, until no Lagrangian cost reduction can be found. Note that after updating the local schedule at each iteration, it is possible that some frames in t_i are not used by any view-switch. In this case, those unused frames will be removed from the structure to save storage.

5. EXPERIMENTATION

We use H.263 tools to encode the first 90 frames of VGA size (640×480) sequence *akko&kayo* of 5 views ($N=5$), at 15 frames per second. To generate data of DSC frames, we use the algorithm in [6], developed using H.263 tools. We select quantization parameters such that I-, P- and DSC frames are reconstructed to the same quality (around 32dB). In addition, the random access period Δ' and switch period Δ are set to be 15 and 3, respectively. For view transition probability $\alpha_{i,j}(k)$, we assume a client remains at the same view with probability α and switches to each adjacent view with probability $(1-\alpha)/2$. A boundary view (view 1 or N) switches to the single neighboring view (view 2 or $N-1$) with probability $(1-\alpha)$. We assume $\alpha = 0.5$ throughout the experiment.

In Fig. 3(a), we compare the performance of frame structures generated by the proposed algorithm using I-, DSC and P-frames (`IPM`), using I- and P-frames (`IP`), and using only I frames (`I-only`), when round-trip delay $RTT = \Delta - \epsilon$ for small positive $\epsilon > 0$. First, we observe that `I-only` had a single tradeoff point, because placing I-frames at all switching points results in no flexibility to trade off between storage and transmission rate. Second, `IPM` offers lower transmission rates by up to 42% than `I-only` *for the same storage*, due to the judicious usage of redundant P- and DSC frames. Third, using DSC frames can generate better tradeoff points than using I-frames with smaller transmission rate up to 8%. The improvement is larger at stringent storage constraint, because



(a) (b)

Fig. 3. Tradeoff between expected transmission and storage (a) using different coding configurations when $RTT = \Delta - \epsilon$; (b) for different RTTs.

DSC frames are more often used by the optimized structure to lower overall storage.

The tradeoff points of the proposed algorithm under different round trip delays RTT's are shown in Fig. 3(b). We observe that for the same storage, larger RTT means larger transmission rate. This is intuitive because larger RTT means larger view span $2\delta + 1$ is required for each structure slice Ξ_i to enable zero view-switching delay for IMVS-ND client.

6. CONCLUSION

In this paper, we address the problem of interactive multiview video streaming with bounded network delay (IMVS-ND), and develop a greedy algorithm to generate a good redundant frame structure to enable bandwidth-efficient view switching with zero view-switching delay. Experimental results demonstrate that the frame structure generated from the proposed algorithm could significantly reduce the expected transmission rate over standard frame structures like I-frames-only structure for given storage cost.

7. REFERENCES

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

[2] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multi-view imaging and 3DTV," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 10–21, November 2007.

[3] A. M. Tekalp, E. Kurutepe, and M. R. Civanlar, "3DTV over IP: end-to-end streaming of multiview video," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 77–87, November 2007.

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

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

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

[7] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Transcation on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, July 2003.

[8] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, "Client-driven selective streaming of multiview video for interactive 3DTV," in *IEEE Transactions on Circuits and Systems for Video Technology*, November 2007, vol. 17, no.11, pp. 1558–1565.