OPTIMAL FRAME STRUCTURE DESIGN USING LANDMARKS FOR INTERACTIVE LIGHT FIELD STREAMING

Wei Cai[o], Gene Cheung[#], Sung-Ju Lee[*], Taekyoung Kwon[o]

[o] Seoul National University, [#] National Institute of Informatics, [*] Hewlett-Packard Laboratories

ABSTRACT

Light field is a large set of spatially correlated images of the same static scene captured using a 2D array of closely spaced cameras. Interactive light field streaming is the application where a client continuously requests successive light field images along a view trajectory of his choosing, and in response the server transmits appropriate data for the client to correctly reconstruct desired images. The technical challenge is how to encode captured light field images into a reasonably sized frame structure a priori (without knowing eventual clients' view trajectories), so that at stream time, expected server transmission rate can be minimized, while satisfying client's view-switch requests. In this paper, using I-frames, redundant P-frames and distributed source coding (DSC) frames as building blocks, we design coding structures to optimally trade off storage size of the frame structure with expected server transmission rate. The key novelty is to facilitate the use of "landmarks" in the structure—popular reference frames cached in the decoder buffer—so that the probability of having at least one useful predictor frame available in the buffer for disparity compensation is greatly increased. We first derive recursive equations to find the optimal caching strategy for a given coding structure. We then formulate the structure design problem as a Lagrangian minimization, and propose fast heuristics to find near-optimal solutions. Experimental results show that the expected server streaming rate can be reduced by up to 93.6% compared to an I-frame-only structure, at twice the storage required.

Index Terms— light field, interactive streaming, optimization

1. INTRODUCTION

Light field [1] is a large set of spatially correlated images of the same static scene taken from a 2D array of closely spaced cameras. Because conventional display terminals show only one image at a time, a client typically browses the light field data by selecting single images in succession across time [2]. In a network streaming scenario then, a server will transmit pre-encoded images corresponding to client's successive view-switch requests along his chosen view trajectory. This network streaming service is called *interactive light field streaming* (ILFS) [3] in the literature.

The technical challenge for ILFS is to encode captured light field images into a reasonably sized frame structure *a priori*, so that during actual streaming session, the expected server transmission rate to the client interactively selecting views is minimized. The problem is challenging because at encoding time, the exact view trajectory that a client will take at stream time is unknown, making it difficult to employ conventional *differential coding* to reduce the transmission rate. Differential coding, typical in compression of temporal frames in single-view video like H.263, assumes a previous decoded frame $\hat{F}_{t-1}$ of time instant $t-1$ is available at decoder for prediction of target image F_t of instant t, so that only (quantized) differential $F_t - \hat{F}_{t-1}$ needs to be coded and transmitted. If view trajectory in spatial frames in ILFS is not known at encoding time, then no frame can be assumed to be available at decoder with certainty for prediction of the target image, and differential coding cannot be applied as is. A simple alternative strategy is to forego differential coding and encode every light field image as an independently coded I-frame. However, this results in a large server transmission rate because no inter-view correlation is exploited for coding gain.

In our previous work [4], we designed redundant frame structures using I-frames, P-frames and distributed source coding (DSC) frames [5] to optimally trade off storage size of the structure with expected server transmission rate. The basic idea is simple: for each pair of views (i, j) that are likely to be requested in succession during ILFS, encode one quantized differential $F_j - \hat{F}_i$ as P-frame $P_j(i)$ *a priori*; i.e., a differentially coded P-frame of view j that uses a decoded frame of view i as predictor. During streaming session then, when a client requests view j after viewing i, essentially only the pre-encoded differential[1] $P_j(i)$ needs to be transmitted to reconstruct view j, rather than an independently coded I-frame I_j, lowering server transmission rate. Pre-encoding differentials for many view pairs, however, incurs a large storage cost. For the extreme case where I-frames are never desired to be sent, $O(M^2)$ differentials for M light field images need to be pre-encoded to cover all pairs. For a large light field, this is too expensive storage-wise.

To decrease the storage cost of the frame structure while keeping server transmission cost low, in this paper we propose to design new coding structures that maximally utilizes *landmark* frames. A landmark frame $\hat{F}_l$ is a popular reference frame for which many differentials $F_j - \hat{F}_l$, $P_j(l)$'s, have been pre-encoded into the structure. So if a client can flexibly choose which frame to use as reference for future frames—cached landmark frame $\hat{F}_l$ in its one-frame decoder buffer or the current displayed frame—then one can achieve low server transmission cost for future requests to many views j's using pre-encoded differentials $P_j(l)$'s. For the same extreme case where I-frame transmission is never desirable, one only needs to pre-encode differentials from a single landmark $\hat{F}_l$ to all other views to achieve no-I-frame transmission for any view-switches, resulting in storage cost $O(M)$ instead of previous $O(M^2)$.

More generally, we investigate the optimal frame structure design problem with a variable number of landmark frames, each with different pre-encoded differentials. We first derive recursive equations to find the optimal caching strategy (which landmark to cache given current displayed view i) for a given structure. Then, we formulate the structure design problem as a Lagrangian minimization, and propose fast heuristics to find near-optimal solutions. Experimental results show that the expected server streaming rate can be reduced by up to 93.6% using our structure with landmarks compared to an I-frame-only structure, at twice the storage required.

The paper outline is as follows. We review related work in Sec-

[1] A small DSC frame W_j that merges different coded versions of view j into one will also need to be transmitted, as detailed in Section 3.

tion 2. We then discuss an interaction model to capture user's view-switching behavior and a structure to satisfy user's view-switching requests in Section 3. We derive the optimal caching strategy given a coding structure in Section 4. We formulate our structure design problem as a Lagrangian minimization and present a fast heuristic algorithm in Section 5. Results and concluding remarks are presented in Section 6 and 7, respectively.

2. RELATED WORK

The uncertainty of which predictor frame will be available at the decoder buffer for differential coding of a target image during encoding time is a major source of difficulty when compressing light field images for ILFS. Early coding structure proposals to address this difficulty include [6, 7]. [6] assumed a user only switches to an adjacent view during an ILFS session, and hence one out of a small subset of adjacent frames must be available at decoder for prediction of the target image during a view-switch. [6] then proposed to differentially encode one SP-frame for each predictor frame, so that the server can transmit an SP-frame corresponding to the predictor frame residing in the decoder during stream time. The identical construction property of SP-frames ensures the same reconstruction of the target image no matter which SP-frame (corresponding to the predictor frame in the decoder cache) was actually transmitted.

For the same assumption of adjacent view-switches, [7] proposed to use DSC frames instead, where the number of least-significant-bit (LSB) bit-planes of transform coefficients that need to be transmitted depends on the quality of the *side information*, *i.e.*, the correlation between the predictor frame at decoder and the target image. The key difference between [6, 7] and our work is that we assume a much more general view-switching model for ILFS, where non-adjacent views can be selected by clients (see example user interface in [2] where non-adjacent views can be selected effortlessly). Keeping server transmission rate low even for more generally accessed light field images is a new technical challenge.

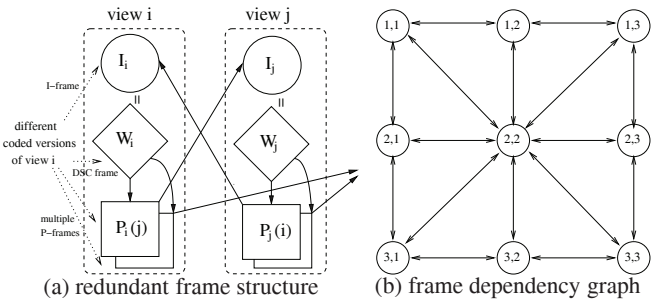
[8] have studied redundant frame structures for *interactive multiview video streaming* (IMVS), where a user can periodically switch to an adjacent view among M captured views, as the video is streamed continuously from server and played back in time uninterrupted at client. Though the notion of interactive navigation in high-dimensional media space is similar, ILFS is free navigation of spatial images in a static scene. That means unlike IMVS, a light field image can be revisited, creating loops in the view trajectory and more difficulty in the coding structure design. We will see how our proposed structure handles recurring frames in Section 3.

3. INTERACTION MODEL & FRAME STRUCTURE

We first describe an interaction model for ILFS that captures the view-switching behavior of clients, and make assumptions on client's decoder capabilities. We then present a redundant frame structure composed of I-, P- and DSC frames that can satisfy possible client's view-switching requests. Finally, we describe a directed graph as an abstraction for a particular instance of the structure.

3.1. Interaction Model for ILFS

Of M available views (x, y)'s on a light field Cartesian grid, we assume all clients start from an initial view $s = (x_s, y_s)$ at the start of an ILFS session. Thereafter, a client switches from view i to view j with probability $p_{i,j}$. In this paper, we restrict permissible view-switches from view i to eight other views j's: four neighboring views in the horizontal and vertical directions, plus four closest *anchor* views of coordinates $(x'K, y'K)$'s, for integer $K > 1$. In other words, m anchor views have coordinates (x, y)'s being multiples of K and lie on a coarser grid. This restriction stems from



(a) redundant frame structure (b) frame dependency graph

Fig. 1. Frame structure and frame dependency graph. I-, P- and DSC frames are shown as circles, squares and diamonds in (a). Differential $P_j(i)$ is shown as an edge $e_{i\to j}$ from node i to j in (b).

observation on typical static scene view-switching interfaces on a computer like [2], where one can drag a mouse to switch to a neighboring view for smooth view transition, or click arrow keys on a keyboard to jump to views on a coarse grid for quick view sampling.

In addition, we assume that a client has average lifetime of T instants; *i.e.*, he will exit the ILFS session after T view-switches on average. A client has either a *fixed* or *flexible* one-frame decoder buffer. Fixed buffer means the current displayed frame is always moved to buffer as reference. Flexible buffer means either the current displayed frame or the previous reference frame can be cached.

3.2. Redundant Frame Structure

To satisfy a client's view-switch request from view i to j, we construct a redundant frame structure as follows. Each view j is first encoded as an independently coded I-frame I_j. In addition, a variable number of differentially coded P-frames $P_j(i)$'s are also encoded, each disparity compensated from a I-frame I_i of predictor view i. Finally, a DSC frame W_j is encoded, using encoded versions of $P_j(i)$'s of the *same* view j as predictors and I-frame I_j as target. DSC frame is encoded with multiple predictors in such a way that it can be perfectly reconstructed to its target if *at least one* predictor frame is available at decoder as side information [5]. The sizes of DSC and I-frame in our setup are roughly twice and ten times the size of a P-frame, respectively. See Fig. 1(a) for an illustration.

Consider first the case when client has a fixed one-frame buffer. When client requests a view-switch from view i to j, if P-frame $P_j(i)$ was pre-encoded into the structure, then a small $P_j(i)$ can be transmitted from server, decoded using previously reconstructed I-frame I_i of view i as predictor. (Any P-frame $P_i(k)$ plus DSC frame W_i also reconstruct I_i perfectly by property of DSC.) We call this 1-*hop transmission*. If $P_j(i)$ is not available, then a large I-frame I_j can be transmitted. We call this 0-*hop transmission*. If $P_j(i)$ is transmitted, DSC-frame W_j must be sent in addition, so that identical I-frame I_j is reconstructed in either case. This is done so that differentially coded frames $P_k(j)$'s that use view j as predictor can predict from a single unified version I_j.

Consider now the case when client has a flexible buffer, where the client can choose whether to use current displayed I-frame I_i, or landmark I_l of view l currently in cache for future reference. When a client switches from view i to j, in 1-hop transmission, pre-encoded differential $P_j(i)$ *or* $P_j(l)$ can now be transmitted (plus DSC frame W_j), if available. Beside 0-hop transmission, for the flexible buffer case, we consider in addition 2-*hop transmission*, where a client can first switch to an intermediate view k before switching to target view j. There are two potential advantages to 2-hop transmission. First, in the absence of pre-encoded differentials $P_j(i)$ or $P_j(l)$ (making 1-

hop transmission impossible), there may exist differentials to an intermediate view k and then to target j, so that the combined differential transmission cost is smaller than 0-hop transmission to target j. Second, the intermediate view k may be an important landmark with pre-encoded differentials to many other views, so that by caching it en route to view j, it will be beneficial for future view-switches.

3.3. Frame Dependency Graph

From previous discussion, it is clear that in the frame structure there exists a degree of freedom at each view j, where the number of differentials $P_j(i)$'s from predictor views i's can be freely chosen. We represent a particular selection of differentials $P_j(i)$'s for each view j in a structure as a direct graph θ called *frame dependency graph*. Specifically, we draw an edge $e_{i\to j}$ from view i to j if there exists differential $P_j(i)$ in the structure. See Fig. 1(b) for an illustration, where view $(2,2)$ in a 3×3 light field has differentials to and from all other views. The optimization is to find a graph θ so that the expected server transmission cost is minimized subject to a server storage constraint. We first define server transmission cost next.

4. OBJECTIVE FUNCTION & CACHING STRATEGY

We now define the objective function—expected server transmission cost—for a client during an ILFS session, given a frame dependency graph θ of frame structure. We study the fixed and flexible decoder buffer cases in order. For the flexible buffer case, by solving the cost function we also find the optimal caching strategy: which I-frame I_l should a client cache as the landmark when switching from view i to j in order to minimize expected server transmission cost.

4.1. Transmission Cost for Fixed One-frame Buffer

For the fixed one-frame buffer case, we write expected transmission cost $c_i^{(t)}(\theta)$ given client is at view i at instant t as:

$$c_i^{(t)}(\theta)=\sum_j p_{i,j}\min\left[\, h1_i^{(t)}(j,\theta),\ h0_i^{(t)}(j,\theta)\,\right] \tag{1}$$

where $h1_i^{(t)}(j,\theta)$ and $h0_i^{(t)}(j,\theta)$ are the costs of 1-hop and 0-hop transmission from view i to j, respectively. The transmission cost at instant 0, $c_s^{(0)}(\theta)$, has in addition a startup cost r_s^I—size of I-frame I_s for initial view s.

1-hop transmission cost $h1_i^{(t)}(j,\theta)$ can be written recursively as the sum of differential coding cost $r_j^P(i,\theta)$ from view i to j plus future cost $c_j^{(t+1)}(\theta)$ at view j of instant $t+1$:

$$h1_i^{(t)}(j,\theta)=r_j^P(i,\theta)+\mathbf{1}(t<T)\,c_j^{(t+1)}(\theta) \tag{2}$$

Differential coding cost $r_j^P(i,\theta)$ equals to the sum of sizes of pre-encoded P-frame $P_j(i)$ and DSC frame W_j if $P_j(i)$ exists in structure θ, and ∞ if it does not; *i.e.*,

$$r_j^P(i,\theta)=\begin{cases}|P_j(i)|+|W_j| & \text{if } e_{i\to j}\in\theta\\ \infty & \text{o.w.}\end{cases} \tag{3}$$

$\mathbf{1}(x)$ is the indicator function that evaluates to 1 if clause x is true and 0 otherwise.

0-hop transmission from view i to j can be similarly written as the sum of independent coding cost r_j^I to view j plus future cost $c_j^{(t+1)}(\theta)$:

$$h0_i^{(t)}(j,\theta)=r_j^I+\mathbf{1}(t<T)\,c_j^{(t+1)}(\theta) \tag{4}$$

r_j^I is simply the size of I-frame $|I_j|$.

These recursive equations can be efficiently solved using dynamic programming (DP). Specifically, each time (1) is solved for cost $c_i^{(t)}(\theta)$, it is stored in DP table entry $C[t][i]$, so that next time the same sub-problem $c_i^{(t)}(\theta)$ is called in the recursion, the solution can be simply looked up. The complexity of evaluating expected server transmission cost for given structure θ is the number of operations in (1) needed to fill each entry in the DP table, multiplied by the total number of entries in table, $T\times M$. Given a client can switch from view i to only eight other views (as discussed in Section 3.1), the number of operations in (1) is constant. Hence the order of complexity for transmission cost evaluation is $O(TM)$.

4.2. Transmission Cost for Flexible One-frame Buffer

For the flexible one-frame buffer case, the transmission cost function is only slightly more involved. We write the expected transmission cost $c_i^{(t)}(l,\theta)$ of a client currently at view i at time t with view l as landmark as follows:

$$c_i^{(t)}(l,\theta)=\sum_j p_{i,j}\min\left[h1_i^{(t)}(l,j,\theta),\ h0_i^{(t)}(l,j,\theta),\ h2_i^{(t)}(l,j,\theta)\right] \tag{5}$$

where in addition to 1- and 0-hop transmission cost as expressed in (1), there is a third option of 2-hop transmission cost $h2_i^{(t)}(l,j,\theta)$. We first write the new 1-hop cost as follows:

$$\begin{aligned}h1_i^{(t)}(l,j,\theta) \;=\; &\min\Big[r_j^P(i,\theta)+\mathbf{1}(t<T)c_j^{(t+1)}(i,\theta),\\ &\quad r_j^P(l,\theta)+\mathbf{1}(t<T)c_j^{(t+1)}(l,\theta)\Big]\end{aligned} \tag{6}$$

where the sum of transition cost from view i to j and recursive cost for the next instant can be one of two possible choices: either current displayed view i or landmark l can be used as predictor to decode target view j. In the first case, view i becomes the new landmark, while in the second case, landmark l is retained.

0-hop transmission for flexible buffer case is similar to (4), with the difference being that the recursive cost $c_j^{(t+1)}(\theta)$ can select either previous view i or landmark l as the next landmark:

$$h0_i^{(t)}(l,j,\theta)=r_j^I+\mathbf{1}(t<T)\min\left[c_j^{(t+1)}(i,\theta),c_j^{(t+1)}(l,\theta)\right] \tag{7}$$

2-hop transmission is similar to (6), but there are now two transition costs: from view i to an intermediate view k, and from view k to target view j. All views k's with 1-hop connection to target view j, $\forall k|e_{k\to j}\in\theta$, should be considered as potential intermediates:

$$\begin{aligned}h2_i^{(t)}(l,j,\theta) \;=\; &\min_{\forall k|e_{k\to j}\in\theta}\Big\{\min\left[r_k^P(i,\theta),r_k^P(l,\theta),r_k^I\right]+r_j^P(k,\theta)+\\ &+\mathbf{1}(t<T)\,c_j^{(t+1)}(k,\theta)\Big\}\end{aligned} \tag{8}$$

In words, 2-hop transmission cost is the cost of arriving at intermediate view k (using current displayed frame i or landmark l as reference, or sending I-frame I_k), plus the size of P-frame $P_j(k)$, plus future recursive cost given frame k is in one-frame buffer.

The order of complexity to compute transmission cost for flexible buffer can be analyzed as follows. First, the size of the DP table has an extra dimension to indicate the landmark view l used at instant t and current view i; i.e., DP table size is now $T\times M\times M$. We use (5) to fill each DP table entry, where again each current view i only has eight possible views to switch to. 1-hop and 0-hop transmissions in (6) and (7) take constant number of operations. 2-hop transmission in (8), on the other hand, can have a worst case of M edges $e_{k\to j}$'s in θ. Hence the complexity of the transmission cost evaluation for the flexible buffer case is $O(TM^3)$.

5. OPTIMIZATION & ALGORITHM

Having defined the expected server transmission cost function for given structure θ, we next define the storage cost $b(\theta)$ of pre-encoded differentials given θ simply:

$$b(\theta) = \sum_{e_{i\to j}\in\theta} |P_j(i)| \quad (9)$$

We do not count storage cost for I- and DSC frames in $b(\theta)$, since we encode one each for each view i in the structure anyway.

We can now formally define our optimization as a constrained optimization: find structure θ with minimum expected server transmission cost such that the storage constraint $\bar{B}$ is observed:

$$\min_{\theta} c_s^{(0)}(\theta) \quad \text{s.t.} \quad b(\theta) \le \bar{B} \quad (10)$$

Note that $\bar{B}$ is the storage available for differentials *beyond* storage already used for I- and DSC frames I_j and W_j for each view j.

5.1. Heuristic Algorithm

Instead of solving the original constrained problem (10), we solve the unconstrained Lagrangian equivalent instead:

$$\min_{\theta} c_s^{(0)}(\theta) + \lambda b(\theta) \quad (11)$$

where $\lambda \ge 0$ is the Lagrange multiplier trading off transmission cost $c_s^{(0)}(\theta)$ and storage $b(\theta)$.

To solve (11), we use a simple heuristic algorithm. We start from a disconnected graph θ with no edges. At each iteration, we greedily find the most "beneficial" single edge $e_{i\to j}$, *or* pair of edges $e_{i\to k}$ and $e_{k\to j}$ and add them to θ. By "beneficial", we mean an edge or edge pair that induces the largest decrease in Lagrangian cost in (11). The algorithm terminates when no more beneficial edge or edge pair can be found.

6. EXPERIMENTATION

To validate the performance of our proposed structures, we set up the following experiment. We downloaded light field image set `bunny` from [2], each image of size 1024×1024. To encode I- and P-frames and DSC frames, we used a H.263-based codec in [5]. Quantization parameters were set so that the Peak Signal-to-Noise (PSNR) of the encoded frames was around 32dB. Instead of generating a P-frame $P_j(i)$ for every possible predictor-predictee view pair (i, j), we assume its size can be approximated as follows:

$$|P_j(i)| = (|I_j| - |P_j(j+1)|)(1 - e^{-\gamma(|i-j-1|)}) + |P_j(j+1)| \quad (12)$$

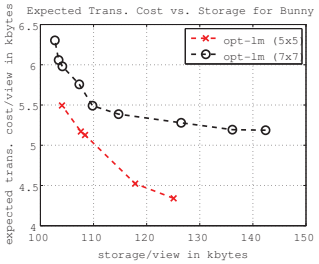
In other words, the further apart view i and j are, the closer in size $P_j(i)$ is to I-frame I_j. We set $\gamma = 0.55$ in the experiment.

View-switching probabilities to adjacent and anchor views were 0.4 and 0.6, respectively. K for anchor views is half the light field image set width. Average lifetime of a ILFS session were half the light field images. Lagrange multiplier λ in (11) was varied to induce different tradeoffs between transmission rate and storage.

We compare performance of our generated structures (`opt-lm`) to three others. `I-only` encodes only one I-frame I_j for each light field image j. `P-adj` encodes in addition four P-frames $P_j(i)$'s for each adjacent horizontal or vertical views. `opt-nlm` is our previously proposed structure in [4] without landmarks (fixed buffer).

6.1. Experimental Results

In Fig. 2, we see the tradeoff between expected server transmission cost and storage per view. In Fig. 2(a), by varying λ, we can induce different tradeoffs for both 5×5 and 7×7 light field. 5×5 has better tradeoffs because regular views in 5×5 are closer to anchor views than in 7×7. In Fig. 2(b), we see the performance of the four structures. At about twice the storage of `I-only`, both previously proposed `opt-nlm` and new `opt-lm` reduced transmission cost drastically over `I-only`, by 50.8% and 93.6%, respectively. Note that the additional reduction in transmission cost of `opt-lm` over `opt-nlm` is 42.8%, which is substantial. The gain of `opt-lm` over `opt-nlm` stems first from the creation of a third option of 2-hop transmission in (5), and then the flexibility of choosing frames for reference for 1-hop (6), 0-hop (7) and 2-hop (8) transmission to reduce transmission cost to target view j.



(a) tradeoffs for diff. LF sizes

structure	storage	trans.
`I-only`	70.8	70.8
`P-adj`	101.7	43.7
`opt-nlm`	131.8	34.8
`opt-lm`	136.1	5.2

(b) performance for diff. structures

Fig. 2. Performance in expected server transmission rate per view versus storage per view, all in kbytes. (a) shows the said tradeoff for 5×5 and 7×7 light fields. (b) shows performance points of different coding structures.

7. CONCLUSION

Designing good frame structures for interactive light field streaming (ILFS) is challenging because at encoding time, the server does not know the particular view navigation path a client will take at stream time. To overcome this uncertainty while maintaining a low server transmission rate, in this paper we design a frame structure composed of I-frames, P-frames and Distributed Source Coding (DSC) frames, so that a likely view pair (i, j) will have differential $F_j - \hat{F}_i$ pre-encoded and ready for transmission when client switches from view i to j. The key novelty is the use of landmarks, popular reference frames cached at client, so that the probability of having at least one useful predictor frame available for disparity compensation is greatly increased. Experiments show that using landmark frames, expected server transmission rate can be decreased by 93.6% compared to I-frame-only structure at twice the storage.

8. REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *ACM SIGGRAPH*, New Orleans, LA, August 1996.

[2] "Stanford Light Field Archive," http://lightfield.stanford.edu/lfs.html.

[3] P. Ramanathan, M. Kalman, and B. Girod, "Rate-distortion optimized interactive light field streaming," in *IEEE Trans. on Multimedia*, June 2007, vol. 9, no.4, pp. 813–825.

[4] W. Cai, G. Cheung, T. Kwon, and S.-J. Lee, "Optimized frame structure for interactive light field streaming with cooperative cache," in *IEEE International Conf. on Multimedia and Expo*, Barcelona, Spain, July 2011.

[5] 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.

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

[7] A. Aaron, P. Ramanathan, and B. Girod, "Wyner-Ziv coding of light fields for random access," in *IEEE International Workshop on Multimedia Signal Processing*, Siena, Italy, September 2004.

[8] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," in *IEEE Trans. on Image Processing*, March 2011, vol. 20, no.3, pp. 744–761.