REDUNDANT FRAME STRUCTURE USING M-FRAME FOR INTERACTIVE LIGHT FIELD STREAMING

Benedicte Motz [*], *Gene Cheung* [#], *Antonio Ortega* [$]

[*] The Graduate University for Advanced Studies, [#] National Institute of Informatics,
[$] University of Southern California

ABSTRACT

A light field (LF) is a 2D array of closely spaced viewpoint images of a static 3D scene. In an interactive LF streaming (ILFS) scenario, a user successively requests desired neighboring viewpoints for observation, and in response the server must transmit pre-encoded data for correct decoding of the requested viewpoint images. Designing frame structures for ILFS is challenging, since at encoding time it is not known what navigation path a user will take, making differential coding very difficult to employ. In this paper, leveraging on a recent work on the merge operator—a new distributed source coding technique that efficiently merges differences among a set of side information (SI) frames into an identical reconstruction—we design redundant frame structures that facilitate ILFS, trading off expected transmission cost with total storage size. Specifically, we first propose a new view interaction model that captures view navigation tendencies of typical users. Assuming a flexible one-frame buffer at the decoder, we then derive a set of recursive equations that compute the expected transmission cost for a navigation lifetime of T views, given the proposed interaction model and a pre-encoded frame structure. Finally, we propose an algorithm that greedily builds a redundant frame structure, minimizing a weighted sum of expected transmission cost and total storage size. Experimental results show that our proposed algorithm generates frame structures with better transmission / storage tradeoffs than competing schemes.

***Index Terms*—** interactive streaming, light field, distributed source coding

1. INTRODUCTION

The advent of *light field* (LF) capturing technologies such as Lytro[1] means that light intensity and direction can now be captured per-pixel using an arrangement of microlenses placed in front of a traditional image sensor. One novel LF application is the generation of a 2D array of densely spaced viewpoint images, which can be navigated freely by a user to create a sense of depth in the captured static 3D scene via motion parallax[2].

However, providing an *interactive LF streaming* (ILFS) service over rate-constrained networks [1–4], where a user successively requests desired views and in response the server transmits pre-encoded data for correct decoding of the requested viewpoint images, is challenging. On one hand, *differential coding*—encoding prediction residuals between a target image and a predictor frame in the decoder buffer—can reduce transmission cost. On the other, at encoding time it is not known what navigation path a user will take, and thus what frames will reside in the decoder buffer for prediction. Thus the challenge is how to employ differential coding for compression efficiency *without* knowing precisely what navigation path a user will eventually take in a ILFS scenario?

In this paper, leveraging on a recent work on the *merge frame* (M-frame) [5, 6]—a new *distributed source coding* (DSC) technique [7] that efficiently merges differences among a set of *side information* (SI) frames into an identical reconstruction—we design redundant frame structures that facilitate ILFS, trading off expected transmission cost with total storage size. Specifically, we first propose a new user interaction model that captures typical view navigation tendencies: a user tends to choose the same navigation direction in consecutive time instants. Assuming a flexible one-frame reference buffer in the decoder, we then derive a set of recursive equations—efficiently computed using *dynamic programming* (DP)—that assess the expected transmission cost for a ILFS session of T view-switches, given the proposed interaction model and a pre-encoded frame structure composed of intra-coded I-, differentially coded P- and M-frames. Finally, we propose an efficient algorithm that greedily builds a redundant frame structure, minimizing a weighted sum of expected transmission cost and total storage size. Using light field data publicly available for the Grand Challenge on Light-Field Image Compression in ICME2016[3], experimental results show that our proposed algorithm generates frame structures with better transmission / storage tradeoffs than competing schemes.

2. RELATED WORK

ILFS was first studied in [1, 2], where the focus was on designing switching mechanisms to adjacent views based on SP-frames [8] and Wyner-Ziv coding. However, the navigation model (which only permits switches to adjacent views) is very limited.

A more general view navigation model was used in [3, 4], where a user can switch to adjacent views *and* pre-defined further-away anchor views, and redundant frame structures were designed for ILFS. DSC frames [7] were used for merging of SI frames. Our work extends from [4], with the following non-trivial differentiations. First, we employ M-frames [5, 6] for merging of SI frames, which have been shown to be more efficient than DSC frames. Second, we capture typical user navigation tendencies—same directional commands in consecutive time instants—in a succinct probabilistic model. Third, we derive a new set of recursive equations to compute the expected transmission cost, given the new user interaction model and a flexible one-frame reference buffer in the decoder.

Redundant frame structures are also used for *interactive multi-view video streaming* (IMVS) [9, 10]. However, in ILFS for static 3D scene, there exist possible loops in a navigation path, making the frame structure design problem more challenging.

[1] https://illum.lytro.com/

[2] http://lightfield.stanford.edu/

[3] http://mmspg.epfl.ch/ICME2016GrandChallenge

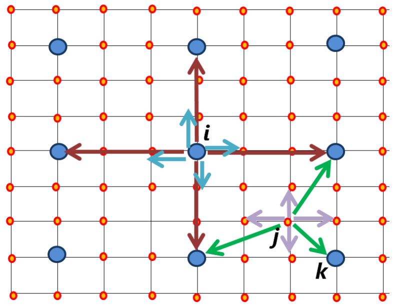


Fig. 1: Example of a fine grid of 9×9 views and a coarse grid of 3×3 views (blue circles). The arrows represent possible view-switches from views i and j.

3. USER INTERACTION MODEL

We propose a *user interaction model*, comprising i) a *view navigation model* which defines permissible view transitions, and ii) a *user behavior model* that describes the likelihood a user selects different permissible view-switches.

3.1. View Navigation Model

For the view navigation model, we assume that a static 3D scene can be observed from a set of N viewpoint images arranged into a $\sqrt{N} \times \sqrt{N}$ 2D grid $\mathcal{G}^f$, called the *fine grid*. From experience using existing static scene view-switching interfaces[2], we notice two possible interaction types: i) switch to an adjacent view, and ii) jump to a distant view for more coarse-grained view sampling. Based on this observation we define the navigation model as follows. Using a set of four local directional commands $\{\mathtt{n}, \mathtt{e}, \mathtt{s}, \mathtt{w}\}$, a user can switch from a fine grid view (i, j) to a vertical $(i \pm 1, j)$ or a horizontal neighbor $(i, j \pm 1)$. This movement called `walk` enables a user to navigate slowly to any available views in a local neighborhood.

We also define a $\sqrt{M} \times \sqrt{M}$ *coarse grid* $\mathcal{G}^c$ on top of $\mathcal{G}^f$, where the views are spaced δ apart. Using a different set of directional commands $\{\mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}\}$, a user can switch from a coarse view (i, j) to a neighboring coarse view $(i \pm \delta, j)$ or $(i, j \pm \delta)$. This movement called `jump` enables a user to navigate quickly from one view neighborhood to another.

A user can also jump from a fine grid view to the nearest upper, right, lower and left coarse grid view using command $\{\mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}\}$. A user at a coarse view can also walk to its adjacent fine grid neighbors using $\{\mathtt{n}, \mathtt{e}, \mathtt{s}, \mathtt{w}\}$. Fig. 1 shows an example of a 3×3 coarse grid (blue circles) on top of a 9×9 fine grid. Possible view-switches from views i and j are illustrated by arrows. Note that from fine grid view j, only *three* closest coarse grid views are accessible with commands $\{\mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}\}$ ($\mathtt{E}$ and $\mathtt{S}$ map to the same view k).

We assume that a user starts a ILFS session at time $t = 0$ at an initial center view s, switches view at each time instant until T switches (called *lifetime*) are performed, upon which he exits the session. A generalization to a probabilistic model for lifetime is straightforward and is left for future work.

3.2. User Behavior Model

We define a user behavior model to assign probabilities to possible view-switches under the described navigation model. Unlike the model in [4] which is memoryless (*i.e.*, the probability $p_{i,j}$ of switching from view i to j does not depend on previous view traversal), we propose a *1-hop memory* model. This means that the probability $p_{k,i,j}$ of switching from current view i to view j depends on previous view k.

Specifically, we define $p_{k,i,j}$ to capture *a user's tendency to select the same navigation direction in consecutive instants*. Denote by $\phi(i, j)$ the direction taken from view i to j; *e.g.*, the direction from view $(1, 1)$ on the fine grid to $(1, 2)$ is $\mathtt{e}$. Denote by q_0 the probability that a user navigates again in the same direction in the fine grid. Denote by q_1 the probability that a user moves from a fine grid to a coarse grid. Assume first that a user is in view i and traversed k previously, where $i, k \in \mathcal{G}^f$. We can define $p_{k,i,j}$ as follows:

$$p_{k,i,j} = \begin{cases} q_1/3 & \text{if } j \in \mathcal{G}^c \\ q_0(1 - q_1) & \text{if } \phi(k, i) = \phi(i, j) \\ (1 - q_0)(1 - q_1)/3 & \text{o.w.} \end{cases} \tag{1}$$

(1) states that a user jumps to each of three permissible coarse grid views with probability $q_1/3$. If the user remains in the fine grid, he will choose the same navigation direction with probability q_0. Otherwise, he will choose the other three adjacent fine grid views with probability $(1 - q_0)/3$.

If $i \in \mathcal{G}^f$ and $k \in \mathcal{G}^c$, then there is no reliable previous direction $\phi(k, i)$, and we assume that the four adjacent fine grid views are equally likely:

$$p_{k,i,j} = \begin{cases} q_1/3 & \text{if } j \in \mathcal{G}^c \\ (1 - q_1)/4 & \text{o.w.} \end{cases} \tag{2}$$

One can similarly define $p_{k,i,j}$ if i is a coarse grid view with corresponding model parameters g_0 and g_1. At the initial view s, we assume that a user navigates to each of the adjacent fine grid view j with probability $p^o_j = 1/4$.

4. ONE-FRAME BUFFER SCHEDULING

We describe the frame types pre-encoded at the encoder and their use to facilitate view-switches. Assuming a flexible one-frame reference buffer at the decoder, we derive recursive equations to compute the expected transmission cost given a frame structure.

4.1. Frame Types in Coding Structure

We use I-frames and P-frames in HEVC [11] for intra- and differentially coded frames. Each view i has one I-frame pre-encoded, denoted as I_i. To facilitate a view-switch from j to i, view i may contain in addition a P-frame $P_i(j)$ using I-frame I_j of view j as predictor.

In general, view i may contain multiple P-frames $P_i(j)$ for view-switches from different views j, and their reconstructions are slightly different due to transform domain quantization of different prediction residuals. To merge their differences to an identical reconstruction, we employ a *merge frame* (M-frame) M_i [5]. In a nutshell, for each quantized transform coefficient $X^i_k(b)$ of frequency k in a block b of I-frame I_i, we encode parameters (c, z) of a floor function $f(x)$, so that the coefficient $X^j_k(b)$ in each reconstructed P-frame $P_i(j)$ is rounded down to the same value, *i.e.*,

$$\left\lfloor \frac{X^i_k(b) - c}{z} \right\rfloor = \left\lfloor \frac{X^j_k(b) - c}{z} \right\rfloor, \ \forall j \text{ s.t. } P_i(j) \text{ exists} \tag{3}$$

Hence the decoding of any $P_i(j)$ with predictor I_j plus M_i will result in an identically reconstructed I_i. Fig. 2 shows a frame structure example for views i and j.

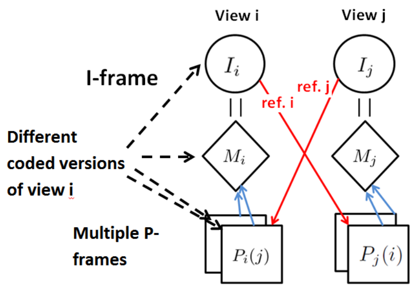


Fig. 2: Example of frame structure for views i and j, with I- (circles), P- (squares) and M-frames (diamonds).

4.2. Flexible One-Frame Reference Buffer

A *flexible one-frame reference buffer* means that, besides a current frame in the display buffer, there is in addition a reference buffer to store one predictor frame. When a user observing view i with frame (view) l in the reference buffer switches to view j, the user can use *either* frame l *or* frame i as predictor to decode P-frame $P_j(l)$ or $P_j(i)$. Essentially, the flexible one-frame buffer enables the user to store *the most valuable reference frame*—one that helps reduce expected transmission cost—as the user navigates the LF data. We assume the size of the reference buffer to be a single frame to keep the to-be-derived recursive equations tractable. In practice a user's reference buffer can be larger, meaning that our computed expected transmission cost is actually an upper bound. Our optimization is thus to minimize a mathematically tractable upper bound of the actual transmission cost.

Using a flexible one-frame buffer, we consider three different transmission types when a user at view i with frame l in the reference buffer switches to view j, each with a different transmission overhead. In *0-hop transmission*, I-frame I_j is transmitted, resulting in an overhead r^I_j. In *1-hop transmission*, P-frame $P_j(i)$ or $P_j(l)$ is transmitted along with M-frame M_j, resulting in overhead $r^P_j(i)$ or $r^P_j(l)$, respectively. In *2-hop transmission*, P-frame $P_\eta(i)$ or $P_\eta(l)$ and M_η are first transmitted to transition to an *intermediate view* η, then P-frame $P_j(\eta)$ and M_j are transmitted to arrive at designation view j. The overhead is thus $r^P_\eta(i)$ or $r^P_\eta(l)$ plus $r^P_j(\eta)$. The purpose of 2-hop transmission is to enable a switch from view i to j without transmitting I-frame I_j, even if P-frames $P_j(i)$ and $P_j(l)$ are not pre-encoded in the structure.

4.3. Computing the Expected Transmission Cost

We now derive equations to compute the expected transmission cost given our proposed user interaction model and a frame structure θ. Denote by $c^{(t)}_{i|k}(l)$ the expected transmission cost from current instant t to lifetime T, given user is at view i and traversed view k just previously, and view l is in the reference buffer. We can write $c^{(t)}_{i|k}(l)$ as:

$$c^{(t)}_{i|k}(l) = \sum_j p_{k,i,j} \min \left[h^{(t)}_i(l,j),\, \dot{h}^{(t)}_i(l,j),\, \ddot{h}^{(t)}_i(l,j) \right] \quad (4)$$

where $h^{(t)}_i(\,)$, $\dot{h}^{(t)}_i(\,)$ and $\ddot{h}^{(t)}_i(\,)$ are the costs of 0-hop, 1-hop and 2-hop transmission, respectively.

The 0-hop transmission cost $h^{(t)}_i(\,)$ is the sum of I-frame cost r^I_j *plus* the recursive cost $c^{(t+1)}_{j|i}(\,)$ if lifetime T has not been reached. The better reference frame between view l and i must be selected for the future. We write $h^{(t)}_i(\,)$:

$$h^{(t)}_i(l,j) = r^I_j + \mathbf{1}(t<T) \min_{\gamma \in \{l,i\}} c^{(t+1)}_{j|i}(\gamma) \quad (5)$$

where $\mathbf{1}(\mathbf{c})$ is an indicator function that equals 1 if clause $\mathbf{c}$ is true and 0 otherwise.

The 1-hop transmission cost is the sum of either P-frame cost $r^P_j(i)$ or $r^P_j(l)$ plus recursive cost $c^{(t+1)}_{j|i}(\,)$. The frame used as predictor to view j will become the new reference in the recursive term. Thus we can write $\dot{h}^{(t)}_i(\,)$:

$$\dot{h}^{(t)}_i(l,j) = \min_{\gamma \in \{l,i\}} \left[r^P_j(\gamma) + \mathbf{1}(t<T)\, c^{(t+1)}_{j|i}(\gamma) \right] \quad (6)$$

We assume here that if P-frame $P_j(\gamma)$ does not exist in structure θ, $r^P_j(\gamma)$ will returns ∞ to signal the violation.

The 2-hop transmission cost is, for an intermediate view η, the sum of either P-frame cost $r^P_\eta(i)$ or $r^P_\eta(l)$, plus P-frame cost $r^P_j(\eta)$, plus recursive cost $c^{(t+1)}_{j|i}(\,)$.

$$\ddot{h}^{(t)}_i(l,j) = \min_\eta \left[r^P_j(\eta) + \mathbf{1}(t<T)\, c^{(t+1)}_{j|i}(\eta) + \min_{\gamma \in \{l,i\}} r^P_\eta(\gamma) \right] \quad (7)$$

Note that view η must be in the reference buffer to decode P-frame $P_j(\eta)$.

Having defined the above, $c^{(0)}_s$ will compute the expected transmission cost starting at the initial view s with an empty reference buffer $\emptyset$. $c^{(0)}_s$ is defined similar to (4):

$$c^{(0)}_s = \sum_j p^o_j \min \left[h^{(0)}_i(\emptyset,j),\, \dot{h}^{(0)}_i(\emptyset,j),\, \ddot{h}^{(0)}_i(\emptyset,j) \right] \quad (8)$$

The computation complexity of computing $c^{(0)}_s$ can be analyzed as follows. Assuming DP tables are used to avoid re-computation of recurring sub-problems, the complexity is bounded by the size of DP tables multiplied by the steps required to compute each entry. The size of DP table is bounded by $O(T8N^2)$ (each view i can have at most 8 different previous traversed views). The steps required to compute (4), (5), (6) and (7) are bounded by $O(N)$. Thus the overall complexity is $O(TN^3)$, which is polynomial time.

5. DESIGNING FRAME STRUCTURE

5.1. Storage Cost

For a given structure θ, the storage cost can be calculated by simply adding up the sizes of all pre-encoded differentials:

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

I- and M-frames are not considred since they are always pre-encoded for each frame in the structure.

Having determined the expected transmission cost and the storage cost for a given structure θ, we next define the optimal frame structure design problem: find a structure θ^* that optimally trades off expected transmission cost and storage for a given weight parameter λ:

$$\min_\theta c^{(0)}_s(\theta) + \lambda b(\theta) \quad (10)$$

5.2. Heuristic Algorithm

To solve (10), we design a greedy algorithm as follows. We initialize our frame structure with pre-encoded I- and M-frames for each view. We then iteratively add the most “beneficial” *single* P-frame (enabling 1-hop transmission) or *pair* of P-frames (enabling 2-hop transmission) that induces the largest decrease in cost function (10). Once no more P-frames can be added to further lower the cost, we exit the algorithm. In order to speed up the algorithm, we employ a simple strategy: we assume that a redundant P-frame closer to the origin is in general more important, so if a P-frame j closer to the initial view is currently sub-optimal to another P-frame candidate k, then P-frames that are further away than j from the initial view and in the same direction will also be sub-optimal and do not need to be tested explicitly. This assumption helps reduce the search complexity for good P-frame additions at each iteration.

6. EXPERIMENTATION

6.1. Experimental Setup

To test the performance of coding structures generated by our algorithm, we downloaded two different light field image sets `swans` and `flowers` from [12][4], where each image is of size 432×624. From this dataset we selected a subset to build a 6×6 fine grid of images, on top of which we built a 2×2 coarse grid. For I- and P-frames, we use HEVC HM-15.0 [11]. Quantization parameters were set so that PSNR of the encoded frames was around 36dB.

Parameters of our user behavior model were set to $q_0 = g_0 = 0.4$ and $q_1 = g_1 = 0.6$. The lifetime of a session T was one third of the number of LF images. We varied λ in (10) to induce different tradeoffs between expected transmission rate and storage.

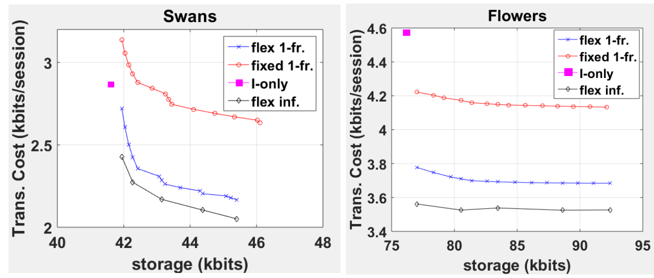
We compare the performance of our generated frame structures (with a flexible 1-frame reference buffer) to three others. The first scheme encodes only one I-frame I_j for each view j. The second uses optimized structures assuming a *fixed* 1-frame reference buffer, meaning that the displayed view is always kept as reference for the next requested view. Finally, the third scheme employs the same optimized structures generated by our optimization, but assumes a reference buffer of *infinite* size during streaming, meaning that any frames previously traversed by a user can be used for referencing. Upon each view request, the server selects the lowest transmission cost option among 0-hop, 1-hop and 2-hop given the content of the user’s reference buffer. Given our user behavior model, we simulated navigation paths of 100 users and then computed the average transmission cost per ILFS session.

Note that if lifetime T is large relative to the number of LF images, then a coding scheme that compresses all LF images without consideration for view interactivity would minimize the average transmission cost per session for a user with infinite buffer size (switching cost per view will be zero when all LF images are in the buffer). Thus we assume here that T is small relative to the number of LF images, which in practice can be very large.

6.2. Experimental Results

In Fig. 3, we see the tradeoff between expected transmission cost per ILFS session and overall storage of the entire frame structures (I-, P- and M-frames). We observe from the curves that for the same storage, using our optimized structures with a flexible 1-frame

[4] This data are part of the ICME16 grand challenge dataset for “light-field image compression”.



(a) (b)

Fig. 3: Expected server transmission rate versus storage size of frame structure for `swans` and `flowers`.

buffer results in an expected transmission cost that is roughly 20% less than that of structures with a fixed 1-frame buffer. This means that though fixed 1-frame scheme is also mathematically tractable (thus amenable to frame structure optimization), employing a flexible 1-frame buffer instead—selectively retaining the most useful reference frame during navigation—has substantial benefit in reducing the exepected transmission cost. Further, we observe that I-frame-only scheme results in relatively small storage sizes compared to our generated redundant frame structures, but has high expected transmission cost. Specifically, using our proposed frame structures with a flexible 1-frame buffer we can reduce transmission cost by about 24% and 37% for `swans` and `flowers`, respectively.

One question in our optimization is the following: given that we assume a flexible 1-frame buffer, how different would the transmission cost be if the user actually has a much larger buffer for referencing? In Fig. 3, we observe that, compared to the infinite buffer case using the same optimized frame structures, our computed expected transmission cost is only 6% higher than the average transmission cost for the infinite buffer case computed via simulation. This shows that our assumption of flexible one-frame buffer to keep the computation tractable is in practice fairly accurate, at least for the lifetime and user behavior model parameters we have chosen in the experiment.

7. CONCLUSION

Designing a frame structure for interactive light field streaming (ILFS) is difficult, because at encoding time it is not known what navigation path a user will take, and hence difficult to employ differential coding to reduce transmission cost. In this paper, we propose to use a recently developed merge frame (M-frame) to merge differences among different differentially coded frames of the same view into one identical reconstruction. We propose a new user interaction model to capture a user’s tendency to navigate in the same direction across multiple time instants. Assuming a flexible one-frame buffer to keep the problem mathematically tractable, we derive a set of recursive equations to compute the expected transmission cost efficiently using dynamic programming (DP). We design an algorithm that finds redundant frame structures, minimizing a weighted sum of expected transmission cost and storage. Experiments show that our frame structures offer good transmission cost / stroage tradeoffs.

8. REFERENCES

[1] 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.

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

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

[4] W. Cai, G. Cheung, S.-J. Lee, and T. Kwon, "Optimal frame structure design using landmarks for interactive light field streaming," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, March 2012.

[5] W. Dai, G. Cheung, N.-M. Cheung, A. Ortega, and O. Au, "Rate-distortion optimized merge frame using piecewise constant functions," in *IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013.

[6] W. Dai, G. Cheung, N.-M. Cheung, A. Ortega, and O. Au, "Merge frame design for video stream switching using piecewise constant functions," in *arXiv:1509.02995 [cs.MM]*, September 2015.

[7] 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.

[8] 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, pp. 637–644.

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

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

[11] Gary J Sullivan, J-R Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[12] Martin Rerabek, Lin Yuan, Léonard Antoine Authier, and Touradj Ebrahimi, "[iso/iec jtc 1/sc 29/wg1 contribution] epfl light-field image dataset," Tech. Rep., ISO/IEC JTC 1/SC 29/WG1, 2015.