OPTIMIZING LANDMARK INSERTIONS FOR INTERACTIVE LIGHT FIELD STREAMING

Yuan Yuan [*], *Gene Cheung* [#], *Pascal Frossard* [$]

[*] University of Alberta, [#] National Institute of Informatics, [$] EPFL

ABSTRACT

Light field imaging enables a user to navigate and observe a static 3D scene from different viewpoints. Downloading the entire data prior to navigation would incur a large startup delay. Instead, previous works propose an interactive light field streaming (ILFS) framework, where a user periodically requests a viewpoint, and in response the server transmits a pre-synthesized and encoded viewpoint image. Using I-frame, P-frame and previously proposed merge frame that facilitates view-switches, the challenge is how to design and pre-encode a storage-constrained frame structure to enable efficient view navigation. In this paper, we initialize "landmarks" into a structure to improve ILFS performance. A landmark is a designated view with P-frames to/from each neighborhood view, so that any viewpoint image can transition to any other viewpoint image by first visiting a landmark, and then from the landmark to the destination view. This results in a transmission cost of only two P-frames. Using a Lloyd's algorithm variant, we first incrementally insert into a frame structure landmarks one at a time at locally optimal locations. We then employ a greedy algorithm to add / subtract P-frames based on a rate-storage criterion. Experimental results show that our proposed structures have noticeably lower expected transmission cost for the same storage than structures generated by a previous greedy algorithm.

Index Terms— Light field imaging, interactive streaming, video coding

1. INTRODUCTION

Light field (LF) cameras such as Lytro[1] employ a 2D array of microlenses before the image sensor to capture multiple light ray intensities and directions per pixel, so that a user can navigate and observe a static 3D scene from different viewpoints post-capture. However, the volume of captured LF data is large, and downloading the entire data prior to user's viewpoint navigation would incur a large startup delay.

Previous works propose an *interactive light field streaming* (ILFS) framework [1–4], where a user periodically requests a desired view, and in response a server transmits a pre-synthesized and encoded viewpoint image for observation. The technical challenge is to design and pre-encode a storage-constrained frame structure to facilitate user-requested view-switches during an ILFS session. Pre-encoding only I-frames for all views would lead to a large transmission cost, while pre-encoding P-frames for all possible user view-switch requests from any view i to j for a LF array of N views would require $O(N^2)$ P-frames and thus expensive in storage cost.

To lower transmission cost while reducing storage requirement, we design new frame structures to facilitate ILFS via optimal selection of *landmarks*. A landmark operates like an airline hub in commercial aviation[2]: by creating direct flights to/from a designated hub for all cities—$O(2N)$ flights for N cities—a passenger can travel from any city to any other city via only two flights (one connecting flight). Similarly, adding P-frames to/from a landmark view means a user's request to switch from any viewpoint image to any other view can be accomplished by decoding two differential P-frames essentially[3]. Hence the number of stored P-frames is only $O(2N)$. The crux is how to select the optimal number and locations of landmark views, and P-frame connections to/from landmarks for the remaining views.

In this paper, we use a Lloyd's algorithm variant [6] to incrementally insert into a frame structure landmarks one at a time at locally optimal locations. We then initialize P-frames connecting the remaining views to/from their closest landmarks. Finally, we greedily add/subtract P-frames from the initialized structure based on a rate-storage criterion. Using two publicly LF datasets, experimental results show that our designed frame structures achieve lower expected transmission cost than greedy structures in [7] for small storage sizes.

The outline of the paper is as follows. We first review related work in Section 2. We discuss our ILFS system and view navigation model in Section 3. We compute the expected ILFS transmission cost in Section 4. We discuss our frame structure design using landmarks in Section 5. Results and conclusion are presented in Section 6 and 7, respectively.

2. RELATED WORK

ILFS is first studied in [1, 2], where new switching mechanisms to adjacent views based on SP-frames [8] and Wyner-Ziv coding are proposed. However, the navigation model (which only permits switches to horizontally and vertically

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

[2] As an example, United Airline has its largest domestic hub in Chicago O'Hare International Airport.

[3] To be discussed in Section 3, an M-frame [5] is also needed after decoding each P-frame to identically merge to a target I-frame reconstruction.

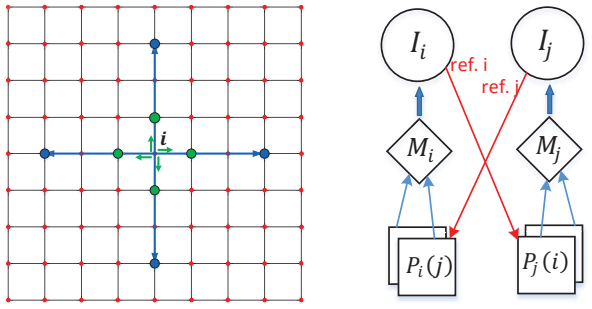
adjacent views) is limited. We discuss our more general view navigation model in Section 3.

More recent studies [3, 7] have employed a more general view navigation model for ILFS, but the focuses are on the use of new *distributed source coding* (DSC) frames [9] and merge frames [5] for view-switching without coding drift. Orthogonally, [10] partitions the 3D scene into segments for efficient coding and streaming. In contrast, we focus on the optimal insertion of landmarks to facilitate ILFS.

Redundant frame structures like the ones generated by our design are also used for *interactive multiview video streaming* (IMVS) [11–13]. However, in ILFS for static 3D scene, there exist possible loops in a navigation path, making the frame structure design problem more challenging.

3. SYSTEM OVERVIEW

We first present a view navigation model for ILFS that describes a typical user's view-switching behavior. We then present I-, P- and merge (M-) frames in our coding structure to facilitate a user's view-switching requests.



(a) (b)

Fig. 1. (a) Example of 2D grid of 9×9 views. The green and blue arrows respectively represent possible view-switching `walk` or `jump` from view i with $K = 3$. (b) Example for frame types of view i and j, with I-(circle), P-(square) and M-frames (diamonds).

3.1. View Navigation Model for ILFS

We assume that the N viewpoints of a static scene captured by a LF camera are arranged into a $\sqrt{N} \times \sqrt{N}$ 2D grid. Common LF user interfaces[4] allow different kinds of view-switches, which we condense into two types: i) switch to a horizontally / vertically adjacent view; ii) jump to a horizontally / vertically view that are K views apart, where K is a constant.

Specifically, at a viewpoint (x, y), one can choose either `walk` or `jump` to a next view. `walk` means that a user can switch from view (x, y) to a vertically $(x \pm 1, y)$ or a horizontally $(x, y \pm 1)$ adjacent view. By `jump`, we mean a user can switch to a view K distance apart, *i.e.* from view (x, y), one can switch to views $(x \pm K, y)$ or $(x, y \pm K)$. An example of view switching with $K = 3$ is shown in Fig. 1(a).

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

We assume that the probabilities of switching to adjacent or distant views are the same. Thus, from view i, a user can switch to a view j with probability $p_{j|i} = \frac{1}{8}$.

3.2. Frame Types in Coding Structure

In our proposed frame structure, by default we assume that each view i has one pre-encoded I-frame, denoted by I_i, which ensures that a server can always enable a user to switch to view i by transmitting I_i, albeit at a large transmission cost. To more efficiently facilitate a view-switch from j to i, view i may contain in addition a P-frame $P_i(j)$, which uses I-frame I_j of view j as predictor. Thus $P_i(j)$ is transmitted only if the user has I_j in the buffer.

In general, view i may contain multiple P-frames $P_i(j)$ to facilitate switches from different views j, and their reconstructions differ slightly due to transform domain quantization of different prediction residuals. To merge their differences to an identical reconstruction (thus avoiding future coding drift), we employ a *merge frame* (M-frame) M_i [5]. M-frame is a new DSC design that uses shift and rounding operations to achieve desired merging results; see [5] for details. An M_i *plus* any decoded $P_i(j)$ will result in an identically reconstructed I_i. See Fig. 1(b) for an illustration. Like I_i, M_i is also pre-encoded by default in our structure.

When a user requests switching from view j to i, the server can either transmit an I-frame I_i, or an M-frame M_i *plus* a P-frame $P_i(j)$. The technical challenge is how to select an appropriate set of P-frames for pre-encoding in the structure to facilitate user's view-switch requests when the storage size is limited. We focus on this problem in the sequel.

4. EXPECTED TRANSMISSION COST

For a given frame structure θ, we can compute the expected transmission cost of an ILFS session using a set of recursive equations. To keep the computation tractable, we first describe a *flexible* one-frame reference buffer model.

4.1. Flexible 1-frame Reference Buffer

A flexible one-frame reference buffer means that, besides a current frame in the display buffer for user observation, there is in addition a reference buffer to store one additional 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 i *or* frame l as predictor to decode P-frame $P_j(i)$ or $P_j(l)$ for view switching. It means that, if there is a landmark view l in the user's reference buffer, using the flexible one-frame buffer a user can switch from view i to j by directly decoding $P_j(l)$ without first decoding $P_l(i)$ to reconstruct the landmark view.

4.2. Expected Transmission Cost

Using a flexible one-frame buffer, we consider three different transmission types during a view-switch: *0-hop*, *1-hop* and *2-hop* transmissions. *0-hop transmission* means that an I-frame

I_j is transmitted for the requested view j, resulting in an overhead $r_j^I = |I_j|$, where $|\cdot|$ stands for the coding rate of a frame. *1-hop transmission* means that a P-frame $P_j(l)$ or $P_j(i)$ is transmitted with an M-frame M_j, resulting in an overhead $r_j^P(l)$ or $r_j^P(i)$, respectively, where $r_j^P(l) = |P_j(l)| + |M_j|$. *2-hop transmission* means that a P-frame $P_\eta(l)$ or $P_\eta(i)$ is transmitted along with an M-frame M_η to transition to an *intermediate view* η, then P-frame $P_j(\eta)$ and M_j are transmitted to arrive at the designation view j. 2-hop transmission enables a user to first switch to a landmark, which may facilitate future view-switching.

Assume that the *lifetime* of an ILFS session is exactly T view-switches. Denote by $c_i^{(t)}(l)$ the expected transmission cost from current instant t to lifetime T, given a user is at view i and with view l in the reference buffer. We can write $c_i^{(t)}(l)$ as:

$$c_i^{(t)}(l) = \sum_j p_{j|i} \ \min\left[h_i^{(t)}(l,j), \dot{h}_i^{(t)}(l,j), \ddot{h}_i^{(t)}(l,j)\right], \quad (1)$$

where $p_{j|i}$ is the transmission probability from view i to view j, and $h_i^{(t)}()$, $\dot{h}_i^{(t)}()$ and $\ddot{h}_i^{(t)}()$ are the costs of 0-hop, 1-hop and 2-hop transmissions, respectively.

The 0-hop transmission cost $h_i^{(t)}()$ is the sum of r_j^I *plus* the recursive cost $c_j^{(t+1)}()$ if lifetime T has not been reached. The more optimal reference frame between views l and i must be selected for the future. We write $h_i^{(t)}()$ as

$$h_i^{(t)}(l,j) = r_j^I + \mathbf{1}(t<T) \min_{\gamma\in\{l,i\}} c_j^{(t+1)}(\gamma), \quad (2)$$

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

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

$$\dot{h}_i^{(t)}(l,j) = \min_{\gamma\in\{l,i\}} \left[r_j^P(\gamma) + \mathbf{1}(t<T) c_j^{(t+1)}(\gamma)\right]. \quad (3)$$

We define $r_j^P(\gamma) = \infty$ to signal a violation if P-frame $P_j(\gamma)$ is not in the structure θ.

The 2-hop transmission cost is, for an intermediate view η, the sum of either P-frame cost $r_\eta^P(l)$ or $r_\eta^P(i)$, plus P-frame cost $r_j^P(\eta)$, plus recursive cost $c_j^{(t+1)}()$.

$$\ddot{h}_i^{(t)}(l,j) = \min_\eta \left[r_j^P(\eta) + \mathbf{1}(t<T) c_j^{(t+1)}(\eta) + \min_{\gamma\in\{l,i\}} r_\eta^P(\gamma)\right] \quad (4)$$

Having defined the above, $c_s^{(0)}(\emptyset)$ will compute the expected transmission cost starting from an initial view s with an empty reference buffer $\emptyset$.

5. FRAME STRUCTURE DESIGN

5.1. Problem Formulation

Given I- and P- frames coded by HEVC [14] and M-frames coded by [5]—all at a sufficiently fine pre-fixed quantization parameter (QP) for a target image quality—our problem is to determine which P-frames to be differentially encoded *a priori* at the server to minimize the expected transmission cost given a storage constraint. For a given structure θ, we define the storage cost $b(\theta)$ as the total coding rate of all the pre-encoded differential P-frames:

$$b(\theta) = \sum_{P_j(i)\in\theta} |P_j(i)| \quad (5)$$

I- and M-frames are not considered since they are pre-encoded by default for each view in the structure.

Having defined the expected transmission cost and the storage cost for a given structure θ, we use Lagrangian relaxation to find the frame structure θ^* that optimally trades off the expected transmission cost and the storage cost, *i.e.*,

$$\theta^* = \arg\min_\theta c(\theta) + \lambda b(\theta). \quad (6)$$

where λ is a given weight parameter. $c(\theta) = c_s^{(0)}(\emptyset)$ is the expected transmission cost computed by Eq. (1) given θ.

5.2. Structure Design Algorithm

To minimize (6), we first initialize structure θ with "landmarks". Similar to an airline hub in functionality, by adding P-frames to/from a landmark view for all neighboring views, any view can transition to any other view by decoding essentially only two P-frames (transition to landmark, then to designation view). Further, having multiple landmarks means that the two P-frames used to arrive at/depart from the landmark can be smaller; however, the transition between two views connected to two different landmarks can become costly. The challenge then is to identify the appropriate number and locations of landmarks.

We iteratively add landmarks one at a time as follows. We first add a single landmark l_o that minimizes a cost function $f(\Psi_o, l_o)$, where Ψ_o stands for the entire LF array and $l_o \in \Psi_o$. At each iteration, given each landmark is associated with a neighborhood of views or a *partition*, we select an existing landmark l and its partition Ψ that result in the largest cost $f(\Psi, l)$ to be split into two sub-partitions Ψ_1 and Ψ_2 (with corresponding landmarks $l_1 \in \Psi_1$ and $l_2 \in \Psi_2$), where $\Psi = \Psi_1 \cup \Psi_2$. Hence the number of landmarks in the structure increases by one. We use the Lloyd's algorithm [6] to refine the splitting until, after assigning P-frames to/from the closest landmarks for all views in Ψ, the objective (6) no more decreases. We stop adding landmarks if the number of landmarks reaches an empirical threshold M.

One choice of the cost function $f(\cdot)$ to be minimized during splitting is a version of (6) computed for views in a partition Ψ. However, computing expected transmission cost is

costly. Hence, we define a simpler cost function $f(\Psi, l)$ for view partition Ψ as follow:

$$f(\Psi, l) = \sum_{i \in \Psi} \left(|P_i(l)| + |P_l(i)| \right). \quad (7)$$

$f(\Psi, l)$ not only stands for the initialized P-frame storage cost, but also represents the total one-step transmission cost to/from landmark l for partition Ψ. The definition also guarantees $f(\Psi_1, l_1) + f(\Psi_2, l_2) < f(\Psi, l)$.

To choose the best landmark l_1 and l_2 and sub-partitions Ψ_1 and Ψ_2 during splitting that minimize $f(\Psi_1, l_1) + f(\Psi_1, l_1)$, we use the Lloyd's algorithm [6] that iterates between two alternating steps until convergence. First, given sub-partitions Ψ_i are fixed, we find each locally optimal landmark l_i by minimizing the following:

$$l_i = \arg\min_{l \in \Psi_i} \sum_{j \in \Psi_i} \left(|P_j(l)| + |P_l(j)| \right). \quad (8)$$

In words, we identify the landmark view l_i that minimizes the total distance between l_i and any view j in partition Ψ_i in terms of P-frame sizes between the two views.

Second, given landmarks l_1 and l_2 are fixed, we assign each view j in partition Ψ to the closer of the two landmarks:

$$z = \arg\min_{i \in \{1,2\}} \left(|P_j(l_i)| + |P_{l_i}(j)| \right), \quad (9)$$

where z is the partition to which view j is assigned. We iteratively solve (8) and (9) until convergence. Empirical data show that the Lloyd's algorithm can converge quickly.

6. EXPERIMENTATION

6.1. Experiment Setup

To test the performance of our designed structures using landmarks, we downloaded two LF image sets `Swans` and `Flowers` from [15][5]. We selected a subset of 9×9 2D grid of images for each set, where each image is of size 432×624. We used HEVC HM 15.0 [14] to code I- and P-frames, and used [5] to code M-frames. Quantization parameters was set so that PSNR of the encoded frames were around 36dB.

We compare our proposed method with a greedy algorithm proposed in [7], where a locally optimal single P-frame or pair of P-frames are iteratively added to the structure at a time to reduce the objective function in (6). Both these two methods used a flexible 1-frame reference buffer and the lifetime T of a session was set to one third of the number of LF images. We varied λ in (6) to induce different tradeoffs between the expected transmission and storage costs.

6.2. Experiment Results

Fig. 2 shows plots of expected transmission cost versus the storage cost for our designed structures (red) and the greedy

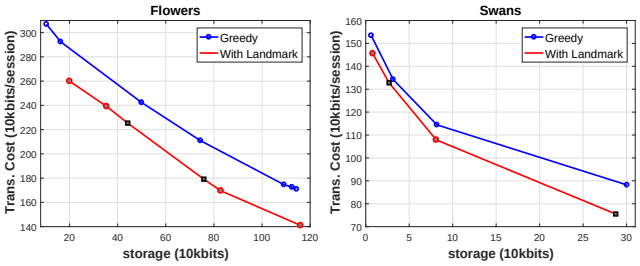


Fig. 2. Expected transmission cost versus storage size of frame structure for `Flowers` and `Swans`.

structures [7] (blue). To reach the same expected transmission cost, our proposed method can save 31.92% and 28.61% storage cost compared to the greedy algorithm for `Flowers` and `Swans`, respectively [16]. This is because, in our method, a landmark is always preferred to be stored in the reference buffer. Thus only one P-frame connected from the landmark to the designated view is needed for each view-switch. Even when views switch across partition boundaries, the 2-hop transmission can enable switching from current landmark to a new landmark. However, the greedy algorithm does not generate landmarks in the frame structure by considering one frame at a time. This is because a landmark view is not useful in reducing the objective until a sufficient number of P-frames to/from neighboring frames are added. The complexity of our proposed method is also much lower than the greedy algorithm, since we have already added enough P-frames into the structure and only a minor adjustment to subtract/add P-frames is required during the greedy step.

On the red curve in Fig. 2, the black square dots correspond to structures with 2 landmarks, where the others correspond to structures with 1 landmark. Although multiple landmarks may increase the transmission cost to deal with partition boundaries, the size of P-frames connecting a landmark to its neighboring views can be smaller, which decreases the storage cost and the transmission cost within a partition. The total cost saving on smaller P-frames will outweighs the increased transmission cost with proper selection of multiple landmarks.

7. CONCLUSION

To efficiently facilitate interactive light field streaming (ILFS), we design a frame structure composed of pre-encoded I-, P- and merge frames using the idea of landmarks: for all neighboring views, adding P-frames to/from a landmark view means that any view can transition to any other view in the neighborhood by decoding only two P-frames. We use a variant of the Lloyd's algorithm to recursively locate landmark views, then greedily add/subtract P-frames subject to a rate-distortion criteria. Experimental results show that our designed frame structures have lower expected transmission costs than structures from a previous proposal [7].

[5] http://mmspg.epfl.ch/EPFL-light-field-image-dataset

8. REFERENCES

[1] 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.

[2] 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.

[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 *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1445–1448.

[5] W. Dai, G. Cheung, N.-M. Cheung, A. Ortega, and O. C. Au, "Merge frame design for video stream switching using piecewise constant functions," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3489–3504, 2016.

[6] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[7] B. Motz, G. Cheung, and A. Ortega, "Redundant frame structure using m-frame for interactive light field streaming," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1369–1373.

[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*, vol. 13, no.7, July 2003, pp. 637–644.

[9] 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.

[10] T. Maugey, I. Daribo, G. Cheung, and P. Frossard, "Navigation domain partitioning for interactive multiview imaging," in *Special Issue on 3D Video Representation, Compression, Rendering, IEEE Transactions on Image Procesing*, vol. 22, no.9, September 2013, pp. 3459–3472.

[11] 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.

[12] ——, "Interactive streaming of stored multiview video using redundant frame structures," in *IEEE Transactions on Image Processing*, vol. 20, no.3, March 2011, pp. 744–761.

[13] B. Motz, G. Cheung, and N.-M. Cheung, "Designing coding structures with merge frames for interactive multiview video streaming," in *22nd International Packet Video Workshop*, Seattle, WA, July 2016.

[14] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. 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.

[15] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, no. EPFL-CONF-218363, 2016.

[16] G. Bjontegaard, "Calcuation of average psnr differences between rd-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April 2001*, 2001.