Light Field Streaming Study Report

Visby Camera Corp.

February 3, 2020

Introduction

Visby Camera Corporation (Visby) has delivered an API to its light field rendering system to Charter Communications; the two together have demonstrated the playback of a video light field into a positionally-tracked virtual reality headset over a DOCSIS network.

Visby has made additional investigations into the light field streaming system to understand how such systems can and will be best deployed in the wild in order to deliver compelling visual experiences given realistic network conditions and other constraints. We specifically are interested in conveying knowledge and understanding of light field streaming over DOCSIS and related learnings: knowledge that will inform how future streaming protocols, networks, etc. are built; and justify further investigations into light field streaming.

For example, consider the network schematic in Figure 1: a light field payload is stored on Content Server. The payload is transmitted to Renderer, which, in real time, renders and serves requested views to Client (e.g., headset or holographic panel) for display. Knowledge and/or control over one link may be levered, e.g., to mitigate deficiencies in the other link. Additionally, the global behavior of the system imposes constraints on the network that can stream light fields successfully.

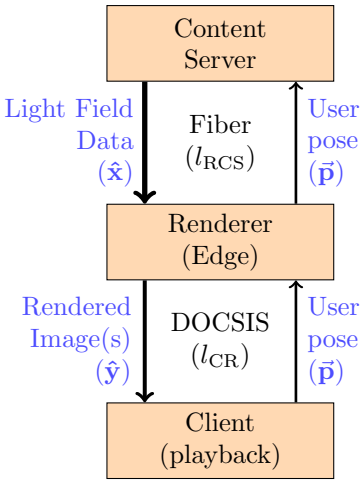


Figure 1: Diagram of potential light field streaming system architecture, separated into Content Server, Renderer and Client components.

The specific studies in this Report use internal Visby light field data sets empirically to explore some fast and implementable light field data decimation strategies. The motivation is to describe and assess practical ways to reduce data transfer bandwidth from Content Server to Renderer, and to understand their ramifications on other salient system variables, such as system latency and quality of User experience.

0.1 Definitions

In the following discussions, we attempt to use the following terminology and notation consistently:

- BPP: "Bits-per-pixel" is a commonly used rate measurement in image compression. Unless otherwise stated, "pixel" means a rendered output pixel, and "bits" means the sum total of bits required to render that output pixel.

- $\mathbf{v}$: a vector.
- $\mathbf{v}_i$: an element of $\mathbf{v}$.
- $|\mathbf{v}|$: the number of elements in $\mathbf{v}$.
- $\mathbf{x}$ and $\hat{\mathbf{x}}$: the uncompressed, full light field, and its sub-sampled counterpart, respectively.
- $\mathbf{y}$ and $\hat{\mathbf{y}}$: an image rendered from $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively.
- Renderer: responsible for converting $\hat{\mathbf{x}}$ into $\hat{\mathbf{y}}$.
- Content Server: responsible for converting $\mathbf{x}$ into $\hat{\mathbf{x}}$.
- Client: responsible for sending frame requests to the Renderer and receiving responses ($\hat{\mathbf{y}}$).
- User: the end user of a head-mounted device connected to the Client.
- Stream: an atomic, independent constituent of an light field source, e.g. a feed or encoding from a single camera of a multi-camera light field acquisition.
- Latency: the time it takes data to travel from one node to another in a network.

0.2 Experimental Clarifications

In the sections that follow, we compute BPP via Eq. 1:

$$\mathrm{BPP}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = 32 \times \frac{|\hat{\mathbf{x}}|}{|\hat{\mathbf{y}}|} \tag{1}$$

This Report (and BPP calculations herein) focuses on uncompressed payloads, though the results can be extrapolated with assumptions about further image or video compression. We assume each rendered frame is a four-channel image with color (RGB) and alpha-mask (A) components.

Our results are derived from two experiences, each with its own subject matter: Ralphie (a dog) and Michael (a man).

1 Lossless Compression: Stream Selection

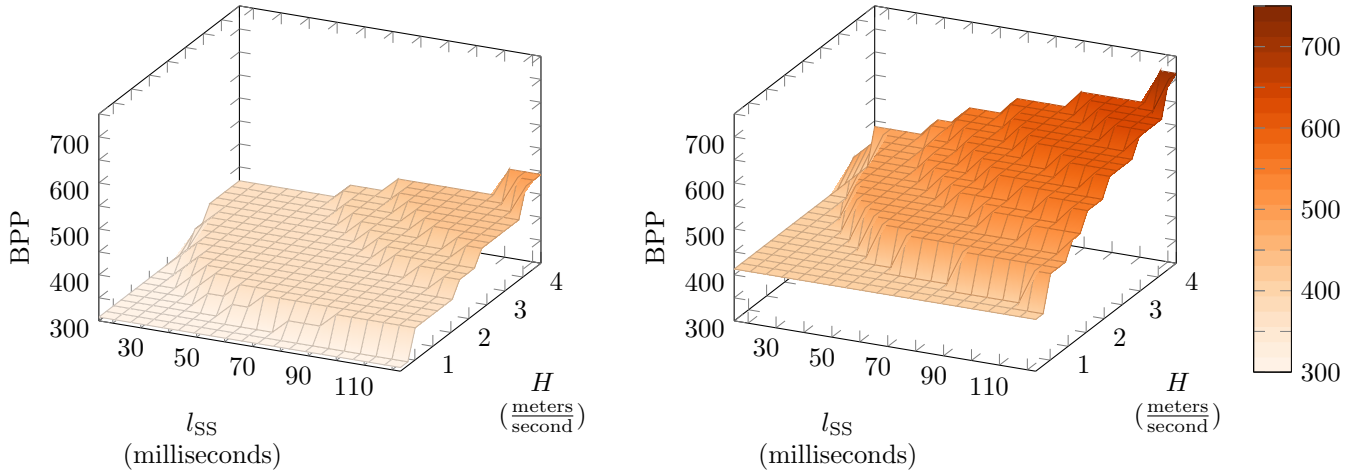
One approach for limiting the bandwidth of transmitted content between Content Server and Renderer nodes in Fig. 1 is to limit the sampling rate of $\mathbf{x}$. We propose and implement a strategy whereby the set of independent light field data streams S comprising $\mathbf{x}$ is sub-selected as a function of User position. We define this quantity as $\vec{\mathbf{p}}(t)$, a 3-vector specifying Cartesian coordinates, which varies with time t. The transmitted content at time t, $\hat{\mathbf{x}}(t)$, would be comprised of $S(\vec{\mathbf{p}}(t)) \subseteq S$. In this analysis, we have not assumed a bound on the User's speed of head rotation.

1.1 Experiments

We conduct a number of numerical experiments using the Ralphie and Michael experiences, whereby $S(\vec{\mathbf{p}}(t))$ is measured at a regular (0.5s) sampling interval in a typical local streaming playback session. For each experiment, the trajectory $\vec{\mathbf{p}}(t)$ spans a typical user viewing box of 2m $\times$ 2m $\times$ 1m in depth, width, and height, respectively, for 100s. We make ensure that the User's head orientation is such that the experience subject is view-centered.

First, we assume that the minimum number of streams available to the Renderer must accommodate the set of streams required at time t_0 as well as any new streams that would be required by the time the request for a stream set finally reaches the Renderer, t_1, at which point the User's position may have changed. We assume that the difference between t_1 and t_0 is l_{SS}:

$$l_{\mathrm{SS}} = l_{\mathrm{CR}} + 2l_{\mathrm{RCS}} \tag{2}$$



(a) Ralphie experience.

(b) Michael experience.

Figure 2: Dependency of BPP on latency l_{SS} and User speed H.

where l_{CR} is the latency between Client and Renderer, and l_{RCS} is the latency between Renderer and Content Server.

Second, we assume that a User's head speed, H, is constant over the latency interval t_0 to t_1, and therefore all sub-intervals of the latency interval.

With an assumed H and l_{SS}, we can use the sampled data to construct scenarios that maximize the number of required streams during the latency interval. Formally, we find an approximate solution to Eq. 3, noting that finding an exact solution is computationally intractable[1]. Our approximation is found by greedily constructing a path traversal through the nodes $S(\vec{\mathbf{p}}(t))$. This path should simultaneously maximize the total size of the set of streams required, and should not exceed the allowed time l_{SS}.

$$S(H, l_{\text{SS}})_{\max} = \max_{T} \left| \bigcup_{i \in T} S(\vec{\mathbf{p}}_i) \right| \tag{3}$$
$$\text{subject to } \frac{1}{H} \sum_{i \in T_{-1}} ||\vec{\mathbf{p}}_i - \vec{\mathbf{p}}_{\text{succ}(i)}||_2 \leq l_{\text{SS}}$$

Here T is an index list representing an induced subpath on the set of sampled positions $\vec{\mathbf{p}}_i$, succ(·) represents a successor of an item in a list, and T_{-1} is the same as T with the last index removed.

For each (l_{SS}, H) pair over a sensible[2] range of values of l_{SS} and H, we approximate the solution $S(H, l_{\text{SS}})_{\max}$ to Eq. 3, and use the BPP calculation in Eq. 1, with $|\hat{\mathbf{x}}| = \frac{|S(H, l_{\text{SS}})_{\max}|}{|S|} \times |\mathbf{x}|$. The results of these experiments are plotted in Fig. 2.

1.2 Discussion

In Fig. 2, we see that the data rate required increases with latency and head speed. Equally unsurprisingly, for a fixed data rate, latency and head speed must vary inversely.

[1]Finding an optimal T is equivalent to finding the longest path in an undirected graph, which is NP-hard.

[2]We assume a round trip traversal of the width of the viewing box in 1 second and use the minimum and maximum from a publicly available Internet latency table[1].

One interesting insight from this analysis is that the range of data rates can vary dramatically depending on the experience: $\sim 300-500$ BPP for Ralphie, and $\sim 400-750$ BPP for Michael. This is due to sharp geometric differences between the respective experiences, the acquiring camera rig geometry, and the relative positioning of experience subject and camera rig. Such prior knowledge can be used to allocate bandwidth as a function of the User-selected experience. If the assumptions about latency and head speed are correct, this data decimation strategy manifestly does not affect the quality of the rendered output.

Another potential use of this data in the context of adaptive streaming is the ability to define bandwidth regimes. That is, if it is known that, for a given experience, a User limits her head speed to a fixed upper bound, and her Client's latency to a Content Server can be bounded, one may potentially operate in one of the BPP regimes shown as isosurfaces in Fig. 2.

2 Lossy Compression: Sub-sampling

Here we present the result of employing a number of hybrid sub-sampling strategies of the input light fields corresponding to the Ralphie and Michael experiences. The goal of this analysis is to convey a sense of the impact of the resultant data rate on the quality of the rendered output.

2.1 Experiments

For each input light field dataset, we compute an image quality metric which quantifies the agreement between the rendered output under sub-sampled conditions and the same output without sub-sampling the input. We call this latter output "ground truth." For each sub-sampling scheme, we target one of the output resolutions shown in Table 1.

Table 1: Output resolutions and corresponding commercial head-mounted displays.

Resolution	Headset(s)
1080x1200	HTC Vive, Oculus Rift
1280x1440	Oculus Rift S
1440x1600	HTC Vive Pro, Oculus Quest

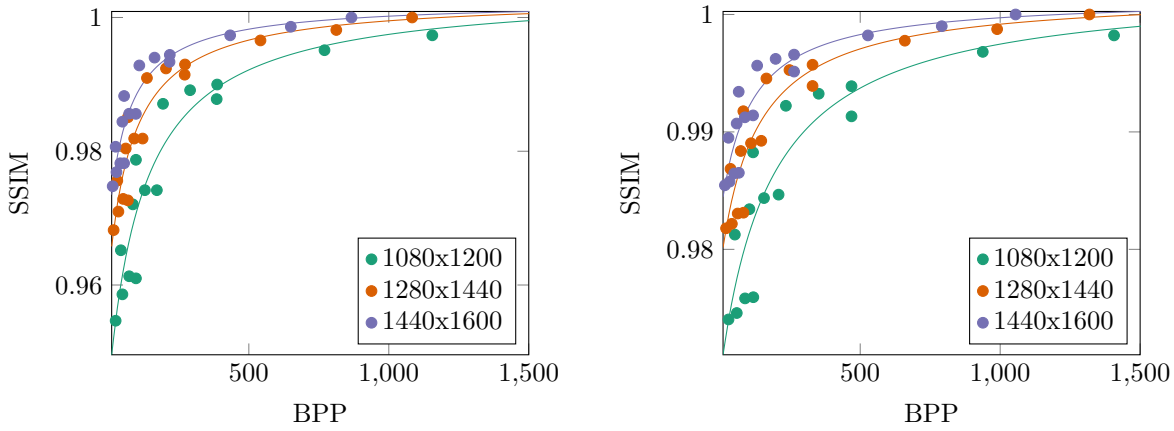
The image quality metrics are SSIM and MPSNR, defined as follows:

$$\mathrm{SSIM}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{M} \sum_{i=1}^{M} \frac{(2\mu_{y,i}\mu_{\hat{y},i} + C_1) + (2\sigma_{y\hat{y},i} + C_2)}{(\mu_{y,i}^2 + \mu_{\hat{y},i}^2 + C_1)(\sigma_{y,i}^2 + \sigma_{\hat{y},i}^2 + C_2)} \tag{4}$$

where the subscripts y, i and $\hat{y}, i$ are used to designate the i-th corresponding $K \times K$ windows in $\mathbf{y}$ and $\hat{\mathbf{y}}$, respectively, and M is the number of windows. For the i-th $K \times K$ window in $\mathbf{y}$, $\mu_{y,i}$ is the average and $\sigma_{y,i}^2$ is the variance. We denote $\sigma_{y\hat{y},i}$ as the cross-correlation between the corresponding windows in $\mathbf{y}$ and $\hat{\mathbf{y}}$. We define the constants $C_1 = (k_1 \mathrm{I}_{\max})^2$ and $C_2 = (k_2 \mathrm{I}_{\max})^2$, where $\mathrm{I}_{\max}$ is the maximum possible pixel value (255 for 8-bit images), and $k_1 = 0.01$ and $k_2 = 0.03$.

SSIM computed against light field renders tend to be quite high. This is due to the presence of a relatively small ratio of foreground to background pixels, and nearly empty (black) background, in the rendered output image. We therefore suggest using an additional measure for comparisons: a masked version of PSNR, MPSNR. This is defined in Eq. 5:

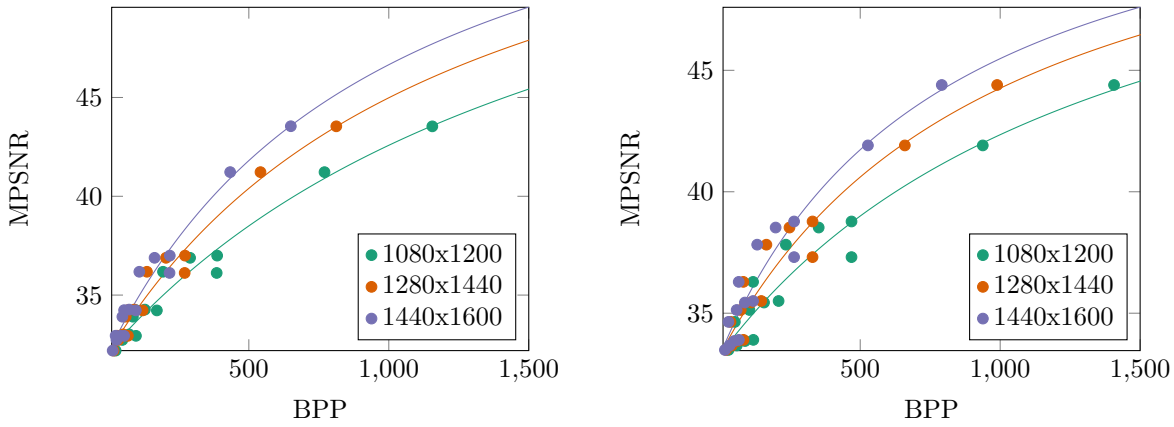
$$\mathrm{MPSNR}(\mathbf{y}, \hat{\mathbf{y}}) = 10 \log_{10} \left(\frac{{\mathrm{I}_{\max}}^2}{\mathrm{MMSE}(\mathbf{y}, \hat{\mathbf{y}})} \right) \tag{5}$$



(a) Ralphie experience.

(b) Michael experience.

Figure 3: Rate Distortion: SSIM vs BPP for varying output resolutions.



(a) Ralphie experience.

(b) Michael experience.

Figure 4: Rate Distortion: MPSNR vs BPP for varying output resolutions.

where MMSE is the masked mean squared error, defined in Eq. 6:

$$\text{MMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N_y} \sum_{i=1}^{N_y} (\mathbf{m}_i \odot (\mathbf{y}_i - \hat{\mathbf{y}}_i))^2 \tag{6}$$

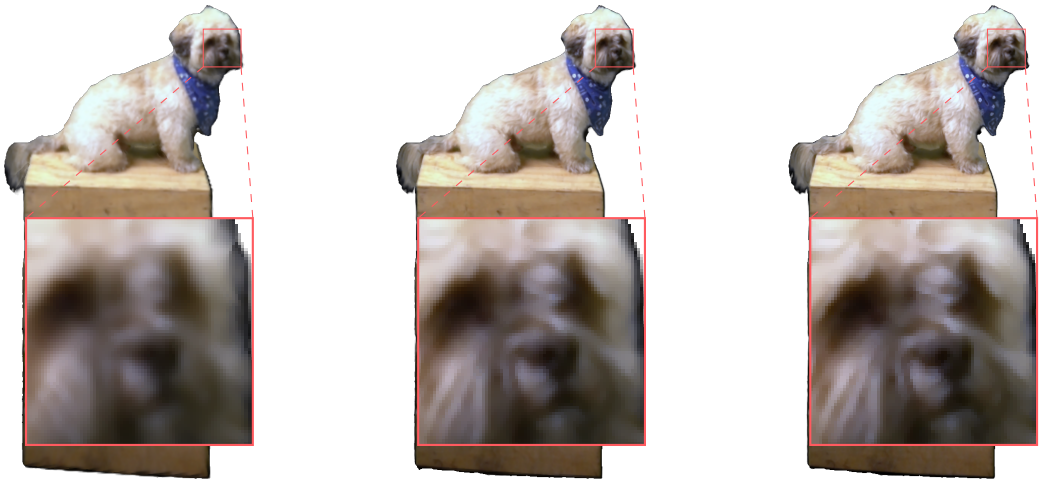
Here, $\mathbf{m}$ is a binary mask representing all non-zero alpha regions in $\mathbf{y}$, N_y is the number of non-zero elements in $\mathbf{m}$, and the masking operation $\mathbf{m} \odot \mathbf{y}$ is the Hadamard product between $\mathbf{m}$ and $\mathbf{y}$.

For each sub-sampled input light field $\hat{\mathbf{x}}$ and its corresponding render $\hat{\mathbf{y}}$, we compute BPP according to Eq. 1. To eliminate bias in the computation of our metric, we take the average across a uniform sampling of viewpoints of the scene along a plane at a nominal distance (say $z = 0$) from the subject.

2.2 Discussion

In Fig. 3, we see that image quality generally increases as the rate increases. The inflection point across all curves seems to be around 300-600 BPP. We believe this should generalize to other experience with similar configurations to Ralphie and Michael.

In Fig. 4 we see the same trend. Also, MPSNR is not trivially close to its maximum value across all values of BPP, as it is for SSIM in Fig. 3. MPSNR may be a better metric to use.



(a) Low: $\sim$ 30 BPP. (b) Medium: $\sim$ 350 BPP. (c) High: $\sim$ 1400 BPP.

Figure 5: Ralphie experience rendered outputs at 1080x1200 resolution with varying levels of sub-sampling applied.

(a) Low: $\sim$ 30 BPP. (b) Medium: $\sim$ 350 BPP. (c) High: $\sim$ 1400 BPP.

Figure 6: Michael experience rendered outputs at 1080x1200 resolution with varying levels of sub-sampling applied.

Figs. 5 and 6 show a few sample renders corresponding to points along the curves in Figs. 3 and 4. The gains in perceptual quality from "Low" to "Medium" sub-sampling seem to be higher than from "Medium"

to "High," suggesting diminishing returns in allocating more bits per output pixel for a given output render size.

3 Future Work

In this section, we summarize and suggest improvements to the findings of this Report.

In Sec. 1, we found that online sub-selecting streams of a light field source can reduce bandwidth requirements by more than 50%, without any loss in quality. Here, we assumed that the switching cost of such an adaptive streaming approach would be negligible in comparison to the network latencies. We also assumed that keeping sub-selected streams in sync with un-selected streams incurs no latency cost. A more accurate analysis would explicitly model these variables.

In Sec. 2, we found that deterministic sub-sampling of the light field source can also reduce bandwidth requirements by up to twofold, at a substantial reduction in visual fidelity. Along this frontier, we may find the best trade-off between distortion and rate by pre-computing rate-distortion curves and utilizing this information to respond to variable network conditions. A promising extension of this work would be to incorporate the online stream-selection strategy in Sec. 1 with the strategies employed here.

For both Secs. 1 and 2, it would be beneficial to examine the bandwidth reduction strategies in the context of light field video and under the assumption of a compressed input. Online (de)compression of the input light field source could dramatically reduce bitrates, but would incur some data fidelity loss along with system latencies in excess of what has been discussed in this Report.

References

[1] ATT. *Global IP Network Latency*. Available at `https://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html` (2020/1/29).