# PROGRESSIVE COMMUNICATION FOR INTERACTIVE LIGHT FIELD IMAGE DATA STREAMING

*Eduardo Peixoto* [*], *Bruno Macchiavello* [*], *Edson Mintsu Hung* [*], *Camilo Dorea* [*], *Gene Cheung* [#]

[*] University of Brasilia, [#] National Institute of Informatics

## ABSTRACT

Light field (LF) imaging captures multiple intensities and directions of light per pixel during acquisition in a 3D scene, so that novel images of different viewpoints or focal points can be synthesized. However, transmitting all LF data before viewer observation incurs a large startup delay. To avoid such delay, we propose a new interactive LF streaming framework, where a client periodically requests viewpoint images, and in response a server synthesizes and transmits each requested image as a carefully chosen sparse linear combination of sub-aperture images. For each received synthesized image, the client "decodes" and recovers a new sub-aperture image using a cache of known sub-aperture images. As the cache of decoded sub-aperture images grows over time, the client becomes capable of synthesizing new view/focal point images, reducing overall transmission cost. Experimental results show that our proposed scheme can deliver synthesized images at high quality, even though only sparse sub-aperture images are used for synthesis. Moreover, compared with a scenario where the requested synthesized images are always transmitted, our proposed scheme achieves a significant reduction in accumulated rate when a sufficient number of images are transmitted.

***Index Terms***— Light field imaging, interactive streaming, network coding

## 1. INTRODUCTION

The advent of light field (LF) sensing technologies like Lytro[1] has enabled the capturing of multiple light rays from different directions for each pixel in an image. The acquired LF data are demosaicked, calibrated and mapped to a series of sub-aperture images, which can be used to synthesize novel images of different viewpoints or different focal points post-capture [1]. However, compared to a conventional RGB image of a similar spatial resolution, the volume of LF data is much larger. Thus, if a server transmits the entire LF data over bandwidth-limited networks to a user prior to user-side image synthesis and observation, this will incur a significant startup delay. This is problematic even if the LF data are first compressed using state-of-the-art LF coding tools [2, 3].

To address this problem, previous works on *interactive light field streaming* (ILFS) have proposed an alternative approach [4–8], where a user periodically requests a desired viewpoint or focal point, and in response a server transmits a pre-synthesized and encoded image for observation. The benefit is that a user can immediately observe synthesized images of the captured static 3D scene via standard image decoding, without the penalty of a long startup delay due to transmission of the entire LF data. However, transmission of only coded synthesized images means that the user never "learns" any underlying sub-aperture images used for synthesis. As a result, the user will always rely on the server for transmission of every viewpoint image, leading to a large aggregate transmission cost when the number of view requests is large.

In this paper, we propose a third alternative called *progressive light field communication* (PLFC), where for each transmitted synthesized image, a user can "decode" and recover a sub-aperture image, which is then added to a *cache* for subsequent user-side image synthesis. Specifically, the server first initializes the user's cache with a set of sub-aperture images that are deemed critical for image synthesis. Then, for each new requested view/focal point image, the server transmits a synthesized image that is a sparse linear combination of sub-aperture images in the user's cache *plus* one carefully chosen new sub-aperture image. Upon receipt of the synthesized image, the user can deduce the new sub-aperture image by solving linear equations, similarly done in network coding [9]. Once a sufficient number of sub-aperture images have been accumulated in the cache, the user can synthesize his own images, reducing overall transmission cost.

Compared to full LF data compression prior to transmission, PLFC does not suffer a long startup delay prior to user's view observation. Yet compared to previous ILFS—equivalent to another extreme case where the cache remains empty at all time—PLFC allows a user to learn and accumulate sub-aperture images in the cache, leading to transmission savings in the long run. To efficiently implement PLFC, in this paper we address the following critical issues: i) initialization of sub-aperture image cache, ii) selection of sparse sub-aperture images for requested image synthesis, and iii) "decoding" of sub-aperture image using received synthesized

Email: eduardopeixoto@ieee.org, {macchiavello, mintsu, camilodorea}@unb.br, cheung@nii.ac.jp

This work was partially supported by CNPq under grant 307737/2015-2.

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

image and cached sub-aperture images. Experimental results show that our implementation of PLFC outperforms a simple ILFS-based differential coding scheme by up to $70\%$ in Bjøntegaard-Delta Bitrate (BD-Rate) [10] considering the accumulated rate for all requested images if a sufficient number of view/focal point images are requested.

## 2. CONVERTING VIEW/FOCAL POINT IMAGES TO / FROM SUB-APERTURE IMAGES

LF data can be represented as $N$ sub-aperture images. The complete set of sub-aperture images is denoted as $\mathcal{S}^o$. Sub-aperture images can be used to synthesize images of different viewpoints or focal points. Though in this work we focus on synthesizing images of different focal points, our proposed PLFC framework can be used for both cases. A synthesized image can be obtained as a linear combination of shifted sub-aperture images [1, 11]. Suppose that the user desires a new focal point image represented by slope $\alpha$. The synthesized image, $\mathbf{v}(\alpha, \mathcal{S}^o)$, can be generated as follows:

$$\mathbf{v}(\alpha, \mathcal{S}^o) = \frac{\sum_{i \in \mathcal{S}^o} w_i^\alpha \mathbf{x}_i^\alpha}{\sum_{i \in \mathcal{S}^o} w_i^\alpha} \tag{1}$$

where $w_i$ is a weight matrix for each sub-aperture image $\mathbf{x}_i$ in $\mathcal{S}^o$ and $\mathbf{x}_i^\alpha$ is the shifted version of $\mathbf{x}_i$. The weights $w_i$ are proportional to the light intensity acquired by each subpixel within the macropixel and are intrinsic to the camera.

If only a sparse subset of sub-aperture images are used, $\mathcal{S} \subset \mathcal{S}^o$, then the estimated view $\mathbf{v}(\alpha, \mathcal{S})$ will not be equal to $\mathbf{v}(\alpha, \mathcal{S}^o)$. Thus we denote by $D_\mathbf{v}(\mathcal{S})$ the distortion of the focal point image $\mathbf{v}$ using a linear combination of sub-aperture images in $\mathcal{S}$ relative to $\mathcal{S}^o$:

$$D_\mathbf{v}(\mathcal{S}) = \|\mathbf{v}(\alpha, \mathcal{S}^o) - \mathbf{v}(\alpha, \mathcal{S})\|_2^2. \tag{2}$$

Eq. (1) can be slightly modified to:

$$\mathbf{v}(\alpha, \mathcal{S}) = \frac{\sum_{i \in \mathcal{S}} p_i \cdot w_i^\alpha \cdot \mathbf{x}_i^\alpha}{\sum_{i \in \mathcal{S}} w_i^\alpha} \tag{3}$$

where $p_i$ are scalar weights applied to all pixels in $\mathbf{x}_i^\alpha$. The main idea is to find the optimal linear combination that minimizes eq. (2).

We now assume that a server and a user know the subset $\mathcal{S}$ and all weights $w$. The server adds a new sub-aperture image, $z$, to the subset and then creates a new synthesized image $\mathbf{v}(\beta, \mathcal{S} \cup z)$ by:

$$\mathbf{v}(\beta, \mathcal{S} \cup z) = \frac{\sum_{i \in \mathcal{S}} \left( p_i \cdot w_i^\beta \cdot \mathbf{x}_i^\beta \right) + w_z^\beta \cdot z^\beta}{\sum_{i \in \mathcal{S}} \left( w_i^\beta \right) + w_z^\beta} \tag{4}$$

If the user receives $\mathbf{v}(\beta, \mathcal{S} \cup z)$, it can use it to estimate the chosen sub-aperture image $z$. Note that, if $\mathbf{v}$ is received exactly, then $z^\beta$ can also be decoded perfectly. If we perform the opposite shift $-\beta$, in order to recover $z$, we are possibly adding some distortion to the estimated image due to numerical imprecision during spatial interpolation required to create a shifted version of an image. However, we are not interested in the decoded sub-aperture image $z$ itself; we are interested in using it for future image synthesis. Hence, the user can add the image $z^\beta$ to the subset, along with the value of the shift $\beta$. When this image is used in order to generate a synthesized view at a new slope $\gamma$, then $z^\beta$ is shifted only by the difference between the shift $\beta$ and the new shift $\gamma$. This way, the additional distortion due to shifting is added only once.

## 3. INTERACTIVE TRANSMISSION FRAMEWORK

We first overview our communication system. We assume that a server has available in storage a full set of LF image data—all sub-aperture images $\mathcal{S}^o$ of a static 3D scene. A user requests periodically a different viewpoint or focal point image of the scene—according to a view interaction model—that can be synthesized as a linear combination of shifted sub-aperture images. Corresponding to each user's request, the server must transmit sufficient image data for the user to render and display each requested image.

Specifically, in our proposed framework, a user maintains a *cache* $\mathcal{C}$ of decoded sub-aperture images that can be used subsequently to synthesize new virtual images. The encoder also has knowledge of this cache by mimicking the decoder operations locally. At the beginning of the streaming session, the server transmits a startup sub-aperture image set $\mathcal{S}^i$ to initialize $\mathcal{C}$. Upon a user's request of a new slope $\beta$, the server can choose among two options:

1. Instruct the user to linearly combine sub-aperture images at the user's cache $\mathcal{C}$ to synthesize the desired focal point image by transmitting the weights $p_i$.
2. Synthesize and transmit the requested synthesized image as an *optimized* linear combination of sub-aperture images in the user's cache $\mathcal{C}$ *plus* one new sub-aperture image. The transmitted image is differentially coded using as a reference frame the image synthesized using only the sub-aperture images available in cache $\mathcal{C}$.

In this proposed framework the encoder works with a user-defined target quality. In option one a synthesized view $\mathbf{v}(\beta, \mathcal{C})$ can be obtained by eq. (3), at a very low bitrate. Nevertheless, $\mathbf{v}(\beta, \mathcal{C})$ may not achieve the target quality. In this case, option two is chosen. By including a new image, $z$, in $\mathcal{C}$ a better virtual image $\mathbf{v}(\beta, \mathcal{C} \cup z)$ can be generated. However, this still does not guarantee the desired quality. Therefore, option two is used iteratively until the target quality is met.

In the next sections we will discuss a view interaction model, cache initialization and the encoder options in details. For simplicity, we assume that the synthesized image always reflects a change in focal point; however, this model can be also used for generating new viewpoints.

### 3.1. View Interaction Model

We assume that a user starts navigation at an initial focal point $\mathbf{u}$ chosen from a 2D array $\mathcal{V}^o$ of $N$ different options. The

probability of switching from a focal point $i$ to $j$ is $p_{i,j}$, the $(i,j)$-th entry of a probability transition matrix $\mathbf{P}$. Suppose the lifetime of an interactive streaming section is $T$ focal-switches. Then the *importance* $\phi(i)$ of a focal point $i$ is defined as:

$$\phi(i) = \sum_{t=1}^{T} \left[\mathbf{1_u}\mathbf{P}^t\right]_i \tag{5}$$

where $\mathbf{1_u}$ is the canonical *row* vector of length $N$ where all entries are $0$ except $u$-th entry is $1$, and $[\,]_i$ denotes the $i$-th entry of a vector. In words, (5) is the sum of probabilities that a user selects focal point $i$ in each of the $T$ switches.

### 3.2. Initialize User Cache

The initial set $\mathcal{S}^i$ that the server transmits at the start of the streaming session must be important enough that each sub-aperture image in $\mathcal{S}^i$ contributes significantly to synthesized image quality. Note that our method will have an initial delay compared to ILFS, however if the total size $|\mathcal{S}^i|$ is sufficiently small, this delay will be significantly lower than transmitting the entire LF. We can select $\mathcal{S}^i$ as follows.

Define the *benefit* of a sub-aperture image $z$ as the *decrease* in distortion if $z$ is added to $\mathcal{S}^i$. We incrementally add sub-aperture image that is the most beneficial until a budget of $K$ images are reached:

$$\max_{z\in\mathcal{S}^o\setminus\mathcal{S}^i} \sum_{\mathbf{v}\in\mathcal{V}^o} \phi(\mathbf{v}) \left(D_{\mathbf{v}}(\mathcal{S}^i) - D_{\mathbf{v}}(\mathcal{S}^i\cup\{z\})\right) \tag{6}$$

The selected images can be transmitted to the user using any off-the-shelf image or video encoder. In our implementation, the first frame selected is encoded as an intra-coded I-frame, while the others are encoded as differentially coded P-frames. All frames are encoded using lossy compression.

### 3.3. Option 1: Instruct View Synthesis

If the server decides not to transmit any synthesized image to the user, then the user can only synthesize the desired virtual image $\mathbf{v}$ using sub-aperture images in $\mathcal{C}$, and the scalar weights $p_i$, resulting in distortion $D_s$:

$$D_s = \|\mathbf{v}(\beta,\mathcal{S}^o) - \mathbf{v}(\beta,\mathcal{C})\|_2^2. \tag{7}$$

### 3.4. Option 2: Synthesize & Transmit

If the server decides to transmit the synthesized focal point $\mathbf{v}$, at a given slope $\beta$, to the user, then it will generate $\mathbf{v}(\beta,\mathcal{C}\cup z)$ as given by eq. (4). As detailed in Section 2, the user can "decode" $z$ from the received synthesized image and add $z$ to its cache $\mathcal{C}$ for subsequent image synthesis. Which $z$ should be used for synthesis of $\mathbf{v}$ deserves careful consideration and will be discussed later.

The server can directly transmit $\mathbf{v}(\beta,\mathcal{C}\cup z)$ as an intra-coded image. We propose a more intelligent scheme leading to a smaller coding cost. The receiver synthesizes a version of the desired image using only sub-aperture images in $\mathcal{C}$, $\mathbf{v}(\beta,\mathcal{C})$. These two images are different versions of the requested image. We then use $\mathbf{v}(\beta,\mathcal{C})$ as the predictor for differential coding of $\mathbf{v}(\beta,\mathcal{C}\cup z)$.

### 3.5. Selection of the new sub-aperture image $z$

Given user already has sub-aperture images in $\mathcal{C}$, $z$ should contribute significantly to quality of synthesized focal point $\mathbf{v}$. This immediate benefit is:

$$B^1_{\mathbf{v}}(\mathcal{C},z) = D_{\mathbf{v}}(\mathcal{C}) - D_{\mathbf{v}}(\mathcal{C}\cup\{z\}) \tag{8}$$

Second, $z$ should contribute to quality of synthesized images requested in the future. Denote by $\mathcal{V}$ the set of focal point images the user has requested so far. We can compute the importance of a view $i\in\mathcal{V}^o\setminus\mathcal{V}$ as:

$$\theta(i,\mathbf{v},t) = \sum_{\tau=1}^{T-t} \left[\mathbf{1_v}\mathbf{P}^\tau\right]_i \tag{9}$$

We can now write the benefit of $z$ for future switches as:

$$B^2(z,\mathcal{C},\mathcal{V}) = \sum_{\mathbf{u}\in\mathcal{V}^o\setminus\mathcal{V}} \theta(\mathbf{u},\mathbf{v},t)\ \left(D_{\mathbf{u}}(\mathcal{C}) - D_{\mathbf{u}}(\mathcal{C}\cup\{z\})\right) \tag{10}$$

We can now combine the two above considerations into one unified criteria for choosing $\mathbf{z}$:

$$\max_{z\in\mathcal{S}^o\setminus\mathcal{C}} B^1_{\mathbf{v}}(\mathcal{C},z) + \mu B^2(z,\mathcal{C},\mathcal{V}) \tag{11}$$

where $\mu$ is a parameter that trades off current and future considerations. An algorithm to optimally select $\mu$ can be a focus of a future study; for this paper we use a small $\mu$, so $B^1_v(\mathcal{C},z)$ has a higher contribution ($\mu = 0.1$).

## 4. EXPERIMENTS

In our experiments, the user requests several images of the same light-field, changing the slope of the desired focal point. For convenience, the first slope requested is set to zero. The change of the next requested slope was generated as Gaussian process of standard deviation $0.3$ up to $100$ different slopes.

We have used some images in the JPEG Pleno image dataset [12]. All images were captured with a Lytro Illum camera, which has a large number of sub-aperture images ($225$). The images are first quantized to $8$ bits and then $2$ pixels are removed from the borders, due to acquisition noise.

We compare our results with an ILFS based method, namely Reference ILFS, where each new synthesized image is encoded as a P-slice using the previously transmitted synthesized image as a prediction. For a fair comparison, we use the same H.264/AVC encoder (JM 19.0) used in the proposed method, working in RGB mode. For the Reference ILFS, all H.264/AVC macroblock partitions are tested, and full search motion estimation is performed. The QP is set to the highest value that still achieves the desire target quality.

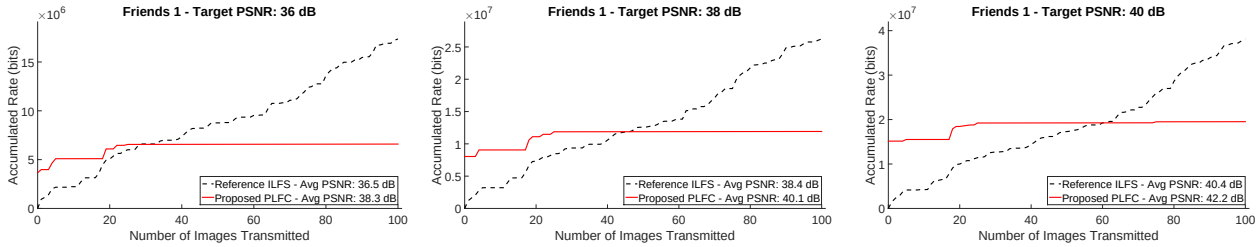


**Fig. 1**. Accumulated Rate transmitted for Friends 1 image with target PSNR : (a) 36 dB; (b) 38 dB and (c) 40 dB.

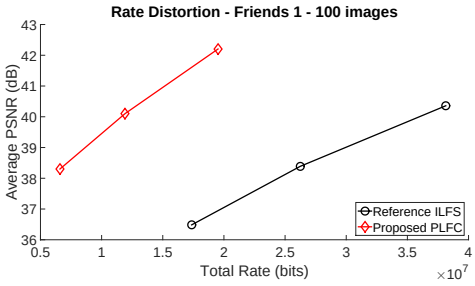


**Fig. 2**. Rate-Distortion after transmitting 100 images.

For the proposed method, namely Proposed PLFC, we also used the H.264/AVC as the encoder. However, when encoding residuals, since we already have a good estimation, we have used only macroblock types $P16 \times 16$ and $PSKIP$, and we have set all motion vectors to zero. In order to improve the quality for shifting sub-aperture images, all images are padded by 10 pixels in each direction. The cache was initialized with 3, 5 or 8 sub-aperture images, for a target PSNR of 36, 38 and 40 dB, respectively. In order to improve the synthesized images using $\mathcal{C}$, in all cases the PSNR of the images in the initial cache $\mathcal{S}^i$ are set to be 2 dB higher than the target.

In this setup, the weights $w_i$ are considered to be known by both ends. Since they are intrinsic to the camera, it is a reasonable assumption that the same camera is used for several transmissions. Nevertheless, if the camera setup changes, these weights are highly spatially correlated and can be compressed efficiently with the appropriate tool.

Fig. 1 shows a plot of the progression of the accumulated rate. As expected, the proposed PLFC algorithm performs better as more images are transmitted. It must be noted that the average PSNR of the Reference ILFS is fairly constant, as it chooses the QP in order to achieve the target quality. In the case of the proposed PLFC, however, the PSNR fluctuates more, as sometimes the images already in the cache can produce a higher quality synthesized image. The minimum PSNR of both algorithms is the same, as both are tuned to stay higher than the target quality.

Fig. 2 shows a rate-distortion plot at the end of the transmission. The rate accounts for the total number of bits used to transmit all images, whereas the PSNR shown is the average PSNR among the images.

As expected, in the extreme case, after a great number of images are transmitted, the proposed PLFC algorithm significantly outperforms the Reference IFLS. It is also interesting to see the progression of the algorithm's performance during transmission. Table 1 shows the bitrate savings, measured as the BD-Rate, of the proposed PLFC algorithm using the Reference ILFS as anchor, after a number of images are transmitted for several LFs. It can be seen that larger gains are obtained as more images are transmitted, but the proposed PLFC algorithm levels with the Reference ILFS after 30 to 50 images are transmitted, depending on the LF.

**Table 1**. BD-Rate after a number of images are transmitted.

| | Number of Images Transmitted | | | | |
|---|---|---|---|---|---|
| LF Image | 20 | 40 | 60 | 80 | 100 |
| Friends 1 | +20.6 | −9.7 | −37.9 | −61.2 | −70.4 |
| Friends 3 | +4.4 | −18.6 | −43.6 | −63.3 | −73.7 |
| Sophie and Vincent | +70.8 | +18.7 | −18.3 | −45.2 | −59.7 |
| Sophie, Krios and Vincent | +32.8 | +18.7 | −10.4 | −44.1 | −57.9 |
| Swans 1 | +30.2 | +15.6 | −16.9 | −52.4 | −66.4 |

## 5. CONCLUSION

In this work we present a novel alternative to light-field transmission, namely progressive light field communication (PLFC). In this aproach, a initial set of sub-aperture images is sent to the user, referred to as the user's cache. Then, for each transmitted synthesized image, the receiver can estimate a new sub-aperture image that can be added to the cache and be used later for image synthesis. The sender has two options, either to send a new synthesized image or to instruct the receiver how to synthesize the requested view/focal point. We also presented algorithms for selecting the sub-aperture images to initialize the cache, and to select the next sub-aperture image to be added to it. Results show that we can generate a synthetized image of good quality with a sparse subset of sub-aperture images. Moreover, when compared to an architecture that always transmits the requested viewpoint, using DPCM , the proposed scheme can reduce the accumulated rate, after a certain number of transmission, by up to 70%. As future studies several details can be improved, such as: $(i)$ an algorithm for optimal selection of the Lagrangian multiplier that balances the immediate and future benefits, $(ii)$ an adaptive cache initialization method with variable size and $(iii)$ a Rate-Distortion function for option selection during encoding.

## 6. REFERENCES


[1] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, vol. 2, pp. 1–11, 2005.

[2] Chuo-Ling Chang, Xiaoqing Zhu, P. Ramanathan, and B. Girod, "Light field compression using disparity-compensated lifting and shape adaptation," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 793–806, April 2006.

[3] G. Alves, F. Pereira, and E. A. B. da Silva, "Light field imaging coding: Performance assessment methodology and standards benchmarking," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–6.

[4] 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.

[5] 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.

[6] 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.

[7] Wei Cai, Gene Cheung, Sung-Ju Lee, and Taekyoung 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.

[8] Benedicte Motz, Gene Cheung, and Antonio 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.

[9] X. Liu, G. Cheung, and C.-N. Chuah, "Structured network coding and cooperative wireless ad-hoc peer-to-peer repair for WWAN video broadcast," in *IEEE Transactions on Multimedia*, 2009, vol. 11, no.4.

[10] G. Bjøntegaard, "Improvements of the BD-PSNR model," ITU-T SG16/Q6, 35th VCEG Meeting, Doc.VCEG-AI11, 2008.

[11] V. Vaish, B. Wiblurn, and N. Joshi amd M. Levoy, "Using plane + parallax for calibrating dense camera arrays," in *Proceedings of CVPR*, 2004.

[12] Martin Rerabek and Touradj Ebrahimi, "New Light Field Image Dataset," in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, 2016.