PRE-DEMOSAIC LIGHT FIELD IMAGE COMPRESSION USING GRAPH LIFTING TRANSFORM

Yung-Hsuan Chao[†], *Gene Cheung*[*], *and Antonio Ortega*[†]

[†] University of Southern California, Los Angeles, CA, USA
[*] National Institute of Informatics, Chiyoda-ku, Tokyo, Japan

ABSTRACT

A plenoptic light field (LF) camera places an array of microlenses in front of an image sensor, in order to separately capture different directional rays arriving at an image pixel. Using a Bayer pattern, data captured at each pixel is a single color component (R, G or B). The sensed data then undergoes demosaicking (interpolation of RGB components per pixel) and conversion to a series of subaperture images. In this paper, we propose a novel LF image coding scheme based on graph lifting transform, where the acquired sensor data are coded in their original form without pre-processing. Specifically, demosaicking is not performed, and instead we first map raw sensed color data directly to subaperture image 2D grids, then encode the color pixels, which are sparse in spatial distribution, via a graph lifting transform. Our method avoids redundancies stemming from demosaicking, and operates in the original RGB domain without color conversion and sub-sampling. The graph lifting transform efficiently encodes irregularly spaced pixels in each subaperture image, resulting in compact representations. Experiments show that at high PSNRs—important for archiving and instant storage scenarios—our method outperforms demosaicking followed by intra-only High Efficiency Video Coding (HEVC) significantly.

Index Terms— Light field imaging, image compression, graph signal processing

1. INTRODUCTION

Light Field (LF) imaging separately captures light rays arriving from different directions at each pixel in an image. With acquired LF data, multi-view rendering and re-focusing become possible post-capture. However, captured LF data are large in volume compared to a conventional color image of the same resolution, and hence efficient compression of LF data is important for storage and transmission.

In the last decade, many hardware designs have been developed for LF acquisition, including multiple camera arrays, aperture cameras, and lenselet-based plenoptic cameras. Among them, the lenselet-based plenoptic camera is the most popular, and has been made commercially available by companies such as Lytro [1] and Raytrix [2]. In a plenoptic camera, a microlens array is placed ahead of an otherwise conventional photo sensor embedded with Bayer color filter. The resulting raw image, called a *lenselet image*, then typically undergoes demosaicking (pixelwise RGB interpolation) and conversion to multiple subaperture images on a 2D array. A subaperture image can be seen as a typical 2D photo, gathering pixels from a specific light direction.

There exist two types of redundancies in the LF data: i) spatial redundancy among neighboring pixels in a subaperture image, *i.e. intra-view correlation*, and ii) angular redundancy among subaperture images of nearby directions, *i.e. inter-view correlation*. Exploiting inter-view correlation in compression leads to high computation complexity (due to motion / disparity prediction) and creates dependencies among coded subaperture images, which is undesirable for random access. In particular, in an archiving scenario, a user may desire to quickly browse through viewpoint images, each of which can be synthesized in acceptably high quality using only a small subset of subaperture images. Thus, speedy extraction of this image subset from the LF data compressed in high quality is important. Furthermore, we note that most standard digital cameras use a low complexity codec (JPEG) operating by default at very high PSNR. In analogy, in this paper we will consider here an intra-view only approach (which leads to faster encoding and better random access), operating at high rates/PSNR.

Recently, many works attempt to exploit spatial correlation in LF images via existing image/video coding tools, *e.g.,* JPEG and HEVC [3–5]. These compression approaches are applied on the full color subaperture images, which are converted from the raw lenselet image as in the aforementioned pipeline, and therefore large redundancies are introduced by demosaicking. Moreover, to incorporate standard codecs, an RGB subaperture image must be converted to 4:2:0 YUV format, which induces distortions due to integer rounding and color sub-sampling.

In this paper, we propose a new coding scheme, where *compression is applied on the original lenselet images captured by the photo sensor*, without the aforementioned pre-processing that increases data volume or distorts captured pixel values. Our work is inspired by schemes proposed in [6–9] for regular images, which also postpone the demosaicking step to the decoder. Specifically, we first map the raw captured pixels directly onto sparse locations in a series of subaperture images. Unlike the input images for compression in [6–9], where R, G, and B pixels are regularly distributed based on the Bayer pattern, the color components after the mapping to subaperture images are irregularly placed, making it difficult to be encoded using conventional schemes, *e.g.,* JPEG. In our work, the irregularly distributed pixels in a subaperture image will be connected as a graph, with the pixel values interpreted as a *graph-signal*. Suitable edge weights are assigned to reflect similarities between connected sample pairs, and the graph-signal is encoded using a graph-based lifting transform proposed in [10]. The transform has been applied previously with promising results for image compression [11–13]. Unlike previous graph-based coding works, we apply a graph lifting transform on irregularly placed pixels in individual subaperture images—the first to do so in the literature. Compared to HEVC-based coding, experiments show noticeable gain at the high PSNR range.

The outline of the paper is as follows. In Section 2, we review the conventional approach in lenselet image compression. Our pro-

The work is funded in part by Sisvel Technology S.r.l

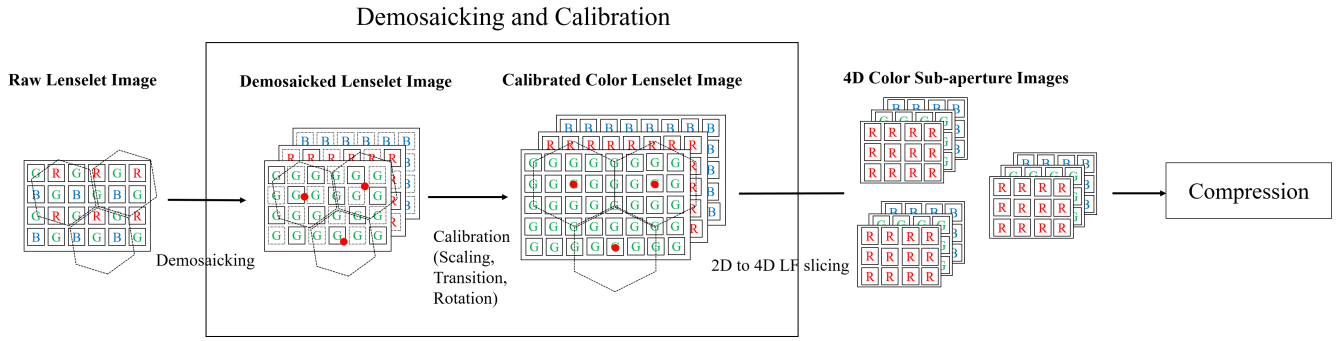


Fig. 1: Conventional coding scheme for light field image. The demosaicking and calibration processes are applied before compression.

posed coding scheme is described in Section 3. In Section 4, the graph-based transform for the LF signal is presented. Experiments and conclusions are presented in Section 5 and 6 respectively.

2. BACKGROUND: LIGHT FIELD IMAGE COMPRESSION

Fig.1 shows an overview of a conventional light field coding scheme. The pre-processing stage, which converts the originally captured lenselet image into an array of full color subaperture images, is based on the method proposed by Dansereau et al. [14, 15]. Through the Bayer filter embedded on the photo sensor, each pixel on the captured lenselet image contains only one color component out of R, G, and B. In order to generate full color images, the missing color components at each pixel have to be interpolated using the nearby pixels where the target colors are available. The process is called *demosaicking*. In this work, we apply the demosaicking approach proposed by Malvar et al. [16]. The amount of pixel values will be increased threefold through the process regardless of the demosaicking algorithms used.

Projected from the microlens array in the plenoptic camera, a lenselet image consists of multiple hexagonally arranged pixel patches, which are called *macro-pixels* (denoted in dash line in Fig.1); each macro-pixel collects lights for one point in a scene arriving from different directions. However, due to manufacturing defects, the arrangement of macro-pixels is usually not aligned with the image coordinates, making it difficult to infer pixel's corresponding position in the scene and the arriving light angle. Therefore, the color lenselet image needs to be calibrated via rotation, translation and scaling, so that each macro-pixel center (denoted with red point in the figure) falls onto an integer pixel location and the arrangement of macro-pixels is aligned to regular grid. Through the calibration, the amount of data will also be increased due to the interpolation involved in scaling.

Each pixel on the calibrated image is indexed by its spatial and angular coordinates. The spatial coordinate is given by the position of the associated macro-pixel and the angular coordinate is the relative location within each macro-pixel. We then collect pixels of the same angular coordinate into one subaperture image, where the pixels are arranged according to their spatial coordinates. Each subaperture image can be viewed as a typical 2D picture, where large correlation exists between neighboring pixels.

3. PROPOSED LIGHT FIELD IMAGE CODING SCHEME

In the pre-procesing stage of the conventional coding scheme, the volume of LF data is increased greatly during demosaicking and the scaling operation of calibration. In order to avoid these redundancies, we propose a new coding scheme for LF in which compression is performed on the original data collected in the original lenselet image instead of the pre-processed pixels in the full color subaperture images. The flow chart is shown in Fig. 2. Without demosaicking, we map raw pixels onto the calibrated lenselet image according to the transformation matrix applied in [14]. Pixels which fall onto non-integer locations after transformation will be rounded to the nearest integer positions. Then, based on the relative locations within the macro-pixels on the calibrated image, pixels are arranged onto multiple subaperture images, where redundancies between spatial neighbors can be exploited. Note that the mapping does not change the amount of pixels nor the intensity values of R, G, and B components.

Since no interpolation is applied, some pixel locations are empty in the subaperture images, as shown in Fig 3. Depending on the camera manufacturing, *i.e.,* different types of macro-pixel misalignment, and the calibration algorithm adopted, the spatial and angular coordinates for each pixel on the captured lenselet image may change accordingly. Therefore, the pattern of pixel distribution in subaperture images is not fixed and also highly irregular. The property is different from the input signal considered in the pre-demosaic image coding schemes discussed in [6–9], where R, G, and B pixels are distributed regularly based on Bayer pattern. Due to such irregularity of spatial distribution for LF data, existing coding techniques, *e.g.,* discrete cosine transform (DCT) and discrete wavelet transform (DWT), are difficult to be applied. This motivates the use of graphs, which can represent both regular and irregular data points as long as the pair-wise relations can be defined properly. In the next section, we will describe the construction of appropriate graphs for graph-based coding of sparsely distributed pixels on subaperture images.

At the decoder side, pixels on subaperture images are decompressed and inverse-mapped back to the original positions on the 2D lenselet image. The image will then be demosaicked and calibrated [14, 15] in order to generate full color 4D LF for further processing, *e.g.,* multi-view rendering and re-focusing. Note that our scheme does not rely on a particular selection of demosaicking and calibration algorithms. Other algorithms, *e.g.,* [17] and [18], can also be applied.

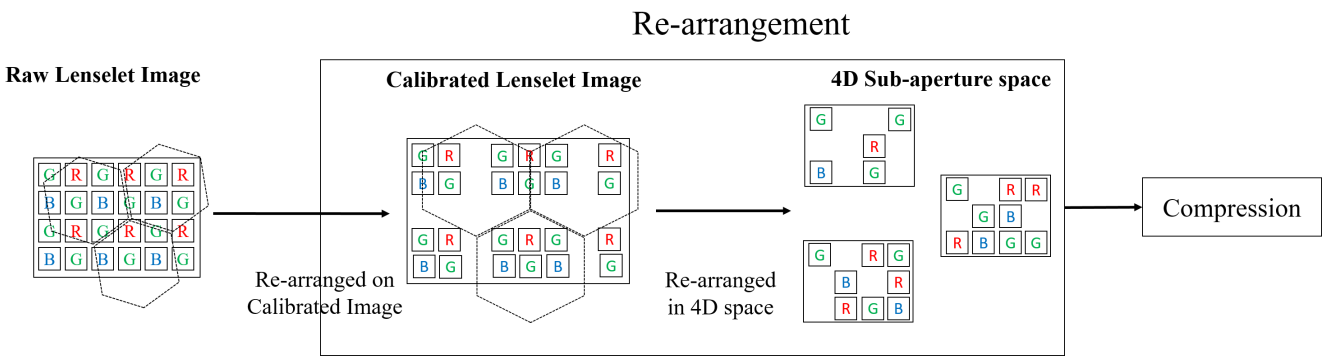


Fig. 2: Proposed LF coding scheme. The demosaicing and calibration, yielding high signal redundancy, are applied after compression.

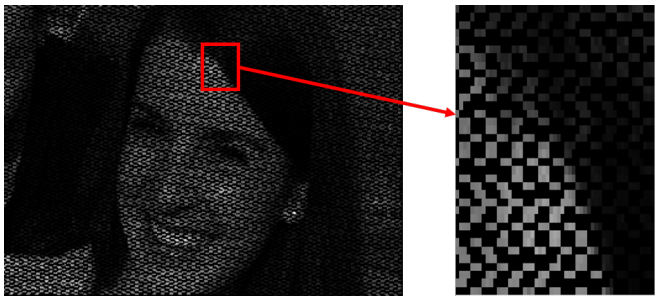
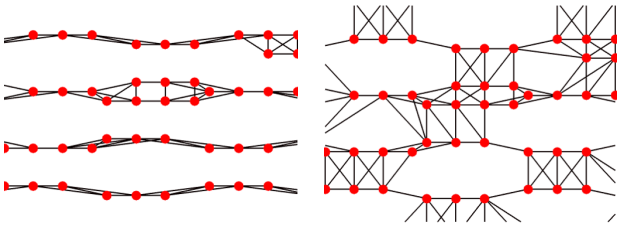


Fig. 3: Sparsely distributed G components on one subaperture image (Figure *Friends1* from EPFL light field dataset)



(a) (b)

Fig. 4: A part of graph constructed for irregularly placed R components. In (a), the one using 4 nearest neighbor method is shown. In (b), each pixel is connected to 2 neighbors in horizontal and vertical orientations respectively

4. GRAPH BASED TRANSFORM FOR LF DATA

In this section, we describe the details constructing graphs for irregularly placed pixels in a subaperture image, and the coding techniques applied. To reduce the implementation complexity and allow parallel processing, each subaperture image is divided into non-overlapped blocks, on which graph-based transform will be applied independently. A weighted graph $G = (V, E)$ consists of a set of nodes $v \in V$, and edges $e_{i,j} \in E$, which reflect the similarities between connected node pairs i and j. The similarity is measured using a non-negative weight value $w_{i,j} \in [0, 1]$. A graph-signal is usually denoted as a vector $f \in \mathbb{R}^N$, where N is the number of nodes in V. In our work, three separate graphs are constructed in each block for R, G, and B components, where each pixel is represented by one node, and the graph-signal contains the associated intensities.

In each subaperture image, similar to natural images, large local redundancies exist among pixels that are close in distance. Hence, the most straightforward approach in exploiting the pair-wise correlation is to connect each pixel with its k nearest neighbors in terms of Euclidean distance. For complexity reduction in the graph-based lifting transform [10], where the computation for each node depends on its connected neighbors, we consider mainly sparse graphs, *i.e.,* small k. However, the graph connection based on k-nearest neighbor with small k can be highly sensitive to the pixel arrangement. For example, in the cropped subaperture image shown in Fig. 4, R components are mostly aligned horizontally. The resulting graph, based on k-nearest neighbor ($k = 4$), thus consists of mostly horizontal links as shown in Fig. 4(a), and is unable to capture local similarity in regions with vertical features, *e.g.,* vertical edges.

In order to exploit similarity in different orientations, yet still keep connection sparse, we instead connect each pixel to equal number of neighbors in horizontal and vertical regions, as shown in Fig. 4(b). Define the Euclidean distance between nodes v_i and v_j as $\text{dist}(i, j)$, the weight $w_{i,j}$ on link $e_{i,j}$ is calculated as

$$w_{i,j} = \exp\left(-\frac{\text{dist}(i,j)^2}{\sigma^2}\right). \quad (1)$$

with the assumption that pixels that are closer in distance are more likely to be similar in pixel intensities.

Once the graphs are constructed, we apply the graph-based lifting transform in each block and on R, G, and B components independently. The bipartition and filter design in the lifting transform are designed based on the method proposed by Martínez-Enríquez et al. [19]. The work has provided high efficiency in coding image/video sequences, and can be applicable to signals with irregular structures. The wavelet coefficients are uniformly quantized and reordered based on the approach in [12]. For entropy coding, we apply the Amplitude and Group Partitioning (AGP) method proposed by Said and Pearlman in [20].

5. EXPERIMENTS

5.1. Experimental Setup

For archival purpose, one should assess the quality of reconstructed lenselet image in the original RGB pattern. However, current coding schemes in literature using HEVC discard under-exposed pixels at the boundary of macro-pixels, so it's difficult to convert the reconstructed subaperture images back to a lenselet image. Hence

for evaluation, we consider the reconstructed full color subaperture images. The full color subaperture image before compression, generated using the demosaicking and calibration described in [14, 15], is used as the ground truth. As a baseline, we consider the HEVC (HM 16.9) encoding of subaperture images in original 4:4:4 RGB and 4:2:0 YUV formats. The configuration used in HEVC is *All-Intra* without deblocking filter and SAO. In our proposed scheme, the same demosaicking and calibration will be applied on the decoded lenselet image in order to generate the reconstructed subaperture images for evaluation.

Each subaperture image is divided into non-overlapped 32×32 blocks. For graph connection, we connect each pixel to 2 neighbors in horizontal and vertical regions respectively. Images from the proposed and baseline schemes are compared in RGB format without sub-sampling. For the 4:2:0 format YUV, the subaperture images are translated back to 4:4:4 RGB format before evaluation. The up-sampling for U and V components is based on nearest neighbor interpolation.

The test images we consider in the experiments are acquired from the EPFL light field database [21], where the raw data are captured with Lytro Illum camera [22]. Each test image is of size 5368×7728. In the baseline scheme, the raw data will be converted into 15×15 full color subaperture images. Each subaperture image is of size 434×625. Therefore for each test image, there are totally $91546875 = 15 \times 15 \times 434 \times 625 \times (1 + \frac{1}{4} + \frac{1}{4})$ pixels needed to be encoded by HEVC with 4:2:0 YUV format. In our scheme, on the other hand, the compression is applied on the original raw data without demosaicking, and therefore only $41483904 = 5368 \times 7728$ pixels are required, saving more than 55% in data size.

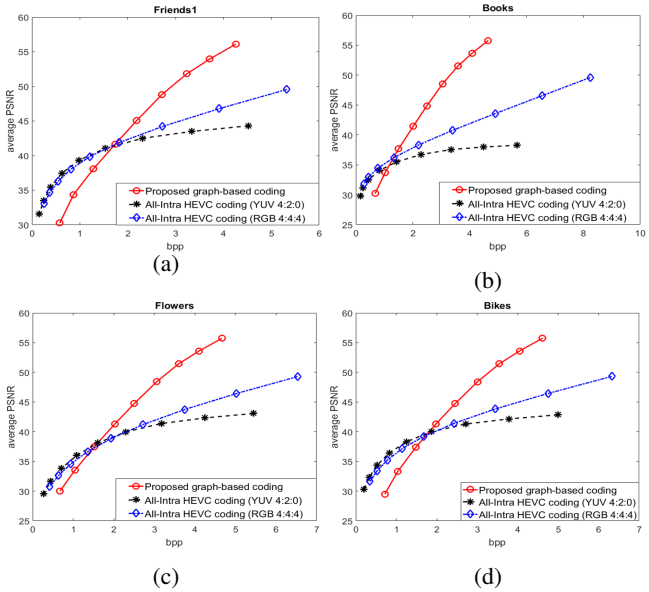


Fig. 5: Average PSNR over R, G, and B components for test images (a) Friends1 (b) Books, (c) Flowers, and (d) Bikes

5.2. Experimental Results

In Fig. 5, we show the PSNR comparison for images *Friends1*, *Books*, *Flowers*, and *Bikes*. The considered QP values range from 4 to 36. For applications like archive and instant storage on cameras, images are stored in very high quality. Therefore, in the evaluation, we consider mainly the high bitrate region. It can be seen that for higher bit rate (bpp > 2), the proposed coding scheme significantly outperforms conventional approach using HEVC. This is because as the bit rate increases, indicating a smaller quantization step, more high frequency components will be kept in the transform domain after quantization. Using conventional approach to encode will incur a large cost to scan all the coefficients, while in the proposed method, only around half the number of coefficients are encoded. For baseline method using 4:2:0 YUV format, PSNR will mostly saturated near $45dB$, which is mainly caused by the color conversion. During the conversion, some details are lost when rounding floating point values, and resolution is reduced as down-sampling are performed. In Table 1, the average PSNR gains over the baseline with HEVC 4:2:0 format are shown for 10 EPFL test images considering only bit rate > 1.5 bpp, where large improvement is achieved using the proposed graph-based coding.

Methods	All-Intra HEVC (4:4:4 RGB)	Proposed graph-based coding
Bikes	0.74	4.90
Black_Fence	2.74	6.69
Books	2.34	9.61
Color_Chart_1	3.86	5.95
Desktop	3.15	5.93
Flowers	0.34	7.09
Friends_1	1.05	5.51
Danger_de_Mort	1.01	4.83
Stone_Pillars_Outside	0.70	6.09
Magnets_1	1.48	3.37

Table 1: The average PSNR gain over HEVC 4:2:0 format for high bit rate (bpp > 1.5) based on Bjontegaard Delta Criterion

6. CONCLUSION

In this paper, we propose a coding scheme for light field image compression based on graph-based lifting transform. The scheme is able to encode the original raw data without introducing redundancies from demosaicking and calibration. Moreover, by dealing with data in the original RGB domain, distortion from color conversion and sub-sampling can be avoided. The coding results at the high bitrate region using the proposed method outperforms the widely applied HEVC based approach. For future work, we will consider more signal characteristics, *e.g.,* edges, in graph connection.

7. REFERENCES

[1] "Lytro illum," https://illum.lytro.com/.

[2] "Raytrix camera," https://www.raytrix.de/.

[3] Feng Dai, Jun Zhang, Yike Ma, and Yongdong Zhang, "Lenselet image compression scheme based on subaperture images streaming," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4733–4737.

[4] Alexandre Vieira, Helder Duarte, Cristian Perra, Luis Tavora, and Pedro Assuncao, "Data formats for high efficiency coding of lytro-illum light fields," in *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*. IEEE, 2015, pp. 494–497.

[5] Dong Liu, Lizhi Wang, Li Li, Zhiwei Xiong, Feng Wu, and Wenjun Zeng, "Pseudo-sequence-based light field image compression," in *Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–4.

[6] Sang-Yong Lee and Antonio Ortega, "A novel approach of image compression in digital cameras with a Bayer color filter

array," in *Image Processing, 2001. Proceedings. 2001 International Conference on*. IEEE, 2001, vol. 3, pp. 482–485.

[7] Chin Chye Koh, Jayanta Mukherjee, and Sanjit K Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1448–1456, 2003.

[8] King-Hong Chung and Yuk-Hee Chan, "A lossless compression scheme for Bayer color filter array images," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 134–144, 2008.

[9] Sang-Yong Lee and Antonio Ortega, "A novel approach for compression of images captured using Bayer color filter arrays," *arXiv preprint arXiv:0903.2272*, 2009.

[10] Sunil K Narang and Antonio Ortega, "Lifting based wavelet transforms on graphs," in *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*. Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, 2009, pp. 441–444.

[11] Eduardo Martínez-Enríquez and Antonio Ortega, "Lifting transforms on graphs for video coding," in *2011 Data Compression Conference*. IEEE, 2011, pp. 73–82.

[12] Eduardo Martínez-Enríquez, Fernando Díaz-de María, and Antonio Ortega, "Video encoder based on lifting transforms on graphs," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 3509–3512.

[13] Yung-Hsuan Chao, Antonio Ortega, and Sehoon Yea, "Graph-based lifting transform for intra-predicted video coding," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 1140–1144.

[14] Donald G Dansereau, Oscar Pizarro, and Stefan B Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1027–1034.

[15] Donald G Dansereau, "Light field toolbox," https://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4.

[16] Henrique S Malvar, Li-wei He, and Ross Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 3, pp. iii–485.

[17] Donghyeon Cho, Minhaeng Lee, Sunyeong Kim, and Yu-Wing Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3280–3287.

[18] Shan Xu, Zhi-Liang Zhou, and Nicholas Devaney, "Multi-view image restoration from plenoptic raw images," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 3–15.

[19] Eduardo Martínez-Enríquez, Jesus Cid-Sueiro, Fernando Diaz-De-Maria, and Antonio Ortega, "Directional transforms for video coding based on lifting on graphs," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[20] Amir Said and William A Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," in *Electronic Imaging'97*. International Society for Optics and Photonics, 1997, pp. 25–37.

[21] "Light-field image dataset," http://mmspg.epfl.ch/EPFL-light-field-image-dataset.

[22] Martin Řeřábek and Touradj Ebrahimi, "New light field image dataset," in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, 2016, number EPFL-CONF-218363.