HIERARCHICAL FOURIER DISPARITY LAYER TRANSMISSION FOR LIGHT FIELD STREAMING

Mikael Le Pendu, Cagri Ozcinar, Aljosa Smolic

V-SENSE, School of Computer Science and Statistics, Trinity College Dublin, Ireland

ABSTRACT

In this paper, we present a novel approach to efficiently transmit light fields in the Fourier Disparity Layer (FDL) representation using a binary hierarchical scheme. The FDL model consists of a set of additive layers which can be simply shifted and summed to render a view of the light field at any angular coordinate. In order to transmit the FDL model, we propose a method for building a binary tree where the root consists of a single compound layer obtained as the sum of all the original layers. Subsequent levels of the tree are obtained by splitting a parent layer into two children layers whose sum is equal to the parent layer. Hence, the FDL model is recursively refined with additional layers at each new level, resulting in a scalable representation. An efficient scheme is proposed to encode a single image in order to split a parent layer into its two children. Thanks to this approach, the total number of images to decode for receiving the complete tree is equal to the number of layers in the original FDL model, which is typically smaller than the number of views required in the traditional light field representation.

Index Terms— Light Fields, Streaming, Compression, Fourier Disparity Layers

1. INTRODUCTION

The emergence of light field capturing and rendering technologies allow advanced user interactions with natural image content, paving the way for new virtual reality applications. While traditional images only contain a projection of the scene in two spatial dimensions, a light field additionally includes two angular dimensions representing all the possible viewpoints along a 2D plane. This 4D image representation enables photo-realistic rendering of a scene at varying viewpoints with additional control over aesthetic parameters such as camera aperture and focus. However, for streaming applications, the storage and transmission of such content is very challenging due to the large amount of information in 4D light fields that contain many views of the same scene.

Light field compression is a very active research topic as evidenced by the standardization initiative in JPEG-Pleno [1] as well as the numerous and diverse methods that have been proposed recently. These methods are generally classified into two main categories. The first category specifically targets the light fields captured with lenslet based plenoptic cameras and encodes the lenslet image directly formed on the sensor. Most of these methods extend the standard HEVC intra predictions with additional prediction modes that exploit the structure and redundancies within the lenslet image [2–8]. The second category of methods encode the light field in the more generic 4D representation. In practice, this means that the data to transmit is a 2D array of 2D images, where each image corresponds to a viewpoint of the scene. Motion compensation prediction schemes of video coding standards have been directly used in [9–11] where the views are encoded as a pseudo video sequence, or in [12] that adapts the multiview extension of HEVC to a 2D array of views. Other approaches leverage view synthesis techniques to predict entire views using, for example, disparity compensation [13–15], or Fourier domain priors [16, 17]. Transform coding schemes adapted to light fields have also been designed in [18] and [19]. While [18] uses the 4D DCT transform, the authors of [19] propose a motion-compensated wavelet decomposition.

However, the number of views to be transmitted may be very large, particularly for dense light fields which are suitable to simulate camera refocusing. This reduces the applicability of these approaches in the context of streaming where both compression rates and decoding complexity are essential. Alternatively, more compact representations have been explored such as a low rank approximation of the 4D light field in [20, 21], a steered mixture of experts model in [22], or local graph transforms in [23]. However, the respective coding schemes still require the full reconstruction of the 4D light field in order to perform renderings such as refocusing.

For more efficiency, we propose in this paper to transmit light fields in the Fourier Disparity Layer (FDL) representation [24] which can model arbitrarily dense light fields with only a few layers. Unlike other compact representations, the FDL model is directly suitable for real-time rendering of either all-in-focus views or refocused images with advanced control over the synthetic camera aperture. Each layer is associated to a single disparity value and mostly contains the texture information at the corresponding depth. However, remaining correlations between layers are not straightforward to exploit with traditional video compression tools. Therefore, instead of encoding the layers as a sequence, we propose a hierarchical scheme based on an FDL binary tree. The sum of all the layers, which corresponds to the rendered central view, is transmitted first as the root compound layer. Then, we encode another image containing the information required to split the root layer in two layers of lower energy. By recursively splitting the compound layers, we finally obtain the original FDL model. At each level of recursion, the obtained layers can be used for light field rendering, hence allowing a scalable representation where the current result is progressively refined by decoding further levels of the tree. The correlations between a given compound layer and the splitting image are similar to a local tone mapping relationship. This directs our choice of encoder towards the method in [25] initially proposed to compress high dynamic range images from their tone mapped, low dynamic range versions.

By evaluating the quality of both the rendered views and the refocused images, we obtain large rate-distortion gains in comparisons to traditional view encoding schemes, including the JPEG-Pleno reference encoders.

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776.

2. FOURIER DISPARITY LAYER BINARY TREE

2.1. Fourier Disparity Layer Model

First, let us present the Fourier Disparity Layer model. In the FDL model, each layer L_k of index k is associated to a disparity value d_k such that any view $I_{u,v}$ of angular coordinates (u, v) can be rendered simply by shifting each layer L_k by a vector $(d_k u, d_k v)$ and by summing all the shifted layers. This is expressed in [24] by the following equation in the Fourier domain:

$$\hat{I}_{u,v} = \sum_{k=1}^{n} e^{2i\pi d_k (u\omega_x + v\omega_y)} \hat{L}_k(\omega_x, \omega_y), \tag{1}$$

where $\hat{I}_{u,v}$ and $\hat{L}_k$ are the Fourier transforms of the view $I_{u,v}$ and the layer L_k respectively, n is the number of layers, and ω_x, ω_y are the spatial horizontal and vertical frequencies respectively. Thanks to this linear relationship, the inverse problem that consists in constructing the FDL model from a set of input views is solved in the Fourier domain for each spatial frequency as a linear least squares problem. Based on the same equation, a calibration of the disparities d_k and the angular coordinates (u, v) of each input view is also presented in [24].

2.2. Binary Tree Construction

For efficient and scalable transmission of the model, we derive an FDL binary tree representation in which each new level of the tree increases the number of layers, and thus the accuracy of the FDL model.

Let us note $L_{l\to r} = \sum_{k=l}^{r} L_k$, the compound layer from a left index l to a right index r. The root of the tree is the sum of all the layers, noted $L_{1\to n}$ (see Fig. 1(a)). Note that from Eq. (1), this corresponds to the central view of coordinates $(u, v) = (0, 0)$. Starting from the root node, the tree is built by recursively splitting parent nodes into two children nodes. Given the compound layer $L_{l\to r}$ of a parent node (with $l < r$), and a split index $s \in [\![l, r-1]\!]$, the two children are defined as $L_{l\to s}$ and $L_{s+1\to r}$ (see Fig. 1(b,c)). In our implementation, the split index is set as $s = \lfloor (l + r)/2 \rfloor$ so that the scene information in the disparity range $[d_l, d_r]$ is distributed evenly between the two children. This node splitting operation can be performed until we reach an original layer $L_k (k \in [\![1, n]\!])$. Therefore, the original FDL model is contained in the leaf nodes.

At every node of the tree, the corresponding layer $L_{l\to r}$ is associated with a disparity value $d_{l\to r} = \sum_{k=l}^{r} d_k$. Therefore, even when only the low levels of the tree are known, an approximation of the light field may be reconstructed from the compound layers and their associated disparity values using Eq.(1).

2.3. Split Images

In this tree representation, the number of nodes, and thus the total number of images is equal to $2n - 1$. However, by definition, every parent layer $L_{l\to r}$, is equal to the sum of its two children layers $L_{l\to s}$ and $L_{s+1\to r}$. Therefore, the two children layers do not need to be directly encoded. Instead, we define a single split image $L^s_{l\to r}$ that only contains the information of the difference between the two layers:

$$L^s_{l\to r} = \frac{L_{l\to s} - \alpha \cdot L_{s+1\to r}}{\alpha + 1}, \tag{2}$$

where α is a positive scalar value. In practice, there exist redundancies between the two children layers. Hence, computing their difference in Eq. (2) reduces the redundancies and the split image can be compressed more efficiently. The scaling parameter α compensates for the fact that both children layers may have significantly different energy when the number of original layers summed to form $L_{l\to s}$ and $L_{s+1\to r}$ are different. In our implementation we fix α as the ratio between the numbers of original layers summed in $L_{l\to s}$ and in $L_{s+1\to r}$. It is expressed as $\alpha = \frac{s-l+1}{r-s}$. Therefore, α does not need to be transmitted.

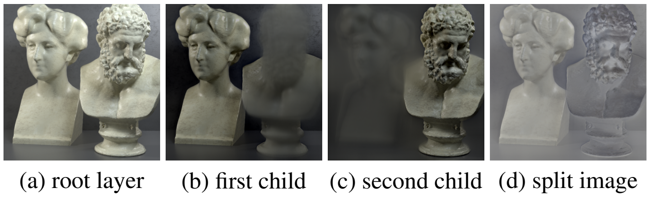
(a) root layer (b) first child (c) second child (d) split image

Fig. 1. Compound layers and split image: (a) root layer $L_{1\to 20}$, (b) first child $L_{1\to 10}$, (c) second child $L_{11\to 20}$ (d) split image $L^{10}_{1\to 20}$ formed from the two children layers. The split image is shown with an offset added to its values to display the negative values.

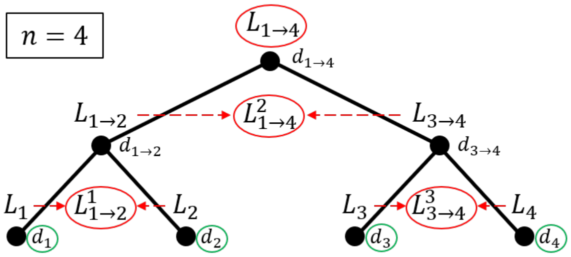

Fig. 2. Example of FDL tree with n=4. Only the n images circled in red and the n disparity values circled in green need to be transmitted to reconstruct the complete tree.

Finally, given Eq. (2) and knowing that $L_{l\to r} = L_{l\to s} + L_{s+1\to r}$, the two children layers are reconstructed as:

$$L_{l\to s} = \frac{\alpha}{\alpha + 1} L_{l\to r} + L^s_{l\to r} \tag{3}$$

$$L_{s+1\to r} = \frac{1}{\alpha + 1} L_{l\to r} - L^s_{l\to r} \tag{4}$$

Fig. 2 illustrates an example of FDL tree with the notations introduced in this section.

3. COMPRESSION OF THE TREE

In order to transmit the complete tree presented in Section 2, we must compress the root layer as well as the $n - 1$ split images required to reconstruct further levels of the tree. After conversion from RGB colorspace to YUV 4:4:4 format with 10 bit encoding, the root layer $L_{1\to n}$ is compressed using the HEVC standard in intra mode. The split images are also encoded in 10 bit YUV 4:4:4 format. However, since these images typically contain positive and negative values, we normalize the image data to the range $[0, 1]$ before the 10 bit encoding. The n disparity values are also transmitted as 32-bit floating point values. We present the normalization process in Section 3.1 and the encoding scheme used for the split images in Section 3.2.

3.1. Image Normalization and Quantization Adjustment

For an image I, its normalized version is $\tilde{I} = (I - m)/M$, where the normalization parameters m and M must be transmitted so that the decoder can perform the inverse operation. The offset m is the minimum value of I and is encoded on 11 bits (10 bit precision

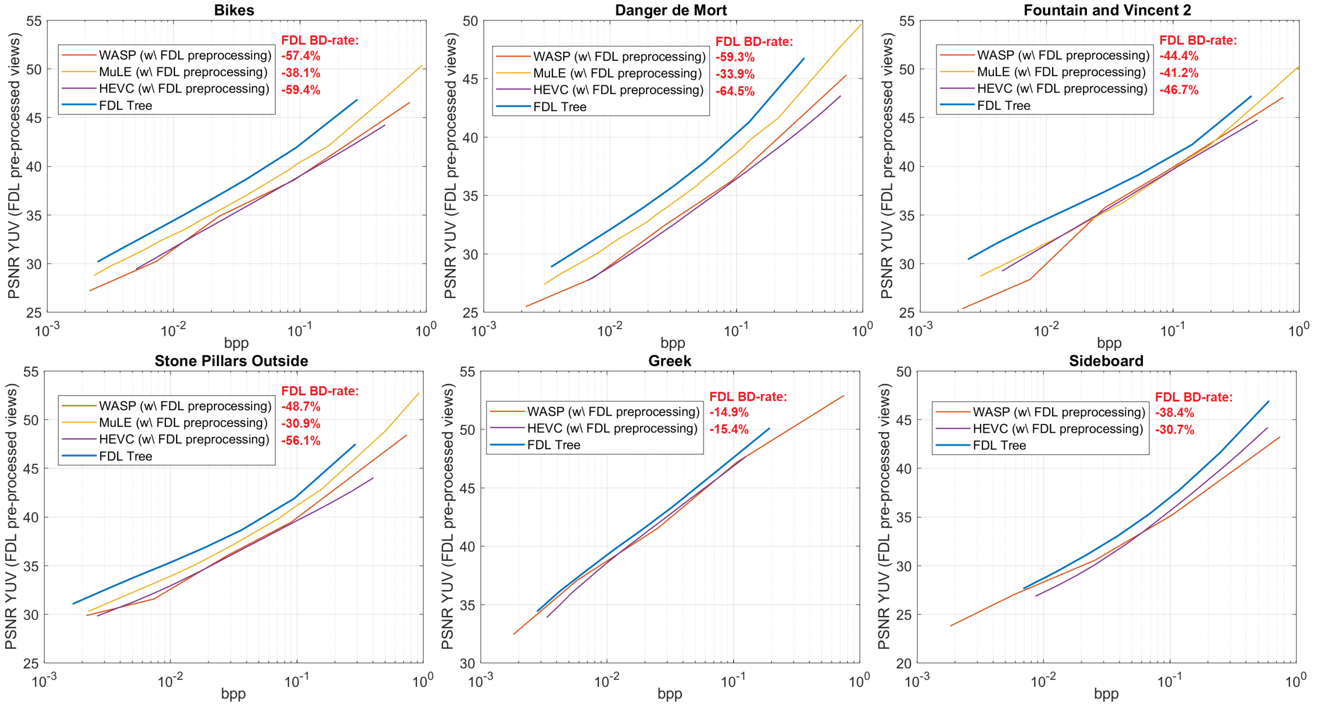

Fig. 3. Rate distortion results with PSNR YUV evaluated on the views obtained with FDL pre-processing. The Bjontegaard rate gains of our FDL Tree method with respect to WASP, MuLE and HEVC are indicated in red next to the corresponding legend items.

and 1 sign bit). Ideally, the parameter M should be computed as $M = \max(I) - m$ so that $\tilde{I}$ covers the full range $[0, 1]$.

However, while this rescaling increases the internal precision within the HEVC encoder, it also affects the quantization and thus the bitrate allocation between the different images to compress. Therefore, we adjust the encoder quantization parameter (QP) to compensate for the scaling of $1/M$. The HEVC QP parameter is an integer defined such that an increase of 6 multiplies the internal quantization step by 2. Therefore, we define a QP offset as:

$$QP_{off} = \left\lfloor 6 \cdot \log_2 \left(\frac{1}{\max(I) - m} \right) \right\rceil \tag{5}$$

where $\log_2$ is the base 2 logarithm. Now, the normalization can be performed using $M = e^{-QP_{off} \cdot \ln(2)/6} \approx \max(I) - m$. Adding the value QP_{off} to the HEVC QP parameter perfectly compensates for the normalisation factor $1/M$.

3.2. Encoder for the Split Images

For an efficient compression of the split images, we observe that redundancies exist between a given parent layer $L_{l\to r}$ and its corresponding split image $L^s_{l\to r}$ (see Fig. 1(a) and (d)). While the texture content of $L_{l\to r}$ also appears in $L^s_{l\to r}$, the dynamic of the images are very different, and the relationship between the two images is comparable to a local tone mapping. However, there is no motion between the images, which makes it possible to use an encoder designed for encoding an image from a given tone mapped version. Various high dynamic range image compression methods have been proposed in the literature to exploit such correlations. In our case, the tone mapping relationship is local and highly non-linear, which excludes methods such as [26–28] that model the tone mapping as a global tone curve. In [29] and [25], on the other hand, no such assumption is made since they perform inverse tone mapping on a per-block basis. In particular, we choose the HEVC based encoder in [25] that has shown higher performances for the most challenging tone mapping operators. It implements a new prediction mode within the HEVC Test Model encoder (HM) that first predicts a non-linear tone curve of a current block based on its neighborhood containing previously reconstructed data for both the original and the tone mapped versions.

4. EXPERIMENTAL RESULTS

For our experiments, we construct FDL models with $n = 20$ layers from 6 light fields of the JPEG-Pleno dataset [30]. We use the 4 Light Fields ‘Bikes’, ‘Danger de Mort’, ‘Fountain Vincent 2’, and ‘Stone Pillars Outside’ captured with a Lytro Illum camera and the 2 synthetic light fields ‘Greek’ and ‘Sideboard’. In order to determine accurate angular coordinates of the input light field views and an optimal set of disparity values d_k for constructing the layers, we use the FDL calibration in [24].

We compare in Sections 4.1 and 4.2 the compression performance of our method with the following encoding schemes:

- JPEG-Pleno VM 2.1 using the WASP mode (encoder based on disparity compensation).
- JPEG-Pleno VM 2.1 using the MuLE mode (encoder based on 4D-DCT transform).
- HEVC encoding of a pseudo-sequence of views following the scanning order in [31]. We use the HEVC Test model encoder (HM) with 10 bit YUV 4:4:4 encoding and using random access configuration).

Finally, we show in Section 4.3 that our coding scheme can be further optimised, particularly for low bitrate transmission, in the context of adaptive streaming.

4.1. Evaluation of Rendered Views

When constructing the FDL model from an array of light field views, the corresponding views rendered with Eq. (1) may differ from the

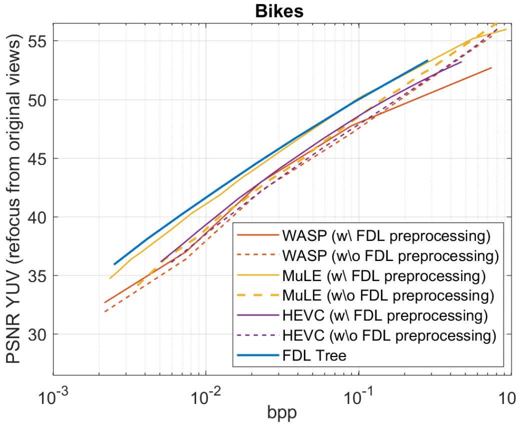

Fig. 4. Rate distortion results on the light field 'Bikes' with PSNR YUV evaluated on the refocused images from the original light field (without FDL pre-processing).

original ones. In particular, for noisy light fields such as the ones captured by the Lytro camera, this FDL processing naturally filters the noise and corrects possible brightness and color inconsistencies between views. Hence, for fair comparisons with our FDL coding scheme, the other methods are applied to the FDL pre-processed array of views (i.e. rendered with the uncompressed FDL model). These pre-processed views are used as the reference for quality evaluation. Comparative rate distortion results are provided in Fig. 3, where the PSNR YUV metric and the bitrate (in bits per pixel) are defined according to the JPEG-Pleno common test conditions [31].

For all the tested light fields large improvements in compression performance are observed in comparison with the other approaches. The Bjontegaard bitrate savings with respect to WASP, MuLE and HEVC are shown in red in Fig. 3, and the Bjontegaard rate and PSNR gains averaged over the tested light fields are reported in Table 1.

	WASP	MuLE	HEVC
BD PSNR	2.27 dB	1.58 dB	2.27 dB
BD Rate	-43.8%	-36.0%	-45.5%

Table 1. Average Bjontegaard rate and PSNR gains of our method with respect to the compared methods.

4.2. Evaluation of Refocused Images

Here, we evaluate the results on the refocused images rendered from the light field, and we assess the impact of the FDL pre-processing of the views on the other compression methods. The ground truth for the PSNR YUV evaluation is a set of 20 refocused images each corresponding to one of the disparity values obtained in the FDL calibration (i.e. the 20 values cover all the disparity range in the light field). Note that, the ground truth images are computed from the original light field without pre-processing. For all the methods, the refocusing is performed using the traditional shift-and-sum algorithm from the reconstructed views. Note that for our method, faster computations could be obtained by directly applying the FDL real-time rendering in [24] without reconstructing the views first. However, this may result in slightly different images and introduce a bias in the quality evaluation because it simulates a continuous aperture rather than a discrete shift-and-sum of sub-aperture images.

The results in Fig. 4 show that our FDL Tree method outperforms the WASP, MuLE and HEVC encoding of the views, either with or without the pre-processing. Furthermore, the latter methods are significantly improved by the FDL pre-processing. This is explained by the fact that the reduced noise and color inconsistencies in the views reduce the bitrate without degrading the refocused images since the refocusing has similar filtering properties.

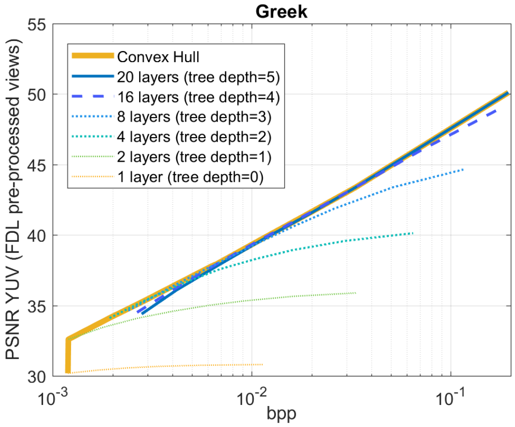

Fig. 5. Results of our method where the encoded FDL is optimised for different numbers of layers (i.e. variable depth of the encoded tree). The ground truth is the light field pre-processed with an FDL of 20 layers. The convex hull shows the overall performance.

4.3. Adaptive Streaming with Variable Tree Depth

Our method is naturally scalable since only a portion of the encoded tree may be decoded to obtain a coarse FDL model with ν layers ($\nu < n$). As described in Section 2.2, this coarse model is obtained by summing layers of the original FDL model initially computed with n layers. However, given a number of layers ν, this procedure may not provide the optimal FDL model for light field reconstruction. In order to compute such an optimal model, the FDL construction in [24] must be applied directly with ν layers.

This can be exploited in the context of adaptive streaming, where several encodings of the content are performed at various bitrates in order to transmit the version that best suits the network bandwidth. In this scenario, several FDL trees with different depths can be encoded, and the lower depth trees can be optimally computed for the corresponding number of layers. We show in Fig. 5 that for low bitrates, the optimised low depth FDL trees allow a higher quality view reconstruction. This improves the overall compression performance for adaptive streaming characterized by the convex hull in Fig. 5.

5. CONCLUSION

We have presented a novel approach to efficiently transmit a light field represented as a Fourier Disparity Layer model rather than a conventional array of views. In the FDL model, the texture information of the 3D scene is distributed over several layers, each corresponding to a depth plane. In order to transmit the layers in a scalable fashion and to exploit their redundancies, we have proposed a binary hierarchical scheme that first encodes a compound layer equal to the sum of all the layers. The original FDL model is recovered by recursively splitting compound layers into two children layers. The information required for this operation is contained in a split image that is transmitted using a state-of-the-art high dynamic range image encoder exploiting the tone mapping like relationship between the split image and the parent compound layer. The proposed scheme was shown to outperform both modes of the latest JPEG-Pleno Verification Model (VM 2.1) as well as the HEVC compression of a pseudo-sequence of views when encoding the FDL pre-processed light field. This pre-processing also improves the compared method when evaluating the refocused images. Finally, further optimisation of our scheme was also proposed for an adaptive streaming scenario.

6. REFERENCES

[1] P. Schelkens, Z. Y. Alpaslan, T. Ebrahimi, K.-J. Oh, F. M. B. Pereira, A. M. G. Pinheiro, I. Tabus, and Z. Chen, "JPEG Pleno: a standard framework for representing and signalling plenoptic modalities," *Proc. SPIE*, vol. 10752, pp. 107521P, 2018.

[2] D. Liu, P. An, R. Ma, W. Zhan, X. Huang, and A. A. Yahya, "Content-based light field image compression method with gaussian process regression," *IEEE Trans. Multimedia*, 2019.

[3] C. Conti, L. D. Soares, and P. Nunes, "Light field coding with field-of-view scalability and exemplar-based interlayer prediction," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 2905–2920, Nov. 2018.

[4] C. Conti, P. Nunes, and L. D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*, Jul. 2016.

[5] R. Monteiro, L. Lucas, C. Conti, P. Nunes, N. Rodrigues, S. Faria, C. Pagliari, E. da Silva, and L. Soares, "Light field hevc-based image coding using locally linear embedding and self-similarity compensated prediction," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*, 2016, pp. 1–4.

[6] Y. Li, M. Sjostrom, and R. Olsson, "Coding of focused plenoptic contents by displacement intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1308–1319, Jun. 2016.

[7] R. Zhong, S. Wang, B. Cornelis, Y. Zheng, J. Yuan, and A. Munteanu, "Efficient directional and l1-optimized intra-prediction for light field image compression," in *IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2017, pp. 1172–1176.

[8] Y. Li, R. Olsson, and M. Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*, 2016.

[9] M. Rizkallah, T. Maugey, C. Yaacoub, and C. Guillemot, "Impact of light field compression on focus stack and extended focus images," in *European Signal Processing Conf. (EUSIPCO)*, Aug. 2016, pp. 898–902.

[10] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo-sequence-based 2-d hierarchical coding structure for light-field image compression," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1107–1119, Oct. 2017.

[11] C. Jia, Y. Yang, X. Zhangy, X. Zhang, S. Wangx, S. Wang, and S. Ma, "Optimized inter-view prediction based light field image compression with adaptive reconstruction," in *IEEE Int. Conf. Image Process. (ICIP)*, 2017.

[12] W. Ahmad, M. Ghafoor, S. A. Tariq, A. Hassan, M. Sjostrom, and R. Olsson, "Computationally efficient light field image compression using a multiview hevc framework," *IEEE access*, vol. 7, pp. ss. 143002–143014, 2019.

[13] X. Jiang, M. Le Pendu, and C. Guillemot, "Light field compression using depth image based view synthesis," in *IEEE Int. Conf. Multimed. Expo Workshops (ICMEW)*, Jul. 2017.

[14] I. Tabus, P. Helin, and P. Astola, "Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000," in *IEEE Int. Conf. Image Process. (ICIP)*, 2017, pp. 4567–4571.

[15] S. Zhao and Z. Chen, "Light field image coding via linear approximation prior," in *IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4562–4566.

[16] F. Hawary, C. Guillemot, D. Thoreau, and G. Boisson, "Scalable light field compression scheme using sparse reconstruction and restoration," in *IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3250–3254.

[17] E. Dib, M. L. Pendu, and C. Guillemot, "Light field compression using fourier disparity layers," in *IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3751–3755.

[18] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. B. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4d dct-based lenslet light field codec," in *IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 435–439.

[19] T.-H. Tran, Y. Baroud, Z. Wang, S. Simon, and D. Taubman, "Light-field image compression based on variational disparity estimation and motion-compensated wavelet decomposition," in *IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017.

[20] X. Jiang, M. Le Pendu, R. Farrugia, and C. Guillemot, "Light field compression with homography-based low-rank approximation," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1132–1145, Oct. 2017.

[21] E. Dib, M. Le Pendu, X. Xiang, and C. Guillemot, "Super-ray based low rank approximation for light field compression," in *Data Compression Conference (DCC)*, Mar. 2019.

[22] R. Verhack, T. Sikora, G. Van Wallendael, and P. Lambert, "Steered mixture-of-experts for light field images and video: Representation and coding," *IEEE Trans. Multimedia*, 2019.

[23] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot, "Geometry-aware graph transforms for light field compact representation," *IEEE Trans. Image Process.*, vol. 29, pp. 602–616, 2020.

[24] M. Le Pendu, C. Guillemot, and A. Smolic, "A fourier disparity layer representation for light fields," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5740–5753, Nov. 2019.

[25] M. Le Pendu, C. Guillemot, and D. Thoreau, "Local inverse tone curve learning for high dynamic range image scalable compression," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5753–5763, Dec. 2015.

[26] G. Ward and M. Simmons, "JPEG-HDR: A backwards-compatible, high dynamic range extension to JPEG," *13th Color Imaging Conference*, pp. 283–290, Jul. 2005.

[27] R. Mantiuk, A. Efremov, K. Myszkowski, and H.-P. Seidel, "Backward compatible high dynamic range MPEG video compression," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 713–723, Jul. 2006.

[28] M. Okuda and N. Adami, "Two-layer coding algorithm for high dynamic range images based on luminance compensation," *J. Vis. Commun. Image. R.*, vol. 18, pp. 377–386, Oct. 2007.

[29] J.-U. Garbas and H. Thoma, "Inter-layer prediction for backwards compatible high dynamic range video coding with SVC," *Picture Coding Symposium (PCS)*, May 2012.

[30] ISO/IEC JTC 1/SC29/WG1 JPEG, "JPEG Pleno database," 2019, Available: https://jpeg.org/plenodb/lf/pleno_lf/.

[31] F. Pereira, C. Pagliari, E. da Silva, I. Tabus, H. Amirpour, M. Bernardo, and A. Pinheiro, *JPEG Pleno Light Field Coding Common Test Conditions*, Standard ISO/IEC JTC 1/SC29/WG1N84025, 84th Meeting, Brussels, Belgium, 2019.