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 multi-view 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.
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:

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

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 $\omega_x, \omega_y$ are the spatial horizontal and vertical frequencies respectively. Thanks to this linear relationship, the inverse problem that consists in computing 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 $L_{l \rightarrow r} = \sum_{k=l}^{r} L_k$, 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 $L_{1 \rightarrow \infty}$ (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 $L_{l \rightarrow r}$ of a parent node (with $l < r$), and a split index $s \in \{l, r-1\}$, the two children are defined as $L_{l \rightarrow s}$ and $L_{s+1 \rightarrow r}$ (see Fig. 1(b,c)). In our implementation, the split index is set as $s = \lceil (l + r) / 2 \rceil$ 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 $L_{l \rightarrow r}$ is associated with a disparity value $d_{l \rightarrow r} = \sum_{k=l}^{r} d_k$. 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 $L_{l \rightarrow r}$ is equal to the sum of its two children layers $L_{l \rightarrow s}$ and $L_{s+1 \rightarrow r}$. Therefore, the two children layers do not need to be directly encoded. Instead, we define a single split image $L_{l \rightarrow r}$ that only contains the information of the difference between the two layers:

$$L_{l \rightarrow r} = \frac{L_{l \rightarrow s} - \alpha \cdot L_{s+1 \rightarrow r}}{\alpha + 1},$$

where $\alpha$ 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 $\alpha$ compensates for the fact that both children layers may have significantly different energy when the number of original layers summed to form $L_{l \rightarrow s}$ and $L_{s+1 \rightarrow r}$ are different. In our implementation we fix $\alpha$ as the ratio between the numbers of layers present in $L_{l \rightarrow s}$ and in $L_{s+1 \rightarrow r}$. It is expressed as $\alpha = \frac{L_{s+1 \rightarrow r}}{L_{l \rightarrow s}}$. Therefore, $\alpha$ does not need to be transmitted.

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

$$L_{l \rightarrow s} = \frac{\alpha}{\alpha + 1} L_{l \rightarrow r} + L_{s+1 \rightarrow r},$$

$$L_{s+1 \rightarrow r} = \frac{1}{\alpha + 1} L_{l \rightarrow r} - L_{s+1 \rightarrow r}.$$
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.

3.1. Normalisation

and 1 sign bit). Ideally, the parameter $M$ should be computed as $M = \max(I) - m$ so that $I$ 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:

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

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 $L_{l,r}$ and its corresponding split image $L_{s,l,r}$ (see Fig. 1(a) and (d)). While the texture content of $L_{l,r}$ also appears in $L_{s,l,r}$, 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
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.

![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).](image)

![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.](image)

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

<table>
<thead>
<tr>
<th></th>
<th>WASP</th>
<th>MuLE</th>
<th>HEVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD PSNR</td>
<td>2.27 dB</td>
<td>1.35 dB</td>
<td>2.27 dB</td>
</tr>
<tr>
<td>BD Rate</td>
<td>-43.8%</td>
<td>-36.0%</td>
<td>-45.5%</td>
</tr>
</tbody>
</table>

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 inconsistenci-
6. REFERENCES


