GRAPH-BASED LIGHT FIELDS REPRESENTATION AND CODING USING GEOMETRY INFORMATION

Xin Su, Mira Rizkallah, Thomas Maugey and Christine Guillemot

INRIA, Campus de Beaulieu, Rennes 35042, France

ABSTRACT

This paper describes a graph-based coding scheme for light fields (LF). It first adapts graph-based representations (GBR) to describe color and geometry information of LF. Graph connections describing scene geometry capture inter-view dependencies. They are used as the support of a weighted Graph Fourier Transform (wGFT) to encode disoccluded pixels. The quality of the LF reconstructed from the graph is enhanced by adding extra color information to the representation for a sub-set of sub-aperture images. Experiments show that the proposed scheme yields rate-distortion gains compared with HEVC based compression (directly compressing the LF as a video sequence by HEVC).

***Index Terms*—** Graph Based Representation (GBR), Graph Fourier Transform (GFT), Compression, Light fields (LF).

1. INTRODUCTION

Light fields (LF) have emerged as a representation of light rays emitted by a 3D scene and received by an observer at a particular point (x, y, z) in space, along different orientations. A variety of capturing devices have been designed based on camera arrays [1], on single cameras mounted on moving gantries, or on arrays of microlenses placed in front of the photosensor to obtain angular information about the captured scene [2, 3].

The problem of LF compression rapidly appeared as quite critical given their significant demand in terms of storage capacity. Classical block-based coding schemes such as JPEG applied for each image of the 2D array of images forming the lumigraph have been quite naturally considered yielding however limited compression performances (compression factors not exceeding 20 for an acceptable quality) [4]. A method based on video compression is presented in [5] where a few views are encoded in Intra while the other views are encoded as P-images in which each block can be predicted from one of the neighboring Intra views with or without disparity compensation, the choice of the prediction mode being made to optimize a rate-distortion measure. A second scheme is presented where several predictions of a view are computed from neighboring views using disparity maps, and averaged to give the final predicted view. The prediction residue is then encoded using classical coding tools (DCT, quantization). Multiview video compression and disparity compensation techniques are considered in [5, 6], and intra coding modes have also been proposed in [7] for LF compression using HEVC. The authors of [8] exploits inter-view correlation by using a homography-based low rank approximation of the LF, showing significant gains compared to HEVC Inter-coding for real LF captured by micro-lenses based devices.

In this paper, we explore the use of GBR for LF. GBR has been proposed for describing the geometry of multi-view images, first for horizontally aligned cameras [9] and more recently for complex camera configurations [10]. Here, we consider GBR to represent LF using 3D geometry information. The graph connections are derived from the disparity and hold just enough information to synthesize other sub-aperture images from one reference image of the LF. Based on the concept of epipolar segment, the graph connections are sparsified (less important segments are removed) by a rate-distortion optimization. The graph vertices and connections are compressed using HEVC [11]. The graph connections capturing the inter-view dependencies are used as the support of a Graph Fourier Transform [12] used to encode disoccluded pixels.

However, the graph mostly represents scene geometry. Texture information is limited to a reference view and disoccluded pixels, which is not sufficient for reaching a high reconstructed LF quality. The bitrate distribution between texture and geometry (i.e. depth) is a key issue in view synthesis from multi-view data and depends on the camera configuration [13]. To enhance the quality of the reconstructed LF, the residuals of a subset of views are added to the graph representation. Experiments with synthetic LF from the dataset in [14] rendered with Blender [15] show that the proposed scheme achieves higher reconstruction quality at low rates compared with traditional video compression by HEVC.

2. LIGHT FIELDS GEOMETRY

We consider the simplified 4D representation of LF describing the radiance along rays by a function $L(x, y, u, v)$ of 4 parameters at the intersection of the light rays with 2 parallel planes. This representation can be seen as an array of multi-view images $\{\mathcal{I}_{u,v}\}$. Each view $\mathcal{I}_{u,v} \in \mathbb{R}^{X\times Y\times 3}$ at position (u, v) is an RGB image with $X \times Y$ pixels. Given a pixel (x, y) in $\mathcal{I}_{u,v}$, its *corresponding* pixel in $\mathcal{I}_{u',v'}$ (the pixel corresponding to the same 3D point in the real world), should have the same color values under the Lambertian assumption. In principle, multiple views of a scene can be rendered from one unique view with the help of scene geometry. This is the core idea of depth image based rendering (DIBR). For instance, given a LF dataset with available depth images $\{\mathcal{Z}_{u,v}\}$, pixel (x', y') in $\mathcal{I}_{u',v'}$ corresponding to the same 3D point as the pixel (x, y) in $\mathcal{I}_{u,v}$ can be located by

$$\begin{aligned} &(x', y') = (x + d_x, y + d_y)\ , \\ &d_x = \tfrac{B*(u-u')*f}{Z_{u,v}(x,y)}\ , d_y = \tfrac{B*(v-v')*f}{Z_{u,v}(x,y)}\ \ , \end{aligned} \quad (1)$$

where B is the distance between neighboring cameras, f is the focal length, $Z_{u,v}(x, y)$ is the depth of pixel (x, y) in $\mathcal{I}_{u,v}$. View $\mathcal{I}_{u',v'}$

thus can be rendered pixel by pixel by Eq.(1). (d_x, d_y) is also known as disparity. In the tests, we consider synthetic LF [14] for which depth information is available. For real LF, depth has to be estimated using for example the methods in [16, 17].

Pixels in different views corresponding to the same 3D point have same or similar color values. In this paper, we represent inter-view dependencies in LF with a graph using geometry information, and use the graph as a support to encode the color information using graph-based transform coding.

3. GRAPH REPRESENTATION

3.1. Graph construction

Let us denote the graph by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where vertices $\mathcal{V} = \{v_i\}$ correspond to each pixel in sub-aperture images $\{\mathcal{I}_{u,v}\}$, and edges $\mathcal{E} = \{e_{ij}\}$ connect pairs of pixels across two images.

Graph connections for reference image. As shown in Fig.1.a, image $\mathcal{I}_{1,1}$ (left bottom corner image marked in red) is selected as the reference view. Pixels on each row of $\mathcal{I}_{1,1}$ are grouped into a set of *straight* horizontal segments based on their depth. One segment has a constant depth. As shown in Fig.1.b, one row in $\mathcal{I}_{1,1}$ has been divided into 3 segments. Every segment in $\mathcal{I}_{1,1}$ is connected to one segment in every sub-aperture image by one graph edge, since the two segments correspond to the same straight segment in the real 3D world. For instance in the toy example in Fig.1.a, the reference view $\mathcal{I}_{1,1}$ is connected with every sub-aperture image $\mathcal{I}_{u,v}$ by graph edges. However, for one straight segment in $\mathcal{I}_{1,1}$, all its connections to other sub-aperture images can be deduced from each other by Eq.(1). Therefore, for one segment in $\mathcal{I}_{1,1}$, only one of its connections is necessary in the final graph structure. In our GBR, we only keep the graph connections between $\mathcal{I}_{1,1}$ and $\mathcal{I}_{1,2}$ (the right sub-aperture image of $\mathcal{I}_{1,1}$), as shown in Fig.1.a, the connections marked as red solid line are kept and the other connections marked as black doted lines are redundant and removed. Fig.1.c gives an illustration of the kept graph connections between $\{\mathcal{I}_{1,1}\}$ and $\{\mathcal{I}_{1,2}\}$.

To simplify the graph representation, each graph connection is represented by a one-dimensional metric namely unidimensional disparity based on the *epipolar segment* concept [10]. The epipolar segment is a line segment consisting of all possible projections of a pixel with varying depth. The unidimensional disparity actually is the distance between the start point of the epipolar segment and the position of the true projection.

Graph connections for disoccluded pixels. Besides the reference image $\mathcal{I}_{1,1}$, the disoccluded pixels which are not visible in $\mathcal{I}_{1,1}$ are also considered in the graph construction. For the sake of simplicity, we only consider the disoccluded pixels in $\mathcal{I}_{U,V}$ (the top right corner image in Fig.1.a), since most of the disoccluded pixels in images $\{\mathcal{I}_{u,v}\}$ $(1 < u < U, 1 < v < V)$ are visible in $\mathcal{I}_{U,V}$. To construct the graph connections for the disoccluded pixels, the same strategy has been applied here. In other words, these disoccluded pixels are treated as "*reference pixels*" for other sub-aperture images.

3.2. Graph sparsification

As presented in [10, 18], the constructed graph in section 3.1 is sparsified based on a rate-distortion model,

$$\mathcal{J}(\mathcal{E}) = \mathcal{D}(\mathcal{E}) + \alpha \mathcal{R}(\mathcal{E}) \;, \tag{2}$$

where $\mathcal{J}$ is the Lagrangian cost (smaller $\mathcal{J}$ values mean better optimal status), $\mathcal{D}$ is the distortion of rendered sub-aperture images and $\mathcal{R}$ is the modeled bitrate cost for coding the graph connections. α is the Lagrangian multiplier which represents the relation between bitrate and rendering quality (distortion). To decrease the computational cost we compute the rendering distortion on only a subset of views. Edges are removed based on the shortest path optimization of

$$\mathcal{E} = \underset{\mathcal{E}}{\text{argmin}}\; \mathcal{J}(\mathcal{E}). \tag{3}$$

Graph sparsification does not only reduce bitrate cost but also corrects errors in the depth, since the optimization modifies graph connections regarding rendering distortion. For real LF with estimated depth, it is very useful due to noise or errors in the estimated depth.

3.3. Graph with Residuals

So far, the constructed graph contains minimum amount of color information, since only the reference view $\mathcal{I}_{1,1}$ and the disoccluded pixels in $\mathcal{I}_{U,V}$ are kept. To enhance the quality of the reconstructed views, residues $r_{m,n}$ between a subset of M rendered images (from the graph) $\tilde{\mathcal{I}}_{m,n}$ and the original true images $\mathcal{I}_{m,n}$, computed as $r_{m,n} = \mathcal{I}_{m,n} - \tilde{\mathcal{I}}_{m,n}$ are added to the graph.

At the decoder, these selected sub-aperture images are also treated as "*reference images*" to render the remaining sub-aperture images. The depth of each straight segment in the reference image $\mathcal{I}_{1,1}$ is estimated from the corresponding graph connections by Eq.(1). Then, the depth of the selected images $\mathcal{I}_{m,n}$ is computed by projection from the estimated depth of $\mathcal{I}_{1,1}$. We compute each remaining sub-aperture image $\mathcal{I}_{m,n}$ by combining $M + 1$ rendered images, one image recovered from the graph and M images warped from the selected *reference images*.

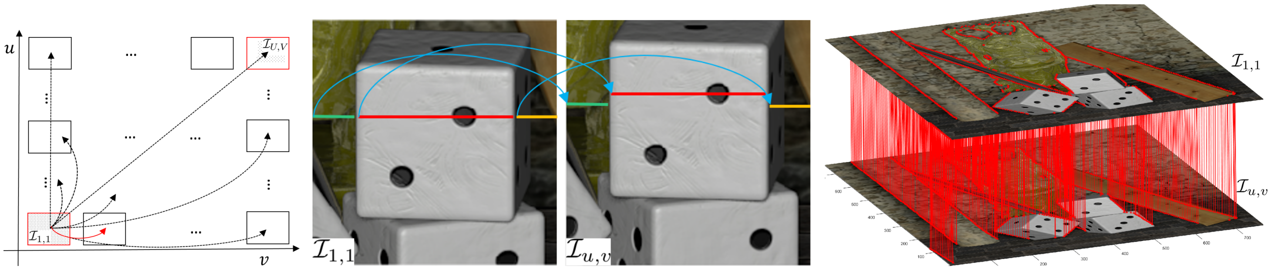
$$\hat{\mathcal{I}}_{u,v} = \frac{1}{\sum w_i} \left(w_0 \tilde{\mathcal{I}}_{u,v} + \sum_{i=1}^{M} w_i \, \tilde{\mathcal{I}}_{u,v} \Big|_{\hat{\mathcal{I}}_{m,n}} \right),$$

$$[w_0, w_1, ...w_M]^T = \mathbf{Rxy} \left(\tilde{\mathcal{I}}_{u,v} \mid_{\hat{\mathcal{I}}_{m,n}}, \mathcal{I}_{u,v} \right) \mathbf{Rxx} \left(\tilde{\mathcal{I}}_{u,v} \mid_{\hat{\mathcal{I}}_{m,n}} \right)^{-1}$$

where weights $[w_0, w_1, ...w_M]^T$ are computed using the minimum mean square error estimation theory. $\mathbf{Rxy} \left(\tilde{\mathcal{I}}_{u,v} \mid_{\hat{\mathcal{I}}_{m,n}}, \mathcal{I}_{u,v} \right)$ is the cross-correlation of the $M + 1$ rendered images and the original image $\mathcal{I}_{u,v}$, and $\mathbf{Rxx} \left(\tilde{\mathcal{I}}_{u,v} \mid_{\hat{\mathcal{I}}_{m,n}} \right)$ is the autocorrelation of the $M + 1$ rendered images.

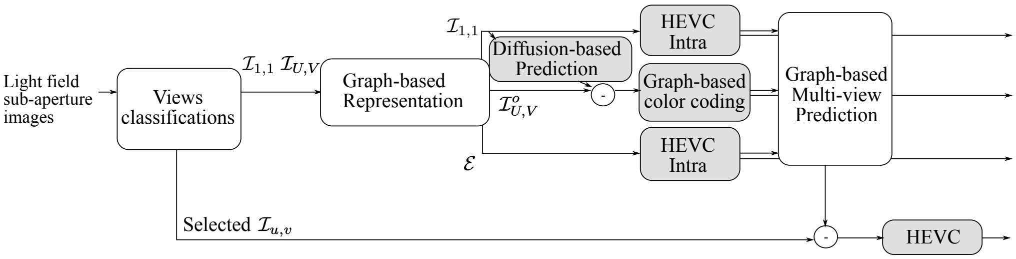
4. CODING SCHEME

The proposed encoder is shown in Fig. 2. As explained in Section 3, from two sub-aperture images, namely the corner images $\mathcal{I}_{1,1}$ and $\mathcal{I}_{U,V}$, we construct the LF graph representation ($\mathcal{G} = (\mathcal{V}, \mathcal{E})$). The graph edges $\mathcal{E}$ are stored in a grey-level image which is coded using HEVC (More details about the graph edges coding can be found in [10]). The vertices are pixels in the images $\mathcal{I}_{1,1}$ and $\mathcal{I}^o_{U,V}$ (the parts of $\mathcal{I}_{U,V}$ that do not appear in $\mathcal{I}_{1,1}$). A part of the graph is depicted in Fig. 3 where blue segments are edges with a small weight (0.5) whereas red ones are edges with high weight(1). While $\mathcal{I}_{1,1}$ is classically compressed using HEVC, the arbitrarily shaped $\mathcal{I}^o_{U,V}$ requires dedicated tools. We propose to compress it using a graph-based compression scheme as follows.



(a) GBR for LF. (b) example of graph connections between $\mathcal{I}_{1,1}$ and $\mathcal{I}_{u,v}$. (c) Graph connections between $\mathcal{I}_{1,1}$ and $\mathcal{I}_{u,v}$.

Fig. 1. Graph based representation (GBR) adapted to LF.



Light field sub-aperture images
Views classifications
$\mathcal{I}_{1,1}$ $\mathcal{I}_{U,V}$
Graph-based Representation
$\mathcal{I}_{1,1}$
Diffusion-based Prediction
$\mathcal{I}^{o}_{U,V}$
$\mathcal{E}$
HEVC Intra
Graph-based color coding
HEVC Intra
Graph-based Multi-view Prediction
-
-
Selected $\mathcal{I}_{u,v}$
HEVC

Fig. 2. Proposed encoder

Let $S = [S_1 S_2]$ be the vector of all color values in $\mathcal{V}$ to be coded, where S_1 comprises the color values of the reference image $\mathcal{I}_{1,1}$ (separately coded with HEVC), and S_2 the color values of the disoccluded pixels $\mathcal{I}^{o}_{U,V}$. The color values in the reference view are initially propagated to the disoccluded pixels using an iterative diffusion method. More precisely, at the first iteration, the pixels at the borders of the disocclusion areas are predicted by computing a weighted average of their 1−hop neighborhood in the reference image. For example, the prediction of a disoccluded pixel p_1 connected to four pixels in the reference view (p_2, p_3, p_4, p_5) is computed as

$$\frac{w_{12}p_2 + w_{13}p_3 + w_{14}p_4 + w_{15}p_5}{w_{12} + w_{13} + w_{14} + w_{15}}$$

where w_{ij} denotes the weight of the connection between the pixels i and j. In practice, the weight values are always 1 except where the depth difference exceeds threshold $\frac{Z_{\max}-Z_{\min}}{20}$ ($Z_{\max}$ and $Z_{\min}$ are maximum and minimum values of the depth image). In that case, a lower weight is assigned to attenuate the color propagation. The predicted pixels are then used to predict other disoccluded pixels in the following iteration.

To code the prediction residuals of the disoccluded pixels, i.e., $R = S_2 - E(S_2/S_1)$, we use the weighted Graph Fourier Transform (wGFT) [12] . The target disocclusion image is divided in 8×8 pixel blocks. In each block, we use the 4-neighbors graph which connects the disoccluded pixels to transform the residuals. More specifically, given the weight matrix W, we define the diagonal degree matrix D, where $D_{ii} = \sum_j w_{ij}$. Lastly, the graph normalized weighted Laplacian matrix L_{norm} is computed as $L_{\text{norm}} = I - D^{-1/2}WD^{-1/2}$. Let Ψ be the matrix whose columns contain the wGFT basis i.e., the eigenvectors of the graph normalized laplacian. The residuals are thus projected on the wGFT basis as $\hat{R} = \Psi R$. The coefficients are quantized for various quality factors following the method in [19], entropy coded then sent to the decoder side.

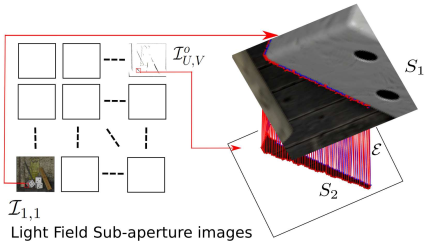


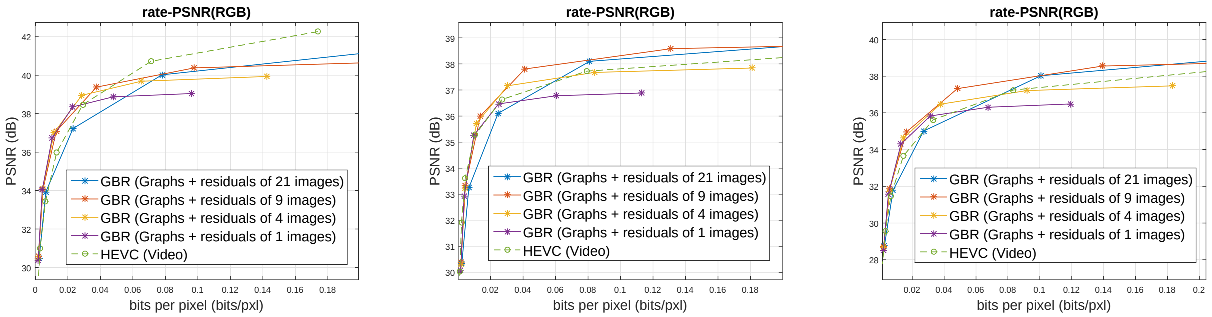
Fig. 3. A part of the graph drawn between the pixels of the reference image $\mathcal{I}_{1,1}$ and the disocclusions image $\mathcal{I}^{o}_{U,V}$. Red and Blue connections have 1 and 0.5 as weights respectively

Because the decoder already received the disparity information in the graph-based representation, it can deduce the exact same locations of disoccluded pixels in the target image as the encoder. It builds the same 4-neighbors graph connecting the disoccluded pixels, computes the edge weights using the disparity information and derives the same transform basis.This computation is required only for few blocks containing the disoccluded pixels. Also, there is no need to send additional side information as done in edge-adaptive approaches [20, 21].

Finally, the remaining views $\mathcal{I}_{u,v}$ are coded as follows. They are first predicted using the graph-based representation. Then, a residual is computed with the true $\mathcal{I}_{u,v}$. This residual is further compressed with HEVC.

5. EXPERIMENTS

We test our GBR on synthetic LF (with $U = 9$, $V = 9$, $X = 768$ and $Y = 768$) from the dataset in [14] rendered with Blender [15]. Three datasets, called *Buddha*, *butterfly* and *monasRoom*, have been tested here.

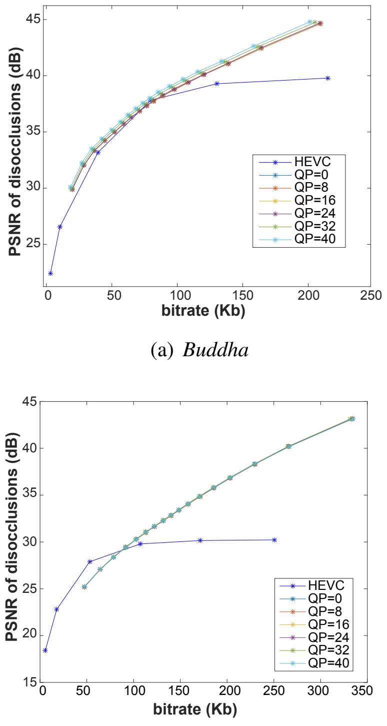


(a) *buddha*

(b) *butterfly*

(c) *monasRoom*

Fig. 5. PSNR-rate performance of the proposed GBR on different datasets, (a) *buddha*, (b) *butterfly* and (c) *monasRoom*.



(a) *Buddha*

(b) *monasRoom*

Fig. 4. Results of coding the disoccluded pixels using a Graph-based approach(QF from 10 to 90) vs HEVC (QP 0 to 40) for *Buddha* and *monasRoom*

5.1. Evaluation of GFT

To show the interest of exploiting inter-view neighboring relations(i.e graph edges) in coding the disocclusions, we first compare the performance of our graph-based compression scheme against HEVC inter-coding. We first code the disoccluded parts along with the reference view as a video sequence using HEVC. We vary the QP from 0 to 40. For each QP, a prediction of the disocclusions is computed(Sec. 4), then the residuals are coded while varying the quality factor from 10 to 90. The bitrate is the one needed to code the disocclusions. The PSNR is measured taking as reference the original disocclusions color values. From the results (Fig. 4), we notice that our approach outperforms HEVC with a higher PSNR for most QP values while preserving acceptable bitrates. Our diffusion method yields a good prediction with *Buddha* since the background mostly consists of smooth regions, and that explains the better coding performance. Whereas for *monasRoom*, the background is made of texture and wrong color values are propagated to the disoccluded areas resulting in residuals harder to code.

5.2. Light field representation and compression

We perform the GBR representation with fixed Lagrangian multiplier $\alpha = 0.5$ in Eq.(2). In this case, the graph sparsification highly depends on the distortion term $\mathcal{D}(\mathcal{E})$. The number of sub-aperture images selected to add residuals is chosen as $\{1, 4, 9, 21\}$ with a regular sub-sampling pattern. The baseline method is the scheme which directly compresses the whole LF dataset as a video sequence with HEVC. Fig.5 shows the PSNR-rate performance of the proposed GBR on different datasets. At low bitrate, the proposed GBR can yield PSNR-rate gain. However, at high bitrate, the GBR scheme is outperformed by HEVC, due to the limited number of selected sub-aperture images. More results (including visual results of rendered views) can be found on the web page `https://www.irisa.fr/temics/demos/lightField/GBR/GBR_LF_2017.html`. The proposed method is only tested on the synthetic light fields data, since the accurate depth or disparity information is needed.

6. CONCLUSION

In this paper, we have adapted the graph based representation (G-BR) [10] to represent light fields (LF). The weighted Graph Fourier Transform (wGFT) is applied on the constructed graph to code the disoccluded pixels. To improve the rendering quality, the residuals of a sub-set of views are added into the graph and further used to render the other views of the LF. Experimental results show rate-distortion gain compared with HEVC based compression. For future work, we will focus on the application of our method to the real light fields.

7. ACKNOWLEDGEMENT

This project has been supported in part by the EU H2020 Research and Innovation Programme under grant agreement No 694122 (ERC advanced grant CLIM).

8. REFERENCES

[1] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy, "High performance imaging using large camera arrays," in *ACM Transactions on Graphics (TOG)*. ACM, 2005, vol. 24, pp. 765–776.

[2] Ren Ng, *Light field photography*, Ph.D. thesis, Stanford University, 2006.

[3] Todor Georgiev, Georgi Chunev, and Andrew Lumsdaine, "Superresolution with the focused plenoptic camera," in *Computational Imaging*, 2011, pp. 78 730X–78 730X–13.

[4] Gavin Miller, Steven Rubin, and Dulce Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," in *Rendering Techniques 98*, pp. 281–292. Springer, 1998.

[5] Marcus Magnor and Bernd Girod, "Data compression for light-field rendering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 338–343, 2000.

[6] Chuo-Ling Chang, Xiaoqing Zhu, Prashant Ramanathan, and Bernd Girod, "Light field compression using disparity-compensated lifting and shape adaptation," *IEEE transactions on image processing*, vol. 15, no. 4, pp. 793–806, 2006.

[7] Yun Li, Marten Sjostrom, Roger Olsson, and Ulf Jennehag, "Efficient intra prediction scheme for light field image compression," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 539–543.

[8] Xiaoran Jiang, Mikaël Le Pendu, Reuben A Farrugia, Sheila S Hemami, and Christine Guillemot, "Homography-based low rank approximation of light fields for compression," in *IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[9] Thomas Maugey, Antonio Ortega, and Pascal Frossard, "Graph-based representation for multiview image geometry," *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1573–1586, 2015.

[10] Xin Su, Thomas Maugey, and Christine Guillemot, "Rate-Distortion Optimized Graph-Based Representation for Multiview Images With Complex Camera Configurations," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2644–2655, 2017.

[11] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[12] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[13] Emilie Bosc, Vincent Jantet, Muriel Pressigout, Luce Morin, and Christine Guillemot, "Bit-rate allocation for multi-view video plus depth," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[14] Sven Wanner, Stephan Meister, and Bastian Goldluecke, "Datasets and Benchmarks for Densely Sampled 4D Light Fields," in *VMV*. Citeseer, 2013, pp. 225–226.

[15] Blender, "Blender," `https://www.blender.org/`, [Online].

[16] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673–680.

[17] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon, "Accurate depth map estimation from a lenslet light field camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1547–1555.

[18] Xin Su, Thomas Maugey, and Christine Guillemot, "Graph-based representation for multiview images with complex camera configurations," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1554–1558.

[19] Jesse D Kornblum, "Using JPEG quantization tables to identify imagery processed by software," *Digital Investigation*, vol. 5, pp. S21–S25, 2008.

[20] Godwin Shen, W-S Kim, Sunil K Narang, Antonio Ortega, Jaejoon Lee, and Hocheon Wey, "Edge-adaptive transforms for efficient depth map coding," in *Picture Coding Symposium (PCS), 2010*. IEEE, 2010, pp. 566–569.

[21] Hilmi E Egilmez, Amir Said, Yung-Hsuan Chao, and Antonio Ortega, "Graph-based transforms for inter predicted video coding," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3992–3996.