Light field compression using depth image based view synthesis

Xiaoran Jiang, Mikaël Le Pendu, Christine Guillemot

LIGHT FIELD COMPRESSION USING DEPTH IMAGE BASED VIEW SYNTHESIS

Xiaoran Jiang, Mikaël Le Pendu, Christine Guillemot

INRIA, Campus de Beaulieu, Rennes, France
name.surname@inria.fr

ABSTRACT

This paper describes a novel light field compression scheme using a depth image-based view synthesis technique. A small subset of views is compressed with HEVC inter coding and then used to reconstruct the entire light field. The residual of the whole light field can be then restructured as a video sequence and encoded by HEVC inter coding. Experiments show that our scheme significantly outperforms a similar view synthesis method which utilizes convolutional neural networks, and does not require training with a large dataset of light fields as required by deep learning techniques. It also outperforms as well the direct encoding of all the light field views.

***Index Terms*—** Light fields, Compression, Depth image based rendering, View synthesis, Convolutional neural networks

1. INTRODUCTION

During the last two decades, there has been a growing interest in light field imaging. Light fields capture the radiance of a dense set of rays emitted by a scene along various directions. This rich scene description enables post-capture image creation with a variety of amazing features such as digital refocusing, change of focal length, change of viewpoint, scene depth estimation, 3D scene reconstruction, to name a few. Effort has been dedicated to light field camera design, going from camera arrays [1] or single cameras mounted on moving gantries, both yielding a wide baseline, to plenoptic cameras using arrays of micro-lenses placed in front of the photosensor leading to light fields with narrow baselines [2], [3].

The problem of light field compression rapidly appeared as quite critical given their significant demand in terms of storage capacity. First methods for compressing synthetic light fields appeared late 90's essentially based on classical coding tools as vector quantization followed by Lempel-Ziv (LZ) entropy coding [4] or wavelet coding as in [5] and [6], yielding however limited compression performances (compression factors not exceeding 20 for an acceptable quality). Predictive schemes inspired from video compression methods have then been naturally investigated, adding specific prediction modes, as in [7] and [8] where the proposed schemes are inspired from H.264 and MVC. The latest HEVC video compression standard has then been naturally considered for light fields, capitalising on advances in the video compression field. With the emergence of plenoptic cameras, two main directions have been followed: either coding the array of sub-aperture images as in [9] after extraction from the lenslet image after de-vignetting and de-mosaicing, or directly encoding of the lenslet images captured by plenoptic cameras [10], [11]. Approaches based on HEVC mostly focus on the introduction of dedicated prediction modes. A scalable extension of HEVC-based scheme is proposed in [12] where a sparse set of micro-lens images (also called elemental images) is encoded in a base layer. The other elemental images are reconstructed at the decoder using disparity-based interpolation and inpainting. The reconstructed images are then used to predict the entire lenslet image and a prediction residue is transmitted yielding a multi-layer scheme.

Instead of directly encoding the light field (the array of sub-aperture images or the lenslet image for light fields captured by plenoptic cameras), the authors in [13] consider the focus stack as an intermediate representation of reduced dimension of the light field and encode the focus stack with a wavelet-based scheme. The light field is then reconstructed from the focus stack using the linear view synthesis approach described in [14]. In [15], the authors propose a homography-based low rank approximation method to construct an intermediate representation of reduced dimension which is then encoded using HEVC.

In this paper, we further investigate light field compression based on view synthesis from a subset of selected views. The problem of light field reconstruction from a subset of views has been addressed in [16] and [17] with two different approaches. The authors in [16] exploit light field sparsity in the angular continuous Fourier domain. Assuming the light field is k-sparse in the angular continuous Fourier domain, it can be represented as a linear combination of k non-zero continuous angular frequency coefficients. The reconstruction algorithm then searches for the frequency values and the corresponding coefficients. The authors in [17] instead propose a learning architecture based on two consecutive convolutional

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

neural networks (CNN). From features extracted from 4 views at the corners of the light field, the first CNN predicts depth maps which are then used to produce by warping 4 estimates of each synthesized view. A second CNN then reconstructs each light field view from these 4 estimates.

Inspired by [17], in this paper we propose to replace the neural network structure by a depth image-based rendering (DIBR) approach for light fields. The disparity maps of the 4 corner views are first estimated by optical flow techniques, before being forward warped to generate the novel disparity map of a projected position. Given this estimated disparity, the color image is finally synthesized by applying backward warping of the 4 corner views. Low rank matrix completion algorithms are applied to fill the zone of disocclusion and cracks, both for the disparity map and color image synthesis. For each novel position, a simple weighted average is considered to fuse the 4 warped estimates.

2. RELATED WORKS

2.1. Disparity estimation for light fields

In the literature, disparity estimation for light fields has been largely studied. Estimation can be performed by using microlens images, sub-aperture images, or EPIs (epipolar plane images). Methods exploiting microlens images such as [18] have advantages of being free from later processing (e.g. demoisaicing, etc.) errors. Some other works mesure the slope on EPIs, or refocused EPIs in order to estimate the scene depth. For this purpose, [19] uses a variational method, whereas [20] and [21] propose to combine defocus and correspondence cue. Occlusion is further explicitly modeled in [21]. Finally, estimations using sub-aperture images [22] [23] assume that these views are well rectified with constant baseline and use block matching techniques.

Most of these methods work on very densely sampled light fields and demand high computational cost. In our scheme of view synthesis from very sparsely sampled views, we adopt optical flow approach, which estimates efficiently disparity between a pair of views. More details will be given in Section 4.1.

2.2. CNN-based view synthesis

The learning based view synthesis described in [17] generates the target view at a novel position from a sparse set of n reference views. This model can be learnt in a machine learning fashion. It is materialized by a sequence of two successive convolutional neural networks. Instead of taking directly the pixel values of the reference views, the input features are extracted in a similar manner as described in [20]. For a given position, the n reference views are warped to this position, using a set of uniformly distributed disparity levels. Then, the pixel-wise mean and standard deviation of all the warped views at each disparity level are computed. For a specific pixel, the correct disparity level should correspond to the maximum mean contrast and the minimum standard deviation. The first CNN is supposedly able to learn this relationship and computes a disparity map for each view to be synthesized, which is then used to warp all the reference views to the novel position. The combination of these warped views is finally realized by the second CNN, which synthesizes the final color pixel values.

In this approach, the handling of occlusion is implicit. The intermediate output disparity map is not necessarily of high quality, but as the two CNNs are trained simultaneously, the second CNN tends to mitigate the final color image rendering error caused by disparity inaccuracy.

3. PROPOSED COMPRESSION SCHEME

Let $L(x, y, u, v)$ denote the 4D representation of a light field, describing the radiance of a light ray parameterized by its intersection with two parallel planes [24], and where (u, v) (with $u = 1 \dots U$ and $v = 1 \dots V$) denote the angular (view) coordinates and (x,y) (with $x = 1 \dots N_x$ and $y = 1 \dots N_y$) the spatial (pixel) coordinates. Let $I_{(u,v)}$ denote a view at the angular position (u, v) in the light field. The main steps of the proposed compression scheme are depicted in Fig. 1.

At the encoder side, only the four sub-aperture images on the extreme corners $\{I_{(1,1)}, I_{(1,V)}, I_{(U,1)}, I_{(U,V)}\}$ are encoded as a sequence by HEVC inter coding. The decoder extracts the compressed version of these four corners $\{I'_{(1,1)}, I'_{(1,V)}, I'_{(U,1)}, I'_{(U,V)}\}$ before the corresponding disparity maps being estimated by using optical flow estimation methods. The other views are then synthesized by a depth image-based rendering approach. The details of the algorithm will be explained in Section 4.

4. DEPTH IMAGE-BASED RENDERING FOR LIGHT FIELDS

4.1. Disparity estimation using DeepFlow

DeepFlow [25] blends a matching algorithm with a variational approach for optical flow. The matching algorithm, a multilayer architecture inspired by convolutional neural networks, called Deep Matching [26], computes dense correspondences between a pair of images. The gradient histograms used by Deep Matching are robust to illumination and color changes, which makes it interesting for light field views captured by plenoptic cameras (e.g. Lytro Illum). Although DeepFlow is specially designed for computing large displacements, it works also fine with light fields captured by lenslet with relatively small baselines.

We use DeepFlow to compute disparity maps of the four corner views. Let $D_X^{(u_i,v_i)\to(u_j,v_j)}$ and $D_Y^{(u_i,v_i)\to(u_j,v_j)}$ denote the disparity on two spatial dimensions from the view

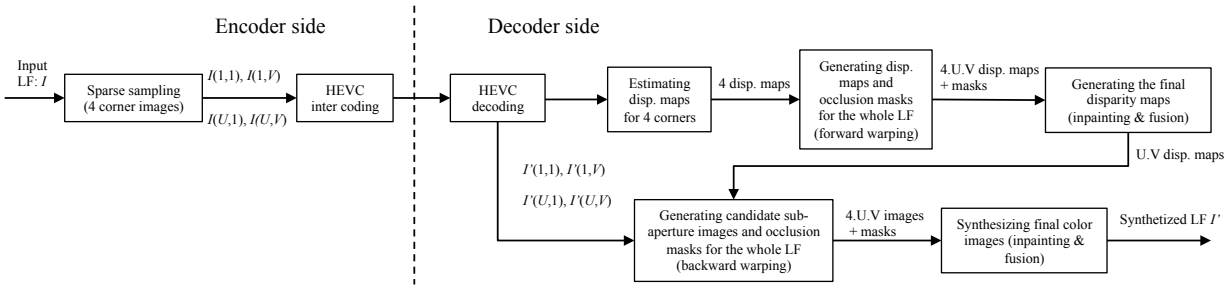

Fig. 1: Coding and decoding scheme overview.

$I_{(u_i,v_i)}$ to $I_{(u_j,v_j)}$, i.e.

$$D^{(u_i,v_i)\to(u_j,v_j)} = \begin{pmatrix} D_X^{(u_i,v_i)\to(u_j,v_j)} \\ D_Y^{(u_i,v_i)\to(u_j,v_j)} \end{pmatrix}. \tag{1}$$

Let us take the top left corner $I_{(1,1)}$ as an example. DeepFlow computes 3 disparity maps $D^{(1,1)\to(1,V)}$, $D^{(1,1)\to(U,1)}$ and $D^{(1,1)\to(U,V)}$. Contrary to other disparity estimation techniques that make use of the whole light field, DeepFlow takes each time only two views, and thus in the resulting disparity maps may reside incoherence. We propose to average these maps in order to mitigate this problem. Here, we take $U = V$. Hence, for a rectified light field, assuming that the scene is lambertian, an ideal flow estimation should give:

$$\begin{aligned} D_X^{(1,1)\to(U,V)} &= D_Y^{(1,1)\to(U,V)} \\ = D_X^{(1,1)\to(U,1)} &= D_Y^{(1,1)\to(1,V)}, \end{aligned} \tag{2}$$

and

$$D_X^{(1,1)\to(1,V)} = D_Y^{(1,1)\to(U,1)} = 0. \tag{3}$$

Then, the disparity information for the view $I_{(1,1)}$ can be described by a single map estimated as:

$$\hat{D}_{(1,1)} = \frac{1}{4U} \cdot \mathbb{1}^\top \cdot \begin{pmatrix} D_X^{(1,1)\to(U,V)} \\ D_Y^{(1,1)\to(U,V)} \\ D_X^{(1,1)\to(U,1)} \\ D_Y^{(1,1)\to(1,V)} \end{pmatrix}, \tag{4}$$

where $\mathbb{1}$ is 4×1 unit matrix.

Assuming that the baseline between views is uniform, which is the case for most plenoptic cameras and camera grid setups, $\hat{D}_{(1,1)}$ can be used to compute both the horizontal and vertical disparities between view $I_{(1,1)}$ and any other view of the light field (see Eq. (5)-(6)). The other three disparity maps $\hat{D}_{(U,1)}$, $\hat{D}_{(1,V)}$ and $\hat{D}_{(U,V)}$ are computed in a similar manner, with the signs adjusted considering their relative positions.

4.2. Disparity maps generation for all views

Given $\hat{D}_{(1,1)}$, $\hat{D}_{(U,1)}$, $\hat{D}_{(1,V)}$ and $\hat{D}_{(U,V)}$, we intend to generate the corresponding disparity map of each view in the whole LF. We project to the novel position (the view to synthesize) the 4 reference disparity maps, which are considered as monochrome images, by using the disparity information itself.

Considering the novel position (u_s, v_s). The disparity estimation based on the input position (u_i, v_i) is computed as:

$$\hat{D}^{(u_i,v_i)}_{(u_s,v_s)}(x', y') = \hat{D}_{(u_i,v_i)}(x, y), \tag{5}$$

with

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} u_s - u_i \\ v_s - v_i \end{pmatrix} \cdot \hat{D}_{(u_i,v_i)}(x, y). \tag{6}$$

This forward warping generates two types of holes. First, cracks are caused by the rounding of x', y' (rounding necessary because the projected image must be sampled at integer positions), leaving small holes between two normally adjacent pixels. Second comes the zone of disocclusion, which is visible in the novel view because of the view position change, but is occluded in the input view.

For each novel position (u_s, v_s), there are 4 such warped estimates. We thus construct the matrix M of $4{\cdot}U{\cdot}V$ columns, each column being a vectorized warped disparity image. The goal is to recover the entire matrix (fill the holes of cracks and disocclusion) by exploiting the low rank prior that is satisfied by the matrix M thanks to the high correlation between the views. The low rank matrix completion problem can be formally expressed as:

$$\begin{aligned} &\min_{\hat{M}} \text{rank}(\hat{M}) \\ &\text{s.t. } P_\Omega(\hat{M}) = P_\Omega(M), \end{aligned} \tag{7}$$

where Ω is the set of indices of the known elements in M.

To solve this, we apply the low rank matrix completion via the Inexact Augmented Lagrangian Multiplier method (IALM) [27]. This method efficiently recovers the holes since the disoccluded areas of different warped views from 4 extreme corners are unlikely to be overlapped, and the positions of cracks are dispersed.

Finally, for each position (u_s, v_s), a simple weighted average of the 4 filled disparity maps is performed to obtain the final estimate:

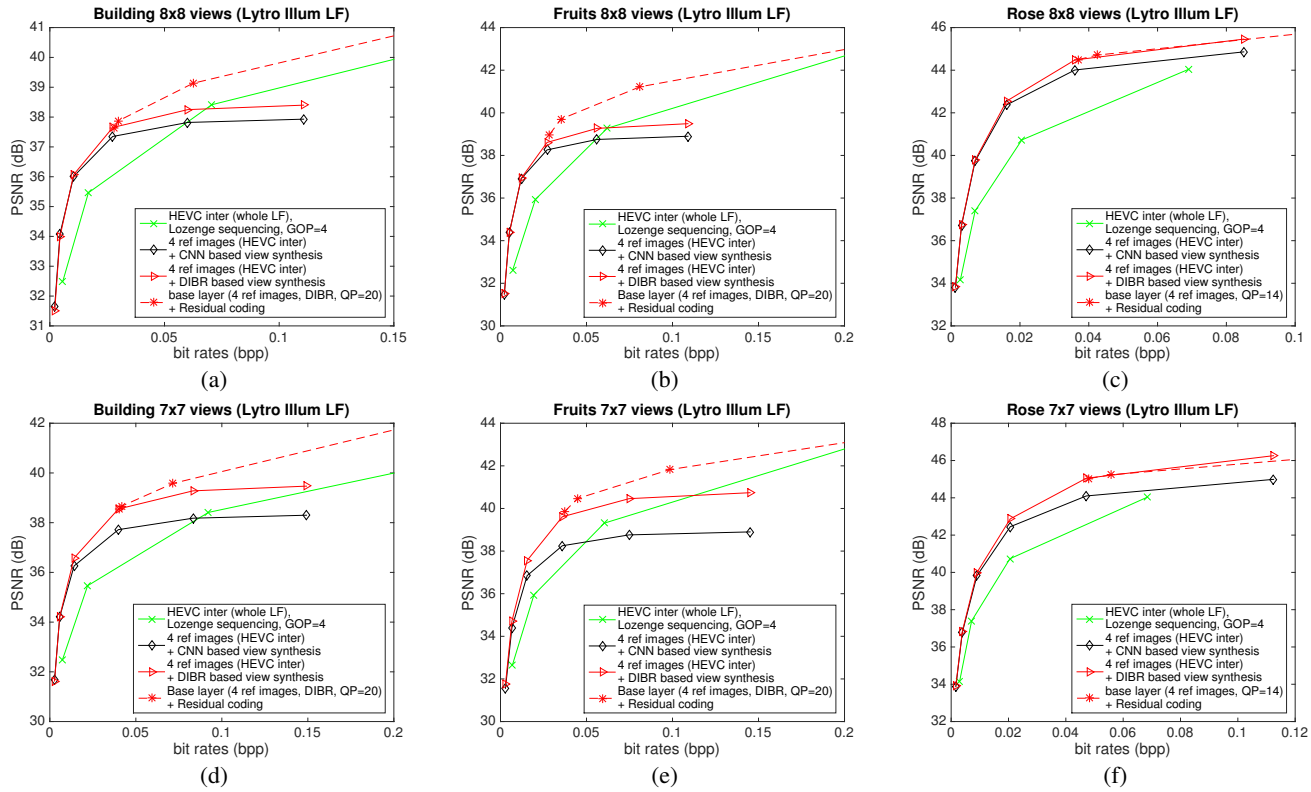

Fig. 2: PSNR-rate performance for DIBR view synthesis compression scheme (Fig. 2a - 2c for 8×8 views and Fig. 2d - 2f for 7×7 views) compared with deep learning based method and direct HEVC inter coding.

Fig. 3: Light Fields used in the tests: "Building", "Fruits" and "Rose". All of the three are captured by Lytro Illum camera and demultiplexed by Lytro Power Tools Beta software.

$$\hat{D}_{(u_s,v_s)} = \sum_i w_{(u_i,v_i)} \cdot \hat{D}^{(u_i,v_i)}_{(u_s,v_s)}, \tag{8}$$

with

$$w_{(u_i,v_i)} = \left(1 - \frac{|u_i - u_s|}{U-1}\right) \cdot \left(1 - \frac{|v_i - v_s|}{V-1}\right). \tag{9}$$

4.3. Color images generation for all views

At this stage, we dispose of disparity $\hat{D}_{(u_s,v_s)}$ for each view (u_s, v_s), and color pixels information from four corner images. To generate the novel view, backward warping is performed separately for each RGB channel:

$$\forall C \in \{R,G,B\}, \hat{I}^{(u_i,v_i)}_{C(u_s,v_s)}(x,y) = \hat{I}_{C(u_i,v_i)}(x',y'), \tag{10}$$

with

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} u_i - u_s \\ v_i - v_s \end{pmatrix} \cdot \hat{D}_{(u_s,v_s)}(x,y). \tag{11}$$

The same inpainting algorithm as in Section 4.2 is applied to fill the holes of warped color images. Note that inpainting is performed independently on each RGB component. The same weighted average (c.f. Eq. (8)-(9)) is finally performed to compute the final color images.

5. EXPERIMENT RESULTS

In this section, we evaluate our compression scheme against the CNN-based view synthesis approach. Both of them are compared with the naive HEVC inter coding of the video with a lozenge sequencing of all sub-aperture views. The used HEVC version is HM-16.10. PSNRs are computed on the luminance component Y.

For a fair comparison, we consider only real light fields captured by a Lytro Illum camera. The three test LFs we use in this comparison are "Building", "Fruits" and "Rose", the central views being shown in Fig. 3. As in [17], the extraction of sub-aperture images from RAW capture (i.e. demultiplexing) is performed by Lytro Power Tools Beta software [28]. The actual angular resolution of Lytro Illum camera is

14×14, but the remote views suffer serious vignetting and distorsion problems. Thus, in our experiments, we take only the 8×8 or 7×7 middle views. Note that the convolutional neural network used in [17] is trained exclusively with the middle 8×8 views of Illum LFs.

In Fig. 2, we observe that for the 8×8 views compression, our DIBR approach for LF synthesis slightly outperforms CNN learning based scheme. At low to middle bit-rates, both of them significantly outperforms the naive HEVC inter coding reference. At high bit-rates, however, the PSNR-rate performance is saturated by view synthesis accuracy (except for "Rose"). In fact, only 4 out of the 64 views are provided, which is very demanding for an accurate synthesis task. Nevertheless, compression performance can be further enhanced by residual coding, as shown by the red dashed lines in Fig. 2. It was generated by using a fixed QP for the base layer (4 corner views) and varying QPs for encoding the residue of all the other views. In these tests, the residue is encoded as a sequence by HEVC inter coding.

The same simulations are also performed for 7×7 views configuration. In this case, the gain of our scheme against CNN based scheme considerably increases. Another configuration such as 5×5 or 4×4 will further accentuate this performance gap, though the curves are not shown in this paper. Note that the neural network is exclusively trained for the 8×8 configuration. This demonstrates that like most of the learning based methods, the CNN based synthesis model is very subject to training data. With the change of angular resolution or the change of LF capture devices (different plenoptic camera LFs, camera grids LFs, synthetic LFs, etc.), the model should be retrained, at the best some fine-tuning operation is required. On the contrary, our method is data-independent.

6. CONCLUSION

In this paper, we have proposed a light field compression scheme using depth image-based rendering approach. Only very sparse samples (4 views at the corners) of light field views are transmitted, the others are synthesized. Despite of the fact that disparity maps are required for view synthesis, this information does not need to be encoded and transmitted, since it is deduced at the decoder side. Compared to the deep learning based method which implicitly deals with occlusions, our scheme treats this explicitly. Our scheme is better in terms of PSNR-rate performance, and is data-independent. With the residue coded, our scheme also significantly outperforms HEVC inter coding for the tested light fields.

7. REFERENCES

[1] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, July 2005.

[2] R. Ng, *Light Field Photography*, Ph.D. thesis, Stanford University, 2006.

[3] T. Georgiev, G. Chunev, and A. Lumsdaine, "Super-resolution with the focused plenoptic camera," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 7873, pp. 78730X–78730X–13, 2011.

[4] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1996, SIGGRAPH '96, pp. 31–42, ACM.

[5] P. Lalonde and A. Fournier, "Interactive rendering of wavelet projected light fields," in *Int. Conference on Graphics Interface '99*. 1999, pp. 107–114, Morgan Kaufmann Publishers Inc.

[6] I. Peter and W. Straßer, "The wavelet stream - progressive transmission of compressed light field data," in *IEEE Visualization 1999 Late Breaking Hot Topics*. 1999, pp. 69–72, IEEE Computer Society.

[7] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 10, no. 3, pp. 338–343, Apr. 2000.

[8] B. Girod, C. L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated Lifting & Shape Adaptation," *Proceedings - IEEE International Conference on Multimedia and Expo*, vol. 1, no. 4, pp. I373–I376, 2003.

[9] M. Rizkallah, T. Maugey, C. Yaacoub, and C. Guillemot, "Impact of light field compression on focus stack and extended focus images," *EUSIPCO*, pp. 898–902, 2016.

[10] C. Conti, P. Nunes, and L. D. Soares, "New HEVC prediction modes for 3D holoscopic video coding," *Proceedings - International Conference on Image Processing, ICIP*, pp. 1325–1328, 2012.

[11] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Efficient intra prediction scheme for light field image compression," *ICASSP*, pp. 539–543, 2014.

[12] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 80–91, 2016.

[13] T. Sakamoto, K. Kodama, and T. Hamamoto, "A study on efficient compression of multi-focus images for dense light-field reconstruction," *VCIP*, 2012.

[14] A. Levin and F. Durand, "Linear view synthesis using dimensionality gap light field prior," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 1831–1838.

[15] X. Jiang, M. Le Pendu, R. A Farrugia, S. Hemami, and C. Guillemot, "Homography-based low rank approximation of light fields for compression," *ICASSP*, 2017.

[16] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light field reconstruction using sparsity in the continuous Fourier domain," *ACM Transactions on Graphics*, vol. 34, no. 1, pp. 1–13, 2014.

[17] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, vol. 35, no. 6, 2016.

[18] Oliver Fleischmann and Reinhard Koch, "Lens-based depth estimation for multi-focus plenoptic cameras," *Pattern Recognition*, pp. 410–420, 2014.

[19] Sven Wanner and Bastian Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Transactions of Pattern analysis and machine intelligence*, vol. 36, no. 3, 2013.

[20] Michael W. Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," Dec. 2013.

[21] Ting-Chun Wang, Alexei Efros, and Ravi Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras.," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[22] Neus Sabater, Mozhdeh Seifi, Valter Drazic, Gustavo Sandri, and Patrick Perez, "Accurate disparity estimation for plenoptic images," in *ECCV workshop on Light Fields for Computer Vision*. IEEE, 2014.

[23] 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 International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, "The lumigraph.," *23rd annual conference on Computer graphics and interactive techniques, ACM*, pp. 43–54, 1996.

[25] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013.

[26] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid, "Deepmatching: Hierarchical deformable dense matching," *International Journal of Computer Vision (IJCV)*, 2016.

[27] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," Tech. Rep., University of Illinois at Urbana-Champaign, 2009.

[28] "Lytro power tools beta," `https://www.lytro.com/imaging/power-tools`, Accessed: 2017-01-10.