An Image Rendering Pipeline for Focused Plenoptic Cameras

Matthieu Hog, Neus Sabater, Benoît Vandame, Valter Drazic

An Image Rendering Pipeline for Focused Plenoptic Cameras

Matthieu Hog, Neus Sabater, Benoît Vandame, Valter Drazic

***Abstract*—In this paper, we present a complete processing pipeline for focused plenoptic cameras. In particular, we propose (i) a new algorithm for microlens center calibration fully in the Fourier domain, (ii) a novel algorithm for depth map computation using a *stereo focal stack* and (iii) a depth-based rendering algorithm that is able to refocus at a particular depth or to create all-in-focus images. The proposed algorithms are fast, accurate and do not need to generate Subaperture Images (SAIs) or Epipolar Plane Images (EPIs) which is capital for focused plenoptic cameras. Also, the resolution of the resulting depth map is the same as the rendered image. We show results of our pipeline on the Georgiev's dataset and real images captured with different Raytrix cameras.**

***Index Terms*—Plenoptic cameras, calibration, depth estimation, refocusing**

I. Introduction

UNLIKE conventional cameras, plenoptic cameras are able to capture the Light Field (LF), thanks to a microlens array placed between the main lens and the sensor. Depending on the position of this microlens array, plenoptic cameras are divided into type 1 or *basic* cameras as presented in [1] (e.g. Lytro [2]) and type 2 or *focused* cameras as presented in [3] (e.g. Raytrix [4]). In this work we focus on the second type.

The task of processing the microlens images captured by a plenoptic camera in order to generate an image (as with a conventional camera) is called *rendering*. During this process, it is possible to modify the virtual focus plane (digital refocusing [1], [5]) or render images with a greater depth of field compared to the main lens optics capabilities [6]. Besides, a LF contains both angular and spatial information of the incoming light rays, allowing to estimate the depth of the captured scene, which is one major benefit of the LF technology.

In this paper we will address the plenoptic depth estimation problem and we will propose a rendering method that relies on depth in order to improve the image quality. The main asset of our approach is that SAIs or EPIs are not computed neither for depth estimation nor rendering, avoiding aliasing artifacts as pointed out by [7]. Instead, captured data in the sensor plane is projected into the rendering planes directly. This strategy is of particular interest for type 2 plenoptic cameras for which SAIs are not as simple to estimate [8] as in type 1.

In addition to depth estimation and rendering, we have also tackled the calibration problem. Indeed, knowing the microlens images centers is an essential step for any plenoptic image processing. We have designed a calibration algorithm, entirely in the Fourier domain, that has proved to be fast, insensitive to noise and robust to different microlens array configurations. Furthermore, our experiments tend to show that the calibration could be used on natural images as well.

To sum up, our contributions are (see Fig. 1 for an overview of our pipeline):

1) A calibration algorithm in the Fourier domain for fast and accurate microlens images center estimation.
2) A novel depth estimation algorithm operating in the refocused image domain that exploits the relationship between the focus planes and the disparities on each slice of what we define as a *stereo focal stack*.
3) A depth-based rendering algorithm that is able to produce accurate results for image refocusing and all-in-focus imaging.

II. Related Work

Regarding plenoptic camera calibration several solutions have already been explored either in the spatial domain [9], or using a combination of Fourier and spatial analysis [10], [11]. However our algorithm estimates all parameters on the frequency domain which has the advantage of being fast and accurate.

In the literature, plenoptic depth estimation has aroused great interest and many research works have been published. We classify them in four different approaches depending on the image type they consider as input: SAIs, microlens images, EPIs or refocused images.

First, SAI-based depth estimation methods rely on the fact that computed SAIs from plenoptic cameras are well rectified images with constant baseline. Among these techniques, we can find local block-matching [12]–[14] and global matching methods [15]–[17]. Then, microlens-based depth estimation methods consider each microlens image as separated camera images with a very small baseline. For this type of methods local and global approaches are also adapted to the plenoptic framework. In [18]–[20] a block-matching algorithm for microlens images is used, and [21]–[23] formalize the problem as an energy minimization task in which cost volumes are computed for each microlens. Another type of method for plenoptic depth-estimation uses EPIs [24]–[28]. Indeed, the slope of the line composed of corresponding pixel in an EPI is proportional to the depth of the pixel [29].

Finally, other approaches use refocused images or images in a focal stack to perform depth computation [30], [31]. However, when defocus cues are used they are usually combined with other measures [15], [26] because of their poor accuracy.

Among plenoptic depth estimation methods we would like to highlight some recent approaches that explicitly estimate

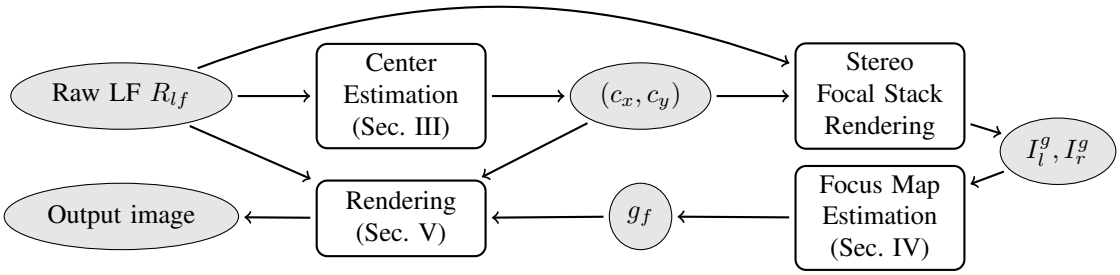


Fig. 1: Proposed pipeline to process a raw LF captured with a focused plenoptic camera. Microlens centers are first estimated. Then a *stereo focal stack* is computed projecting directly the data from the sensor. The *stereo focal stack* allows to estimate a focus map, which is used in the final rendering step.

occlusions and seem to give the best results. Occlusions can be detected by studying the variance of the pixel re-projections on several views as in [32] or an occlusion coefficient can be used in a regularisation framework. For instance, [33] statistically computes the probability of a pixel to be in an occlusion boundary, [34] uses the log likelihood of the probability of the pixel color to appear in the projected views and [35] compare depth and variance of occluding candidates normalized by the region mean (to handle uniform areas). It is also possible to learn occlusion and depth simultaneously as in [36].

While plenty of contributions on plenoptic depth estimation significantly improving the state-of-the-art have been published the last years, very rare are the papers addressing the problem for type 2 plenoptic setup. This may be due to the fact that most of the available data come from Lytro (type 1) cameras. Moreover, the proposed approaches almost all rely on re-sampling the captured LF to a traditional SAI-based representation. Yet, SAIs or EPIs are not well adapted to focused plenoptic cameras. Indeed, it has been proved [7] that SAIs suffer from strong aliasing artifacts (even with antialiasing filters) which affects depth estimation, but above all, estimating SAIs or EPIs for focused plenoptic cameras is a complex process not without error [8] (see Fig. 2) since an accurate depth estimate of the scene would be necessary (chicken-egg problem).

Microlens-based methods [18] can be a good alternative for focused plenoptic cameras provided the size of the microlens are big enough as it happens to be with the Georgiev's prototype, but this is not always the case.

In this context, our motivation is to provide a depth estimation method that works with arbitrarily sized microlenses and operates in the refocusing image domain without the need of SAIs or EPIs. This approach has the advantage of creating depth maps of the resolution of the final rendered image and that are not affected by SAI rendering artifacts.

Concerning plenoptic image rendering, many algorithms have been proposed since the pioneer work of [1] using the Fourier Slice Theorem. On the one hand, there are approaches using SAIs (shift and add) [11], [37] and on the other hand, the approaches using the projection rendering algorithm [10], [38] which are the closer works to ours.

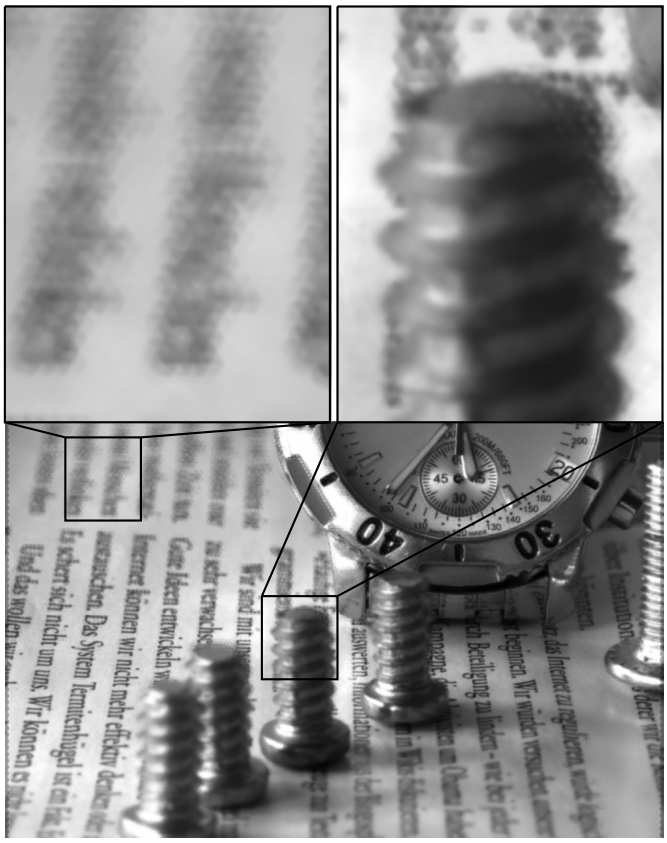


Fig. 2: SAI from a focused plenoptic camera estimated with the approach in [8]. Note the microlens artifacts on the two zoom-ins that may be detrimental to depth estimation (please see on the electronic version). Unlike type 1, type 2 plenoptic cameras require the depth to be know to render SAIs without errors. Our approach circumvents this problem estimating depth without SAIs.

III. Plenoptic Image Calibration

In this work, plenoptic calibration refers to estimating the microlens image centers (see [39] for a complete plenoptic camera calibration method). More precisely, in our calibration, we compute the microlens image diameter D, the translation offset $\mathbf{o} = (o_x, o_y)$ and the rotation α with respect to the coordinate system given by the sensor array (see Fig. 3).

So, the microlens image center coordinates (c_x, c_y) in the pixel coordinates are computed as:

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} o_x \\ o_y \end{bmatrix} + \frac{D}{\sqrt{3}} \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{1}$$

where $(x, y) \in \mathbb{Z}^2$ are the elements of an integer grid.

Our approach leans on the observation that a white plenoptic image I_w can be modeled as a sum of three 2D cosines (see Fig. 4), oscillating at different angles:

$$I_w(x,y) = \frac{1}{3} \sum_{d=0}^{2} \cos\left(\frac{2\pi}{D} c^d(x,y)\right), \tag{2}$$

$$c^d(x,y) = \cos\left(\frac{d\pi}{3} + \alpha\right)(x - o_x) + \sin\left(\frac{d\pi}{3} + \alpha\right)(y - o_y).$$

Consequently, its Fourier transform $F(I_w)$ is a Dirac comb function (Fig. 5). In this work we propose to estimate all calibration parameters α, D and $\mathbf{o}$ directly from $F(I_w)$. Let F^m and F^p be the magnitude and phase of $F(I_w)$.

Let $\xi_i^0 \in \mathbb{Z}^2$ be the pixel coordinates of the i-th peak (local maxima) of F^m obtained by thresholding. In practice our threshold is fixed to $100 \cdot$Variance(F^m). Note however, that the peaks of the Dirac comb in the frequency domain need to be evaluated with great accuracy (much below the pixel size) in order to obtain precise microlens image centers. Inspired by [40], the final peak locations $\xi_i = \xi_i^0 + \Delta\xi_i^0 \in \mathbb{R}^2$ are estimated with sub-pixel accuracy.

More precisely, each component of $\Delta\xi^0 = (\Delta\xi_u^0, \Delta\xi_v^0)$ is computed as

$$\begin{aligned} &\text{sign}\big(F^m(\xi^-) - F^m(\xi^+)\big) \, \frac{M}{M + F^m(\xi^0)}, \\ &\text{where } M = \max\big\{F^m(\xi^+), F^m(\xi^-)\big\}, \end{aligned} \tag{3}$$

and $\xi^+ = (\xi_u^0 + 1, \xi_v^0)$, $\xi^- = (\xi_u^0 - 1, \xi_v^0)$ when estimating $\Delta\xi_u^0$ and $\xi^+ = (\xi_u^0, \xi_v^0 + 1)$, $\xi^- = (\xi_u^0, \xi_v^0 - 1)$ when estimating $\Delta\xi_v^0$.

The number of peaks, card(i), is equal to six in the ideal case of the white image being a sum of pure cosines (Eq. 2) but many replicas appear on real images. In that case, we select the six peak locations with most energy and at the same distance from the center (Fig 5).

The microlens images diameter and the rotation are then computed as

$$D = \frac{N}{\frac{1}{6}\sum_{i=1}^{6} \rho_i}, \tag{4}$$

$$\alpha = \frac{1}{6} \sum_{i=1}^{6} \mod\left(\theta_i, \frac{\pi}{3}\right), \tag{5}$$

where θ_i and ρ_i are the polar coordinates of ξ_i, N is the size of the white input image and mod is the modulo function.

Finally, the lines along which the three cosines oscillate (color lines in Fig. 4), intersect at the offset phase $\mathbf{o}$. Considering only three peaks among the six not being symmetric, we define the oscillation lines as

$$L_{i=1,2,3}(\mathbf{x}) := x \sin\left(\theta_i + \frac{\pi}{2}\right) - y \cos\left(\theta_i + \frac{\pi}{2}\right) + \frac{F^p(\xi_i)}{2\pi} = 0, \tag{6}$$

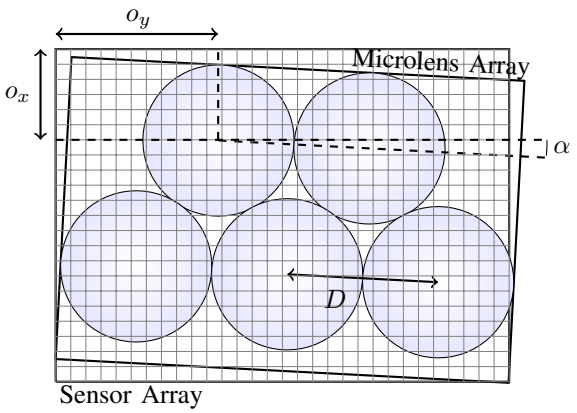


Fig. 3: Microlens images (of diameter D) are arranged in a hexagonal grid and pixels in a squared grid. Microlens images are misaligned with respect to the pixel grid. There is a rotation of angle α and a translation offset (o_x, o_y) between the origins of both grids placed at the most top-left pixel and microlens image respectively.

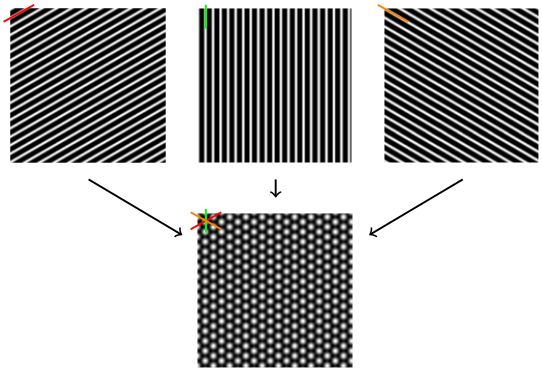


Fig. 4: An ideal white plenoptic image is a sum of three 2D cosine images. The intersection of the lines along which the 3 cosines oscillate defines the offset $\mathbf{o}$ of the white image.

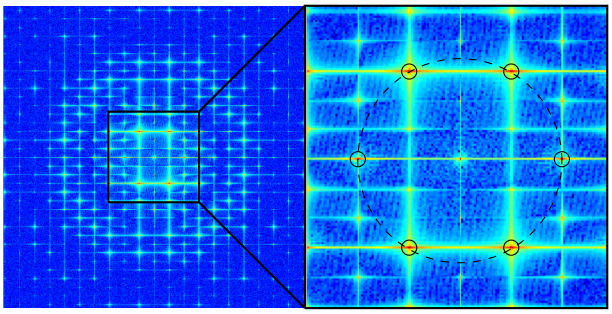


Fig. 5: Peaks in the (log) frequency spectrum of a real white image. Each peak has its symmetrical. The Fourier transform of an ideal white image is a perfect Dirac comb but many replicas appear on real white images. The six concentric peaks with highest energy are selected.

where $\mathbf{x} = (x, y)$. We write the three line equations as $A\mathbf{x} = B$ and its solution $\mathbf{o} = (A^T A)^{-1} A^T B$ is estimated by least squares.

In order to validate our method, we have tested our algorithm with synthetic white plenoptic images (generated using Eq. 2). We have verified its robustness with many parameter values for $\mathbf{o}$, α and D simulating different plenoptic camera configurations. We have also added Gaussian noise of variance V and simulated different camera apertures by Gamma-correcting the white image by a factor Γ in order to study the performance of our algorithm in more complicated situations. For the sake of comparison, we have also estimated in all of our tested synthetic images the microlens images centers with the method of [9], a full spatial calibration method. Complete results can be found on Tab. I. In average, both methods are comparable but we have observed a larger robustness in our solution (smaller variance) when changing the parameters. Also the algorithm in [9] has failed in two cases.

TABLE I: Comparison of the calibration error between our approach and the one proposed in [9], across several parameter variations.

Parameters					Mean Error	
D	α	o_x, o_y	Γ	$V_{(10^{-2})}$	Ours	[9]
10.1	0.03	-5 0	1.7	0.01	0.0821	**0.0365**
10.2	//	//	//	//	**0.1357**	0.3741
10.5	//	//	//	//	**0.3139**	0.3150
10.8	//	//	//	//	0.1886	**0.0209**
10	0.01	-5 0	1.7	0.01	0.0887	**0.0817**
//	0.04	//	//	//	0.0923	**0.0355**
//	0.05	//	//	//	**0.0929**	0.3288
//	0.1	//	//	//	0.0937	**0.0349**
10	0.03	-5.1 0	1.7	0.01	0.0911	**0.0280**
//	//	-5.5 0	//	//	**0.0982**	0.3165
//	//	-5 0.1	//	//	0.0905	**0.0379**
//	//	-5 0.5	//	//	**0.1168**	0.3489
10	0.03	-5 0	2	0.01	0.0882	**0.0590**
//	//	//	1.5	//	**0.0916**	0.3580
//	//	//	1	//	**0.0954**	0.4166
//	//	//	0.7	//	**0.0972**	Fail
10	0.03	-5 0	1.7	0.015	0.0900	**0.0410**
//	//	//	//	0.03	0.0893	**0.0776**
//	//	//	//	1	**0.0905**	0.3749
//	//	//	//	10	**0.0904**	Fail

IV. Proposed Depth Estimation Method

In this section we present our method for depth estimation. Unlike other methods we compute a *focus map* which gives the in-focus value of each pixel without using defocus cues but stereo matching. The novelty of our method is that we first compute a so-called *stereo focal stack*, then stereo matching is performed for each of the pairs of images of the stereo focal stack and, finally, the obtained disparities are combined to obtain a focus map.

A. Stereo Focal Stack Computation

A focal stack is a collection of photographs focused at different depths. In order to render each image (slice) of the focal stack I^g, refocused at the focal value g, we use the projection algorithm as in [10], [38]. This is, each pixel (x, y) of the raw LF R_{lf} belonging to the microlens with center coordinates (c_x, c_y), is projected at position

$$(X, Y) = \Big(s(g(x - c_x) + c_x),\ s(g(y - c_y) + c_y)\Big)\,, \quad (7)$$

where s controls the size of the rendered image. Formally, the refocused image is computed as

$$I^g(m,n) = \frac{1}{W(m,n)} \sum_{x,y} K(X - m, Y - n) R_{lf}\,(x, y)\ ,$$
$$\text{where } W(m,n) = \sum_{x,y} K(X - m, Y - n)\,,$$
$$\text{and } K(u,v) = \begin{cases} \frac{1}{u^2+v^2}\,, & \text{if } ||(u,v)|| < 0.5 \\ 0\,, & \text{otherwise.} \end{cases} \quad (8)$$

K being a fixed kernel with a very small support (4 closest pixels from the projected coordinates).

Now, a *stereo focal stack* is rendered using Eq. 7 and Eq. 8 but separately for pixels (x, y) belonging to the left part and the right part of the microlens images (see Fig. 6). This strategy creates a stereo pair of images I^g_l and I^g_r for each focus value g.

Note that the stereo pairs suffer from aliased blur and may contain missing channel values. Indeed, given a fixed size s for the rendered images the spatial resolution of the projected points (X, Y) on the refocus plane depends on the depth of the scene as pointed out in [38]. Inspired by this same work, demosaicking is done during the rendering step and we project with Eq. 7 the color channels separately.

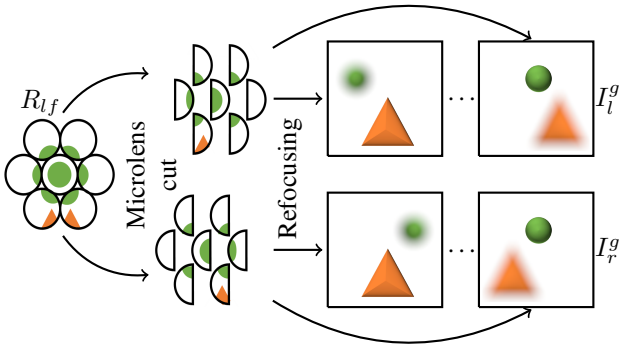


Fig. 6: Stereo focal stack computation. Points in the raw LF are projected separately depending on their positions on the microlens. Points belonging to the left (resp. right) side of the microlens are projected into I^g_l (resp. I^g_r). For each g, I^g_l and I^g_r is a rectified pair of stereo images such that points at the focus plane g appear sharp.

B. Focus Map Estimation from the Stereo Focal Stack

Proposition 1. *Let g_f be the g value for which a certain point on the scene is in-focus. Then, for any focus value g, the difference between g and g_f is proportional to the disparity Δ^g of this point in the stereo pair of images I^g_l and I^g_r. Also, a point appears in-focus in the refocused images I^g_l and I^g_r (i.e. $g = g_f$) if and only if its corresponding disparity is null ($\Delta^g = 0$).*

Proof. Let us consider a point in the scene that is seen by two microlenses (Fig. 7). The same reasoning is valid for more microlenses but we consider only two for the sake of clarity. Let x_1 and x_2 be the x-coordinates in R_{lf} of this point and δ the distance between them. Using Eq. 7 we know that for each g, the disparity Δ^g of the corresponding points in I_l^g and I_r^g is

$$\begin{aligned}\Delta^g &= X_2 - X_1 = s(g(x_2 - c_2) + c_2) - s(g(x_1 - c_1) + c_1) \\ &= s(g(\delta - D) + D)\,.\end{aligned} \tag{9}$$

Now $\Delta^g = 0$ if and only if

$$g = \frac{D}{D - \delta}\,, \tag{10}$$

which turns out to be the the value g_f that allows a point to be in-focus (i.e. image points of a same scene point are projected at the same position).

From Eq. 9 and 10 we get the relationship between the refocusing parameter g used for rendering and g_f the value at which the point is in-focus

$$g = \frac{1}{s(\delta - D)}\Delta^g + g_f\,. \tag{11}$$

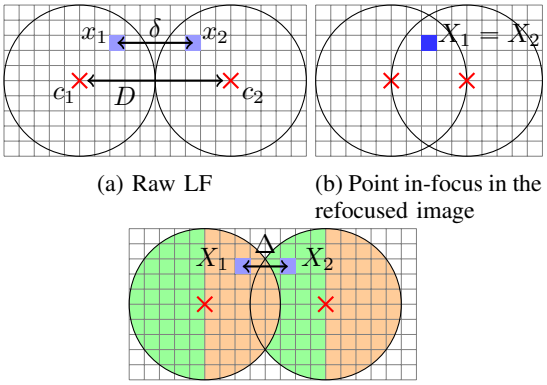
□

From the previous proposition we know that there is a linear relationship between g and Δ^g and estimating the focus g_f is equivalent to estimating the value g such that $\Delta^g = 0$. In practice, g_f is estimated as the root of the line passing through two points (g_1, Δ_1^g) and (g_2, Δ_2^g) for two particular focus values g_1 and g_2. Precisely,

$$g_f = \Delta_2^g - g_2 \frac{g_1 - g_2}{\Delta_1^g - \Delta_2^g}\,. \tag{12}$$

In order to compute Eq. 12, it is sufficient to render two pairs of stereo images at g_1 and g_2 and to estimate the corresponding disparities Δ^{g_1} and Δ^{g_2}. Notice, however, that it is possible to estimate the corresponding disparity Δ^g for each slice of the focal stack. In that case g_f is the root of the regression line of all (g, Δ^g). This solution produces slightly more accurate results at the expense of greatly increasing the computational cost. This is why in our algorithm the focus map at each point is estimated using Eq. 12 which is a good trade-off between accuracy and complexity.

Note that, our algorithm does not compute SAIs but projects the information into the refocusing plane. Also, our focus map has the same size than the rendered images (same s value). Besides, it is interesting to point out that since the projection is done in a stereo focal stack it creates a parallax at each slice g. Thanks to this parallax, any binocular stereo algorithm can be exploited for evaluating Δ^g. Depending on the desired accuracy and complexity one algorithm or another can be used. In this work we have used the algorithm presented in [41] because it is real-time and accurate.



(a) Raw LF (b) Point in-focus in the refocused image

(c) Point out-of-focus in the refocused image. The left (green) and right (orange) parts of the microlenses form the left and right slices of the focal stack respectively.

Fig. 7: Projection of a scene point visible on two microlens images. Both points are projected at the same position ($\Delta^g = 0$) when the point is in-focus ($g = g_f$) and there is a shift $\Delta^g \neq 0$ when the point is not in-focus.

V. Rendering using a Focus Map

A. Adaptive Splatting for Refocusing

Inspired by [7], [10] we define a *splatting kernel* K' to be used instead of K in Eq. 8. K' adaptively changes for each point of the scene. In particular, we exploit the focus map obtained previously to define a Gaussian splatting kernel

$$K'(u, v) = \exp\Big(- \frac{\sqrt{(X - u)^2 + (Y - v)^2}}{\lambda\,|g_f(u, v) - g| + \varepsilon} \Big)\,, \tag{13}$$

where ε is a very small value to avoid dividing by zero and λ controls the ratio among the spatial distance to (X, Y) and the g focus difference. The Gaussian kernel K' aims to penalize distant points from (X, Y) while its standard deviation results from the difference in absolute value between the refocusing value g and the in-focus value of the point $g_f(u, v)$.

The idea behind the weighting is that the kernel K' has a small support when the point is in-focus (i.e. $g = g_f(u, v)$). On the contrary, the farther g is from g_f, the bigger the support of K' which increases blurriness at that particular point. Besides, the splatting strategy also helps to densify the rendered image. Indeed, we know that in particular cases several values of (x, y) are projected to the same point (X, Y) or different values of (X, Y) but very close from each other creating areas with few, or no pixel contributions [38].

One problem that rises, when using splatting is the spreading of background out-of-focus pixels intensities on foreground pixels, creating unwanted artifacts around edges of foreground objects. To overcome this issue we use a bilateral filtering strategy. We alter the kernel in such way that background pixel values are not propagated on the kernel area where the depth is inferior to the depth of the splatted pixel.

The depth reshaped kernel is defined as $K'' = K' \circ S$ where

$$S(u, v) = \begin{cases} 1\,, & \text{if } g_f([X], [Y]) > g_f(u, v), \\ 0\,, & \text{otherwise.} \end{cases} \tag{14}$$

Thus, S is not null when the point (u, v) is behind $([X], [Y])$. The refocusing is performed as in Eq. 8, replacing K with K''.

B. Gathering for All-in-Focus Rendering

Splatting can also be seen backwards. Instead of spreading the ray values around the splatting coordinates, it is possible for a pixel (m, n) in the refocused image, knowing g_f, to compute the corresponding set of coordinates (x, y) of pixels in the raw LF that *see* (m, n):

$$(x, y) = \left(\frac{\frac{u}{s} - c_x}{g_f(u, v)} + c_x \quad , \quad \frac{\frac{v}{s} - c_y}{g_f(u, v)} + c_y \right). \tag{15}$$

We call this approach gathering, in the sense that we seek to fetch and integrate ray-pixels from the raw LF rather than projecting them into a refocused image. Doing so, we integrate only the pixel describing the same scene point, creating an image that is sharp everywhere. This is similar to the approach proposed in [19], but in our case, the depth information is contained directly in the refocused image domain, not in the raw LF domain. Formally, the all in-focus image is computed as

$$I(m, n) = \frac{1}{W(m, n)} \sum_{c_x, c_y} K_{c_x, c_y}(m, n) R_{lf}([x], [y])$$
$$\text{where } W(m, n) = \sum_{c_x, c_y} K_{c_x, c_y}(m, n) \text{ and}$$
$$K_{c_x, c_y}(m, n) = \begin{cases} \frac{1}{\{x\}^2 + \{y\}^2}, & \text{if } ||(x, y) - (c_x, c_y)|| < \frac{D}{2}, \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

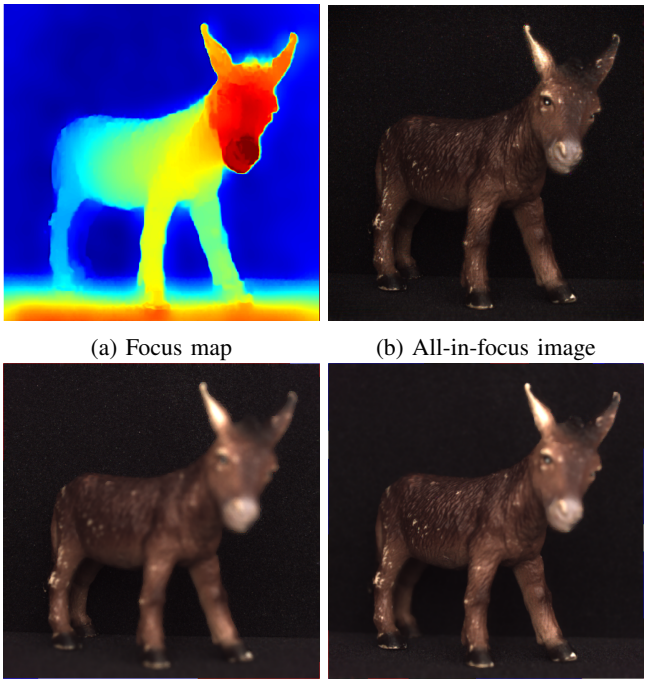
where $\{a\}$ is the decimal part of a and K checks if the back-projected pixel is visible on a microlens image of center (c_x, c_y) (i.e. it is null if the back-projected pixel coordinates (x, y) is outside of the microlens image) and otherwise, weights the pixel contribution according to its distance from the (non-integer) back-projected image coordinates. Carrying the microlens image visibility test for all microlens images can be extremely heavy. However, a pixel (m, n) can only be seen within a small radius around $(m/s, n/s)$ in the raw LF. That is why, in practice, the search for the microlens images can be bounded to few microlenses.

Note that depending on its depth, one pixel may receive incomplete color channel information. In that case, we interpolate with the neighbourhood pixels in the raw LF.

VI. Experimental results

In this section we show the results of our depth estimation and rendering algorithms. We show experiments on Raytrix R5 data we have captured, on a Raytrix R11 dataset[1] and on the Georgiev's dataset[2] for comparison purposes. As far as we know these are the only available type 2 plenoptic datasets providing raw data which is the input of our pipeline.

[1]Available online : https://www.raytrix.de
[2]Available online : http://www.tgeorgiev.net



(a) Focus map (b) All-in-focus image

(c) Image refocused at $g = 2.6$ (d) Image refocused at $g = 4$

Fig. 8: *Donkey* experiment. Data captured with a Raytrix R5 camera. We invite the reader to zoom-in to see details. Supplementary material includes more results of our R5 dataset.

In our experiments we divide the raw LF by its corresponding white image to correct vignetting, and we fix $s = 0.5$. Our focus map is neither filtered nor regularized.

Fig. 8a shows the focus map of one of the images of the Raytrix R5 dataset. The two focus slices are rendered at $g = 2.5$ and $g = 5.5$. The corresponding all-in-focus image in Fig. 8b is entirely sharp, attesting on the validity of the focus map. Fig. 8c and Fig. 8d compare two images refocused using adaptive splatting, refocused on the background and the donkey flank respectively. We notice that adapting the kernel allows to recover the details of the objects in-focus while showing a uniform blur in the out-of-focus areas. Supplementary material includes the raw data and calibration, as well as more examples of depth estimation and rendering of our R5 dataset.

Fig. 9 compares the obtained focus maps with the manufacturer depth map for a Raytrix R11 camera. We notice that the errors introduced by our method are fairly different than Raytrix's. Indeed our algorithm performs better in uniform areas (e.g. region between the arm and head of the pilot). However our algorithm is more sensitive to the reflexion halos (e.g. specularities into the ship's wheel). Also, our algorithm generally has a better edge preservation while Raytrix depth maps suffer from *fattening* (e.g. branches of the forest). Fig.11 compares an all-in-focus image provided by Raytrix with ours. In general, our refocused images are comparable to Raytrix quality.

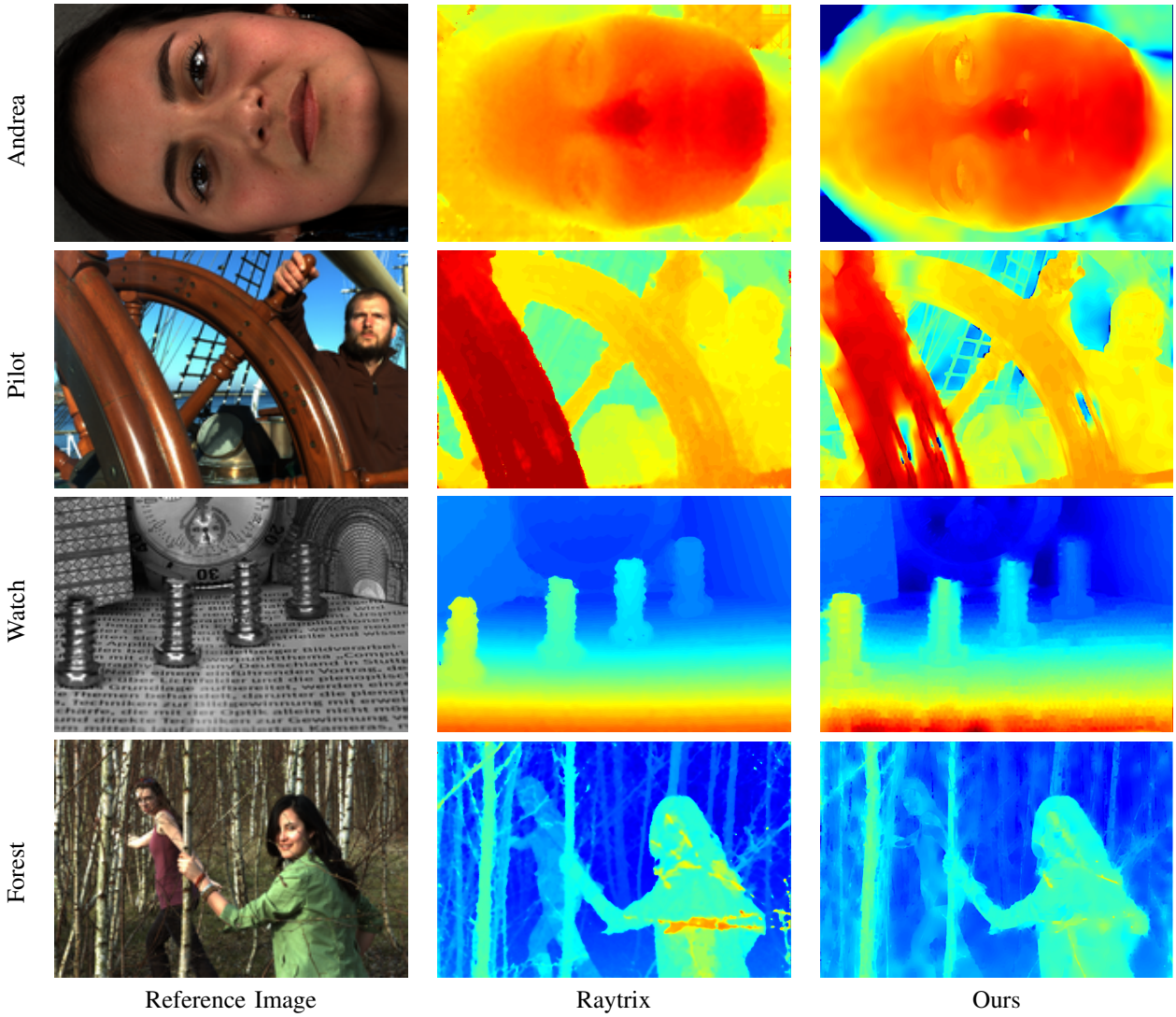


Fig. 9: Comparison with R11 test images provided by Raytrix. In general, our method deals better with objects borders and poor textured region but it provides erroneous disparities in specular regions.

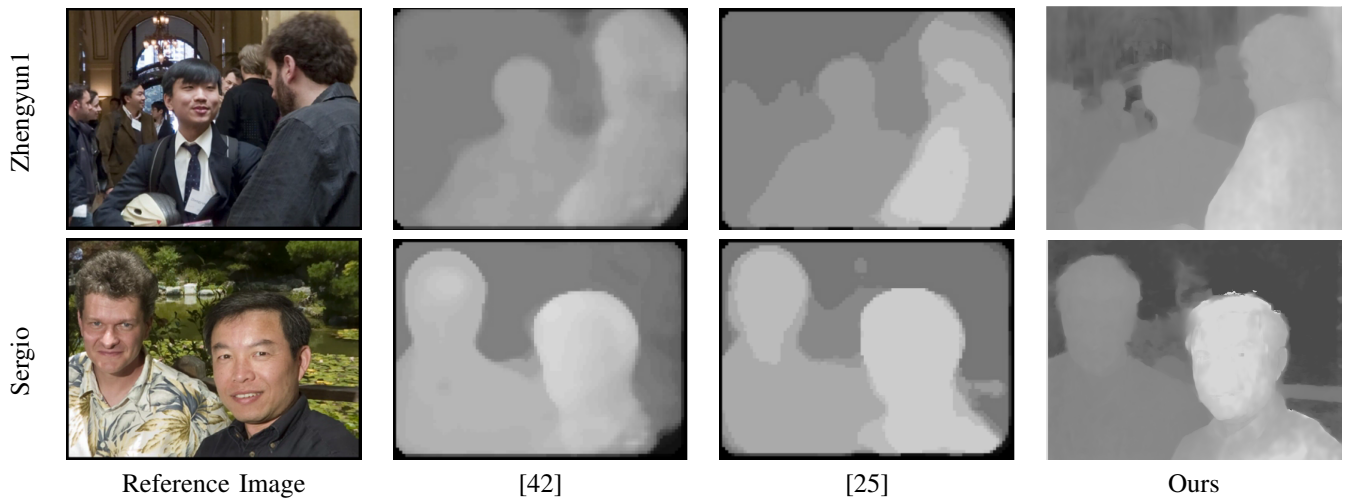


Fig. 10: Comparison with [42] and [25] using the Georgiev's dataset. Our approach is able to discern more depths and is more accurate. See the level of detail in the backgournd of 'Zhengyun1' or the faces in 'Sergio'.

On Fig. 10, we compare our depth maps with the results in [25], [42]. We can see that our method allows to recover more depth planes than the two microlens-bases approaches [25], [42]. This is due to the fact that our depth measurement is done on the image domain, with a bigger baseline than in the microlens domain. The depth maps for the rest of the Georgiev's dataset are available in the supplementary material.

Regarding the refocusing, Fig. 12 shows how adaptive splatting compensates for angular aliasing and the sparse image sampling (pixels with no contribution are visible as 0 channel values) that arises when using a fixed splatting (Fig. 12a). The images on Fig. 12b to 12d show the effect of changing the blur parameter λ.

Discussion

Our microlens centers calibration algorithm has been tested with real white images and natural images captured with a Raytrix camera in addition to our simulated white images. We have observed that both estimations provide very close microlens image center positions for the great majority of the scenes we captured. Besides, our calibration method is fast (less than 0.2 seconds for Raytrix R5 images in our Matlab implementation) and could be used to monitor the microlens image center positions dynamically on a plenoptic video. These two features are of major interest if the plenoptic camera has a zoom lens or interchangeable lenses. In that situation, calibration could be done "on the fly" from the captured sequence which is not possible with existing methods.

Usually, the resolution of the final depth map is substantially smaller than the size of the raw LF for most of the state-of-the-art methods. In fact, the resolution depends on the considered image type for depth estimation (SAIs, microlens images, EPIs or refocused images). For instance, the size of each Lytro SAI is 328×328 pixels which produces rather small depth maps (without super-resolution algorithms). In that sense, depth estimation on the refocused image plane provides the best resolution. In our approach, the resolution of the depth map and the rendered image are tuned with parameter s. The interesting point is that the depth map perfectly matches in terms of spatial resolution the rendered image which is a real advantage for depth-based editing tasks or in the rendering process itself as we have seen in Sec. V.

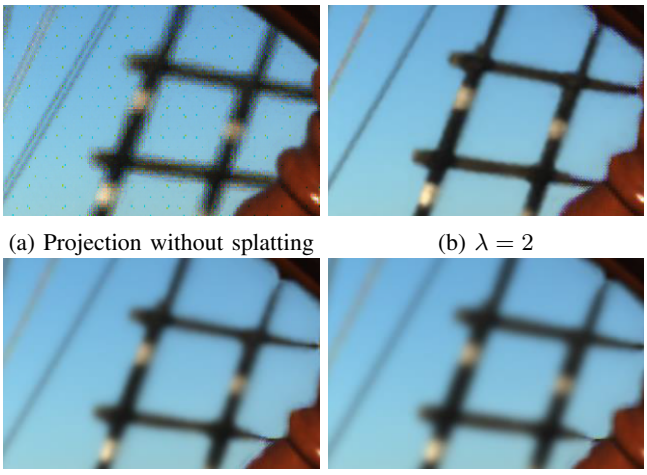
Even if the presented pipeline is particularly adapted for Raytrix cameras, we have not taken into account the tri-focal property of the microlens array. Taking it in consideration during the splatting process will surely improve the rendering image quality.

Note that different disparity estimation algorithms could be used in our framework [43]. However, the goal of this paper is not to compare such methods but to show that the depth estimation problem in a plenoptic camera can be treated as a stereo problem via a stereo focal stack and without estimating SAIs. In fact, our depth estimation strategy could also be applied to type 1 data but it would not be optimal. In that case, our half-apertures would not capture all the angular information as all the SAIs do in a type 1. We believe however that this is a good alternative for focused plenoptic cameras for which SAIs are not available without errors.

Moreover, it is interesting to point out that the proposed approach is somehow related to coded apertures [44]. Indeed, cutting the microlens images in half is physically equivalent to mask half of the aperture of a conventional camera. In particular, [45] compares the use of stereo aperture masks and depth from defocus using several mask and show that the

(a) Raytrix (b) Ours

Fig. 11: R11 all-in-focus rendering on test image Andrea. The fine details on the eyelashes are well recovered using our technique. Note that Raytrix uses a color and contrast correction and potentially a sharpening filter on their output images.



(a) Projection without splatting (b) $\lambda = 2$

(c) $\lambda = 3$ (d) $\lambda = 5$

Fig. 12: Refocusing via point projection without splatting using Eq. 8 (a) and adaptive splatting for different λ (b-d) on a region of the test image R11 *Pilot*. We can see that the adaptive strategy compensates for angular aliasing and that blur intensity can be controlled by the parameter λ.

second provides a better depth discrimination. As the study focused on the setup where the focus depth is *fixed*, it would be interesting to see how this conclusion hold in our case, since the scene depth is *triangulated* using several artificial focus depths.

Finally, regarding the complexity, we believe our algorithm is significantly lighter than other methods. Our Matlab implementation for generating the two slices of the focal stack runs in approximately 2.5 and 7 seconds per stereo slice for Raytrix R5 and Raytrix R11 images respectively. We believe that a proper GPU implementation of our depth estimation and rendering can be done in real-time provided the used stereo algorithm is real-time.

VII. Conclusion

We have introduced a novel pipeline for processing focused plenoptic camera images. First we have presented a detailed description of our calibration algorithm that fully estimates all parameters in the Fourier domain allowing a fast and robust microlens images center estimation on white and natural images. Then, we have proposed a new algorithm for depth estimation from a stereo focal stack. Our algorithm does not require estimating SAIs or EPIs but can bring into play any stereo algorithm. Moreover, it provides a depth map in the refocused image domain, and does not require any knowledge about the camera parameters (except the microlens images centers, estimated at the beginning of our pipeline). Finally, our image rendering is guided by the estimated scene depth and allows to refocus the images or render all-in-focus images. We have tested our algorithm on images captured with a Raytrix camera but our modelling is not restricted to it and could be applied to other focused plenoptic cameras. Further work will include combining the defocus cues introduces by the stereo focal stack in order to improve the depth measurement, especially in the specular areas and for occlusions.

References

[1] R. Ng, "Digital light field photography," 2006.

[2] "https://www.lytro.com."

[3] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *Computational Photography (ICCP), 2009 IEEE International Conference on*. IEEE, 2009, pp. 1–8.

[4] "http://www.raytrix.de."

[5] T. G. Georgiev, A. Lumsdaine, and S. Goma, "High dynamic range image capture with plenoptic 2.0 camera," *Signal recovery and synthesis*, p. SWA7P, 2009.

[6] F. P. Nava and J. Lüke, "Simultaneous estimation of super-resolved depth and all-in-focus images from a plenoptic camera," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009*. IEEE, 2009, pp. 1–4.

[7] C.-K. Liang and R. Ramamoorthi, "A light transport framework for lenslet light field cameras," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 2, p. 16, 2015.

[8] S. Wanner, J. Fehr, and B. Jaehne, "Generating epi representations of 4d light fields with a single lens focused plenoptic camera," in *International Symposium on Visual Computing (ISVC, oral presentation)*, 2011.

[9] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1027–1034.

[10] J. Fiss, B. Curless, and R. Szeliski, "Refocusing plenoptic images using depth-adaptive splatting," in *Computational Photography (ICCP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–9.

[11] D. Cho, M. Lee, S. Kim, and Y.-W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3280–3287.

[12] E. H. Adelson and J. Y. A. Wang, "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 2, pp. 99–106, 1992.

[13] T. E. Bishop and P. Favaro, "Plenoptic depth estimation from multiple aliased views," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1622–1629.

[14] N. Sabater, M. Seifi, V. Drazic, G. Sandri, and P. Perez, "Accurate disparity estimation for plenoptic images," in *ECCV Workshop on Light Fields for Computer Vision*, 2014.

[15] M.-J. Kim, T.-H. Oh, and I. S. Kweon, "Cost-aware depth map estimation for lytro camera," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 36–40.

[16] S. Heber, R. Ranftl, and T. Pock, "Variational shape from light field," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer Berlin Heidelberg, 2013, pp. 66–79.

[17] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. S. 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] T. Georgiev and A. Lumsdaine, "Focused plenoptic camera and rendering," *Journal of Electronic Imaging*, vol. 19, no. 2, pp. 021 106–021 106, 2010.

[19] C. Perwass and L. Wietzke, "Single lens 3d-camera with extended depth-of-field," in *Proc. SPIE*, vol. 8291, 2012, p. 829108.

[20] C.-W. Chang, M.-R. Chen, P.-H. Hsu, and Y.-C. Lu, "A pixel-based depth estimation algorithm and its hardware implementation for 4-d light field data," in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 786–789.

[21] S. Tulyakov, T. H. Lee, and H. Han, "Quadratic formulation of disparity estimation problem for light-field camera," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013, pp. 2063–2067.

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

[23] M. Uliyar, G. Putraya, S. Ukil, S. Basavaraja, K. Govindarao, and M. Veldandi, "Pixel resolution plenoptic disparity using cost aggregation," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3847–3851.

[24] S. Wanner, C. Straehle, and B. Goldluecke, "Globally consistent multi-label assignment on the ray space of 4d light fields," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1011–1018.

[25] M. Uliyar, G. Putraya, and S. Basavaraja, "Fast epi based depth for plenoptic cameras," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013, pp. 1–4.

[26] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 673–680.

[27] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, "Depth from shading, defocus, and correspondence using light-field angular coherence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1940–1948.

[28] D. Dansereau and L. Bruton, "Gradient-based depth estimation from 4d light fields," in *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, vol. 3. IEEE, 2004, pp. III–549.

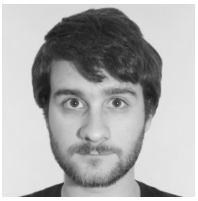
[29] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 7–55, 1987.

[30] Y.-H. Kao, C.-K. Liang, L.-W. Chang, and H. H. Chen, "Depth detection of light field," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1. IEEE, 2007, pp. I–893.

[31] A. Mousnier, E. Vural, and C. Guillemot, "Partial light field tomographic reconstruction from a fixed-camera focal stack," *arXiv preprint arXiv:1503.01903*, 2015.

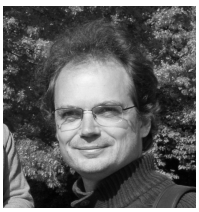
[32] C. Chen, H. Lin, Z. Yu, S. Bing Kang, and J. Yu, "Light field stereo matching using bilateral statistics of surface cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1518–1525.

[33] H. Lin, C. Chen, S. Bing Kang, and J. Yu, "Depth recovery from light field using focal stack symmetry," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3451–3459.
[34] W. Williem and I. Kyu Park, "Robust light field depth estimation for noisy scene with occlusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4396–4404.
[35] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3487–3495.
[36] S. Heber and T. Pock, "Convolutional networks for shape from light field," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3746–3754.
[37] M. Seifi, N. Sabater, V. Drazic, and P. Perez, "Disparity guided demosaicking of light field images," in *IEEE International Conference on Image Processing (ICIP)*, 2014.
[38] Z. Yu, J. Yu, A. Lumsdaine, and T. Georgiev, "An analysis of color demosaicing in plenoptic cameras," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 901–908.
[39] Y. Bok, H.-G. Jeon, and I. S. Kweon, "Geometric calibration of micro-lens-based light-field cameras using line features," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.
[40] B. G. Quinn, "Estimating frequency by interpolation using fourier coefficients," *Signal Processing, IEEE Transactions on*, vol. 42, no. 5, pp. 1264–1268, 1994.
[41] V. Drazic and N. Sabater, "A precise real-time stereo algorithm," in *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*. ACM, 2012, pp. 138–143.
[42] T. Georgiev and A. Lumsdaine, "Reducing plenoptic camera artifacts," in *Computer Graphics Forum*, vol. 29, no. 6. Wiley Online Library, 2010, pp. 1955–1968.
[43] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.
[44] A. Levin, R. Fergus, F. Durand, and W. Freeman, "Image and depth from a conventional camera with a coded aperture," in *TOG*, vol. 26, no. 3, 2007, p. 70.
[45] A. Levin, "Analyzing depth from coded aperture sets," in *ECCV*. Springer, 2010, pp. 214–227.



Matthieu Hog received the BSc in 2013 from the Université de Saint-Etienne, France and the MSc in 2015 from the Université de Saint-Etienne and the Gøjik University College, Norway. He is currently pursuing his PhD studies in INRIA (Institut National de Recherche en Informatique et en Automatique) and Technicolor Research & Innovation in Rennes, France. His research interests are image processing, computer vision and computational imaging.

Neus Sabater received the BSc degree in 2005 from the Universitat de Barcelona, Spain, and the MSc and PhD degrees in 2006 and 2009, respectively, in image processing from the Ecole Normale Suprieure de Cachan, France. She was a postdoctoral researcher at the California Institute of Technology before being appointed at Technicolor Research & Innovation in 2011 where she is currently a Senior Scientist. Her research interest include image processing, computer vision and computational photography.



Benoît Vandame received the MSc degree in 1997 from the Université de Saint-Etienne, France, and the PhD degrees in 2004 from the Université de Nice, dedicated to image processing applied to large datasets of astronomical images. He was a postdoctoral researcher at the European Southern Observatory (ESO München, Germany) before being appointed at the Canon Research Center in Rennes, France. In 2014 he joins the Technicolor Research & Innovation. His research interest include image processing, computer vision and computational photography.

Valter Drazic received the MSc degree in Optical Engineering from Telecom Physique, Strasbourg in 1988, a MSc degree in Physics from Université Louis Pasteur, Strasbourg in 1989 and a PhD degree from Karlsruhe Institute of Technology in 1993. He is currently Principal Scientist with Technicolor Research & Innovation. His main research activities spanned from illumination and imaging optics, displays, computer vision, acquisition technologies and computational imaging.