Real-Time Refocusing Using an FPGA-Based Standard Plenoptic Camera

Christopher Hahne, Andrew Lumsdaine, *Senior Member, IEEE*, Amar Aggoun, and Vladan Velisavljevic, *Senior Member, IEEE*

***Abstract*—Plenoptic cameras are receiving increased attention in scientific and commercial applications because they capture the entire structure of light in a scene, enabling optical transforms (such as focusing) to be applied computationally after the fact, rather than once and for all at the time a picture is taken. In many settings, real-time interactive performance is also desired, which in turn requires significant computational power due to the large amount of data required to represent a plenoptic image. Although GPUs have been shown to provide acceptable performance for real-time plenoptic rendering, their cost and power requirements make them prohibitive for embedded uses (such as in-camera). On the other hand, the computation to accomplish plenoptic rendering is well structured, suggesting the use of specialized hardware. Accordingly, this paper presents an array of switch-driven finite impulse response filters, implemented with FPGA to accomplish high-throughput spatial-domain rendering. The proposed architecture provides a power-efficient rendering hardware design suitable for full-video applications as required in broadcasting or cinematography. A benchmark assessment of the proposed hardware implementation shows that real-time performance can readily be achieved, with a one order of magnitude performance improvement over a GPU implementation and three orders of magnitude performance improvement over a general-purpose CPU implementation.**

***Index Terms*—.**

I. Introduction

OVER the last two decades, several studies have reported methods to computationally render varyingly focused images from a single lightfield photograph [1]–[8]. In addition to spatial information, lightfields contain directional information, acquired by capturing an array of two-dimensional (2-D) spatial images with either multiple conventional cameras [1], [9]–[11] or by attaching a micro lens array (MLA) to a single image recording device [2], [12], [13]. In science, lightfield cameras are also known as plenoptic cameras derived from the Latin and Greek roots meaning “full view” [13], [14]. For industrial applications, MLAs are preferred to simple pinholes or coded-aperture patterns due to improved light-gather capability and to multiaperture systems due to compact form-factor. A study carried out by Ng *et al.* [15] has found that the maximum directional information is recorded when placing the microlenses one focal length away from the image sensor. However, a follow-up study reinvestigated this and showed that it is possible to flexibly tradeoff directional and spatial resolution by shifting the MLA with respect to the sensor [4], [16]. In this paper, we refer to the former design as the standard plenoptic camera (SPC) and the latter as the focused plenoptic camera (FPC). While researchers have developed a number of approaches to plenoptic camera design [17], [18], the rendering (or focusing) process remains computationally intensive, posing a core challenge to the computer vision field.

One motivating industrial performance-sensitive application for plenoptic cameras is in cinematography, where the use of plenoptic source video can greatly enhance the flexibility and creativity in capture and production. For example, since the optical parameters are not irrevocably set at the time the video is captured, focus or depth of field can easily be adjusted in postproduction. Moreover, new creative effects can be applied, including nonphysical optical effects. Plenoptic video can also be used to create stereo pairs for three-dimensional (3-D) viewing—with the important advantage over stereo capture that different videos can be created for different devices, each having parallax suited for the particular device [19]. Finally, 2-D and 3-D production can use significantly different effects for directing the viewer’s attention (depth of field is not as useful in 3-D as 2-D, for example). With plenoptic source video, 2-D and 3-D can be rendered from the same source, with different creative effects for each. We note that Lytro, one of the earliest manufacturers of plenoptic cameras, has recently announced a video lightfield camera to the broadcast and cinematography market [20]. In any of these scenarios, high rendering performance is essential. For preview and for postproduction, rendering of each video frame must be accomplished at the video frame rate, regardless of the effects and adjustments being applied.

Manuscript received June 11, 2017; revised January 10, 2018; accepted March 1, 2018. This work was supported by the EU’s 7th Framework Programme under Grant EU-FP7 ICT-2010-248420. *(Corresponding author: Christopher Hahne.)*

C. Hahne is with the trinamiX GmbH (BASF), Ludwigshafen 67063, Germany (e-mail: info@christopherhahne.de).

A. Lumsdaine is with the Northwest Institute for Advanced Computing, University of Washington, Seattle, WA 98195 USA (e-mail: al75@uw.edu).

A. Aggoun is with the School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton WV1 1LZ, U.K. (e-mail: a.aggoun@wlv.ac.uk).

V. Velisavljevic is with the School of Computer Science and Technology, University of Bedfordshire, Luton LU1 3JU, U.K. (e-mail: vladan.velisavljevic@beds.ac.uk).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2018.2818644

An early attempt at high-performance rendering was based on the projection slice theorem, which rendered images with lower dimensional slices of the lightfield in the Fourier domain [3], [21]. This procedure is also known as Fourier slice photography (FSP). Although FSP has the potential to be efficient when rendering a large number of focused images from the same lightfield, there are significant overheads in this approach that limit its practical application. Real-time rendering in the spatial-domain has been achieved with graphical processing units (GPUs) [22], but the cost and power associated with GPUs make their use in embedded settings (for example) impractical. Accordingly, it is the goal of this study to devise and demonstrate a special-purpose hardware architecture that performs real-time rendering in the spatial-domain based on serially incoming video frames. We propose an array of semisystolic finite impulse response (FIR) filters designed for high data throughput. Moreover, we realize the rendering convolution kernel in FIR fashion by introducing switches to the filter distribution network. For power efficiency and configuration flexibility, the proposed design is implemented with a field programmable gate array (FPGA). As distinguished from previous studies, our hardware design accomplishes a computation time of less than 100 μs for a single refocused frame with 3201-by-3201 pixel resolution when running at 100-MHz pixel clock frequency. This outperforms earlier studies in the field, which we further demonstrate with benchmarks against a GPU and a CPU MATLAB implementation.

The organization of this paper is as follows. Section II presents recent developments in the field of FSP and SPC lightfield modeling to serve as a starting point for refocusing in spatial-domain. Section III imposes requirements on the filter module architecture and presents a solution based on switch-driven FIR filters. The proposed hardware design is examined in Section IV, using a hardware description language (HDL) for FPGAs (see supplementary material) and by benchmarks with an alternative GPU-based implementation. Conclusions and suggestions for further work are presented in Section V.

II. Related Work

A. Background

A lightfield can be retrieved by light rays intersecting two consecutively-placed 2-D planes of known relative position [9]. Intersections of a single ray at two 2-D planes yield four coordinates in total, thus making up a four-dimensional (4-D) light ray parametrization. Because of its simplicity, this conceptual model has gained popularity among scientists in the field of computer vision. A related one-plane parameterization based on position and angle can also be used [4], [16]. In the celebrated work by Ng *et al.* [3], a raw captured 4-D lightfield is transformed to the Fourier domain to achieve refocusing using the projection-slice theorem. Unfortunately, the process of taking Fourier transforms, interpolating for slicing, and then taking inverse transforms introduces significant computational overhead, making FSP unsuitable for real-time rendering. This assumption was confirmed by Mhabary *et al.* [21], who have worked to advance FSP by employing a fractional Fourier transform. However, the authors conclude that the integral projection operator in the spatial-domain is faster when computing only a single refocused image from a lightfield. The suitability of refocusing in the spatial-domain was further confirmed by Lumsdaine *et al.* who demonstrated real-time rendering performance using GPU hardware [22]. For these reasons, our approach in this paper is based on rendering in the spatial-domain.

The main concept of computation time improvements using FPGAs builds on the principle of parallelization and pipelining [23]. A pipeline comprises chained processor blocks fed with serialized data that are processed sequentially. Speed up is obtained by processing data chunks in one processor unit while subsequent data chunks are handled in preceding units. Hence, the benefit of pipelining is that serialized data chunks are processed at the same time while processor units perform different tasks. While data serialization limits a specific task to be computed with one single operation at a time, e.g., one pixel after another, parallelized data streams allow a computing system to perform at least two operations of the same type simultaneously. Parallelization can be thought of as duplicating processor pipelines, which requires synchronized parallel data streams as input signals. Letting the degree of parallelization be ι, the computation time in image processing may be minimized to $\mathcal{O}\left(K^2/\iota\right)$ if 2-D image dimensions consist of K samples each and provided that both computation systems run at the same clock frequency. Consequently, the one-dimensional (1-D) parallelization limit is reached where $\iota = L$ for image rows and $\iota = K$ for image columns, which is the ideal scenario in terms of parallelizing data processes.

Early work in the field of embedded plenoptic imaging was reported by Rodríguez-Ramos *et al.* [24], who employed an FPGA to process plenoptic data with the aim of analyzing wavefront measurements. Another interesting approach, reported by Wimalagunarathne *et al.* [25], proposed a design to render computationally focused photographs from a set of multiview images using infinite impulse response filters. Work on real-time rendering from FPC captures was presented in [22]. The first reported hardware design for performing real-time rendering from SPC captures was presented by Hahne *et al.* [6]. Shortly thereafter, Pérez *et al.* [7] published an article addressing the same topic. The authors demonstrated significant computation time improvements compared with run times based on a central processing unit (CPU) system that was programmed using an object-oriented language. A theoretical comparison of our method with that of Pérez *et al.* [7] is carried out at the end of Section III.

B. SPC Ray Model

Development of a computationally efficient refocusing algorithm requires knowledge about the ray geometrical properties in a plenoptic camera. To conceive a refocusing hardware architecture in spatial-domain, we employ a ray model reported by Hahne *et al.* [8], which is based on paraxial optics. The model is depicted in Fig. 1 and builds on the assumption that image sensor plane and MLA are separated by one focal length f_s such that the MLA is focused to infinity, which is in accordance with

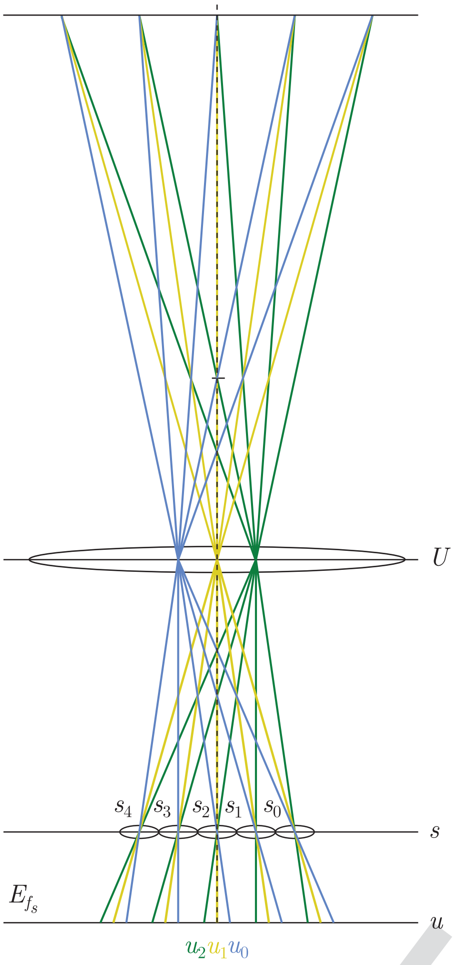

Fig. 1. SPC ray model (borrowed from [8]) with microlens chief rays traveling through the MLA plane s and main lens plane U, which is depicted as a thin lens. Lightfield intensities captured at the sensor plane are denoted as $E_{f_s}\,[s_j\,,\,u_{c+i}]$ for the 1-D case. Chief ray colors in a microimage indicate angular samples u_{c+i}.

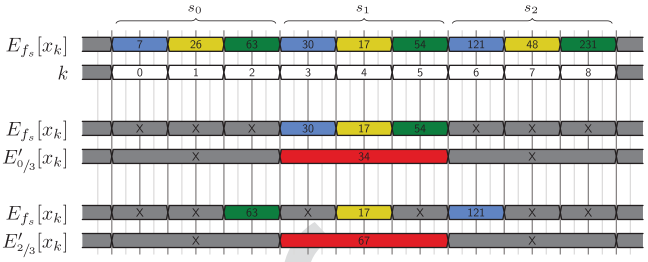

Fig. 2. Processing requirements for the hardware architecture. The diagram shows exemplary input illuminance values E_{f_s} (see Fig. 1) subdivided into microimages s_j and synthesized output values E'_a at a desired refocused image plane a.

Ng's concept of a plenoptic camera [15]. To understand lightfield imaging in an SPC, as in the Lytro setup [20], one may regard a main lens image of an object plane to be focused on the MLA plane. In this case, the focused light rays converge to the microlens and diverge when leaving it to form a microimage (see Fig. 1). A pixelated light-sensitive detector placed behind the MLA captures angular portions of the incident-divergent beam. Each angular sample in this microimage corresponds to the same focused spatial point in space observed from different views. This point's intensity is recovered when integrating all microimage samples.

We denote a lightfield captured by an SPC in the following way. For clarity, only the horizontal cross-section is regarded hereafter. In the angular domain u, we start counting samples from microimage centers (MICs), which serve as a reference positions $c = (M-1)/2$ where M denotes a consistent total number of samples for each microimage in one dimension. Microimages are seen to be radially symmetric and horizontally indexed by $c+i$, with $i \in [-c \mathinner{\ldotp\ldotp} c]$. Horizontal lightfield positions are then given as $[s_j\,,\,u_{c+i}]$ with j as the 1-D index of a respective micro lens s_j. All microimages together form a light field image with its cross-sectional representation $E_{f_s}\,[s_j\,,\,u_{c+i}]$ where E_{f_s} denotes a pixel's illuminance. As demonstrated in [8], a horizontal cross-section of a lightfield image can be refocused by employing

$$E'_a\left[s_j\right] = \sum_{i=-c}^{c} \frac{1}{M}\, E_{f_s}\left[s_{j+a(c-i)}\,,\,u_{c+i}\right], \quad a \in \mathbb{Q} \tag{1}$$

where a adjusts the synthetic focus. Equation (1) can also be applied to the vertical dimension.

Since images acquired by an SPC do not feature the $E_{f_s}\,[s_j\,,\,u_{c+i}]$ notation, it is convenient to define an index translation formula considering the lightfield photograph to be of two regular sensor dimensions $[x_k\,,\,y_l]$ as if taken by a conventional sensor. Indices are then converted by

$$k = j \times M + c + i \tag{2}$$

in the horizontal dimension meaning that $[x_k]$ is formed by $[x_{j\times M+c+i}]$ to replace $[s_j\,,\,u_{c+i}]$. This concept of index translation may be similarly extended to the vertical domain.

III. Filter Design

An efficient hardware design that enables an FPGA to refocus in real-time may be conceptualized on the basis of the lightfield ray model presented in Section II. The upper data line of Fig. 2 depicts discrete and quantized illuminance values $E_{f_s}\,[x_k]$ of a single horizontal row that is part of a calibrated lightfield image. Lightfield calibration implies MIC detection and rendering procedures to obtain a consistent microimage size (M). The computational refocusing synthesis given in Section II reveals that pixels involved in the integration process expose interleaved neighborhood relations, which exclusively depend on a. This phenomenon is illustrated by the data flow diagram in Fig. 2, where respective pixels are highlighted for two exemplary refocusing settings: $a = 0/3$ and $a = 2/3$. Here, each color corresponds to a chief ray in the model in Fig. 1, with $M = 3$ where yellow represents the MIC pixel. In this section, a hardware architecture is devised that accomplishes signal processing according to (1) as depicted in Fig. 2.

On the supposition that a horizontal cross-section of a captured lightfield $E_{f_s}\,[x_k]$ is a linear, time-invariant system, the integral projection in (1) may be represented as a discrete FIR convolution formula. Following the $[s_j, u_{c+i}]$ to $[x_k]$ translation in Section II, 1-D refocusing can be given by

$$E'_a\left[x_k\right] = \sum_{i=0}^{M-1} \frac{1}{M} E_{f_s}\left[x_{k'+i(aM-1)}\right], \quad a \in \mathbb{Z} \tag{3}$$

with

$$k' = (k+1) \times M - 1 \tag{4}$$

taking care of a correct integral projection, which inevitably reduces the number of samples in the rendered output image. Equation (3) aims at complying with the classical FIR filter notation, however with indices in subscripts for consistency reasons and to let x signify the domain and coordinate direction. Upon closer examination, one may notice that the impulse response is represented by a constant coefficient $1/M$, which is a consequence of weighting pixels equally during the integration process. Note that $i \in [0 \,..\, M-1]$ in the following.

In contrast to (3), we seek to reproduce an output image with a resolution numerically equal to that of the raw sensor image. To compensate for sample reduction in the integral projection process, the overall sensor resolution may be retained by upsampling the spatial-domain during image formation. Besides, it will be shown hereafter that our proposed upsampling scheme enables interpolation of refocused depth planes.

To break down the complexity, we devise one filtering function per refocusing slice a that qualifies for FIR filter implementation. Regardless of the microimage resolution M, a filter that computes a refocusing slice with $a = 0$ in horizontal direction reads

$$E'_{0/M}\left[x_k\right] = \sum_{i=0}^{M-1} \frac{1}{M} E_{f_s}\left[x_{k-i-\bmod(k+1,\,M)}\right] \tag{5}$$

when $k \in \{0, \ldots, K-1\}$. Term $\bmod(k+1, M)$ comprises a nearest-neighbor (NN) interpolation ensuring that the numerical output image resolution matches that of the input. A synthetically focused image where $a = 1$ is formed by

$$E'_{M/M}\left[x_k\right] = \sum_{i=0}^{M-1} \frac{1}{M} E_{f_s}\left[x_{k+i(m-1)}\right]. \tag{6}$$

Synthesis equations for different $a = a'/M$ are retrieved by reverse-engineering. Probably, the most straightforward refocusing filter kernel function is given by

$$E'_{1/M}\left[x_k\right] = \sum_{i=0}^{M-1} \frac{1}{M} E_{f_s}\left[x_{k-i}\right] \tag{7}$$

which computes refocusing slice $a = 1/M$. When implementing (7) as an FIR filter, it becomes obvious that the number of filter taps amounts to M. A VHDL implementation using this filter type with $M = 5$ is provided in supplementary material. In the following, we demonstrate a refocusing hardware architecture that is adapted to an SPC with $M = 3$. Then, a photograph refocused with $a = 2/3$ is computed by

$$E'_{2/3}\left[x_k\right] = \sum_{i=0}^{3-1} \frac{1}{3} E_{f_s}\left[x_{k-i+|\lceil \bmod(k+1,3)/3\rceil - 1|\times(i-1)}\right] \tag{8}$$

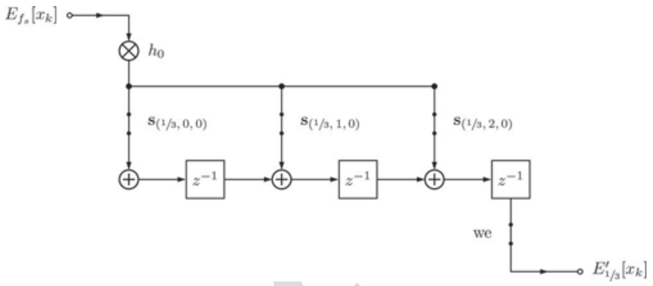
$E_{f_s}[x_k]$
h_0
$\mathbf{s}_{(1/3,\,0,\,0)}$
$\mathbf{s}_{(1/3,\,1,\,0)}$
$\mathbf{s}_{(1/3,\,2,\,0)}$
z^{-1}
z^{-1}
z^{-1}
we
$E'_{1/3}[x_k]$

Fig. 3. 1-D semisystolic FIR filter for sub-pixel shift $a = 0/3$.

where $\lceil\cdot\rceil$ is the ceiling and $|\cdot|$ the absolute value operator. An exemplary step in the computation of $E'_{2/3}[x_k]$ would be

$$E'_{2/3}\left[x_3\right] = \frac{1}{3} E_{f_s}\left[x_3\right] + \frac{1}{3} E_{f_s}\left[x_2\right] + \frac{1}{3} E_{f_s}\left[x_1\right]. \tag{9}$$

Here, fractions $1/3$ can be regarded as multipliers, denoted as h_0, which are identical for each pixel such that $h_0 = 1/M$. On the condition that incoming images are underexposed and clipping is prevented, it is noteworthy that multipliers are redundant and thus can be left out.

A. Semisystolic Modules

Equations (5)–(8) are implemented with a systolic filter design. Systolic arrays broadcast input data to many processing elements (PEs). As shown, all wired connections in a systolic filter contain at least one latch driven by the same clock signal. *semisystolic* designs omit these latches. All of the remaining designs that we consider are semisystolic, but latches can be added for systolic FPGA implementation purposes. Descriptive information about systolic arrangements can be found in [26].

A positive side effect of the systolic filter is that it can be exploited for an NN-interpolation in microimages. By letting the upsampling factor be the number of microimage samples M, the resolution loss in integral projection is compensated, since incoming and outgoing resolution are the same. Naturally, the interpolation method can be more sophisticated, which in turn requires intermediate calculations, causing delays and an increasing number of occupied logic gates. Closer inspection of (6) reveals that pixels that need to be integrated are interlaced. Thereby, gaps between merged pixels grow with ascending a and extend the filter length. The omission of pixels within gaps is realized with switches. A switch-controlled semisystolic FIR filter design of (5) with multiplier h_0 is depicted in Fig. 3. In this design, switch states are controlled by bits in a 2-D vector field denoted as $\mathbf{s}_{(a,\,w,\,p)}$ that is given by

$$\mathbf{s}_{(0/3,\,w,\,p)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

if $a = 0/3$. Depending on refocusing parameter a, switch state matrices $\mathbf{s}_{(a,\,w,\,p)}$ contain binary numbers with columns indexed by w for the state of each switch in the FIR filter and with rows indexed by p, which loads a new row of switch states when

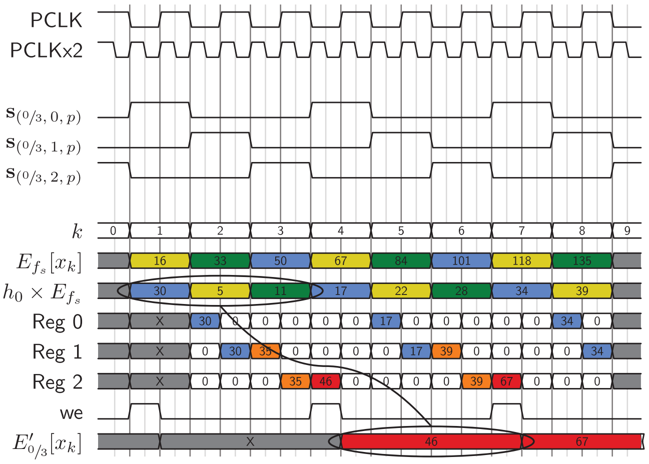

Fig. 4. Timing diagram of FIR filter module with $a = 0/3$.

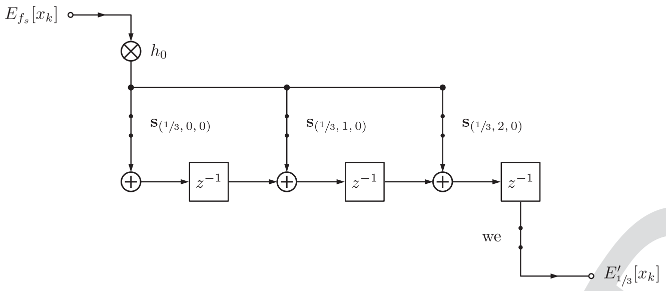
$E_{f_s}[x_k]$
h_0
$\mathbf{s}_{(1/3,\,0,\,0)}$
$\mathbf{s}_{(1/3,\,1,\,0)}$
$\mathbf{s}_{(1/3,\,2,\,0)}$
z^{-1}
z^{-1}
z^{-1}
we
$E'_{1/3}[x_k]$

Fig. 5. 1-D semisystolic FIR filter for sub-pixel shift $a = 1/3$.

incremented. In addition, a write enable switch helps to prevent intermediate falsified values from being streamed out.

For better comprehension, a timing diagram in Fig. 4 visualizes the computational concept of the FIR design from Fig. 3. Here, the pixel clock signal is given as PCLK. Furthermore, the proposed architecture employs the doubled pixel clock PCLKx2 with a time period $T_{\text{PCLKx2}} = T_{\text{PCLK}}/2$ to shift and add pixel values in a single pixel clock cycle T_{PCLK}. It is also seen that a new row of switch states is called by incrementing p every pixel clock cycle. Numbers in the data streams represent unsigned decimal 8-bit gray-scale values, which are multiplied with $h_0 = 1/3$. Pixel colors match those of the SPC ray model in Fig. 1 representing chief ray positions in microimages with $M = 3$. Orange color highlights interim results and red signifies 1-D refocused output data. Oval circles indicate that the sum of divided microimage pixels is reflected in the output pixel $E'_{0/3}[x_k]$. The filter includes an NN-interpolation upsampling the micro image resolution by factor 3. To refocus with $a = 1/3$, another FIR filter module is conceived based on (7) and depicted in Fig. 5. In reference to the previous FIR filter where $a = 0/3$, it becomes obvious that the arrangements are identical except for different switch states. The switch state matrix $\mathbf{s}_{(1/3,\,w,\,p)}$ is given by

$$\mathbf{s}_{(1/3,\,w,\,p)} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{11}$$

which means that switches remain closed at all times. A corresponding timing diagram is shown in Fig. 6. Fig. 7 depicts an FIR filter according to (8), which occupies more PEs due to the fact that the distance between added pixels grows. The corresponding switch state matrix $\mathbf{s}_{(2/3,\,w,\,p)}$ is as follows:

$$\mathbf{s}_{(2/3,\,w,\,p)} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{12}$$

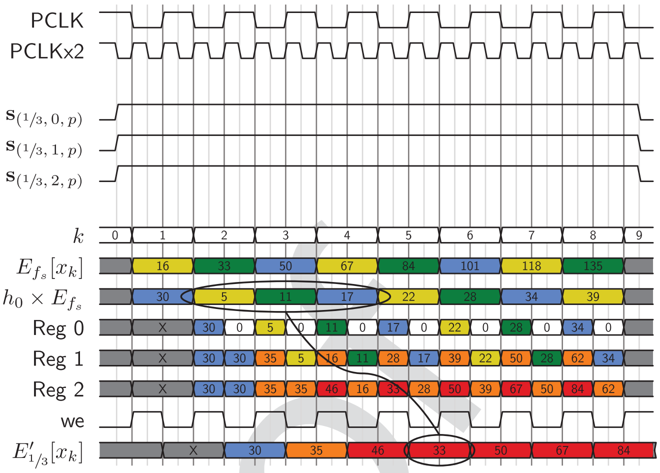

Fig. 6. Timing diagram of FIR filter module with $a = 1/3$.

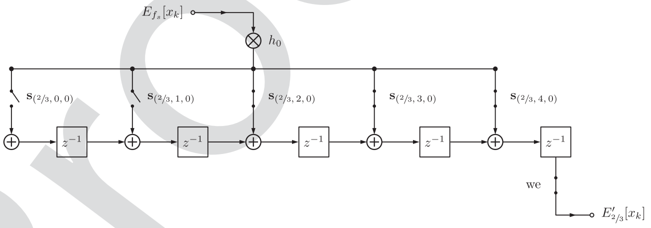
$E_{f_s}[x_k]$
h_0
$\mathbf{s}_{(2/3,\,0,\,0)}$
$\mathbf{s}_{(2/3,\,1,\,0)}$
$\mathbf{s}_{(2/3,\,2,\,0)}$
$\mathbf{s}_{(2/3,\,3,\,0)}$
$\mathbf{s}_{(2/3,\,4,\,0)}$
z^{-1}
z^{-1}
z^{-1}
z^{-1}
z^{-1}
we
$E'_{2/3}[x_k]$

Fig. 7. 1-D semisystolic FIR filter for sub-pixel shift $a = 2/3$.

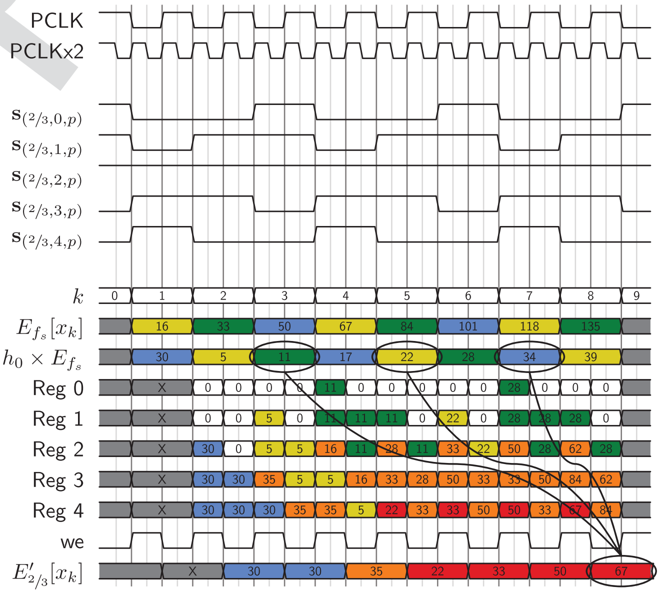

Fig. 8. Timing diagram of FIR filter module with $a = 2/3$.

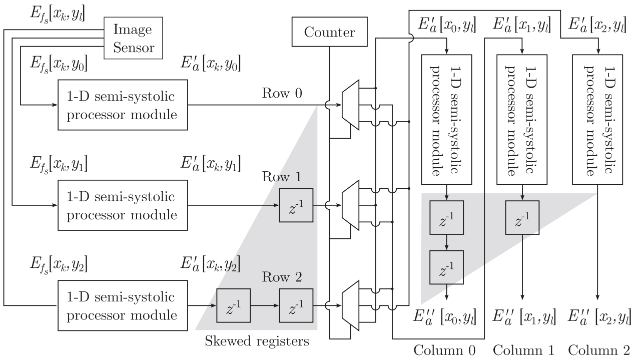

Fig. 9. Parallelized 2-D processing module array with $\iota = 3$.

producing a filter behavior shown in Fig. 8. As Fig. 7 demonstrates, a large 1-D semisystolic filter module may imply long wires when broadcasting multiplier outputs. Long wires would cause a low-pass filter behavior in the signal transmission, which affects the readability of falling and rising edges and therefore has to be avoided. To keep wires short in the broadcast net, incoming bit words can be distributed to several synchronized latches (buffers) before being merged in adders.

B. 2-D Module Array

The proposed FIR filter modules process data in 1-D and thus in horizontal or vertical directions only. Fig. 9 shows a 2-D construct of 1-D semisystolic processor modules to accomplish refocusing by processing data in both dimensions. In this example, the degree of parallelization amounts to $\iota = 3$, but could be scaled as desired until limits are reached ($\iota = L$ for image rows, $\iota = K$ for image columns).

The data flow in Fig. 9 is described in the following. First, pixels coming from the sensor are fed into horizontal processor blocks representing semisystolic FIR filter modules as proposed in the previous section. All semisystolic processor modules are identical whereas the type relies on the refocusing parameter a. In the second stage, horizontally processed data rows $E'_a\,[x_k\,,y_l]$ are delayed using skewed registers and assigned to another arrangement of semisystolic modules making it possible to form an incoming image column (e.g., $E'_a\,[x_0\,,y_l]$). Here, demultiplexers are driven by a pixel counter to assist in the correct assignment of pixels values. This assures that pixels from different rows sharing index k are sent to the same vertical processing unit that produces an image column (e.g., $E''_a\,[x_0\,,y_l]$) of the final refocused image. For synchronization purposes, an additional array of skewed registers can be optionally placed behind column processor blocks.

In order to estimate the computation time, it is assumed hereafter that the hardware system refers to the ideal case of maximum parallelization where $\iota = L$ or $\iota = K$ for each dimension, respectively. Besides, it is supposed that color channels are also parallelized causing no extra time delay. The shift and integration for a single output pixel refocused with $a = 1/M$ takes M pixel clock cycles in 1-D when using twice the pixel clock to process them. Taking this as an example, the overall number of steps η to compute a single image $E''_{1/3}$ with K-by-L resolution is given by

TABLE I
BENCHMARK OF PROPOSED ARCHITECTURE

	Proposed design	Pérez *et al.* [7]	GPU	Matlab
Clock frequency	100 MHz	100 MHz	1.35 GHz	3.40 GHz
Time to compute frame	96.2 μs	105.9 ms	1.38 ms	12.1 s

$$\eta = 2(\Lambda + M) + 2(K - 1) + L - 1 \tag{13}$$

where Λ represents a single clock cycle step to compute the mathematical product of an incoming pixel value. The total computation time $\mathcal{O}$ for a single image can be obtained by

$$\mathcal{O}(\eta) = \eta \times T_{\text{PCLK}}\,. \tag{14}$$

This duration reflects the theoretical time that elapsed from the moment the first pixel $E_{f_s}\,[x_k\,,y_l]$ entered the logic gate until the final output pixel $E''_a\,[x_k\,,y_l]$ is available. When pipelining the data stream, output pixels of a subsequent image arrive directly after that letting the overall computation time for a single frame be represented by the delay time of the computational focusing system. Once the first refocused photograph is received, the number of remaining computational steps η_{sub} for every following image amounts to:

$$\eta_{\text{sub}} = L - 1 + K - 1\,. \tag{15}$$

To assess performance limits of the presented architecture, we performed a benchmark comparison between this approach, the FPGA-based implementation of Pérez *et al.* [7], and a GPU-based approach [22]. In this comparison, a 3201-by-3201 pixel image ($K = L = 3201$) with 291-by-291 microlenses was computationally refocused in 105.9 ms at 100-MHz clock frequency. Thereby, the microimage resolution is $M = 11$ and the output image resolution amounts to 589-by-589, which is less than $1/6$ of the incoming image. Conversely, the proposed semisystolic method numerically preserves the incoming spatial resolution by employing an NN-interpolation in $\eta = 1 + 11 + 3200 + 1 + 11 + 3200 + 3200$ steps yielding $\mathcal{O}(\eta) = 96.2\,\mu$s computation time for a single frame when running at 100 MHz pixel clock. Each subsequent frame, however, can be processed in $\eta_{\text{sub}} = 3200 + 3200$ steps, which is available at every $\mathcal{O}(\eta_{\text{sub}}) = 64\,\mu$s. In comparison, an identical implementation based on the GPU implementation by Lumsdaine *et al.* [22] takes approximately 1.38 ms on average, whereas a MATLAB implementation takes approximately 12.1 s per image on average as seen in the overview in Table I.

In this comparison, we employed the Spartan-6 XC6SLX45 chip using the ISE WebPACK design software from Xilinx. The refocusing shader were executed on a Fermi architecture GeForce 480M GTX with 2 GB of GDDR5 RAM running at 1200 MHz, connected to a 256 bit bus [22]. For the CPU environment, we used MATLAB 7.11.0.584 (R2010b) on an Intel Core i7-3770 CPU @ 3.40 GHz without multithreading.

IV. VALIDATION

In this section, we evaluate the functionality of the proposed FPGA-based refocusing hardware design. For that purpose, the

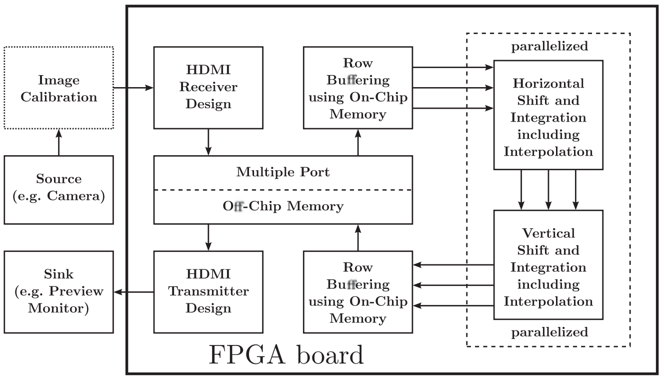
Image Calibration
Source (e.g. Camera)
Sink (e.g. Preview Monitor)
HDMI Receiver Design
Row Buffering using On-Chip Memory
parallelized
Horizontal Shift and Integration including Interpolation
Multiple Port
Off-Chip Memory
Vertical Shift and Integration including Interpolation
HDMI Transmitter Design
Row Buffering using On-Chip Memory
parallelized
FPGA board

Fig. 10. Block diagram (borrowed from [6]) for experimental validation. Single arrows denote serialized whereas three arrows indicate parallelized data streams. Row buffers are employed to simulate data parallelization in the experiment.

TABLE II
UTILIZATION SUMMARY FOR XC6SLX45–CSG324

On-chip	Power [mW]	Used	Available	Utilization [%]
Clocks	82.97	8	—	—
Logic	2.68	957	27 288	4
Signals	12.82	1646	—	—
IOs	461.29	84	218	39
PLLs	314.69	2	4	50
MCBs	189.00	1	2	50
Quiescent	79.02	—	—	—
Total	1142.46	—	—	—

VHSIC HDL (VHDL) is used to configure the FPGA where VHSIC stands for *very high speed integrated circuit*. A schematic file, generated from a VHDL compiler, is then flashed onto the FPGA chip model XC6SLX45. Fig. 10 contains a block diagram illustrating the implemented processing architecture used to validate the design proposed in the previous section. The FPGA board features high-definition multimedia interface (HDMI) connectors such that video frame transmission is accomplished using the transition minimized differential signaling (TMDS) protocol. TMDS receiver and transmitter designs have been integrated on the FPGA to fulfill deserialization, serialization just as decoding and encoding tasks. Off-chip memory is used for buffering decoded and serialized video frames outside the FPGA since the amount of image data exceeds internal memory storage.

In our implementation, a row of switch settings is loaded from a look-up table (LUT) every clock cycle starting from the first row again after the last one is reached. The switch-state LUTs can be stored in block random-access memorys (BRAMs), which are part of the FPGA. The integration of multiplier h_0 is also achieved using on-chip memory, making it called *stored product*. In accordance with the TMDS protocol specification, a decoded pixel value is of 8-bit depth per color channel, which yields a manageable number of 256 possible results when dividing by M. Thus, quotients can be precalculated for a specific divisor M and stored in one BRAM per color channel for each image row. Note that these BRAMs are read-only memories.

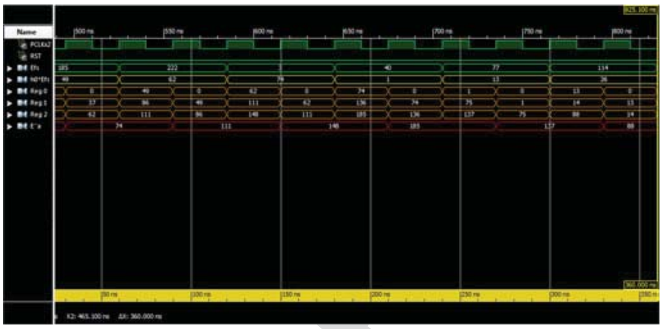

Fig. 11. Timing diagram example from ISE simulator.

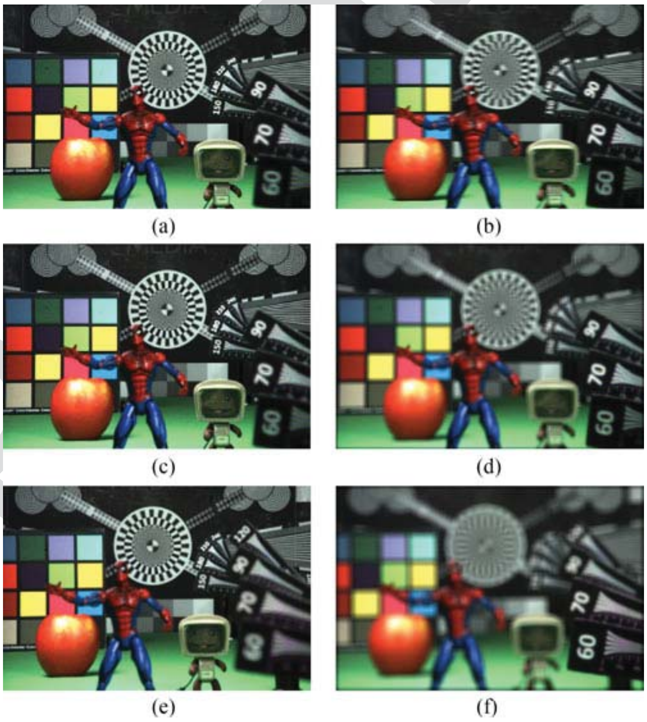

Fig. 12. Refocused photographs using the proposed architecture. (a) $E'_{0/3}$. (b) $E'_{5/3}$. (c) $E''_{0/3}$. (d) $E''_{5/3}$. (e) $E''_{0/5}$. (f) $E''_{8/5}$. Input and output spatial image resolutions amount to 843-by-561 pixels with $M = 3$ in (a)–(d). Intermediate horizontally processed images are shown in (a) and (b) whereas (c) and (d) depict fully refocused images after horizontal and vertical processing with varying a. In comparison, output images in (e) and (f) with 1405-by-935 pixel resolution expose improved synthetic blur by using a linear interpolation of whole microimages with $M = 5$. Reducing a lightfield's angular sampling rate M extends the depth of field [8] and leads to blur aliasing in case of angular undersampling [15].

A screenshot from an exemplary timing diagram simulation where $a = 1/3$ and $T_{\text{PCLK}} = 60$ ns is provided in Fig. 11 with the code attached to this article. This VHDL-implemented hardware simulation shows that the filter behaves as expected, justifying the conceived architecture. PCLKx2 can be obtained with a phase-locked loop (PLL). An overview of the implemented design comprising a single FIR filter with $a = 1/5$ is presented in Table II where it can be seen that inputs/outputs (IOs) and PLLs make up by far most of the power consumption. This is due to the included HDMI transceiver, memory controller block (MCB) and color conversion modules. Parts

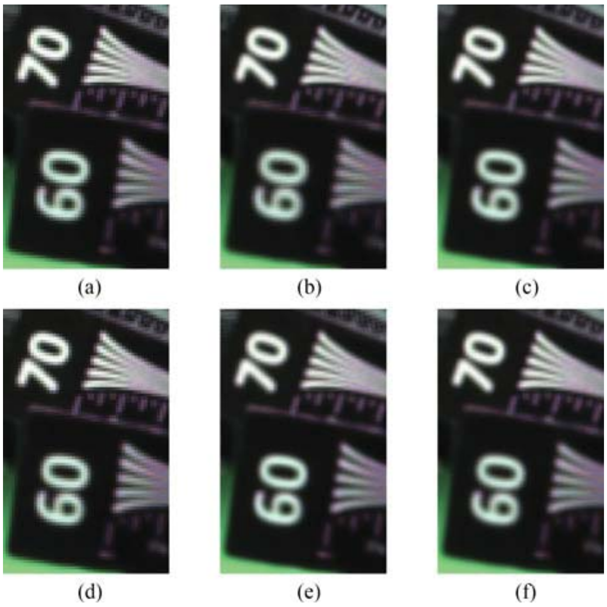

Fig. 13. (a) NN interp. $E''_{5/5}$ (while refocusing). (b) NN interp. $E''_{4/5}$ (while refocusing). (c) Lin. interp. $E''_{5/5}$ (while refocusing). (d) NN interp. $E''_{5/5}$ (after refocusing). (e) NN interp. $E''_{6/5}$ (while refocusing). (f) Lin. interp. $E''_{5/5}$ (after refocusing). Resolution comparison where (a), (c), (d) and (f) show the same region refocused with $a = 5/5$ using different interpolation techniques during and after shift and integration. Images in (b) and (e) are NN-interpolated versions with varying a indicating significant variation of the spatial resolution when compared with (a) and (d). Effective resolution is more consistent when using linear interpolation [e.g., compare (d), (e), and (f)].

of these modules may be omitted or replaced by on-board integrated circuits (ICs) in a prototyping stage. Furthermore, Table II gives indication that adding more FIR filters for full parallelization (maximum L and K) is noncritical to power, but may be limited to the number of logic slices in a Spartan-6 device.

Presented refocusing synthesis formulas require all microimages to be of a consistent size. This is not the case, however, in raw lightfield photographs. As indicated with the experimental architecture in Fig. 10, microimage cropping remains an external process performed prior to streaming the data to the FPGA. Embedding this process on an FPGA is essential for prototyping, but left for future work. To comply with FIR filter designs in Section III, the microimage size is reduced to $M = 3$ and $M = 5$ for comparison. Lightfield images have been acquired by our custom-built plenoptic camera with an MLA of 281 microlenses per row and 188 per column. Insightful details on the camera calibration can be found in [27].

Fig. 12 depicts refocused photographs computed by the proposed 2-D module array to accomplish real-time refocusing. Intermediate results after processing images in a horizontal direction are seen in Fig. 12(a) and (b). Their fully refocused counterparts are found in Fig. 12(c) and (d). Closer inspection of Fig. 12(d) indicates aliasing in blurred regions. This is due to an undersampled directional domain as there are only 3-by-3 samples per microimage ($M = 3$) in the incoming lightfield capture. Aliasing in synthetic image blur is an observation Ng already pointed out in his thesis [15]. To combat the aliasing problem, the author suggests to sufficiently increase the microimage sampling rate M. Fig. 12(e) and (f) shows refocused images obtained from a raw capture with a native microimage resolution of 5-by-5 pixels ($M = 5$) using a linear interpolation instead of NN. There, it can be seen that aliasing artifacts are satisfyingly suppressed. A comparison of output image resolutions using the inherent NN-interpolation of proposed FIR filters is provided in Fig. 13. Results in Fig. 13(a)–(f) suggest that interpolating microimages while refocusing with $a \in \mathbb{Z}$ using (6) corresponds to a conventional 2-D image interpolation. On the contrary, an effective resolution enhancement can be observed when comparing Fig. 13(a) where $a = 5/5$ with Fig. 13(b) where $a = 4/5$, which are both computed from the same raw image using NN-interpolation. Given that respective objects are acceptably well covered by their depth of field and exhibit best focus, it is possible to state that improved resolution is obtained by refocusing with noninteger numbers ($a \notin \mathbb{Z}$). This effective resolution variation is a consequence of the microimage repetition and the interleaving filter kernel for the refocusing synthesis yielding identical values for adjacent output pixels when $a \in \mathbb{Z}$, but varying intensities for contiguous pixels if $a \in \mathbb{R}$. This can be seen by inspecting output data streams $E'_a[x_k]$ of the timing diagrams in Figs. 4 and 6. To work toward consistency in spatial resolutions for varying a, it is thus essential to employ linear interpolation prior to distributing microimage pixels through the FIR broadcast net. A positive side effect in upsampling microimages is that refocused image slices $E''_a[x_k, y_l]$ are not only interpolated in spatial-domain, but also subsampled along depth as demonstrated in [8].

V. Conclusion

This paper demonstrated methods to derive optimized FIR refocusing filter kernels for a time- and cost-efficient hardware implementation. Simulating the conceived architecture proved that real-time refocusing can be accomplished with a computation time of $96.24\,\mu$s per frame reducing the delay time by 99.91 % in comparison with a previous state-of-the-art attempt. By interpolating microimages, it was shown how to retain the numerical sensor resolution in refocused photographs. The proposed architecture can serve as a groundwork for application-specific integrated circuit chips.

A limitation of the results is that timing delays have been simulated and need to be verified using chip analyzing tools. As the number of required PEs grows with higher image resolutions, it may exceed the gate count capacity of the FPGA in full parallelization. Besides this, care needs to be taken to prevent long wires in the broadcast net. For the hardware system's reliability, it is also recommended to convert semisystolic arrays into a full-systolic architecture. To achieve consistency in microimage size (M), cropping of the same has to be integrated as a preceding processing stage on the FPGA chip. Furthermore, a bilinear interpolation ought to be implemented to replace microimage repetition (NN-interpolation) and work toward consistent effective resolutions in refocused images, although this will cause additional delays.

A competitive design approach may conceive a refocusing architecture based on the FSP theorem. It is, however, expected that the Fourier transform produces larger time delays. A considerable alternative to an FPGA-based implementation is the employment of a GPU as this takes less design effort, however, by inducing larger delays and more power consumption.

Deployment of proposed design to an FPC is thought to be impractical, since there is a fundamental difference between SPC and FPC with regards to the optical design (number of microlenses and focus position of MLA). On the algorithmic level, SPC refocusing is a pixel-based integration whereas an FPC requires the integration of overlapping areas of shifted microimage patches such that a refocusing algorithm has to be designed specific to the type of plenoptic camera.

References

[1] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proc. 27th Ann. Conf. Comput. Graph. Interactive Tech.*, ser. SIGGRAPH '00. New York, NY, USA, 2000, pp. 297–306. [Online]. Available: http://dx.doi.org/10.1145/344779.344929

[2] R. Ng, M. Levoy, M. Brèdif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Stanford University, Tech. Rep. CTSR 2005-02, 2005.

[3] R. Ng, "Fourier slice photography," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 735–744, Jul. 2005. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073256

[4] A. Lumsdaine and T. Georgiev, "Full resolution lightfield rendering," Adobe Systems, Inc., Tech. Rep., Jan. 2008.

Q5 [5] C. Perwass and L. Wietzke, "Single lens 3D-camera with extended depth-of-field," *Proc. SPIE*, vol. 8291, February 2012. [Online]. Available: http://dx.doi.org/10.1117/12.909882

Q6 [6] C. Hahne and A. Aggoun, "Embedded FIR filter design for real-time refocusing using a standard plenoptic video camera," *Proc. SPIE*, vol. 9023, 2014. [Online]. Available: http://hdl.handle.net/10547/313167

[7] J. Pérez, E. Magdaleno, F. Pérez, M. Rodríguez, D. Hernández, and J. Corrales, "Super-resolution in plenoptic cameras using FPGAs," *Sensors*, vol. 14, no. 5, pp. 8669–8685, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/5/8669

[8] C. Hahne, A. Aggoun, V. Velisavljevic, S. Fiebig, and M. Pesch, "Refocusing distance of a standard plenoptic camera," *Opt. Exp.*, vol. 24, no. 19, pp. 21 521–21 540, Sep. 2016. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-24-19-21521

[9] M. Levoy and P. Hanrahan, "Light field rendering," Stanford University, Stanford, CA, USA, Tech. Rep., 1996.

Q7 [10] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, "A real-time distributed light field camera," in *Proc. 13th Eurographics Workshop Rendering*. Aire-la-Ville, Switzerland, Switzerland, 2002, pp. 77–86. [Online]. Available: http://dl.acm.org/citation.cfm?id=581896.581907

[11] K. Venkataraman *et al.*, "PiCam: An ultra-thin high performance monolithic camera array," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 166:1–166:13, Nov. 2013. [Online]. Available: http://doi.acm.org/10.1145/2508363.2508390

[12] G. Lippmann, "Épreuves réversibles donnant la sensation du relief," *Académie Des Sci.*, pp. 446–451, Mar. 1908.

[13] E. H. Adelson and J. Y. Wang, "Single lens stereo with a plenoptic camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 99–106, Feb. 1992.

[14] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Proc. Comput. Models Visual Process.*, Cambridge, MA, USA: MIT Press, 1991, pp. 3–20.

[15] R. Ng, "Digital light field photography," Ph.D. dissertation, Stanford University, Stanford, CA, USA, July 2006.

[16] T. Georgiev and A. Lumsdaine, "The focused plenoptic camera," in *Proc. Int. Conf. Comput. Photography*, 2009.

[17] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, "Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1276377.1276463

[18] Z. Xu, J. Ke, and E. Y. Lam, "High-resolution lightfield photography using two masks," *Opt. Exp.*, vol. 20, no. 10, pp. 10 971–10 983, May 2012. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-20-10-10971

[19] C. Hahne, A. Aggoun, V. Velisavljevic, S. Fiebig, and M. Pesch, "Baseline and triangulation geometry in a standard plenoptic camera," *Int. J. Comput. Vis.*, vol. 126, no. 1, pp. 21–35, Jan. 2018.

[20] Lytro, Inc., "Lytro-press releases," 2016. [Online]. Available: https://www.lytro.com/press/releases/lytro-brings-revolutionary-light-field-technology-to-film-and-tv-production-with-lytro-cinema. Accessed on: Aug. 24, 2016.

[21] Z. Mhabary, O. Levi, E. Small, and A. Stern, "Fast and exact method for computing a stack of images at various focuses from a four-dimensional light field," *J. Electron. Imaging*, vol. 25, no. 4, 2016, Art. no. 043002. [Online]. Available: http://dx.doi.org/10.1117/1.JEI.25.4.043002

[22] A. Lumsdaine, G. Chunev, and T. Georgiev, "Plenoptic rendering with interactive performance using gpus," *Proc. SPIE*, vol. 8295, pp. 829 513–829 513–15, 2012. [Online]. Available: http://dx.doi.org/10.1117/12.909683

[23] D. G. Bailey, *Design for Embedded Image Processing on FPGAs*. Hoboken, NJ, USA: Wiley, 2011.

[24] L. F. Rodríguez-Ramos, Y. Marín, J. J. Díaz, J. Piqueras, J. García-Jiménez, and J. M. Rodríguez-Ramos, "FPGA-based real time processing of the plenoptic wavefront sensor," in *Proc. Adaptative Opt. Extremely Large Telescopes*, 2010, Paper 7007.

[25] R. Wimalagunarathne, A. Madanayake, D. G. Dansereau, and L. T. Bruton, "A systolic-array architecture for first-order 4-D IIR frequency-planar digital filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 3069–3072.

[26] Xilinx, Inc., "A 1D Systolic FIR," 2002. [Online]. Available: http://www.iro.umontreal.ca/aboulham/F6221/Xilinx.htm. Accessed on: Dec. 12, 2015.

[27] C. Hahne, "The standard plenoptic camera – Applications of a geometrical light field model," Ph.D. dissertation, University of Bedfordshire, Luton, U.K., Jan. 2016.

Christopher Hahne received the B.Sc. degree from the University of Applied Sciences, Hamburg, Germany, in 2012, and the Doctoral degree from the University of Bedfordshire, Luton, U.K., in 2016, in a bursary-funded Ph.D. program.

He is affiliated with BASF subsidiary trinamiX GmbH, Ludwigshafen, Germany, where he currently works as the Manager of Simulation & Software on adaptive three-dimensional sensing. He worked at R&D departments of Rohde & Schwarz GmbH & Co. KG, Munich, Germany, in 2010, and Arnold & Richter Cinetechnik GmbH & Co. KG, Munich, Germany, in 2011. Subsequently, he became a Visiting Student with Brunel University, London, U.K., in 2012. Q8

Andrew Lumsdaine (SM'15) is an internationally recognized expert in the area of high-performance computing who has made important contributions in many of the constitutive areas of HPC. In particular, he has contributed in the areas of HPC systems, programming languages, software libraries, and performance modeling. His work in HPC has been motivated by data-driven problems (large-scale graph analytics), as well as more traditional computational science problems. In addition, outside of the realm of HPC, he has done seminal work in the area of computational photography and plenoptic cameras. In his career, he has authored or coauthored more than 200 articles in top journals and conferences and holds 15 patents. He has also contributed important software artifacts to the research community, especially in the area of message passing interface (MPI). He is active in a number of standardization efforts with important contributions to the MPI specification, the C++ programming language, and the Graph 500. Q9

Amar Aggoun received the “Ingenieur d’état” degree in electronics engineering from Ecole Nationale Polytechnique d’Alger, Algiers, Algeria, and the Ph.D. degree in electronic engineering from the University of Nottingham, Nottingham, U.K.

He is currently the Head of School of Mathematics and Computer Science and Professor of Visual Computing with the University of Wolverhampton, Wolverhampton, U.K. His academic carrier started at the University of Nottingham where he held the positions of Research Fellow in low power DSP architectures and Visiting Lecturer in electronic engineering and mathematics. In 1993, he joined De Montfort University as a Lecturer and progressed to the position of Principal Lecturer in 2000. In 2005, he joined Brunel University as a Reader with Information and Communication Technologies. From 2013 to 2016, he was at the University of Bedfordshire as the Head of School of Computer Science and Technology. He was also the Director of the Institute for Research in Applicable Computing which oversees all the research within the School. His research is mainly focused on three-dimensional (3-D) Imaging and Immersive Technologies and he successfully secured and delivered research contracts worth in excess of 6.9M, funded by the research councils UK, Innovate UK, the European commission and industry. Amongst the successful project, he was the initiator and the principal coordinator and manager of a project sponsored by the EU-FP7 ICT-4-1.5-Networked Media and 3-D Internet, namely live immerse video-audio interactive multimedia. He holds 3 filed patents, authored or coauthored more than 200 peer-reviewed journals and conference publications and contributed to two white papers for the European Commission on the future internet.

Dr. Aggoun also served as an Associate Editor for the IEEE/OSA JOURNAL OF DISPLAY TECHNOLOGIES. Q10

Vladan Velisavljevic (M’06–SM’12) received the Ph.D. degree in the field of signal and image processing from École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2005.

He is a Reader (Associate Professor) in visual systems engineering with the School of Computer Science and Technology and the Head of the Centre for Research in Signals, Sensors and Wireless Technology, University of Bedfordshire, Luton, U.K., since 2011. Previously, he was a Senior Research Scientist with Deutsche Telekom Laboratories, University of Technology Berlin, Germany, in 2006–2011, and a Doctoral Assistant with LCAV, EPFL, Switzerland, in 2001–2005. He has authored or coauthored more than 60 peer-reviewed journal and conference publications and two book chapters.

Dr. Velisavljevic serves as an Associate Editor for *Elsevier Signal Processing: Image Communication* and for *IET Journal of Engineering* and he is a Co-Chair of the IEEE ComSoc MMTC Interest Group on 3-D Processing and Communications. He was a General Chair of the IEEE MMSP 2017, a Lead Guest Editor for special issue on Visual Signal Processing for Wireless Networks at the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING in February 2015 and special session organizer at 3DTV-Con 2015 and IEEE ICIP 2011. He was also Associate Editor for IEEE ComSoc MMTC R-Letters and Member of the Review Board for the IEEE ComSoc Multimedia Communications TC. He has served as a TPC member and reviewer for a number of conferences and journals.