A Real-time Coprime Line Scan Super-resolution System for Ultra-fast Microscopy

Runbin Shi, Justin S. J. Wong, Edmund Y. Lam, *Fellow, IEEE*, Kevin K. Tsia, and Hayden K.-H. So, *Senior Member, IEEE*

***Abstract*—A fundamental technical challenge for ultra-fast cell microscopy is the trade-off between imaging throughput and resolution. In addition to throughput, real-time applications such as image-based cell sorting further requires ultra-low imaging latency to facilitate rapid decision making on a single-cell level. Using a novel coprime line scan sampling scheme, a real-time low-latency hardware super-resolution system for ultra-fast time-stretch microscopy is presented. The proposed scheme utilizes analog-to-digital converter with a carefully tuned sampling pattern (shifted sampling grid) to enable super-resolution image reconstruction using line scan input from an optical front-end. A fully-pipelined FPGA-based system is built to efficiently handle the real-time high-resolution image reconstruction process with the input subpixel samples while achieving minimal output latency. The proposed super-resolution sampling and reconstruction scheme is parametrizable and is readily applicable to different line scan imaging systems. In our experiments, an imaging latency of 0.29 µs has been achieved based on a pixel-stream throughput of 4.123 giga pixels per second, which translates into imaging throughput of approximately 120 000 cells per second.**

***Index Terms*—CLSS, Line scan super resolution, optical microscopy, FPGA, ADC**

I. Introduction

RECENT advances in high-throughput cell microscopy promise a new generation of image-based biomedical applications otherwise impossible with traditional biomolecular assays [1]–[4]. In [5], for instances, researchers have demonstrated a high-speed cell sorting system that is activated through analysis of cell images alone. In [6], researchers have also demonstrated that by using image-derived markers alone, cell types can be classified and clustered effectively. When compared to traditional genetic, epigenetic or transcriptomic based analytic methods [7], [8], image-based systems are able to examine cells and profile their phenotypes at much higher throughput, thus making them promising tools to address problems that demand examination of large populations of cells, such as for cell-based drug screen [9], rare cell detection [10], and single cell analytic [11]. The challenge with these systems, however, is that researchers must be able to derive adequate information from the cell images in real time to facilitate various forms of image analytics while avoiding significant impact to the overall imaging throughput.

In this work, a real-time coprime line scan super-resolution (CLSS) system is presented. CLSS is capable of producing images with improved spatial resolution in the horizontal direction in real-time from scan lines that are sampled by analog-to-digital converter (ADC) at limited or reduced sampling frequency. One key innovation of CLSS rests on its novel sampling technique, which collects data samples in each line at a frequency that is coprime to the line repetition frequency to produce a shifted sampling grid. High resolution images are then produced computationally from this low resolution shifted sampling grid. By controlling the spatial resolution of the scan lines, CLSS effectively enhances resolution in the horizontal direction by leveraging over-sampled scan lines in the vertical direction.

The design of CLSS was motivated by the limited real-time performance of the asymmetric-detection time-stretch optical microscopy (ATOM) system [12]. The original ATOM system was capable of imaging over 100 000 cells per second with an approximate 1 µm axial resolution. However, the superior throughput-resolution performance relied heavily on the analog bandwidth of the ADC that samples the photo-detector (PD) output. In the original system, an 80 giga sample per second (GSPS) high speed oscilloscope with limited on-board buffer was used to capture the time-stretched laser pulses for image formation, heavily limiting real time applications of the proposed scheme. Lowering the ADC bandwidth may substantially improve its throughput performance but at a cost of reduced image resolution in the horizontal direction orthogonal to the cell flow. Multiframe based image super-resolution (SR) has been well discussed in the conventional 2-D image processing [13]. Inspired by this technique, several SR methods have been proposed previously for the 1-D line-scan imaging system that improve the reduced horizontal resolution via combining multiple low-resolution lines (frames) which has an intended sub-pixel staggering pattern in vertical direction [14]–[18]. In particular, a 2-D sensor based on pixel staggering is proposed in [18] that was dedicated to the cell imaging. Yet none of the previous work is capable of producing high resolution images in real-time at the ultra-fast image acquisition rate of this work.

In this work, in place of the original 80 GSPS sampling oscilloscope, a 4 GSPS streaming ADC directly connected to a field-programmable gate-array (FPGA) is used. By optimizing the spacing and frequency of the line scan together with the ADC sampling frequency, we demonstrate an implementation of CLSS that improves horizontal resolution by $9\times$ over its baseline. Fig. 1 shows an example cell images that illustrate the improved image quality. Furthermore, by taking advantage of the tight system integration, a real-time super-resolution

R. Shi, J.S.J. Wong, E.Y. Lam, K.K. Tsia and H.K.-H. So are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: rbshi@eee.hku.hk, jsjwong@hku.hk, elam@eee.hku.hk, tsia@hku.hk, hso@eee.hku.hk)

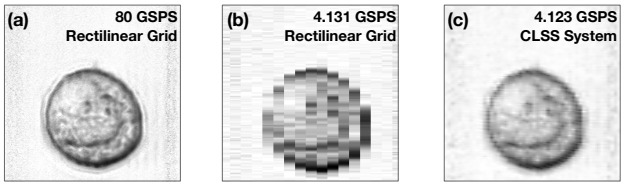

Fig. 1. (a) High-resolution cell image sampled with an 80 GSPS oscilloscope; (b) Low-resolution image sampled with 4.131 GSPS ADC; (c) Image obtained in CLSS system and the sample rate is $\approx$ 4.123 GSPS.

(SR) image construction hardware has been implemented directly in the controlling FPGA with the raw ADC readings as input. With a tight integration between optics, microfluidics and electronics, and a fully-pipelined image construction hardware, the resulting system is capable of producing super-resolution images with a latency of 0.29 µs and no impact to the imaging throughput.

It is worth noting that while the proposed system was motivated and demonstrated with the ATOM system, the proposed CLSS scheme is a generic super-resolution methodology that is applicable to many line scan imaging systems similar to ATOM. As such, the main contribution of this work rests on the following 3 aspects:

- We propose a novel coprime line scan super-resolution system that allows efficient enhancement to image resolution in the horizontal direction with minimal impact to the overall imaging system throughput;
- We demonstrate system-level co-optimization techniques among the optic, microfluidic, and electronic subsystems, which facilitates the CLSS sampling scheme in real-time with low system overhead;
- We present the design and implementation of a fully-pipeline image construction FPGA hardware that is capable of producing images at line speed by using a novel line buffering scheme with minimal on-chip memory.

In Section II, we present the key innovation of CLSS mechanism. Then, in Section III, we describe the design of the CLSS-based SR system on ATOM microscopy. Section IV proposes the design of SR-image reconstruction module on FPGA that permits the high-throughput and real-time performance. Extensive experimental evaluations are then given in Section V, followed by concluding remarks in Section VI.

II. CLSS System Overview

The proposed CLSS system works with image acquisition systems that have a continuous line scan input such as the ATOM high-speed microscopy system[12]. In the case of ATOM, cells flowing in the microfluidic imaging channel are imaged by periodic spatially-dispersed laser pulses as shown in Fig. 2. Each resulting pulse is subsequently spectrally-encoded with the information of one image line and is time-stretched by group velocity dispersion (GVD) [19] before it is transformed by a photodetector (PD) into a continuous analog electronic signal, which serves as the input to the CLSS system. The continuous analog signal is sampled and digitized into pixels by the ADC at a frequency synchronized to the laser pulses. The pixel synchronization clock $\mathtt{CLK}_{\mathrm{samp}}$ with frequency f_{samp} is derived from the line-scan clock $\mathtt{CLK}_{\mathrm{line}}$ which follows the phase and frequency (f_{line}) of the source laser pulses. The resulting pixel stream from the ADC is subsequently delivered to the FPGA in which the SR image is constructed.

The key innovation of CLSS rests on its sampling method. In a conventional system, images are reconstructed from continuous line scan input with samples taken at frequency f_{samp} equal to the integer multiple of the line repetition frequency f_{line}, i.e.

$$f_{\mathrm{samp}} = k \cdot f_{\mathrm{line}}$$

for some integer k. As a result, every k samples from the input will become a row (line) of pixels in the resulting image. As visualized in Fig. 3(a), this sampling technique results in a *rectilinear grid* of samples in the resulting image. The horizontal spatial resolution of this rectilinear sampling grid is thus determined by the ratio $k = f_{\mathrm{samp}}/f_{\mathrm{line}}$.

In contrast, the CLSS system chooses a sampling frequency that is *non-integer multiple* of the line repetition frequency. In particular, we set a special f'_{sample} that is a non-integer multiple of f_{line}. By doing so, a small spatial shift of sample position is introduced at the beginning of each line as Fig. 3(b) shows. With this deliberate shift, the pixels in neighboring lines are obtained with a sub-pixel displacement.

The actual ratio between f'_{samp} and f_{line} in the CLSS system is characterized by the tuples $(\mathbf{p}, \mathbf{q})$. The parameter $\mathbf{p}$ and $\mathbf{q}$ are chosen such that $\mathbf{p}$ pixels are sampled uniformly in every $\mathbf{q}$ scan lines. In other words, we have the relationship between the f'_{samp} and f_{line} as follows:

$$f'_{\mathrm{samp}} = \frac{\mathbf{p}}{\mathbf{q}} \cdot f_{\mathrm{line}} \tag{1}$$

in which f_{line} is a fixed parameter from the optical front-end of the imaging system and f'_{samp} decides the ADC sampling frequency based on $(\mathbf{p}, \mathbf{q})$. The parameters $(\mathbf{p}, \mathbf{q})$ must be chosen with the following two constraints:

i) $\mathbf{p}$ and $\mathbf{q}$ should be *relatively prime*.
ii) $\frac{\mathbf{p}}{\mathbf{q}} \cdot f_{\mathrm{line}} \leq f_{\mathrm{ADC}}$

In this way, each of the $\mathbf{q}$ lines has a unique sampling-position shift but returns to the initial position after $\mathbf{q}$ lines, forming a periodical sample shift pattern. To illustrate this, we set $(\mathbf{p}, \mathbf{q})$ to (11,3), thus the line and sample clocks with f_{line} and f_{samp} have waveforms as shown in Fig. 4(b) and (d) respectively. The corresponding sample grid is depicted in Fig. 3(b). We can observe that 11 pixels are sampled uniformly over 3 lines and the sample pattern repeats every 3 lines. The second constraint (ii) states that the values of $(\mathbf{p}, \mathbf{q})$ should guarantee that f'_{samp} is less than or equal to the maximum operating frequency of the sampling ADC (f_{ADC}).

III. Analogue front-end of CLSS for ATOM Microscopy and SR methods

The CLSS system as shown in Fig. 2 can be separated into two sections containing (i) the analogue front-end with clock generation, synchronization circuitries, and ADC; (ii) the FPGA based digital image processing unit for CLSS image reconstruction. In this section, we take the ATOM system as a concrete example to describe the setup and implementation

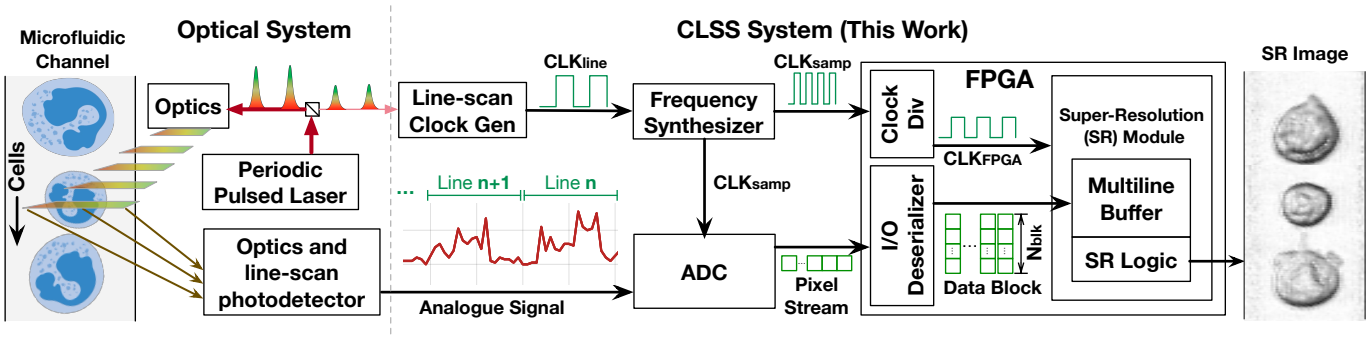

Fig. 2. Block diagram of the proposed CLSS system connected to the optical front-end of line-scan cell microscopy. Discrete lines of continuous analogue signal are obtained by the photodetector (PD), and digitized pixel values are acquired by an ADC at frequency f_{samp} of $\texttt{CLK}_{\text{samp}}$ synchronized to $\texttt{CLK}_{\text{line}}$.

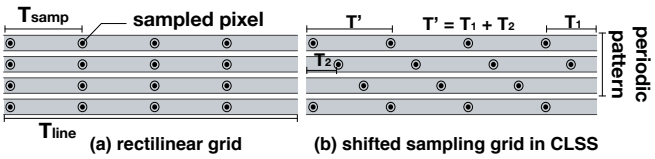

Fig. 3. (a) is the rectilinear grid sampled with traditional method; (b) is shifted sampling grid obtained by the novel CLSS method. Pixel sampling frequency in the two cases are f_{samp} and f'_{samp} respectively, and their corresponding sampling clocks are illustrated by Fig. 4(c)(d).

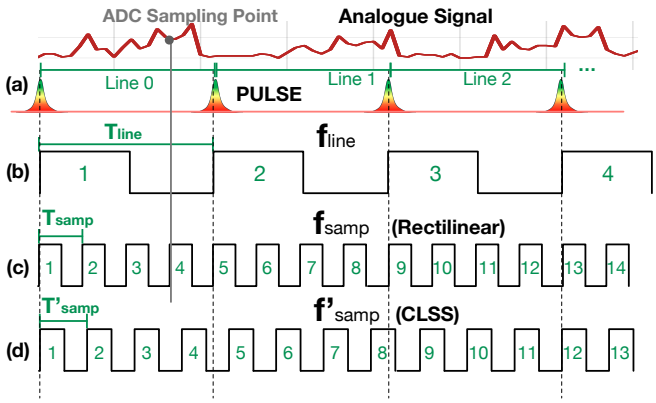

Fig. 4. Timing diagram of clock signals in the imaging system. (a) are the laser pulses, and (b) is the line clock ($\texttt{CLK}_{\text{line}}$). (c) is the traditional sample clock with f_{samp} that is an integer multiple of f_{line} in (b). (d) shows the shifted sample clock with f'_{samp} in CLSS system which is coprime to f_{line}.

of part (i) leading up to the CLSS image processing unit in the next section.

A. Clock synchronization and synthesis

Since each pulse from the laser source corresponds to one scan-line, a small fraction of the periodic laser pulses is split from the source to generate the line-scan clock ($\texttt{CLK}_{\text{line}}$). The `Line-scan Clock Gen` block in Fig. 2 converts the laser pulses into electrical pulses using a photodetector, then the pulses are amplified and stretched to form the square-wave clock ($\texttt{CLK}_{\text{line}}$) with frequency f_{line}. This is the base reference clock that is used to derive all subsequent clocks for both the ADC sampling clock ($\texttt{CLK}_{\text{samp}}$) and FPGA internal clocks for CLSS logics generalized as $\texttt{CLK}_{\text{FPGA}}$.

One of the key component that governs the behavior of the CLSS algorithm is the `Frequency Synthesizer` block in Fig. 2 which generates $\texttt{CLK}_{\text{samp}}$ for the ADC. The Phase Lock Loop (PLL) inside the `Frequency Synthesizer` takes this task, since it allows clock frequency synthesis from an input reference frequency (f_{in}) to an output clock frequency (f_{out}), where f_{out} could be non-integer times of f_{in}. Thus, in CLSS system, the `Frequency Synthesizer` generates $\texttt{CLK}_{\text{samp}}$ with the $(\mathbf{p}, \mathbf{q})$ as the parameters setting and $\texttt{CLK}_{\text{line}}$ as the input reference clock. It is important to note that a PLL with low output clock jitter should be used to minimize spatial inaccuracy of pixels acquired by the ADC, which will in turn affect the accuracy of the shifted pixel pattern and the quality of CLSS image reconstruction.

B. Analogue front-end setup for ATOM system

In the optical front-end of the ATOM microscopy, the actual pulsed laser used is tuned to have a stable repetition rate of 12.08 MHz.A corresponding line-scan clock ($\texttt{CLK}_{\text{line}}$) with the same frequency is generated by a custom circuit consists of analogue comparators and clock buffers.

We invoke a high-speed ADC (EV8AQ160) [20] for pixel digitization, which has a maximum sampling frequency of 5 GHz. Based on the CLSS mechanism, we select the parameters $(\mathbf{p}, \mathbf{q}) = (1024, 3)$ such that f'_{samp} has a frequency of 4.123 GHz (12.08 MHz$\times$1024/3). A PLL based frequency synthesizer (Valon-5009) [21] is used to generate the desired $\texttt{CLK}_{\text{samp}}$ to drive and synchronize the ADC at the correct sampling frequency. With the above sampling settings, continuous lines of cell images with shifted sampling pattern for CLSS processing are obtained and the throughput of the pixel stream matches the sample rate at around 4.123 giga pixel per second.

C. Image Reconstruction with Shifted Sampling Grid

We propose three SR image reconstruction methods for the CLSS system: (a) *line-interleave*, (b) *simple-interpolation*, and (c) *full-interpolation*. Fig. 5 illustrates the three methods in detail. The source and reconstructed pixels are arranged in a regular isotropic sub-pixel grid for ease of visualization. The elements highlighted in orange represent raw pixel samples from the ADC that form each of the lines with a low horizontal resolution (LR-lines). For each method, sub-pixels

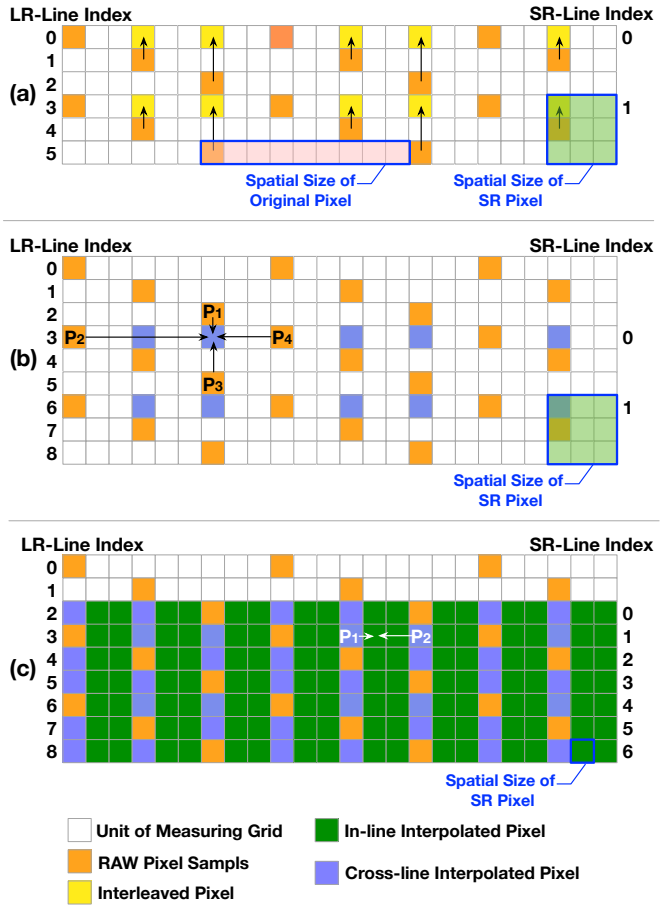

Fig. 5. Illustration of the three proposed SR-image reconstruction methods with the CLSS system. (a) Line-interleave; (b) simple-interpolation; (c) full-interpolation. The source input to the CLSS system containing "low-resolution lines" are denoted as LR-Line, and the reconstructed super-resolution lines are denoted as SR-Line. The elements highlighted in orange represents the raw pixel samples within each input SR-line.

are obtained and reconstructed into super-resolution lines (SR-lines). The reconstructed sub-pixels are highlighted in different colors as shown in Fig. 5(a-c) depending on the method used. Note that for methods (a) and (b), the final 2-D image from the stacked SR-Lines will have improved horizontal resolution but reduced vertical resolution.

1) Line-interleave: This method is the most light-weighted scheme and can be achieved without any arithmetic operations. As shown by the example in Fig. 5(a), the source input pixels from every 3 lines ($\mathbf{q} = 3$) are combined with regular interleaving arrangement to form a single line with $3\times$ the horizontal resolution. This scheme is feasible in general as the pattern of the shifted sampling position repeats every $\mathbf{q}$ lines, and for each cycle of pattern ($\mathbf{q}$ lines), one super-resolution line (SR-Line) is produced. Therefore, in the ATOM example, the line-interleave method reduces the line count by $3\times$ while increasing the horizontal pixel count by $3\times$. Although this scheme alters the image aspect ratio from $9:1$ (LR-lines) to $1:1$ (SR-lines), the resultant image is isotropic and hence allows both vertical and horizontal image features to be resolved equally.

2) Simple-interpolation: While the basic line-interleave method is simple to implement, it relies on the assumption that small vertical traversal across the actual image causes negligible changes in pixel value. In reality, this introduces pixel errors that could manifest as image artifacts, and the magnitude of error depends on the source line frequency and $\mathbf{q}$. To minimize pixel error in the output SR image, we have implemented a bilinear-interpolation method for CLSS that can more accurately estimate the pixel values at the actual vertical position. As Fig. 5(b) shows, the sub-pixel is calculated as the weighted mean of the four surrounding pixels in the vertical and horizontal directions. The weight (w_n) of each referenced pixel is inversely proportional to its spatial distance to the sub-pixel. In the ATOM cell microscopy case, we set $w_n = \{\frac{6}{12}, \frac{1}{12}, \frac{3}{12}, \frac{2}{12}\}$. Like the line-interleave case, the number of lines is reduced by $3\times$, and the horizontal pixel count is increased by $3\times$, resulting in an isotropic image with $1:1$ aspect ratio.

3) Full Interpolation: For the scenarios that need the best possible resolution in both vertical and horizontal directions, we present the *full-interpolation* scheme. Instead of composing each SR-line with $\mathbf{q}$ LR-lines as in the previous two schemes, every single sub-pixels along the original LR-lines are reconstructed to form a corresponding SR-line. The sub-pixels shown in Fig. 5(c) are computed in two interpolation stages. First, *cross-line* sub-pixels (highlighted in purple) are computed using four surrounding pixels in the vertical and horizontal directions as in method (b). Then, the remaining sub-pixels (highlighted in green) are computed *in-line* by interpolating two horizontal neighboring pixels consisting of two *cross-line* sub-pixels, or a mix of *cross-line* sub-pixel and source pixel sample. Since the entire isotropic sub-pixel grid is filled, the reconstructed image is also isotropic, and the overall resolution has increased $9\times$ in the horizontal direction compared to the source image with LR-lines.

IV. Super-resolution Hardware on FPGA

High-throughput CLSS image reconstruction is essential in maintaining real-time image acquisition and analysis in ultra-fast cell microscopy. For example, the ADC in the ATOM system continuously digitizes analogue signals into 8-bit pixels at rates beyond 4 GB/s. At such throughputs, a general-purpose computer is no longer a viable option for complex real-time image processing on the incoming pixel stream such as CLSS. FPGA, on the other hand, is perfectly suited to such high-throughput processing scenario, owing to its high-speed I/O and abundant programmable logic. The main challenge with real-time processing on FPGA is its relatively limited internal clock frequency that is incapable of matching the ADC sampling frequency directly. However, high-speed I/Os on FPGAs are usually equipped with hardware deserializer which can transform serial pixel stream into blocks of parallel pixel data (Fig. 2). This allows batches of pixels to be processed in parallel at lower clock frequencies while maintaining the same overall throughput as the incoming pixel stream. The actual frequency (f_{FPGA}) of the FPGA internal clock (CLK_{FPGA}) depends on the number of pixels parallelized within each data

block (denoted as N_{blk}). For example, the FPGA deserializer in the ATOM system is configured to collect 16 parallel pixels per block ($N_{blk} = 16$). Therefore, f_{FPGA} is given by $4.123\,\text{GHz} \div 16 = 257.7\,\text{MHz}$.

A. Multiline Pixel Storage

In the CLSS system, the FPGA super-resolution module plays a central role in transforming the shifted sampling grid image into SR image with isotropic 2-D pixels. However, for the process to work effectively, the source pixel data must first be arranged and stored in a way that allows parallel read/write patterns required by the CLSS SR methods. For line-interleave and the interpolation methods, multiple LR-lines are accessed in parallel to construct one SR-line. As demonstrated in Fig. 5, three LR-lines ($\mathbf{q} = 3$) are needed for each SR-line in the line-interleave method, and five LR-lines ($2 \times \mathbf{q} - 1 = 5$) are accessed to construct one SR-line in both interpolation methods. Since the sequential LR-lines are delivered to FPGA in stream, an on-chip buffer is set to dynamically store multiple LR-lines for the 2-D stencil access of SR-pixel construction.

Register (REG) and Block RAM (BRAM) are the two major FPGA on-chip storage components that can be used. Since composing a complex 2-D buffer with massive REGs usually results in poor timing, BRAM is employed for the buffer of CLSS, that operates in a frequency (f_{FPGA}) of 257.7 MHz. However, BRAM comes at the cost of reduced flexibility in terms of data access parallelism, where only two read/write ports are available for independent addressing and data accessing. The generic multiline buffer design that constructs each line buffer with a BRAM is not suitable for CLSS case. Since the pixel number of an LR-line may not be a multiple of N_{blk}, the last pixel block of one LR-line may encapsulate pixels belong to the subsequent line. These pixels cause the block access misaligned to the BRAM boundary and result in the data-coverage risk between two contiguous blocks. A line-buffer design method, Stream Windowing on Interleaved Memory (SWIM), is proposed in [22], that circumvents the BRAM-misalignment issue via constructing one line buffer with multiple BRAMs in a specific width. CLSS leverages the SWIM method during the buffer design that fulfills the continuous pixel block access in the SR-line construction.

In the case of CLSS with parameters $(\mathbf{p}, \mathbf{q}) = (1024, 3)$ and $N_{blk} = 16$ for the ATOM microscopy, the LR-line has a periodic sampling pattern that repeats every three lines, and the lines of a period contain $\{342, 341, 341\}$ pixels respectively. In Fig. 6, we propose the multiline buffer design and data arrangement for this case based on the SWIM method. Each line buffer is composed of multiple physical BRAM partitions. Each partition is labeled with $\mathtt{B}_{\mathtt{n,b}}\langle\mathtt{Addr}\rangle$, where `n`, `b`, and `Addr` denote the line index, partition index and data address respectively. The width of each BRAM is denoted as $W_{n,b}$ and labeled under each partition.Since the line width is not a multiple of N_{blk} (16), the last input block of each line may exceed the total line length. The remaining partial block contains pixels that belong to the start of the next line, and they are denoted as $r_n(m)$ in Fig. 6, where m is the width in terms of number of pixels. The content of $r_n(m)$ is written directly into the first BRAM partition in the next line ($\mathtt{B}_{\mathtt{n+1,0}}\langle\mathtt{0}\rangle$), that is set to the same width ($W_{n+1,0}$). Thus, the subsequent block is separated and written into $\mathtt{B}_{\mathtt{n+1,1}}\langle\mathtt{0}\rangle$ and $\mathtt{B}_{\mathtt{n+1,0}}\langle\mathtt{1}\rangle$ without BRAM-misalignment issue.

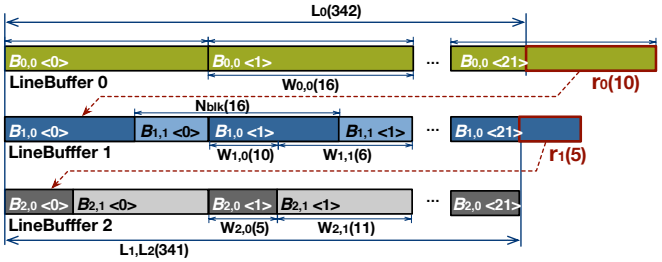

Fig. 6. SWIM BRAM partition scheme for CLSS in ATOM microscopy, with the parameters set to $(\mathbf{p}, \mathbf{q}) = (1024, 3)$ and $N_{blk} = 16$. Each BRAM Partition is labeled as $\mathtt{B}_{\mathtt{n,b}}\langle\mathtt{Addr}\rangle$, where `n`, `b`, and `Addr` denote the line index, partition index and data address respectively.

In this particular case, $\mathbf{p}$ (1024) is set to be divisible by N_{blk} (16) such that at the end of every $\mathbf{q}$ (3) lines, the whole block aligns exactly to the end of the line with no remainder. Therefore, the subsequent pixel block will start a new round with the same storage pattern.

B. Image Reconstruction with Line-interleave

There are two design requirements for the CLSS line-interleave hardware module: First, the module must be capable of receiving pixel blocks continuously and performing interleave simultaneously without stall cycles; Second, the design should have low output latency such that real-time applications, including cell sorting, that require rapid decision making in a cell-by-cell basis is feasible. A reasonable design scheme that could satisfy both requirements is *double buffering*, in which two line-buffer groups are employed and each has a capacity of $\mathbf{q}$ LR-lines. During the reconstruction process, the two groups alternate between read and write operations. Therefore, while one group is outputting results from the previous round, the other group can be filled with new data for the next round, and vice-versa. This allows continuous interleave operation while having a latency time required to fill one group with $\mathbf{q}$ LR-lines completely.

Based on the double-buffering, the hardware working scheme for line-interleave SR module is presented in Fig. 7. Fig. 7(a) shows that two buffer groups (`BufferGroup0` and `BufferGroup1`) receive the LR lines and output SR line alternatively. Each of the two buffer groups adopts the SWIM BRAM partition scheme demonstrated in Fig. 6 and is capable of processing pixel block seamlessly. Note the output latency is labeled on the diagram that is equivalent to the period of $\mathbf{q}(3)$ LR lines. The specific memory transaction of pixel data in and out of the multiline buffer is illustrated in Fig. 7(b). During the write cycle of one particular Line-buffer group (`BufferGroup1`), it stores one input block ($\mathtt{BLK}_{\mathtt{in}}$) for each clock cycle while 2-D pixel array from the other filled Line-buffer group (`BufferGroup0`) is being fetched in parallel to an output buffer for constructing the line-interleaved output pattern (`OutBlk0,1,2`). After `BufferGroup1` is filled

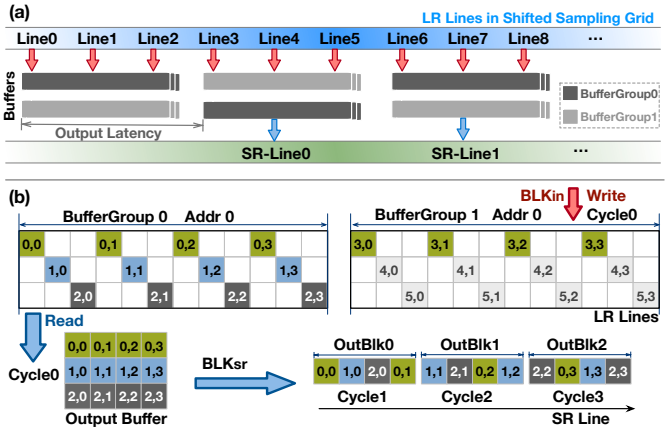

Fig. 7. (a) shows the working scheme of double buffering in the ATOM CLSS system. (b) shows an example of memory transaction of pixel data in and out of the buffer groups. The memory elements are highlighted in color, and the positions represent the shifted sampling grid. Each pixel element is labeled with its row (line) index and horizontal pixel index respectively.

Fig. 8. The rotating line access pattern of the line-rolling buffer.

with all three lines in `Cycle2`, the writing process switches over to `BufferGroup0` and the pixel fetch process begins on `BufferGroup1`. The detail composition of the hardware module is described in Appendix A.

C. Image Reconstruction with Interpolation

In the interpolation-based SR cases, the hardware module is composed of two phases corresponding to the multiline buffer and arithmetic unit. The first phase fetches *All Related Pixels* (ARP) for the sub-pixel computation in the current clock cycle, which contains N_{blk} pixels from each of the $(2 \times \mathbf{q} - 1)$ lines. The second phase performs the sub-pixel computation (SPC) with pixels obtained in the ARP-fetch phase. The ARP-fetch phase employs a multiline buffer that also adopts the SWIM BRAM organization described in Section IV-A. For the simple-interpolation case, *triple buffering* is adopted because $(2 \times \mathbf{q} - 1)$ LR-lines are accessed concurrently instead of $\mathbf{q}$ lines in the line-interleave case. The triple buffering scheme uses three line-buffer groups and each of them stores $\mathbf{q}$ LR-lines. In each round of data access, pixel blocks with $(2\times\mathbf{q}-1)$ LR-lines are read out from two buffer groups in parallel, and the incoming block is stored in the remaining one group. After each round, the role of the buffer-groups rotates in a round-robin manner, similar to the line-interleave case with double-buffering.

The full-interpolation scheme is different from simple-interpolation as each LR-line is interpolated into one SR-line in the full-interpolation case. Furthermore, $(2\times\mathbf{q}-2)$ LR-lines are reused during interpolation of the two successive lines. As shown in Fig. 5(c), in the full-interpolation scheme for ATOM microscopy, four LR-lines are reused in two successive SR-lines computation. To enable such access pattern, a *Line-rolling buffer* is developed for the ARP-fetch phase hardware, that is composed of $2 \times \mathbf{q}$ line-buffers. As demonstrated in Fig. 8, the line-rolling workflow contains continuous rounds (Fig. 8(a)-(c) presents the behavior of one cycle in each of `Round0-2` respectively). In each round, the buffer receives one LR-line and constructs an SR-line along with the arithmetic units. Thus, $(2 \times \mathbf{q} - 1)$ line-buffers perform read access, and the rest one stores the input block. For example, in `Round0` (Fig. 8(a)), LR-line 5 is written into `LineBuffer 5`, while the previous five LR-lines are read from `LineBuffer 0-4` to interpolate the pixels in SR-line 2. In `Round1` (Fig. 8(b)), the store location of the incoming LR-line 6 wraps-around to the top, and the LR-line is written into `LineBuffer 0` that replaces the no longer needed LR-line 0. Concurrently, pixel blocks are read from `LineBuffer 1-5` to interpolate SR-line 3. The subsequent `Round2` (Fig. 8(c)) follows the same rolling behavior where read and write access lines continue to rotate. Such line-rolling buffer is capable of performing seamless pixel store and fetch for the full-interpolation case.

Fig. 9 shows the workflow of the pipelined full-interpolation hardware with the simplified parameters $(\mathbf{p}, \mathbf{q}) = (32, 3)$ and $N_{blk} = 4$. In Fig. 9(a), the colorful squares represent the memory element in the line-rolling buffer, and the spatial sampling position of each pixel is indicated by the shifted grid. In the present cycle, $\mathtt{BLK}_{\mathrm{in}}$ of LR-line 7 is written into `LineBuffer 1` and it replaces the pixels of LR-line 1 stored in it. Note that although three pixels of $\mathtt{BLK}_{\mathrm{in}}$ are written to `Addr0` of `LineBuffer 1`, it does not wipe the pixel (7,0) stored at `Addr0` in the previous cycle. This is because the `LineBuffer 1` adopts the SWIM method and is composed of two BRAMs with different widths (distinguished by dark/light color in Fig. 9(a)) that exactly fit the misaligned storage pattern. A similar situation also occurs on the other line buffers. In the same cycle, all related pixels for interpolating the SR block ($\mathtt{BLK}_{\mathrm{sr}}$) of SR-line 4 are read from the line-rolling buffer. The ARP-fetch phase is further pipelined into three stages. In stage(1), each line-buffer outputs eight $(2 \times N_{blk})$ pixels from two consecutive addresses. Then in stage(2) the hardware selects the proper pixels from each line-buffer output that are used in full-interpolation. In stage(3) the line-reordering is performed to transform the line-buffer output sequence to the spatial sequence in an image. Subsequently, the reordered pixel-blocks are delivered to the arithmetic units for *Sub-Pixel Computation* (SPC). As Fig. 9(b) shows, the hardware of SPC phase is pipelined into two stages that corresponding to cross-line and in-line interpolation respectively. The underlying hardware design for full-interpolation SR module is presented in Appendix B.

V. Experiment and Evaluation

A. System Integration

As Fig. 10 presents, the experimental CLSS system was implemented and integrated into the ATOM microscopy [12]. In the ATOM system (Fig. 10(a)), the cell microfluidics flow

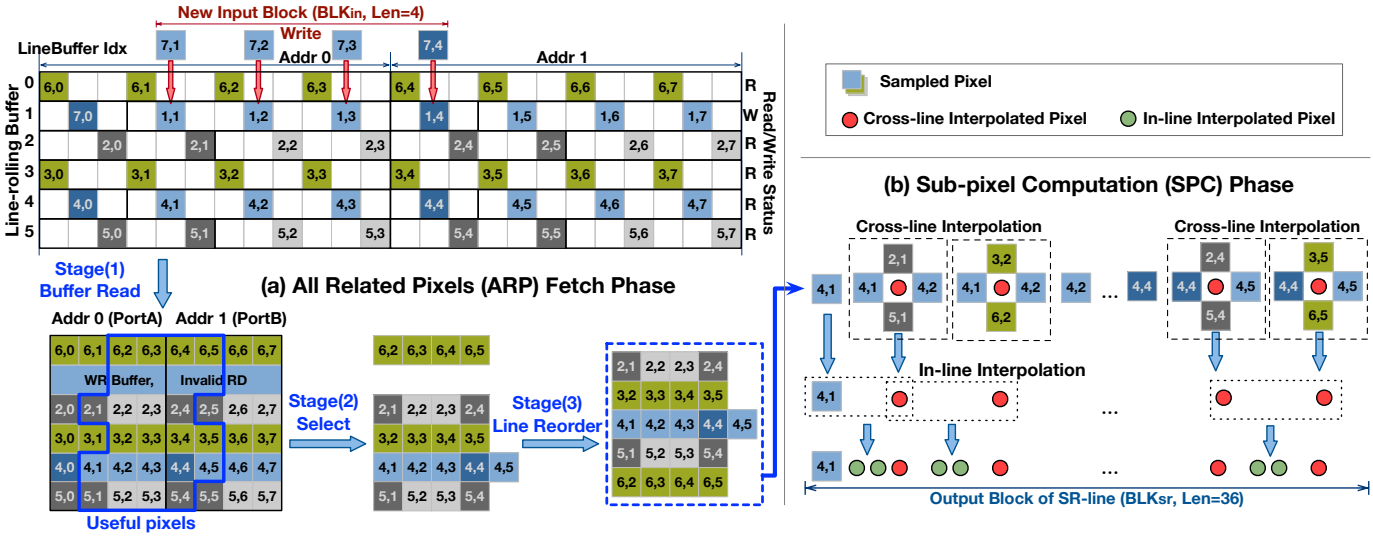

Fig. 9. An example to demonstrate the hardware interpolation pipeline. The ARP-fetch phase output the pixel blocks that are required by SR in the current cycle. The SPC phase computes the sub-pixel value according to the interpolation pattern described earlier in Fig. 5. (a) shows the hardware workflow of ARP-fetch phase that is pipelined to three stages. (b) shows SPC phase hardware workflow that contains the computation steps for cross-line and in-line interpolation.

pass the pipe channel in a stable flow rate. The cells are line-scanned by the periodically pulsed laser (Fig. 10(b)) and then transformed to analog electronics signal by the photodetector. The ADC module (EV8AQ160) is connected to the photodetector; it digitizes the analog signal to the pixel stream as the input of the digital system (Fig. 10(d)). Following the CLSS sampling scheme, a frequency synthesizer (Valon-5009, Fig. 10(c)) is employed to provide the synchronized sampling clock ($\mathtt{CLK}_{\mathrm{samp}}$) for the ADC and FPGA. The frequency synthesizer takes in the reference clock ($\mathtt{CLK}_{\mathrm{line}}$) that inherited from the line-scan laser pulse and outputs the $\mathtt{CLK}_{\mathrm{samp}}$ with a frequency of $1024/3$ times that of $\mathtt{CLK}_{\mathrm{line}}$. The SR image reconstruction module in the CLSS system was implemented on the ROACH-2 platform [23] that receives the pixel stream in shifted sampling grid and constructs them to SR-lines. The FPGA (Virtex-6 XC6VSX475T) on ROACH-2 is programmed with the CLSS SR hardware, along with the communication module that delivers the SR-lines to the host server via a high-speed network.

B. Imaging Quality

To compare and contrast the imaging quality enhancement of the CLSS schemes, Fig. 11 shows a detailed comparison of the cell images obtained in ATOM microscopy with and without CLSS. To enable comparison of fine details within each cell image, a consistent scaling and cropping is applied to all images around the cell region of interest. Cells of three cancer types, oesophageal cancer (OAC), oligosaccharyltransferase (OST) and leukemic monocyte THP-1 are used in the imaging experiments and one comparison group for each cell type is provided in Fig. 11. Each group contains five images (Column(1)-(5)) obtained from different sampling grids and SR-reconstruction methods, as labeled by the headers of each column.

Images in Column(1) are sampled in rectilinear grid with a frequency of 4.131 GHz, which is $342\times$ (integer multiple) the $\mathtt{CLK}_{\mathrm{line}}$ frequency. The LR-lines with traditional pixel sampling are directly stacked to form non-isotropic cell images with a resolution of 150×17. Images in Column(2) are generated by applying $9\times$ bilinear interpolation in the horizontal direction to the base images in Column(1). Column(3)-Column(5) are reconstructed based on the CLSS shifted sample grid, where the image lines are sampled in a frequency of 4.123 GHz, that is $(1024/3)\times$ (non-integer multiple) the frequency of $\mathtt{CLK}_{\mathrm{line}}$. Based on the shifted sampling grid, the SR-lines are reconstructed using the three methods described in Section III-C. Apparently, the SR-images obtained from the CLSS system reveals an overall increase of texture details than the original images with rectilinear grid (Column(1) and (2)). This observation highlights the effectiveness of CLSS scheme that compensates the under-sampled horizontal pixels with the over-sampled vertical pixels. For images in Column(3) obtained by the line-interleave SR method, we can observe artifacts in the high-contrast horizontal edge area (such as cytoderm and nucleus of the cells), due to the error caused by spatial pixel shift. In column(4), images are reconstructed using the improved simple-interpolation SR method. The interpolated images have the same resolution (50×50) as in Column(3), however, the interpolation better preserves the spatial accuracy of pixel values and hence eliminated image artifacts in areas with high vertical contrast. Column(5) is obtained via full-interpolation, as such the resolution is 150×150. Texture details in this case are improved over the simple-interpolation case as the resolution has further increased. However, due to the resolution increase, the effect of the non-uniform cross-line interpolation pattern (Fig. 5(b)) becomes more apparent. Therefore, slight periodic variation in blurriness of vertical edge is visible. Comparing Column(5) to Column(2) that have the same resolution, the CLSS scheme apparently improves image details in the horizontal direction, whereas the interpolation in Column(2) creates excessive horizontal blurring that has no benefit to the overall image quality.

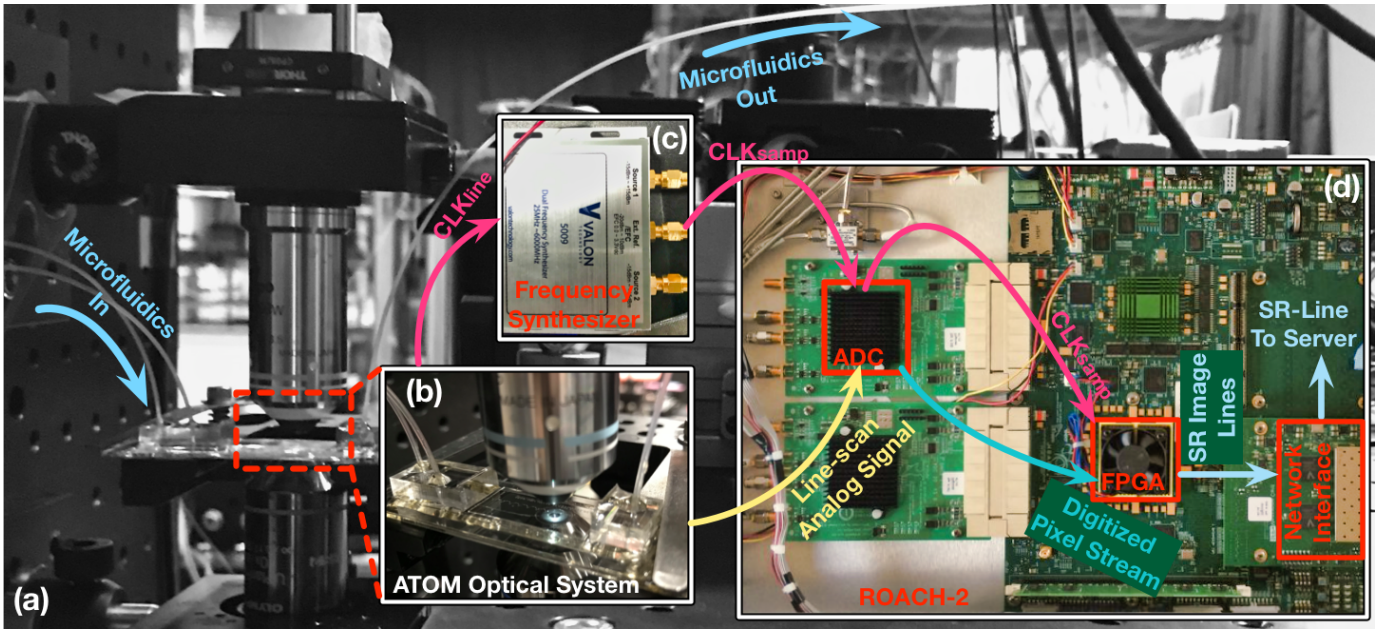

Fig. 10. Experiment setup of the CLSS system for ATOM microscopy corresponding to the diagram in Fig. 2. (a) shows the microfluidics flow in ATOM; (b) shows the line-scan optical system; (c) shows the frequency synthesizer for the coprime f'_{samp} generation; (d) shows the ADC module and FPGA-centric SR reconstruction system.

TABLE I
SUMMARIZED PARAMETERS AND QUALITY FOR DIFFERENT IMAGING METHODS ON ATOM.

Imaging Method	Column of Fig. 11	Sampling Method	Sampling Frequency	Spatial Resolution (H, V μm)	Isotropic	Imaging Quality
Original Sampling	(1)	Rectilinear	4.131 GHz	3.20, 0.36	No	Apparent artifacts
Original Interpolation	(2)	Rectilinear	4.131 GHz	0.36, 0.36	Yes	Smoother than Column(1), but blurred
Line-Interleave	(3)	CLSS	4.123 GHz	1.06, 1.08	Yes	Reveal texture detail; Artifacts exist
Simple-Interpolation	(4)	CLSS	4.123 GHz	1.06, 1.08	Yes	Sharp; Less artifacts than Column(3)
Full-Interpolation	(5)	CLSS	4.123 GHz	0.36, 0.36	Yes	Highest resolution; Sharp; Slight artifacts

Table I summarizes the property and quality of each imaging method for an intuitional comparison. The spatial resolution in the experiments is given, and the image is isotropic if the resolution in horizontal and vertical (h/v) axes are identical. With both the line-interleave and interpolation SR methods in the CLSS, the resultant SR images are isotropic due to the proper $\mathbf{q}$ value we selected considering the ADC maximal frequency and the flow rate of microfluidics that determine the spatial h/v resolution respectively. Note that the shifted sampling grid of CLSS increases the sampling density in the horizontal axis that benefits to revealing higher frequency information than the rectilinear grid. Nevertheless, the subsequent SR-interpolation cannot capture the missing part of the high-frequency feature from the original cells. Thus, we consider the effective resolution improvement in CLSS scheme as $\mathbf{q}$ times in the horizontal axis that is undersampled in ATOM-like line-scan microscopes.

C. SR Module Hardware Evaluation

The imaging throughput of the CLSS system depends on the timing performance and efficiency of the SR module on the FPGA. Thus we evaluated the SR hardware in terms of resource usage, maximal operating frequency (f_{max}) and the output latency, which have been carefully considered during the SR module design.

To reflect the performance of the proposed design, we also implemented the SR module with the naive RTL as the baseline for comparison. In the baseline design, the SR function is described using the high-level behavioral RTL that let the FPGA synthesis tool generate the design automatically. In contrast, the proposed SR module is implemented with the low-level description of the underlying FPGA component, that directs the synthesis tool to generate the pre-designed hardware. The post-place-and-route results of both designs are given by the Xilinx ISE synthesis tool and listed in Table II.

TABLE II
HARDWARE IMPLEMENTATION RESULTS OF SR MODULE

Interpolation Method	SR Reconstruction Method	LUT	REG	BRAM	DSP	f_{max} (MHz)	Latency (cycle)
This work	Line-interleave	1149	1156	14	0	400	66
	Simple-interpolation	2551	4913	21	128	361	71
	Full-interpolation	5000	6591	26	320	361	74
Naive RTL (baseline)	Line-interleave	18245	16606	0	0	260	66
	Simple-interpolation	40623	27135	0	128	183	71
	Full-interpolation	33743	20378	0	320	105	74

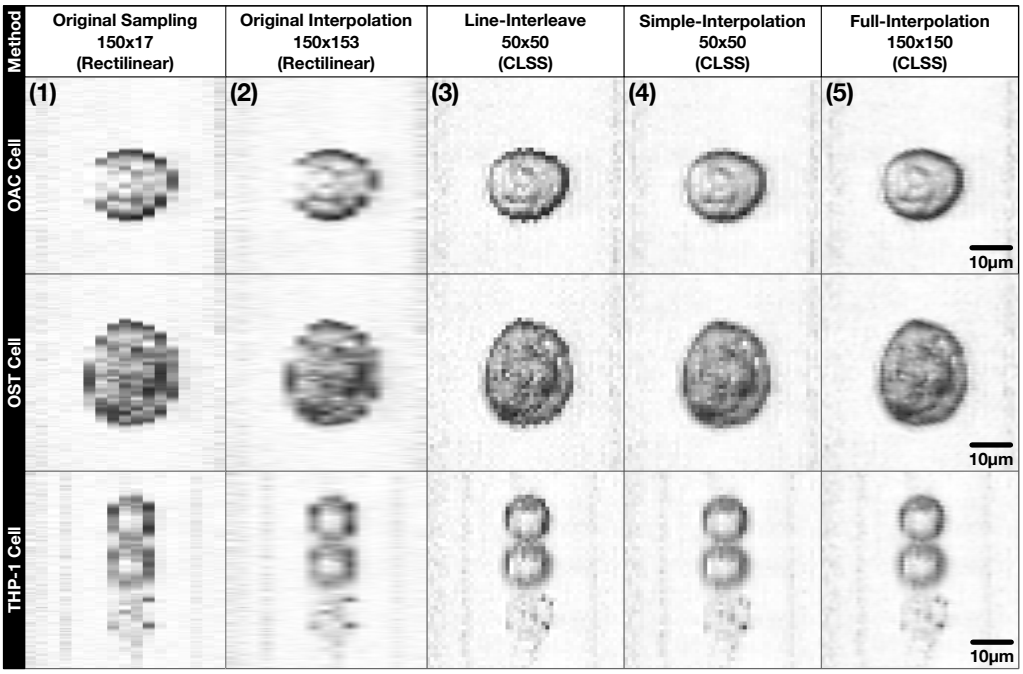

Fig. 11. Cell images of three cancer types obtained in ATOM microscopy with different imaging methods. Column(1) shows images obtained using the original rectilinear grid without CLSS sample shift, where the vertical resolution is approximate $9\times$ of the horizontal resolution; Column(2) is the images with horizontal interpolation based on Column(1) images to extend the horizontal pixel count by $9\times$; Column(3)-(5) are SR images obtained with the CLSS schemes proposed in this paper with line-interleave, simple-interpolation and full-interpolation respectively. More cell images are available online (`http://www.eee.hku.hk/%7Erbshi/sr_res.html`) for further reference.

1) Resource usage: The SR logic circuit is constructed using the following FPGA building blocks: Look-up table (LUT), Register (REG), BRAM and DSP. Usage of these resources can reveal the efficiency of the hardware design method. For the line-interleave SR, the usage of LUT and REG in the baseline design is $15.9\times$ and $14.4\times$ the usage of our design respectively. The ratio is $15.9\times$, $5.5\times$ in the simple-interpolation case and $6.7\times$, $3.1\times$ in the full-interpolation case. This is because we adopted partitioned BRAMs and pre-designed logic to compose the line buffer, whereas the baseline method invoked massive REGs as the storage component that led to a complex controller logic in realizing the buffer behaviors. Comparing the logic resource usage of three SR methods, the LUT usage in simple-interpolation case is $2.2\times$ of that in line-interleave hardware, because the triple-buffering scheme consumes more logic resource on the `GroupMUX` than the double-buffering. The line-rolling buffer in full-interpolation consumes the most LUTs due to its complex `SplitMUX` and `LineReorderMUX`. In addition, massive REGs are consumed in the fully-pipelined SPC phase hardware, thus the interpolation-based methods consume more REGs than the line-interleave case. The arithmetic units in SPC hardware also costs the DSP resource on the FPGA. In terms of BRAM usage, the full-interpolation case uses the same number of line buffer as the line-interleave case, but its BRAM usage is approximately twice as much. This is because line-interleave adopts the *simple dual port* (SDP) BRAM, which consumes only half a unit of BRAM resource on the Xilinx Virtex-6 architecture. On the other hand, the full-interpolation case uses *true dual port* (TDP) BRAM which consumes one complete unit of BRAM hardware.

2) Operating Frequency and Throughput: Given the sampling frequency of 4.123 GSPS in the ATOM system, the achievable f_{max} of the internal FPGA clock should be at least 257.7 MHz (f_{samp}/N_{blk}) to sustain the data throughput. The f_{max} of FPGA design is estimated using the timing models for Virtex-6 within Xilinx ISE. In the naive RTL implementation, complex and long routings among distributed components (REGs and LUTs) lead to a much lower f_{max}. In contrast, our method relies on dedicated BRAM hardware with optimal timing performance and high f_{max} optimized at transistor-level of the FPGA. For example, by comparing our full-interpolation hardware with the baseline design, the automatic synthesis with naive RTL results in a 130 MHz f_{max}, and it violates the 257.7 MHz minimum requirement. Our scheme meets the required f_{max} at 361 MHz which is bounded by the f_{max} of the DSP in the SPC hardware. In terms of throughput, the CLSS system is capable of processing 12 million LR-lines per second (≈342 pixels/line) with a sampling speed of 4.123 GSPS throughput. Assuming a normal sized cell covers approximately 100 LR-lines as Fig. 11 shows. The experimental CLSS system is capable of seamlessly processing around 120 000 cells per second.

3) Latency: The processing latency is defined as the time interval between the raw LR-line input and the SR-line output. In the proposed CLSS system, latency in terms of clock-cycle ($T_{latency}$) can be deterministically calculated using the imaging parameters:

$$T_{latency} = \mathbf{p}/N_{blk} + d_{post} + d_{spc} \tag{2}$$

The first fractional term ($\mathbf{p}/N_{blk}$) represents the latency of the multiline buffer. d_{post} is the pipeline latency in the post-read data processing (e.g., line-reordering), which is 2 cycles in the line-interleave and simple-interpolation case, and 4 cycles in the ARP-fetch stage for the full-interpolation case. For the interpolation-based SR hardware, there is an extra latency term (d_{spc}) introduced by the SPC-phase. The values of d_{spc} for simple-interpolation and full-interpolation are 5 cycles and 8 cycles respectively. Given the FPGA operating frequency of 257.7 MHz for the ATOM-CLSS system, the clock latency translates to a time latency of 0.29 µs.

D. Comparison with Related Works

A wide spectrum of techniques has been developed by researchers to enhance image resolutions, ranging from computational methods that rely on deep learning[28], [29] to advanced sub-diffraction-limit optical imaging techniques such as STED[25], STORM[26], or PALM[27]. This work addresses the limitation arising from the underlying image formation electronic systems by carefully adjusting the sampling period to produce a shifted sampling grid. When compared to previous frame-based super-resolution systems[28], [29], CLSS relies on only a few lines of image input for resolution enhancement. This results in substantial reduction in super-resolution processing latency, while the optimized streaming hardware design allows real-time imaging throughput at 3931 frame per second.

VI. Conclusions

In this paper, we propose CLSS, a coprime line scan super-resolution scheme for ultra-fast microscopy. With CLSS, shifted sampling grid is obtained that facilitates the reconstruction of texture detail with the tailored image super-resolution approaches. The hardware design method is presented that covers the entire CLSS imaging flow and achieves an ultra-high throughput and real-time image reconstruction. Applied to the ATOM microscopy, our CLSS design is capable of performing image super-resolution with a throughput of 120 000 cells per second and improved the horizontal resolution by at least 3× over the baseline system. Furthermore, the whole CLSS system is parametrizable and hence our scheme is applicable to a wide range of line-scan based microscopy systems.

VII. Acknowledgement

This work was supported in part by funding from the Research Grants Council (RGC) of Hong Kong (CRF C7047-16G, GRF 17245716, GRF 17203217, GRF 17259316, GRF 17209017, GRF 17208918), the Innovation and Technology Support Programme (Tier 3) (ITS/204/18) and the Croucher Innovation Award.

Appendix A
Hardware of Line-interleave SR Module

Fig. 12 presents the hardware components for line-interleave SR module, where (a) shows the buffer structure and (b) shows the detail of the control logic. Corresponding to the workflow in Fig. 7, the `GroupMUX` selects the data read from the buffer groups in interleave and then sends the data to the `OutputBuffer` registers. The `PatternMUX` composes the line-interleaved pixel blocks ($\mathtt{BLK}_{\mathrm{sr}}$) in the subsequent $\mathbf{q}$ clock cycles. The control logic in Fig. 12(b) governs the control signals for buffer and multiplexers. The essential part of the control logic is the `InputBlockCounter` that counts from 0 to ($\mathbf{p}/N_{blk} - 1$) repeatedly and the value `CNT` increases by one for each input block ($\mathtt{BLK}_{\mathrm{in}}$). The `CNT` is then sent to the subsequent logic to generate the control signals. The `Buffer Signal Generator` is referenced from the SWIM work [22] that generates the signal for each BRAM partition in the multiline buffer. The `Toggle REG` is a one-bit register that toggles when `CNT` returns to zero. It serves as the control signal of the `GroupMUX`. The control signal of `PatternMUX` is generated via a modulo operation on `CNT` and $\mathbf{q}$.

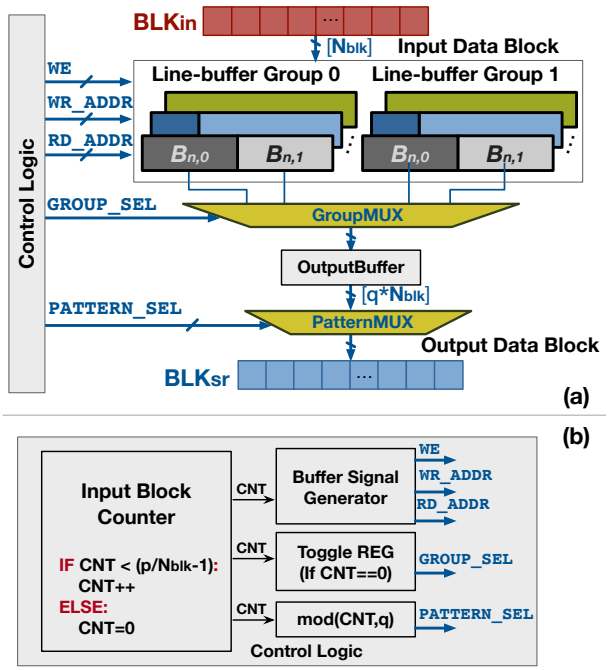

Fig. 12. (a) shows the line-interleave hardware components. (b) shows the detail of the control logic.

Appendix B
Hardware of Full-interpolation SR Module

Fig. 13 presents the full-interpolation hardware SR module, where (a) shows the line-rolling buffer for ARP-fetch phase and (b), (c) shows the SPC phase hardware for cross-line and in-line interpolation respectively. As Fig. 9(a) shows, the line-rolling buffer receives the continuous input pixel-blocks and outputs 2-D blocks for interpolation. The `SplitMUX` is connected to the dual ports of all BRAM partitions in the line-rolling buffer and selects the proper pixels in each line for interpolation, as the Stage(2) in Fig. 9(a). The `LineReorderMUX` performs the Stage(3) behavior that adjusts the line order in the 2-D pixel array and sends them to the SPC phase hardware.

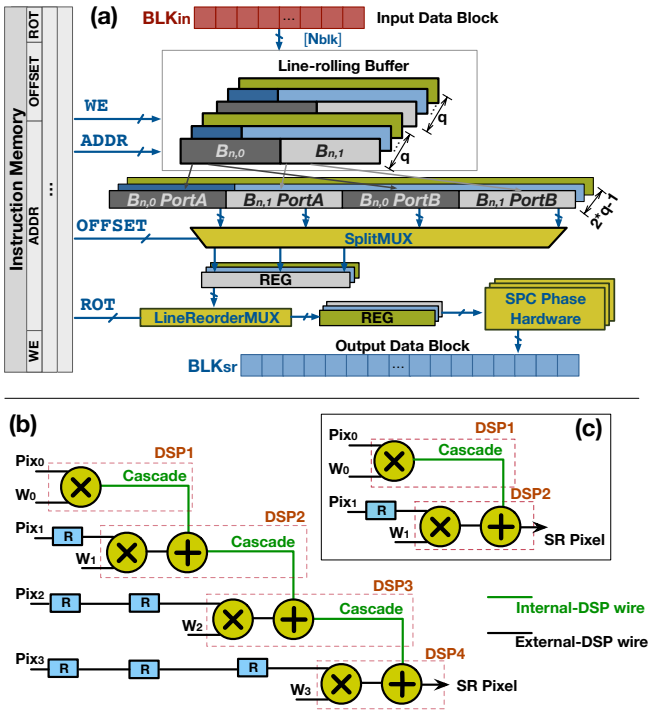

Fig. 13. Full-interpolation hardware components. (a) shows the line-rolling buffer with specific instruction control method; (b) and (c) show the DSP-based arithmetic unit for cross-line and in-line interpolation respectively.

The line-rolling behavior adds complexity to the control-signal generation, especially the memory-access signals for multiple BRAMs. In addition, using control logic as run-time signal generator brings significant hardware overhead. Therefore, an `Instruction Memory` is used to provide the control signals, so that the instruction for a control period can be generated offline and loaded into the memory during power-up. The dedicated instruction memory contains 4 subsets of control signals, as shown in Fig. 13(a). Write enable (`WE`) section includes a one-bit signal for each BRAM partition that indicates whether to store the incoming pixel in the present cycle. Address (`ADDR`) section is connected to the two ports of each BRAM to control the memory access position. The `OFFSET` section indicates a pixel-wise offset for each `SplitMUX` to select the correct set of N_{blk} pixels from $2\times N_{blk}$ output pixels of a line buffer. Rotation (`ROT`) section provides the control signal for the `LineReorderMUX` that constructs the 2-D block with a proper line-order. Since the data access pattern recurs during continuous processing, the instructions are periodically executed every $(2 \times \mathbf{q}/N_{blk})$ cycles.

The SPC phase hardware in Fig. 13(b)(c) performs two interpolation steps (cross-line and in-line interpolation) as described earlier in Section III-C3. Both interpolation methods compute the weighted-mean values, that can be mapped to the *multiply-accumulate* (MACC) arithmetic unit. We leveraged the embedded *digital signal processor* (DSP) units on FPGA for MACC operations instead of using inefficient look-up tables (LUTs) based implementation. Besides, we took advantage of the primitive-level programming [30] to enable the dedicated cascade connections between the DSPs as shown in Fig. 13(b)(c). This provides the best timing performance via dedicated inter-DSP routing connections to transmit the partial result for accumulation. With the above optimizations, only additional registers are consumed to construct the pipeline stages.

REFERENCES

[1] J. C. Caicedo *et al.*," *Nature Methods*, vol. 14, pp. 849–863, Aug 2017.
[2] A. K. S. Lau, H. C. Shum, K. K. Y. Wong, and K. K. Tsia, "Optofluidic time-stretch imaging – an emerging tool for high-throughput imaging flow cytometry," *Lab Chip*, vol. 16, pp. 1743–1756, 2016.
[3] H. Mikami *et al.*, "High-speed imaging meets single-cell analysis," *Chem*, vol. 4, no. 10, pp. 2278–2300, 2018.
[4] N. Meng, E. Lam, K. K. M. Tsia, and H. K.-H. So, "Large-scale multi-class image-based cell classification with deep learning," *IEEE journal of biomedical and health informatics*, 2018.
[5] N. Nitta *et al.*," *Cell*, vol. 175, pp. 266–276.e13, Sep 2018.
[6] A. H. Tang *et al.*, "Microfluidic imaging flow cytometry by asymmetric-detection time-stretch optical microscopy (ATOM)," *JoVE (Journal of Visualized Experiments)*, no. 124, p. e55840, 2017.
[7] I. C. Macaulay, C. P. Ponting, and T. Voet, "Single-cell multiomics: Multiple measurements from single cells," *Trends in Genetics*, vol. 33, no. 2, pp. 155–168, 2017.
[8] F. Tang, K. Lao, and M. A. Surani, "Development and applications of single-cell transcriptome analysis," *Nature Methods*, vol. 8, pp. S6–S11, Mar 2011.
[9] T. Riss, R. Moravec, and A. Niles, "Selecting cell-based assays for drug discovery screening," *Cell Notes*, vol. 13, pp. 16–21, 2005.
[10] R. T. Krivacic *et al.*, "A rare-cell detector for cancer," *Proceedings of the National Academy of Sciences*, vol. 101, no. 29, pp. 10501–10504, 2004.
[11] A. K. Lau *et al.*, "High-throughput image-based single-cell analysis by ultrafast asymmetric-detection time-stretch optical microscopy," in *Bio-Optics: Design and Application*, pp. BW1A–4, Optical Society of America, 2015.
[12] T. T. Wong *et al.*, "Asymmetric-detection time-stretch optical microscopy (ATOM) for ultrafast high-contrast cellular imaging in flow," *Scientific Reports*, vol. 4, p. 3656, January 2014.
[13] Q. Yuan, L. Zhang, and H. Shen, "Multiframe super-resolution employing a spatially weighted total variation model," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 3, pp. 379–392, 2012.
[14] A. Chan, E. Y. Lam, and K. K. Tsia, "Pixel super-resolution in optical time-stretch microscopy using acousto-optic deflector," in *OSA Topical Meeting in Bio-Optics: Design and Application*, p. BW2A.7, April 2015.
[15] A. C. Chan, E. Y. Lam, and K. K. Tsia, "Pixel super-resolution of time-stretch imaging by an equivalent-time sampling concept," in *High-Speed Biomedical Imaging and Spectroscopy: Toward Big Data Instrumentation and Management*, vol. 9720 of *Proceedings of the SPIE*, p. 972004, February 2016.
[16] A. C. Chan, H.-C. Ng, S. C. Bogaraju, H. K.-H. So, E. Y. Lam, and K. K. Tsia, "All-passive pixel super-resolution of time-stretch imaging," *Scientific Reports*, vol. 7, p. 44608, March 2017.
[17] R. Shi, A. C. Chan, E. Y. Lam, and H. K.-H. So, "Image super-resolution for ultrafast optical time-stretch imaging," in *Congress of the International Commission for Optics*, pp. W1F–08, August 2017.
[18] C. W. Liu, A. Feizi, N. Sarhangnejad, G. Gulak, and R. Genov, "Superresolution line scan image sensor for multimodal microscopy," *IEEE transactions on biomedical circuits and systems*, no. 99, pp. 1–12, 2018.
[19] A. K. Lau *et al.*, "Interferometric time-stretch microscopy for ultrafast quantitative cellular and tissue imaging at 1 μm," *Journal of biomedical optics*, vol. 19, no. 7, p. 076001, 2014.
[20] e2v Technologies, *EV8AQ160 QUAD ADC Datasheet DS0846*.
[21] Valon Technology, *5009 Dual RF Frequency Synthesizer Module*.
[22] J. S. J. Wong, R. Shi, M. Wang, and H. K.-H. So, "Ultra-low latency continuous block-parallel stream windowing using FPGA on-chip memory," in *2017 International Conference on Field Programmable Technology (ICFPT)*, pp. 56–63, IEEE, Dec. 2017.
[23] Collaboration for Astronomy Signal Processing and Electronics Research, "Roach2: Reconfigurable Open Architecture Computing Hardware."
[24] S. W. Hell and J. Wichmann, "Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy," *Optics letters*, vol. 19, no. 11, pp. 780–782, 1994.

[25] V. Westphal, S. O. Rizzoli, M. A. Lauterbach, D. Kamin, R. Jahn, and S. W. Hell, "Video-rate far-field optical nanoscopy dissects synaptic vesicle movement," *Science*, vol. 320, no. 5873, pp. 246–249, 2008.
[26] M. J. Rust, M. Bates, and X. Zhuang, "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM)," *Nature methods*, vol. 3, no. 10, p. 793, 2006.
[27] E. Betzig *et al.*, "Imaging intracellular fluorescent proteins at nanometer resolution," *Science*, vol. 313, no. 5793, pp. 1642–1645, 2006.
[28] X. Liu *et al.*, "A microfluidic cytometer for complete blood count with a 3.2-megapixel, 1.1-μm-pitch super-resolution image sensor in 65-nm bsi cmos," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 4, pp. 794–803, 2017.
[29] X. Huang, J. Guo, X. Wang, M. Yan, Y. Kang, and H. Yu, "A contact-imaging based microfluidic cytometer with machine-learning for single-frame super-resolution processing," *PloS one*, vol. 9, no. 8, p. e104539, 2014.
[30] A. Ehliar, "Optimizing xilinx designs through primitive instantiation," in *Proceedings of the 7th FPGAworld Conference*, pp. 20–27, ACM, 2010.

Runbin Shi received the B.Eng. and M.Eng. degrees from Soochow University, Suzhou, China, in 2013 and 2016. Since September 2016, he has been pursuing a Ph.D. degree at the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interest is on image super-resolution and its real-time implementation for biomedical application.

Justin S. J. Wong received the MEng and PhD degree in Electrical and Electronic Engineering from Imperial College London, UK, in 2006 and 2011. He worked as Research Associate in the Circuits and Systems Group at Imperial College London until 2014 and received the Charted Engineer (CEng) qualification. He then worked in ZMP Inc., Tokyo, Japan, on real-time sensor and imaging systems for autonomous vehicles until 2016. He is currently the Vice President Chief Engineer of Conzeb Ltd. in Hong Kong and is collaborating closely with HKU to develop FPGA based ultra-high throughput real-time imaging and classification systems for cancer diagnostic. His research interests include ultra-highspeed real-time image processing, super-resolution, and convolutional neural network (CNN) based image classification on FPGAs and GPUs.

Edmund Y. Lam (F'15) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University. He is currently a Professor in Electrical and Electronic Engineering and Associate Dean of Engineering at the University of Hong Kong. He has broad research interests in computational optics and imaging, with over 300 journal and conference publications. He is a Fellow of IEEE, OSA, SPIE, IS&T, and HKIE, and a recipient of the IBM Faculty Award. He also serves as a senior area editor of IEEE Signal Processing Letters and an associate editor of IEEE Transactions on Biomedical Circuits and Systems.

Kevin K. Tsia received the Ph.D. degree from the Department of Electrical Engineering, University of California, Los Angeles (UCLA), CA, USA, in 2009. He is currently an Associate Professor in the Department of Electrical and Electronic Engineering, and the Medical Engineering Program, at the University of Hong Kong, Hong Kong. His research interest covers a broad range of subject matters, including ultrafast real-time spectroscopy and microscopy for biomedical applications such as imaging flow cytometry and MHz optical coherence tomography. His previous research works, such as energy harvesting in silicon photonics and the World's fastest optical imaging system, have attracted worldwide press coverage and featured in many science and technology review magazines such as MIT Technology Review and EE Times and Science News. He received the Early Career Award 20122013 by the Research Grants Council, Hong Kong. He also received the Outstanding Young Research Award 2015 at HKU as well as 14th Chinese Science and Technology Award for Young Scientists in 2016. He is the author or coauthor of more than 120 journal, conference papers, and book chapters. He holds 2 granted and 4 pending U.S. patents on ultrafast optical imaging technologies.

Hayden K.-H. So (S'03-M'07-SM'15) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998, 2000, and 2007, respectively. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering, University of Hong Kong. He received the Croucher Innovation Award, in 2013, for his work in a power-efficient high-performance heterogeneous computing system, the University Outstanding Teaching Award (Team), in 2012, and the Faculty Best Teacher Award, in 2011. He has served as the Technical Program Chair for various international conferences, including the 2014 International Conference on Field-Programmable Technology (FPT), the 2014 International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART), and the 2015 IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP). He also served as the Multiprocessor Systems and Networks on Chip Track Co-Chair for the International Conference on Reconfigurable Computing and FPGAs (ReConFig) and a Guest Editor for the Journal of Signal Processing Systems.