electronics

MDPI

Article

FPGA Implementation of Shack–Hartmann Wavefront Sensing Using Stream-Based Center of Gravity Method for Centroid Estimation

Fanpeng Kong [1], Manuel Cegarra Polo [2] and Andrew Lambert [1,*]

[1] School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2612, Australia
[2] Japan Aerospace Exploration Agency, Tokyo 182-8522, Japan
* Correspondence: a-lambert@adfa.edu.au

Abstract: We present a fast and reconfigurable architecture for Shack–Hartmann wavefront sensing implemented on FPGA devices using a stream-based center of gravity to measure the spot displacements. By calculating the center of gravity around each incoming pixel with an optimal window matching the spot size, the common trade-off between noise and bias errors and dynamic range due to window size existing in conventional center of gravity methods is avoided. In addition, the accuracy of centroid estimation is not compromised when the spot moves to or even crosses the sub-aperture boundary, leading to an increased dynamic range. The calculation of the centroid begins while the pixel values are read from an image sensor and further computation such as slope and partial wavefront reconstruction follows immediately as the sub-aperture centroids are ready. The result is a real-time wavefront sensing system with very low latency and high measurement accuracy feasible for targeting on low-cost FPGA devices. This architecture provides a promising solution which can cope with multiple target objects and work in moderate scintillation.

Keywords: adaptive optics (AO); Shack–Hartmann wavefront sensor (SHWFS); wavefront sensing; field-programmable gate array (FPGA)

Citation: Kong, F.; Cegarra Polo, M.; Lambert, A. FPGA Implementation of Shack–Hartmann Wavefront Sensing Using Stream-Based Center of Gravity Method for Centroid Estimation. *Electronics* **2023**, *12*, 1714. https://doi.org/10.3390/electronics12071714

Academic Editors: Andres Upegui, Andrea Guerrieri and Laurent Gantel

Received: 27 February 2023
Revised: 27 March 2023
Accepted: 28 March 2023
Published: 4 April 2023

1. Introduction

The Shack–Hartmann wavefront sensor (SHWFS) is one of the most widely used wavefront sensors (WFSs) in adaptive optics (AO) systems [1]. The micro lenslet array (MLA) samples the input wavefront spatially, and by determining the local slope on each sub-aperture, the entire wavefront can be reconstructed successively. The local slopes over individual sub-apertures are calculated linearly from the displacements of spots focused by the micro lenses from their optical axes. Therefore, the accuracy of the spot displacement estimation is directly related to the overall performance of the SHWFS, as error from the estimation will pass through various stages to the final wavefront reconstruction stage. Center of Gravity (CoG)-based methods have been largely used for determining the centroids of the spots by measuring the center of mass within a window associated to a particular micro lenslet. Matched filter [2], minimum mean-square-error estimator [3], and maximum-likelihood methods [4] have been studied to estimate the centroid or wavefront slope as well. If the source is an extended object instead of a point source, cross-correlation methods can also be used to estimate the sub-image displacement [5–7]. A comparison of some commonly used centroiding algorithms, including thresholding (TCoG), weighted centroid (WCoG), correlation, and quad cell, is given by Thomas et al. [8]. For closed-loop AO systems and fast wavefront sensing, the CoG-based methods are still preferred over other methods due to their robustness and easy implementation.

Conventional CoG estimations are often corrupted due to various noise sources such as photon noise, readout noise of the image sensor, and the finite and coarse sampling of

the pixels. These CoG-derived methods also fail to estimate the local wavefront if the spot breaks into parts due to strong turbulence with lower Fried coherence length r_0 or when scintillation from distant volumes of turbulence is presented. Using Gaussian approximation, Rousset [9] pointed out that the noise variance in local wavefront estimation due to sensor readout noise increases with the increased number of pixels for CoG calculation. On the other hand, Irwan et al. showed that the error variance due to photon noise also diverges as detector size increases, even for a perfect CCD array, and even without readout noise and effects due to finite pixel size. These analyses mean that a small CoG calculation window is necessary in order to reduce the error contribution of photon and sensor readout noise. However, if the window for the CoG calculation is too small, signal truncation error will be introduced when the spot moves to the window edge, which in turn leads to a limited dynamic range of the SHWFS. Therefore, the optimal CoG calculation window size often needs to balance between the noise errors and dynamic range. To isolate the useful signal for CoG calculation, the traditional CoG methods have been improved by thresholding the signal [10,11] or adding weight to emphasize the signal [12,13]. The error sources for centroid computation of a point source on a CCD-based sensor were analyzed by Ma et al. [14] and the best threshold level is given. Other methods using iterative detection of spot location and centroid estimation have been reported [15,16]. While they improve the best area for the CoG calculation, these iterations will introduce extra delay for the centroid measurements and eventually reduce the bandwidth of a closed-loop AO system. Recent efforts to improve the performance of Shack–Hartmann WFS include direct wavefront reconstruction as a continuous function from a bitmap image of the Shack–Hartmann pattern [17], and using artificial neural networks [18] and learning-based methods [19] for spot detection. To overcome the limitation of Shack–Hartmann WFS in certain situations, e.g., under strong scintillation, a diffractive lenslet array can also been used to replace the physical MLA, leading to a more flexible and adaptable Shack–Hartamnn WFS [20]. Talmi and Ribak [21] showed that gradient calculation over the whole aperture is possible by direct demodulation on the grid without reverting to Fourier Transforms. This method is especially suited to very large arrays due to the saving of computation by removing the two inverse Fourier Transforms. Importantly, they considered that incomplete spots, for example, at the edge of the aperture, would create bias on the complete reconstruction, and showed that processing these in a sensible way in the image domain could have a lesser effect on the whole reconstruction.

In addition to the effort to improve the accuracy of centroid estimation algorithms, other researchers also tried to increase the wavefront sensing speed by utilizing special hardware such as GPU [22,23] or field-programmable gate array (FPGA) devices for implementation. For example, FPGA devices have been used both in complex AO systems to process data where the timing is crucial [24–26], or used to implement centroid estimation and reconstruction [27–29], or even to develop full AO application including driving the wavefront corrector [30,31]. In comparison with conventional CPU [32] or GPU-based solutions, FPGA devices provide a cost-effective way to achieve a high throughput, low latency and reconfigurable wavefront sensing, and AO system thanks to their parallel computation power.

In our previous work [33], we proposed an improved stream-based center of gravity (SCoG) method for centroid estimation which is suitable to be implemented on FPGA devices. By extending the conventional CoG method to evaluate the center of gravity around each incoming pixel, the SCoG method can use an optimal CoG window matching the size of the spot behind the MLA without the common trade-off between increased bias error and reduced noise errors. In addition, the accuracy of the centroid estimation by SCoG is not compromised when the spot moves to the sub-aperture edge or even crosses the boundary, since the CoG operation centers on each individual pixel. The SCoG is also able to detect multiple centroids within one sub-aperture when the size of the CoG window is chosen appropriately because of its whole sensor centroid calculation.

While complicated and advanced CoG methods [12,13,15,17] have been proposed to improve the accuracy of the centroid estimation, little work has been reported to discuss their appropriate implementations to meet the low latency and high bandwidth requirement in real-time AO systems. However, most real-time implementations of the Shack–Hartmann WFS [22,23,28,32] use the basic thresholding CoG method. In this paper, we present a complete Shack–Hartmann wavefront sensing system implemented on FPGA hardware with very low latency and high accuracy using the superior SCoG for centroid estimations. A parallel slope calculation and a robust least-squares wavefront reconstruction are also implemented in a pipe-lined way after the centroid estimation. The paper is organized as follows: In Section 2, the theory of stream-based center of gravity deriving from conventional CoG methods, special treatments of multiple or missing centroids in sub-apertures, and the modal wavefront reconstruction are explained. The hardware implementations of the SCoG module, centroids segmentation module, and least-square modal wavefront reconstruction module are described in detail in Section 3. In Section 4, the resource usage and latency of the FPGA implementation are analyzed. Performance of the centroiding algorithm is compared with a traditional CoG method using an artificially generated image of spots followed by an examination of the wavefront reconstruction performance of the whole Shack–Hartmann WFS system. Conclusions and future work are summarized in Section 5.

2. Theoretical Background

Some notations used in this section to describe the stream-based center of gravity algorithm are listed in Table 1.

Table 1. Notations.

Symbol	Description
(i, j)	pixel indices
(r, c)	sub-aperture indices
(p, q)	stream centroids indices
$\mathcal{A}(r, c)$	sub-aperture
$\mathcal{C}(p, q)$	stream centroid
$\hat{C}_x(r, c), \hat{C}_y(r, c)$	centroid estimation in $\mathcal{A}(r, c)$
$s_x(r, c), s_y(r, c)$	average slope in $\mathcal{A}(r, c)$
$I(x_i, y_j)$	image intensity at pixel (x_i, y_j)

2.1. Conventional Center of Gravity

The geometric diagram of a single lenslet from a MLA is shown in Figure 1. The local wavefront tilt θ in a sub-aperture $\mathcal{A}(r, c)$ causes a shift Δx of the focal spot from its reference on-axis position $(C_{x,ref}(r, c), C_{y,ref}(r, c))$ when a plane wavefront is used. The size of the diffraction-limited spot is determined by the f-number of the lenslet and equals to $2.44\lambda f_{ML}/D_{ML}$ where f_{ML} and D_{ML} are the focal length and diameter of the micro lens respectively. By determining the local slopes at all sub-apertures, a continuous wavefront map can be reconstructed using either zonal- or modal-based reconstruction methods.

The local slope over sub-aperture $\mathcal{A}(r, c)$ can be calculated by:

$$s_x(r, c) = \frac{\Delta_x(r, c)}{f_{ML}} = \frac{\hat{C}_x(r, c) - C_{x,ref}(r, c)}{f_{ML}} \tag{1a}$$

$$s_y(r, c) = \frac{\Delta_y(r, c)}{f_{ML}} = \frac{\hat{C}_y(r, c) - C_{y,ref}(r, c)}{f_{ML}} \tag{1b}$$

where $\Delta_x(r, c)$, $\Delta_y(r, c)$ are the displacement of the spot in x and y directions from its reference location.

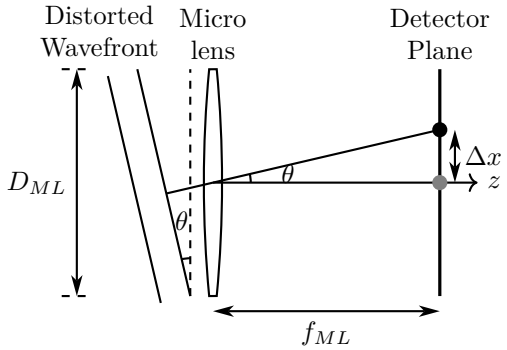

Figure 1. Schematic of a single micro lens in the Shack–Hartmann wavefront sensor.

In Equation (2), the centroid in the sub-aperture $\mathcal{A}(r,c)$ can be measured in x and y directions using the traditional CoG definition as:

$$\hat{C}_x(r,c) = \frac{\sum_{x_i}\sum_{y_j} x_i I(x_i,y_j)}{\sum_{x_i}\sum_{y_j} I(x_i,y_j)} \tag{2a}$$

$$\hat{C}_y(r,c) = \frac{\sum_{x_i}\sum_{y_j} y_j I(x_i,y_j)}{\sum_{x_i}\sum_{y_j} I(x_i,y_j)} \tag{2b}$$

where r,c is the coordinate of the sub-aperture, x_i, y_j is pixel index, and $I(x_i,y_j)$ is the pixel intensity. In a traditional instrument, the CoG is only calculated on the central pixel location $i = kr, j = kc$ per lenslet.

2.2. Stream-Based Center of Gravity

The conventional CoG method can be extended by evaluating the centroid estimation on each pixel of the signal:

$$\hat{C}_x(i,j) = \frac{\sum_{m=-M}^{M}\sum_{n=-N}^{N} F_x(m,n) I(x_i,y_j)}{\sum_{m=-M}^{M}\sum_{n=-N}^{N} I(x_i,y_j)} \tag{3a}$$

$$\hat{C}_y(i,j) = \frac{\sum_{m=-M}^{M}\sum_{n=-N}^{N} F_y(m,n) I(x_i,y_j)}{\sum_{m=-M}^{M}\sum_{n=-N}^{N} I(x_i,y_j)} \tag{3b}$$

where $F(m,n)$ is a linear filter ranging from $-M$ to M. $\hat{C}_x(i,j), \hat{C}_y(i,j)$ represents the estimated centroid value for the pixel (i,j) within a square window of side size of $2M+1$.

$$F_x(m,n) = \begin{pmatrix} -M & -M+1 & \cdots & M \\ -M & -M+1 & \cdots & M \\ \vdots & \vdots & \ddots & \vdots \\ -M & -M+1 & \cdots & M \end{pmatrix} \tag{4a}$$

$$F_y(m,n) = \begin{pmatrix} -N & -N & \cdots & -N \\ -N+1 & -N+1 & \cdots & -N+1 \\ \vdots & \vdots & \ddots & \vdots \\ N & N & \cdots & N \end{pmatrix} \tag{4b}$$

If the centroid estimation value at a pixel (i_p, j_q) equals to zero, then a spot is centered on this pixel. In most cases, however, the centroid is less likely to sit on an exact pixel but rather between two pixels. A potential spot is detected around pixel (i_p, j_q) if a zero-crossing (from positive to negative) of CoG values happens horizontally between pixels $(i_p - 1, j_q)$ and (i_p, j_q) and vertically between pixels $(i_p, j_q - 1)$ and (i_p, j_q) at the same time.

The sub-pixel shift in the x directions from pixel (i_p, j_q) can be interpreted linearly by the CoG values at pixel (i_p, j_q) and its left pixel $(i_p - 1, j_q)$, while the sub-pixel shift in y direction can be interpreted similarly by the CoG values at pixel (i_p, j_q) and its above pixel $(i_p, j_q - 1)$:

$$\Delta_x(i_p, j_q) = \frac{\hat{C}_x(i_p, j_q)}{\hat{C}_x(i_p - 1, j_q) - \hat{C}_x(i_p, j_q)} \tag{5a}$$

$$\Delta_y(i_p, j_q) = \frac{\hat{C}_y(i_p, j_q)}{\hat{C}_y(i_p, j_q - 1) - \hat{C}_y(i_p, j_q)} \tag{5b}$$

Therefore, the (p, q)th centroid $\mathcal{C}(p, q)$ where a potential spot is located can be described by $(\hat{x}_p, \hat{y}_q)$ as below:

$$\hat{x}_p = i_p + \Delta_x(i_p, j_q) \tag{6a}$$

$$\hat{y}_q = j_q + \Delta_y(i_p, j_q) \tag{6b}$$

Note that the calculation of integer and decimal parts of centroid $\mathcal{C}(p, q)$ are through separate steps and the integer parts, i.e., pixel indices, are determined first. (i_p, j_q) can be obtained directly by the sign changes of the numerators in Equation (3) as the denominators which represent the sum of energy within the kernel window are always positive. If only whole pixel resolution is concerned, calculations of the numerators are sufficient to locate the pixel indices.

In this work, an estimate of centroid is made based on each and every pixel streamed from the image sensor, for which best centroids are tagged resulting in a stream of centroids or SCoG synchronous with the stream of pixels. Several immediate advantages of SCoG over conventional CoG-based centroid estimation methods can be noticed [33]. Since the CoG window is floating with the incoming pixels and will center around each potential genuine centroid, bias errors due to asymmetric CoG filter are largely avoided. In addition, the size of the CoG window $2M + 1$ can be optimised by matching with the diffraction-limited spot size to minimize the influence of irrelevant pixels so that noise errors are minimised.

It is worth noting the unique characteristics of the centroids detected by the stream-based CoG algorithm. Using conventional sub-aperture-based CoG methods, only one centroid will be estimated for each sub-aperture, even when multiple spots exist due to, for example, binary star structure or strong turbulence, resulting in a "broken" spot (r_0 is less than the microlens diameter). In addition, the measured centroids belonging to each sub-aperture are rather apparent from conventional CoG methods. However, the stream of centroids from SCoG arises in order based on the position of the spot occurrence in the input image frame as it is read from the image sensor, row-by-row, and column-by-column. Therefore, the SCoG centroids stream need to be further processed in order to be used in the conventional zonal or modal wavefront reconstruction algorithms.

2.3. Segmentation of the Streamed Centroids

There are two problems that need to be addressed for the stream of centroids $\mathcal{C}(p, q)$ in order to get the centroid estimation for each sub-aperture associated with traditional CoG-based methods. First, the occurrence of centroid $\mathcal{C}(p, q)$ follows the lower row number to a higher row number or a lower column number to a higher column number depending on the pixel reading sequence of the particular sensor. For a particular centroid $\mathcal{C}(p, q)$, it needs to be assigned to a sub-aperture $\mathcal{A}(r, c)$ if its values are within the window of $\mathcal{A}(r, c)$ defined by:

$$r = \left\lfloor \frac{i_p}{w} \right\rfloor, c = \left\lfloor \frac{j_q}{w} \right\rfloor \tag{7}$$

where $\lfloor \cdot \rfloor$ is the floor operation and w is the window width in pixels.

Secondly, it is possible that multiple centroids or no valid centroid are detected within one sub-aperture $\mathcal{A}(r, c)$, perhaps because of multiple objects, obstruction, or scintillation.

For the multiple centroids case, different strategies can be used, such as using the centroid with the highest energy (sum of intensity as the denominator in Equation (3)) or taking the average of all centroids.

In our current implementation, we used the average of all centroids in the sub-aperture, hence effectively measuring the G-tilt of the local wavefront:

$$\left.\begin{aligned}\hat{C}_x(r,c) &= \langle \hat{x}_p \rangle \\ \hat{C}_y(r,c) &= \langle \hat{y}_q \rangle\end{aligned}\right\} \quad \text{where} (i_p, j_q) \in \mathcal{A}(r,c) \tag{8}$$

where $\langle \cdot \rangle$ denotes the average operation and $\in$ means the integer part of $\mathcal{C}(p,q)$ that falls within the pixel range of sub-aperture $\mathcal{A}(r,c)$.

On the other hand, if a centroid is missing in a sub-aperture $\mathcal{A}(r,c)$, it is possible to generate an average one from its surrounding sub-apertures. However, if one or more surrounding sub-apertures also miss valid centroids, the chain of average operation will expand to further sub-apertures which could soon become too complicated to manage. The other method to treat the sub-aperture with missing centroid is to inherit the corresponding centroid from a previous frame, which could be traced back to an original reference centroid if none of the previous frames contain a valid centroid.

Once all the valid sub-apertures in one MLA row have been processed to select a representative centroid estimation, they need to be re-ordered to match the sequence of the physical geometry, so the following wavefront reconstruction can be started even though the remaining lenslets have not yet arrived. Therefore, the further processing of the streamed centroids follows a *sorting–processing–reordering* procedure.

2.4. Wavefront Reconstruction

From the discrete slopes at all the valid sub-apertures as given by Equation (1), a continuous wavefront map can be reconstructed using *zonal*, *modal*, or *FFT-based* methods [34,35]. Using the modal wavefront reconstruction, the incoming wavefront $W(x,y)$ over the pupil can be decomposed by a set of orthogonal functions, such as Zernike polynomials:

$$W(x,y) = \sum_{k=1}^{N} a_k Z_k(x,y) \tag{9}$$

where a_k represents the weight of the kth Zernike term $Z_k(x,y)$ and N is the total number of Zernike modes used to approximate the actual wavefront.

In Equation (1), the local slope of wavefront on the individual sub-apertures is expressed as a relation between the local sub-image displacements for each axis $(\Delta_x(r,c), \Delta_y(r,c))$ and the MLA focal length f_{ML}. Considering only the x axis results in the following equation:

$$s_x(r,c) = \left.\frac{\partial W(x,y)}{\partial x}\right|_{(r,c)} = \frac{\Delta x(r,c)}{f_{ML}} \tag{10}$$

By combining Equations (9) and (10), the gradients of the wavefront over each sub-aperture can be related with a weighted sum of Zernike polynomials as follows:

$$\frac{\Delta x(r,c)}{f_{ML}} = \sum_{k=1}^{N} a_k \left.\frac{\partial Z_k(x,y)}{\partial x}\right|_{(r,c)} \tag{11}$$

Considering the slopes in both x and y directions for all the sub-apertures, Equation (11) can be expressed in the following matrix form:

$$\boldsymbol{s}_{2M\times 1} = \boldsymbol{W}_{2M\times N}\boldsymbol{a}_{N\times 1} \tag{12}$$

where M is the total number of valid sub-apertures and N is the number of Zernike modes used for wavefront reconstruction. $\boldsymbol{s}$ is the slope vector of dimensions $2M \times 1$ and $\boldsymbol{a}$ is the

Zernike mode coefficients vector of dimension $N \times 1$. $\boldsymbol{W}$ is a matrix of dimension $2M \times N$ whose elements are the partial derivative of a Zernike mode on either x or y direction. Equation (12) defines $2M$ linear equations. To obtain the Zernike coefficients vector $\boldsymbol{a}$ from measured slope vector $\boldsymbol{s}$, the pseudo-inverse of matrix $\boldsymbol{W}$ is used:

$$\boldsymbol{a}_{N\times 1} = \boldsymbol{E}_{N\times 2M}\boldsymbol{s}_{2M\times 1}; \tag{13}$$

The pseudo-inverse matrix $\boldsymbol{E}$, also known as the calibration matrix, has a dimension of $N \times 2M$ and can be calculated using the least squares estimation method:

$$\boldsymbol{E} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T \tag{14}$$

In Section 3.4, the detailed implementation of the modal wavefront reconstruction is explained.

3. Implementation

In this section, a parallel implementation of the stream-based center of gravity algorithm and the modal wavefront reconstruction suitable for FPGA devices are described. By taking advantage of parallelism and storage resources of FPGA devices, the CoG operation can be evaluated at each incoming pixel in real-time. The wavefront reconstruction can also start partially as soon as the centroids estimation of sub-apertures becomes available and complete at a very short delay after acquiring one image frame (and certainly before the start of the next frame).

The overall block diagram of the SHWFS system design using the stream-based CoG method is shown in Figure 2. The complete implementation consists of four main modules corresponding to Sections 2.2–2.4: *SCoG* module computes centroid on all incoming pixels and presents a stream of valid centroids; *SEG* module sorts the centroids to confined sub-apertures and handles multiple centroids or centroids missing in some sub-apertures; *SLOPE* module calculate the local slopes from measured centroids and can also be used to generate a reference centroid grid; *RECON* module conducts a modal wavefront reconstruction from the measured slopes. The SCoG and SEG modules together provide similar functions to other conventional CoG methods. The auxiliary pre-processing, display, and control modules shown in light blue blocks are necessary to configure the image sensor, set system parameters, and visualize various results but not directly related to the research interest and therefore are omitted in the following discussion.

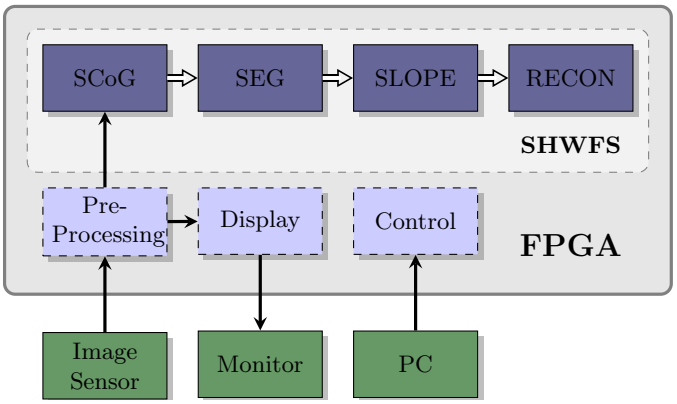

Figure 2. Top level block diagram of the SHWFS implementation where the implementation of the SHWFS is described here.

3.1. SCoG Module

In Figure 3, the 2D pixel array of an image sensor at the focal plane of the MLA is shown. The sub-aperture window SA corresponds to individual lenslet with its size and number of pixels (5×5 for this example) determined by the lenslet and pixel sizes.

Traditional CoG-based methods use all the pixels (except for the windowing CoG) inside the SA window for the calculation and generally need to read the whole image frame before computation. The SCoG operates on a much smaller pixel set (the blue dashed window) which could be optimized by matching the window with the spot size (3×3 for this example). In the FPGA devices, First-In, First-Out (FIFO), or Look Up Table (LUT) can be used to buffer several rows and columns of pixels, so as the last required pixel arrives (light green and light blue pixel for the first and last possible centroids in SA_{11}), the computation of the CoG will start.

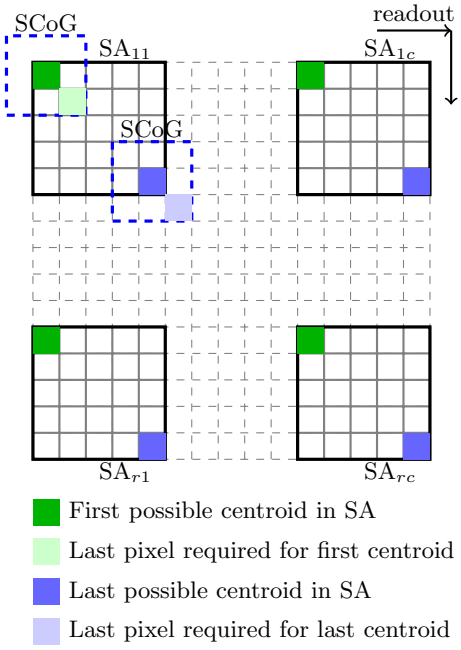

Figure 3. Illustration of the sub-aperture and SCoG window on the image sensor pixel array.

A high level logic diagram for calculating the stream centroid according to Section 2.2 on each incoming pixel is shown in Figure 4, where pixel Data D_{IN} is synchronous with the master clock. The circuit in the yellow box represents a 2D convolution operation and it calculates either the numerators or denominators in Equation (3) depending on the chosen coefficients C_{11} to C_{33} in the multipliers, which represent the matrix elements in Equation (4). The SCoG window is assumed with a 3×3 size, and therefore three buffers are used to store three rows of pixel values and three Flip-Flop (FF) registers are used to buffer three columns of pixels. As the last required pixel within the SCoG window arrives at the output of the first buffer, all the required pixel values will appear at the nine registers' output simultaneously at the next clock cycle. This implementation makes logical sense but is impractical, as all the multiplications and additions need to be finished in one clock cycle. This causes challenges to meeting timing constraints in FPGA implementation. In Appendix A, matrix multiplication is separated to two 1D filter operations, which reduces the usage of multipliers. Furthermore, the implementation of the two 1D filter operations are fully pipe-lined to best improve the timing closure in trade of more latency.

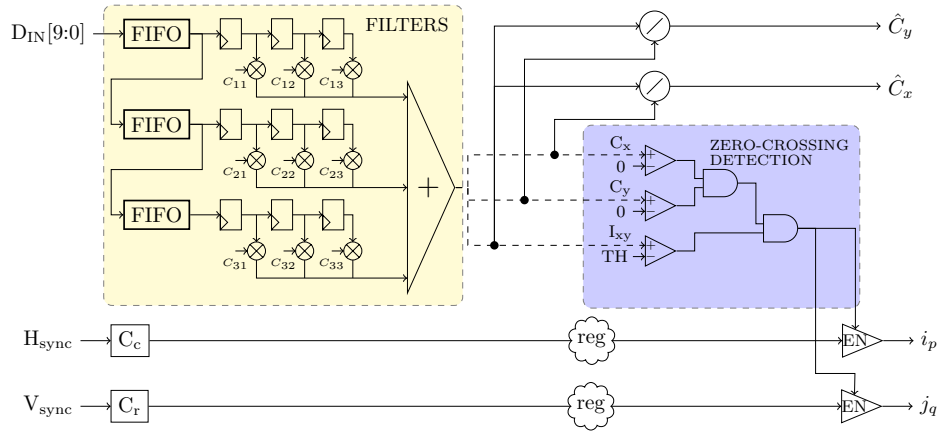

Figure 4. Diagram for the implementation of SCoG module used for calculate integer pixel centroid values. Depending on the coefficient values $\boldsymbol{C}$, the numerators and denominators in Equation (3) are calculated from the FILTERS circuit.

The zero-crossing detection circuit is shown in the blue box where the numerators and denominators in Equation (3) are present simultaneously at the input. The horizontal and vertical zero-crossing can be determined by checking the sign of the numerators while the sum of the intensity, i.e., the denominator, can be compared with a threshold value I_{TH} to eliminate fake centroids due to noise pixels. The two dividers calculate the centroid at pixel (i_p, j_q) as described by Equation (3). The registers on the H_sync and V_sync lines are necessary to introduce the same number of clock delays for the counters of row and column numbers as those FILTERS, and ZERO_CROSSING DETECTION circuits. The length of these is a direct measure of the latency of the calculation.

The implementation of the sub-pixel interpolation in the x and y directions according to Equations (5) and (6) is shown in Figure 5. To interpret the sub-pixel shift $\Delta_x(i_p, j_q)$, the centroid estimation from the previous pixel is required, which can be buffered by a FF register. For the sub-pixel shift in the y direction, however, the pixel in the previous row is needed, which means the whole row of centroids value in y direction needed to be buffered in a FIFO with depth that is the same as the column number.

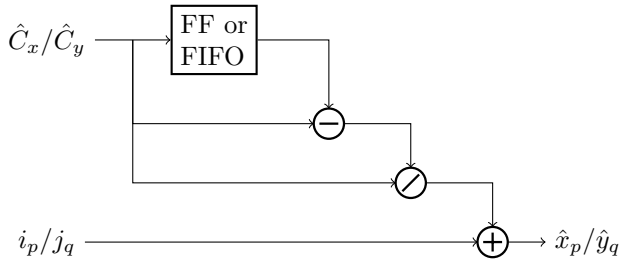

Figure 5. Diagram for the implementation sub-pixel interpolation in x and y direction.

3.2. Segmentation Module

The stream-based CoG algorithm itself is able to determine multiple spots presented in a sub-aperture if the filter size is not so large as to cover all the spots. Recall that the filter size is matched to the spot size, not the sub-aperture size Equation (3). How to use these extra centroids to achieve a better local slope estimation (Z-tilt instead of G-tilt) or reconstruct wavefront from multiple sources is a simple sorting procedure is beyond the scope of this paper. Here, we try to render a traditional array of centroids corresponding to the lenslet array from the SCoG detected centroids through the Segmentation module using the following steps.

3.2.1. Centroid Sorting Module

It is assumed that all the detected centroids in one sub-aperture truly belong to this sub-aperture. In other words, there are no spots crossing lenslet boundaries due to strong aberration. In order to associate a detected centroid to its sub-aperture, Equation (7) was used to calculate the row and column index of the sub-aperture. When the number of pixels per lenslet window and integer of power is two, the division can be simplified to bit shifts.

3.2.2. Treatment of Multiple Centroids and Missing Centroid

As discussed in Section 2.3, there are different ways to treat the scenarios where multiple centroids or no centroid is detected in some sub-apertures. For the current FPGA implementation, the average of all the centroids within a sub-aperture is used to represent an overall displacement for the multiple centroids situation. If a centroid is missing from a sub-aperture, then the centroid position from previous frame is inherited, which can trace back to an initial default reference centroid positions for each sub-aperture.

Once the centroids for the last sub-aperture in a row have been examined following by the slope calculation, the partial wavefront reconstruction can start immediately. The above segmentation steps can be represented by the pseudocode in Algorithm 1:

Algorithm 1: Pseudocode for *sorting–processing–reordering*.

Data: stream centroid $\mathcal{C}$
Result: ordered centroid for each sub-aperture $\mathcal{A}$
initialization;
foreach stream centroid $\mathcal{C}$ **do**
  $r := floor(\mathcal{C}_x/w)$;
  $c := floor(\mathcal{C}_y/w)$;
  append $\mathcal{C}$ to centroid list of $\mathcal{A}(r, c)$;
  if time of the last possible $\mathcal{C}$ in $\mathcal{A}$ **then**
    if no centroid in $\mathcal{A}(r, c)$ **then**
      Assign the centroid from previous frame;
    else if multiple centroids in $\mathcal{A}(r, c)$ **then**
      Average all the centroids;
    if last $\mathcal{A}$ in a MLA row **then**
      reorder all the centroids in this row;
    end
  end
end

3.3. Slope Calculation Module

After the SCoG and Segmentation modules, there will be an averaged centroid for each sub-aperture similar to conventional CoG-based methods. The SLOPE module calculates local slopes of wavefront in each sub-aperture according to Equation (1). An initial reference grid is stored in this module, where the reference center is designed to be in the center of each sub-aperture. This module is also capable of updating the reference centroids grid by averaging a certain frames of centroids data. This is particularly useful when applying the wavefront sensing system to an imaging system where a flat wavefront is not accessible. Therefore, the reference grid must be calculated by averaging the centroids from a number of long exposure images.

3.4. Wavefront Reconstruction Module

The wavefront reconstruction module essentially computes a matrix multiplication between the inverse matrix and the slope vectors according to Equation (12). As the slopes for each sub-aperture are measured, they are multiplied by their corresponding matrix elements and accumulated with previous multiplication. The calibration matrix E

is determined once the geometric configurations is fixed and therefore can be calculated in advance. When the slopes of all the sub-apertures in the same row are measured and multiplied, the next section of the matrix corresponding to the slopes of next row of sub-apertures are loaded. The loading of new matrix elements is controlled by a finite state machine (FSM).

In Figure 6, different areas related with the SHWFS are shown. The gray sensor area includes all the selected active imaging sensor pixels, on which the SCoG is operated. The resulted detected centroids from the SCoG module are represented on this sensor coordinate system. Since most AO systems have either a circular or annular Pupil, a square ROI was chosen to include the Pupil. After removing the ROI origin offset (os_x, os_y) from the centroids calculated by the SCoG module, the local centroids were obtained on the ROI coordinate system. Following that, the segmentation module can sort the centroids to their associated sub-apertures, followed by the slope module to obtain the average slope on individual sub-apertures. For the wavefront reconstruction, only slopes on those valid sub-apertures (green) were used. A binary sub-aperture mask indicating the validness of an sub-aperture was used to decide whether to accumulate matrix multiplication result to the partial wavefront reconstruction in the RECON module.

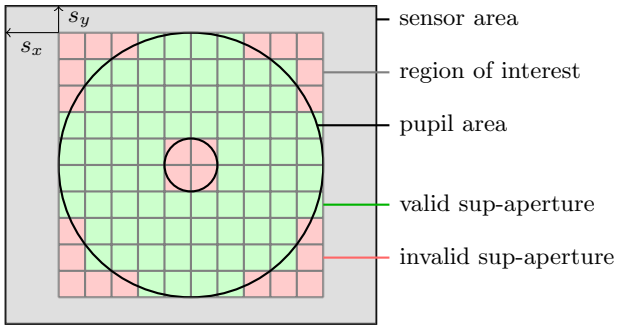

Figure 6. Illustration of various regions on the sensor area. The gray area is the sensor area on which the stream-based CoG operates. The region of interest is selected such that it covers the pupil of the imaging system. The green segments indicate those valid MLA lenslets within the pupil, while the red segments are invalid sub-apertures for wavefront reconstruction.

4. Results

4.1. FPGA Design

The hardware platform used to implement the proposed Shack–Hartmann wavefront sensor using the stream-based CoG method is shown in Figure 7. The FPGA board is a Trenz Electronic TE0712 core board (TE0712-01-200-2C), which includes a Xilinx 7 series FPGA Artix-7 chip XC7A200T.

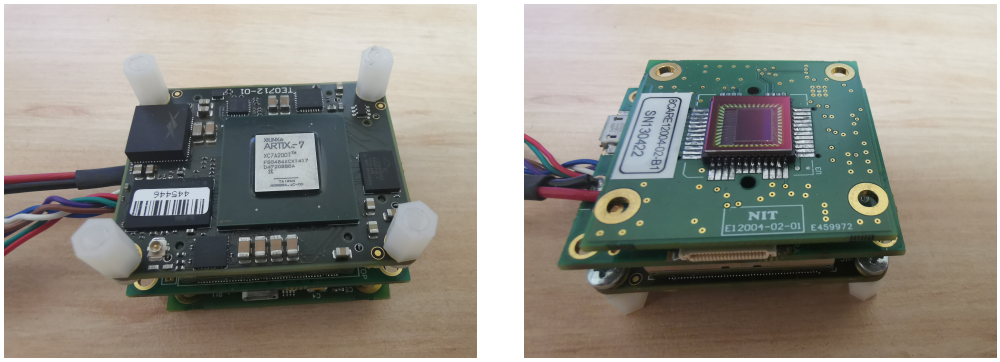

Figure 7. Hardware platform used to implement the SCoG-based SHWFS.

The FPGA implementation was developed in the Xilinx Vivado Design Suite, with most parts of the different modules designed using VHDL directly. Some common IP

blocks, such as FIFOs, Dividers, and Block Memories provided by Xilinx, have also been used in various places. This implementation strategy reduces the migration complexities and difficulties when re-targeting the design to a different FPGA platform.

The arithmetic operations starting from the calculation of the sub-pixel shift use fixed-point numbers. The absolute centroid position has a precision of 12 integer bits and 8 fractional bits, while the slope calculation uses 1 sign bit and 16 fractional bits. The intermediate calculations use full precision, i.e., integer and fractional bits are extended accordingly. The final Zernike amplitudes are rounded to 1 sign bit, 7 integer bits, and 8 fractional bits.

4.1.1. Resource Usage

The resource usage for the SCoG module, centroids segmentation module, slope calculation module, and modal wavefront reconstruction module for a SHWFS with 10×10 sub-apertures, each having 30×30 pixels. as is shown in Table 2. The filter size is 5×5 and the wavefront reconstruction has been performed on the first 9 Zernike modes based on Noll's notation. The SEG module uses more LUT RAMs than other components because centroids from previous frame are buffered to solve the possible missing centroid situations in some sub-apertures. The RECON module uses the most Block RAMs only because the calibration matrix are stored inside this module instead of external RAM.

Table 2. Resource usage of different modules on Artix 7 XC7A200T.

Resources	SCoG	SEG	Slope	RECON	Total
Slice LUT	1554	7744	33	3731	13,062 (10%)
Slice REG	3567	16,184	19	10,100	29,870 (11%)
Block RAM	3.5	0	0	26	29.5 (8.1%)
DSPs	0	0	2	18	20 (2.7%)

The resource usage for the SCoG module with four different filter sizes is shown in Figure 8. The implementation for slope calculation and wavefront reconstruction are common and similar to those in the literature [27,28] and therefore are not shown here separately. As the filter size increases, the slice usages as LUT and registers increase linearly. The Block RAMs are mainly used to buffer a certain number of rows pixels depending on the filter size, and therefore also increase linearly.

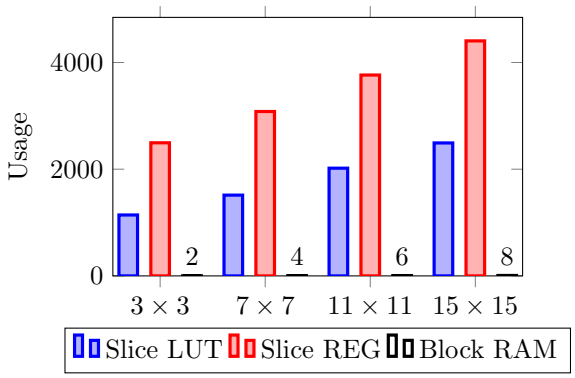

Figure 8. Resource usage for the SCoG module with four different filter sizes. Contributing resources for each filter size include Slice LUT, Block RAM, and Slice Registers.

4.1.2. Latency

The various time delays from the pixel read to the calculation of wavefront are shown in Figure 9. The SCoG computes the centroid on each incoming pixel and the time delay from a when pixel is read to its associated CoG value being calculated t_{SCoG} is:

$$t_{SCoG} = M \times T_{ROW} + M \times T_{CLK} + c \times T_{CLK} \quad (15)$$

where M is half of the filter size and c is a constant related to the delays caused by pipe-line implementation and division. The detected centroids will be sorted to a sub-aperture according to Algorithm 1, which introduces a delay t_{Sort} mostly due to divisions. When the CoG value of the last pixel in a row of sub-apertures is sorted, multiple centroids (if presented) in each individual sub-aperture will be averaged after a further delay of t_{SEG}. At this point, each sub-aperture will have its centroid estimation and the following timing delay is similar to those reported in [28]. The slope calculation can be started as soon as the segmentation is done, as the operation is basically a subtraction of the sub-aperture centroid with its reference center and following multiplication with a factor related with lenslet f-number. The time delay from the segmentation to the slope calculation is denoted as t_{SLOPE}.

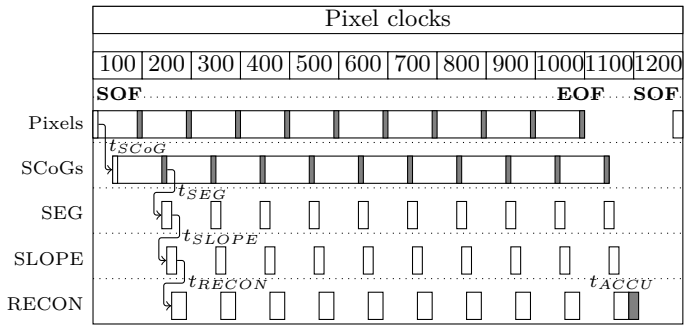

Figure 9. Time delay between different modules in the SHWFS implementation. Note that the whole wavefront sensing operation finishes shortly after End-Of-Frame (EOF) and before the next Start-Of-Frame (SOF).

The matrix multiplication for the wavefront reconstruction starts as soon as the slopes of all the sub-apertures in the same row becomes available and introduces a time delay of t_{RECON} until the partial reconstruction finishes. As the multiplication on the last row finishes, a time of t_{ACCU} is required to finish the final accumulation and serialization of the Zernike coefficients vectors.

With a larger filter size, the closure of the timing constraint becomes more challenging, even with fully pipe-lined implementation as described in Appendix A. In our implementation, the timing constraint for a clock running at 100 MHz can be met with a maximum filter size of 65 pixels. Considering the micro lenslet array has a pitch size ranging from 100 µm to 500 µm (equivalent f-number 20 to 50) and the common pixel size for CMOS imaging sensors of 5 to 10 µm, the spot size for a wavelength of 500 nm is about 4 to 20 pixels and each sub-aperture $\mathcal{A}$ is up to 50 × 50 pixels. Therefore, the timing constraint with this maximum filter size is sufficient for most of the SHWFS for astronomy AO systems. We note that most SHWFS limit the spot size to maximize the detectability, but meteorology look for better precision with the luxury of more light.

The appealing low latency and high accuracy of our proposed Shack–Hartmann WFS is only possible because of the deep exploitation of the parallelism power offered by FPGA devices. Due to the heavy computation overload introduced by the per-pixel convolution operations in Equation (3), CPU-based implementation will suffer from significant time delay. For a configuration of 10 × 10 sub-apertures with each sub-aperture consisting of 30 × 30 pixels and an implementation running at 50 MHz, the latency from the end of the frame to the finish of wavefront reconstruction to 9 Zernike terms is only 820 ns. The same software implementation of the Shack–Hartmann WFS, getting the benefits of CoG calculation at each pixel, written in Python 3.7 running on Ubuntu 18.04 (Intel Core i7-8750 CPU 2.20 GHz × 12, 32 GB RAM) (without exploiting specific optimisation like multithread programming) takes about 2.855 s (SCoG 2.85 s, SEG 0.19 ms, SLOPE, and RECON 4.77 ms). Image fetching time from memory in the CPU implementation has been omitted in the above comparison. The reader should therefore note that it is not sensible to implement the streaming per-pixel estimation of SCoG on a traditional CPU architecture for real-

time applications. The latency of our FPGA-based Shack–Hartmann WFS implementation using SCoG for centroiding estimation is on par with similar FPGA implementation using TCoG [28] (740 ns) and is much smaller than the GPU-based centroiding extraction [22] (2 ms) and CPU-based Shack–Hartmann WFS implementation [32] (40 µs excluding transfer time).

4.2. Experimental Results

To evaluate the effectiveness of the hardware implementation of the proposed SHWFS using SCoG algorithm, artificial SHWFS spots images were generated from numerical simulation and stored inside the FPGA ROM. Following that, a camera module written in HDL was used to mimic an image sensor by reading the stored image and outputting common CMOS sensor signals such as pixel clock, frame/line valid, and pixel data bus. This arrangement allows the experiments to still run on the hardware platform and quantify the system performance in a reproducible manner.

4.2.1. Centroiding Performance

Comparison of the SCoG method with other CoG-based and cross-correlation methods for measuring the spot displacement under various noise scenarios has been reported in our previous research [33]. The focus here is to confirm that the argument of SCoG is superior to conventional CoG method still holds with the fixed-point number representation being used in various modules. The spot profile from the lenslet is modeled using the following 2D Gaussian function:

$$P(x,y) = \frac{N_{ph}}{2\pi\sigma_{spot}^2} \exp\left[-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma_{spot}^2}\right] \tag{16}$$

where (x_0, y_0) represents the ground truth of the spot center and N_{ph} is the number of photons. σ_{spot} is the standard deviation and the FWHM of the spot is $2\sqrt{2\ln(2)}\sigma_{spot}$.

A Gaussian spot with a FWHM of 2 pixels and 100 photons moving across a sub-aperture of 30 pixels is simulated as shown in Figure 10. On top of the Gaussian signal, photon (Poisson) noise and sensor readout noise are introduced as well. The readout noise is modelled by a Gaussian distribution with a mean value of 0.8 e^- and standard deviation of 0.2 e^- under assumption of 100% quantum efficiency. The final signal is then digitized to 8 bits. The centroids estimation results from a conventional TCoG with a thresholding value of 15 and the SCoG with a filter size of 5 pixels are shown in Figure 10. While the accuracy of the TCoG varies and at some sampling position results in a totally wrong estimation, the SCoG always has a close agreement with the ground truth along the path. The error variance of the centroid estimation from TCoG is about 1.795 Pixel^2, while the number from SCoG estimation is only about 0.002 Pixel^2.

To evaluate a case where traditional CoG methods fail, in Figure 11 we present a spots pattern from a random Kolmogorov phase screen with r_0 of 14 cm. A telescope with a diameter of 4.2 m, which leads to $D/r_0 = 30$, is used. The pupil was sampled into 10×10 sub-apertures by a MLA with pitch size of 42 cm. Here, r_0 is smaller than the MLA pitch size, and therefore scintillation is expected. All the detected centroids together with the centroid representation for each sub-aperture after segmentation are annotated on the spots in blue and red colors. Noticeably, the spot breaks as the local turbulence is too strong, i.e., r_0 is less than the lenslet size, as shown in the two inset sub-apertures, and the SCoG is able to identify these individual broken spots. As stated in the previous section, the average of these centroids was used to represent the mean local wavefront. However, they could be used in other, better ways [36] to yield a more precise wavefront reconstruction with further study. The SCoG works here, despite scintillation and the accompanying signal spread.

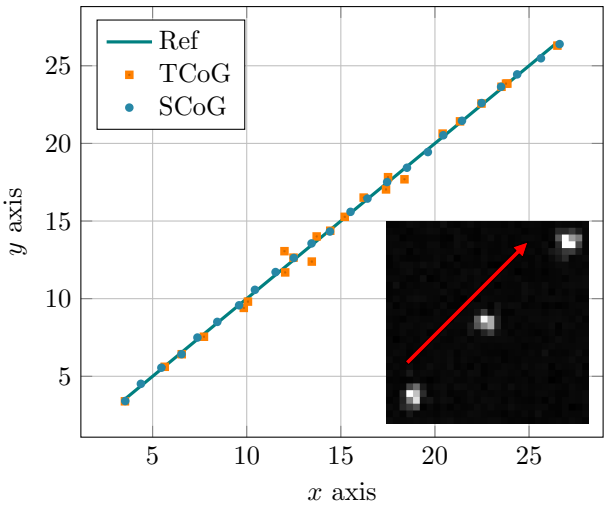

Figure 10. Comparison of hardware implemented SCoG with a traditional thresholding CoG (TCoG) for detecting a moving Gaussian spot.

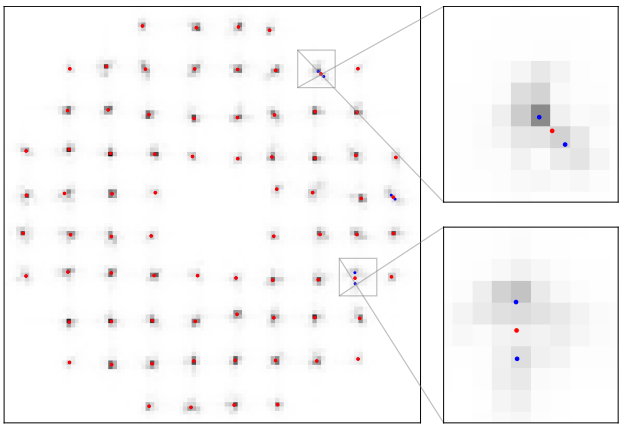

Figure 11. All the detected centroids (blue) and average ones for each sub-apertures (red) are annotated on the SHWFS spots pattern (invalid sub-apertures are not shown here). Color for the spots is inverted for better visibility.

4.2.2. Wavefront Reconstruction Results

In a similar way, to demonstrate the performance of the whole SHWFS design including the wavefront reconstruction, the SHWFS spots pattern was generated numerically from a known phase screen as shown in the top left in Figure 12. The wavefront is randomly generated by a combination of the first 9 Zernike modes, after setting a wavefront error budget of [0 nm, 100 nm, 100 nm, 36 nm, 36 nm, 36 nm, 6 nm, 6 nm, and 6 nm] at a wavelength of 635 nm and over a pupil with diameter of 1.5 mm. The MLA used to sample the wavefront has a pitch size of 150 µm and focal length of 4.1 mm, which models the Thorlabs MLA150-5C micro lenslet array. The pixel size of the detector is set to be 5 µm and therefore there are 10×10 sub-apertures and each sub-aperture has 30×30 pixels. The phase residual between the reconstructed wavefront and the original wavefront is shown in the top right in Figure 12 and the modal reconstruction has a RMSE of 0.023 rad. A comparison of the Zernike coefficients from the original and reconstructed wavefront is given in the bottom in Figure 12.

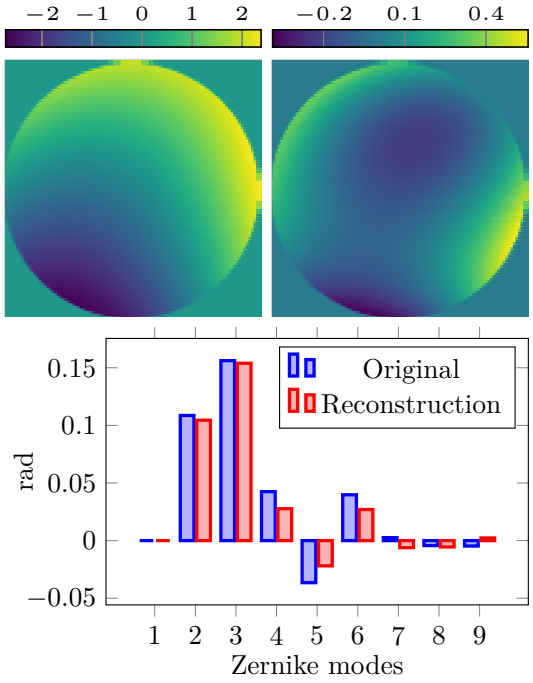

Figure 12. Top left: original phase; Top right: phase residual; Bottom: comparison of original and reconstructed Zernike coefficients using Noll's index notation [37].

5. Discussion

In this paper, we have presented a complete hardware implementation of Shack–Hartmann wavefront sensor in a modular design. The spots displacement was measured using the previously reported Stream-based CoG algorithm, which reduces the centroid estimation error when the signal is cut by the sub-aperture boundary and noise-induced error by using a spot-matched floating CoG window. The slope calculation and wavefront reconstruction using a least-square fit method starts immediately as long as the required sub-aperture measurements are ready. The result is a very low-latency and real-time wavefront sensing system with superior performance to conventional CoG-based SHWFS facilitated by the parallelism power from FPGA devices.

The FPGA implementation of Stream-based Center-of-Gravity algorithm can be modified without much effort to adapt other filter-like algorithms, for example, the cross-correlation for determining image shifts under extended sources proposed by Poyneer [6]. The First Fourier coefficient (1FC) algorithm [38] examines the phase symmetry in the Fourier domain to evaluate the spot shifts. Both the CCF and 1FC algorithms can be modified to their streamed version with minimal effort based on the implementation presented in this work.

The current treatment of multiple centroids identified in one sub-aperture is to average them, which can be further improved by other algorithm events to increase the dynamic range such as [36]. Future work also includes testing the proposed SHWFS implementation with the on-board image sensor in laboratory optics systems and eventually including the wavefront correction component, such as a deformable mirror, to realize a closed-loop AO system.

Author Contributions: Conceptualization, F.K. and A.L.; methodology, F.K., M.C.P. and A.L.; software, F.K.; validation, F.K., M.C.P. and A.L. formal analysis, F.K.; investigation, F.K., M.C.P. and A.L.; resources, A.L.; data curation, F.K.; writing—original draft preparation, F.K.; writing—review and editing, M.C.P. and A.L.; visualization, F.K.; supervision, M.C.P. and A.L.; project administration, A.L.; funding acquisition, A.L. All authors have read and agreed to the published version of the manuscript.

Funding: Parts of this research were funded by US AFOSR grant FA9550-18-1-0471.

Data Availability Statement: All data supporting this publication is held by UNSW in accordance with its Handling Research Material & Data Procedure.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Pipe-Lined Filter Implementation

The filters chosen for the SCoG algorithm are symmetric and separable, which is helpful to reduce the resource usage. For instance, $F_x(m,n)$ can be separated into two 1-D filters as following:

$$F_x(m,n) = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} -M & -M+1 & \dots & M \end{bmatrix} \tag{A1}$$

Similarly, $F_y(m,n)$ can be separated by:

$$F_y(m,n) = \begin{bmatrix} -M \\ -M+1 \\ \vdots \\ M \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \tag{A2}$$

The column vector is applied to each incoming column of pixels first, followed by a pipeline adder for the row vector. However, the multiple sum operations after the multiplication of the vertical vectors lead to complicated combination logic between two synchronous FFs and as a result, difficult timing closure. One solution to improve the timing performance of this path is to break down the operations to be completed in more clock cycles, i.e., a pipelined implementation as shown in Figure A1. Because the 1D filter $[-M\,-M+1\,\dots\,M]^T$ in Equation (A2) is also vertically symmetrical, it is possible to further reduce the number of multipliers and delay FFs by summing the symmetrical pixels first. Similar optimization can be applied to the horizontal 1D filter operation. However, as our target FPGA device has plenty of resources for our current implementation, we have not done such optimization. The output after the column vector, DV, is then passed to be multiplied by the horizontal vector.

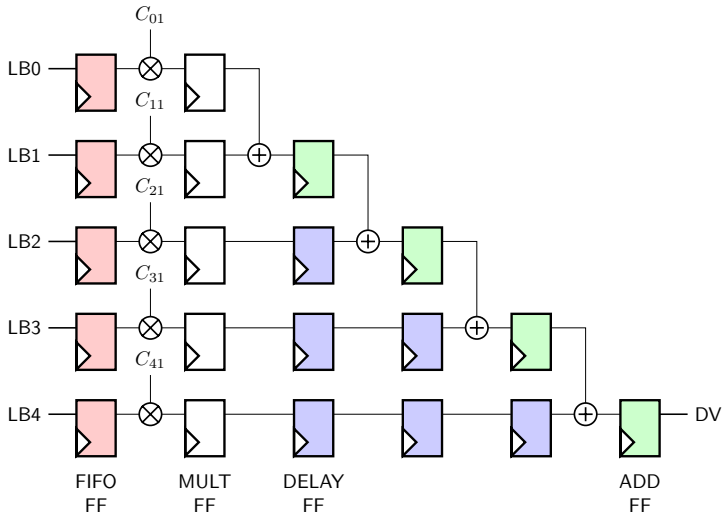

Figure A1. A fully pipe-lined design for the vertical filter operation. LB is the output of different line buffers and DV represents the output after applying the column vector.

The implementation of the filter with the horizontal vector is a bit different as the product of the previous vertical filter operation is present sequentially with each clock cycle. It is straightforward to think that we need to use registers to buffer the data, and then do the pipe-lined multiplication and addition as described for the vertical filtering. However, a better way, in terms of simplicity and resource usage, to implement this filter is to use the Transpose structure, which is a common technique in the design of finite impulse response (FIR) filter. As shown in Figure 4, the input DV, is multiplied by all the coefficients of the horizontal vector on each clock cycle. The results of multiplications are added and propagate through a chain of registers, with the result from the later coefficient at the furthest FF. The elegance of this Transpose structure, is that the adder chain can be easily placed and routed because it matches with the Xilinx FPGA structure, where arithmetic functions are placed in columns. In fact, the adder and register can be absorbed to the DSP48A slices which are used for the multiplication, which not only reduces the resource usage but also improves the speed.

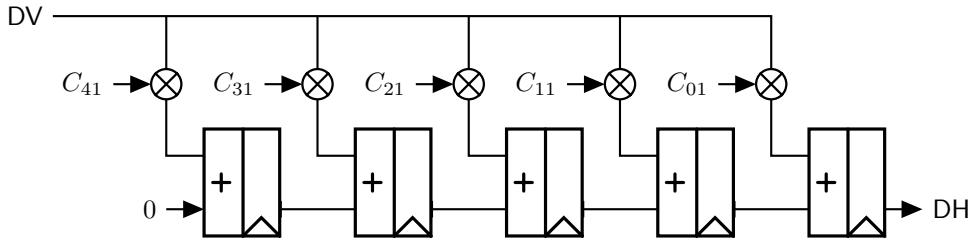

Figure A2. The horizontal filter implemented with a Transpose structure which takes the output of the column vector operation, DV, and produces DH in its output.

References

1. Platt, B.C.; Shack, R. History and Principles of Shack-Hartmann Wavefront Sensing. *J. Refract. Surg.* **2001**, *17*, S573–S577. [CrossRef] [PubMed]
2. Leroux, C.; Dainty, C. Estimation of Centroid Positions with a Matched-Filter Algorithm: Relevance for Aberrometry of the Eye. *Opt. Express* **2010**, *18*, 1197–1206. [CrossRef] [PubMed]
3. van Dam, M.A.; Lane, R.G. Wave-Front Slope Estimation. *J. Opt. Soc. Am. A* **2000**, *17*, 1319–1324. [CrossRef] [PubMed]
4. Barrett, H.H.; Dainty, C.; Lara, D. Maximum-Likelihood Methods in Wavefront Sensing: Stochastic Models and Likelihood Functions. *J. Opt. Soc. Am. A* **2007**, *24*, 391–414. [CrossRef]
5. Michau, V.; Rousset, G.; Fontanella, J. Wavefront Sensing from Extended Sources. In *Real Time and Post Facto Solar Image Correction*; Radick, R.R., Ed.; SAO/NASA Astrophysics Data System (ADS): online, 1993; p. 124.
6. Poyneer, L.A. Scene-Based Shack-Hartmann Wave-Front Sensing: Analysis and Simulation. *Appl. Opt.* **2003**, *42*, 5807–5815. [CrossRef]
7. Knutsson, P.A.; Owner-Petersen, M.; Dainty, C. Extended Object Wavefront Sensing Based on the Correlation Spectrum Phase. *Opt. Express* **2005**, *13*, 9527–9536. [CrossRef]
8. Thomas, S.; Fusco, T.; Tokovinin, A.; Nicolle, M.; Michau, V.; Rousset, G. Comparison of Centroid Computation Algorithms in a Shack–Hartmann Sensor. *Mon. Not. R. Astron. Soc.* **2006**, *371*, 323–336. [CrossRef]
9. Rousset, G. Wave-front sensors. In *Adaptive Optics in Astronomy*; Roddier, F., Ed.; Cambridge University Press: Cambridge, UK, 2004; Chapter 5, pp. 91–130.
10. Arines, J.; Ares, J. Minimum Variance Centroid Thresholding. *Opt. Lett.* **2002**, *27*, 497–499. [CrossRef]
11. Li, X.; Li, X.; Wang, C. Optimum Threshold Selection Method of Centroid Computation for Gaussian Spot. *Proc. SPIE* **2015**, *9675*, 967517. [CrossRef]
12. Nicolle, M.; Fusco, T.; Rousset, G.; Michau, V. Improvement of Shack-Hartmann Wave-Front Sensor Measurement for Extreme Adaptive Optics. *Opt. Lett.* **2004**, *29*, 2743–2745. [CrossRef]
13. Baker, K.L.; Moallem, M.M. Iteratively Weighted Centroiding for Shack-Hartmann Wave-Front Sensors. *Opt. Express* **2007**, *15*, 5147–5159. [CrossRef]
14. Ma, X.; Rao, C.; Zheng, H. Error Analysis of CCD-based Point Source Centroid Computation Under the Background Light. *Opt. Express* **2009**, *17*, 8525–8541. [CrossRef]
15. Yin, X.; Li, X.; Zhao, L.; Fang, Z. Automatic centroid detection for Shack-Hartmann Wavefront sensor. In Proceedings of the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, 14–17 July 2009; pp. 1986–1991. [CrossRef]
16. Spiricon. *Hartmann Wavefront Analyzer Tutorial*; Spiricon, Inc.: North Logan, UT, USA, 2004.

17. Bezzubik, V.; Belashenkov, N.; Soloviev, O.; Vasilyev, V.; Vdovin, G. Hartmann-Shack Wavefront Reconstruction with Bitmap Image Processing. *Optics Lett.* **2020**, *45*, 972. [CrossRef]
18. Li, Z.; Li, X. Centroid Computation for Shack-Hartmann Wavefront Sensor in Extreme Situations Based on Artificial Neural Networks. *Opt. Express* **2018**, *26*, 31675–31692. [CrossRef]
19. Hu, L.; Hu, S.; Gong, W.; Si, K. Learning-Based Shack-Hartmann Wavefront Sensor for High-Order Aberration Detection. *Opt. Express* **2019**, *27*, 33504. [CrossRef]
20. Lechner, D.; Zepp, A.; Eichhorn, M.; Gładysz, S. Adaptable Shack-Hartmann Wavefront Sensor with Diffractive Lenslet Arrays to Mitigate the Effects of Scintillation. *Opt. Express* **2020**, *28*, 36188. [CrossRef]
21. Talmi, A.; Ribak, E.N. Direct Demodulation of Hartmann-Shack Patterns. *J. Opt. Soc. Am. A* **2004**, *21*, 632–639. [CrossRef]
22. Mocci, J.; Busato, F.; Bombieri, N.; Bonora, S.; Muradore, R. Efficient Implementation of the Shack-Hartmann Centroid Extraction for Edge Computing. *J. Opt. Soc. Am. A* **2020**, *37*, 1548. [CrossRef]
23. Mompeán, J.; Aragón, J.L.; Prieto, P.M.; Artal, P. GPU-based Processing of Hartmann-Shack Images for Accurate and High-Speed Ocular Wavefront Sensing. *Future Gener. Comput. Syst.* **2019**, *91*, 177–190. [CrossRef]
24. Goodsell, S.J.; Fedrigo, E.; Dipper, N.A.; Donaldson, R.; Geng, D.; Myers, R.M.; Saunter, C.D.; Soenke, C. FPGA developments for the SPARTA project. *Proc. SPIE* **2005**, *5903*, 5903OG. [CrossRef]
25. Mauch, S.; Reger, J.; Reinlein, C.; Appelfelder, M.; Goy, M.; Beckert, E.; Tünnermann, A. FPGA-accelerated adaptive optics wavefront control. *Proc. SPIE* **2014**, *8978*, 897802. [CrossRef]
26. Mauch, S.; Barth, A.; Reger, J.; Reinlein, C.; Appelfelder, M.; Beckert, E. FPGA-accelerated adaptive optics wavefront control part II. *Proc. SPIE* **2015**, *9343*, 93430Y. [CrossRef]
27. Mauch, S.; Reger, J. Real-Time Spot Detection and Ordering for a Shack-Hartmann Wavefront Sensor with a Low-Cost FPGA. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 2379–2386. [CrossRef]
28. Thier, M.; Paris, R.; Thurner, T.; Schitter, G. Low-Latency Shack-Hartmann Wavefront Sensor Based on an Industrial Smart Camera. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 1241–1249. [CrossRef]
29. Kincses, Z.; Orzó, L.; Nagy, Z.; Mező, G.; Szolgay, P. High-Speed, SAD Based Wavefront Sensor Architecture Implementation on FPGA. *J. Signal Process. Syst.* **2011**, *64*, 279–290. [CrossRef]
30. Saunter, C.D.; Love, G.D.; Johns, M.; Holmes, J. FPGA Technology for High-Speed Low-Cost Adaptive Optics. *Proc. SPIE* **2005**, *6018*, 60181G. [CrossRef]
31. Cegarra Polo, M. Adaptive Optics For Small Aperture Telescopes. Ph.D. Thesis, The University of New South Wales, Sydney, NSW, Australia, 2015.
32. Mocci, J.; Quintavalla, M.; Trestino, C.; Bonora, S.; Muradore, R. A Multiplatform Cpu-Based Architecture for Cost-Effective Adaptive Optics Systems. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4431–4439. [CrossRef]
33. Kong, F.; Polo, M.C.; Lambert, A. Centroid Estimation for a Shack-Hartmann Wavefront Sensor Based on Stream Processing. *Appl. Opt.* **2017**, *56*, 6466–6475. [CrossRef]
34. Hardy, J. *Adaptive Optics for Astronomical Telescopes*; Oxford Series in Optical and Imaging Sciences; Oxford University Press: Oxford, UK, 1998.
35. Poyneer, L.A.; Gavel, D.T.; Brase, J.M. Fast Wave-Front Reconstruction in Large Adaptive Optics Systems with Use of the Fourier Transform. *J. Opt. Soc. Am. A* **2002**, *19*, 2100. [CrossRef]
36. Leroux, C.; Dainty, C. A Simple and Robust Method To Extend the Dynamic Range of an Aberrometer. *Opt. Express* **2009**, *17*, 19055–19061. [CrossRef]
37. Noll, R.J. Zernike Polynomials and Atmospheric Turbulence. *J. Opt. Soc. Am.* **1976**, *66*, 207–211. [CrossRef]
38. Lambert, A.; Cegarra Polo, M. Real-time algorithms implemented in hardware for centroiding in a Shack-Hartmann Sensor. In Proceedings of the Imaging and Applied Optics, Arlington, VI, USA, 7–11 June 2015; Optical Society of America: Washington, DC, USA, 2015; p. AOW3F.2. [CrossRef]