



AFRL-AFOSR-VA-TR-2022-0144

**Exploring the use of Event Based Imaging Sensors on Turbulence
Characterisation and Adaptive Optics**

**Cohen, Gregory
WESTERN SYDNEY UNIVERSITY
GREAT WESTERN HWY
WERRINGTON, NSW, 2747
AUS**

**05/11/2022
Final Technical Report**

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
Air Force Office of Scientific Research
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE 20220511		2. REPORT TYPE Final		3. DATES COVERED <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;">START DATE 20180815</td> <td style="width: 50%; border: none;">END DATE 20210814</td> </tr> </table>		START DATE 20180815	END DATE 20210814
START DATE 20180815	END DATE 20210814						
4. TITLE AND SUBTITLE Exploring the use of Event Based Imaging Sensors on Turbulence Characterisation and Adaptive Optics							
5a. CONTRACT NUMBER		5b. GRANT NUMBER FA9550-18-1-0471		5c. PROGRAM ELEMENT NUMBER 61102F			
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER			
6. AUTHOR(S) Gregory Cohen							
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) WESTERN SYDNEY UNIVERSITY GREAT WESTERN HWY WERRINGTON, NSW 2747 AUS					8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/AFOSR IOS	11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-AFOSR-VA-TR-2022-0144		
12. DISTRIBUTION/AVAILABILITY STATEMENT A Distribution Unlimited: PB Public Release							
13. SUPPLEMENTARY NOTES							
14. ABSTRACT This grant was highly successful in spite of experimental problems associated with the pandemic. The teams has constructed a very useful model that can simulate the operation of event-based sensors in both Shack-Hartman and Holographic wavefront sensors. Simulations show that both would benefit significantly from using these sensors over current CCDs that use deterministic and full-frame readout. On the experimental side, they have clearly demonstrated the advantage of using event-based sensors for the spot image analysis in Shack-Hartmann wavefront sensors, with an increase in speeds of up to 2 orders of magnitude shown to be possible. Further study with more optimized and updated version of the sensor would be warranted. The PIs generated 2 high quality papers with 2 more expected.							
15. SUBJECT TERMS							
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT SAR			
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U		18. NUMBER OF PAGES 59			
19a. NAME OF RESPONSIBLE PERSON GEOFFREY ANDERSEN					19b. PHONE NUMBER (Include area code)		

FINAL PERFORMANCE REPORT

AFOSR GRANT NUMBER FA9550-18-1-0471

Exploring the use of Event Based Imaging Sensors on Turbulence Characterisation and Adaptive Optics

**Western Sydney University
UNSW Canberra**

Principal Investigator: Gregory Cohen

Authors Fanpeng Kong
Gregory Cohen
Andrew Lambert

Reporting Period:
15 Aug 2020 - 14 Aug 2021

Contents

List of Figures	2
List of Tables	3
1 Summary	4
2 Introduction	5
3 Methods, Assumptions and Procedures	8
3.1 Simulation based approach	8
3.1.1 Simulation of DVS pixels	8
3.1.2 Simulation of wavefront sensors	9
3.2 Real-time processing of event data	10
3.3 Compact AO system with event-based SHWFS	14
3.3.1 System components	14
3.3.2 Proposal	14
3.3.3 Compact adaptive optics system	15
4 Results and Discussion	18
4.1 EBS in Shack-Hartmann WFS	18
4.1.1 Pre-processing of the events	18
4.1.2 Events-based tracker	19
4.1.3 Wavefront reconstruction from event-based spot estimation	21
4.1.4 Experimental results	22
4.2 EBS in Holographic Modal WFS	23
4.3 EBS in Curvature/Geometric WFS	25
5 Conclusion	26
6 Publications and presentations	28
7 References	29

List of Figures

1	Principle of the event-based image sensor used in this work.	5
2	(a) Sample frame from the slow-motion video and (b) time surface rendered from simulated events using IEBCS.	9
3	Left: Wavefront of Astigmatism 45°; Event-rendered frames simulated from SHWFS (left), Holographic Modal WFS (right) and Curvature/Geometric WFS (bottom).	10
4	Compact AO system using Thorlabs cage system and event-based Shack-Hartmann WFS.	14
5	AO system configuration with (a) 90 degrees and (b) small angle reflections from DM.	16
6	Deformable mirror and multiactuator adaptive lens.	16
7	Cage system AO with event-based Shack-Hartmann wavefront sensor.	17
8	Optical setup of the XSAO based on the MAL (a) optical layout and (b) picture of the assembled module.[1]	17
9	Event stream simulated from a rotating Gaussian spot.	19
10	Spot tracking with average position and Gaussian blob.	21
11	Reconstruction to 9 Zernike modes with input of Astigmatism.	23
12	Left: Experiment setup for evaluating Shack-Hartmann wavefront sensor with an event-based camera; Right top left: laser beam profile; Right top right: spots from APS frames; Right bottom: DAVIS 240C event camera.	24
13	2D histogram of recorded events from a spherical oscillation (a) raw data (b) after applying a background activity filter of 1 ms.	24
14	Comparison of global tilt using the averaged position tracker (ON events only) and APS frames.	25
15	Reconstructed spots from a holographic modal WFS encoded with 4 Zernike modes on APS frames (left) and DVS event rendered frames (right).	26
16	Reconstruction to 9 Zernike modes with input of Astigmatism.	27
17	Experiment setup for evaluating a Holographic Modal WFS with an event camera.	27

List of Tables

1	Comparison of DVS pixel simulators	9
2	Various C/C++ frameworks provide tools to build event-based algorithms . . .	12
3	Components used in compact AO system	15

1 Summary

This research report covers the work on “Exploring the use of Event based Imaging Sensors on Turbulence Characterisation and Adaptive Optics” between 15 August 2020 to 14 August 2021. It is the third annual performance report and also the final performance report for this AFOSR funded project starting from 07 January 2019.

Neuromorphic cameras contain specialized pixels that only generate data in response to changes in log-illumination around a setpoint that is variable for each independent pixel. The data is emitted in an asynchronous fashion, and only from pixels that detect a change of a specific magnitude controlled by a variable threshold on the device. This results in a lower power sensor that produces far less data than a conventional imaging device, and additionally allows for high-speed processing of events from the camera. Recent applications in space imaging have demonstrated the applicability of these cameras for space observing tasks through conventional optical telescopes. This work built on that success to investigate the applications and potential uses for these cameras for adaptive optics tasks.

The objective of this research is to investigate the potential and applications of event-based neuromorphic cameras for adaptive optic systems, specifically for use in small and large telescope configurations. These devices offer the potential for high-speed adaptive optic corrections using conventional equipment configurations and may also allow for new and innovative approaches to adaptive optics that make use of the spatiotemporal output of the cameras.

The research project aims to achieve the following objectives:

- Develop hardware, software, and algorithms to integrate an event-based camera into an existing adaptive optics configuration.
- Explore prime-focus telescope optics, including evaluating the evolution of the Point Spread Function (PSD) directly. Physical parameters of the turbulence in the optical path will be determined from the signal processing of the event-based sensor output.
- Explore the event based sensor in a plane ajar from the pupil of the telescope as in traditional curvature wavefront sensing, but unlike traditional systems that require two planes of the system to be imaged, we will explore the spatio-temporal variation at one plane only to extract the same information about the aberrations.
- Install and test the sensor as a part of a small amateur class telescope AO system.

During the past year, the continued work on the event-based Shack-Hartmann wavefront sensor includes looking for better event-based processing and tracking techniques to improve the quality of spots displacement estimation. The application of event-based cameras in adaptive optics system has also been expanded to other types of wavefront sensors like the holographic modal wavefront sensor, Curvature/Geometric wavefront sensors, with some preliminary results achieved. In addition, the build of a compact adaptive optics system using event-based Shack-Hartmann wavefront sensor has been started. Due to the high temporal bandwidth of the event stream, FPGA devices have been selected as the platform to implement the event processing and wavefront reconstruction and corrections etc.

This research has led to the development of a novel and high-speed adaptive optics system capable of operating at speeds well above conventional and existing equipment. Through leveraging the unique characteristics of event-based sensors, the project will attempt to develop new methods of performing adaptive optics tasks and estimating parameters of turbulence.

2 Introduction

This is the third annual performance report on the program, “Exploring the use of Event Based Imaging Sensors on Turbulence Characterisation and Adaptive Optics”, being carried out at the Western Sydney University (WSU) and University of New South Wales Canberra (UNSW Canberra). The purpose of this study is to undertake research in adaptive optics and advanced wavefront sensors by using revolutionary Event-Based image Sensors (EBS) and image processing techniques. This project involves laboratory-based experiments and on-sky observations based on the existing optical infrastructure from UNSW Canberra. Considering the high data rate from the event-based image sensor, real-time image processing techniques such as the use of Field Programmable Gate Array (FPGA) technology might be necessary, which will be part of the continuation of our previous research on fast and compact adaptive optical system implemented on FPGA devices.

The structure and function of a pixel in one type of event cameras, the Dynamic Vision Sensor [2], is shown in Figure 1. The DVS pixels work asynchronously from each other and each pixel continuously detects the relative brightness change. If the relative illumination change exceeds a pre-defined threshold, a positive or negative spike will be generated depending on the change is increase or decrease.

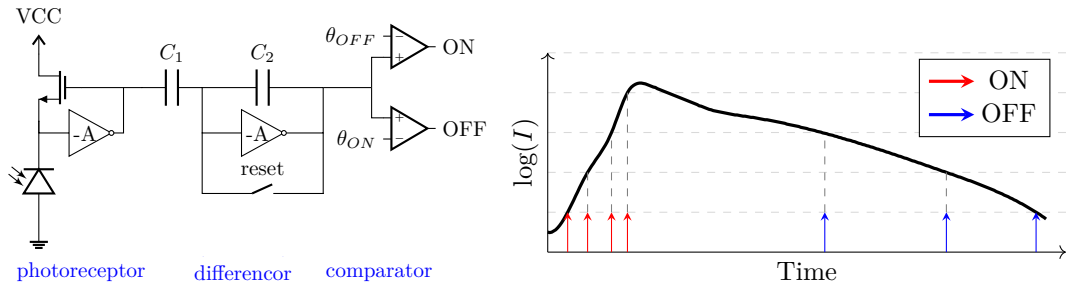


Figure 1: Principle of the event-based image sensor used in this work.

The output of the sensor can be described by a series of events $(e_k)_{k \in N}$, with each event represented by:

$$e_k = [\mathbf{u}, p, t]^T \quad (1)$$

where $\mathbf{u} = [x, y]$ denotes the pixel location generating the event, t is the timestamp when the event occurs and $p \in -1, +1$ is the polarity of the event indicating an increase (ON event) or decrease (OFF event) of the brightness. Naturally the event carries the temporal information, *i.e.*, derivative of brightness since the occurrence of the last event.

Compared with the conventional frame-based image sensors where all the pixels are integrated over a fixed exposure time and read out on the pixel clock cycles, the event-based image sensors only respond to the logarithmic changes of pixel values exceeding a threshold. The output therefore is a continuous stream of events tagged with the occurrence of the event location, timestamp and polarity (increase or decrease). This unique operation principle of the event-based image sensors offers various advantages over their counterpart frame-based sensors [3]: *high temporal resolution* to one microsecond; *low latency* because of the individual asynchronous operation of each pixel; *low power consumption* owing to the removal

of data redundancy found in frame-based image sensors; and *high dynamic range* due to the photoreceptors' operating on a logarithmic scale.

As the quality of event stream from event cameras often depends on several factors including camera settings, light condition and input scenes, a numerical simulation approach has been taken for the initial study of the event-based cameras in various wavefront sensors and can be used as a benchmark for the later experiments. The wavefront sensors are at the heart of any adaptive optics system to assess the distortion or aberration of an optical wavefront as optical phase of a wavefront cannot be directly sensed. Traditional WFS will integrate a camera frame and then analyse the spatial variations in their own unique way. This means they lose spatial-temporal changes that might spread the image information during the exposure. Such changes will require increase exposure times, and hence less temporal resolution in the wavefront estimation. The EBS provides these changes throughout the equivalent "exposure" time, and has the potential for much higher bandwidth WFS and AO as a result.

In this work, we explore the applications of event-based cameras in various Wavefront Sensors (WFS) and how the unique characteristics of the event cameras can help to improve the performance or overcome some challenges in these traditional wavefront sensors:

- Shack-Hartmann WFS

The output of the Shack-Hartmann WFS is sparse since only the sub-aperture spots (when used with point source) are useful for local slope estimation while the rest of the background does not carry useful information. Therefore the event camera only generates events aligned to the moving spots and yields higher bandwidth than conventional frame-based image sensors. The high dynamic range of the EBS can improve the photon budget problem that haunts sensors with large numbers of sub-apertures or small sized telescopes, and the low light level performance will provide subtle changes in the spot patterns.

- Holographic Modal WFS

In the holographic modal wavefront sensor (HMWFS), multiple superimposed holograms need to share the available dynamic range of the recording material (hologram film). Even with volume hologram recording, the diffraction efficiency of individual hologram decreases dramatically as it scales by α/M^2 where M is the number of holograms and α is a proportionality constant which depends on recording material and scheme [4]. Being a diffracting optical element, the narrow working bandwidth of the Holographic Modal WFS also makes it challenging to work under low light conditions. The EBS could potentially improve the applicability of the HMWFS in a similar way to the SHWFS as it is relatively independent of light level.

- Curvature/Geometric WFS

The event data naturally carries temporal derivative of the illumination and the spatial derivatives can also be retrieved by simple manipulations of adjacent events [5]. These information could provide new ways to solve the Poisson equation in the Curvature WFS [6], e.g. through the intensity transport equation. Similarly, by examining the statistics of the event data on the intra/extra planes in a Geometric WFS, it is also possible to solve the ray propagation problem and therefore measure the local slopes [7]. Diversity of planes in an optical system allow for the sampling of the Intensity Transport and consequently solving the Intensity Transport Equation for phase curvature. If

one can rely on the balanced changes occurring in two different planes of the optical system for this process, then two EBS should deliver opposing changes in each plane. In such as case, only one plane and only one EBS would be required.

3 Methods, Assumptions and Procedures

3.1 Simulation based approach

The quality of the event stream generated from an event-based camera is often dependant on various camera settings and raises uncertainty when evaluating different algorithms. The current manufacturing technology used for the event cameras also introduce limitations when exploring event cameras for conventional wavefront sensors. For example, the pixels from most event cameras have a very low fill factor around 22 % which might not be a problem for many computation vision tasks but certainly decreases the wavefront sensing from Shack-Hartmann or Curvature/Geometric wavefront sensors. In this work, we introduced numerical based approach to simulate the output of an event camera used in different wavefront sensors. Therefore, any event-based wavefront sensing data can be compared with its frame-based counterpart as well as the input ground truth. Later experimental results obtained from the real event camera can be analysed against the simulation results to find limitations imposed due to manufacture or hardware imperfections.

The numerical based approach can be described by following steps:

1. A phase screen generator generate varying wavefront, *e.g.* Kolmogorov phase screen or single Zernike modes.
2. The phase screens are passed to different wavefront sensors (Shack-Hartmann WFS, Holographic Modal WFS or Curvature/Geometric WFS) and the resultant frames from these WFS are stored.
3. Event-based simulators [8, 9] are used to generate event streams from the WFS frames in Step 2.
4. Wavefront sensing from event data is explored for different wavefront sensors using event data from Step 3.
5. Wavefront sensing from frames generated in Step 2 are used to compare with wavefront sensing from events data.

3.1.1 Simulation of DVS pixels

Although it has been more than 10 years since the first widely used event camera prototype DVS128 sensor being reported [2], the availability of event cameras still can not be compared to conventional image sensors and a wide adoption in industry applications is yet to be seen. Together with the lack of readily available event datasets, several pixel-level event-based simulators have been developed to meet the gap. Using a DVS pixel simulator also guarantees a reliable and reproducible datasets for initial exploration which is particular important for this research to assess the applicability of event cameras for various conventional wavefront sensing setups.

The ESIM simulator [8] is one of the widely used event camera simulators which converts the temporal light flux received by a pixel $F(t)$ to its logarithmic value $V_F(t) = \ln(F(t) + \epsilon)$. Linear interpolation between t and $t + dt$ is used to determine when the threshold is crossed. However in the ESIM simulator design, an event will be generated instantly from an infinite contrast which is in contrary to the physical DVS pixel. In addition, the latency does not depend on the sensor itself but rather on the rendering frame rate, *i.e.*, is a function of dt [10].

To address the shortcomings of ESIM, the v2e (video2events) toolbox introduces pixel-level Gaussian event threshold mismatch, finite intensity-dependent bandwidth and intensity-dependent noise [11]. The v2e simulator achieves these advancements by optimising the transfer function to match the behavior of a real sensor and using a slow-motion network interpolating $F(t)$ for a higher rendering rate.

The IEBCS simulator developed at ICNS WSU [10] further simplifies the latency and noise model by directly mapping noise distributions from real pixels. An arbiter in the readout circuit is also modelled to more closely simulate the behaviour of a real DVS pixel. A comparison of the modelling of a DVS pixel at different stages by these three simulators is summarised in Table 1:

Table 1: Comparison of DVS pixel simulators

Name	Scene	Log Conversion	Amplifier	Thresholds	timestamp	Arbiter
ESIM	Unreal/OpenGL	Linear Response	Delay	Gaussian	-	-
	SuperSiomo	Shot Noise	Leak Noise	Gaussian	-	-
V2E		Log. Response 2nd Order Filter				
IEBCS	Blender	Log. Response	Const. Latency	Gaussian	Gaussian	Bottleneck Synchronous
		1st Order Filter	Delay			

As an example of the IEBCS simulator, the National Geographic’s hummingbird slow-motion video is used to generate events. In Figure 2, an original frame from the video and time surface frame rendered from the simulated events using the IEBCS simulator are shown. (A time surface is a construct to integrate events so an traditional image may be formed. Note this integration time can be implemented arbitrarily after event capture.) In this study, the IEBCS event camera simulator is used to simulate events generated from varying wavefront observed from different wavefront sensor setups.



Figure 2: (a) Sample frame from the slow-motion video and (b) time surface rendered from simulated events using IEBCS.

3.1.2 Simulation of wavefront sensors

Simulation packages for wavefront sensors including Shack-Hartmann WFS, Holographic Modal WFS and Curvature/Geometric WFS have been developed through this research. In these simulation packages written in Python, a varying input wavefront, represented either by a combination of Zernike Polynomials or as a Kolmogorov phase screen, passes through optics system corresponding to different types of wavefront sensors and generate intensity images on the image sensors. For Shack-Hartmann wavefront sensor, the intensity image is

an array of spots formed by the micro lenslet array and represents downsampled Pupil point spread functions (PSFs). For Holographic modal wavefront sensor, intensity images of pairs of spots corresponding to system PSFs biased by individual Zernike Polynomials are generated on the image sensor plane. As in the Curvature/Geometric wavefront sensor simulation, intra and extra focal images axially separated from the focal plane are created. From these changing intensity images, events are generated using the IEBCS event camera simulation toolbox. In this simulation setup, wavefronts measured using event data can be cross compared with that reconstructed from the conventional intensity images. The ground truth data from the input wavefront is also available for assessment of wavefront sensing using event data.

In Figure 3, the wavefront of a Zernike aberration Astigmatism 45° is created to generate different frames on the Shack-Hartmann WFS, Holographic Modal WFS and Curvature/Geometric WFS. Simulated events from these frames are used to render the event frames for an illustration.

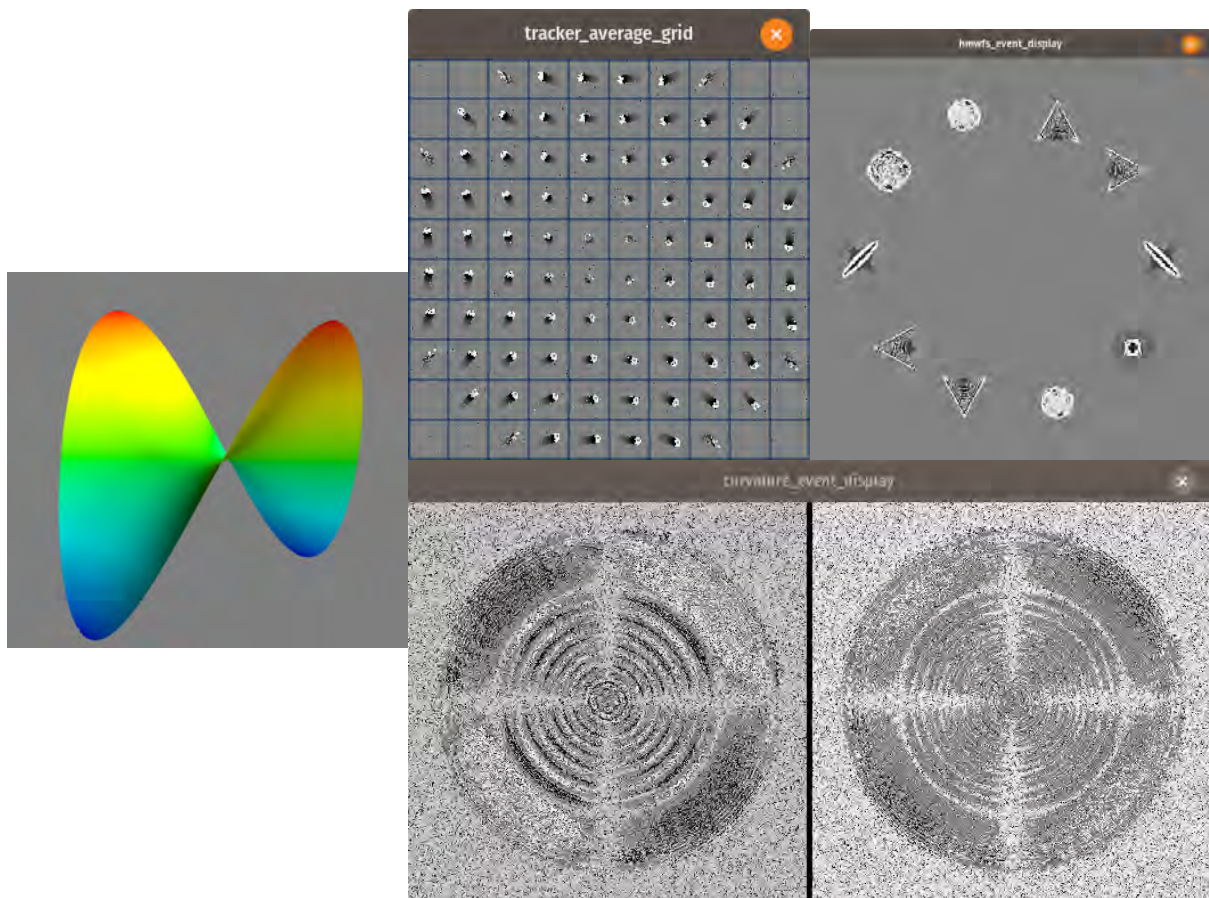


Figure 3: Left: Wavefront of Astigmatism 45° ; Event-rendered frames simulated from SHWFS (left), Holographic Modal WFS (right) and Curvature/Geometric WFS (bottom).

3.2 Real-time processing of event data

The output from event-based cameras are spatially sparse with high temporal resolution with sub-millisecond precision. This means that the current computer vision algorithms and frameworks, developed for conventional frame-based cameras which generate spatially dense images with relative low temporal speed, can not be used directly to process event data without modification of the event data. While it is possible to reconstruct frames from event data

and feed them into conventional frame-based computer vision frameworks, the asynchronous sparse nature of the event data is missed, and the potential computation overload of processing at such "frame rates" must be considered. Therefore specialised event driven algorithms need to be built from the ground up to better use the advantages of event cameras over conventional cameras.

Current event driven frameworks are mainly used in web servers and graphic user interface applications and they can handle up to several thousands events per second. The event cameras such as ATIS or DAVIS can generate events at a rate of several million events per second under a natural environment. Therefore current event-driven frameworks are not compatible with processing the high event rate from event based cameras and dedicated event-based computer vision frameworks need to be designed from scratch. Several existing frameworks dedicated to event-based computer vision have been developed by either event camera vendors or third party research institutions. For example, cAER and the Dynamic Vision (DV) System developed by iniVation are used to interface their DVS and DAVIS event cameras and allow users to create custom modules to process the event data. The kAER framework developed by Prophesee are used to control their Prophesee and ATIS event cameras. Tarsier, a modular, header-only C++ framework designed by Marcireau et al. [12] utilises static polymorphism using templates instead of static polymorphism as seen in other event-based frameworks. With its design methodology, event buffers are not required and latency also is decreased because the data pipeline is assembled during compile time. Together with the Sepia library for file IO, Chameleon library for display, Tarsier for collections of partial event handlers, these header-only libraries provide novel event-based frameworks which closely map operations to the nature of the output from event cameras. Table 2 highlights the design difference of several popular C/C++ based frameworks for processing event data.

In the Tarsier framework, functions are used to represent handlers, *e.g.* denoting f_* as a partial event handler. Since f_* generates events, it is an observable for a complete event handler g . Binding g to f_* yields the complete event handler f_g . When called, it performs the calculations associated with f_* , then calls g . Any number of partial event handlers can be chained to build an algorithm, as long as the last handler is complete. For example, with g_* now a partial event handler, and h a complete event handler, one can build the pipeline f_{g_h} . For each child, its direct parent is an observable generating events. For each parent, its child is a complete event handler (g_h is a complete event handler and a child for f_*). The syntax can be extended to partial event handlers generating multiple event types: $f_{*,*}$ is a partial event handler with two observable types [12].

In this work, the Sepia, Tarsier and Chameleon modular C++ framework is heavily used for conducting real-time processing on the event data generated from iniVation Davis cameras. The Tarsier code shown in Code 1 demonstrates a simple Tarsier pipeline bounded to a Sepia observable. `tarsier::make_replicate` first duplicates the event data read from an event file "input.es". The first event handler calculates the number of events per second while in the second event handler a background activity filter is applied first before the same event rate operation is performed.

Table 2: Various C/C++ frameworks provide tools to build event-based algorithms

Name	Open source	Operators connection	Dependencies	Communication and execution	Event types
Tarsier	Yes	Compile-time, C++ templates	-	Event-wise function calls, single thread	Template event types, contiguous memory
cAER	Yes	Run-time, XML	Boost, libpng, libusb, libuv	Event buffers, single thread	Hard-coded event types, contiguous memory
kAER	No	Run-time, C++/Python	Boost, OpenCV, Python, Qt	Event buffers, constant time intervals, single thread	Hard-coded event types, contiguous memory
Event-driven YARP	Yes	Run-time, C++/XML	Libace	IP packets, multiple programs	Hard-coded event types, contiguous memory or polymorphic event types, non-contiguous memory
Dynamic vision system	Yes	Run-time, XML	Boost, libusb, OpenCV, OpenSSL	Event buffers, multiple threads	Hard-coded event types, contiguous memory

Source Code 1: This program outputs event rate before and after applying a background activity filter (BAF).

```
1  #include "sepia.hpp"
2  #include "mask_isolated.hpp"
3  #include "replicate.hpp"
4  #include <iostream>
5
6  int main(int argc, char* argv[]) {
7      std::size_t count = 0;
8      uint64_t previous_t = 0;
9      std::size_t f_count = 0;
10     uint64_t f_previous_t = 0;
11     const auto header =
12         sepia::read_header(sepia::filename_to_ifstream("input.es"));
13     sepia::join_observable<sepia::type::dvs>(
14         sepia::filename_to_ifstream("input.es"),
15         tarsier::make_replicate<sepia::dvs_event>(
16             [&](sepia::dvs_event dvs_event) {
17                 ++count;
18                 if (dvs_event.t >= previous_t + 1000000) {
19                     std::cout << count << " events / second" << std::endl;
20                     previous_t = dvs_event.t;
21                     count = 0;
22                 }
23             },
24         tarsier::make_mask_isolated<sepia::dvs_event>(
25             header.width,
26             header.height,
27             1e3,
28             [&](sepia::dvs_event dvs_event) {
29                 ++f_count;
30                 if (dvs_event.t >= f_previous_t + 1000000) {
31                     std::cout << f_count << " events / second after BAF"
32                                     << std::endl;
33                     f_previous_t = dvs_event.t;
34                     f_count = 0;
35                 }
36             })
37         ));
38
39     return 0;
40 }
```

3.3 Compact AO system with event-based SHWFS

Currently, we are towards building a compact adaptive optics system using Thorlab's cage system and an event-based Shack-Hartmann wavefront sensor as shown in Figure 4.

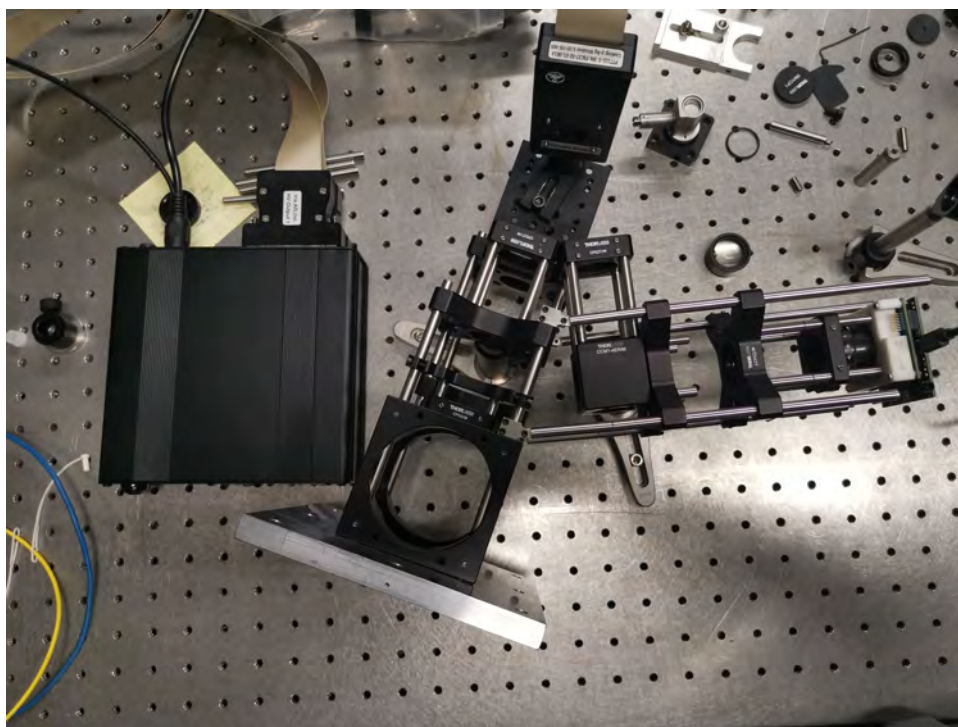


Figure 4: Compact AO system using Thorlabs cage system and event-based Shack-Hartmann WFS.

3.3.1 System components

The system components seen in Figure 4 draw on those available at UNSW Canberra. Principle components include an Iris AO PT111 deformable mirror and controller box, and Thorlabs microlens array, in addition to the Davis 240C EBS. The revised compact adaptive optics system, shown in Figure 7 employs components tabulated in Table ??, and interchanges the corrector from Dynamic Optics as a transmissive Deformable lens. The component not shown in either photo is the FPGA solution developed over a number of years at UNSW Canberra that controls the whole system with low-latency. This FPGA system is to be revised in future work to accommodate the events from the EBS rather than traditional image sensor frames. The FPGA system is documented in a draft journal paper attached to this report. The wavefront sensor is shown here as a Shack Hartmann WFS but the other types mentioned in this report may be easily interchanged.

3.3.2 Proposal

Adaptive optics (AO) systems for small sized ground based telescopes with a primary mirror ranging from 20 cm to 2 m often have different requirements and challenges compared to the complicated AO systems aiming at large size telescopes (1.5 m to 8 m). Small telescopes have to share the limited light budget between the wavefront sensor and science camera and result in bandwidth limited AO system due to the necessary longer exposure time. Using an event-based camera instead of a traditional frame-based CMOS or CCD camera in a Shack-Hartmann

Table 3: Components used in compact AO system

Items	Vendor	Model	Price (original)	Price (AUD)
Adaptive lens	Dynamic Optics	AOL 1810	9000 EUR	14676
Lens	Thorlabs	AC254-100-A-ML	106.05 USD	147
Lens	Thorlabs	AC254-100-A-ML	106.05 USD	147
Lens	Thorlabs	AC254-100-A-ML	106.05 USD	147
Lens	Thorlabs	AC254-030-A-ML	113.62 USD	157
Beamsplitter	Thorlabs	BS040	193.70 USD	268
Micro Lenslet Array	Thorlabs	MLA150-7AR-M	489.12 USD	677
Science camera	ATIK cameras	ATIK 414EX Mono	2899 AUD	2899
WFS camera	iniVation	Davis346	4900 EUR	7600
Cage system rods	Thorlabs	ER3	6.8 USD	9.4
Cage system rods	Thorlabs	ER3	6.8 USD	9.4
Sum				26736.8

wavefront sensor (SHWFS), the whole AO system will benefit from the high dynamic range and temporal bandwidth as well as reduced data redundancy from the event camera. In conjunction with a FPGA-based platform for the SHWFS and control system implementation, the resultant could be an ultra fast AO system suitable for small telescopes.

The other aspect of AO systems targeting at small telescopes is the necessary compactness and transportability. Although conventional wavefront correctors such as deformable mirrors (DMs) can provide high quality of wavefront shaping with various speed, the setup within a compact AO system are found challenging not to mention the often bulky electronics driving system. The emerging transmissive adaptive lenses [13] provide an alternative wavefront correction method to the deformable mirrors while delivering low order Zernike correction similar to COTS Deformable mirrors. As a result the optical configuration of the AO system could be significantly simplified.

3.3.3 Compact adaptive optics system

In Figure 5, two common optical setup for a single conjugate AO module suitable for small sized telescopes are shown.

The configuration shown in Figure 5a has 90° reflection from the DM and therefore does not introduce Astigmatism aberration due to asymmetrical reflection within the optics path. In addition, common commercial off-the-shelf parts, *e.g.* cage systems from Thorlabs, can be used to construct the opto-mechanical system because of the simple configuration. However, there is a 50 % light loose from the first beamsplitter (BS1) which makes this setup less favourable considering the already limited light budget.

The small angle reflection setup, shown in Figure 5b, can use all the light from the telescope but has extra astigmatism introduced to the science camera and the wavefront sensor. The opto-mechanical setup is also less flexible and rigid due to the miss of available parts to hold the DM in the small angle.

1. Deformable mirror and Multiactuator Adaptive lens

Multiactuator adaptive lens (MAL) is a particular type of transitive wavefront corrector based on piezoelectric actuators that deform two thin glass windows among which there

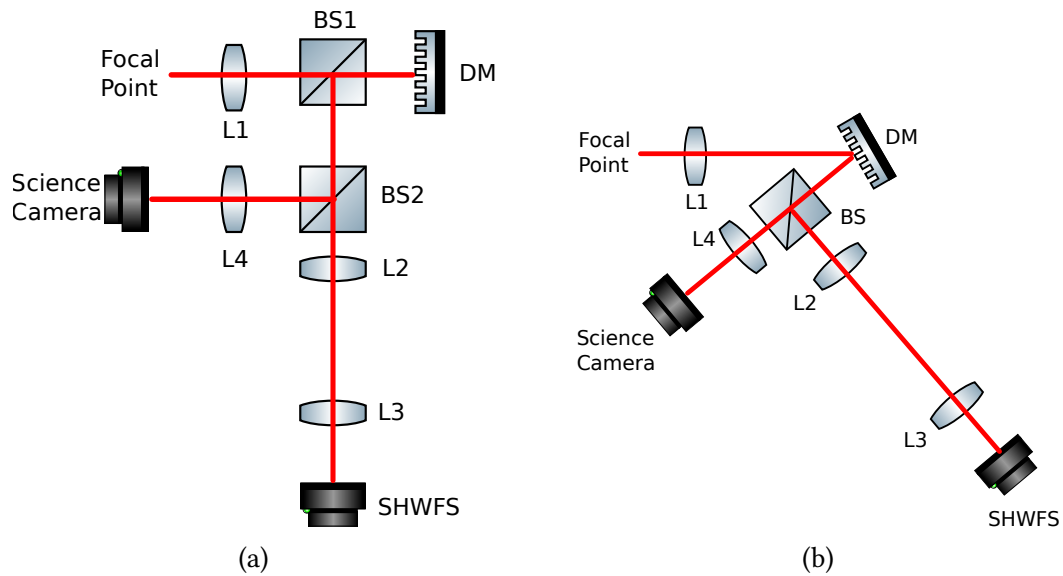


Figure 5: AO system configuration with (a) 90 degrees and (b) small angle reflections from DM.

is a refractive index-matching liquid [1]. The MAL shown in Figure 6b from Dynamic Optics can correct up to the 4th order of Zernike polynomial with a response time less than 5 ms. Compared to the PT111 deformable mirror from Iris AO, the transmissive MAL can simplify the compact AO system design greatly.

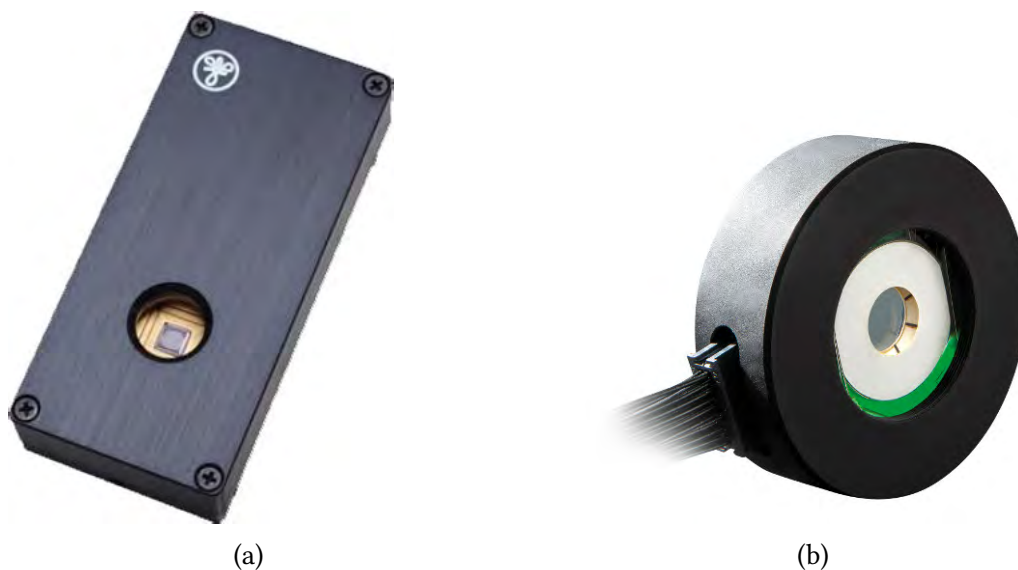


Figure 6: Deformable mirror and multiactuator adaptive lens.

2. Proposed design of the compact AO system

Currently, the compact AO system with an event-based Shack-Hartmann wavefront sensor is shown in Figure 7, where the PT111 DM is used in a small angle configuration. The black box shown on the optical bench is the electronics driver for the PT111 and it is almost the half size of the AO system itself.

If a multiactuator adaptive lens is used to substitute the PT111, the compact AO system

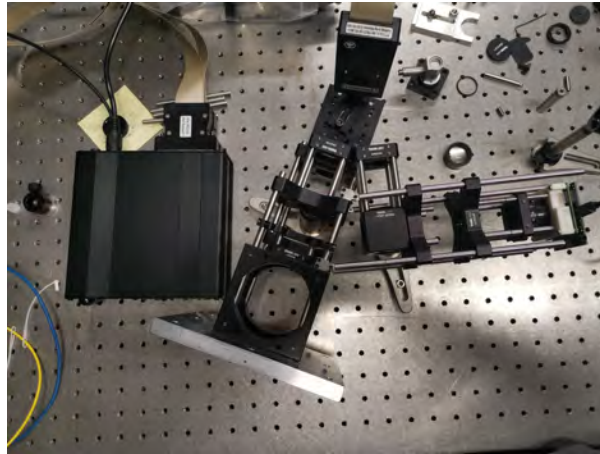


Figure 7: Cage system AO with event-based Shack-Hartmann wavefront sensor.

could be simplified similarly to the extremely small adaptive optics (XSAO) [1] as shown in Figure 8.

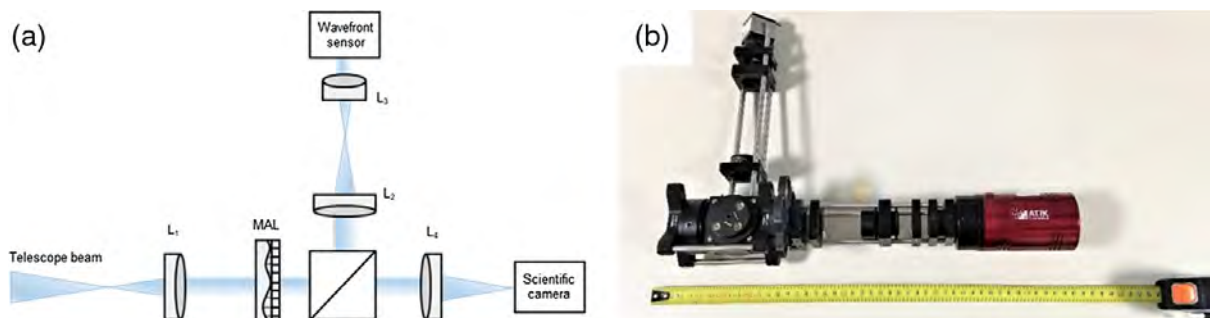


Figure 8: Optical setup of the XSAO based on the MAL (a) optical layout and (b) picture of the assembled module.[1]

4 Results and Discussion

In the past year, we have continued our study on applying the event-based camera in Shack-Hartmann WFS and expanded our work to Holographic Modal WFS and Curvature/Geometric WFS. The research progress in each type of wavefront sensors are detailed in the rest of this section.

4.1 EBS in Shack-Hartmann WFS

Through this study, we will investigate the usefulness of these event-based image sensors in the existing wavefront sensing techniques where currently traditional CMOS or CCD image sensors are exclusively used, and explore the potential benefits as well as hidden limitations under their controversial working principles. For example, in the Shack-Hartmann wavefront sensor (SHWFS), Center of Gravity (CoG) based centroiding methods or cross-correlation derived methods are widely used to determine the displacement of the spots pattern from the micro lenslet array (MLA). The event data however could provide an alternative more robust and less computational solution to measure these movements as the events are directly associated with the spatial changes of the spot location. The other example is in the Curvature and Geometric wavefront sensing where two image sensors located at the symmetrical *intra* and *extra* focal planes to measure the intensity redistribution as the wavefront propagates to estimate the local curvature of the wavefront. If event-based image sensors are used on the two focal planes, simplified manipulation of events data is possible to lead to similar curvature estimation but at a much higher temporal speed. The other direction of the study is to explore characterisation of the turbulence profile such as phase structure function and power spectrum parameters *etc.* directly from the event-based image sensor's output as each event carries the temporal derivative of the brightness.

4.1.1 Pre-processing of the events

In any event camera, noisy events can be generated from background activities such as thermal noise and junction leakage currents. These noisy events not only reduce the quality of the event stream but also occupies the bandwidth and unnecessary computation overload. The difference between background activity events and the real activity events is that the background activity event often lacks temporal correlation with events in their spatial neighborhood. Therefore, the background noisy events can be filtered out by removing events generated by a pixel without the spatiotemporal correlation with the events generated by neighboring pixels [14]. In Figure 9, the original event stream from a rotating Gaussian spot with a FWHM of 2 pixels is shown in the top left. There are large amount of noisy events generated outside the trace of the rotating spot. The event stream after applying the background activity filter is shown in the top right. It can be seen that a large portion of the background noisy events but not all have been filtered out. The residual noisy events are due to the spatiotemporal support from their neighbouring noisy events.

In [15], Afshar *et al.* proposed to use a time surface activation test as a generalized nearest neighbor filter where the radius can be selected according to the size of moving object instead of being a fixed 1 pixel. To start, a exponentially decayed time surface is updated with the arrival of each new event. On this time surface, the events withing a region of interest (ROI) are examined: if the number of activated events on this ROI is larger than a threshold, the ROI (together with the event) is accepted; otherwise, the ROI is rejected and the event is filtered

out. The advantage of this surface activation test over the background filter is that these parameters can be fine tuned to match the moving spot. The event stream after applying this surface activation test is shown in the bottom of Figure 9. Compared with the background activity filter, the residual events are much cleaner.

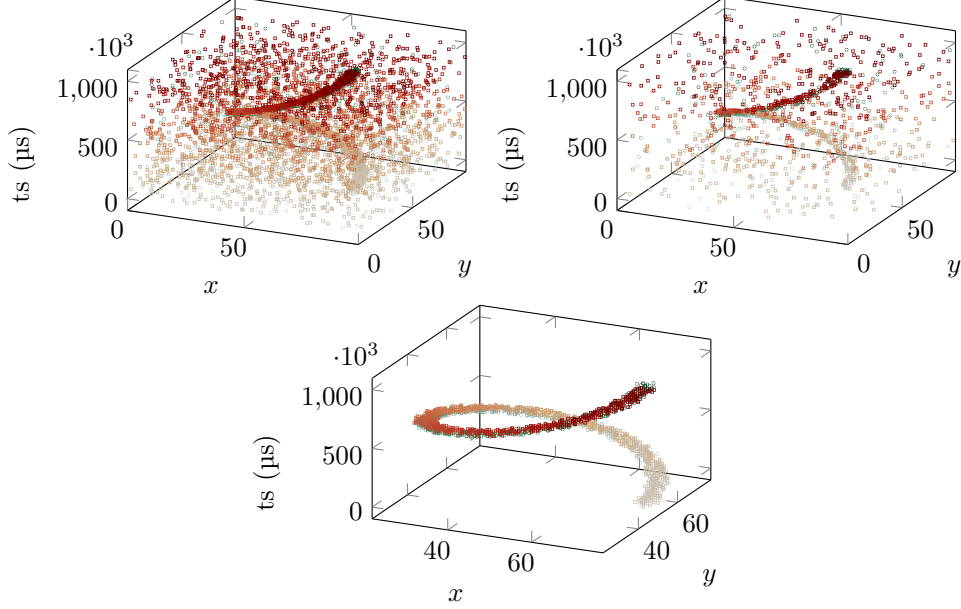


Figure 9: Event stream simulated from a rotating Gaussian spot. Top left: original event stream; Top right: event stream after applying a background activity filter; Bottom: event stream after applying the time surface activation test.

4.1.2 Events-based tracker

In this section, an event stream is generated from the rotating Gaussian spot using the WSU ESIM event-based simulator [9]. Two event-based trackers are examined to estimate the position of the Gaussian spot below.

1. Average position tracker

To track the spot movement using the generated events, we start with a simple algorithm by calculating the average position of the given events. Every time an increase event $\mathbf{e}_i = [x_i, y_i, t_i]$ is generated, the average position $(\hat{x}_{t_i}, \hat{y}_{t_i})$ at the event timestamp is updated:

$$\hat{x}_{t_i} = m\hat{x}_{t_{i-1}} + (1 - m)x_i \quad (2)$$

$$\hat{y}_{t_i} = m\hat{y}_{t_{i-1}} + (1 - m)y_i \quad (3)$$

where m is an inertia ranging from 0 to 1 and its value varies depending on the event rate from a specific application. It is worth noting that only the ON events are used for this estimation otherwise the trailing OFF events are likely to pull the average position back towards the location from which the spot came.

2. Gaussian blob tracker

The second tracking method used a Gaussian blob presented in [16]. The idea behind this method is that the cluster of generated events from a moving object represent the spatial distribution of the observed shape, which can be roughly approximated by a Gaussian distribution.

The parameters of the Gaussian blob $\mathcal{N}(\mu, \Sigma)$ consist of the object's position, size and orientation. The Gaussian mean μ and the covariance matrix Σ which denotes the size and orientation are updated when new events fall within a certain scope. The assignment of each new event to a potential tracker depends on a probability value $p_i(\mathbf{u})$ which can be calculated for each tracker $\mathcal{A}_i$:

$$p_i(\mathbf{u}) = \frac{1}{2\pi} |\Sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{u}-\mu_i)^T \Sigma_i^{-1} (\mathbf{u}-\mu_i)} \quad (4)$$

where μ_i and Σ_i represent the i th tracker's location and covariance matrix:

$$\mu_i = [\mu_{ix}, \mu_{iy}]^T \quad (5)$$

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (6)$$

The probability for all the trackers will be ranked from high to low with each incoming event and if the probability of tracker with the highest rank is larger than a predefined threshold δp , the position and size of this Gaussian tracker will be updated as follows:

$$\mu_t = a_1 \mu_{t-1} + (1 - a_1) \mathbf{u} \quad (7)$$

$$\Sigma_t = a_2 \Sigma_{t-1} + (1 - a_2) \Delta \Sigma \quad (8)$$

where a_1 and a_2 are update factors.

The covariance difference $\Delta \Sigma$ is calculated by:

$$\Delta \Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (9)$$

The activity of each tracker can be defined by the following:

$$\mathcal{A}_i(t) = \begin{cases} \mathcal{A}_i(t - \Delta t) \exp(-\frac{\Delta t}{\tau}) + p_i \mathbf{u} & \text{if } e(\mathbf{u}, t) \text{ belongs to tracker } i \\ \mathcal{A}_i(t - \Delta t) \exp(-\frac{\Delta t}{\tau}) & \text{otherwise} \end{cases} \quad (10)$$

Original application of the blob tracking method can use different kernels depending on the scene and object of interest but in our particular application, however, Gaussian blob is sufficient as they are good approximation to the local PSFs.

3. A comparison of the tracking performance

The tracking results from the average position tracker and the Gaussian blob tracker are shown in Figure 10. For the average position tracker, the tracking results using only ON

or OFF events and both ON and OFF events are compared. It is found that using only ON events for the average position yields the best tracking accuracy. The ON and OFF events do not always have the same noise characteristics and this is an accurate modeling of the actual hardware of the event-based image sensor. The decreased performance from averaging the OFF events can be attributed to the worse signal to noise (SNR) ratio from the OFF events. The Gaussian blob tracker use both the ON and OFF events and therefore the accuracy is also worse than the simple average position tracker with only ON events.

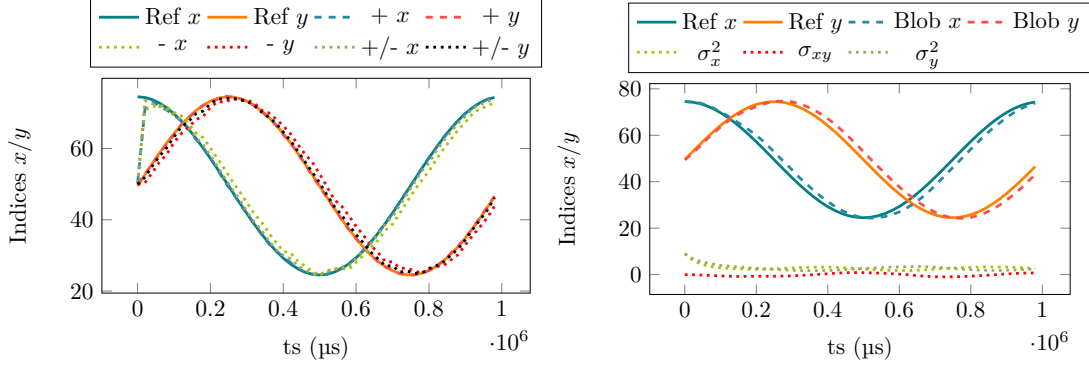


Figure 10: Spot tracking with average position and Gaussian blob.

4.1.3 Wavefront reconstruction from event-based spot estimation

Expanding the average position tracker or Gaussian blob tracker to a Shack-Hartmann WFS with $M \times M$ sub-apertures requires the same amount of individual trackers. Suppose $\mathcal{A}(r, c)$ where $r, c \in M$ is a sub-aperture, the corresponding tracker for this sub-aperture is denoted by $\mathcal{T}(r, c)$. For each incoming event $e[x, y, p, t]^T$, it is first sorted to its bounding sub-aperture according to $[x, y]^T$ and the number of pixels in each sub-aperture $w \times w$. Then the spots position for this sub-aperture will be updated using either the average position tracker or the Gaussian blob tracker. Essentially this means the update of local slope in each sub-aperture will be asynchronous and independent from each other, in the similar way to the event triggering from individual pixels. To use a conventional least-square fitting modal reconstructions, the latest slope values from all the sub-apertures at a fixed time interval t_{rec} are used. The time interval t_{rec} , or its reciprocal the reconstruction frequency f_{rec} no longer is limited by the frame rate found in traditional frame-based image sensors. Though the reconstruction frequency f_{rec} can increase up to the same temporal resolution as the event camera (about 1 MHz¹), the actual reconstruction frequency should math the temporal frequency of the turbulence, or more practically to the maximum speed of the deformable mirror. The reconstruction from event-based spot estimation is illustrated in Algorithm 1:

Using the numerical simulation approach described in 3.1, a Shack-Hartmann WFS with 10×10 sub-apertures and each sub-aperture consists of 30×30 pixels is simulated. The input aberration consists single mode of Zernike mode aberration with strength varying between $[-\pi : \pi : -\pi]$. 400 frames are generated for each aberration and the frames updates at a time interval of 1000 μs , resulting in a total of 0.4 ms simulation. The event stream are then generated from these 400 frames using the event-based simulator described in [9].

¹Notice that this limitation found in most of the event cameras is due to the hardware logic used to attach timestamps for event. The output of the event image sensor are not limited to this speed.

Algorithm 1: Wavefront reconstruction from event-based spot estimation

Data: event stream $e_k[x, y, p, t]^T$ **Result:** Wavefront $\mathbf{W}$

```
1 initialization;
2 foreach event  $e[x, y, p, t]^T$  do
3    $r := \text{floor}(x/w)$ ;
4    $c := \text{floor}(y/w)$ ;
5   update centroid position  $\hat{C}(r, c)$  using tracker  $\mathcal{T}(r, c)$  append  $\mathcal{C}$  to centroid list of
      $\mathcal{A}(r, c)$ ;
6   if  $t = \text{new } t_{rec}$  then
7     | Modal reconstruction using all  $\hat{C}(r, c)$ 
8   end
9 end
```

The events are then used to estimate the spot position in each sub-aperture using both the average and Gaussian blob tracker. The modal reconstruction is conducted on the same time interval as the original frames so the reconstruction from the event and frame-based estimations can be compared with each other. In Figure 11, the wavefront reconstruction from frames and events for a tilt aberration (Z_2) and Astigmatism 45° aberration (Z_5) is shown. The wavefront reconstruction from the Gaussian blob tracker has a large latency from the ground truth, *i.e.*, frame-based reconstruction, which could be attributed to the poor tracking accuracy from due to the OFF events. Average position tracker yields a better reconstruction consistent with the ground truth, although the trackers need a certain amount of events generated before they are updated to following the spots from their initial positions. A closer examination on the comparison between individual sub-aperture tracking between the frame and event-based estimation shows that the difference in the corner sub-apertures are much larger. This is due to the different intensity profile of the local PSFs in these sub-apertures which means the tracker parameters need to be optimized accordingly.

4.1.4 Experimental results

Currently, we are evaluating the application of event-based image sensor in a Shack-Hartmann wavefront sensor setup. In Figure 12, the experimental setup for capturing Shack-Hartmann spots on an event-based camera is shown. The Thorlabs DMP40 deformable mirror is used to introduce lower order aberrations such as Tip/Tilt and Defocus resulting in simple spots movement pattern like global shift, shrinking and expanding as pin cushion. The regular and structural movement helps the development and verification of event-based algorithm to measure the spots location in SHWFS. The Thorlabs WFS20 Shack-Hartmann WFS is used to verify the introduced aberration first and then as a comparison reference with the event-based SHWFS at a later stage.

The event camera DAVIS 240C includes both APS and DVS sensor on the same sensor array, and the histogram shown in Figure 13 are from the APS frames and DVS events during a test of horizontal DM oscillation. The red circle indicates the theoretical Pupil size based on the optical configuration. Noticeably the spots from the sub-apertures closer to the Pupil edge in the APS frames are much dimmer than those sub-apertures in the central area, due to the less preferred beam profile. However the DVS events generated in these edge sub-apertures are

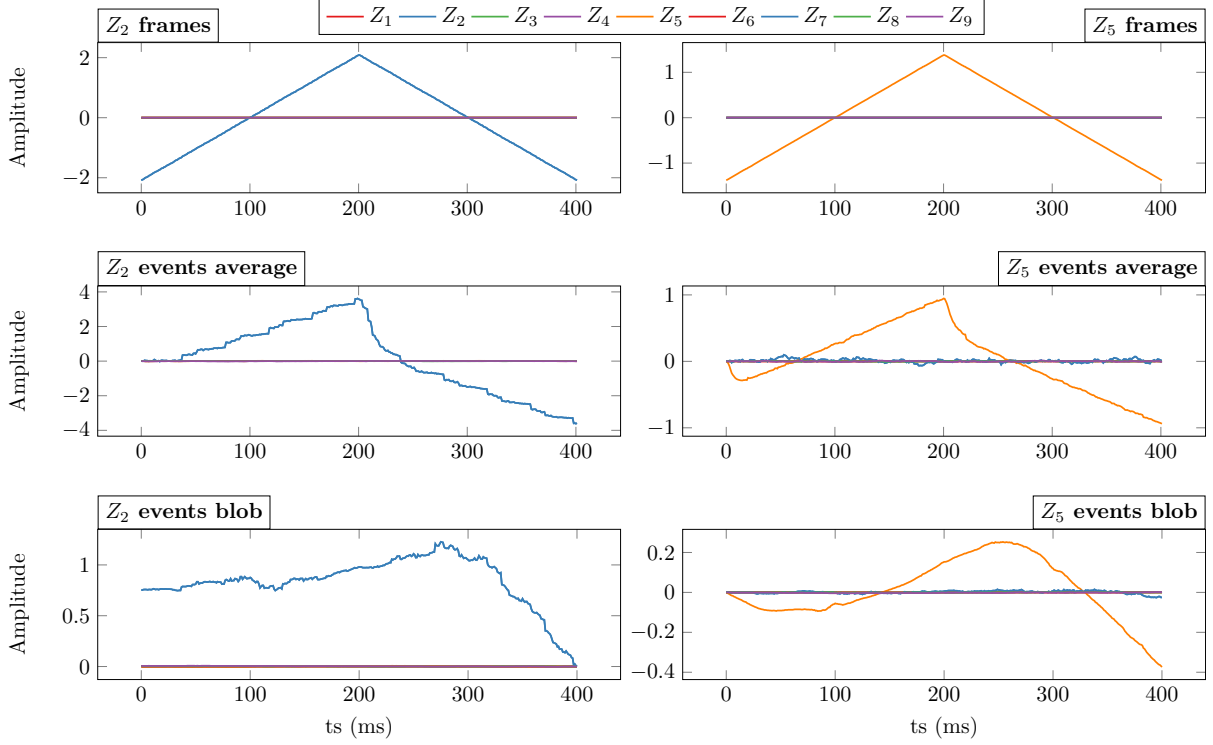


Figure 11: Reconstruction to 9 Zernike modes with input of Astigmatism.

much consistent with the central sub-apertures.

In Figure 14, the tilt estimations from events data using an average position tracker (ON events only) and APS frames are compared with each other. The DMP40 oscillated horizontally resulting in a global tilt in x direction. Depending on the value of inertia parameter, the estimation in y direction could become extremely noisy. The estimations from the frame and event data indicate the same number of cycles. The lag between the estimations are a result of mismatch between timestamps used for frames and events, which can be manually aligned upon calibration. It is worth to note that the pixels in DAVIS 240C have a fill factor of 22 %, comparing the absolute strength estimation is not considered currently. The important thing to note here is the event data provides more than 1000 estimations in one second while the conventional APS sensor data only yields about 12 frames of estimation in the corresponding time.

4.2 EBS in Holographic Modal WFS

Holographic modal wavefront sensor can directly measuring the strength of encoded Zernike modes in the multiplexed hologram by comparing the intensity different of the pairs of diffracted spots [17, 18, 19]. However, one of the challenges faced by Holographic modal wavefront sensors is the reduced diffraction efficiency as the number of multiplexed holograms increases. In addition, the HMWFS being a diffracting optical element is expected to work under narrow bandwidth which means not all the photon within the spectrum can be efficiently used. Event using volume hologram recording for the HMWFS, the diffraction efficiency for individual hologram can vary between $(2.2/M)^2$ and $(6.5/M)^2$ using the exposure schedule proposed by Pu *et al.* [4] for a photopolymer film. For example, in Figure 15, a HMWFS encoded with 4 Zernike modes (8 multiplexed holograms) is reconstructed with different diffraction efficiency

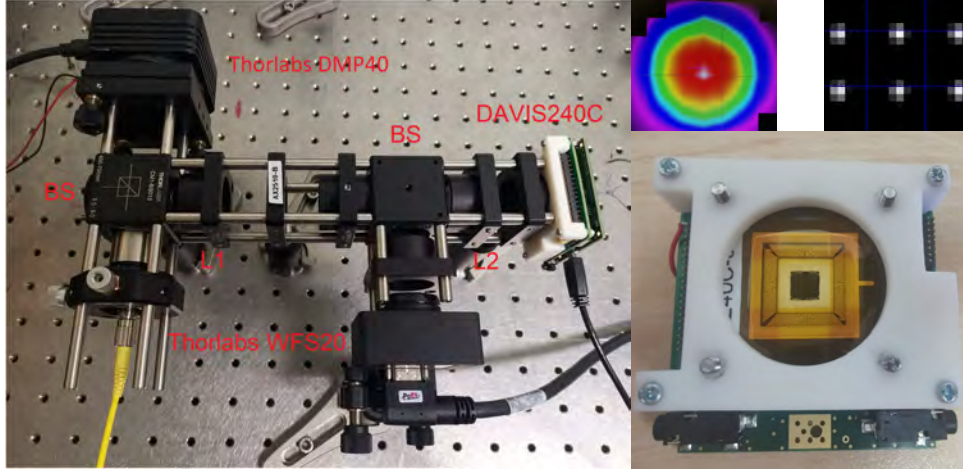


Figure 12: Left: Experiment setup for evaluating Shack-Hartmann wavefront sensor with an event-based camera; Right top left: laser beam profile; Right top right: spots from APS frames; Right bottom: DAVIS 240C event camera.

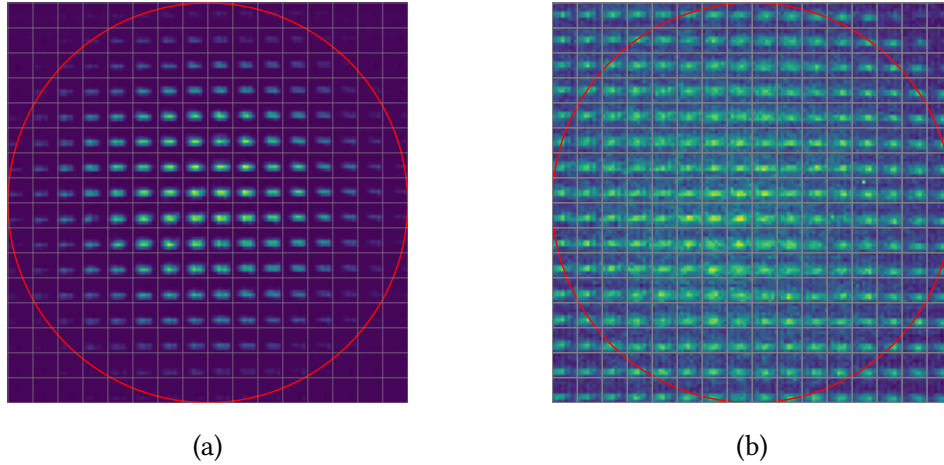


Figure 13: 2D histogram of recorded events from a spherical oscillation (a) raw data (b) after applying a background activity filter of 1 ms.

for each individual hologram. The later recorded holograms have much lower diffraction efficiency due to the insufficient dynamic range left within the hologram film. As a result, their diffracted spots can not be observed probably without saturating those with higher diffraction efficiencies. The DVS pixel however does not suffer from this trade-off as it only detects change, as shown in the right figure.

Using the numerical simulation approach described in Section 3.1, a holographic modal WFS encoded Zernike modes Z_4 to Z_8 is simulated. Single input aberrations including Z_4 Defocus, Z_5 Astigmatism 45° and Z_7 Coma Y with varying strength are used as input the Holographic modal WFS. An exponentially decayed time surface is generated from the simulated event stream, on which the weighted average intensity method [20] is used to measure the relative size of pairs of spots. The same HMWFS reconstruction method is applied on the frames as well to measure the unified strength of input aberrations. Comparison from the frame and event measurements is shown in Figure 16. The common discontinuities appeared in all three test at the same time stamps need to be further analyzed as they might be a result of the event

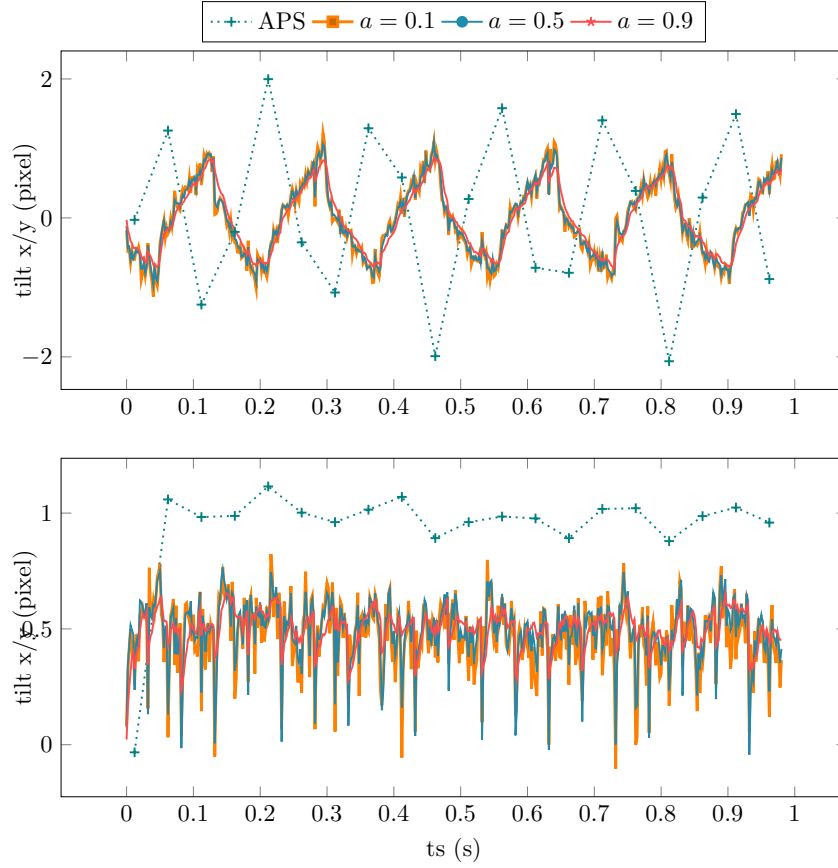


Figure 14: Comparison of global tilt using the averaged position tracker (ON events only) and APS frames.

simulator or the weighted average intensity calculation on the time decay surface need to be refined.

We have set up the following experiment in the lab for evaluating a Holographic Modal WFS with an event camera as shown in Figure 17. Evaluation of holographic modal WFS with an event camera is still underway and the focus will be qualifying how many holograms can be multiplexed on film with detectable output on the event camera. Inherent is all this work is the low-light level performance of the EBS. Preliminary results show the best performance is at extremely low-light, but we have yet to quantify exactly how low, and the interplay between sensor bias and reset settings and this performance.

4.3 EBS in Curvature/Geometric WFS

This section describes the work of event-based camera for Curvature and Geometric wave-front sensing. Due to evolving restrictions in Australia caused by the pandemic, access to the laboratory to proceed with this work has been interrupted. We endeavour to develop the preliminary simulation and findings from these WFS into a journal paper as soon as we can revisit the laboratory experiments. Preliminary measurements suggest the temporal derivative of the phase curvature can be easily extracted from the events, and correlation between changes in one plane and the inverse changes in the other are extremely good. This leads us to the realisation of using a single EBS in only one of the extra-focal planes, being sufficient for estimation of the changes to the phase curvature.

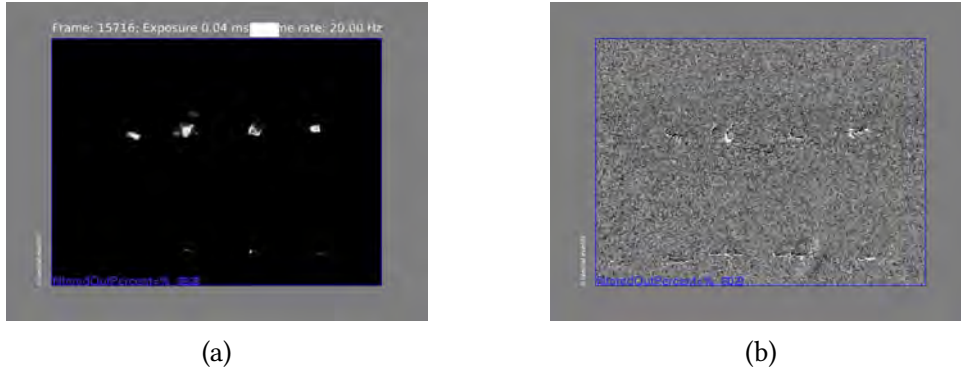


Figure 15: Reconstructed spots from a holographic modal WFS encoded with 4 Zernike modes on APS frames (left) and DVS event rendered frames (right).

5 Conclusion

To conclude, we have continued our investigation of applying an event camera in Shack-Hartmann wavefront sensor by examining different noisy filters and tracking algorithms to improve the accuracy of spot estimation from event data. Our experiments in the laboratory and on-sky using small (0.3m) telescopes showed an excellent tendency for the EBS to perform best at very low-light level. This is an added bonus for small optical systems and for more dense microlens arrays that would yield higher spatial resolution estimates of the wavefront. We also expanded our study of event camera in the Holographic modal WFS and Curvature/Geometric WFSs. A numerical simulation approach has been established as initial source for examining the event camera in conventional WFSs and as a benchmark for the later experiment results. We started to build a compact adaptive optics system using Thorlabs cage system and event-based Shack-Hartmann wavefront sensor. FPGA devices have been chosen as the platform for event acquisition, processing and control of the adaptive optics system.

In the next phase of this project, we will try to finish the construction of the compact AO system and close the AO correction loop by driving one of our deformable mirrors. This will require inciteful algorithm development to use the event streams within the FPGA architecture of the control system. The improvement of event camera to holographic modal WFS will be qualified. In addition, using event data for Curvature sensing or slope estimation in the Geometric WFS will be further explored. In preliminary experiments we also noticed that the low light-level behaviour of the event camera can yield information on the evolution of the Airy pattern of a point spread function and as such we also expect to derive turbulence related information directly from the likes of a stellar image when spatial resolution of the EBS improves.

In summary, this project has been successful in showing the EBS WFS can enhance the performance of traditional WFS, but algorithm development to further take advantage of the event streams in temporal evolution of the wavefront is yet to be fully examined. Through simulation, laboratory experiment and on-sky use of the EBS WFS we can continue the research. We fully appreciate there is much more to develop in terms of EBS-based WFS and performance of adaptive optics systems as a result.

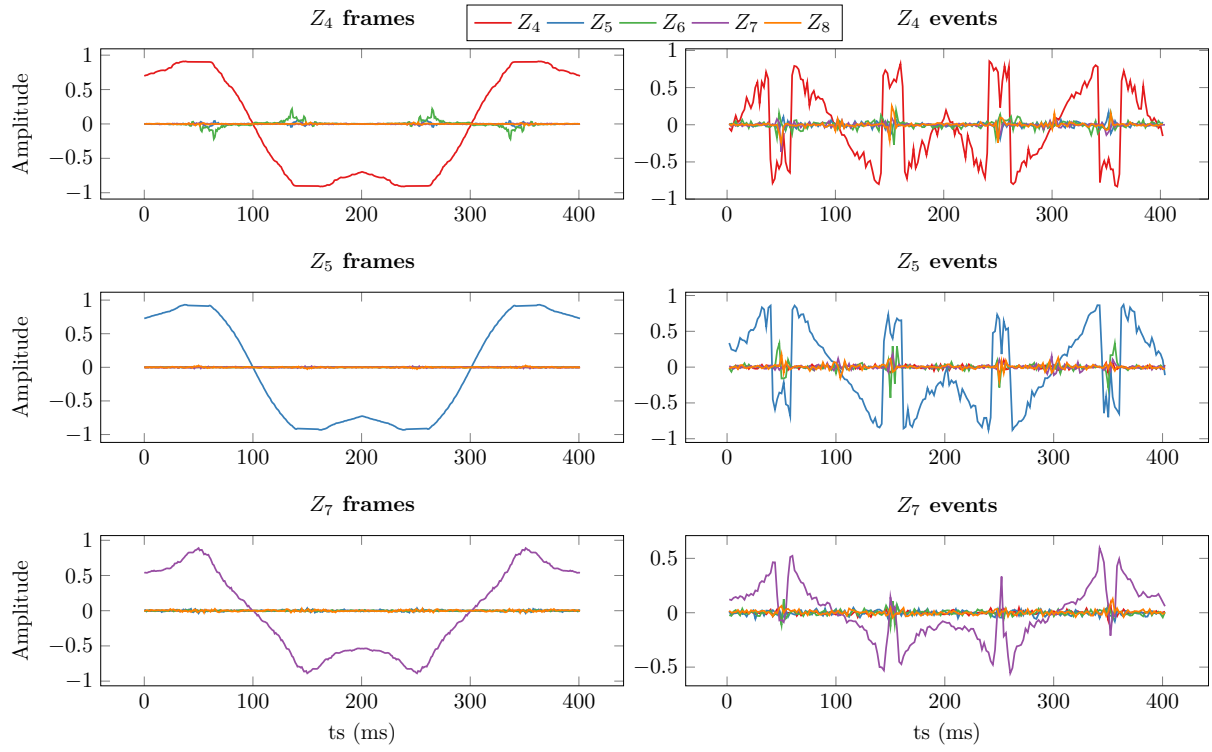


Figure 16: Reconstruction to 9 Zernike modes with input of Astigmatism.

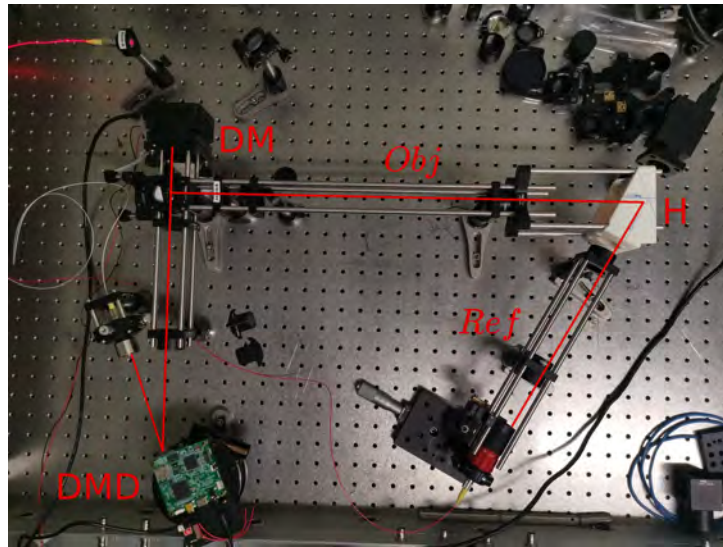


Figure 17: Experiment setup for evaluating a Holographic Modal WFS with an event camera.

6 Publications and presentations

Conference Publication

- Fanpeng Kong, Andrew Lambert, and Gregory Cohen. Shack-Hartmann wavefront sensing using spatial-temporal data from an Event-Based sensor. In *XII Workshop on Adaptive Optics for Industry and Medicine (AOIMXII)*, page nil, 10 2019
- Fanpeng Kong and Andrew Lambert. Holographic modal wavefront sensor performance in imaging systems with annular pupils. In *Imaging and Applied Optics 2020*, page AOM4C.4. Optical Society of America, 2020
- Andrew Lambert and Fanpeng Kong. Event-based sensing for direct turbulence parameter estimation. In *Imaging and Applied Optics 2021 (COSI, IS, MATH, pcAOP)*, page nil, Jul 2021

Journal Publication

- Fanpeng Kong, Andrew Lambert, Damien Joubert, and Gregory Cohen. Shack-hartmann wavefront sensing using spatial-temporal data from an event-based image sensor. *Optics Express*, 28(24):36159, 2020
- Fanpeng Kong, Manuel Cegarra Polo, and Andrew Lambert. FPGA implementation of shack-hartmann wavefront sensing using stream-based center of gravity method for centroid estimation. 2020

7 References

- [1] Martino Quintavalla, Mattia Spagnol, Lyu Abe, Marcel Carbillet, Eric Aristidi, Jacopo Mocci, Riccardo Muradore, and Stefano Bonora. XSAO: an extremely small adaptive optics module for small-aperture telescopes with multiactuator adaptive lens. *Journal of Astronomical Telescopes, Instruments, and Systems*, 6(02):1, 2020.
- [2] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, Feb 2008.
- [3] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-Based Vision: a Survey. *arXiv e-prints*, page arXiv:1904.08405, Apr 2019.
- [4] Allen Pu, Kevin Curtis, and Demetri Psaltis. Exposure schedule for multiplexing holograms in photopolymer films. *Optical Engineering*, 35(10):2824–2829, 1996.
- [5] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural networks : the official journal of the International Neural Network Society*, 27:32–7, 11 2011.
- [6] Francois Roddier, Claude Roddier, and Nicolas Roddier. Curvature sensing: a new wavefront sensing method. *Proc. SPIE*, 0976:203–209, 1988.
- [7] Theam Yong Chew. *Wavefront sensors in Adaptive Optics*. PhD thesis, University of Canterbury, 2008.
- [8] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 969–982. PMLR, 29–31 Oct 2018.
- [9] Damien Joubert. Design of a simulated event-based sensor to evaluate its potential for space applications. v1.02, 2020.
- [10] Damien Joubert, Alexandre Marcireau, Nic Ralph, Andrew Jolley, André van Schaik, and Gregory Cohen. Event camera simulator improvements via characterized parameters. *Frontiers in Neuroscience*, 15(nil):nil, 2021.
- [11] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events, 2021.
- [12] Alexandre Marcireau, Sio-Hoi Ieng, and Ryad Benosman. Sepia, tarsier, and chameleon: a modular c++ framework for event-based computer vision. *Frontiers in Neuroscience*, 13(nil):nil, 2020.
- [13] Stefano Bonora, Yifan Jian, Pengfei Zhang, Azhar Zam, Edward N. Pugh, Robert J. Zawadzki, and Marinko V. Sarunic. Wavefront correction and high-resolution in vivo oct imaging with an objective integrated multi-actuator adaptive lens. *Optics Express*, 23(17):21931, 2015.

- [14] Alireza Khodamoradi and Ryan Kastner. $O(n)$ -space spatiotemporal filter for reducing noise in neuromorphic vision sensors. *IEEE Transactions on Emerging Topics in Computing*, nil(nil):1–1, 2018.
- [15] Saeed Afshar, Andrew P Nicholson, Andre van Schaik, and Gregory Cohen. Event-based object detection and tracking for space situational awareness, 2019.
- [16] Xavier Lagorce, Cedric Meyer, Sio-Hoi Ieng, David Filliat, and Ryad Benosman. Asynchronous event-based multikernel algorithm for high-speed visual features tracking. *IEEE Transactions on Neural Networks and Learning Systems*, 26(8):1710–1720, 2015.
- [17] Mark A. A. Neil, Martin J. Booth, and Tony Wilson. New modal wave-front sensor: a theoretical analysis. *J. Opt. Soc. Am. A*, 17(6):1098–1107, Jun 2000.
- [18] F. Ghebremichael, G. P. Andersen, and K. S. Gurley. Holography-based wavefront sensing. *Applied Optics*, 47(4):A62, 2008.
- [19] Fanpeng Kong and Andrew Lambert. An improved holographic wavefront sensor. In *Imaging and Applied Optics 2016*, page AOM4C.4. Optical Society of America, 2016.
- [20] Fanpeng Kong and Andrew Lambert. Improvements to the modal holographic wavefront sensor. *Appl. Opt.*, 55(13):3615–3625, May 2016.
- [21] Fanpeng Kong, Andrew Lambert, and Gregory Cohen. Shack-Hartmann wavefront sensing using spatial-temporal data from an Event-Based sensor. In *XII Workshop on Adaptive Optics for Industry and Medicine (AOIMXII)*, page nil, 10 2019.
- [22] Fanpeng Kong and Andrew Lambert. Holographic modal wavefront sensor performance in imaging systems with annular pupils. In *Imaging and Applied Optics 2020*, page AOM4C.4. Optical Society of America, 2020.
- [23] Andrew Lambert and Fanpeng Kong. Event-based sensing for direct turbulence parameter estimation. In *Imaging and Applied Optics 2021 (COSI, IS, MATH, pcAOP)*, page nil, Jul 2021.
- [24] Fanpeng Kong, Andrew Lambert, Damien Joubert, and Gregory Cohen. Shack-hartmann wavefront sensing using spatial-temporal data from an event-based image sensor. *Optics Express*, 28(24):36159, 2020.
- [25] Fanpeng Kong, Manuel Cegarra Polo, and Andrew Lambert. FPGA implementation of shack-hartmann wavefront sensing using stream-based center of gravity method for centroid estimation. 2020.



Shack-Hartmann wavefront sensing using spatial-temporal data from an event-based image sensor

FANPENG KONG,^{1,2,*}  ANDREW LAMBERT,¹ DAMIEN JOUBERT,² AND GREGORY COHEN²

¹SEIT, University of New South Wales, Canberra, ACT 2612, Australia

²ICNS, MARCS, Western Sydney University, Sydney, NSW 2747, Australia

*f.kong@adfa.edu.au

Abstract: An event-based image sensor works dramatically differently from the conventional frame-based image sensors in a way that it only responds to local brightness changes whereas its counterparts' output is a linear representation of the illumination over a fixed exposure time. The output of an event-based image sensor therefore is an asynchronous stream of spatial-temporal events data tagged with the location, timestamp and polarity of the triggered events. Compared to traditional frame-based image sensors, event-based image sensors have advantages of high temporal resolution, low latency, high dynamic range and low power consumption. Although event-based image sensors have been used in many computer vision, navigation and even space situation awareness applications, little work has been done to explore their applicability in the field of wavefront sensing. In this work, we present the integration of an event camera in a Shack-Hartmann wavefront sensor and the usage of event data to determine spot displacement and wavefront estimation. We show that it can achieve the same functionality but with substantial speed and can operate in extremely low light conditions. This makes an event-based Shack-Hartmann wavefront sensor a preferable choice for adaptive optics systems where light budget is limited or high bandwidth is required.

© 2020 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Event-based image sensors are novel bio-inspired vision sensors which mimic the function of human eyes and work radically differently from conventional frame-based CCD or CMOS image sensors. While all the pixels in a traditional image sensor produce a linear representation of the illumination over the same fixed exposure time, the event-based image sensor only presents events on those pixels detecting an illumination change which exceeds a threshold [1,2]. The per-pixel detection of brightness change operates individually and asynchronously among all the pixels in an event-based image sensor. Advantages of event-based image sensors over traditional frame-based image sensors include reduced data redundancy, very high temporal resolution, low latency and low power consumption [3]. Event cameras also have a higher dynamic range, similar to logarithmic cameras, as they work on the logarithmic response region of the photodiode to detect the relative illumination change. As event-based image sensors only provide output from activated pixels, output data is fast and sparse. External algorithms can make better sense of this information than deciphering accumulated events over a fixed integration period.

The unique characters of event cameras have led to their adoption in many real-time interaction systems such as robotics or wearable electronics [4]. Event-based vision is a growing research field and various algorithms utilizing the event data have been proposed or adapted to execute common computer vision tasks such as feature detection and tracking, optical flow estimation [5–7], 3D reconstruction and pose estimation, motion segmentation and recognition [3], to name a few. Recent efforts of extending the usage of event cameras in astronomy include attempts to

detect and track space objects (LEO and GEO) during both day and nighttime [8,9] and to build real-time and continuous event-based astrometry system [10]. The unique characteristics of these event-based sensors promise alternative solutions to characterize the turbulence and wavefront sensing at an expected high speed in adaptive optics and wavefront sensing regime [11]. However, regardless of the increased interest for using event camera in computation field, little work has been done to explore the usage of event-based image sensors in the fields of wavefront sensing and adaptive optics.

Early event cameras like the Dynamic Vision Sensor (DVS) [1,12] respond to the brightness changes in the scene and produce a stream of asynchronous events using an Address-Event Representation (AER). The later improved Dynamic and Active Pixel Vision Sensor (DAVIS) can output traditional image frames as well as events by combining a conventional Active Pixel Sensor (APS) with DVS in the same pixel [13,14]. A different approach to measure the absolute intensity is taken by the Asynchronous Time Based Image Sensor (ATIS) [2] where an exposure measurement circuit will be triggered by an active change detection circuit contained in the same pixel. Therefore, the event-based algorithms can be sorted into two broad categories: one relying only on event data and one that uses a combination of event and frame (or gray scale data from the ATIS) data.

In a Shack-Hartmann wavefront sensor [15,16], a micro lenslet array spatially samples the input wavefront into individual sub-apertures. Local wavefront tilts result in the focal spots being displaced from their reference positions. In Shack-Hartmann wavefront sensors where traditional frame-based cameras are used, center-of-gravity (CoG) derived methods [17–19] are commonly used to determine the position of the spots for point-like objects. By computing the displacements of all spots from valid sub-apertures, local wavefront slopes are determined. From the discrete wavefront slopes, a continuous wavefront can be reconstructed using zonal or modal methods [20]. By integrating an event-based camera into a Shack-Hartmann wavefront sensor, the advantages of event-based image sensors over frame-based image sensors can be exploited in the context of wavefront sensing which can lead to a system with low latency, high bandwidth and better performance under low light conditions.

In this paper, we explore the application of event cameras in a Shack-Hartmann WFS by using the event data to directly estimate the spot displacement. Two event-based tracking algorithms are compared to track the movements of individual spots at the focal plane of a Micro Lenslet Array (MLA). The first method uses the average position of events over a user defined time period within each sub-aperture to provide an estimation of the spot movement, while the second method relies on a Gaussian blob tracker [21] to monitor the spot's position, size and orientation. Both trackers update the position of spots when new event is generated. The sub-aperture centroid updates asynchronously at a very high temporal speed similar to the event stream. The estimation of spot displacement is purely based on the events data and does not rely on absolute illumination information. For comparison, the synchronous image frames generated from the APS on the same pixel are used to calculate the spot location with traditional center-of-gravity method.

Three are several features of the current available event cameras from major vendors (iniVation AG, Prophesee), having large pixel size (from 1840μm), limited resolution (normally less than VGA) and small fill factor (around 20%) that could possibly be limitations or enablers. The fidelity of the output event stream under the same varying scene also heavily depends on the various bias settings of the event camera and adjusting these is not always a trivial task. The small fill factor will certainly impact the accuracy and consistency of the spot tracking and the following wavefront reconstruction. For these reasons, we use an event camera simulator to benchmark the performance of the estimated spot position from event data in comparison with that derived from image frames, but to verify the proposed event-based Shack-Hartmann wavefront sensor design, laboratory experiments are carried out using an iniVation DAVIS 240C event camera.

The paper is organized as following: first, the operation principles of event-based image sensors are reviewed in Section 2. In Section 3, the two event-based spot tracking algorithms are compared with the ground truth using an event camera simulator. Then the wavefront reconstruction from the asynchronously updated sub-apertures is explained. The experimental evaluation of a custom Shack-Hartmann WFS using an event camera is presented in Section 4. The paper ends with a discussion and conclusion in Section 5.

2. Principle of event-based image sensors

Each DVS pixel includes a fast logarithmic photoreceptor circuit, a differencing circuit which amplifies changes precisely and two-transistor comparators [1], as shown in Fig. 1. The inverting amplifier in the photoreceptor stabilizes the voltage across the photodiode and outputs a voltage proportional to the log of the light intensity. The capacitive-feedback inverting amplifier in the differencing circuit is balanced with a reset switch which shorts its input and output, effectively rejecting the absolute level of the illumination and outputting a relative change. The comparators compare the change with a global threshold and if the change exceeds a threshold, an event is triggered and transmitted from the chip to its peripheral bus. Normally the event data is tagged with the spatial location of the pixel (x and y), a timestamp t and the polarity of the change (ON or positive for increase and OFF or negative for decrease).

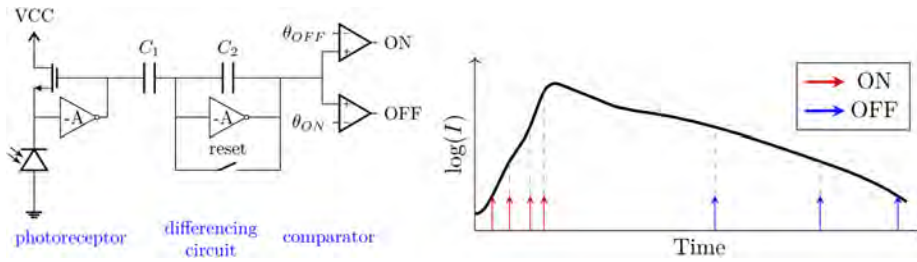


Fig. 1. Abstracted DVS pixel diagram and principle of its operation [1].

The series of events e_i generated by the event-based image sensor can be mathematically described by:

$$\mathbf{e}_i = [\mathbf{u}_i, t_i, p_i]^T \quad (1)$$

where i is the index of the event, $\mathbf{u}_i = [x_i, y_i]^T$ represents the spatial location of the pixel triggering the event, t_i is the time when the event occurred and $p_i \in \{-1, +1\}$ is the polarity of the event indicating an increase (ON event) or decrease (OFF event) of the brightness.

Event-based algorithms which operate on a event-by-event basis require history of recent events which can be generated by various index or time-based surfaces as reported in [22]. Following the notation in [9], an exponentially-decayed event time surface can be computed by the following steps. A matrix $\mathbf{T}_i$ contains the timestamps of the most recent events at each pixel and is initialized to $-\infty$. Each time a new event $\mathbf{e}_i$ is generated, the timestamp at the corresponding pixel $\mathbf{u}_i$ is updated by t_i in matrix $\mathbf{T}_i$ while the timestamps at other locations are unchanged, as described by Eq. (2):

$$\forall \mathbf{e}_i, \forall \mathbf{u}_k \in \mathbf{U} : \mathbf{T}_i(\mathbf{u}_k) = \begin{cases} \mathbf{T}_{i-1}(\mathbf{u}_k) & \mathbf{u}_k \neq \mathbf{u}_i \\ t_i & \mathbf{u}_k = \mathbf{u}_i \end{cases} \quad (2)$$

Defining $\mathbf{U}$ as the matrix of all the pixels of the sensor and k is the index of the pixel. The exponentially-decayed time surface $\mathbf{S}_i$ is calculated by Eq. (3):

$$\forall \mathbf{u}_k \in \mathbf{U} : \mathbf{S}_i(\mathbf{u}_k) = e^{(\mathbf{T}_i(\mathbf{u}_k) - t_i)/\tau} \quad (3)$$

Here τ is a decay constant of the time surface which defines of “lasting effectiveness” of an event on the following events. All the values in matrix S_i are initialized to 0. It is worth noting that the timestamp matrix T_i and exponentially-decayed time surface S_i have the same resolution as the sensor’s pixel array but they both update per new event.

3. Event-based Shack-Hartmann wavefront sensing

The first step to evaluate the applicability of an event-based camera for a Shack-Hartmann WFS is whether the spot location can be precisely and consistently measured from the event data. For this purpose, an event camera simulator, developed at ICNS (Western Sydney University) which models the event generation better than the reported ESIM [23] under high contrast situations [24], is used to generate the events from a rotating Gaussian spot (with a FWHM of 2 pixels), as shown in Fig. 2(a). The histogram of all the generated events from one cycle of the rotation is shown in Fig. 2(b).

Event-based image sensors often output a significant number of noise events besides the real activity generated events due to the inherent shot noise in photons and thermal noise. The event camera simulator also models the behaviour of this inherited limitation of DVS pixels, as can be seen by the noisy events outside the track of the rotating spot in Fig. 2. It is therefore common practise to filter these noisy events before using the event data for tracking or classification tasks. Advanced machine learning or neural network based filtering algorithms can reduce the noisy events dramatically while retaining the events related to signal [25–27].

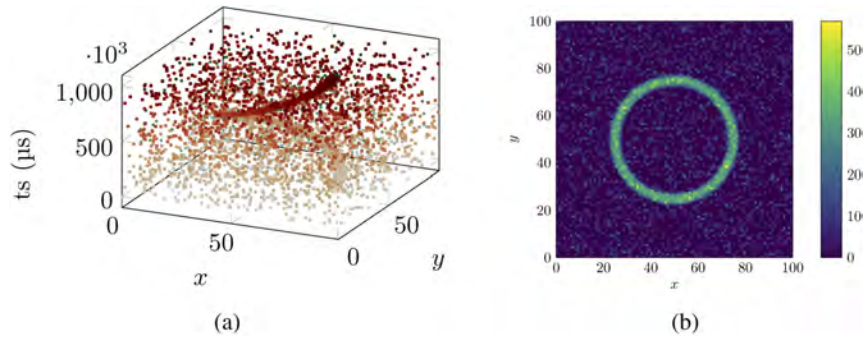


Fig. 2. (a) Event stream simulated from a rotating Gaussian spot; (b) Histogram of the generated events within the 1 s simulated rotation cycle.

The determination of an event stream from an observed scene is not always straightforward as the metric will mostly depend on the application of the event camera. For instance, in [9] where event cameras are used in space imaging applications to track satellites, three parameters (*Sensitivity*, *Specificity* and *Informedness*) are measured based on relative event density. These measurements require manually labelling the trajectory of the tracked satellite objects. For the simulation in this work, since the path of the rotating Gaussian spot is known, we can measure the quality of the generated event stream by the normalized difference between the true and noisy events Q . This quality is measured separately for the ON (Q_P) and OFF (Q_N) events, as described by Eq. (4):

$$Q_P = (TP - FP)/(TP + FP) \quad (4a)$$

$$Q_N = (TN - FN)/(TN + FN) \quad (4b)$$

where TP and TN are the numbers of true ON and OFF events on the trace of the rotating Gaussian spot which represent those genuine events from the rotating activity. Similarly, FP and FN are the numbers of false ON and OFF events outside the rotation trace denoting background

noise events. The event stream shown in Fig. 2 has a $Q_P = 0.42$ and $Q_N = -0.28$ respectively before applying any filters. Negative Q_N means that there are more false OFF events generated from background noise than true OFF events generated by the rotating Gaussian spot.

3.1. Event pre-processing

Background activity filter (BAF) is widely used to remove uncorrelated events generated by background activities caused by the transistor switch leakage or noise of the device [28]. For each new event $\mathbf{e}_i$, the timestamp t_i is stored in all 8 neighboring pixels' timestamp memory $\mathbf{T}_i(\mathbf{u}_k)$, $\forall \mathbf{u}_k \in \mathbf{u}_i \pm 1$. If the present timestamp is within t_{TH} of the previous timestamp stored at this pixel location $\mathbf{u}_i$, then this event is accepted; otherwise the event is filtered out. The removal of uncorrelated events not only minimizes the effect of noisy events on the tracking algorithm but also reduces the computation overload. The event stream after applying a background filter is shown in Fig. 3(a). Compared to the event stream shown in Fig. 2(a), there are much less noisy events and Q_P and Q_N have become to 0.42 and -0.26 . The false events are distributed around the whole sensor and removing them reduces the tracker deviation errors. On the other hand, the true events are generated on the trace of the moving object; removing some of them only reduces the number of valid samplings. Hence the BAF is not an optimal filter for our application: while it reduces tracking errors by removing false events, it also reduces temporal resolution. Therefore even it seems there are less noisy events after applying the BAF as shown in Fig. 3(a), the quality of the event stream does not actually improve much since the filter removes both noisy and valid events.

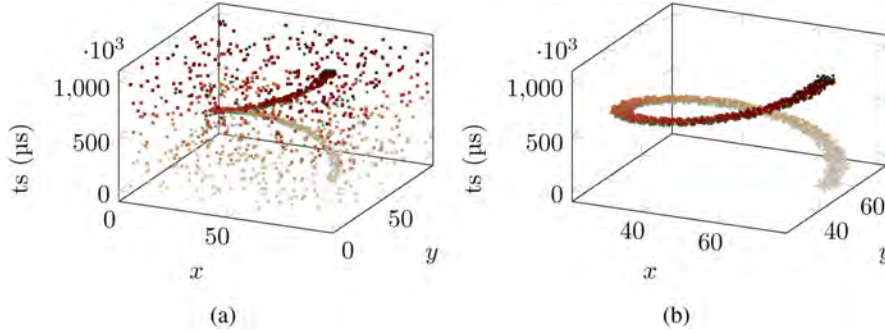


Fig. 3. Scatter plots of event stream after applying (a) background activity filter and (b) time surface activation test.

A time surface activation test was used by Afshar *et al.* as an alternative pre-processing step to the commonly used background activity filter for event-based space imaging applications [9]. A region of interest (ROI) of size $R \times R$ around the event $\mathbf{e}_i$ is selected upon receiving each new event. Therefore the ROI_i generated by event $\mathbf{e}_i = [x_i, y_i, t_i, p_i]^T$ is defined as a disc containing the time surface values $\mathbf{S}_i$ at time t_i for all pixels within a distance R to the location of the event $\mathbf{e}_i$. If the number of recently activated pixels (defined by a time interval Φ) on the time surface within the ROI_i is larger than a threshold L , then this ROI_i and together with the event $\mathbf{e}_i$ are accepted for the next stage of processing. The correlation of the generated events from focal spots moving is well-defined by both the spatial structure of the local point spread function (PSF) and the rate of motion. Therefore, the size of the ROI R is chosen to match the FWHM of the local PSF. The event stream of the rotating Gaussian spot after applying the time surface activation test is shown in Fig. 3(b). There are no visible noisy events outside the spot track and Q_P and Q_N have been both increased to 0.97.

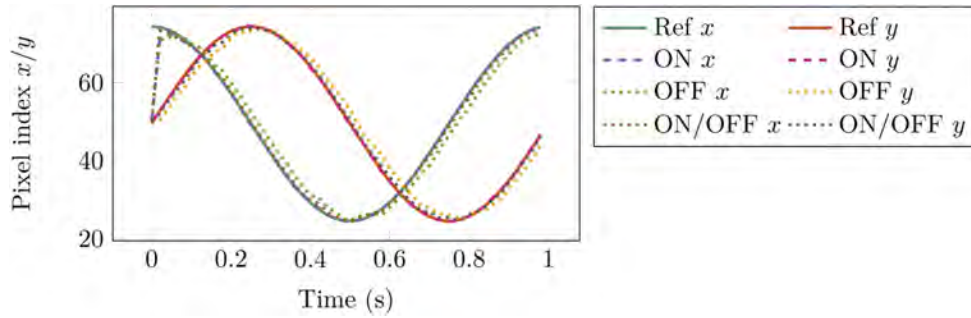


Fig. 4. Average position estimation using ON, OFF and both ON/OFF events.

3.2. Spot tracking algorithms

Several event-based object tracking algorithms have been reported in literature to track simple objects like circles, lines or blob-like sources to complex user-defined shapes [3]. Because the objects to be tracked here, local PSF in sub-apertures, have well-defined spatial and temporal profiles which do not change dramatically, two simple event-based trackers are selected to examine their effectiveness of tracking the spots displacement. The first tracker named average position tracker uses the histogram information of the events and the second tracker called Gaussian blob tracker uses a Gaussian blob to estimate the position and size of the spot.

3.2.1. Spot tracking I: using average position

To track the spot movement using the generated events, we start with a simple algorithm by calculating the average position of the given events. Every time an increase event $\mathbf{e}_i = [x_i, y_i, t_i, p_i]^T$ is generated, the average position $(\hat{x}_{t_i}, \hat{y}_{t_i})$ at the event timestamp is updated:

$$\hat{x}_{t_i} = m\hat{x}_{t_{i-1}} + (1 - m)x_i \quad (5)$$

$$\hat{y}_{t_i} = m\hat{y}_{t_{i-1}} + (1 - m)y_i \quad (6)$$

where m is an inertia ranging from 0 to 1 and its value varies depending on the event rate from the specific application. It is worth noting that only the ON events are used for this estimation otherwise the trailing OFF events are likely to pull the average position back towards the location from which the spot came. The result for tracking the spot using an average position tracker is shown in Fig. 4.

3.2.2. Spot tracking II: using Gaussian blob tracking

The second tracking method used a Gaussian blob presented in [29]. The idea behind this method is that the cluster of generated events from a moving object represent the spatial distribution of the observed shape, which can be roughly approximated by a Gaussian distribution.

The parameters of the Gaussian blob $\mathcal{N}(\mu, \Sigma)$ consist of the object's position, size and orientation. The Gaussian mean μ and the covariance matrix Σ which denotes the size and orientation are updated when new events fall within a certain scope. The assignment of each new event to a potential tracker depends on a probability value $p_i(\mathbf{u})$ which can be calculated for each tracker $\mathcal{A}_i$:

$$p_i(\mathbf{u}) = \frac{1}{2\pi} |\Sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{u} - \mu_i)^T \Sigma_i^{-1} (\mathbf{u} - \mu_i)} \quad (7)$$

where μ_i and Σ_i represent the i th tracker's location and covariance matrix:

$$\mu_i = [\mu_{ix}, \mu_{iy}]^T \quad (8)$$

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (9)$$

The probability for all the trackers will be ranked from high to low with each incoming event and if the probability of tracker with the highest rank is larger than a predefined threshold δp , the position and size of this Gaussian tracker will be updated as follows:

$$\mu_t = a_1 \mu_{t-1} + (1 - a_1) \mathbf{u} \quad (10)$$

$$\Sigma_t = a_2 \Sigma_{t-1} + (1 - a_2) \Delta \Sigma \quad (11)$$

where a_1 and a_2 are update factors.

The covariance difference $\Delta \Sigma$ is calculated by:

$$\Delta \Sigma = \begin{bmatrix} (x - \mu_{tx})^2 & (x - \mu_{tx})(y - \mu_{ty}) \\ (x - \mu_{tx})(y - \mu_{ty}) & (y - \mu_{ty})^2 \end{bmatrix} \quad (12)$$

The activity of each tracker can be defined by the following:

$$\mathcal{A}_i(t) = \begin{cases} \mathcal{A}_i(t - \Delta t) \exp(-\frac{\Delta t}{\tau}) + p_i \mathbf{u} & \text{if } e(\mathbf{u}, t) \text{ belongs to tracker } i \\ \mathcal{A}_i(t - \Delta t) \exp(-\frac{\Delta t}{\tau}) & \text{otherwise} \end{cases} \quad (13)$$

The original application of the blob tracking method used different kernels depending on the scene and object of interest but in our particular application here a Gaussian blob is sufficient as a good approximation to the local PSF. The results for tracking the spot using a Gaussian blob tracker is shown in Fig. 5.

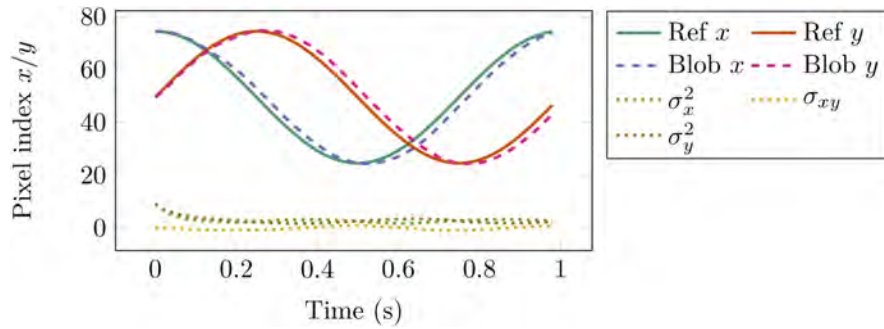


Fig. 5. Estimation of spot motion using blob tracker.

For example, the moving objects are the local PSF behind each lenslet which is approximated by a Gaussian. The size and orientation of the Gaussian blob do not need to be updated since the tracked objects are diffraction limited PSFs, though the update would be useful if elongated spots need to be tracked when using a laser guide star for example. In addition, all the trackers are expected to remain in their sub-apertures and any probability calculation is not necessary for associating each new event with a tracker since the spot array is sparse.

3.2.3. Comparison of the average position tracker and Gaussian blob tracker

At a given time t , the Gaussian spot's position is $P_t = (x_t, y_t)$. The spot estimation using either average position or blob tracker at a timestamp closest to t , can be described by $\hat{Q}_t = (\hat{x}_t, \hat{y}_t)$.

The distance between the estimated spot position $\hat{Q}_t$ and the ground truth position P_t gives the estimation error:

$$\text{distance}(P, Q) = \sqrt{(\Delta x)^2 + (\Delta y)^2} = \sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2} \quad (14)$$

The estimation error for the average position tracker using only ON events, OFF events or both events and the blob tracker are shown in Fig. 6. The average position tracker using just the ON events gives the best accuracy of tracking.

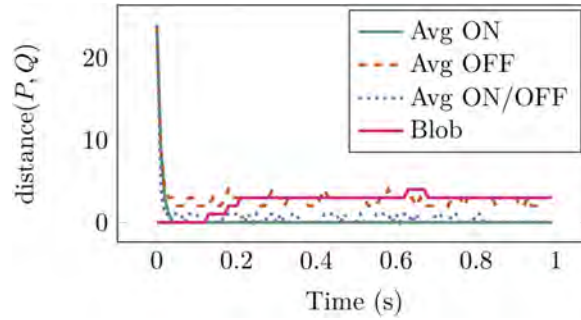


Fig. 6. Blob tracking and average position errors by using ON, OFF and both ON/OFF events.

The response time of a particular tracking algorithm is defined as how long it takes the tracker to follow up the object to within one pixel as the object passes by the initial position of the tracker. From Fig. 6, the blob tracker appears to have a faster response time than the average position trackers only because it is placed at the initial position of the spot while the average trackers are placed at the center of the object plane. There does not appear any important differentiation between techniques due to speed of event production.

3.3. Wavefront reconstruction

In Fig. 7, histograms of events generated from individual varying Zernike modes from Z_2 Tilt X to Z_{10} Trefoil X are shown. The local event histogram in each sub-aperture indicate the changing direction and strength of local wavefront slope. Less events are generated if the local wavefront stays similar with varying Zernike strength, as evidenced by the central sub-apertures in Defocus and the Astigmatisms.

Due to the asynchronous nature of the event stream, the spot positions in individual sub-apertures will be updated asynchronously as new events arrive. Assume the spot position estimated from the events using either the average position tracker or the Gaussian blob tracker is denoted as $\hat{\mathbf{u}}_i(r, c)$ for the sub-aperture $\mathcal{A}(r, c)$ at the time of the event $\mathbf{e}_i$. The local slope over the sub-aperture $\mathcal{A}_i(r, c)$ can be calculated by:

$$\mathbf{g}_i(r, c) = \frac{\Delta_i(r, c)}{f_{ML}} = \frac{\hat{\mathbf{u}}_i(r, c) - \mathbf{u}_{ref}(r, c)}{f_{ML}} \quad (15)$$

where $\Delta_i(r, c)$ is the displacement of the spot from its reference position $\mathbf{u}_{ref}(r, c)$ and f_{ML} is the focal length of the MLA. $\mathbf{G}_i$ defines the slope map of all the valid sub-apertures which will be updated per new event. If the event-by-event based processing paradigm is strictly followed, a continuously updated wavefront map $\mathbf{W}_i$ can be reconstructed from $\mathbf{G}_i$. The computation for the wavefront reconstruction is expensive because of the matrix multiplications involved, and while the per-event reconstruction is achievable using dedicated hardware like FPGA devices, the wavefront corrector like deformable mirrors will not be able to operate at the event speed.

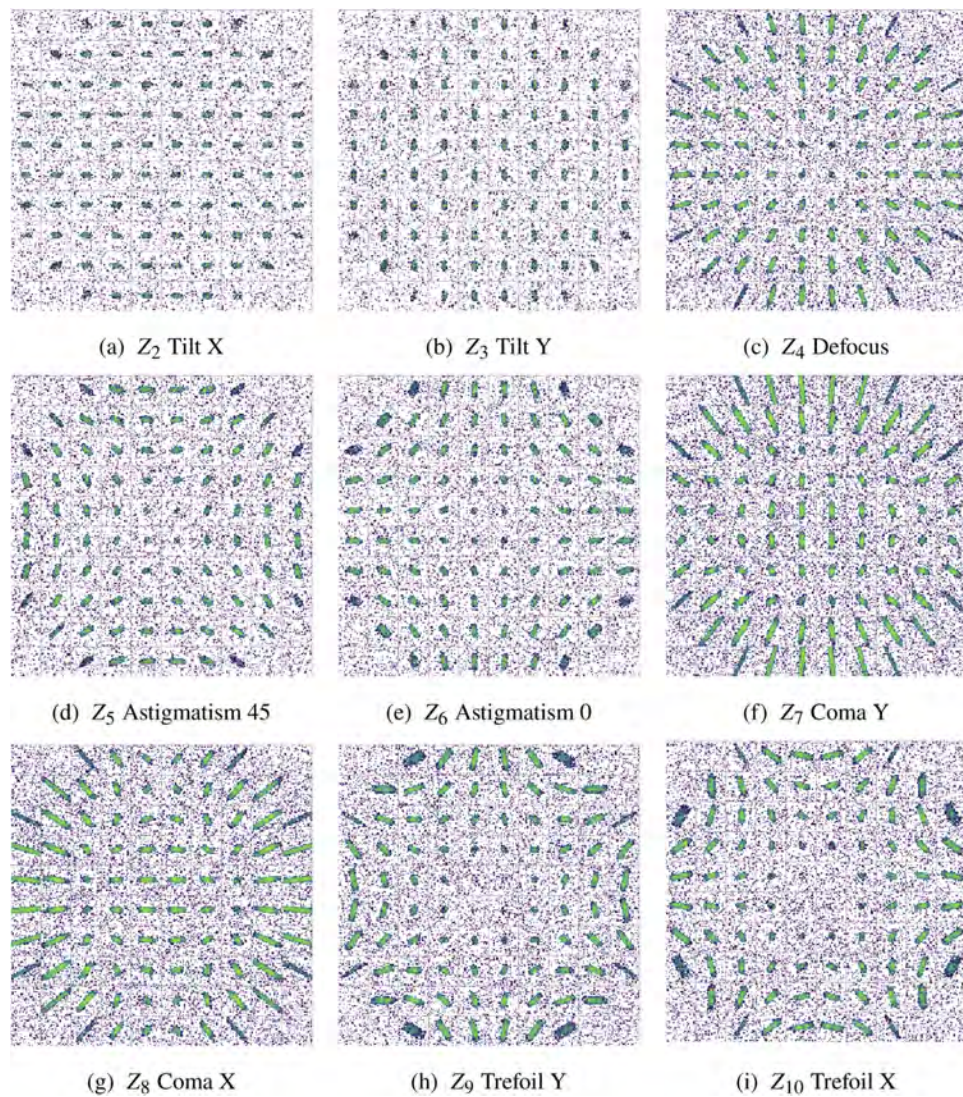


Fig. 7. Histograms of events simulated from varying Zernike modes.

Therefore, the slope map at a user defined time interval τ_R is used for the wavefront reconstruction. τ_R can be selected based on the temporal speed of the aberrated wavefront or the maximum operation speed of the deformable mirror, and is most likely to be significantly less and definitely no longer than the traditional exposure time for a SHWFS.

In Fig. 8, we provide a wavefront reconstruction of 8 Zernike coefficients from the event stream when presenting a cycle of a specific aberration (Z_4 , Z_5 , Z_6). Tilt X and Tilt Y are not displayed here from the reconstructed Zernike modes. There is an erroneous coupling between Z_4 and Z_6 in the reconstruction, which requires further investigation. We have checked for bias in the tracker estimation, so postulate instead that this may be related to row ordering when attending to events in much the same way as an electronic rolling shutter. The other feature around the discontinuous movement are effects of the tracker parameters.

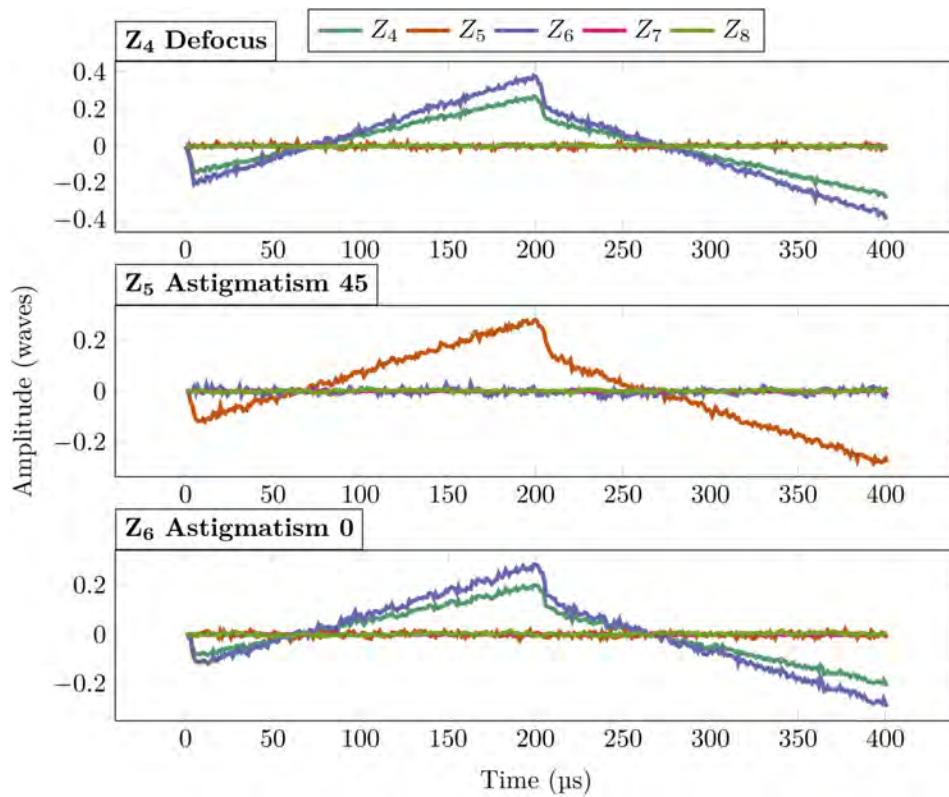


Fig. 8. Wavefront reconstructions of the first 8 Zernike modes when cycling individual Zernike Z_4 Defocus, Z_5 Astigmatism 45 and Z_6 Astigmatism 0.

4. Experimental results

4.1. Experiment setup

To verify the proposed event-based Shack-Hartmann wavefront sensor in Section 3, an iniVation DAVIS 240C event camera is inserted into the optical setup shown in Fig. 9. The Thorlabs DMP40 deformable mirror is used to introduce lower order Zernike aberrations such as Tip/Tilt and Defocus resulting in simple spot movement patterns like global shift, and shrinking or expanding as pin cushion. The regular and structural movement helps the development and verification of event-based algorithm to measure the spots location in SHWFS. The Thorlabs

WFS20 Shack-Hartmann WFS located at the other arm is used to measure the static system aberration in the common optics path, which will be compensated by the DM.

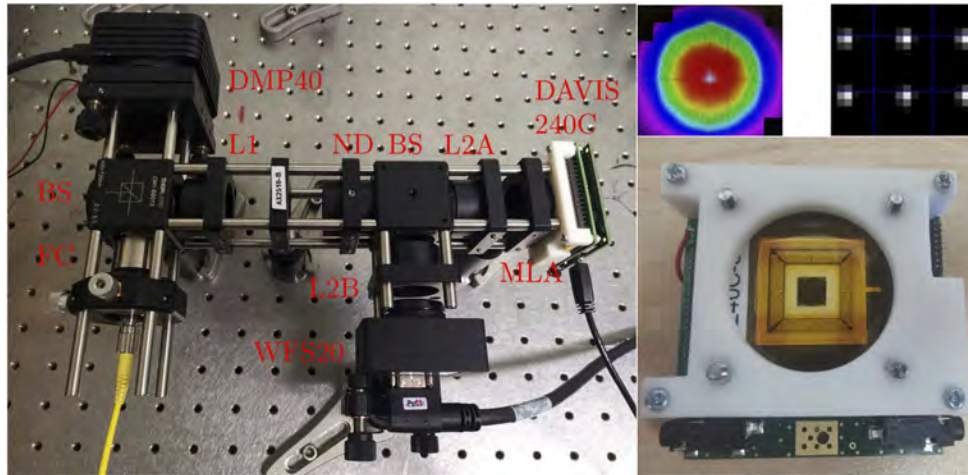


Fig. 9. (a) Optical setup for the event-based Shark-Hartmann wavefront sensing experiments (BS: beamsplitter, L: achromatic lens, ND: neutral density filter, MLA: micro lenslet array). (b) Davis 240C event camera with customized 3D printed mount compatible with Thorlab's 30mm cage systems.

The Pupil diameter on the DM is 10mm and the focal lengths of convex lens L1 and L2A/L2B are 100mm and 30mm respectively. The reduced Pupil size on the DAVIS 240C and WFS20 after this imaging relay system is therefore 3.3mm. The micro lenslet array (MLA, Thorlabs MLA150-5C) located in front of the event camera DAVIS 240C has a 150μm pitch size and the pixel size of the sensor is 18.5μm, which results in 22×22 sub-apertures each of which consists of 8×8 pixels. In comparison, the pitch size of the MLA used in the WFS20 (Thorlabs MLA300-14AR) is 300μm and the pixel size of CMOS sensor is 5μm, which leads to 11×11 sub-apertures and 60×60 pixels per sub-aperture. A custom 3D printed mount is used to mount the DAVIS 240C camera into the Thorlabs cage system. The laser beam profile measured by the WFS20 and the spots pattern from the APS in the DAVIS 240C are also shown in Fig. 9.

4.2. Global tilt

In the first experiment, the DMP40 is driven to oscillate horizontally 20 cycles in a 4 s recording, resulting all the spots to move in x direction globally. A sample of an APS image frame from the DAVIS 240C with an exposure time of 0.35 ms is shown in Fig. 10(a). The red circle indicates the Pupil on the DAVIS 240C event camera where the 22×22 valid sub-apertures can be seen. After reducing the light level by 1000 times with a ND3 filter, an exponentially-decayed time surface from the DVS event stream is shown in Fig. 10(b). The length of the time window for the time surface is 50 ms starting at about 1 s and the decay constant used here is 10 ms. The histogram of events within the same 50 ms time period is shown in Fig. 10(c). Compared to the time surface image, the horizontal trace within each sub-aperture in the histogram reveals the global motion introduced by the DM.

In Fig. 11, the averaged displacements of spots in all valid sub-apertures in x direction within a 1 s recording time period from events data using an average position tracker (ON events only) and APS frames using Center-of-Gravity are compared with each other. The estimations from the frame and event data indicate the same number of cycles. The lag between the estimations are a result of mismatch between timestamps used for frames and events, which can be manually

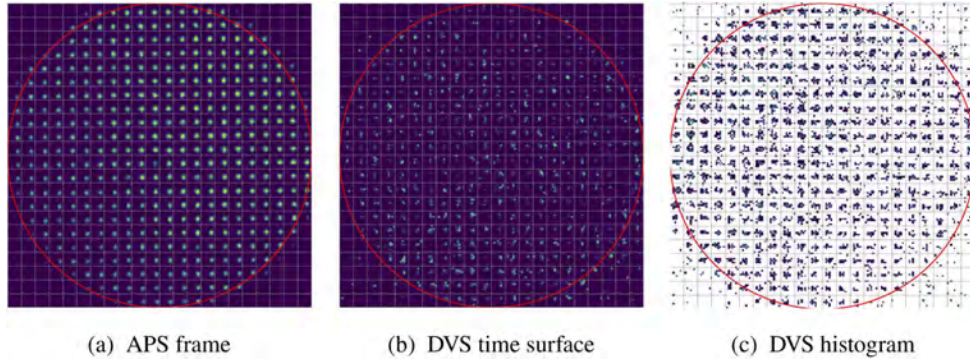


Fig. 10. (a) An APS frame from DAVIS 240C; (b) An exponentially-decayed time surface within a time period of 50 ms with a decay constant of 10 ms; (c) Histogram of events within the same 50 ms time period.

aligned upon calibration. The important thing to note here is the event data provides more than 1000 estimations in one second while the conventional APS sensor data only yields about 12 frames of estimation in the corresponding time. It also clearly exhibits the hysteresis in the DMP40 operation evidenced by the different rising and falling time in each cycle in Fig. 11.

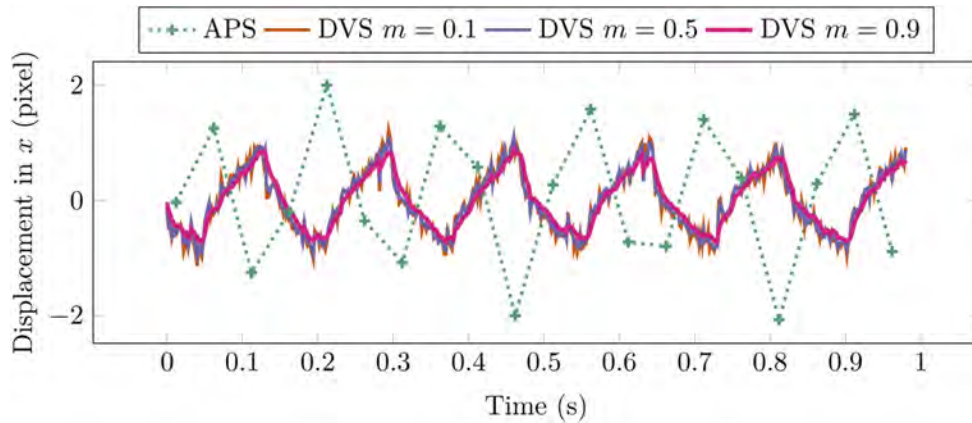


Fig. 11. Comparison of the averaged spot displacements in all valid sub-apertures from APS frames and DVS events by cycling the DMP horizontally.

4.3. Performance under low light conditions

Event cameras normally have much higher dynamic range than frame-based CCD or CMOS cameras. For example, the DVS pixels in the DAVIS 240C camera have a dynamic range of 120 dB compared to the 55 dB from the APS pixels [14]. The DVS pixel detects the change of intensity on a logarithmic scale and generates an event as soon as the change exceeds the user-defined threshold regardless of the absolute intensity level. The DVS pixel can reliably and reproducibly generate events with scene illumination down to less than 0.1 lux [1]. This makes an event camera a better candidate to work under low light condition since it is not necessary to increase the “exposure time” or gain and therefore slow down the speed or reduce SNR in order to an useful signal.

Starting from the same light intensity level as in Fig. 10, a number of different neutral density (ND) filters are inserted into the optic system to reduce the light level and observe the influence on the APS frames and DVS events. Shown in Fig. 12 are the central 4×4 sub-apertures from the APS frames and histograms of DVS events. The DAVIS 240C uses the same exposure time of 0.35 ms and starts to lose spots from some sub-apertures after reducing the light level by 10 times using the ND1 filter. The DVS pixels on the contrary still output a large amount of events as shown in the histogram of DVS events within a time slice of 100 ms which is half of the oscillation cycle. In fact, the events become much cleaner without even applying the background activity filter or time surface activation test after further reducing the light level by 100 and 1000 times using the ND2 and ND3 filters. However, reducing the light level further to 10000 times using both the ND1 and ND3 filters led to inconsistent event output, possibly because some sub-apertures' light level fall below the minimum lux requirement of the DVS pixels. The power incident per lenslet when using the ND3 filter is about 7.44 pW and we measured about 200 events generated per sub-aperture per second for the Tilt X oscillation.

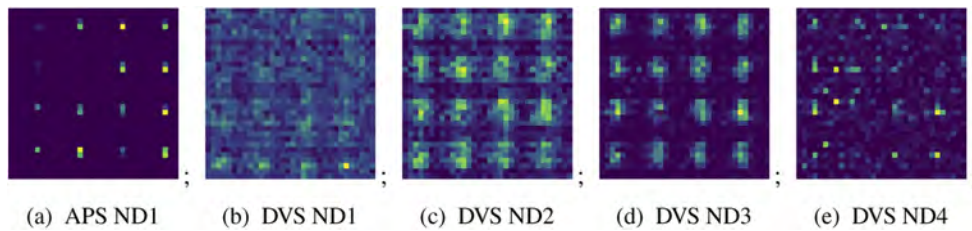


Fig. 12. From left to right: central 4×4 sub-apertures from the APS frame after inserting ND1 filter; Histograms of DVS events after inserting ND1, ND2, ND3 and ND4 filters.

In Fig. 13, the averaged displacement in x direction under Tilt X cycling after inserting three different ND filters (ND1, ND2 and ND3) is shown. In all three cases, where the light level has been decreased by 10, 100 and 1000 times, the 20 oscillation cycles from the DMP40 can be observed. These are three sequential experiments with mirror relaxed after each drive and therefore the exact oscillation speed might differ slightly. The estimation has more jitters and undervalues the motion in the case of higher light level using the ND1 filter so our recommendation

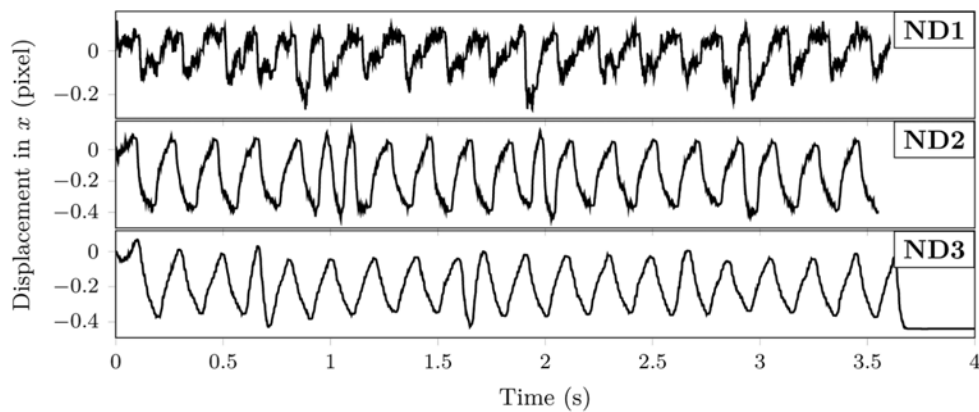


Fig. 13. Averaged displacement in x direction estimated from events after reducing light level by 10, 100 and 1000 times using ND1, ND2 and ND3 filters. The excitation is 20 cycles of Tilt X aberration.

is to operate the event-based SHWFS at low light levels. Therefore, before trying different bias settings of the event camera to optimise the event generation or applying more complex filters to improve the quality of the event stream, one can actually try to reduce the illumination to an appropriate level first. This is in contrast to the frame-based image sensors which normally struggle with low light conditions in Shack-Hartmann wavefront sensors on small telescopes or with larger number of sub-apertures.

4.4. Wavefront reconstruction

We next evaluated the response of the event camera to cycling the strength of a specific Zernike aberration and the histogram of events from varying Zernikes shown in Fig. 14 exhibit the expected spatial structure as shown in Fig. 7 from simulation. Note the extra cluster of events on the

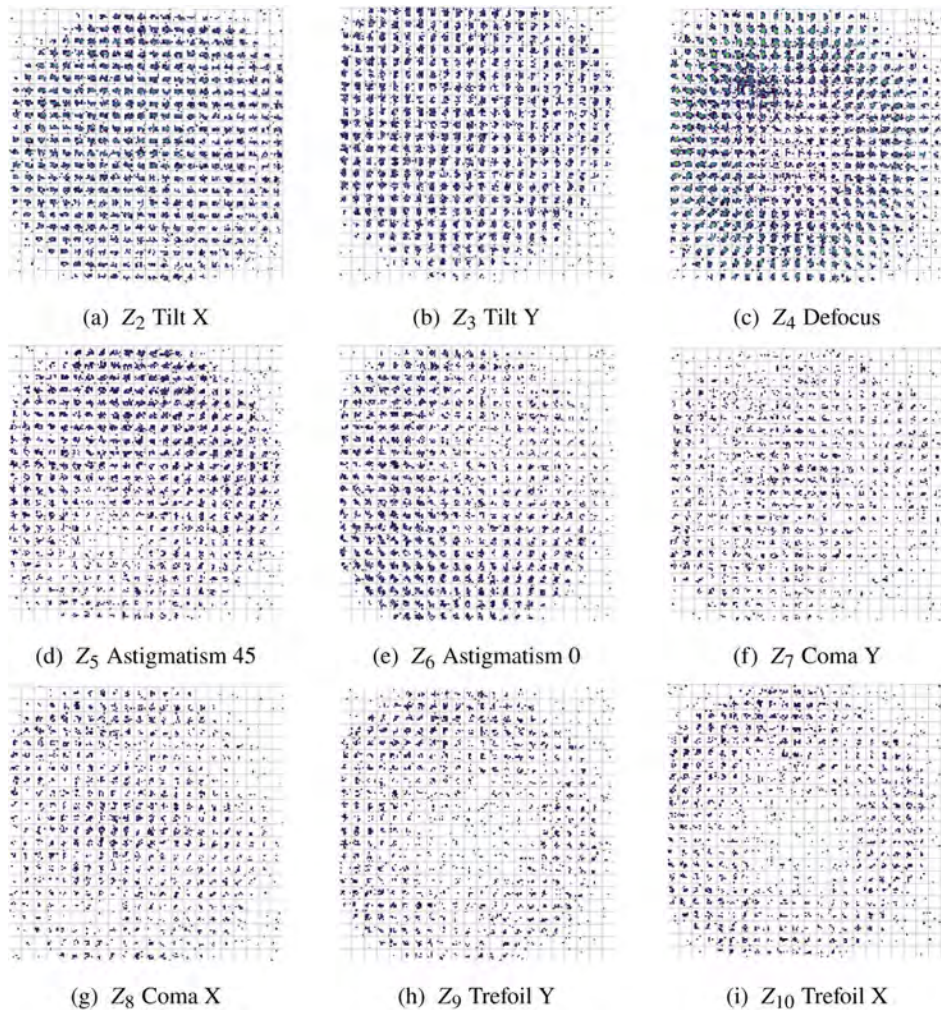


Fig. 14. Histograms of events from varying Zernike modes recorded on DAVIS 240C event camera. See [Visualization 1](#), [Visualization 2](#), [Visualization 3](#), [Visualization 4](#), [Visualization 5](#), [Visualization 6](#), [Visualization 7](#), [Visualization 8](#), [Visualization 9](#), and [Visualization 10](#), in the supplementary material for videos generated from events rendered frames with varying individual Zernike modes.

upper left histogram shown in Fig. 14(c) from the cycling Defocus which is caused by the second reflection between the MLA and previous optical element. This dim reflection is not detected by the APS frames but is observed by the DVS with its superior low light performance and high dynamic range. To view the events rendered frames in action with varying individual Zernike modes, see [Visualization 1](#), [Visualization 2](#), [Visualization 3](#), [Visualization 4](#), [Visualization 5](#), [Visualization 6](#), [Visualization 7](#), [Visualization 8](#), [Visualization 9](#), and [Visualization 10](#) in the supplementary material.

From the events data and APS frames with varying strength of a single Zernike aberration (Z_4, Z_5), modal reconstructions of the wavefront are performed using the first 11 Zernike modes. In Fig. 15, it shows that using spot estimations from the event data, there is an unfortunate coupling between Z_4 and Z_6 . On the other hand, the wavefront reconstruction using the centroid estimations from APS frames for varying Defocus does not show any crosstalk with Z_6 Astigmatism 0. This coupling is consistent with the simulation results shown in Fig. 8. We have exhaustively checked for any bias in the tracker, and postulate this is an effect of readout ordering.

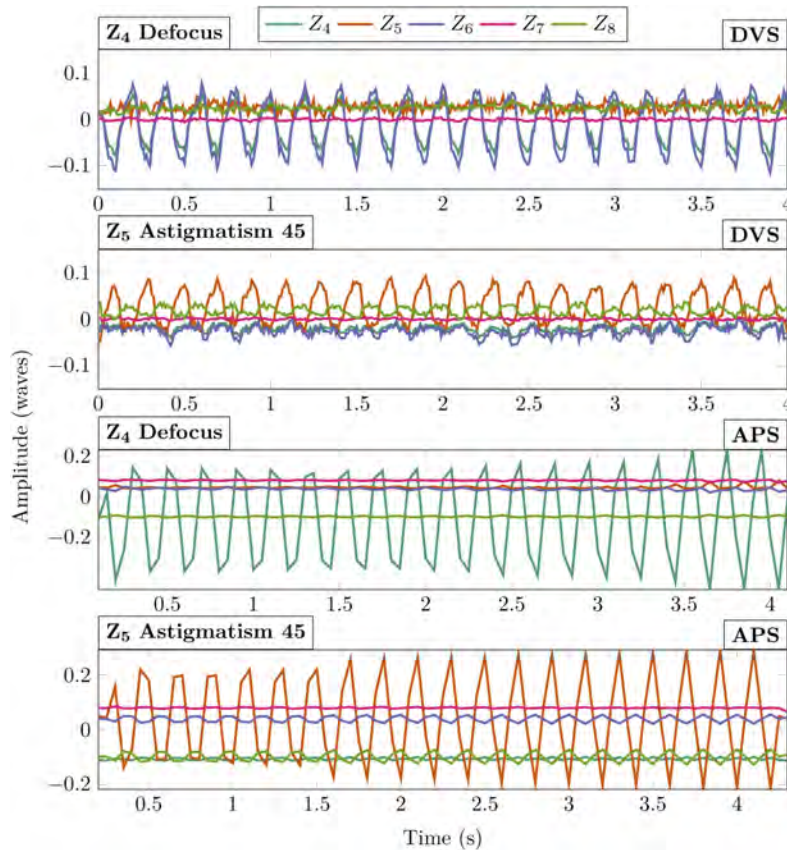


Fig. 15. Wavefront reconstructions coefficients of the first 8 Zernike modes in the presence of cycling individual Zernike Z_4 Defocus, Z_5 Astigmatism 45 and Z_6 Astigmatism 0 using the DVS event data.

5. Conclusion

Event-based image sensors are novel bio-inspired vision sensors which have drawn increased interest from many research fields. The sparse output from these image sensors led to very high

temporal resolution, low latency and power consumption which are appealing for wavefront sensors and adaptive optics system too. In this paper, we examined using two different event-based trackers to estimate the spot movements within a Shack-Hartmann wavefront sensor. The simulation results indicate that the event-based average position tracker using just ON events is capable of determining the spot's position accurately and consistently, after filtering noisy events using a time surface activation test. These noise events would bias the traditional wavefront estimation as they could not be identified as sparse in the image accumulated during the longer exposure time. Current commercial event cameras available on the market have several limitations such as small fill factor and limited resolution which hold off their performance in the Shack-Hartmann wavefront sensors. However, these limitations are likely to be reduced as the manufacturing technology of the event-based image sensor matures and tech giant company like Samsung starts to invest in the field [30]. Nevertheless, the experimental results demonstrated that an event-based Shack-Hartmann wavefront sensor has the potential to operate at very high bandwidth, extremely low light condition with a very low power consumption. The methods of correlation-based WFS when dealing with extended objects [31] are equally extendable here if the event-based tracking algorithms, for example the multikernel algorithm in [29], are adapted to track extended objects. Event-based cameras are applicable to many other types of wavefront sensors than SHWFS and we are currently evaluating their integration with Curvature wavefront sensing.

Funding

Air Force Office of Scientific Research (FA9550-18-1-0471).

Disclosures

The authors declare no conflicts of interest.

References

1. P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits* **43**(2), 566–576 (2008).
2. C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-State Circuits* **46**(1), 259–275 (2011).
3. G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-Based Vision: a Survey," arXiv e-prints arXiv:1904.08405 (2019).
4. T. Delbruck, "Neuromorphic vision sensing and processing," in *2016 46th European Solid-State Device Research Conference (ESSDERC)*, (2016), pp. 7–14.
5. R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," *Neural Netw.* **27**, 32–37 (2012).
6. T. Brosch, S. Tschechne, and H. Neumann, "On event-based optical flow detection," *Front. Neurosci.* **9**, 137 (2015).
7. G. Haessig, A. Cassidy, R. Alvarez, R. Benosman, and G. Orchard, "Spiking Optical Flow for Event-Based Sensors Using IBM's Truenorth Neurosynaptic System," arXiv e-prints arXiv:1710.09820 (2017).
8. G. Cohen, S. Afshar, B. Morreale, T. Bessell, A. Wabnitz, M. Rutten, and A. van Schaik, "Event-based sensing for space situational awareness," *J. Astronaut. Sci.* **66**(2), 125–141 (2019).
9. S. Afshar, A. P. Nicholson, A. van Schaik, and G. Cohen, "Event-based object detection and tracking for space situational awareness," arXiv e-prints arXiv:1911.08730 (2019).
10. G. Cohen, S. Afshar, and A. van Schaik, "Approaches for astrometry using event-based sensors," in *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, (2018), p. 25.
11. A. Lambert and G. Cohen, "Study of a spatio-temporal sensor for turbulence characterisation and wavefront sensing (conference presentation)," in *Environmental Effects on Light Propagation and Adaptive Systems*, (2018).
12. P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 30mW asynchronous vision sensor that responds to relative intensity change," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, (2006), pp. 2060–2069.
13. R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 120 dB 10mW 12 μs-latency sparse output vision sensor for mobile applications," in *2013 Symposium on VLSI Circuits Digest of Technical Papers*, (2013), pp. c186–c187.
14. C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240×180 130 dB 3μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits* **49**(10), 2333–2341 (2014).

15. J. Hartmann, "Bermerkungen ueber den bau und die justierung von spektrographen," *Zeitschrift fuer Instrumentenkunde* **20**, 47 (1900).
16. R. V. Shack and B. C. Platt, "Production and use of a lenticular hartmann screen," *J Opt Soc Am* **61**, 656–660 (1971).
17. S. Thomas, T. Fusco, A. Tokovinin, M. Nicolle, V. Michau, and G. Rousset, "Comparison of centroid computation algorithms in a Shack–Hartmann sensor," *Mon. Not. R. Astron. Soc.* **371**(1), 323–336 (2006).
18. F. Kong, M. C. Polo, and A. Lambert, "Centroid estimation for a Shack-Hartmann wavefront sensor based on stream processing," *Appl. Opt.* **56**(23), 6466–6475 (2017).
19. J. C. Shelton, "Denominator-free control algorithms for adaptive optics," in *Adaptive Optics and Applications*, (1997), pp. 455–459.
20. J. Hardy, *Adaptive Optics for Astronomical Telescopes*, Oxford series in optical and imaging sciences (Oxford University, 1998).
21. F. Barranco, C. Fermuller, and E. Ros, "Real-time clustering and multi-target tracking using event-based sensors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2018), pp. 5764–5769.
22. S. Afshar, T. J. Hamilton, J. Tapson, A. van Schaik, and G. Cohen, "Investigation of event-based surfaces for high-speed detection, unsupervised feature extraction, and object recognition," *Front. Neurosci.* **12**, 1047 (2019).
23. H. Rebecq, D. Gehrig, and D. Scaramuzza, "ESIM: an open event camera simulator," in *Proceedings of The 2nd Conference on Robot Learning*, vol. 87 of Proceedings of Machine Learning Research A. Billard, A. Dragan, J. Peters, and J. Morimoto, eds. (PMLR, 2018), pp. 969–982.
24. D. Joubert, "Design of a simulated event-based sensor to evaluate its potential for space applications," (2020). V1.02.
25. V. Padala, A. Basu, and G. Orchard, "A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth," *Front. Neurosci.* **12**, 118 (2018).
26. H. Liu, C. Brandli, C. Li, S.-C. Liu, and T. Delbruck, "Design of a spatiotemporal correlation filter for event-based sensors," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, (2015), pp. 722–725.
27. R. Ghosh, A. Gupta, A. Nakagawa, A. Soares, and N. Thakor, "Spatiotemporal filtering for event-based action recognition," arXiv e-prints arXiv:1903.07067 (2019).
28. T. Delbruck, "Frame-free dynamic digital vision," in *International Symposium on Secure-Life Electronics*, vol. 1 (University of Tokyo, 2008), pp. 21–26. In: Proceedings of International Symposium on Secure-Life Electronics, Advanced Electronics for Quality Life and Society, Univ. of Tokyo, Mar. 6-7, 2008.
29. X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Trans. Neural Netw. Learning Syst* **26**(8), 1710–1720 (2015).
30. B. Son, Y. Suh, S. Kim, H. Jung, J.-S. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo, Y. Roh, H. Lee, Y. Wang, I. Ovsianikov, and H. Ryu, "4.1 a 640×480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, (2017), pp. 66–67.
31. L. A. Poyneer, "Scene-based Shack-Hartmann wave-front sensing: Analysis and simulation," *Appl. Opt.* **42**(29), 5807–5815 (2003).

FPGA implementation of Shack-Hartmann wavefront sensing using Stream-based Center of Gravity method for centroid estimation

Fanpeng Kong, *Member, IEEE*, Manuel Cegarra Polo, and Andrew Lambert, *Senior Member, IEEE*

Abstract—We present a fast and reconfigurable architecture for Shack-Hartmann wavefront sensing implemented on FPGA devices using stream-based center of gravity to measure the spot displacements. Calculation of 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.

Index Terms—Adaptive optics(AO), Shack-Hartmann wavefront sensor(SHWFS), wavefront sensing, field-programmable gate array(FPGA).

I. INTRODUCTION

The Shack-Hartmann wavefront sensor(SHWFS) is one of the most widely used wavefront sensors (WFSs) in adaptive optics (AO) systems. 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 [1], minimum mean-square-error estimator [2] and maximum-likelihood methods also 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 [3], [4], [5]. A comparison of some commonly used centroiding algorithms including thresholding (TCoG), weighted centroid (WCoG), correlation and quad cell are given by Thomas *et al.* [6]. 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 as well as 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 [7] 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, 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 [8], [9] or adding weight to emphasize the signal [10], [11]. The error sources for centroid computation of a point source on a CCD-based sensor are analyzed by Ma *et al.* [12] and the best threshold level is given. Other methods using iterative detection of spot location and centroid estimation have been reported [13], [14]. 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.

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 [15] 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 [16], [17], [18], or used to implement centroid estimation and reconstruction [19], [20], [21], or even to develop full AO application including driving the wavefront corrector [22], [23]. In comparison with conventional CPU or GPU based solutions, FPGA devices provide a cost-effective

F. Kong, M. Cegarra Polo and A. Lambert are with the School of Engineering and Information Technology, The University of New South Wales Canberra, 2600 Canberra ACT, Australia (email: f.kong@adfa.edu.au; m.cegarrapolo@adfa.edu.au; a.lambert@adfa.edu.au)

Manuscript received April 19, 2019; revised August 26, 2019.

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 [24], 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.

In this paper, we present a complete Shack-Hartmann wavefront sensing system implemented on FPGA hardware using the 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 II, the theory of stream-based center of gravity deriving from conventional CoG method, 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, least-square modal wavefront reconstruction module are described in detail in Section III. In Section IV, 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 summarised in section V.

II. THEORETICAL BACKGROUND

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

TABLE I: 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)

A. 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.

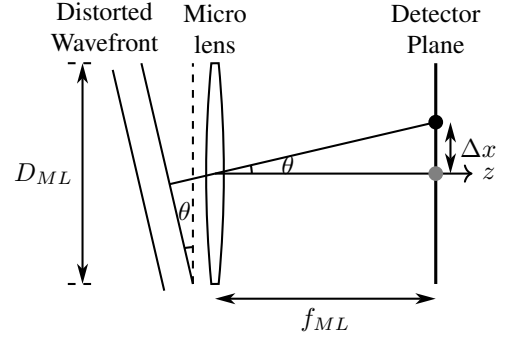


Fig. 1: Schematic of a single micro lens in the Shack-Hartmann wavefront sensor

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

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

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

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

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

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

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

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.

B. Stream-based Center of Gravity

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

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

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

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 $2M + 1$.

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

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

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 $(i_p, j_q - 1)$:

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

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

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:

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

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

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. And if only whole pixel resolution is concerned, calculations of the numerators are sufficient to locate the pixel indices.

Several immediate advantages of SCoG over conventional CoG based centroid estimation methods can be noticed [24]. 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 “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. But 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.

C. 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 higher row number, lower column number to 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:

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

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 by using the centroid with the highest energy (sum of intensity as the denominator in Eq. (3)) or taking the average of all centroids.

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

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

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 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 are not yet arrived. Therefore these further processing of the streamed centroids follows a *sorting – processing – reordering* procedure.

D. 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 [25], [26]. 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:

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

where a_k represents the weight of the k th 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:

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

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:

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

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

$$\mathbf{s}_{2M \times 1} = \mathbf{W}_{2M \times N} \mathbf{a}_{N \times 1} \quad (12)$$

where M is the total number of valid sub-apertures and N is the number of Zernike modes used for wavefront reconstruction. $\mathbf{s}$ is the slope vector of dimensions $2M \times 1$ and $\mathbf{a}$ is the Zernike mode coefficients vector of dimension $N \times 1$. $\mathbf{W}$ 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 $\mathbf{a}$ from measured slope vector $\mathbf{s}$, the pseudo-inverse of matrix $\mathbf{W}$ is used:

$$\mathbf{a}_{N \times 1} = \mathbf{E}_{N \times 2M} \mathbf{s}_{2M \times 1}; \quad (13)$$

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

$$\mathbf{E} = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T \quad (14)$$

In Section III-D, the detailed implementation of the modal wavefront reconstruction is explained.

III. IMPLEMENTATION

In this section, a parallel implementation of the stream-based center of gravity algorithm and the modal wavefront reconstruction suitable for FPGA devices is described. By taking the 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 Section II-B to II-D: *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 provides similar functions as 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 following discussion.

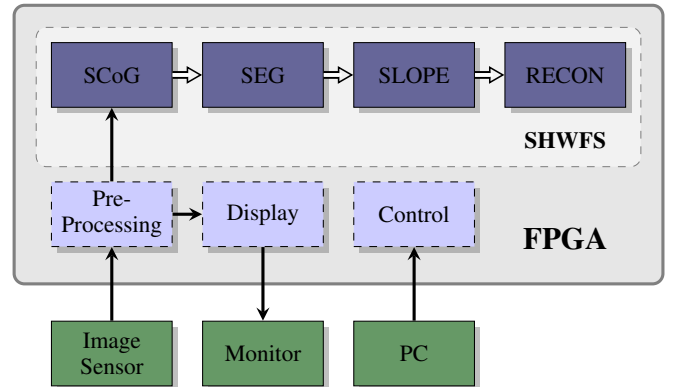


Fig. 2: Top level block diagram of the SHWFS implementation where the implementation of the SHWFS is described here.

A. 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 sets (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.

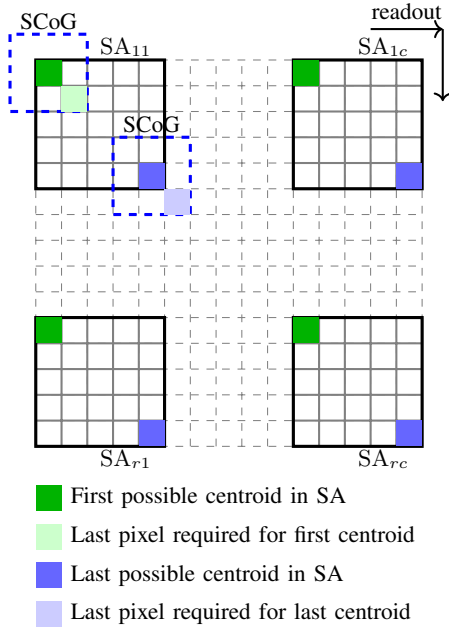


Fig. 3: Illustration of the sub-aperture and SCoG window on the image sensor pixel array.

A high level logic diagram for calculating the stream centroid according to Section II-B 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 3 buffers are used to store 3 rows of pixel values and 3 Flip-Flop (FF) registers are used to buffer 3 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 9 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 constraint 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.

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 the same as the column number.

B. 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 but 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:

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, Eq. 7 is used to calculate the row and column index of the sub-aperture. When the number of pixels per lenslet window is an integer of power two, the division can be simplified to bit shifts.

2) *Treatment of multiple centroids and missing centroid*: As discussed in Section II-C, 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

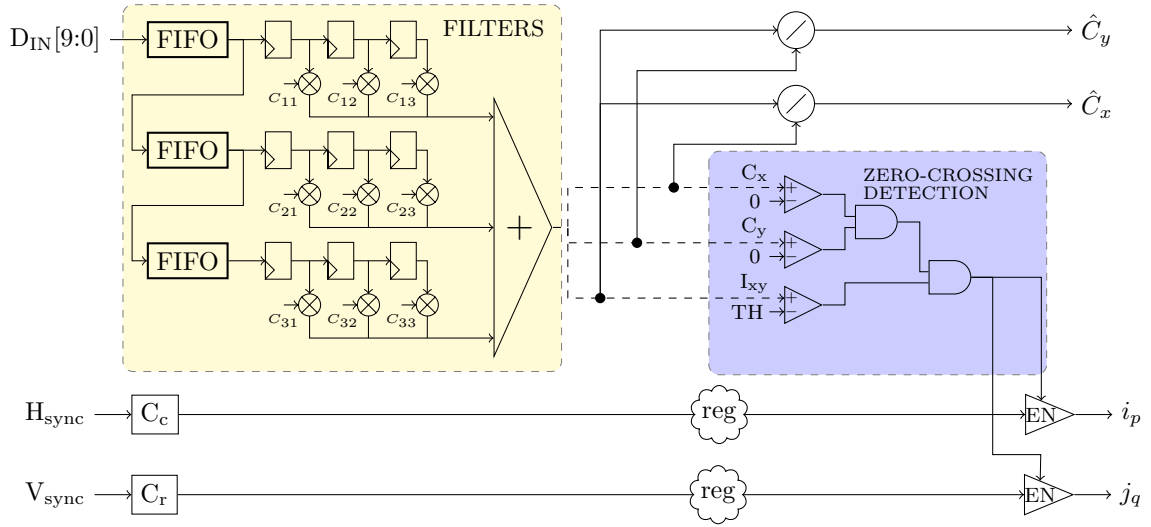


Fig. 4: Diagram for the implementation of SCoG module used for calculate integer pixel centroid values. Depending on the coefficient values C , the numerators and denominators in Equation (3) are calculated from the FILTERS circuit.

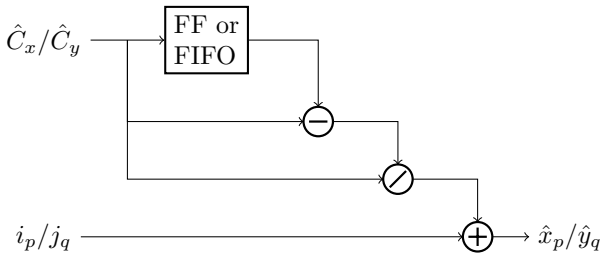


Fig. 5: Diagram for the implementation sub-pixel interpolation in x and y direction.

trace back to an initial default reference centroid positions for each sub-aperture.

Once the centroids for the last sub-aperture in a row has 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:

C. 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 particular 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.

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

Data: stream centroid C
Result: ordered centroid for each sub-aperture $\mathcal{A}$

```

1 initialization;
2 foreach stream centroid  $C$  do
3    $r := \text{floor}(C_x/w)$ ;
4    $c := \text{floor}(C_y/w)$ ;
5   append  $C$  to centroid list of  $\mathcal{A}(r, c)$ ;
6   if time of the last possible  $C$  in  $\mathcal{A}$  then
7     if no centroid in  $\mathcal{A}(r, c)$  then
8       Assign the centroid from previous frame;
9     else if multiple centroids in  $\mathcal{A}(r, c)$  then
10      Average all the centroids;
11    if last  $\mathcal{A}$  in a MLA row then
12      reorder all the centroids in this row;
13    end
14  end
15 end

```

D. 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

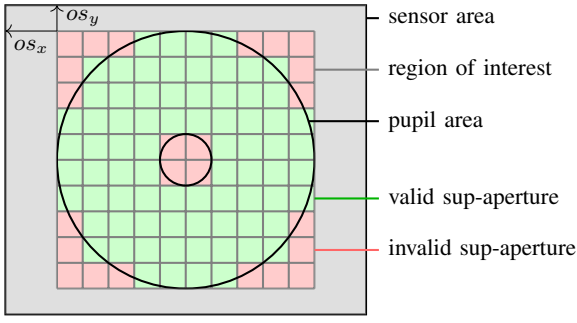


Fig. 6: Illustration of various regions on the sensor area. The gray area is the sensor area on which the stream-based CoG operates. The region of interest is selected such that it covers the Pupil of the imaging system. The green segments indicate those valid MLA lenslets within the Pupil while the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

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 is 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 are obtained on the ROI coordinate system. Then 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) are used. A binary sub-aperture mask indicating the validness of an sub-aperture is used to decide whether to accumulate matrix multiplication result to the partial wavefront reconstruction in the RECON module.

IV. RESULTS

A. 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.

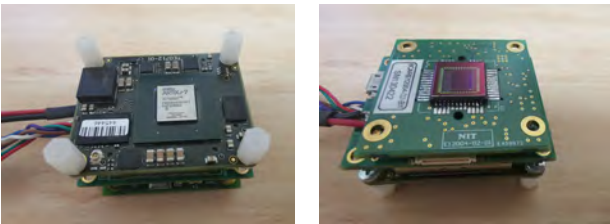


Fig. 7: Hardware platform used to implement the SCoG based SHWFS.

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 is shown in Table II. 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 II: Resource usage of different modules on Artix 7 XC7A200T.

Resources	SCoG	SEG	Slope	RECON	Total
Slice LUT	1554	7744	33	3731	13062 (10%)
Slice REG	3567	16184	19	10100	29870 (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 [20], [19] and therefore 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 increases linearly.

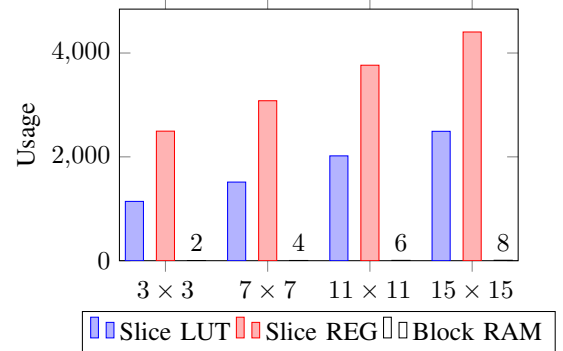


Fig. 8: Resource usage for the SCoG module with four different filter sizes.

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 in 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 introduce 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 [20]. 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} .

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.

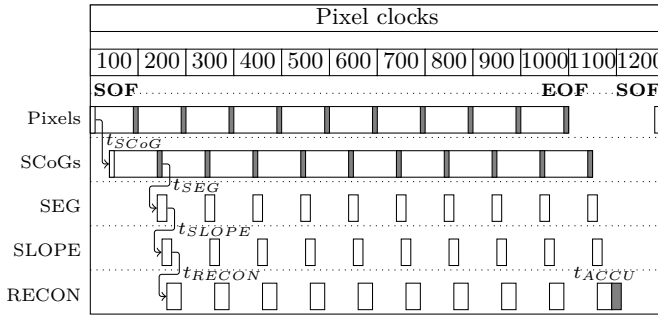


Fig. 9: Time delay between different modules in the SHWFS implementation. Note that the whole wavefront sensing operation finishes shortly after End-Of-Frame (EOF) and before the next Start-Of-Frame (SOF).

With larger filter size, closure of 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 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 most SHWFS limit the spot size to maximize the detectability but metrology look for better precision with luxury of more light.

B. Experimental Results

To evaluate the effectiveness of the hardware implementation of the proposed SHWFS using SCoG algorithm, artificial SHWFS spots images are generated from numerical simulation and stored inside the FPGA ROM. Then a camera module written in HDL is 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.

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 [24]. 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:

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

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 totally wrong estimation, the SCoG always has a close agreement with the ground truth along the path.

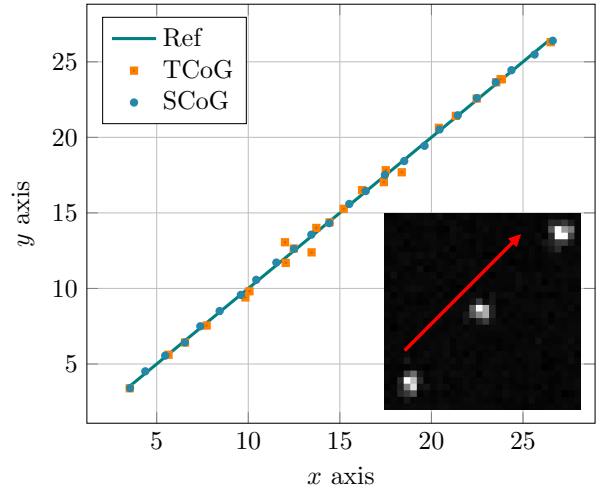


Fig. 10: Comparison of hardware implemented SCoG with a traditional thresholding CoG (TCoG) for detecting a moving Gaussian spot.

To evaluate a case where traditional CoG methods fail, we present in Figure 11, 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 is 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 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 previous section, the average of these centroids are used to represent the mean local wavefront. But they could be used in other better ways [27] to yield a more precise wavefront reconstruction with further study. The SCoG works here despite of scintillation and accompanying signal spread.

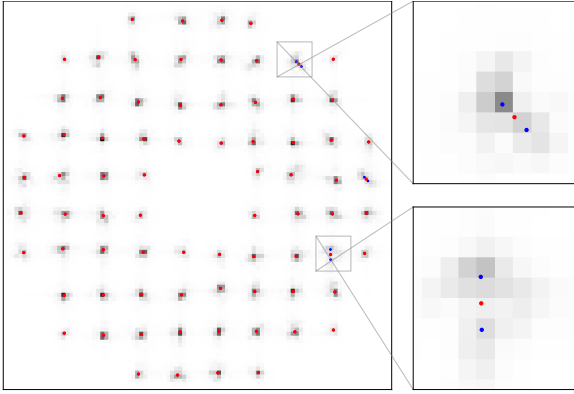


Fig. 11: All the detected centroids (blue) and average ones for each sub-apertures (red) are annotated on the SHWFS spots pattern (invalid sub-apertures are not shown here). Color for the spots are inverted for better visibility.

2) *Wavefront reconstruction results:* In a similar way, to demonstrate the performance of the whole SHWFS design including the wavefront reconstruction, SHWFS spots pattern is generated numerically from a known phase screen as shown in 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 μ m 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 μ m and therefore there are 10 $\times$ 10 sub-apertures and each sub-aperture has 30 $\times$ 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.

V. CONCLUSION

In this paper, we have presented a complete hardware implementation of Shack-Hartmann wavefront sensor in a modular design. The spots displacement is measured using the previously reported Stream-based CoG algorithm, which

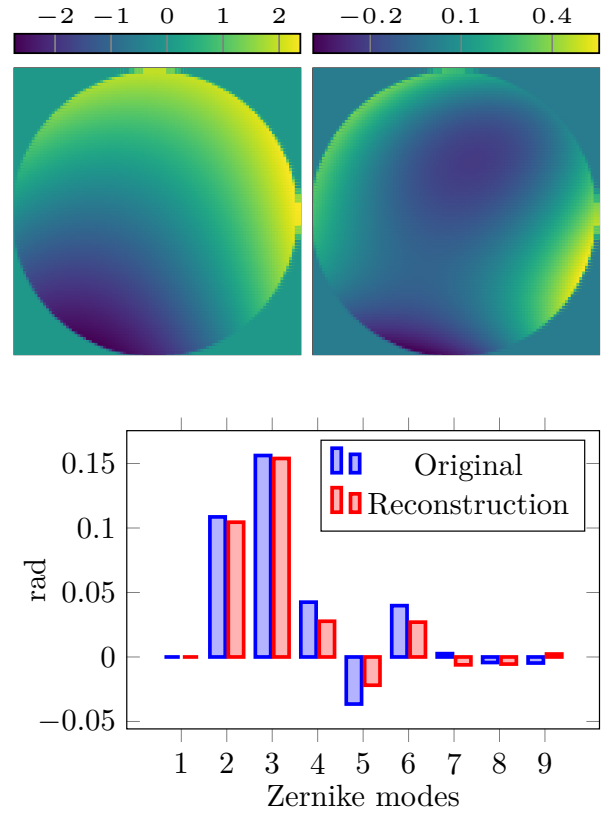


Fig. 12: Top left: original phase; Top right: phase residual; Bottom: comparison of original and reconstructed Zernike coefficients using Noll's index notation [28].

reduces the centroid estimation error when signal is cut by 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 [4]. The First Fourier coefficient (1FC) algorithm [29] 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 algorithms event to increase the dynamic range such as [27]. 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.

APPENDIX 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:

$$F_x(m, n) = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} [-M \quad -M+1 \quad \dots \quad M] \quad (17)$$

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

$$F_y(m, n) = \begin{bmatrix} -M \\ -M+1 \\ \vdots \\ M \end{bmatrix} [1 \quad 1 \quad \dots \quad 1] \quad (18)$$

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 13. Because the 1D filter $[-M \quad -M+1 \dots M]^T$ in Equation (18) 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.

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.

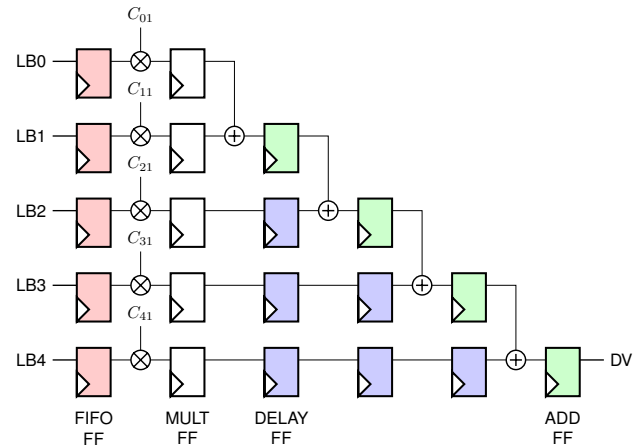


Fig. 13: A fully pipe-lined design for the vertical filter operation. LB is the output of different line buffers and DV represents the output after applying the column vector.

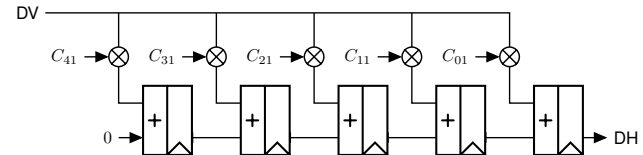
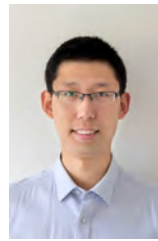


Fig. 14: The horizontal filter implemented with a Transpose structure which takes the output of the column vector operation, DV, and produces DH in its output.

REFERENCES

- [1] C. Leroux and C. Dainty, "Estimation of centroid positions with a matched-filter algorithm: Relevance for aberrometry of the eye," *Opt. Express*, vol. 18, no. 2, pp. 1197–1206, Jan 2010. [Online]. Available: <https://doi.org/10.1364/OE.18.001197>
- [2] M. A. van Dam and R. G. Lane, "Wave-front slope estimation," *J. Opt. Soc. Am. A*, vol. 17, no. 7, pp. 1319–1324, Jul 2000. [Online]. Available: <https://doi.org/10.1364/JOSAA.17.001319>
- [3] V. Michau, G. Rousset, and J. Fontanella, "Wavefront Sensing from Extended Sources," in *Real Time and Post Facto Solar Image Correction*, R. R. Radick, Ed., Dec. 1993, p. 124.
- [4] L. A. Poyneer, "Scene-based Shack-Hartmann wave-front sensing: Analysis and simulation," *Appl. Opt.*, vol. 42, no. 29, pp. 5807–5815, Oct 2003. [Online]. Available: <https://doi.org/10.1364/AO.42.005807>
- [5] P. A. Knutsson, M. Owner-Petersen, and C. Dainty, "Extended object wavefront sensing based on the correlation spectrum phase," *Opt. Express*, vol. 13, no. 23, pp. 9527–9536, Nov 2005. [Online]. Available: <https://doi.org/10.1364/OPEX.13.009527>
- [6] S. Thomas, T. Fusco, A. Tokovinin, M. Nicolle, V. Michau, and G. Rousset, "Comparison of centroid computation algorithms in a Shack-Hartmann sensor," *Monthly Notices of the Royal Astronomical Society*, vol. 371, no. 1, pp. 323–336, 2006. [Online]. Available: <https://doi.org/10.1111/j.1365-2966.2006.10661.x>
- [7] G. Rousset, "Wave-front sensors," in *Adaptive Optics in Astronomy*, F. Roddier, Ed. Cambridge University Press, Nov 2004, ch. 5, pp. 91–130.
- [8] J. Arines and J. Ares, "Minimum variance centroid thresholding," *Opt. Lett.*, vol. 27, no. 7, pp. 497–499, Apr 2002. [Online]. Available: <https://doi.org/10.1364/OL.27.000497>
- [9] X. Li, X. Li, and C. Wang, "Optimum threshold selection method of centroid computation for Gaussian spot," *Proc. SPIE*, vol. 9675, pp. 967517–967517–8, 2015. [Online]. Available: <https://doi.org/10.1117/12.2199247>
- [10] M. Nicolle, T. Fusco, G. Rousset, and V. Michau, "Improvement of Shack-Hartmann wave-front sensor measurement for extreme adaptive optics," *Opt. Lett.*, vol. 29, no. 23, pp. 2743–2745, Dec 2004. [Online]. Available: <https://doi.org/10.1364/OL.29.002743>

- [11] K. L. Baker and M. M. Moallem, "Iteratively weighted centroiding for Shack-Hartmann wave-front sensors," *Opt. Express*, vol. 15, no. 8, pp. 5147–5159, Apr 2007. [Online]. Available: <https://doi.org/10.1364/OE.15.005147>
- [12] X. Ma, C. Rao, and H. Zheng, "Error analysis of CCD-based point source centroid computation under the background light," *Opt. Express*, vol. 17, no. 10, pp. 8525–8541, May 2009. [Online]. Available: <https://doi.org/10.1364/OE.17.008525>
- [13] X. Yin, X. Li, L. Zhao, and Z. Fang, "Automatic centroid detection for Shack-Hartmann wavefront sensor," in *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2009, pp. 1986–1991. [Online]. Available: <https://doi.org/10.1109/AIM.2009.5229758>
- [14] Spiricon, *Hartmann Wavefront Analyzer Tutorial*, rev. d ed., Spiricon, Inc., 2004.
- [15] J. Mompeán, J. L. Aragón, P. M. Prieto, and P. Artal, "GPU-based processing of Hartmann-Shack images for accurate and high-speed ocular wavefront sensing," *Future Generation Computer Systems*, vol. 91, no. nil, pp. 177–190, 2019. [Online]. Available: <https://doi.org/10.1016/j.future.2018.09.010>
- [16] S. J. Goodsell, E. Fedrigo, N. A. Dipper, R. Donaldson, D. Geng, R. M. Myers, C. D. Saunter, and C. Soenke, "FPGA developments for the SPARTA project," *Proc. SPIE*, vol. 5903, pp. 5903 – 5903 – 12, 2005. [Online]. Available: <https://doi.org/10.1117/12.617837>
- [17] S. Mauch, J. Reger, C. Reinlein, M. Appelfelder, M. Goy, E. Beckert, and A. Tünnermann, "FPGA-accelerated adaptive optics wavefront control," vol. 8978, 2014, pp. 897 802–897 802–12. [Online]. Available: <https://doi.org/10.1117/12.2038910>
- [18] S. Mauch, A. Barth, J. Reger, C. Reinlein, M. Appelfelder, and E. Beckert, "FPGA-accelerated adaptive optics wavefront control part ii," *Proc. SPIE*, vol. 9343, pp. 93 430Y–93 430Y–13, 2015. [Online]. Available: <https://doi.org/10.1117/12.2079010>
- [19] S. Mauch and J. Reger, "Real-time spot detection and ordering for a Shack-Hartmann wavefront sensor with a low-cost FPGA," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 10, pp. 2379–2386, Oct 2014. [Online]. Available: <https://doi.org/10.1109/TIM.2014.2310616>
- [20] M. Thier, R. Paris, T. Thurner, and G. Schitter, "Low-latency Shack-Hartmann wavefront sensor based on an industrial smart camera," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 5, pp. 1241–1249, May 2013. [Online]. Available: <https://doi.org/10.1109/TIM.2012.2223333>
- [21] Z. Kincses, L. Orzó, Z. Nagy, G. Mező, and P. Szolgay, "High-speed, SAD based wavefront sensor architecture implementation on FPGA," *Journal of Signal Processing Systems*, vol. 64, no. 3, pp. 279–290, 2011. [Online]. Available: <https://doi.org/10.1007/s11265-010-0487-4>
- [22] C. D. Saunter, G. D. Love, M. Johns, and J. Holmes, "FPGA technology for high-speed low-cost adaptive optics," *Proc. SPIE*, vol. 6018, pp. 60 181G–60 181G–7, 2005. [Online]. Available: <https://doi.org/10.1117/12.669386>
- [23] M. Cegarra Polo, "Adaptive optics for small aperture telescopes," Ph.D. dissertation, The University of New South Wales, 2015.
- [24] F. Kong, M. C. Polo, and A. Lambert, "Centroid estimation for a Shack-Hartmann wavefront sensor based on stream processing," *Appl. Opt.*, vol. 56, no. 23, pp. 6466–6475, Aug 2017. [Online]. Available: <https://doi.org/10.1364/AO.56.006466>
- [25] J. Hardy, *Adaptive Optics for Astronomical Telescopes*, ser. Oxford series in optical and imaging sciences. Oxford University Press, 1998. [Online]. Available: http://books.google.com.au/books?id=-0aAWyckS_8C
- [26] L. A. Poyneer, D. T. Gavel, and J. M. Brase, "Fast wavefront reconstruction in large adaptive optics systems with use of the fourier transform," *Journal of the Optical Society of America A*, vol. 19, no. 10, p. 2100, 2002. [Online]. Available: <https://doi.org/10.1364/josaa.19.002100>
- [27] C. Leroux and C. Dainty, "A simple and robust method to extend the dynamic range of an aberrometer," *Opt. Express*, vol. 17, no. 21, pp. 19 055–19 061, Oct 2009. [Online]. Available: <https://doi.org/10.1364/OE.17.019055>
- [28] R. J. Noll, "Zernike polynomials and atmospheric turbulence," *J. Opt. Soc. Am.*, vol. 66, no. 3, pp. 207–211, Mar 1976. [Online]. Available: <https://doi.org/10.1364/JOSA.66.000207>
- [29] A. Lambert and M. Cegarra Polo, "Real-time algorithms implemented in hardware for centroiding in a Shack-Hartmann sensor," in *Imaging and Applied Optics 2015*. Optical Society of America, 2015, p. AOW3F.2. [Online]. Available: <https://doi.org/10.1364/AOMS.2015.AOW3F.2>



America Air Force Office of Scientific Research (AFOSR) for adaptive optics and advanced wavefront sensors using event-based cameras.

Fanpeng Kong Dr. Fanpeng Kong received his PhD in Electrical Engineering from UNSW Canberra under the supervision of Associate Professor Andrew Lambert. His research focuses on adaptive optics for Astronomy, especially wavefront sensing of atmospheric turbulence using the combination of holographic or Shack-Hartmann wavefront sensor (WFS) and field-programmable gate array (FPGA) devices. He is currently working as a Postdoctoral Fellow at UNSW Canberra and adjunct researcher at WSU on a project funded by the United States of



Manuel Cegarra Polo is a Research Associate at the Space Research group in the School of Engineering and Technology (UNSW Canberra), where he collaborates in different projects to develop, improve and maintain the existing optical and RF sensors, that will play a key role in the ground based tracking of the next UNSW Canberra CubeSats missions scheduled from 2016 to 2020. Prior to joining UNSW he was working at CSIC, the government agency for scientific Research in Spain, and in Research and Development departments in different Spanish technological companies, with the role of electronic engineer in the fields of wireless communications, embedded processing based in FPGA, ultrasound technologies, radio communications and related.

Manuel Cegarra Polo Manuel Cegarra Polo is a Research Associate at the Space Research group in the School of Engineering and Technology (UNSW Canberra), where he collaborates in different projects to develop, improve and maintain the existing optical and RF sensors, that will play a key role in the ground based tracking of the next UNSW Canberra CubeSats missions scheduled from 2016 to 2020. Prior to joining UNSW he was working at CSIC, the government agency for scientific Research in Spain, and in Research and Development departments in different Spanish technological companies, with the role of electronic engineer in the fields of wireless communications, embedded processing based in FPGA, ultrasound technologies, radio communications and related.



for imaging through turbulence applications in astronomy, surveillance, space surveillance, satellite engineering and ophthalmology.

Andrew J. Lambert Andrew J. Lambert received the B.Sc. (Hons.) degree in Physics from the University of Otago, Dunedin, New Zealand, in 1984, and the Ph.D. degree in Electrical Engineering from UNSW, Australia in 1997. He is an Associate Professor with UNSW Canberra Space in the School of Engineering and Information Technology, The University of New South Wales, Canberra, A.C.T., Australia. His research interests include optical and digital image and signal processing, high-speed processing hardware, electro-optics and adaptive optics