ARL-CR-0867 ● FEB 2022

DEVCOM ARMY RESEARCH LABORATORY

Apparatus and Techniques for the Deployment and Characterization of Event-Based Sensors

prepared by Jonah P Sengupta
The Johns Hopkins University
Baltimore, MD

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-CR-0867 ● FEB 2022

Apparatus and Techniques for the Deployment and Characterization of Event-Based Sensors

prepared by Jonah P Sengupta
The Johns Hopkins University
Baltimore, MD

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
February 2022	Contractor Report	June–September 2021

4. TITLE AND SUBTITLE

Apparatus and Techniques for the Deployment and Characterization of Event-Based Sensors

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jonah P Sengupta

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The Johns Hopkins University
Baltimore, MD

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DEVCOM Army Research Laboratory
ATTN: FCDD-RLW-TD
Aberdeen Proving Ground, MD 21005

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

ARL-CR-0867

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

ORCID ID: Jonah Sengupta, 0000-0003-3555-7207

14. ABSTRACT

Event-based (EB) cameras offer a size- and weight-efficient sensing platform that does not emit a signature but can be used in applications requiring high-speed solutions. Despite the promise of EB cameras, little progress has been made in maturing the technology in the realm of defense deployment. Current sensing modalities, such as radar and lidar, have been equipped with infrastructure to allow expedient testing in experimental range scenarios as well as vetted to understand the limitations of each technology with reference to velocity profiles. A series of equipment, test fixtures, and techniques were recently developed to characterize and deploy EB sensors in high-speed scenarios. Initial experiments have been conducted to characterize scene velocities, communication latencies, and hardware capabilities. Camera collection interactions under 50 μs have been seen, rotational velocities of 200 rpm are detected, and a complete profile of delays was acquired.

15. SUBJECT TERMS

event-based (EB) sensing, high-speed detection, embedded processing, general-purpose input–output, GPIO

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	22	Jonah P Sengupta
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER (Include area code) (410) 278-6219

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

Autonomous and semi-autonomous platforms in military applications operate under severe constraints, are required to perform with utmost precision, and face scenarios with great consequence. Much like their operator, for whom they are meant to protect, these systems gather sensory data, perceive obstacles and salient information, and perform a degree of contextual understanding. However, the timescales and magnitudes of these information processing tasks far exceed those of human capabilities. Therefore, novel sensing and computational paradigms that contribute little power and weight cost but operate in fast timescales while being robust to environmental factors are of tantamount importance.

Event-based (EB) cameras derive their functionality from the mammalian retina: a thin neural layer in the back of the eye that transduces light to electrical signals.[1] Like the retina, these optical sensors logarithmically compress incident light and subsequently only respond to changes in the resulting signal. In contrast to traditional complementary-metal-oxide-semiconductor sensors, which encode integrated amounts of light into frames of digital values, EB sensors asynchronously transmit these "spikes" based on temporally changing scene illumination and reflectances. Such operation greatly reduces the data rate from the sensor and allows for large-dynamic-range, low-latency, and energy-efficient sensing, making it an ideal solution for high-speed application.

Despite the promise of EB cameras, little progress has been made to mature the technology in the realm of high-speed detection. Current sensing modalities such as radar and lidar have been equipped with infrastructure to allow expedient testing in range scenarios as well as vetted to understand the limitations of each technology with reference to high-speed velocity profiles. To get from being "range-ready" to extracting maximum detection frequency and sensor latency, additional work is needed to progress EB sensing to the same stage as other technologies that have been exploited for situational awareness in scenarios constrained by size, weight, and power.

This report details a series of equipment, test fixtures, and techniques recently developed to characterize and deploy EB sensors in range- and autonomous-vehicle-based scenarios. The following section describes the design methodology and rationale of equipment used to characterize and operationalize EB cameras. Section 3 presents results from deployment of the test fixtures, and the report concludes with Section 4.

2. Test Fixtures

Typical metrics and specifications of candidate sensors for high-speed applications include dynamic range, latency, signal-to-noise ratio, and bandwidth. This aforementioned effort to create testing infrastructure is to allow for rapid characterization and deployment of EB sensors.

2.1 Motor Mount

The motion of objects in the visual plane can be decomposed into two components: rotational and translational movement.[2] Optical flow is the projection of such velocities into a 2D space on the image sensor. The former component of 3D velocity is not dependent on the relative depth of the object, unlike the latter. Parallax is the effect where the relative position of the object influences its perceived motion and translational velocity. Hence, objects further away from a camera move slower than closer ones.

A main thrust in perceiving threats in a high-speed application is understanding relative velocities and characterizing the motion of objects in the camera's field of vision. Careful characterization of such features is needed to understand the limitations of the sensing front-end and algorithmic back-end. As a result, a testbed and experimental setup were designed to aid this purpose. Currently, other efforts to characterize EB sensors use graphical displays to generate arbitrary visual stimuli. Despite the theoretical ability to generate any arbitrary pattern, a graphical approach also incurs a refresh rate that generates spurious events. Hence, an electro-mechanical method based on a stepper motor was pursued. Another goal of the effort is to make the designed equipment compatible within a laboratory setting, specifically a customary 1-inch optical table. In addition, the testbed was made to be programmable to allow for rotational velocities. An ArduinoMega development board equipped with an ATMega2560 microcontroller was used to control the digital signaling that configured the motor frequency.

The test fixture, seen by the camera in Fig. 1, is composed of three main components: the Arduino microcontroller and driver circuit, optical-table motor mount and stepper motor, and mounted stimulus. The ArduinoMega board, seen with the driving circuit in Fig. 2, was used to stimulate the motor. The motor mount was made from two pieces of 1/2-inch aluminum welded to form a 90° joint. The motor, seen in Fig. 3, was then affixed to the aluminum mount via a 3D printed part. High-tensile foam pieces were used to reduce motor vibration within the part cavity. An optical stimulus was mounted onto the motor by coupling a 1-inch-diameter foam disk to a 3D printed part that fit on the motor's shaft. Patterns were then printed on to paper, which was then taped to the disk to produce the optical

stimulus. The stepper motor used in this setup was a NEMA17 model. A DAVIS346 model EB sensor from iniVation[3] was used in the monocular and stereo capture of the motor-mount diagnostic rotations. A DRV8825 stepper driver was used to create the micro-stepped current waveform used to stimulate the bipolar AC motor. An on-board potentiometer was used to limit the current of the integrated circuit (IC) to 2 A to prevent damage the motor circuit. Motor stimulus requires a relatively large voltage (>9), which was provided by an external power supply. In future designs a battery pack could be used to provide a similar supply voltage and increase portability of the testbed.

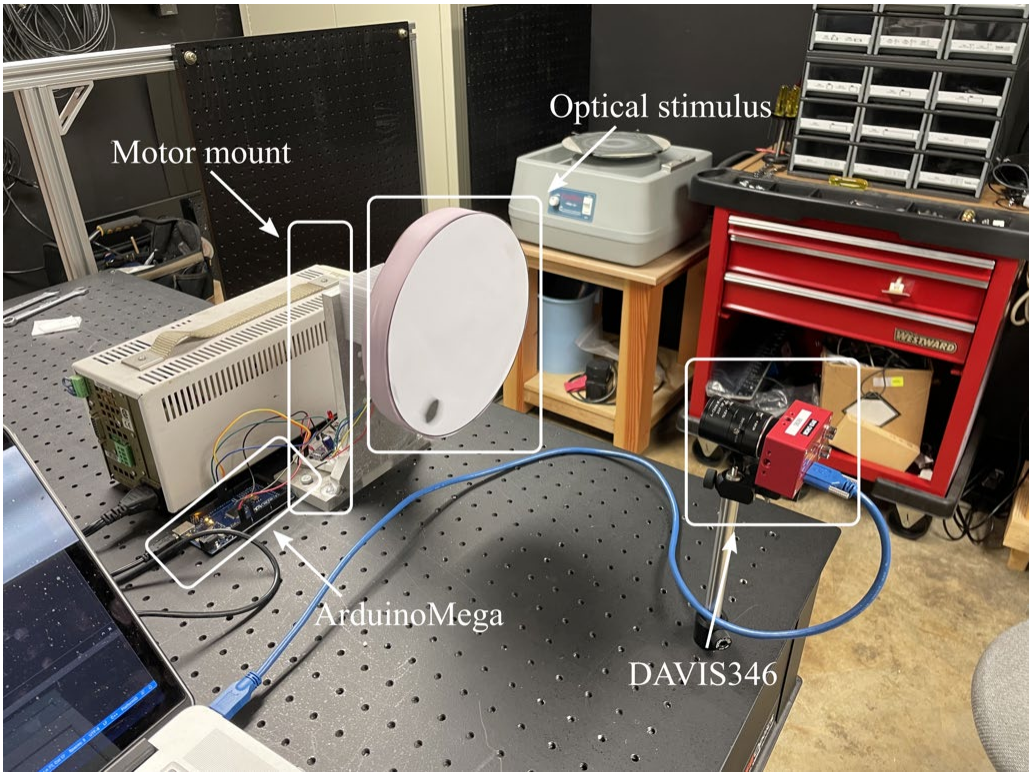

Fig. 1 Lab setup of the motor-mount assembly with parts labeled

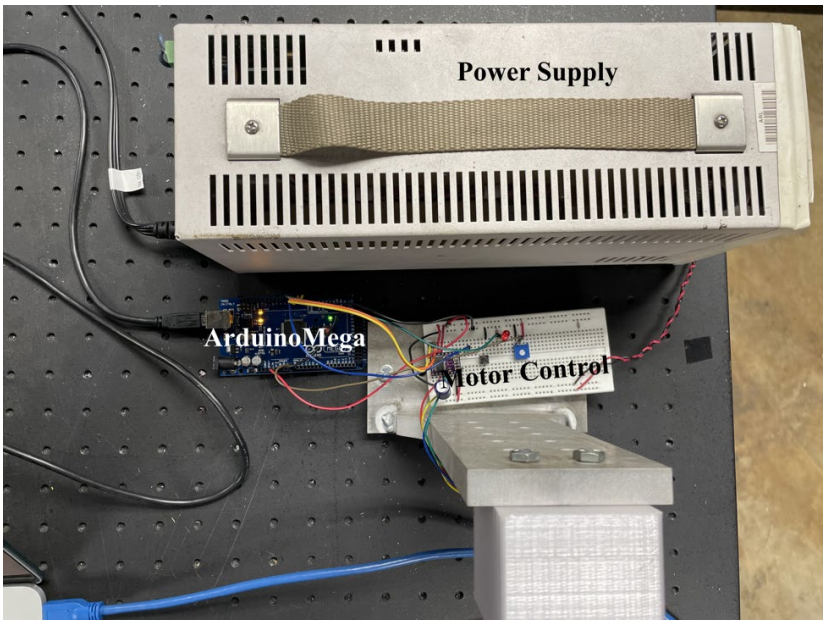

Fig. 2 Control and power elements of the motor-mount assembly

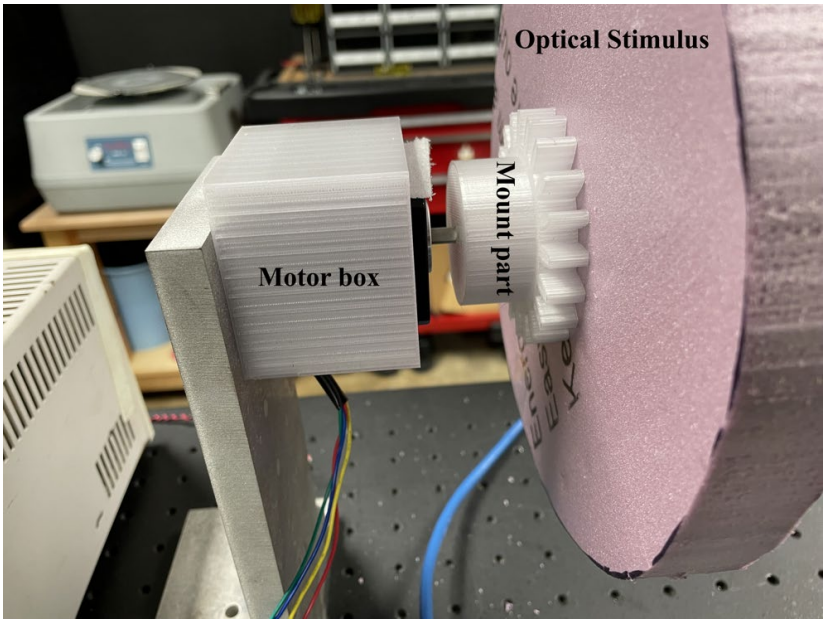

Fig. 3 Motor and mounting part for optical stimulus rotation

2.2 PiBox

Developing technology and sensors for expedient deployment in high-speed systems is highlighted by rigorous testing in a variety of combat-like scenarios. High-speed, reliable communication among sensing, control modules, and countermeasures is essential to make these aforementioned systems “range-ready”.

Typical communication practices include network-based approaches such as Wi-Fi or Ethernet-enabled Transmission Control Protocol (TCP)/User Diagram Protocol interfaces or hardware-based paradigms such as transistor–transistor logic (TTL) signaling. Sensing platforms equipped with both of these interfaces are not only able to be deployed readily in a range scenario but can be scaled to be compatible in application infrastructure. Hence, the following effort was undertaken to provide these communication capabilities for commercial off-the-shelf EB sensors.

An embedded platform to append to the EB sensor was needed to realize the range-ready specifications and provide a means to run computational tasks and salient algorithms locally. The choice of such a platform was constrained by a set of software and hardware criteria. For one, a singular USB-C cable is used to provide the DAVIS-346 a channel for communication to a host computer and power for camera electronics. Also, any platform should be relatively low-cost to mitigate loss in incidental, destructive scenarios on range. The platform should also have network connectivity capabilities and a general-purpose input–output (GPIO) interface for hardware level signaling. In addition, it should be compact enough to allow for future efforts on autonomous platforms. In terms of software requirements, the chosen platform needs to be compatible with libraries, such as libcaer, used to interface to the EB cameras.

An embedded solution that satisfies all the requirements is the Raspberry Pi Model 4. The Internet-of-Things device is equipped with a Broadcom BCM2711 capable of many different flavors of Linux. Instead of RaspianOS, Ubuntu Server 20.04 was used to allow compatibility with ROS, DV, and libcaer. Pigpio, an open-source application programming interface (API) to handle GPIO, was chosen for hardware signaling using the header pins on the board. Additionally, the Pi is able to establish network communication via Ethernet and Wi-Fi devices. According to RaspberryPi specification, the GPIO voltage regime is 3.3 V despite the presence of 5 V on the input–output header. This becomes an issue due to the 5-V/TTL standard of range devices. To solve this issue, a companion board was made with a pair of level-shifter ICs to translate signals from the 5- to 3.3-V domain (i.e., for trigger inputs) and 3.3- to 5-V domain (trigger outputs).

As seen in Fig. 4, the box and printed-circuit-board shelf were designed, printed, and assembled with the requisite hardware. The PiBox has six Bayonet Neill–Concelman (BNC) connectors to interface with three trigger inputs and three trigger outputs. When powered, it also can be armed by closing the switch on the top of the box. Two indicator LEDs designate when the box is armed and when the DAVIS capture has been triggered. There is a button affixed to the top of the box to allow for manual triggers, which helps troubleshooting during setup. This

version of the box uses female-to-female jumpers that allows for arbitrary configurations of the 10 GPIO signals.

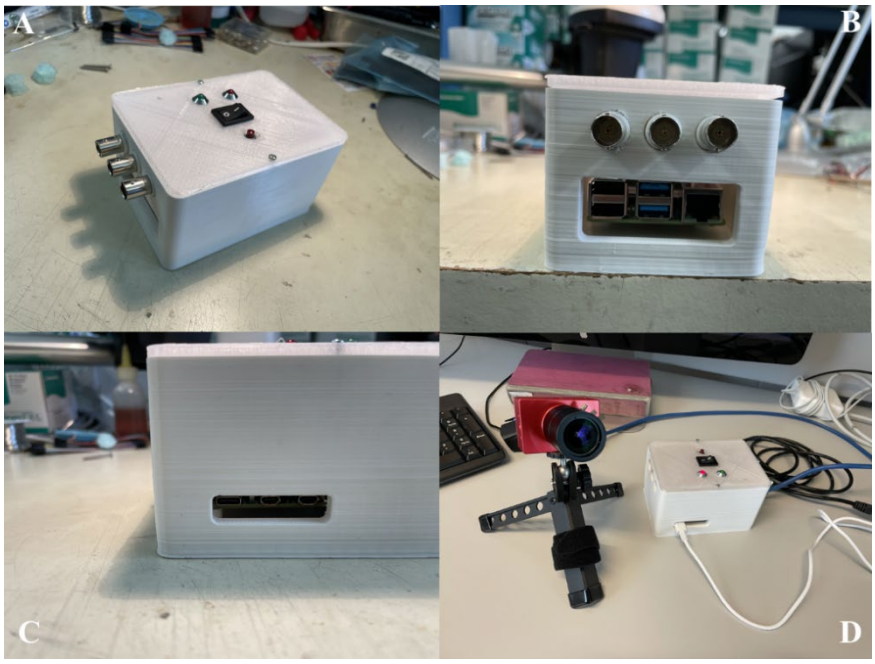

Fig. 4 A) Full view of PiBox assembly, B) box opening for USB and Ethernet connectivity along with BNC trigger interface, C) power and display connections, and D) PiBox interfaced with DAVIS346 via USB (blue cable) and host PC via Ethernet (black cable)

3. Results

3.1 Motor Mount

The motor mount assembly was demonstrated and characterized using a singular black-dot pattern and a pair of DAVIS346 sensors arranged in a stereo configuration. Sensors were calibrated by capturing a video of a moving, 15-mm, 8 × 9 checkerboard pattern. By synchronizing the streams via physical HiRose cable that resets timestamps in conjunction, calibration was achieved using the StereoCalibrationModule in MATLAB.[4] The intrinsic parameters of each sensor were resolved in addition to the extrinsic parameters of the stereo setup (Fig. 5). A routine was then written for the Arduino-based control module, which modulated the rotational velocity of the motor. The motor was programmed to spin at six distinct velocities (75, 100, 125, 150, 200, and 250 rpm) for 10 revolutions each. After capturing the raw data from the sensors, a spatial filter was implemented in MATLAB to pass only events within a specific region of interest (ROI). This ROI can be programmed to be rectangular or circular geometry. A 3D point cloud of the rotating target was then extracted for each camera, as seen in Fig. 6. The blue and

orange streams correlate to the left and right sensors, respectively. Each point cloud was then projected on to the axis pertaining to the X-addresses in Fig. 7. A sinusoid of increasing frequency can be seen that correlates to the programmed routine using the Arduino-driver circuit just described.

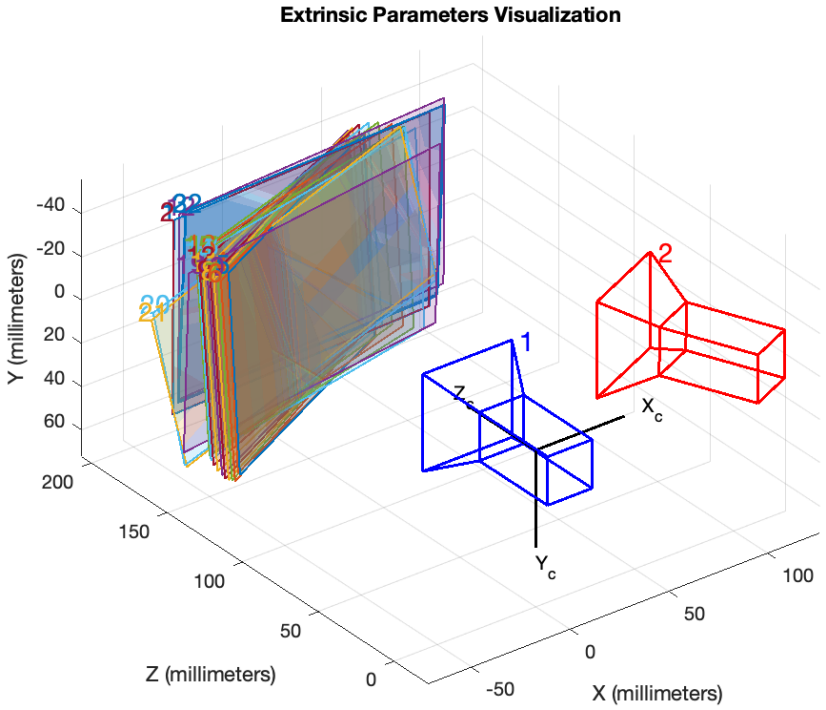

Fig. 5 Graphical depiction of DAVIS346 stereo organization. The setup baseline was approximately 10 cm with the motor mount having a 200-cm standoff from the sensors.

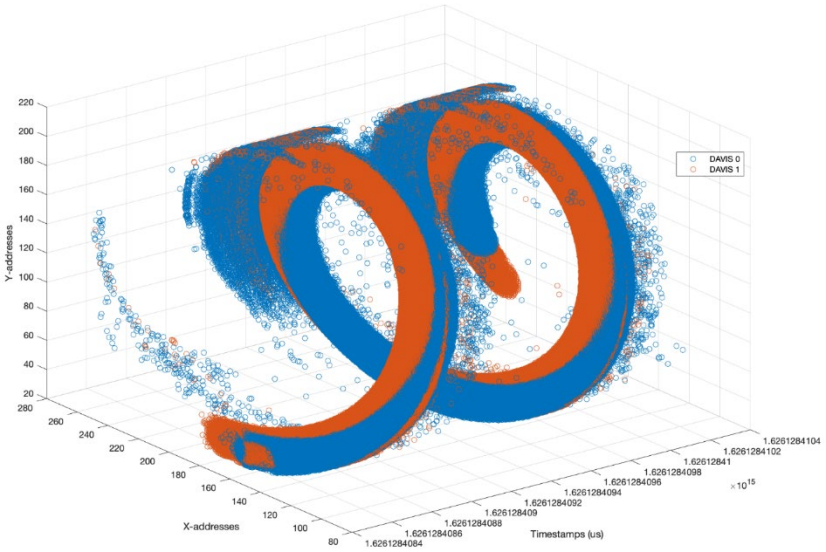

Fig. 6 Spatially filtered, 3D EB point-cloud of rotating dots. The orange stream corresponds to the left camera (DAVIS0) and the blue stream is the captured data from the

right (DAVIS1). The approximate disparity between the dots in the X and Y address dimensions can be used to approximate the distance to the motor mount.

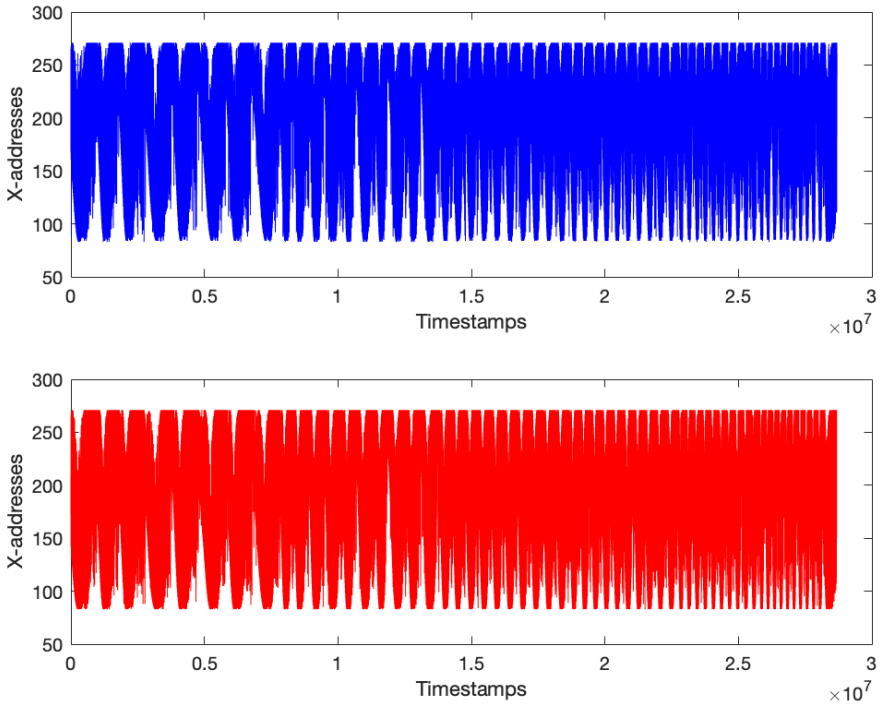

Fig. 7 EB data from each DAVIS projected onto the X-address axis. The signal is composed of the frequencies programmed using the Arduino driver circuit.

An EB algorithm was used to extract the target optical flow.[5] Similar to the frame-based Lucas–Kanade approach, the algorithm examines the spatiotemporal neighborhood of each event and fits a plane to the local field. By minimizing the L2 loss, pruning outliers, and iterating, a representative plane can be found and used to find the approximate flow-field. The mean flow magnitude for both sensors can be seen in Fig. 8, in which five distinct levels corresponding to the five rotational velocities are seen. The sixth-fastest velocity was not reliably examined using the experimental setup.

Precise angular and absolute error can be extracted using the true velocity, camera parameters, and the found optical-flow results. This in-depth algorithm inspection using the motor-mount stand is a priority for future experiments. Additionally, a larger dataset of stimuli and velocity profiles will be compiled to characterize the EB sensor limits.

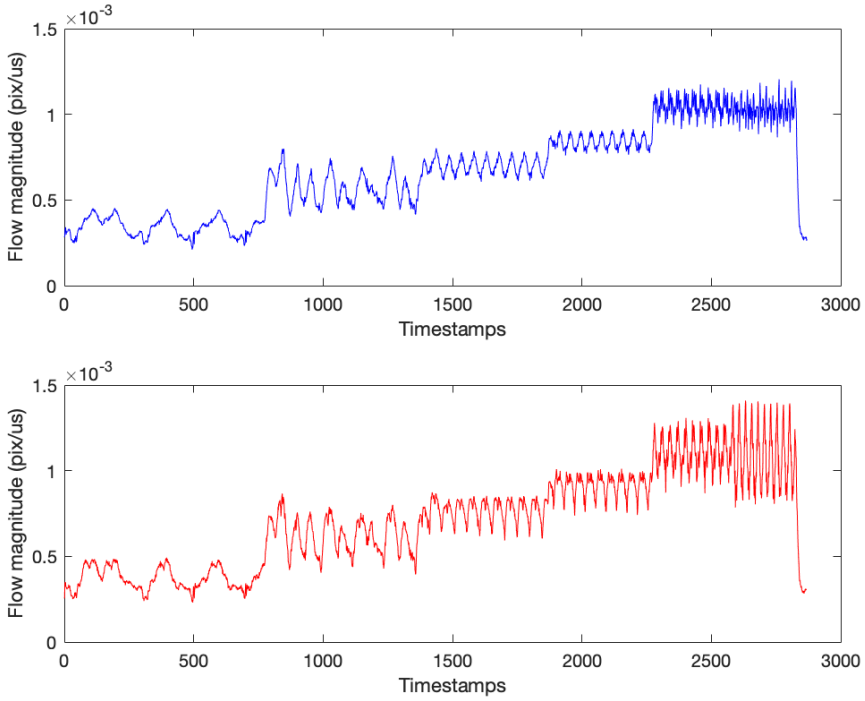

Fig. 8 Mean optical flow extracted from the recorded rotating dot using the motor-mount stand

3.2 PiBox Testing

A series of experiments set in a laboratory environment were devised to troubleshoot and characterize the PiBox. High-speed object detection requires low-latency decision-making and processing steps. Therefore, initial steps of the PiBox consisted of understanding the latencies involved with the hardware triggering used for process initialization, optical detection, and communication interfaces such as TCP via Ethernet. RaspberryPi GPIO were used to handle the hardware signaling used for trigger-in-and-out interfaces. Multiple candidate software libraries exist to interact with the Broadcom CPU GPIO pins, but the library pigpio was chosen for its interpretable API, adequate documentation, and C interface. EB-based sensors use the C-based library libcaer to configure sensor biases, readout parameters, and onboard microprocessor capabilities.

Prior to interfacing with the camera using libcaer, pigpio was used to diagnose communication and hardware delays. Two distinct approaches were used to characterize hardware latencies: flag polling and interrupt service routines (ISRs). The former approach uses the main procedure to continuously read the GPIO input to detect an edge. Pigpio's API can also be used to specify an ISR: a specific function to be evaluated upon receiving an edge on a specific GPIO input. The library accomplishes this by interfacing with sysfs, a file system used by the Linux

kernel to publish information about hardware devices, drivers, and other processes. Upon detecting the input edge, both methods then trigger an output pin to provide a concrete hardware event to characterize the latency. Example trigger-in-and-out waveforms are shown in Fig. 9. Ten samples using both methods were collected to construct the distribution shown in Fig. 10.

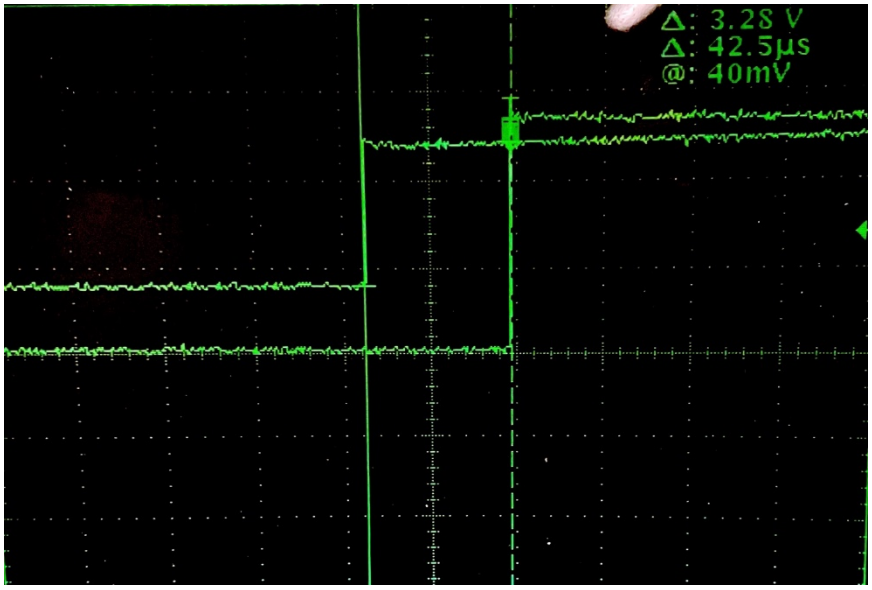

Fig. 9 Oscilloscope waveform capturing latency recording for ISR procedure

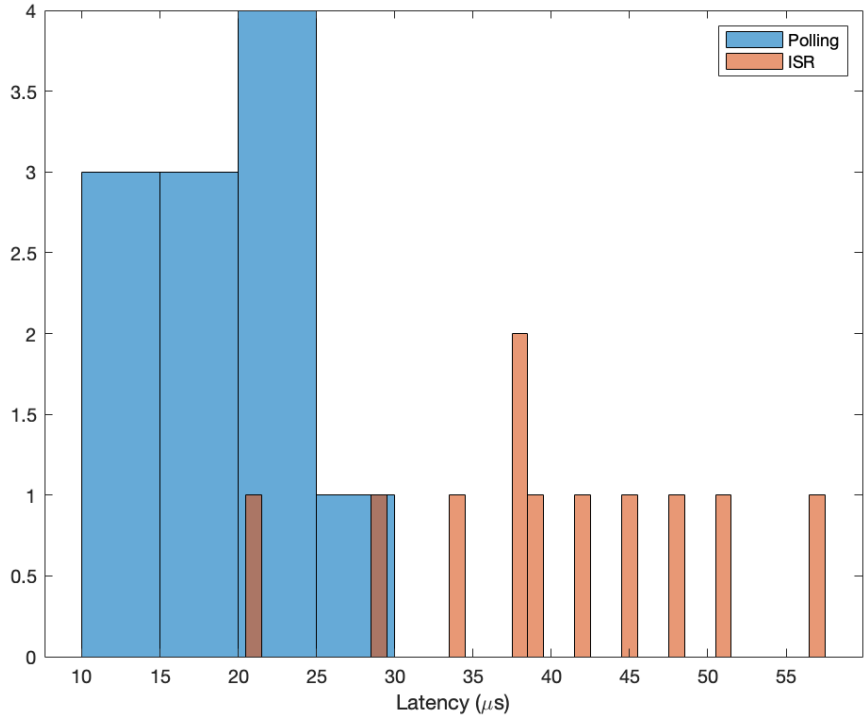

Fig. 10 Distribution of hardware signal latencies: (blue) polling method, (orange) ISR method

Next, Ethernet communication latencies were experimentally captured. A point-to-point connection via TCP was established using C-based socket interfaces between a host PC and the PiBox. The experimental procedure consisted of creating this connection, ensuring its integrity, capturing a hardware trigger in, bouncing a byte to and from the server (host PC), and sending a trigger out on the client (PiBox) once the byte is returned. In this way a lower bound on latency was modeled because this only consists of a single client with minimal data transmission. Similar to the hardware trigger experiment, a series of 10 trials was conducted to create a distribution on latencies.

Finally, basic algorithm latency was understood via detection of an LED blink. Two processes were constructed from the main thread: one set a GPIO signal that enabled an LED and the other captured data from the DAVIS346. Each event datum consists of a timestamp, hence in a sparse setting, timestamps would have large differences. On the contrary, when a large temporal contrast is present, multiple events with close timestamps will be produced. This fact was exploited to produce an algorithm that could detect an LED flash while only using 10 s of events. When an LED blink was detected, the second process read the output of a pipe that contained the trigger time and used the current time to calculate the latency. A series of trials was completed to find a mean algorithm latency.

Figure 11 shows a synopsis of the approximate contribution of each communication and computation step performed on the PiBox. Table 1 outlines the mean and standard deviation for each step further. In comparison with the algorithm and network latency, hardware signaling contributes little to the overall figure. In addition to its large mean figure, algorithm latency has a substantial amount of jitter, which can lead to varying amount of time consumed. This can be attributed to the interaction of acquiring events from a sensor that has noise mechanisms and is not fully optically isolated (i.e., AC light sources are present). In contrast, network and trigger latencies are 4–5 times more uniform, suggesting a stronger guarantee of performance.

In a culmination of the effort, the PiBox was deployed in multiple range scenarios. Independent of the captured interactions, the PiBox successfully coordinated the capture of EB data with submillisecond latency. Local capture was used to mitigate the network latencies, and raw data was used to template algorithms in other software environments. Hence, minimal delays were incurred.

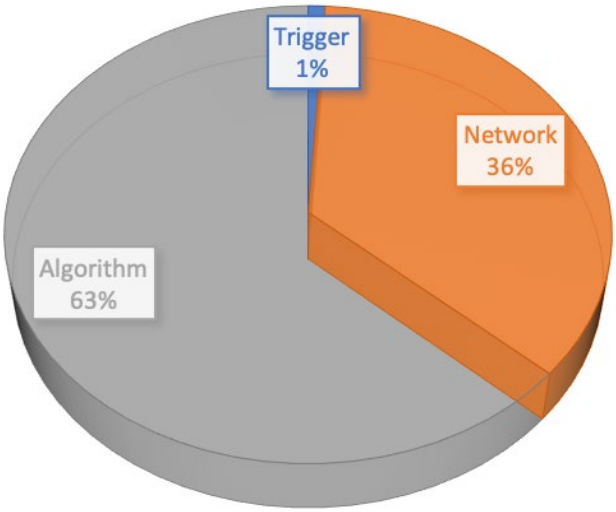

Fig. 11 PiBox mean latency contribution

Table 1 Mean and standard deviation of latency (in microseconds) trials for each PiBox component

Metric	Trigger	Network	Algorithm	Total
Mean	18.53	663.67	1181.82	1864.012
Std. dev.	4.06	95.25	1036.22	1135.53
σ/μ	0.22	0.14	0.88	…

4. Conclusions and Future Steps

A series of support equipment and techniques was demonstrated for the characterization and deployment of EB sensors. Specific-velocity detection limits needed to be specified to operationalize EB sensing in high-speed scenarios. Hence, a lab-compatible apparatus was designed and used to provide fine control over the rotational motion of a high-contrast target. Stepper-motor frequency was programmable via Arduino microcontroller and used to understand the limitations of sensing and algorithms used for velocity extraction. Using a baseline stereo setup, a peak rotational velocity of 200 rpm was observed. Despite the promise of initial results, further investigation is needed, including the optimization of camera biases, lensing, and algorithm parameters for optical flow computation.

Fine-grained hardware signaling and robust communication interfaces are needed not only for range testing of sensing solutions, but for eventual deployment in field platforms. Hence, a prototype based on a RaspberryPi was devised to provide EB sensors a pathway to interface with TTL triggers and networks devices. The PiBox can stream and record data from up to two EB sensors, receive and transmit six TTL signals, and communicate to a host PC via Ethernet or Wi-Fi. Initial tests were

conducted to understand the latencies involved with hardware interfaces, network communication, and onboard processing. After a series of trials, it was found that the majority of time was consumed by algorithm processing. This suggests that intelligent design of processing capabilities or coordinated pursuit of high-performance software libraries and hardware platforms are needed. Furthermore, a large portion of latency was induced by network communication. Thus, it becomes evident that localized computation without off-platform communication becomes important when high-speed decisions are needed.

5. References

1. Gallego G, Delbruck T, Orchard G, Bartolozzi C, Taba B, Censi A, Scaramuzza D. Event-based vision: a survey. 2019. arXiv preprint arXiv:1904.08405.

2. Forsyth D, Ponce J. Computer vision: a modern approach. Prentice Hall; 2011. p. 792.

3. Brandli C, Berner R, Yang M, Liu SC, Delbruck T. A 240×180 130 db 3 µs latency global shutter spatiotemporal vision sensor. IEEE Journal of Solid-State Circuits. 2014;49(10):2333–2341.

4. Mathworks. Using the stereo camera calibrator app: MATLAB & Simulink [accessed 2021 Sep 27]. https://www.mathworks.com/help /vision/ug/using-the-stereo-camera-calibrator-app.html.

5. Benosman R, Ieng SH, Clercq C, Bartolozzi C, Srinivasan M. Asynchronous frameless event-based optical flow. Neural Networks. 2012;27:32–37.

List of Symbols, Abbreviations, and Acronyms

2D	two-dimensional
3D	three-dimensional
AC	alternating current
API	application programming interface
BNC	Bayonet Neill–Concelman (connector)
CPU	central processing unit
EB	event-based
GPIO	general-purpose input–output
IC	integrated circuit
ISR	interrupt service routine
LED	light-emitting diode
lidar	light detection and ranging
PC	personal computer
ROI	region of interest
TCP	Transmission Control Protocol
TTL	transistor–transistor logic
USB	Universal Series Bus

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLD DCI TECH LIB
2 (PDF)	DEVCOM ARL FCDD RLW TD J SENGUPTA B KRZEWINSKI