ARL-TN-1190 ● JAN 2024

Scalable Hardware and Software Procedures for N-ocular Event-Based Stereo System Calibration

by Jonah P Sengupta, Ethan K Eldridge, and Matthew R Devine

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

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-TN-1190 ● JAN 2024

Scalable Hardware and Software Procedures for N-ocular Event-Based Stereo System Calibration

Jonah P Sengupta, Ethan K Eldridge, and Matthew R Devine
DEVCOM Army Research Laboratory

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	END DATE
January 2024	Technical Note	04-01-2023	10-31-2023

4. TITLE AND SUBTITLE
Scalable Hardware and Software Procedures for N-ocular Event-Based Stereo System Calibration

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Jonah P Sengupta, Ethan K Eldridge, and Matthew R Devine

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-TD Aberdeen Proving Ground, MD 21005	ARL-TN-1190

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID IDs: Jonah P Sengupta, 0000-0003-3555-7207; Matthew R Devine, 0000-0002-4139-5884

14. ABSTRACT
Pre-calibrated stereo-based vision approaches offer the capability to resolve 3D feature information instantaneously without the need to register image information and compute camera geometric transformations. Moreover, event-based cameras (EBCs) promise to enhance this capability due to their inherent compression of scene information and high temporal resolution. However, this ability requires careful calibration of the stereo system. Due to their recent inception, current approaches to enable this capability for EBCs are not disseminated amongst the broad community. This report presents a set of hardware and software techniques that can be used and scaled to calibrate N-ocular wide-baseline stereo systems of EBCs. The design of two hardware instruments and two software algorithms are detailed. These approaches are ultimately applied to extract the intrinsic and extrinsic parameters of a four-camera EBC system with millimeter resolution.

15. SUBJECT TERMS
Photonics, Electronics, and Quantum Sciences, camera calibration, event-based cameras, stereo vision, N-ocular system, neuromorphic vision

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	36

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Jonah P Sengupta	(410) 278-5183

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

In recent years, event-based cameras (EBCs) have emerged as a promising alternative to traditional machine vision cameras as they transmit visual information at microsecond-scale latencies over six decades of intensity while consuming milliwatts of power.[1] EBCs can yield such impressive metrics by leveraging in-pixel processing, which only outputs data upon changes in input light intensity. Recent robotics, machine vision, and artificial intelligence research has shown EBCs can be leveraged for high-frequency vibrometry, microfluidics, generative high-frame-rate video synthesis, autonomous obstacle avoidance, and visual odometry.[2–6] However, due to its recent inception and unique data representation, EBCs are without the standardized software suites and established procedures present in traditional frame-based cameras that are needed for high-accuracy photometry, characterization, and camera calibration.[7] In particular, the latter is required to compensate for nonlinear distortion introduced by camera optics, acquire relative camera pose for stereo and robotics applications, and produce high-fidelity depth and triangulation estimates.[8,9]

The desired goal for camera calibration is to retrieve a set of three distinct parameters: the extrinsic, intrinsic, and lens distortion parameters. A pinhole camera model is encompassed by the following set of equations[7,10]:

$$\alpha \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Extrinsic parameters (R and t) describe how the camera coordinate system is rotated and translated with respect to an established world coordinate system. Intrinsic parameters (K) project 3D points within the camera coordinate system to the local image space. Finally, lens distortion parameters compensate for non-affine transformations that are seen when using short focal length or wide-angle optics such as fish-eye lenses. In stereo or N-ocular (containing N cameras) systems, rigid transformations between cameras (relative extrinsic parameters) need to be found as well.

In both monocular and N-ocular scenarios, camera calibration involves presenting a series of 2D or 3D patterns (typically a checkerboard or periodic dots) with known dimensions to one or all cameras simultaneously. Since the size of each element in the presented pattern is known, 3D coordinates of each element corner or centroid can be constructed. For traditional cameras, well-established feature extraction methods (such as edge, corner detectors, and Hough transforms) can be used to localize the checkerboard corner or dot centroid with subpixel accuracy. Once these 3D–2D correspondences have been established, there are conventional closed-form solutions that can be refined with optimization approaches, which can be used to find camera intrinsic parameters and lens distortion coefficients.[11,12] By presenting the pattern to all cameras in the N-ocular simultaneously, similar calibration methods can be extended to extract the rigid transformation between each pair in the apparatus.[12]

Calibrating a singular or set of EBCs is like the methods described previously, but with one important caveat. Since the camera is only sensitive to changes in illuminance, static planar or 3D patterns cannot be used to establish 3D–2D correspondences. Instead, prior work used custom instruments and event-based feature extractors to achieve the same purpose. Researchers traditionally used a blinking checkerboard pattern on an electronic display to present to the EBC. Since the whole pattern is oscillating at a fixed frequency, the EBC captures an image that appears to be a static checkerboard.[13,14] These images can be integrated and processed with traditional feature extractors to provide the requisite 3D–2D correspondences. Prior work also used sets of blinking LEDs arrayed on the surface of a cube. Like dot patterns used for traditional cameras, each LED provides a stable point source that can be integrated and localized in the resultant image.[15] Finally, other approaches have used a self-calibration method that leveraged a moving setup with a complementary metal oxide semiconductor (CMOS) camera and EBC pair.[6] Features in the CMOS camera can be correlated to scene contours detected by the EBC. In tandem with knowledge of each camera's intrinsic parameters, these correspondences can be used to compute the relative pose of the EBC.

Despite the great progress made by event-based vision researchers, calibration of N-ocular event-based systems has always been an ancillary topic. There has not been any literature devoted to the construction of hardware and software approaches for the purpose of calibration. This work presents two different calibration instruments, two feature extraction methods, and application of the techniques to a setup of four EBCs. Section 2 describes the design and implementation of hardware to establish transient, planar patterns. Section 3 outlines the feature extractors used to establish the requisite 3D–2D correspondences. Section 4 presents the calibration results for each software method before Section 5 concludes the work and outlines future steps.

2. Blinking Pattern Hardware

In a stereo system, it is well established that depth estimation accuracy and resolution is directly proportional to the camera baseline or linear separation of the cameras:

$$\Delta z = \frac{z^2}{f \cdot b} \Delta d$$

such that b is the baseline, z is depth, f is focal length, and d is disparity. However, for setups with wider baselines, the system's field of view (FOV) is further constricted such that minimum working distance (z_{min}) is directly proportional to the baseline as well:

$$z_{min} = \frac{b}{(\tan\left(\frac{\theta_{FOV,r}}{2}\right) + \tan\left(\frac{\theta_{FOV,l}}{2}\right))}$$

such that $\theta_{FOV,(r,l)}$ are the right and left cameras' angular FOVs, respectively. Thus, calibration techniques used for stereo systems with wider baselines need to be functional from a longer distance from the system. For this reason, blinking checkerboards on liquid-crystal display or organic LED displays are not feasible and do not scale in these situations as it would require larger, unwieldy pieces of hardware.

As an example: for a stereo setup of video graphics array (VGA) cameras with pixel size = 15 µm, focal length = 7 mm, baseline = 0.65 m, and 75% stereo FOV, it can be shown that a minimum working distance (WD) of 0.95 m and calibration instrument size of 0.54 m is required when a 5 × 6 checkerboard pattern is used. Tablets and displays that can accommodate such a pattern do exist. However, when scaling the baseline to 1 m, a minimum of a 0.84-m (~33-inches) display is required. This would require using a large LED display, which is not feasible for calibration in many situations. Conversely, using a 2 × 2 blinking dot pattern in the latter scenario would require a 0.75-m instrument whose electronics are inexpensive, and a structure that can be 3D printed. The math to accompany the preceding claims can be found in Appendix A.1. The two approaches presented here use the blinking dot methodology to create calibration instruments that can be easily scaled to accommodate wider baselines.

2.1 Blinking LED Pattern

Figure 1 shows a prototype for an LED calibration pattern that can be used to calibrate an event-based stereo system. Each LED is 10 mm wide and arranged in a 2 × 2 array with 0.2 m spacing. This was designed to accommodate a system like the one described prior, but with baseline of 0.65 m. Minimum WD to achieve a 50% coverage with the overlapping stereo FOV is 0.95 m. Each LED row is driven by a dedicated pull-down transistor. The gate of each device is tied to a digital signal that is controlled by a microcontroller on the back of the LED panel. Upon deassertion of these signals, the LED cathode is pulled high by a pull-up resistor that is in parallel with each LED. Sizes of these resistors were chosen to attenuate current draw through the LED. Anode voltage of 5 V was provided by the microcontroller reference.

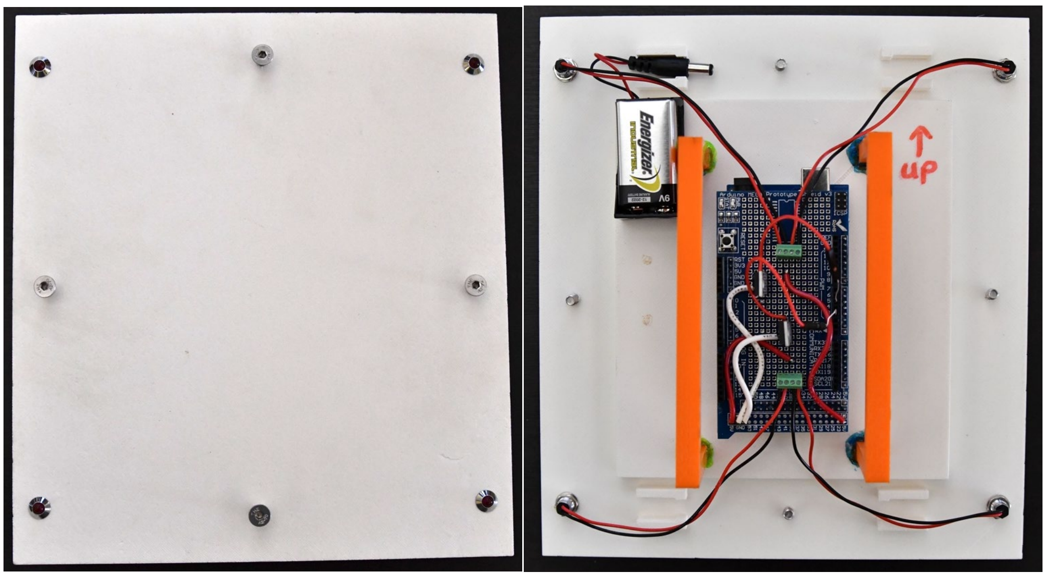

Fig. 1 LED calibration pattern front (left), and LED calibration back: circuitry and battery (right)

Distinct switching devices for the top and bottom rows provide independent frequency programming. The different top row frequency allows the calibration algorithm to orient the panel and determine the pattern pose relative to the camera. The programmable frequency range is bounded by the frequency response of the EBC. For a given set of camera biases, characterization of these devices has shown a maximum periodic response on the order of 1–2 kHz.[1] Battery power is used to ensure the calibration instrument can be maneuvered at will and leveraged in a variety of scenarios.

2.2 Rotating Dot Pattern

The flashing dot pattern described previously is an effective, scalable solution for calibrating wide-baseline event-based stereo systems under a variety of indoor- or low-lighting conditions. However, techniques that use active illumination are required to scale their output optical power to compensate for the longer WDs needed for wide-baseline stereo systems. Furthermore, outdoor calibration of the EBC becomes difficult with the LED pattern. Due to the bright background, a lower temporal contrast signature is produced by each LED at the required WD making the event-based frequency detection task more complex. A potential solution would be replacing the LEDs with higher-power versions and modifying the switching circuitry. However, this only highlights the lack of scalability seen for the blinking LED pattern in outdoor environments with longer baseline stereo systems.

In contrast, the EBC is stimulated in a similar manner by an instrument that uses reflective light and a rotating element to shutter an aperture. Figure 2 shows the two layers of 3D-printed material used for the approach. The top layer closely resembled the pattern used in the LED calibration technique: it is a 2 × 2 square array of circles with identical diameters. The second layer is a circular disk with two concentric circles whose circumferences are composed entirely of dots. These dots have diameters greater than or equal to the dimensions of circles on the first layer. A third layer comprised of reflective material (lightweight polycarbonate mirrors) is placed underneath the circular patterns. During operation, the second layer is rotated by a motor while the first and third are static. When a dot in the second layer is aligned with a circle on the first, the aperture is fully open, and light is reflected by the third layer. When the dot and circle are unaligned, light is partially absorbed by the second layer and the illuminance detected by the EBC is reduced. Thus, oscillating stimuli are produced by the pattern with frequency:

$$f_{stim} = f_{mot} \cdot N_{dots}$$

such that f_{mot} is the rotating frequency of the motor and N_{dots} is the number of dots on the circumference of a circle on the rotating layer. To produce the two unique frequencies needed for a backend algorithm, the circular and square patterns on the second and first layer, respectively, are offset from one another. This is done to ensure the top and bottom row of circles on the first layer are aligned with two different circles on the second layer. These circles are composed of a different number of dots; hence, they produce two distinct frequencies. The geometric expressions used to derive the pattern shapes and sizes can be found in Appendix A.2.

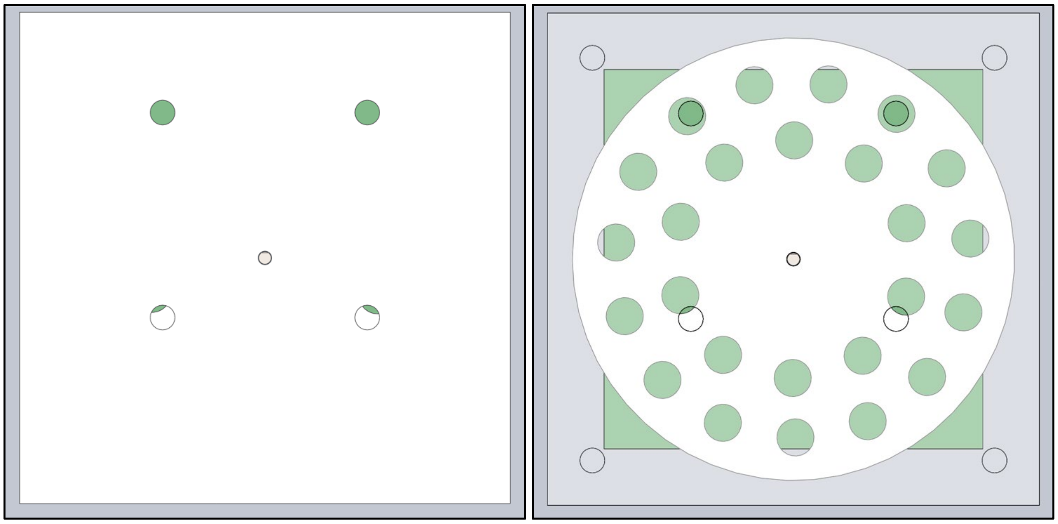

Fig. 2 Rotating disk 3D design: first layer with square dot pattern (left) and second layer with two concentric dotted circles and third layer of reflective material (right)

3. Event-Based Calibration Software

As mentioned previously, the goal of a calibration algorithm is to first establish 2D localization of the stimuli within the image plane before using a series of these 3D–2D correspondences from each camera to extract camera intrinsic and extrinsic parameters. The approach adopted to calibrate the event-based stereo systems using the aforementioned hardware leveraged a blend of custom and open-sourced algorithms. The subsequent sections describe the two feature detection methods used and give an overview of the iterative routine deployed to extract the extrinsic parameters of the stereo system.

3.1 Pattern Detection

3.1.1 Frequency Detection and Clustering

The first approach of pattern detection used an event-based algorithm that detected temporal frequencies within the camera event stream and clustered similar signatures to produce centroids. These steps were realized in the Metavision Intelligence software development kit (SDK). This software platform realizes a multilanguage application programming interface that can interface with Prophesee EBCs and recorded data sets. In addition, it also includes a set of event-based computer vision algorithms that can be leveraged to filter noise, detect salient features, and track objects in the event stream. Two of these algorithms were leveraged in tandem to detect and localize the calibration pattern in the image plane.

The first step of the detection routine involves detecting temporal frequencies by recording the time intervals between events for each pixel address. A valid measurement requires N_{filt} time interval or period measurements. Moreover, these period samples need to be within T_{meas} of each other or else the sample is invalidated. Low- and high-frequency noise events are also filtered out by instituting a minimum and maximum bound on detection frequencies. These parameters lead to the detection of stable temporal stimuli. If all conditions are met, then the algorithm produces an output event that has the same address as the pixel but is appended with the detect frequency.

These frequency events are then clustered to localize the center of each temporal stimuli. Clusters are established by connecting frequency events that pass a set of spatiotemporal conditions.

1. These events must be within one pixel distance of one another.
2. The detected frequencies must be within F_{diff} Hz.
3. The frequency event (absolute time when it was generated) must occur within T_{diff} of the candidate cluster's last update.

If a frequency event meets all these conditions, the cluster is subsequently moved in the direction of the event address and the cluster size grows by one. If the cluster contains more than N_{ev}, the algorithm outputs detect feature centers with subpixel accuracy.

3.1.2 Stereo Detection Routine

Stereo calibration requires the simultaneous capture of the calibration pattern within each EBC's FOV. However, this assumes that camera output and pattern detections are closely synchronized in time. Prophesee's Metavision SDK has not implemented the requisite software needed for stereo calibration, so a custom procedure needed to be developed. As a result, a custom Python method was designed that can be scaled for parallel capture of calibration patterns in N-ocular stereo systems. This latter behavior involved deploying a parent thread, which synchronized the pattern capture across each camera thread. Figure 3 provides a graphical depiction of the relationship between the parent and child threads.

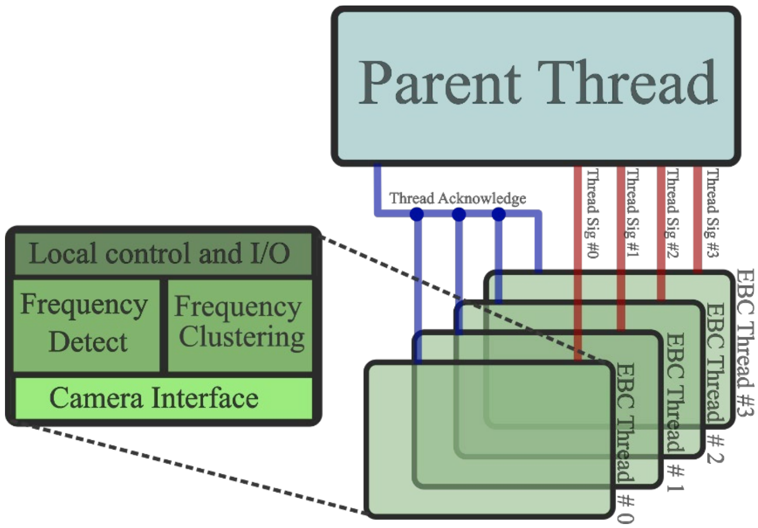

Fig. 3 Structure of multithreaded pattern capture. Each child thread (green) interfaces with an EBC via USB serial connection. It then performs frequency detection algorithms locally before signaling to the parent thread (red). The parent thread (cyan) coordinates parallel capture of the patterns via the thread acknowledge (blue) and terminates the program upon successful registration of the images.

The number of threads deployed is contingent on an accompanying JSON file that maps the EBC serial numbers to the detected devices on the USB interface. This allows the user to configure and deploy one to N cameras in parallel where N is the number of devices simultaneously connected to the PC. Each camera thread reads from the USB 3.0 serial interface, decodes the information, and batches read events in T_{int} intervals. These events are accumulated into a 2D frame, which is output to a display window that provides the user with feedback regarding the orientation of the pattern with respect to that camera's FOV. Moreover, since the camera's clocks are synchronized, frame generation is approximately aligned as well. However, a caveat: display windows may become misaligned due to how Python deploys the thread.

In tandem, each thread has an instance of the event-based frequency detection and clustering algorithm. The temporally binned batch of events are passed through the local algorithm, which returns a set of valid clusters. Once four clusters with reasonable separation are detected, the thread asserts a flag to the main thread. The main thread waits for all child threads to indicate that the pattern is detected before asserting a signal in return. Upon reception of this signal from the parent thread and local reaffirmation that the four clusters are still valid, four actions are taken by the local thread:

1. Cluster centers are stored by each local thread.
2. The accumulated event image is saved.

3. A local pattern counter increments.
4. The child deasserts the detection flag to the parent.

This four-phase handshake has two advantages. One, it is agnostic to the number of child threads. Two, it enables the simultaneous capture of calibration points across all camera threads despite the asynchronous nature of the data and thread instances.

Robust extraction of extrinsic parameters requires the registration of patterns spanning the stereo FOV and in a variety of poses. Thus, there is a programmable pause inserted after each successful stereo capture to allow the user to reorient the pattern. After N_{patt} patterns are captured, the counter in each camera thread triggers recording of the images and 2D detections to the disk. After these files are saved, each local thread terminates, which enables the main thread to exit.

3.1.3 Hough Circle Detection and Localization

As mentioned prior, each camera captures the detected patterns in a set of images that are saved locally. These images are then postprocessed by another algorithm to produce another set of 2D detections. A complementary, frame-based approach was used for two reasons: to highlight the potential benefit of the event-based algorithm, which leverages the high-resolution temporal information, and to provide a secondary means that can be used in tandem to provide a final extrinsic estimate.

Calibration patterns were detected in the saved images by applying a sequence of noise-filtering and feature-extraction steps. First, a median filter with kernel size of 3 was used to remove uncorrelated spatial noise in each frame. Next, a Hough transform was applied to extract circles within the image. As shown in the left panel of Fig. 4, events stimulated by rotating disk or blinking LED patterns appear as small circles within the image when integrated over time. Thus, a Hough circle transform was used to extract circular features along with their respective radii and centroids. More information about the Hough transform can be found elsewhere.[16] Analysis was conducted on a pair of EBC images. Once at least four circles with non-overlapping radii were extracted in each frame, the algorithm deemed that a good set of centers was found. The detected patterns were then appended to output data, which was ultimately used for stereo calibration.

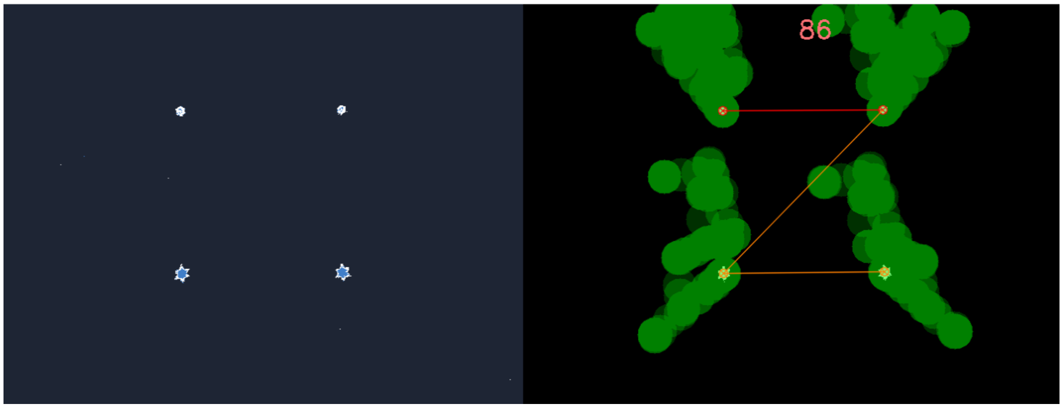

Fig. 4 EBC registration of LED calibration pattern (left) and Metavision intrinsic calibration routine with pattern pose and past patterns embedded (right)

3.2 Intrinsic and Extrinsic Parameter Computation

After collecting the calibration points using either the event- or frame-based method, two algorithms were used in succession to extract the intrinsic and extrinsic parameters for the stereo setup, respectively.

3.2.1 Monocular Calibration

Open-source, vendor software from Prophesee was used to extract the camera matrices for each EBC. This used OpenCV functions to compute the optimal camera matrix and lens distortion coefficients for provided 3D–2D correspondences. After computing these parameters, the reprojection error (RPE) is computed for each set of detected patterns. The patterns that correspond to the highest RPE are removed and camera calibration is conducted again. This is repeated until the root-mean-squared (RMS) RPE converges. Once this occurs, the calibration routine is completed, and intrinsic parameters are returned. An example capture of successive patterns in the software is seen in Fig. 4.

3.2.2 Stereo Calibration

Since stereo calibration is out of scope of the current vendor software suite, a custom routine had to be developed. The general procedure closely mirrors the method used for monocular calibration. However, stereo calibration requires the camera matrices and detected patterns for each EBC in the desired pair. Detected patterns for the stereo calibration were provided by one of the two synchronized detection algorithms previously described. After loading the detections, they were each reordered to ensure the pattern is oriented the same in both camera views. Specifically, this step ensured that extracted feature centers for each pattern are ordered by top right (TR), top left (TL), bottom right (BR), and bottom left (BL) features. These ordered patterns and camera matrices are then used as input to the OpenCV stereo calibration routine. This function returns the stereo pair's essential, fundamental, rotation, and translation matrices along with the RPE for each pair of detections.

Like the monocular calibration routine implemented in the vendor software, the custom algorithm uses an iterative approach to refine the extrinsic parameters. This approach is parameterized by four values: number of maximum iterations, minimum RPE, initial discard threshold (θ_{thr}), and discard threshold decrease rate (R_θ). The former two values determine when to terminate the algorithm. The latter two are involved with determining if a pair of detected patterns should be discarded. If an RPE is above the median RPE by θ_{thr}, then it is discarded. However, if the RMS RPE converges prematurely (either before reaching the minimum RPE or maximum number of iterations), then the $\theta_{thr}$is multiplied by the R_θ. This ensures that the algorithm does not settle at a local minimum and estimates the extrinsic parameters with nonoptimal patterns. Also, the algorithm uses median RPE rather than mean RPE as the means to filter out bad patterns. This metric was chosen as the median more accurately captures the central tendency of the RPE distribution and is less influenced by bad RPE outliers. It was also seen that using the mean RPE resulted in convergence, which included pattern pairs with larger RPEs.

After successfully proceeding through the stereo calibration algorithm, the projection matrix for each EBC can be constructed. As shown in Section 1, this is done by performing a matrix multiplication between the camera matrix and the extrinsic parameters. Approximate poses of each pattern in 3D space can then be computed using the 2D detections and projection matrices for each camera. This is realized using the OpenCV triangulation method and subsequently applying the extrinsic parameters to the 3D points. Resulting pose information with respect to the stereo setup is shown in Section 4.

4. Results

4.1 Monocular Calibration Verification

The hardware calibration patterns were verified in tandem by using each to extract the intrinsic parameters of a single EBC. Table 1 outlines the pattern geometries, temporal frequencies, and experimental information used. The WD was scaled linearly to ensure the projected pattern size was the same for each instrument. LED and rotating disk method (ROT) denote the parameters used for calibration with the blinking LED and rotating disk patterns, respectively.

Table 1 Experimental parameters for monocular EBC calibration

Method	Spacing (mm)	fspec (Hz)	fnorm (Hz)	WD (mm)
LED	200	250	100	890
ROT	83	75	50	365

Pattern images were acquired using the Metavision software suite, but calibration was conducted in a similar manner as described in the stereo routine. Dot centroids and pattern centers were extracted from each image using a Hough circle transform. This was then used in tandem with the 3D geometry of the patterns to compute the intrinsic parameters of the camera. Patterns that produced outlier RPE metrics were trimmed, and calibration results were recomputed. This iterative approach yielded the RPE and intrinsic parameters are reported in Table 2.

Table 2 Intrinsic parameters and median RPE for each calibration method

Method	f_x (mm)	f_y (mm)	c_x	c_y	k_1	k_2	k_3	k_4	k_5	RPE
LED	8.298	8.298	326	226	–0.0046	–0.0005	–0.001	–0.0011	3.75E–06	0.129
ROT	8.211	8.211	345	228	–0.0185	–0.0006	0.00042	–0.0015	4.31E–05	0.19759

Notes: Focal lengths are computed by dividing the camera scale factors by the pixel size. $f_{x,y}$ is the lens focal length in mm, c_x and c_y are the coordinates of the principal point in the image plane, k_1–k_5 are the lens distortion coefficients.

As shown in Table 2, monocular calibration with the two instruments resulted in similar intrinsic parameters, most notably the focal lengths and principal points. The rotating disk instrument resulted in calibration patterns with higher RPE. Since calibration took place in an indoor environment, this could be attributed to opening the EBC aperture to accommodate stimuli capture. Using the ROT also requires the need for better alignment with the sensor: any yaw or pitch in the instrument with respect to the camera will also transform the shapes of the apertures and introduce potential localization errors when using the Hough method. This is due to the instrument's use of reflectance rather than active light transmission, thus making circles look like ellipses when the instrument is transformed in specific ways.

4.2 Stereo Calibration of a Quadrocular Apparatus

Stereo camera calibration was subsequently conducted by capturing the blinking LED pattern simultaneously with pairs of EBCs and using the previously described methods to extract the stereo extrinsic parameters. This approach was deployed on the quadrocular EBC setup shown in Fig. 5. The commercial off-the-shelf EBC used in this experiment was a SilkyEvCam-VGA from CenturyArks whose core is the 640 × 480 PPS3MVCD image sensor from Prophesee.[17] Each EBC was equipped with a 4–12-mm telephoto lens, USB 3.0 cable, and modified hardware signaling cable to ensure synchronization of event timestamps. In this setup, the top-right camera was the designated master. For these cameras, the master EBC sources the timestamp clock, which was used by the other three cameras to synchronize their timestamps. Width- and lengthwise spacing of the EBCs on the stand were 476.25 and 679.45 mm, respectively.

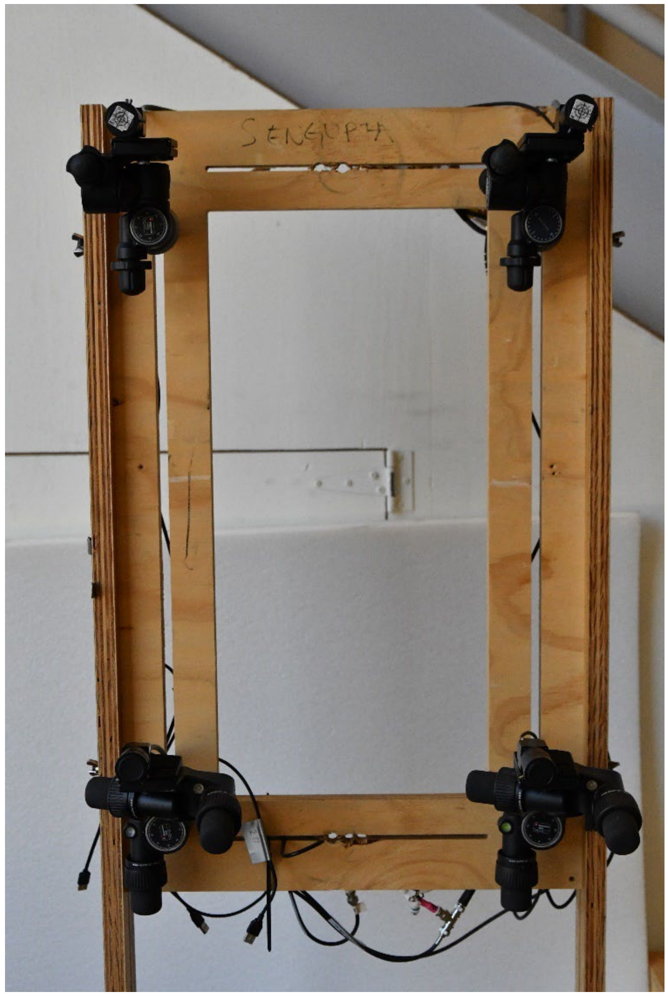

Fig. 5 Quadrocular EBC stereo setup: four SilkyEvCam cameras are mounted on a custom stand. Counterclockwise starting from the top right, serial nos. for each part are: 291, 290, 289, and 288.

Prior to stereo calibration, each EBC was first calibrated individually using the Metavision detection and monocular calibration routines to extract lens distortion parameters and camera matrices. Blinking LED frequencies were programmed to 250 and 100 Hz for the top and bottom rows, respectively. Given the wider baselines seen for this setup, focal length was adjusted to roughly 8 mm to allow for a shorter WD. Fifty patterns were captured and used by the vendor software to compute the intrinsic parameters. Results are shown in Table 3. Each calibration resulted in RMS RPE less than half a pixel and computed focal length close to the configured value. However, the computed principal point (c_y, c_x) varied from the established origin of 320, 240. This could be attributed to a lack of uniformity of pattern presentations across the EBC's FOVs.

Table 3 Intrinsic calibration results for each camera in quadrocular setup. Each calibration was conducted using the blinking LED pattern and provided vendor software.

EBC	No. of patterns	Focal length (mm)	Principal point X (pix)	Absolute X error (pix)	Principal point Y (pix)	Absolute Y error (pix)	RMS RPE
288	50	7.80	315	5	260	20	0.29
289	48	7.94	315	5	260	20	0.29
290	48	7.86	330	10	249	9	0.30
291	47	7.73	302	18	255	15	0.21

Due to communication bottlenecks and lack of laptop processing capabilities, the present version of the algorithm could not accommodate simultaneous calibration of all four cameras. Instead, three pair-wise calibrations were conducted: TR/TL (291/290), TR/BL (291/289), and TR/BR (291/288). Fifty pattern detections and corresponding images were captured by each pair of cameras. After processing the captured image data through the Hough transform detection algorithm, each set of pattern detections were used to extract the stereo pair's extrinsic parameters using the custom calibration algorithm. Convergence data for each stereo pair and detection algorithm is shown in Fig. 6. For this calibration routine, the stereo algorithm terminated after 10 iterations, or if a median RPE of 0.5 was achieved across all the patterns used. θ_{thr} was used to filter the patterns. R_θ was set to 0.75 to prevent premature convergence.

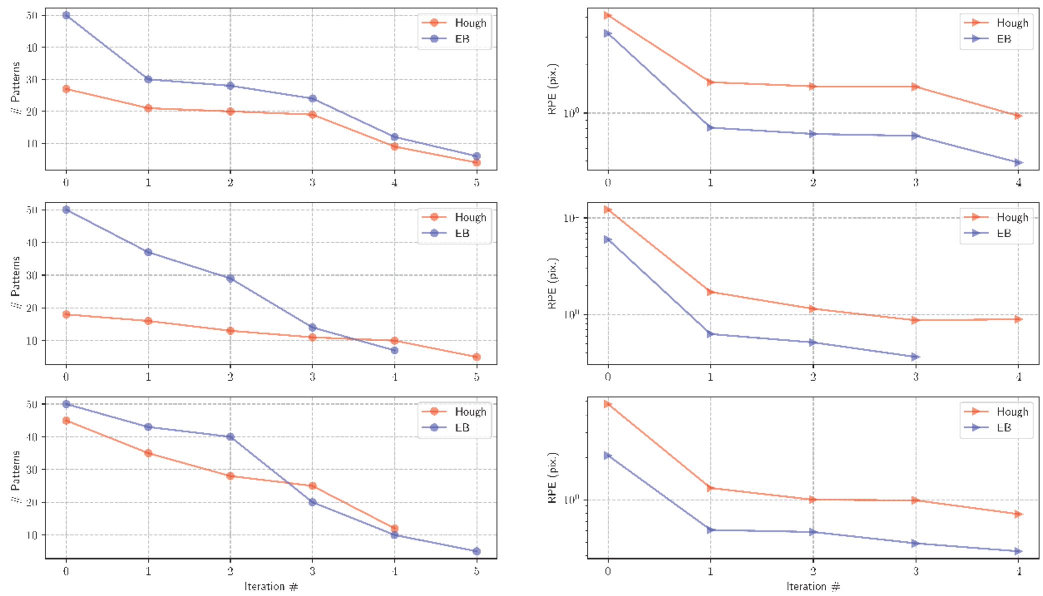

Fig. 6 **Stereo calibration algorithm convergence results for the three stereo pairs. Top, middle, and bottom rows correspond to TL/TR (290–291), BR/TR (289–291), and BL/TR (288–291) pairs, respectively. The first and second columns correspond to the number of patterns used for calibration and median RPEs, respectively.**

Since the Hough method's postprocessed images were extracted using the event-based detection method, the approach consistently produced fewer initial patterns than the event-based detection approach. However, both methods yielded final calibration results that used a similar number of patterns. Ultimately, both algorithms only needed to leverage 9–14 patterns to achieve calibration results that resulted in a median RPE of 0.5–1 pixels. Another consistent observation was the large initial drop in RPE after the first stereo calibration iteration. This is due to discarding the outliers that were acquired during pattern transitions. Contours produced by operator movement routinely caused spurious LED localization results. Subsequent large drops in median RPE were because of decreasing the threshold used to discard bad patterns.

Tables 4 and 5 present the stereo calibration results when using the event-based and Hough transform detection methods, respectively.

Table 4 Extrinsic calibration results using event-based clustering method: bottom right data reflects the mean extrinsic error and RPE for all pair-wise calibrations. Event-based algorithm parameters reflect the min/max frequency, minimum cluster size, and maximum cluster temporal difference, respectively.

Pair	Initial patterns	Final patterns	Algorithm parameters	Computed distance (mm)	Measured distance (mm)	Absolute extrinsic error (mm)	Median RPE
290–291	50	10	100,250,10, 10000	487.68	476.25	11.43	0.43
289–291	50	14	100,250,10, 10000	821.94	829.57	7.63	0.36
288–291	50	12	100,250,10, 10000	680.72	679.25	1.47	0.49
					Average:	**6.84**	**0.43**

Table 5 Extrinsic calibration results using Hough transform detection method: bottom right data reflects the mean extrinsic error and RPE for all pair-wise calibrations. Hough parameters reflect the OpenCV HoughCircles function arguments.[18]

Pair	Initial patterns	Final patterns	Hough parameters	Computed distance (mm)	Measured distance (mm)	Absolute extrinsic error (mm)	Median RPE
290–291	45	12	1,50,5,4,2,15	473.46	476.25	2.79	0.79
289–291	18	10	1,50,5,4,2,15	855.98	829.57	26.40	1.09
288–291	27	9	1,50,5,4,2,15	698.50	679.25	19.25	0.96
					Average:	**16.15**	**0.95**

Since the approximate distance between cameras can be measured from the test fixture, the difference between the computed and actual locations can be extracted. This absolute extrinsic error (AEE) can then be used in tandem with the median RPE to evaluate the accuracy of stereo calibration routine. Results in Tables 4 and 5 show that the event-based detection method consistently outperformed the Hough transform with respect to median RPE and AEE. This can be attributed to the event-based algorithm's capability to consume and leverage temporal information. Fine-grained detection of precise temporal frequencies allows the algorithm to reject other uncorrelated stimuli, extract pattern pixels, and compute cluster centroids with subpixelic accuracy.

In contrast, the Hough transform processes images of accumulated events, which essentially discards the temporal information produced by the blinking LED pattern. Instead, this method operates using the edges of each circle, which potentially incurs additional localization error. A future improvement to the Hough transform algorithm would be to use the accumulation time into the processing flow. This information could be used to filter out any pixels that exceed $\frac{f_1}{fps}$ or are below $\frac{f_2}{fps}$.

After extracting each camera pairs' extrinsic parameters, the relative pose of each pattern can be found. Pattern pose is extracted by computing the 3D point of each LED on the pattern and creating a shape. Triangulation of these points is performed using the OpenCV triangulation method. This function uses an iterative least-squares method to find the 3D point that best projects to the 2D point in each camera view given each camera's projection matrix. Projection matrices were found by computing the matrix product, the camera matrix, rotation matrix, and translation vector. Upon completion, the triangulation function yields a Nx4x4 tensor of homogenous coordinates where N is the number of patterns used for calibration of the stereo pair. After converting the 4D points to 3D coordinates, each set of points was transformed with respect to the stereo coordinate frame and used to draw a pattern polygon.

The resulting patterns are shown in Figs. 7 and 8 for each stereo pair from the Z- and X-axis perspectives, respectively. Figure 8 is of particular interest as it shows the approximate WD used for stereo calibration of each pair. For the TL/TR pair (290/291), a shorter WD of 1 m was used since that pair had the shortest stereo baseline. In contrast, the BR/TR (289/291) pair required a WD of 1.5 m due to a wider baseline of 0.83 m.

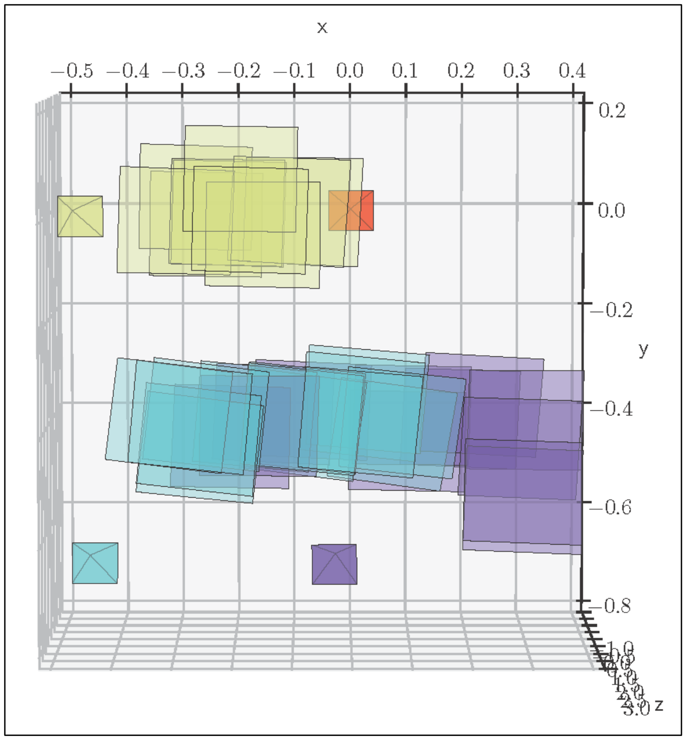

Fig. 7 Calibration pattern poses for each pair of stereo calibrations from Z-axis perspective using extrinsic parameter data extracted using the event-based detection method. Each pattern is colored with respect to the pair: yellow = TL/TR pair, blue = BR/TR, and purple = BL/TR.

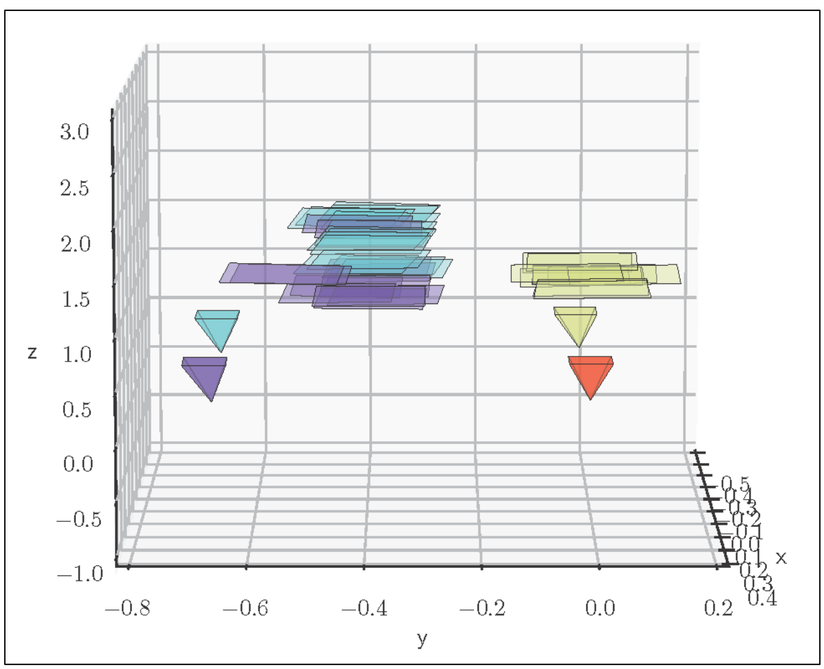

Fig. 8 Calibration pattern poses for each pair of stereo calibrations from X-axis perspective using extrinsic parameter data extracted using the event-based detection method. Each pattern is colored with respect to the pair: yellow = TL/TR pair, blue = BR/TR, and purple = BL/TR.

Figure 9 outlines a comparison of the extrinsic parameters computed using the event-based and Hough transform methods. This figure highlights the AEEs presented in Tables 2 and 3. It is shown that the event-based method (green) yields extrinsic parameters that more closely align with the actual baselines seen in the test apparatus. These small disparities matter, as they directly impact the ability of downstream algorithms to accurately track and localize objects in 3D space.

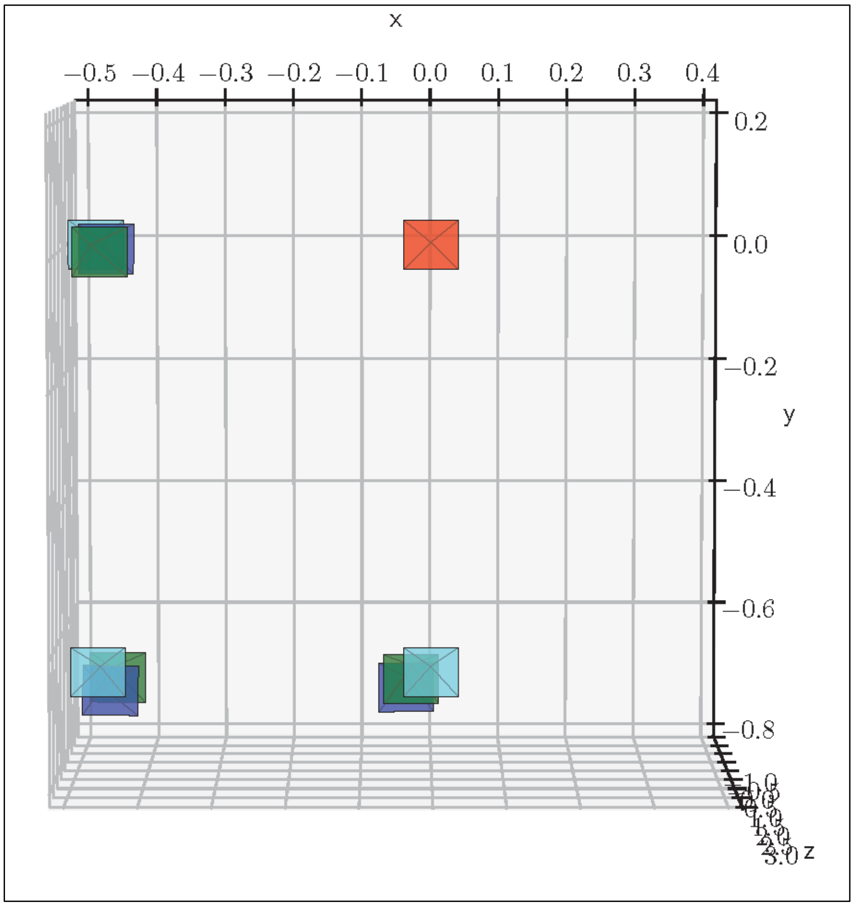

Fig. 9 Camera positions computed from the stereo calibration algorithm with respect to ground truth (cyan). Green and blue camera pyramids correspond to extrinsic parameters extracted using the event-based and Hough transform detection methods, respectively.

5. Conclusion

This report presents a set of custom hardware instruments and software algorithms that were used to calibrate event-based stereo systems. EBCs are a relatively new technology whose operation is not compatible with traditional calibration techniques or established software procedures. Furthermore, current approaches used to calibrate EBCs focus on single or small baseline systems. Thus, a pair of instruments were designed and deployed to extract the intrinsic and extrinsic parameters of stereo systems.

A blinking LED array was used to provide separate stimuli to the EBC with distinct, accurate temporal frequencies. This approach was effective under indoor or dim lighting conditions but fails to scale for wider baselines or outdoor use. In contrast, a spinning disk with a shuttered aperture was used to stimulate the array. This approach used a reflective backing that yielded better performance in outdoor settings. These approaches were used in tandem with a custom stereo calibration algorithm that deployed one of two pattern detection approaches: an event-based temporal frequency detector or frame-based Hough transform. When deployed to calibrate a quadrocular system, the iterative stereo algorithm yielded accurate extrinsic parameters when using the event-based detector.

Future efforts will focus on scaling and deploying the rotating disk method for calibrating large baseline stereo setups (>1 m separation) in outdoor settings. In addition, multithread software used for parallel capture of temporal patterns will be optimized to ensure the simultaneous calibration of greater than two cameras. After that is accomplished, the software suite and 3D designs for the hardware instruments will be consolidated and open-sourced to improve the calibration capability of the event-based imaging community.

6. References

1. Lichtsteiner P, Posch C, Delbruck T. A 128× 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. IEEE J Solid-State Circuits. 2008;43(2):566–576.

2. Niwa R, Fushimi T, Yamamoto K, Ochiai Y. Live demonstration: event-based visual microphone. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 4053–4054.

3. Ni Z, Pacoret C, Benosman R, Ieng S, Régnier S. Asynchronous event-based high speed vision for microparticle tracking. J Microsc. 2012;245(3):236–244.

4. Pan L, Scheerlinck C, Yu X, Hartley R, Liu M, Dai Y. Bringing a blurry frame alive at high frame-rate with an event camera. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 6820–6829.

5. Falanga D, Kleber K, Scaramuzza D. Dynamic obstacle avoidance for quadrotors with event cameras. Sci Robot. 2020;5(40):eaaz9712.

6. Censi A, Scaramuzza D. Low-latency event-based visual odometry. 2014 IEEE International Conference on Robotics and Automation (ICRA); 2014. IEEE.

7. Bouguet J-Y. Camera calibration toolbox for Matlab (1.0). Caltech Library; 2004 May 4. http://www.vision.caltech.edu/bouguetj/calib_doc/.

8. Carneiro J, Ieng SH, Posch C, Benosman R. Event-based 3D reconstruction from neuromorphic retinas. Neural Netw. 2013;45:27–38.

9. Shah M, Eastman RD, Hong T. An overview of robot-sensor calibration methods for evaluation of perception systems. Proceedings of the Workshop on Performance Metrics for Intelligent Systems; 2012 Mar. p. 15–20.

10. Hartley R, Zisserman A. Multiple view geometry in computer vision. Cambridge University Press; 2003.

11. Zhang Z. A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell. 2000;22(11):1330–1334.

12. Sturm P. Algorithms for plane-based pose estimation. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition; 2000 June. CVPR 2000 (Cat. No. PR00662) vol. 1, p. 706–711. IEEE.

13. Benosman R, Ieng SH, Rogister P, Posch C. Asynchronous event-based Hebbian epipolar geometry. IEEE Trans Neural Netw. 2011;22(11):1723–1734.

14. Muglikar M, Gehrig M, Gehrig D, Scaramuzza D. How to calibrate your event camera. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 1403–1409.

15. Camuñas-Mesa LA, Serrano-Gotarredona T, Ieng SH, Benosman RB, Linares-Barranco B. On the use of orientation filters for 3D reconstruction in event-driven stereo vision. Front Neurosci. 2014;8:48.

16. Illingworth J, Kittler J. A survey of the Hough transform. Comput Vis Graph Image Proces. 1988;44(1):87–116.

17. CenturyArks Co, Ltd. SilkyEvCam (VGA). CenturyArks Co, Ltd; n. d. [accessed 2023 Aug 15]. https://centuryarks.com/en/silkyevcam-vga/.

18. HoughCircles. OpenCV: feature detection. HoughCircles; c2023 [accessed 15 Aug 2023]. https://docs.opencv.org/4.x/dd/d1a/group__imgproc__feature.html

Appendix. Calibration Feature Requirements and Design Derivation

A.1 Stereo System Pattern Constraints

The following is the derivation of expression to guide calibration instrument design for a given stereo setup. Input design definitions are:

- pixel size in meters (y_{pix})
- sensor size ($N_{pix,sensor}$)
- baseline (b)
- focal length (f)
- pixels per detected feature ($N_{pix,feat}$)
- minimum separation between features ($N_{pix,sep}$)
- aggregate, overlapping field of view (FOV) from the stereo pair expressed as percentage desired for stereo calibration (pct_{fov})
- number of rows for the calibration pattern ($Nrows_{cal}$)

Total number of pixels in the y dimension needed for valid calibration detection is:

$$N_{pix,pattern} = \left(N_{pix,feat} + N_{pix,sep}\right) \cdot Nrows_{cal} + N_{pix,feat}$$

Sensor size in meters is:

$$y_{sensor} = y_{pix} \cdot N_{pix,sensor}$$

Angular field of view can be found using the geometric relation:

$$\theta_{FOV} = 2 * \tan(\frac{y_{sensor}}{2f})$$

Given an additional distance (Δz) beyond the minimum working distance for the stereo setup with baseline b and angular field of view, the largest object dimension that can be detected in the overlapping FOV is:

$$y_{obj} = 2\Delta z \cdot \tan\left(\frac{\theta_{FOV}}{2}\right)$$

Projecting the object dimension to the image plane:

$$y_{img} = \frac{f \cdot y_{obj}}{z_{min} + \Delta z} = \frac{2f \cdot \Delta z \cdot \tan\left(\frac{\theta_{FOV}}{2}\right)}{z_{min} + \Delta z}$$

Dividing both sides by y_{pix} , the expression can be placed in terms of pixels:

$$N_{pix,obj} = \frac{2f \cdot \Delta z \cdot \tan\left(\frac{\theta_{FOV}}{2}\right)}{(z_{min} + \Delta z) y_{pix}}$$

By setting $N_{pix,obj} = \left(pct_{fov} \cdot N_{pix,sensor}\right) = N_{pix,fov}$, an expression can be derived to find the minimum additional working distance to encompass the desired region within the overlapping stereo FOV. After substituting and rearranging:

$$\Delta z_{min} = \frac{N_{pix,fov} \cdot z_{min}}{\left(2f \cdot \tan\left(\frac{\theta_{FOV}}{2}\right) \cdot \frac{1}{y_{pix}}\right) - N_{pix,fov}}$$

Thus, total working distance is:

$$z_{wd} = \Delta z_{min} + z_{min}$$

Given the desired feature size and separation for a calibration algorithm, one can then forward project these metrics to extract instrument feature size and dimension:

$$y_{feat} = \frac{z_{wd} \cdot N_{pix,feat} \cdot y_{pix}}{f}$$

$$y_{pattern} = \frac{z_{wd} \cdot N_{pix,pattern} \cdot y_{pix}}{f}$$

A.2 Rotating Disk Design

The following expressions are used to design the rotating dot calibration pattern. The expressions derived in Appendix A.1 can be used to derive the minimum feature size and dimension of the instrument. From there, the geometry and design specifics of the rotating disk instrument can be derived. These expressions outline this design process. Input design definitions are:

- motor frequency (f_{mot})
- top row frequency (f_2)
- bottom row frequency (f_1)
- pattern square length ($l_{sq} = y_{pattern}$)
- pattern feature size ($d_{dot} = y_{feat}$)

The outside circle on the second layer that is aligned with the top row is denoted as circle 2 while the inner circle is circle 1. To stimulate the camera with frequency $f_{2,1}$, the number of dots for each circle is:

$$N_{dot,(1,2)} = \frac{f_{(1,2)}}{f_{mot}}$$

The circumference of each circle on the second layer can be derived by having knowledge of $N_{dot,(1,2)}$ and the dot diameter, d_{dot}. Assuming each dot is evenly distributed along circumference with spacing equal to dot diameter, the circle circumference can be expressed as:

$$c_{(1,2)} = 2N_{dot,(1,2)}d_{dot}$$

Radii of the circles on the second layer can be found by:

$$r_{(1,2)} = \frac{c_{(1,2)}}{2\pi} = \frac{N_{dot,(1,2)}d_{dot}}{\pi} = \frac{f_{(1,2)}d_{dot}}{\pi f_{mot}}$$

The only remaining parameter to compute is the offset of the center of the circles with respect to the bottom edge of the square pattern on the first layer, c_{cir}. This can be found using Fig. A-1.

From the figure, the radii of circles 1 and 2 can be related to the pattern spacing and offset parameter using the Pythagorean Theorem:

$$r_1{}^2 = c_{cir}{}^2 + \frac{l_{sq}{}^2}{4}$$

$$r_2{}^2 = (l_{sq} - c_{cir})^2 + \frac{l_{sq}{}^2}{4}$$

The offset can be computed using either equation. Choosing the expression for radius 1,

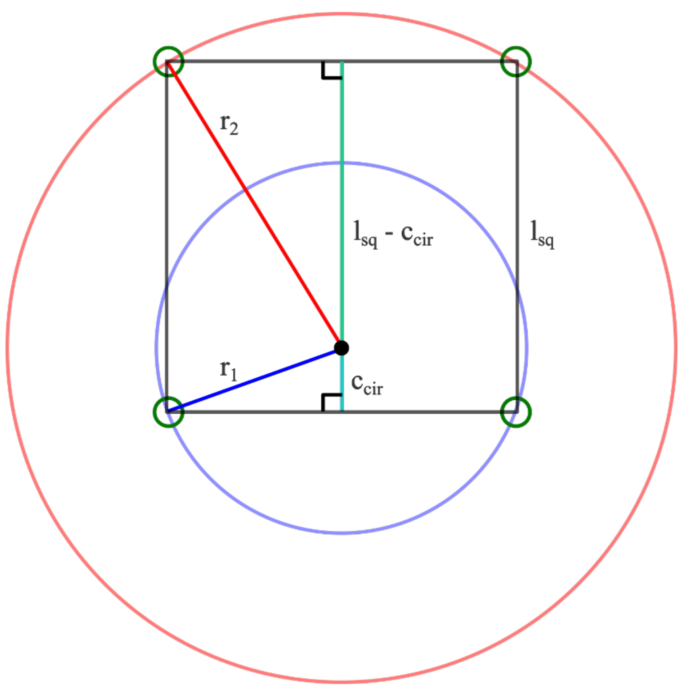

Fig. A-1 Geometric diagram used to derive the parameters for rotating disk design

$$c_{cir} = \sqrt{{r_1}^2 - \frac{{l_{sq}}^2}{4}} = \sqrt{(\frac{f_1 d_{dot}}{f_{mot}})^2 - \frac{{l_{sq}}^2}{4}}$$

If an imaginary result is computed for the offset parameter, the design parameters need to be adjusted. This includes reducing the pattern size or motor speed and increasing the desired frequency or dot size.

List of Symbols, Abbreviations, and Acronyms

2/3/4D	two-/three-/four-dimensional
AEE	absolute extrinsic error
BL	bottom left
BR	bottom right
COMS	complementary metal oxide semiconductor
EBC	event-based camera
FOV	field of view
LED	light-emitting diode
max	maximum
min	minimum
PC	personal computer
RMS	root mean squared
ROT	rotating disk method
RPE	reprojection error
SDK	software development kit
TL	top left
TR	top right
USB	Universal Serial Bus
VGA	video graphics array
WD	working distance

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB
28 (PDF)	DEVCOM ARL FCDD RLA TA G THOMSON T KOTTKE C WOLFE FCDD RLA TD J SENGUPTA A BARD N BRUCHEY M DEVINE R DONEY E ELDRIDGE J FLENIKEN R GUPTA S HALSEY M KEELE D KLEPONIS B KRZEWINSKI B LINNE M DUFFY K MASSER F MURPHY D PETTY C RANDOW S SCHRAML R SPINK J STEWART K STOFFEL N TEE D WEBB M ZELLNER W CLARK