This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Quadrocular, Neuromorphic Stereo Triangulation and Asynchronous Data Fusion for 3D Object Tracking

Jonah P. Sengupta
DEVCOM Army Research Laboratory, Army Research Directorate
2800 Powder Mill Road, Adelphi, MD 20783
jonah.p.sengupta.civ@army.mil

Abstract

Neuromorphic vision sensors (NVS) incorporate pixel-level circuitry that is only sensitive to local gradients in intensity. As a result, NVS' reduce background clutter and increase signal-noise ratio for long-standoff scenarios while providing the high temporal resolution needed for tracking of high-speed objects. This paper presents a hardware apparatus and novel algorithm to explore NVS passive ranging capability. A quadrocular, neuromorphic stereo system was constructed, calibrated, and used to capture temporal contrast signatures from high-velocity targets. Resulting data, computed calibration parameters, and a spatiotemporal correspondence metric were subsequently used to compute a pair-wise triangulation of object tip location. An asynchronous, adaptive Kalman Filter was then designed and deployed to fuse 3D track information from each camera pair and produce a high accuracy estimate of tip location and velocity. Preliminary results show this approach reduces track error over other standard methods by 30-50% while yield target tracks greater than 1MHz equivalent sampling rate under short standoffs.

1. Introduction

Tracking high-speed objects is a foundational cornerstone for many commercial systems that have legacy use in sports [4], astronomy [19], robotics [20], and public safety [9]. Historically, RADAR has been the key sensing paradigm leveraged for these applications. Sensor capability can be scaled to detect objects beyond the horizon or for short-standoff ranging. However, realizing these applications on attritable autonomous systems becomes complicated with this current approach. Due to the requisite active emission, scaling the RADAR transmitter for medium- to long- range tracking of these high-speed objects makes the system unsuitable on platforms which are size, weight, and power (SWaP) constrained. Electro-optical/infrared (EO/IR) phenomenologies can be leveraged in these scenarios, but with trade-offs. For one, the spectral radiant signature of the high-speed target can be exploited for long-standoff detection, but fine object resolution at long-range with simultaneously wide angular field of view requires large format focal plane arrays. As such, scanning the FPAs at the temporal resolution needed for tracking high-speed objects comes with large power and data bandwidth penalties. Secondly, upon approach after resolution of the target, the apparent angular rate of these targets accelerates. Thus, frame rate of the sensor places a hard bound on the temporal resolution of the track.

Neuromorphic vision sensing (NVS) provides a potential technical route to resolve this technological gap in EO/IR ranging for high-speed object tracking [10, 12]. NVS, otherwise commonly known as an event-based sensor (EBS), leverages pixels which only produce data with respect to requisite changes in the visual scene. This in-pixel processing transmits data corresponding to the high-speed target and largely removes the low spatiotemporal frequencies present in visual backgrounds. As such, NVS can devote the requisite temporal and spatial resolution needed for high-speed object tracking applications without sacrificing performance with respect to SWaP constraints.

1.1. Related Work

Since the advent of the technology, the academic community has developed algorithms and processing approaches to deploy NVS for object tracking. These include unsupervised clustering methods applied to pedestrian and commercial surveillance [13], feature-based, discriminative approaches for dynamic object recognition [14], and multi-hypothesis trackers used for celestial targets [3]. Moreover, monocular tracking applications have been expanded to multi-camera or multi-view scenarios. By incorporating calibration techniques or fusing inertial information, depth and 3D scene information can be computed solely from event-based data [5, 6, 23].

1.2. Our Contribution

This paper presents novel hardware and algorithmic approaches to deploying NVS systems for 3D tracking of high-speed objects. It builds upon prior work [17] and [3] by using 2D tracks of high-speed objects to build 3D estimates of tip locations under short-standoff. Also, to the authors knowledge, this work also presents the first demonstration of a quadrocular, NVS system. These unique tracks from the quadrocular system are then fused together with a novel, adaptive, asynchronous Kalman filter algorithm. This and the triangulation steps are presented in Section 2. The algorithm was then verified using the hardware system and experimental approach outlines in Section 3. Preliminary results and comparative analysis of the algorithm is shown in Section 4 before concluding in Section 5.

2. Algorithmic Framework

The quadrocular triangulation and fusion framework can be segmented into four steps:

1. 2D object detection and tracking ($x_i^a(t) \in \mathbb{R}^2$)
2. Spatiotemporal correspondence across tip pairs ($x_i^{(a,b)}(t) \in \mathbb{R}^{(2\times2)}$)
3. Discrete linear transform and linear least-squares to resolve 3D points ($X_i^{AB}(t) \in \mathbb{R}^3$)
4. Feed 3D points into asynchronous Kalman filter with adaptive noise modeling for final 3D estimate ($X_i(t) \in \mathbb{R}^3$)

The correspondence and triangulation steps of the algorithm require knowledge of each camera's intrinsic and extrinsic parameters, formally K_i and $[R_i|t_i]$ where R_i and t_i denote the rotation and translation properties i_{th} respectively. Information about the custom techniques deployed to resolve these parameters are described in Section 3.

After the cameras are geometrically calibrated, the system was then deployed to capture the event-based visual signature of the high-speed targets. NVS data is quite dissimilar to traditional visual data: instead of synchronously-bundled frames, NVS output continuous time, discrete value datum in address-event representation or (AER). In this form, event $\mathbf{e_i}$ is a vector composed of:

$$\mathbf{e_i} = (x_i, y_i, ts_i, p_i) \forall i \in (1, N_{ev}) \tag{1}$$

where x_i, y_i are the x,y address, p_i is polarity, and ts_i is the timestamp (in units of microseconds) of the i_{th} event.

Object tracks and features were subsequently extracted from the resulting event-stream using the Event-based, Multi-camera, Multi-Object Tracker (EB-MCMOT) pipeline [17]. This algorithm first applies a series of spatiotemporal filters and feature extractors to the event data [3]. These steps seek to remove non-correlated data that is a byproduct of noise and pass event data corresponding to salient features, such as corner keypoints. After filtering, the data is further compressed into a series of clusters which bundles events that close to one another in space and time. Object tracks are finally established by grouping these clusters based on another spatiotemporal metric and their individual velocity profiles. The governing assumption is that clusters belonging the the same object are close to each other and move similarly in the same small period of time. The output of the pipeline is a series of event-based object tracks $\mathbf{O_i}$ formally composed as:

$$\mathbf{O_i} = (ts_i, \mathbf{cx_i}, \mathbf{tx_i}, \mathbf{sx_i}, \theta_i, r_i) \forall i \in (1, N_{ev}) \tag{2}$$

where ts_i is the time of last track,$\mathbf{cx_i}, \mathbf{tx_i}, \mathbf{sx_i}$ are the 2D centroid, tip, and tail vector locations, and θ_i, r_i are the objects scalar angle and length respectively. For a ballistic object candidate, tip and tail features are classified based on the object velocity angle and the clusters which correspond to the leading and falling edge.

2.1. Triangulation

After the event stream from each is processed through EB-MCMOT, the resulting object tracks are then processed in parallel during the triangulation step. In the presented system, there are four, synchronized NVS capturing the target. These cameras were then coupled into pairs in order to produce six distinct 3D estimates. As such, the algorithm maintains three parallel threads and four memory elements which stores a history of recent tip tracks from each camera.

As a symbolic example, the first thread receives tracks from camera A and performs correspondence using tracks from B, C, and D. Correspondingly, the second thread consumes tracks from camera B and reads data from memories C and D. Finally, the third thread receives tracks from camera C and performs correspondence with camera D. In general, the number threads can be expressed as:

$$N_{th} = N_{cam} - 1 \tag{3}$$

where thread $N_i \forall i \in (1, N_{th})$ attempts to correspond from $N_{cam} - i - 1$ cameras. Similar to other work [6], correspondence is established with a series of temporal and spatial metrics. First, the two events are evaluated against a temporal window:

$$\delta t_{i,j}^{a,b} = t_i^a - t_j^b \tag{4}$$

$$I_t(a, b, \theta_t) = 1 \Rightarrow \ \delta t_{i,j}^{a,b} > \theta_t \tag{5}$$

If the indicator function $I_t(a, b, \theta_t)$ evaluating the two most recent tracks from cameras A and B is valid, then the spatial test is applied. This involves using the camera pair's Fundamental matrix (derived from the pair's Projection matrices), $F_{a,b}$. Formally, the track from camera A is projected onto an epipolar line $l_{i,j}^b$ in camera B's image plane.

$$l_{i,j}^b = \mathbf{F}_{a,b}\mathbf{x}_i^a \tag{6}$$

The distance from this line to the temporally relevant tip track in camera B is then evaluated by multiplying the homogeneous coordinate with the epipolar line and ensuring the result is below a threshold.

$$I_s(a, b, \theta_s) = 1 \Rightarrow \mathbf{l}_{i,j}^b \mathbf{x}_j^b < \theta_s \tag{7}$$

Thus, if this projection is valid (0 for perfect epipolar correspondence), then a correspondence is then established across cameras A and B for events i and j. This method is a subset of the techniques performed by other works [6].

These two 2D tracks are then used to resolve a 3D estimate of the tip position. The classical triangulation problem from pair wise matches is posed as:

$$\begin{bmatrix} y_i^a \mathbf{p_3^a}^T - \mathbf{p_2^a}^T \\ \mathbf{p_1^a}^T - x_i^a \mathbf{p_3^a}^T \\ y_j^b \mathbf{p_3^b}^T - \mathbf{p_2^b}^T \\ \mathbf{p_1^b}^T - x_i^b \mathbf{p_3^b}^T \end{bmatrix} X_{i,j}^{a,b} = 0$$

where $p_n^{a,b}$ represent the n^{th} row of the projection matrix from camera A, B and $X_{i,j}^{a,b}$ is the desired 3D estimate. Using Singular Value Decomposition, the least squares error of this homogeneous linear system can be minimized and the 3D coordinate can be found. Each thread then records its $N_{th} - i$ 3D estimates in internal buffers which are then consumed by the data fusion algorithm.

2.2. Asynchronous Data Fusion

The data fusion processing block has the objective to reduce intra-track jitter, enhance track accuracy with respect to ground truth and produce a singular 3D estimate of target position which benefits from the temporal resolution of each track. Historically, the Kalman Filter has been the first choice to fuse information from disparate sensing paradigms [22]. Using prior knowledge of noise mechanisms, the approach reduces error from its input measurements by adapting confidence in the measured and predicted state by minimizing residual cost. It is an intuitive choice for this application as well due to the linear, kinematic nature of high-speed targets.

The simple Kalman Filter makes multiple assumptions, including that the transition states are linear and that sensor measurements are synchronous and single-rate. However, in most scenarios (including this problem statement of this paper), one or both are violated. In the latter assumption, it is common that the KF needs to fuse information from sensors that are not only multi-rate (such as low-frequency GPS and high-speed INS), but also are anisochronous. Methods to compensate for this include using individual filters for each sensor and combining using adaptive weighting schemes, downsampling the high-frequency device, or interpolating/predicting data from the low-frequency device using the high-frequency measurements [7, 8, 11, 15].

However, these scenarios are usually typically integrating information from complementary sensors rather than those that produce redundant states. There are those who recommend treating each measurement independently with individual sensor noise models, but that requires individual Kalman gain terms which balloons computation time and memory resources [7]. Instead, the alternative approach was taken to fuse all tracks into a singular state by pseudo-synchronizing the data via temporal advancement and cultivating an adaptive noise model to capture the non-linearity of the NVS and tracker.

2.2.1. Temporal Advancement

Due to the event-based nature of the NVS and tracker, individual tracks from each pair-wise stereo system occurs with respect to the output the EB-MCMOT for the specified cameras. Despite using similar parameters for each camera, the temporal resolution of the triangulation is determined as much by temporal contrast signature, stochastic nature of the analog processing chain in the pixel, and the bias conditions of the camera [16]. As such, event rates and the resulting tracks differ greatly. As such, the temporal advancement step of the algorithm seeks to align each track in time using the current velocity estimate from the Kalman filter. This produces a higher-dimensional measurement, Kalman gain, and sensor noise model. While this compounds computational complexity due to the need to invert a higher dimensional matrix, the increased information promises to reduce measurement noise and the residual error that seeks to be minimized by the gain term.

The temporal advancement step requires three key pieces of data from the sensors and filter: the most recent 3D estimate from each stereo pair, the time of occurrence, and velocity state of the filter. The latter two variables are being stored and read from an internal buffer in the algorithm. New tracks overwrite the memory and trigger the most recent states to be queried. The update times of these recent states are used to determine the temporal difference between the current track and the newest track from the i^{th} pair:

$$\delta t_i = ts_j - ts_i \forall i \in (1, N_{tracks}) \tag{8}$$

such that ts_j is the timestamp from the newest 3D estimate. Once most recent tracks are fetched from the buffer, the velocity vector is appended to construct intermediate state

vector $\hat{z}_k(t)$:

$$\hat{z}_k(t) = \begin{bmatrix} x_0(ts_0) \\ y_0(ts_0) \\ z_0(ts_0) \\ \vdots \\ x_n(ts_n) \\ y_n(ts_n) \\ z_n(ts_n) \\ \hat{v_x}(t) \\ \hat{v_y}(t) \\ \hat{v_z}(t) \end{bmatrix} \in \mathbb{R}^{N_{cam} \cdot 3+3}$$

The temporal deltas computed from the most recent tracks are then used to populate a the temporal advancement matrix $U(t) \in \mathbb{R}^{(N_{cam} \cdot 3) \times (N_{cam} \cdot 3+3)}$:

$$U(t) = \begin{bmatrix} 1 & 0 & 0 & \dots & \dots & \dots & \delta t_0 & 0 & 0 \\ 0 & 1 & 0 & \dots & \dots & \dots & 0 & \delta t_0 & 0 \\ 0 & 0 & 1 & \dots & \dots & \dots & 0 & 0 & \delta t_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & 1 & 0 & 0 & \delta t_n & 0 & 0 \\ \dots & \dots & \dots & 0 & 1 & 0 & 0 & \delta t_n & 0 \\ \dots & \dots & \dots & 0 & 0 & 1 & 0 & 0 & \delta t_n \end{bmatrix}$$

Finally, the velocity and temporal delta then update the most recent tracks and align the sensor states by performing a matrix-vector multiplication:

$$\tilde{z}_k = U(t)\hat{z}_k(t) \tag{9}$$

This high-dimension sensor state is then used in the update step of the Kalman Filter and will later be shown to produce a smoother estimate than the time-multiplexed, independent sensor Kalman Filter.

2.2.2. Adaptive Noise Modeling

In most scenarios, a fixed, stationary noise model can be applied to the sensor inputs. However, NVS' are inherently non-linear, stochastic elements which do not abide by these traditional modeling approaches [21]. Attempts to model noise characteristics of the sensors have been shown to have efficacy for fusion but the application to this redundant tracking space remains to be seen. In this scenario, the noise model needs to encompass the sensor and algorithm characteristics. Moreover, as these systems are applied in real-world scenarios, the characteristics of the NVS shift demonstrably with respect to illumination and stimulii [16]. Thus, an online, time-varying noise-model was chosen to represent the stochastic nature of the measurement. This approach can be deconstructed into five steps:

1. Online linear regressive fit of each time-series track.
2. Using the temporal data and linear model to predict the track
3. Computing the residual error between the actual and predicted value.
4. Online estimation of residual error statistics
5. Given the track index, using the error statistics to populate and update the appropriate elements in the noise model.

Within the algorithm, N_{tracks}-sized buffer of noise parameters are stored. This includes the current linear fit, residual error statistics, and number of points:

$$\sigma_{tracks,i} = (\beta_0, \beta_1, e_y, n) \forall i \in (1, N_{tracks}) \tag{10}$$

Formally, the online regressive fit is computed with the following method. For simplicity, the x-term represents the x-axis of the linear fit, which is time, while y is the coordinate (x,y, or z). Equations 11-13 are computed for the x,y variables, while 14-15 are computed once.

$$\mu_x^t = \mu_x^{t-1} + \frac{(x_t - \mu_x^{t-1})}{n^{t-1}} \tag{11}$$

$$e_x = x - \mu^{t-1} \tag{12}$$

$$\sigma_x^t = \sigma_x^{t-1} + \frac{(x - \mu_x^t)(x - \mu_x^{t-1})}{n^{t-1}} - \frac{\sigma^{t-1}}{n^{t-1}} \tag{13}$$

$$c_x^t = c_x^{t-1}\frac{n^{t-1}}{n^t} + \frac{e_x \cdot e_y}{n^{t-1}} \tag{14}$$

$$\beta_1 = \frac{c_x^t}{\sigma_x^t},\ \beta_0 = \mu_y^t - \beta_1 \mu_x^t \tag{15}$$

Computing the predicted track from the temporal value involves using the model coefficients and the current timestamp (x-term):

$$\tilde{y} = \beta_1 x + \beta_0 \tag{16}$$

Thus, the residual, signed error is computed:

$$\tilde{e_y} = y - \tilde{y} \tag{17}$$

Through the use of Welford's online algorithm, the mean and standard deviation of this residual error can be found:

$$\mu_e^t = \mu_e^{t-1} + \frac{\tilde{e_y} - \mu_e^{t-1}}{n} \tag{18}$$

$$\sigma_e^t = \sigma_e^{t-1} + \frac{(\tilde{e_y} - \mu_e^t)}{n} - \frac{\sigma_e^{t-1}}{n} \tag{19}$$

The standard deviation of this error represents how smooth the track is, while the mean outlines if the track truly abides by the linear model. Thus, the error deviation can be used then to populate the sensor noise metric for each track.

$$\sigma_{x,n} = \alpha\sqrt{\sigma_{e,x}} \tag{20}$$

$$\sigma_{y,n} = \alpha\sqrt{\sigma_{e,y}} \tag{21}$$

$$\alpha = \min(\frac{\alpha_0}{N_n}, \alpha_{min}) \tag{22}$$

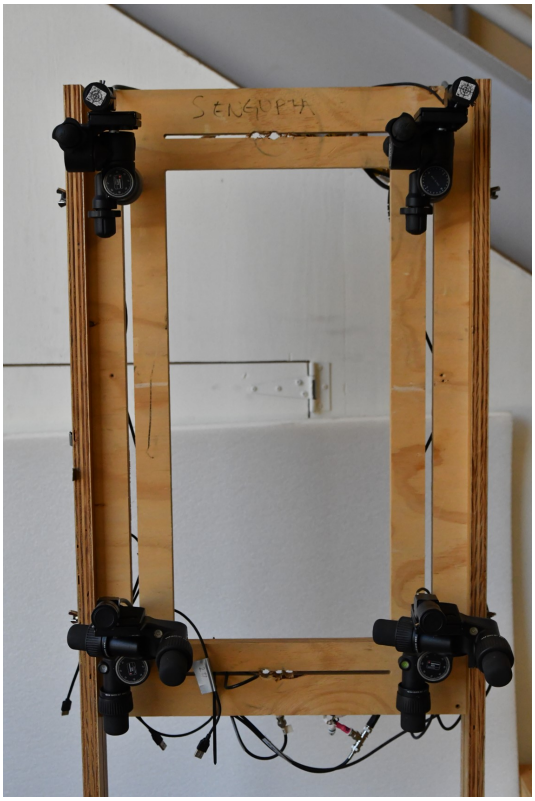


Figure 1. Quadrocular event-based stereo system: four SilkyEvCamVGA event-based cameras (S/N - 291-288 CCW from TL), 0.48m and 0.76m horizontal and vertical baseline respectively

Moreover, the deviation is also scaled with respect to the number of samples received from the specific track. Intuitively, more confidence into the error estimate is placed as more measurements are received. Thus, a convergence will take place over the course of each run. As a result, tracks with less data will be noisier and the Kalman gain will reduce the impact of those measurements.

3. Experimental Setup

Figure 1 depicts the experimental setup used to capture the high-speed object signatures. An event-based quadrocular, stereo system (EBQSS) was constructed by integrating four commercial-off-the-shelf (COTS) NVS parts. Specifically, the SilkyEvCamVGA EBC was chosen as it provides a compromise between resolution and response time [2, 16].

Camera serial numbers were used to delineate between camera tracks. Specifically, the S/N for the EBQSS spanned 288-291 (moving from bottom right clockwise). Each camera captures, decodes, and timestamps the data from the NVS. Timestamps from each camera were synchronized within the EBQSS using a clock output from the master camera (S/N - 291). Each EBC was also equipped with a varifocal length lens that allowed whose focal length can be adjusted from 4-12mm. This allowed for flexibility during experimentation to investigate the inherent tradeoff between angular and instantaneous fields-of-view. Data is ultimately transported from each EBC to a host PC via USB 3.1 interface. Custom software was designed on-top of the Prophesee Metavision SDK to configure, start, and record data from all cameras in parallel.

This same software was also expanded to use the data from the EBQSS for the purpose of geometric calibration. A custom hardware fiducial was designed to provide the necessary temporally modulated, precisely arranged spatial stimulus. This consisted of an array of blinking LEDs whose frequency and brightness was controlled by an onboard microcontroller. The software was then used to synchronously detect and record the transmitted patterns in each camera FOV. After a user-specified amount of patterns were recorded, a post-processing calibration routine was run to first compute the intrinsic camera parameters for each EBC before extracting the extrinsic camera parameters for each stereo pair in the EBQSS. These extrinsic parameters were optimized for each pair by iteratively subsampling the detected keypoint pairs and minimizing the re-projection error. Results of this procedure are shown in Figure 2. More information about the calibration procedure and algorithms can be found in the supplemental material [18].

The left panel shows the captured pattern poses with respect to each stereo pair (color coded by camera). The right panel shows the ground truth and computed camera poses. Overall, the event-based detection methodology paired with the iterative calibration procedure yielded a pose error $<$10mm.

High-speed object surrogates were used for rapid prototyping and large dataset cultivation. The surrogate system chosen for this endeavor was an airbow which leverages the rapid release of compressed air to launch arrows at fast speeds. Specifically, when the system is pumped to 3000 psi, arrow muzzle velocities exceed 190m/s [1]. The airbow was situated less than 15m away from the test stand shown in Figure 1. Thus, the muzzle velocity was verified via high-speed videography to only decay by less than 10% when passing through the test fixture.

4. Results

The test fixture described in Section 3 was used to capture multiple airbow signatures at multiple camera bias points and stereo system combinations. These included configuring all the NVS at low and high sensitivity settings with multiple binocular and trinocular pairings before enabling all cameras simultaneously. Figure 3 shows the output of the EB-MCMOT following the capture of the target with a fully active EBQSS. The left panel represents frames from each NVS by temporally integrating the events over a 10ms. Blue and red squares correspond to inactive and active clus-

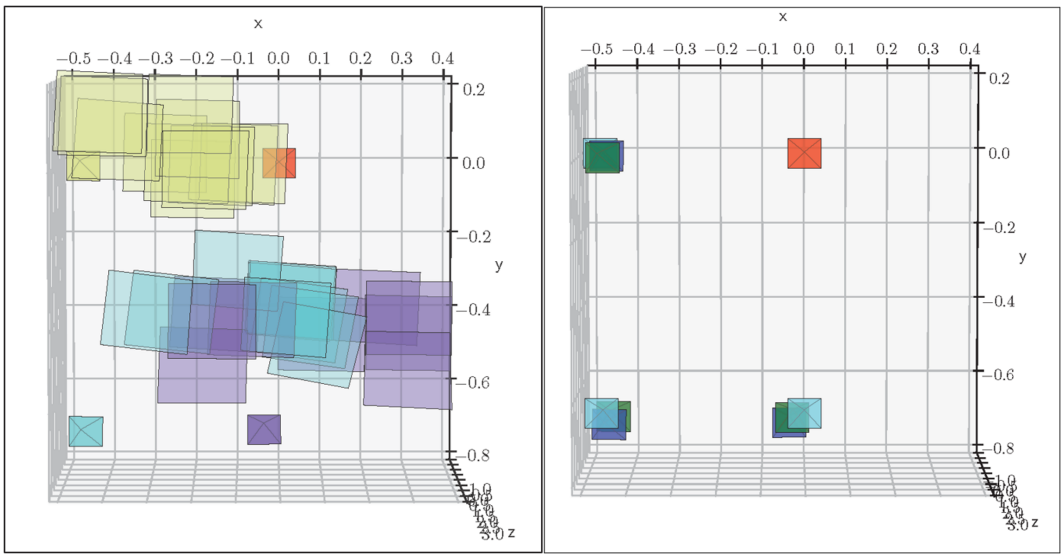


Figure 2. Results of the calibration step. Left - computed sensor positions from calibration algorithm with patterns for each pair overlaid and color-coded with respect to the camera pair. Right - computed extrinsics (green/blue) plotted against ground truth (cyan). More information on the methodology can be found elsewhere (citation)

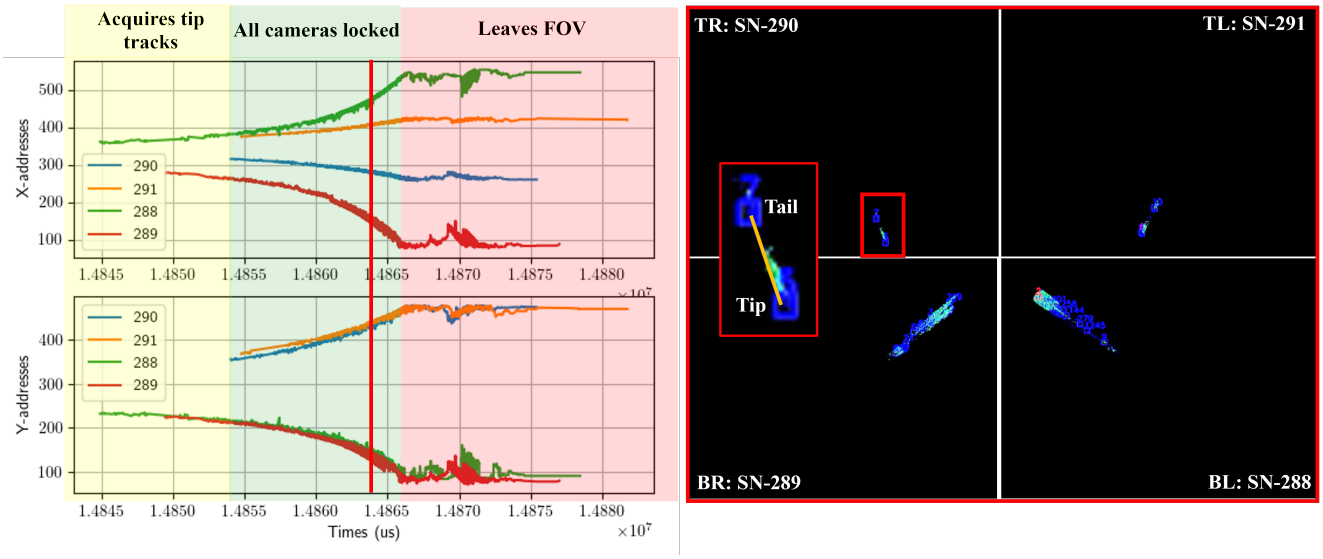


Figure 3. Tracks of high-speed object as it approaches the platform. **Left panel -** Time series of each camera track over time. Serial numbers and positions (from platform perspective) are inset. Different phases of detection and tracking are labeled. **Right panel -** 1ms integrated events and tracks from EB-MCMOT at the sample denoted by the red line in the left panel. Green pixels correspond to events, blue and red boxes correspond to active and inactive clusters respectively. An example object from the TR camera with labeled tip and tail features is inset.

ters of event keypoints respectively. The velocity of the associated object determines the directionality of the target which is leveraged by EB-MCMOT to label tip and tail locations. The X- and Y- coordinate of the tip tracks over time are shown in the right panel. To note, a linear velocity towards the the camera translates to the apparent acceleration shown in the trajectories due to the projective relationship with depth.

These four tracks were then sampled pair-wise to produce six gray tracks shown in Figure 4. The z-coordinate of each track is tightly grouped and represents the target with similar velocities ranging from 131 to 161 m/s. However, there are discrepancies with respect to the x- and y-coordinate the of the track. To visualize this, a simple

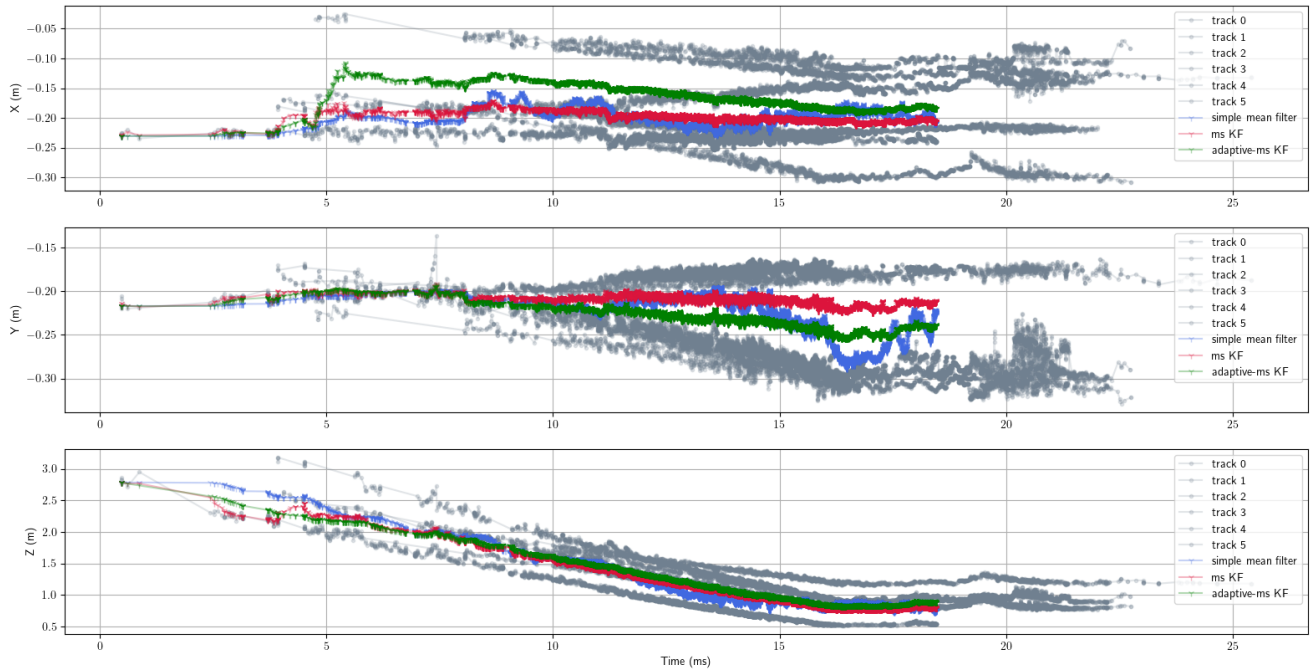


Figure 4. Six tracks (gray) plotted against the three fusion methods: simple mean, the multi-state Kalman filter, and adaptive, multi-state Kalman filter. The X,Y,Z coordinates are plotted against the track time on the first, second, and third axes respectively.

Data Fusion Filter	**RMSE - X(m.)**	**RMSE - Y(m.)**	**RMSE - Z(m.)**
Simple Mean	0.012	0.018	0.127
Single-state KF	0.021	0.021	0.133
Temporally-averaged KF	0.015	0.014	0.092
Multi-state KF	**0.004**	**0.003**	**0.069**
Adaptive, multi-state KF	0.009	0.004	0.071

Table 1. Root mean squared error (RMSE) of each fusion method. Error was averaged over the same temporal interval for each fusion method and each coordinate. The multi-state fusion algorithms had the superior smoothness metric.

ground truth was devised using the trigger timing and positioning of the airbow. When overlaid, there is a large deviation present in the x- and y- position for some tracks. To note, the ground truth was estimated via linear projection and did not account for droop in the target over its trajectory. Hence, the y-tracks could be valid but can only be confirmed upon further testing with high-speed videography.

Three simple fusion methods were applied to the six, redundant tracks in order to provide baseline approaches. The first is a simple, alpha-beta (mean) filter which uses a fixed update constant and treats each of the six tracks as an independent update.

$$z^t = (1-\alpha)\cdot z^{t-1} + \alpha \cdot x_i(t_i)^t \tag{23}$$

where $x_i(t_i)$ is the i^{th} tip track with timestamp t_i and t is updated upon each tip estimate that is timestamped at $t_i = t$ given filter rate constant, α.

The second baseline approach is deploying a linear Kalman Filter using one sensor noise matrix and a low, 3-dimensional measurement vector. In this scenario, only the most recent track was used to update the filter state. The final baseline method was a simple method was devised to pre-condition the signals prior to the single-state Kalman Filter. This involved applying a simple box filter to the six tracks in order to temporally average and compress the signals into one track.

$$z_t = \frac{1}{N_{tracks}} \sum_{i=0}^{N_{tracks}} \frac{1}{\delta t} \int_t^{t+\delta t} x_i(\tau) d\tau \tag{24}$$

where δt is the size of the temporal window used to integrate the tracks.

The temporal advancement procedure described before was applied to yield a high-dimensional measurement state for the Kalman Filter. In these scenarios, shown by the red

curve in Figure 4, the transition state of the KF imposes a constant velocity model to the target. In combination with the high-dimension measurement state which allows for high-frequency optimization of the residual error, the resulting state estimate exhibits low jitter and is approximately linear.

Finally, the online estimation of sensor noise method was applied on top of the temporal advancement. The resulting state estimate is shown by the green curve in Figure 4. Sensor noise for each track was initialized to a naive diagonal matrix. Afterwards, the online linear regression was applied to approximate the residual error and its respective statistics. As highlighted prior, the standard deviation was then used to re-populate the sensor noise matrix for the specific track on the fly. In essence, the adaptive noise matrix represents the linearity of the track at that point in time and is leveraged by the Kalman gain to reduce update weights for those that deviate from this linearity imposed by the model.

Once again, the resulting track estimate from the adaptive methodology exhibits reasonable linearity in all three dimensions. However, this fit to the linear model of the target can be quantified and used to compare all fusion methods. Without presence of accurate, ground truth, a global error metric cannot be leveraged. Alternatively, each track estimate could be evaluated based on its linear nature. To do so, a first-order, linear model was approximated for the state estimate. Similar to equations 16 and 17, the error is found by applying the approximated linear model and find the track residual error. The root-mean-squared error (RMSE) of this residual for the entire track is then used as the metric to compare fusion methods. RMSE for each fusion method is shown in 1.

Overall, the multi-state KF which used temporal advancement performed the best with respect to intra-track RMSE. The method adhered to a linear model with less than 5mm of deviation in the x- and y-coordinate and 7cm in the z-dimension. This latter metric is around a 100% improvement over the simple mean and 3-dimension Kalman Filter methods. Encouragingly, the performance does not decay considerably when the adaptive noise methodology is applied on top of temporal advancement. In fact, the adaptive, multi-state KF is the second-most accurate with respect to intra-track RMSE and is only 2mm less accurate in the Z-direction. Thus, learning the sensor noise model without *a-priori* assumptions on the measurement has comparable performance and promises more robust application than the former.

Another key point to extract from the fusion methodology is the resulting temporal resolution of the track estimate. Neuromorphic vision sensors are promising due to their optimized bandwidth and high-temporal resolution devoted to target signatures. Individually, it has been seen that tracks can be produced with more than 10kHz sampling rates when high-speed targets are at short standoff. When fusing tracks from many of these cameras, this benefit compounds and produces an event-based track with temporal resolution less than 10μs with equivalent sampling rate of 100kHz. In fact, the fusion methodologies shown yield an average sample rate of 1.25MHz (0.79μs) at less than 3m away. This tracking performance and temporal resolution rivals high-performance, high-speed video at a fraction of the size, weight, or power cost.

5. Conclusions and Future Steps

This paper presented a novel algorithmic approach to estimate 3D object position using event-based data from a quadrocular stereo system. The algorithm first ingests temporally synchronized tracks from the four cameras. Using spatiotemporal correspondence metrics and the calibrated parameters, six preliminary 3D tracks are produced for the target tip. These tracks are then fused using the backend Kalman Filter which utilizes two new approaches: temporal advancement and online linear regression. These two approaches showed to reduce intra-track RMSE while yielding a sampling rate that exceeds 1MHz when the target is less than 3m away from the test fixture.

There are multiple experimental and quantitative next steps to prove the fidelity of this approach. First, a new test will be executed that will include high-speed video to produce ground truth object position over time. This will provide a means to evaluate the accuracy of the algorithm and compare against other approaches. More over, the fusion approach can be compared against a triangulation method which expands the over-defined linear system shown in Section 2.1 to handle all four 2D estimate simultaneously. Such a study would showcase the potential benefit of using the fusion method directly.

Overall, the presented hardware and software approach presents a framework to first compute 3D states from NVS data and fuse data from many sensors to yield a position estimate with superior accuracy and temporal resolution. With the described future steps, this approach can be used to yield a low power high-performance tracking system that can leverage data from distributed platforms.

References

[1] Benjamin airbow - benjamin airbow 450 quiver. https://www.benjaminairguns.com/benjamin-airbow. Accessed: 2024-08-30. 5

[2] Silkyevcam-vga description. https://centuryarks.com/en/silkyevcam-vga/. Accessed: 2024-08-30. 5

[3] Saeed Afshar, Andrew Peter Nicholson, Andre Van Schaik, and Gregory Cohen. Event-based object detection and tracking for space situational awareness. *IEEE Sensors Journal*, 20(24):15117–15132, 2020. 1, 2

[4] Brian Armstrong and Brad S Holeman. Target tracking with a network of doppler radars. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):33–48, 1998. 1

[5] Luis A Camuñas-Mesa, Teresa Serrano-Gotarredona, Sio-Hoi Ieng, Ryad Benosman, and Bernabé Linares-Barranco. Event-driven stereo visual tracking algorithm to solve object occlusion. *IEEE transactions on neural networks and learning systems*, 29(9):4223–4237, 2017. 1

[6] João Carneiro, Sio-Hoi Ieng, Christoph Posch, and Ryad Benosman. Event-based 3d reconstruction from neuromorphic retinas. *Neural Networks*, 45:27–38, 2013. 1, 2, 3

[7] Francois Caron, Emmanuel Duflos, Denis Pomorski, and Philippe Vanheeghe. Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects. *Information fusion*, 7(2):221–230, 2006. 3

[8] Shimin Feng and Roderick Murray-Smith. Fusing kinect sensor and inertial sensors with multi-rate kalman filter. In *IET Conference on Data Fusion & Target Tracking 2014: Algorithms and Applications (DF&TT 2014)*, pages 1–8. IET, 2014. 3

[9] P David Fisher. Improving on police radar. *IEEE Spectrum*, 29(7):38–43, 1992. 1

[10] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1

[11] Christopher Hide, Terry Moore, and Martin Smith. Adaptive kalman filtering algorithms for integrating gps and low cost ins. In *Plans 2004. Position location and navigation symposium (ieee cat. no. 04ch37556)*, pages 227–233. IEEE, 2004. 3

[12] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128x128 120 db 15us latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43 (2):566–576, 2008. 1

[13] Martin Litzenberger, Christoph Posch, D Bauer, Ahmed Nabil Belbachir, P Schon, B Kohn, and H Garn. Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, pages 173–178. IEEE, 2006. 1

[14] Bharath Ramesh, Shihao Zhang, Hong Yang, Andres Ussa, Matthew Ong, Garrick Orchard, and Cheng Xiang. e-tld: Event-based framework for dynamic object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3996–4006, 2020. 1

[15] JZ Sasiadek, Q Wang, and MB Zeremba. Fuzzy adaptive kalman filtering for ins/gps data fusion. In *Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No. 00CH37147)*, pages 181–186. IEEE, 2000. 3

[16] Jonah P Sengupta and Stephen D Burks. Demystifying event-based camera latency: sensor speed dependence on pixel biasing, light, and spatial activity. In *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXXV*, pages 273–291. SPIE, 2024. 3, 4, 5

[17] Jonah P Sengupta, Martin Villemur, and Andreas G Andreou. Efficient, event-driven feature extraction and unsupervised object tracking for embedded applications. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2021. 2

[18] Jonah P Sengupta, Julian D Fleniken, Ethan K Eldridge, and Matthew R Devine. Scalable hardware and software procedures for n-ocular event-based stereo system calibration. *ARL Technical Note*, (1190):1–22, 2024. 5

[19] Melvin L Stone and Gerald P Banner. Radars for the detection and tracking of ballistic missiles, satellites, and planets. *Lincoln laboratory journal*, 12(2):217–244, 2000. 1

[20] Shigeki Sugimoto, Hayato Tateda, Hidekazu Takahashi, and Masatoshi Okutomi. Obstacle detection using millimeter-wave radar and its visualization on image sequence. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pages 342–345. IEEE, 2004. 1

[21] Ziwei Wang, Yonhon Ng, Cedric Scheerlinck, and Robert Mahony. An asynchronous kalman filter for hybrid event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 448–457, 2021. 4

[22] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 3

[23] Alex Zihao Zhu, Yibo Chen, and Kostas Daniilidis. Real-time time synchronized event-based stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 433–447, 2018. 1