AAS 24-197

EVENT DATA PROCESSING AND FILTERING FOR FAR-FIELD SPACECRAFT RENDEZVOUS

Sofia G. Catalan[*], Brandon A. Jones[†], and Michael A. Shoemaker[‡]

While the flight heritage of frame-based cameras results in their continued use for satellite missions, event-based cameras have promising advantages for on-orbit use. With improvements in dynamic range capabilities and high return rates, event cameras can continue to provide data in lighting conditions where traditional cameras struggle. This paper uses simulated event data and range measurements to estimate relative spacecraft position and velocity in far-field rendezvous scenarios. The integrated software pipeline described in this work compares the Extended Kalman Filter, the Extended Kalman Probability Hypothesis Density Filter, and the Extended Target Probability Hypothesis Density Filter with single and multiple target tracking cases to estimate translational states between the observing camera and target spacecraft.

INTRODUCTION

Computationally fast and accurate state estimation methods are necessary for spacecraft operations that employ autonomous data acquisition and processing given on-board sensors. With potential delays caused by the ever-increasing demand for communications resources, decreasing the reliance on data transfers to ground operators will be required for future spacecraft missions such that they include more autonomy in their operations. By enabling satellites to perform actions, responding autonomously to what the on-board sensors observe, the latency between critical mission events and the corresponding actions that the spacecraft must take can be significantly reduced. Developing methods to support spacecraft autonomy can be done through simulation-based testing of sensor components prior to the costly hardware in the loop tests. This paper presents how event camera simulation tools are used with state estimation methods to emulate an on-board processing capability in support of Guidance, Navigation, and Control (GN&C).

In the orbital environment, providing accurate state estimates for rendezvous and proximity operations is critical to ensure flight safety. For satellites that have docking and capture operations as part of their mission, a set of on-board sensors can include Light Detection and Ranging (LiDAR), imaging cameras, laser rangefinders, Global Positioning Systems (GPS) and Inertial Measurement Unit (IMU) sensors, etc. to inform translational and rotational states.[1,2] Event sensors, also referred to as neuromorphic cameras from their biological inspiration, can supplement the use of these other sensors, providing an additional optical-based data source in support of state

[*] Graduate Research Assistant, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 2617 Wichita Street, Austin, TX 78712

[†] Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 2617 Wichita Street, Austin, TX 78712

[‡] Aerospace Engineer, Navigation and Mission Design Branch (Code 595), NASA Goddard Space Flight Center, Greenbelt, MD 20771

estimation efforts. Their applications toward ground robots, drones, and autonomous cars have been explored previously.[3–5] The use of event cameras in spaceflight applications is an emerging area of research, ranging from ground-based satellite tracking,[6] in-orbit characterization of lightning strikes,[7] simulation-based object detection,[8] and crater-based lunar navigation.[9] Previous work[10] integrated software tools to simulate event measurements for a relative spacecraft scenario. In the developed tools, a user can specify models for the camera, lighting conditions, and motion (both relative translation and rotation), to then provide event measurements for filters.

Continuing from previous work, this paper addresses challenges in processing event measurements as applied to far-field rendezvous applications. While the high data rate outputs of event sensors are beneficial for filtering, the computational constraints of processing this data in real time can pose a challenge. To handle the data load of a simulated event camera, measurement preprocessing steps are implemented and integrated into the estimation workflow to reduce the sequential processing of individual events. Instead, generated event sets can be used directly in an extended-target based filter (e.g. the Extended Target Probability Hypothesis Density (ET-PHD) filter[11]) or further reduction can be done to process an averaged measurement centroid using point object-based filters (e.x. the Extended Kalman Filter (EKF) and the Probability Hypothesis Density (PHD) filter[12]).

Event Camera Background

Event sensors operate differently at a hardware level from traditional, frame-output cameras that provide a snapshot of a full, representative image of the current field of view (FOV) at a given time. Frame-based cameras store the brightness of each channel, typically red, green, and blue, at a common timestamp to record a single image. Event sensors, however, operate with independently acting pixels such that any observed change in brightness for each pixel is recorded directly. These event detections allow the camera to observe the environment with an increased sensitivity to contrast, since the sensor array is no longer limited to the dynamic range constraint of a typical frame-based camera.[13]

The ability to observe very bright and shadow-obscured objects in the same FOV is of interest to on-orbit relative navigation because of the expected variations in lighting conditions in space. Difficulties in frame-based observations can be caused by the expected reflections of common materials used in spacecraft, oversaturation and blooming caused by the sun or other light sources, and self-shadowing due to large components like solar panels. Since frame-based cameras have dynamic range constraints that may result in observability issues, this work aims to develop methods for event cameras to support navigation in these lighting conditions.

There are several commercially available event cameras that provide different types of data outputs based on their respective hardware and software implementations.[*,†] Some event sensors are packaged together with other sensors in a single camera unit, providing events along with image frames and IMU information, while other cameras provide only the detected events. While the outputs and metadata of the different cameras can vary, the common event output they provide is stored as

$$\mathbf{z}_k = [u, v, p]^T \tag{1}$$

for a single event that occurs at time t_k, located at the pixel coordinate (u, v), and a polarity change

[*] Prophesee EVK4 camera specifications: https://www.prophesee.ai/event-camera-evk4/

[†] iniVation Dynamic Vision Sensors: https://inivation.com/technology/

p. Polarity in this context refers to a positive (brightness increase) or a negative (brightness decrease) change for a particular pixel. The software controlling the sensitivity of the individual pixels allows for a user-configurable input for the brightness threshold required for an activation for each pixel.

Figure 1 compares the outputs of frame and event cameras given a rotating target simulation, where the time axis shows the data rates for the respective sensors. In this simplified case, the event data stream effectively tracks the black dot's motion, whereas the lower frame rate output does not capture any information in the intermediary times between frames. However, the event dataset does not include the redundant information of the external parts of the target. For a static camera setup (as in this example), only producing measurements for moving targets may be a significant benefit of the event sensor since any non-moving components or the environment will not produce measurements.

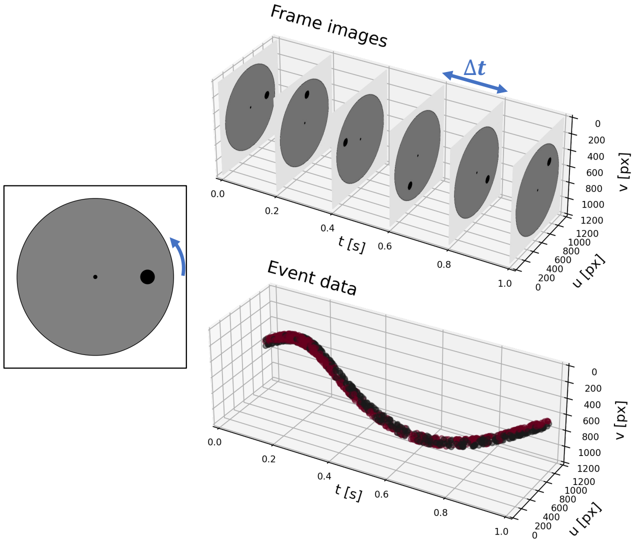


Figure 1. Frame and event camera outputs.[10]

SIMULATION ENVIRONMENT

This work relies on event camera simulators that use sequential image sets to emulate the response of a real sensor, with configurable sensor size, clutter rates, and other parameters that can be based on a real camera. This frame-to-event process enables the use of widely available image/video datasets used heavily by the machine learning community in the development of event-based processing work, leveraged primarily by the autonomous vehicle community.[5] However, for modeling on-orbit conditions, having the ability to generate rendered frame images enables testing different spacecraft models, trajectories, material properties, and lighting conditions to see how an event sensor would respond. A simulation-based approach especially helps to model conditions that cannot be readily replicated in a laboratory environment, for example, the relative orbital dynamics between two satellites in space. While still limited by the fidelity of the simulator, the goal of this research is to develop data processing methods using simulation outputs that can be later applied to outputs obtained by a real event camera in the lab.

To generate event measurements for use in the filtering methods tested in this work, a software workflow integrates several tools to provide replicable outputs with user-configurable fidelity:

- true trajectory generation using the relative orbital dynamics model for each spacecraft,
- image renders from Blender[14] that model the spacecraft location based on each trajectory and user-specified object 3D models, attitude, material properties, and lighting conditions, and
- simulated event measurements for a sensor configured to use a real camera sensor size, sensor clutter models, and output data rates.

Further information about the configurable inputs in this current software workflow can be found in previous work,[10] however, for consistency in this paper, the camera parameters emulate the Prophesee EVK4 through the ICNS Event Based Camera Simulator (IEBCS).[15]

STATE REPRESENTATION AND DYNAMICS MODEL

This work uses Hill-Clohessy-Wiltshire (HCW) dynamics to model the relative motion between an observing camera (denoted as the "chief") and a spacecraft model (the "deputy"). Estimating the deputy's motion $\mathbf{x}(t) = [x, y, z, \dot{x}, \dot{y}, \dot{x}]^T = [\mathbf{r}^T, \mathbf{v}^T]^T$ is the primary goal of the employed estimation filters in this work. With this dynamics model, the chief is assumed to be in a circular orbit with respect to the primary body, and for this paper, no additional forces are modeled (maneuvers, perturbations, and unmodeled random errors in acceleration). To simplify the transformations between the deputy's relative coordinates and the camera frame, the camera is pointed such that the boresight is aligned to the $+y$ axis, the vertical matches the $+x$ axis, and the horizontal corresponds to the $+z$ axis of the HCW frame.

The relationships between the position and velocity coordinates result in useful constraints that can be leveraged to limit the ballistic trajectory of the deputy to a bounded ellipsoid,[16] as shown in Fig. 2. The deputy's initial conditions can be set such that it remains in the camera's FOV for an extended period of time, allowing for testing the implemented estimation filters without extended gaps in measurements.

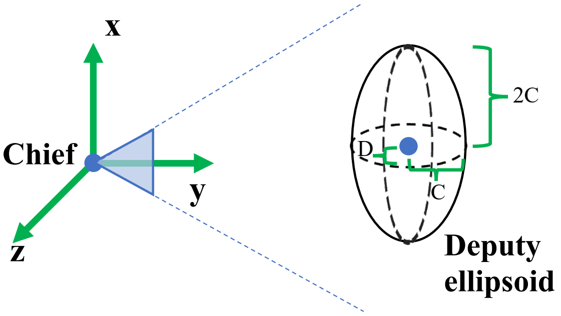


Figure 2. Geometry of bounded deputy trajectory

The geometry of the ellipse is related to the deputy's initial conditions through:

$$y_c = y_0 - \frac{2\dot{x}_0}{\omega}, \tag{2}$$

$$C = \sqrt{(3x_0 + \frac{2\dot{y}_0}{\omega})^2 + (\frac{\dot{x}_0}{\omega})^2}, \tag{3}$$

$$D = \sqrt{\frac{\dot{z}_0}{\omega}^2 + z_0^2}, \tag{4}$$

where ω is the mean motion of chief satellite's orbit around the primary body, y_c is the y coordinate of the ellipse center, with dimensions C and D. Through this geometry, the constraints on the initial conditions are

$$y_0 = y_c + 2\dot{x}_0 > 0, \tag{5}$$

$$z_0 = 0, \tag{6}$$

$$\dot{y}_0 = 2\omega x_0, \tag{7}$$

$$\dot{z}_0 \neq 0, \tag{8}$$

which reduces the free variables to x_0, y_c, and $\dot{z}_0$. These user inputs are then set according to the modeled camera FOV to maintain observability of the deputy throughout the desired simulated trajectory.

MEASUREMENT MODELS

The (u, v) event pixel coordinates are converted to the horizontal and vertical bearing angles (α, β), respectively, based on the pinhole camera model:[17]

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \operatorname{atan}\left(\left(u - \frac{N_u}{2}\right), f\right) \\ \operatorname{atan}\left(\left(-v + \frac{N_v}{2}\right), f\right) \end{bmatrix}, \tag{9}$$

where N_u and N_v are the horizontal and vertical sensor sizes in pixels, and f is the focal length in pixels. The sign change in the vertical angle accounts for the flip between a positive pixel defined downwards in the camera frame while the x axis is positive upwards. Assume that the measurement has additive random noise with $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_2)$, resulting in $\mathbf{z}_{bearing} = [\alpha, \beta]^T + \epsilon$. The current work does not use the polarity of the events, but future extensions can consider its inclusion with an appropriate measurement model.

Relating the deputy state $\mathbf{x}$ to the bearing measurements is done via the measurement model:

$$\mathbf{h}(\mathbf{x})_{bearing} = \begin{bmatrix} \operatorname{atan2}(z, y) \\ \operatorname{atan2}(x, y) \end{bmatrix}. \tag{10}$$

For these conversions, the operator $\operatorname{atan2}(\cdot)$ is used to return the correct quadrant information. From the bearing angles to the state, the measurement Jacobian is computed with

$$\mathbf{H}(\mathbf{x})_{bearing} = \begin{bmatrix} 0 & \frac{-z}{y^2+z^2} & \frac{y}{y^2+z^2} & 0 & 0 & 0 \\ \frac{y}{x^2+y^2} & \frac{-x}{x^2+y^2} & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11}$$

Due to the relative orbital dynamics, the bearing angle measurements alone do not provide enough information to converge on the modeled true trajectory because these measurements are not unique, leading to potential maps to other possible trajectory solutions. Previous work enables bearing-only relative estimation, however, this requires constraining the types of trajectories that can be modeled to include specific maneuvers for continued, full observability of the state.[18] To enable testing more variations in trajectories, and in the future, to apply this work towards a wider span of possible proximity operations missions, additional relative range information is required. The inclusion of range allows for testing trajectories where the modeled spacecraft do not have maneuvering capabilities, while also testing the required precision and frequency of range data needed to provide accurate state solutions.

To include range information in this paper's simulations, consider the use of an on-board laser rangefinder to provide a relative range measurement with additive error $\epsilon \sim \mathcal{N}(0, \sigma^2)$ such that $z_{range} = \rho + \epsilon$ and

$$h(\mathbf{x})_{range} = \rho = \sqrt{x^2 + y^2 + z^2}, \tag{12}$$

with the corresponding Jacobian

$$\mathbf{H}(\mathbf{x})_{range} = \left[\begin{array}{cccccc} x/\rho & y/\rho & z/\rho & 0 & 0 & 0 \end{array}\right]. \tag{13}$$

For simplicity, the range sensor is modeled as collocated with the event camera such that their coordinate origins are equivalent. Additionally, assume that the range data is provided for the center of the chaser spacecraft but at a different, lower frequency of returns than the bearing angle measurements. The two sensors will provide measurements at different rates and corresponding time steps since they are independently acting, emulated software elements on the chief spacecraft.

ESTIMATION METHODS FOR ON-ORBIT APPLICATIONS

The types of estimation filters applicable to specific problems depend on what kinds of dynamics, measurements, and associated error distributions are modeled. To build up to the capability to process event sensor outputs, three estimators are compared in this section. Point-source measurements are handled well by the EKF and the PHD filter. However, the PHD filter can account for clutter returns and enables tracking multiple objects, which the EKF cannot handle. Clutter, in this context, are spurious measurements produced by the sensor hardware that are not representative of the tracked object. Building upon the PHD further, the ET-PHD can account for having multiple measurements resulting from each observable target in the FOV.

Extended Kalman Filter (EKF)

The EKF can provide state estimates of a single object when provided measurements, the nonlinear measurement models, and corresponding Jacobians presented previously. The EKF serves as a baseline filter for further testing, particularly for the idealized case where there are no clutter measurements and only one measurement is returned for an object. A basic addition of residual-based outlier rejection can be implemented for the EKF to prevent ingesting bad measurements, for example, clutter returns. However, the EKF's limitation to single target tracking does not allow for its use in tracking scenarios when there are possibly multiple targets that can generate multiple measurements per object. Particularly for ranges where the extent of the object becomes observable, the EKF would require a preprocessing step to group observations and compute a centroid prior to performing measurement updates.

Probability Hypothesis Density (PHD) Filter

Handling measurements that either map to real targets or spurious, clutter returns can be characterized by Random Finite Set (RFS)-based filtering methods like the Gaussian Mixture (GM)-PHD filter from Vo and Ma.[12] Here, the Extended Kalman-GM-PHD[12] is implemented, however, for shorthand, this filter will be referred to as the PHD in this paper.

The PHD implementation extends previous work[19,20] to develop a numerically tractable multi-target tracking capability for cluttered measurements. The GM model is used to represent the RFS intensity functions of the form:

$$v_k(\boldsymbol{x}_k) = \sum_{j=1}^{J_k} w_k^{(j)} \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{m}_k^{(j)}, \boldsymbol{P}_k^{(j)}\right),$$

where the index k represents time, j denotes the GM component index, and w, $\boldsymbol{m}$, and $\boldsymbol{P}$ represents the scalar weight, vector mean, and matrix covariance of the GM component, respectively. Clutter can be modeled as a Poisson RFS, defined by the expected number of clutter measurements and assumes a uniform density over the FOV of the sensor.

By representing the multitarget state as a GM, the intensity functions corresponding to the prior and posterior are evaluated to include the probability of detection and survival, while also considering the included clutter models. Through the weight updates given measurements, this filter is able to maintain computational tractability because it does not explicitly include a costly data association step to match measurements to states. Instead of attempting to label which measurement corresponds to which target at each time step, the PHD updates each of the GM components using every available measurement without labeling. Additionally, Vo and Ma also provide a GM reduction method via a merge, prune, and truncate procedure that pares down the number of GM components based on their weights to reduce the computational burden of using GMs.

Intensity functions representing object birth and spawn can also be included in the PHD, further adding to capabilities that the EKF cannot handle. Since the multitarget state representation is of a GM, it is straightforward to append the list of components with new objects when they become observable in the FOV through birth. Birth models represent objects entering the FOV and spawns consider the case where new objects are generated from previously existing objects. In this paper, spawning is not implemented.

A key limitation with the PHD is that it assumes a single target can, at most, return a single measurement. This constraining factor would require clustering and centroiding steps for preprocessing, similar to the EKF. If used for a scenario with multiple measurements per object, the PHD would result in a higher estimate for the number of objects since it would treat these sensor returns as measurements from separate objects.

Extended Target Probability Hypothesis Density Filter

To enable tracking objects that can each return multiple measurements, for example, for close range objects (relative to their size and camera parameters), the ET-PHD filter was developed by Granstrom et al.[11] Because of the asynchronous, independently operating pixels in the sensor array of event cameras, it is possible for multiple measurements to be stored with the same time tag that originate from the same object. Similarly, if an object detection method is used for an image frame, bearing angles that map to an object's extent can be processed to replace the use of only an averaged centroid for a given timestep.

Instead of the process used in the PHD, where the set of GM components are all updated given a measurement, the ET-PHD generates partitions of measurement sets to update the GM component weights based on these partitions. The partition set-based update accounts for the possibility that multiple measurements map to the same object, which prevents the inflation of the predicted number of objects based on the total weight of the GM. However, the use of nonlinear bearing angles and range measurements for event-based tracking of on-orbit spacecraft require modifications to the ET-PHD's method of updating GM component weights. The following equations use notation derived from previous work.[11]

For a GM of size J following the propagation step, each component is represented by a respective weight $w_{k|k-1}^{(j)}$, mean $\mathbf{m}_{k|k-1}^{(j)}$, and covariance $\mathbf{P}_{k|k-1}^{(j)}$. In the measurement update, the weight assignment includes an evaluation of

$$\phi_W^{(j)} = \mathcal{N}\left(\mathbf{z}_W; \mathbf{H}_W \mathbf{m}_{k|k-1}^{(j)}, \mathbf{H}_W \mathbf{P}_{k|k-1}^{(j)} \mathbf{H}_W^{\mathrm{T}} + \mathbf{R}_W\right), \tag{14}$$

where $\mathbf{z}_W$ is the vertical vectorial concatenation of measurements in the partition set W, $\mathbf{H}_W$ is the $|W|$-times stack of the linear measurement mapping $\mathbf{H}_k$:

$$\mathbf{H}_W = \left[\mathbf{H}_k^{\mathrm{T}}, \mathbf{H}_k^{\mathrm{T}}, \ldots, \mathbf{H}_k^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{15}$$

and $\mathbf{R}_W$ is the measurement error covariance, similarly stacked as previous, through the block diagonal matrix with $|W|$-times matrix elements $\mathbf{R}_k$:

$$\mathbf{R}_W = \operatorname{blkdiag}\left(\mathbf{R}_k, \mathbf{R}_k, \ldots, \mathbf{R}_k\right) \tag{16}$$

$$= \begin{bmatrix} \mathbf{R}_k & \mathbf{0} & \cdots \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{R}_k \end{bmatrix}, \tag{17}$$

with $\mathbf{0}$ matrices that match the dimension of $\mathbf{R}_k$.

Equation 14 provides an assessment of the likelihood that the current set of measurements is produced by a target with a state corresponding to the j^{th} component of the GM. However, Eq. 14 assumes a linear mapping between the mean state $\mathbf{m}$ to the measurement $\mathbf{z}$ through the use of $\mathbf{H}_W$. To enable processing nonlinear measurements, $\mathbf{H}_W \mathbf{m}_{k|k-1}^{(j)}$ is replaced with an evaluation of the expected measurement model directly for either the bearing angles or the range measurement, which is repeated $|W|$ times vertically, resulting in

$$\mathbf{h}_W(\mathbf{m}_{k|k-1}^{(j)}) = [\mathbf{h}(\mathbf{m}_{k|k-1}^{(j)})^T, \ldots, \mathbf{h}(\mathbf{m}_{k|k-1}^{(j)})^T]^T. \tag{18}$$

Instead of the linear map in Eq. 14, $\mathbf{H}_W$ is replaced by

$$\boldsymbol{\mathcal{H}}_W = \left[\mathbf{H}(\mathbf{m}_{k|k-1}^{(j)})_k^{\mathrm{T}}, \ldots, \mathbf{H}(\mathbf{m}_{k|k-1}^{(j)})_k^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{19}$$

where $\mathbf{H}(\mathbf{m}_{k|k-1}^{(j)})_k^{\mathrm{T}}$ is the evaluation of the measurement Jacobian matrices with respect to the j^{th} GM mean state $\mathbf{m}_{k|k-1}^{(j)}$. These substitutions result in an implementation that handles nonlinear measurement models

$$\phi_W^{(j)} = \mathcal{N}\left(\mathbf{z}_W; \mathbf{h}_W(\mathbf{m}_{k|k-1}^{(j)}), \boldsymbol{\mathcal{H}}_W \mathbf{P}_{k|k-1}^{(j)} \boldsymbol{\mathcal{H}}_W^{\mathrm{T}} + \mathbf{R}_W\right), \tag{20}$$

where Eq. 16 is equivalently used. This measurement stacking operation and Eq. 20 assume that the measurements remain close to the centroid of the tracked object instead of having measurements that are representative of the object's shape. This constraint agrees with the far-range bearing scenario that is the focus of this current paper, however handling events that show the contour of the deputy spacecraft is a subject of ongoing work and would allow for processing observations of closer-range, resolved measurements.

FILTER TESTS

As an intermediary test of the dynamics and measurement models with the implemented filters, the Blender-generated events are first excluded from the results in this section. From these measurement models, simulated bearing and range measurements were generated at different times to assess filter performance independent of event-handling methods. These tests also enable comparisons of filter performance for cases with and without clutter returns included in the measurement sets. For the dynamics propagation with HCW, the chief is modeled to follow a circular orbit with an altitude of 400 km, where the Earth's radius is assumed to be 6378 km and the gravitational parameter μ is 398600 km^3/s^2.

Single Target Test

Figure 3 shows the modeled trajectory given the initial conditions for the centroid of a single deputy spacecraft $\mathbf{x}(t_0) = [0.5, 70, 0, -0.017, -.00262, 0.002]^T$ in units of m and m/s for the position and velocity components, respectively. Through the imposed trajectory constraints, the deputy's trajectory is designed to remain in the camera FOV over the span of a 3 hour simulation. This example simulates different data rates between the bearing angle measurements (1 Hz) and range measurements (0.1 Hz), with 1 measurement each per timestep. For the simulated measurements, $\mathbf{R}_{bearing} = (\frac{1}{3} * \pi/180)^2\ \mathbf{I}_2$ radians^2 and $R_{range} = \frac{1}{3}^2$ m^2. Clutter measurements are not included in this test to provide the simplest set of inputs to the filters tested.

Each filter is initialized with a random sample from the multivariate Gaussian with mean $\mathbf{x}(t_0)$ and covariance $\mathbf{P}_0 = \text{blkdiag}\left(100^2\ \mathbf{I}_3, 10^2\ \mathbf{I}_3\right)$ with units m^2 and (m/s)^2. The PHD filters tested did not include a birth model so the initial GM only includes a single component with weight $w = 1$. For this test, the probability of detection $P_D = 1$, and the probability of survival $P_S = 1$.

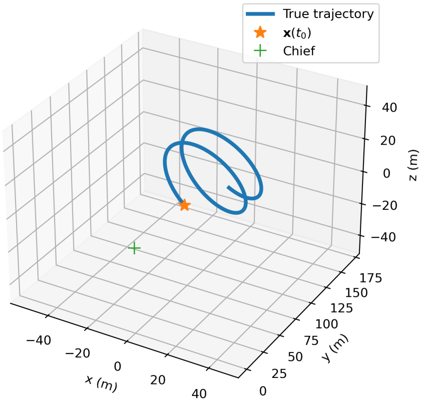
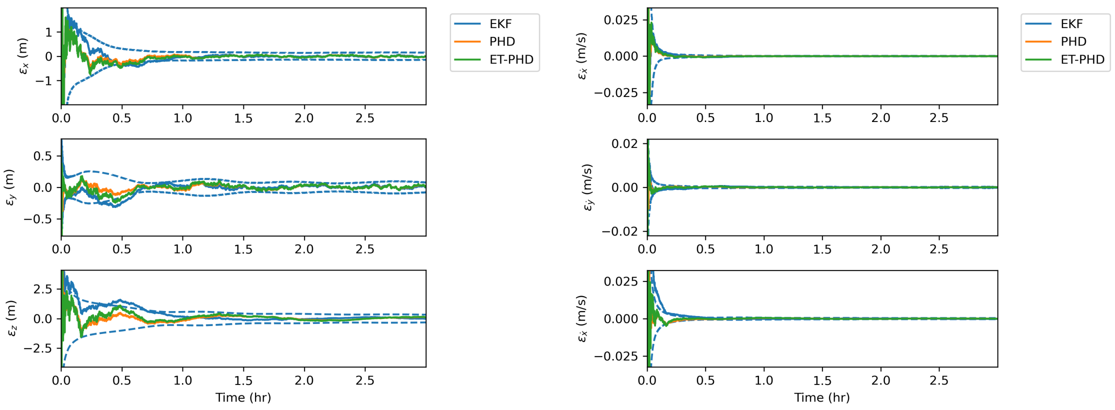


Figure 3. Single target trajectory

Figure 4 shows the position and velocity errors for single test runs of the three implemented filters for a simulation without clutter included. Given bearing angles and range measurements generated by the center of a simulated satellite, the filters converge to accurate state solutions in the duration of the simulated orbit. The results show that the filters are able to obtain accurate solutions without full state observability at each time step, since the bearing angles and range measurements were simulated at different rates. Note that the dashed lines provide the 3σ covariance for the EKF only, where all filters converge within the bounds with unbiased solutions.



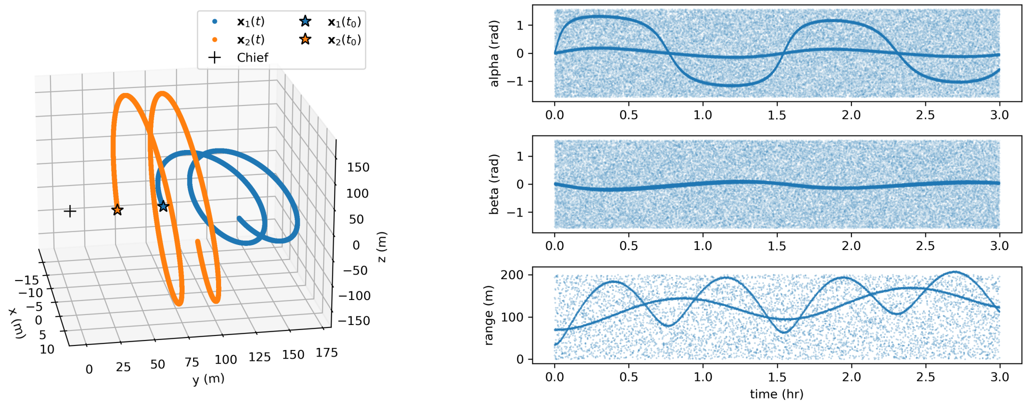
(a) Filter position errors

(b) Filter velocity errors

Figure 4. State error results given HCW dynamics with bearing angles and range measurements of a simulated satellite

Multiple Target Test with Clutter

Handling clutter returns directly in a multiple target test takes full advantage of the PHD-based filters' capabilities. With the inclusion of two targets and clutter measurements, a trajectory case is used as an example below in Fig. 5(a), where $\mathbf{x}_1(t_0) = [0.5, 70, 0, -0.017, -.00262, 0.002]^T$ and $\mathbf{x}_2(t_0) = [0.75, 35, 0, -0.00849, -.00339, 0.002]^T$. This example uses the same data rates 1 Hz for bearing and 0.1 Hz for range, with the inclusion of uniform clutter with constant Poisson rates, $\lambda_{bearing} = 6$ and $\lambda_{range} = 6$. At timesteps where the camera returns observations, 3 bearing measurements are generated for each object, and similarly, the rangefinder returns 3 measurements per object when it is active. The difference in the modeled measurement and clutter rates for the two sensors can be observed in Fig. 5(b), which shows the observation densities in their respective sensor volumes: $[-\pi/2, \pi/2]$ rad^2 for bearing angles and $[0 - 200]$ m for range. The measurement error covariances remain the same from the previous test, and similarly, $P_D = 1$ and $P_S = 1$.

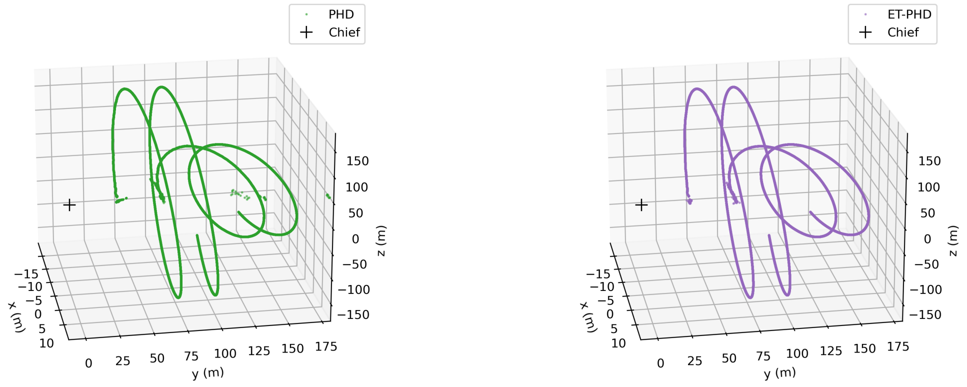


(a) Intersecting true trajectories

(b) Cluttered measurements

Figure 5. Multiple target PHD and ET-PHD test inputs

In this example, the two objects periodically result in intersections in the α bearing angle and in range. As a product of the similarly sized ellipsoid used to generate these trajectories, their β angles overlap throughout the simulation. This test is meant to provide a stress case for the PHD filters, as for both cases, the initial GM for the multitarget state only included a prior for deputy 1, which was randomly sampled from the multivariate Gaussian with mean $\mathbf{x_1}(t_0)$ and covariance $\mathbf{P}_0 = \text{blkdiag}\left(100^2\,\mathbf{I}_3, 10^2\,\mathbf{I}_3\right)$, with weight $w = 1$. The other object was left to be handled by the birth model included in the filter, which is another benefit for using the RFS-based construction of these multitarget tracking methods. Their ability to instantiate tracks and provide state estimates, while also accounting for object spawn and death, are key filtering components that can be broadly applied in target tracking problems. Both filters used a single component GM to represent new born objects with weight $w = 10^{-10}$, $\mathbf{x} = \mathbf{I}_6$, and $\mathbf{P} = 10^3\,\mathbf{I}_6$. In this case, the PHD and the ET-PHD are able to resolve the two individual trajectory tracks, as shown in Fig. 6. In these plots, the points corresponding to the initial part of the two trajectories have some error that improves over time with additional measurements.



(a) PHD estimate

(b) ET-PHD estimate

Figure 6. Trajectory estimates of the PHD filters

Beyond inspection of the output trajectories, the sum of the GM component weights shows the cardinality estimates of the PHD filters.[12] Since the PHD and ET-PHD filters used here do not carry a label for each GM component, which maintains an association-free paradigm, having a metric for cardinality is useful to track how many objects the respective filter is tracking within the FOV. Figure 7 shows that the ET-PHD is more accurate than the PHD in the assessment for the number of tracked objects. In this plot, the ET-PHD is expected to obtain the value of 2 to accurately represent the number of observed targets. In comparison, the PHD on average would obtain a value of 6, since each target returns 3 observations and the PHD does not have a way of accounting for multiple measurements per object. This is a direct result of the measurement set-based update, which enables the ET-PHD to account for the multiple measurement per object case. While the resulting trajectories in Fig. 6 appear to be comparable in accuracy given enough measurements, further inspection of the GM components show the ET-PHD's advantages compared to the PHD.

To isolate the cause in the weight sum variations observed in Fig. 7, additional testing was completed to consider the effects of different inputs to the two filters. As mentioned previously, the measurement partition generation of the ET-PHD allows for set-based updates. This process

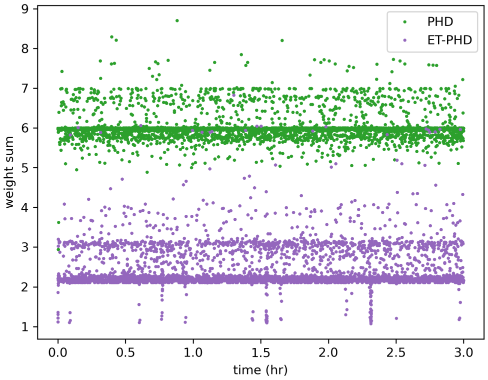
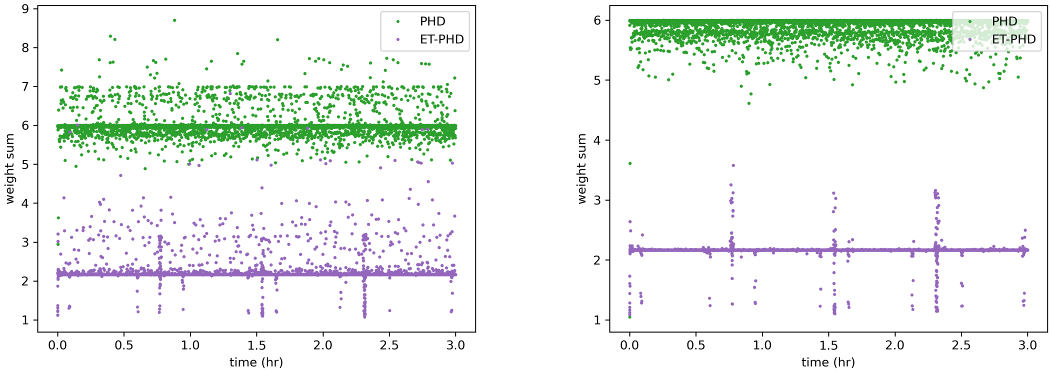


Figure 7. Filter cardinality estimates

includes a truncation for considering fewer partitions than possible to save computation time. In Monte Carlo-based tests done by Granstrom et al,[11] parameters $P_U = 0.8$ and $P_L = 0.3$ were found to provide adequate partitions that result in accurate multitarget state estimates. These parameters provide bounds on the Mahalanobis distance-based partitioning.[11] With all other inputs set equivalently, Fig. 8(a) shows the output of setting $P_U = 0.99$ and $P_L = 0.01$ to include more measurement partitions that previously were excluded out of the set updates. This reduces the variation in the ET-PHD periodic vertical trends more visible. This case is an example where the computational cost of including more set partitions to the filter significantly improves the cardinality estimate. It is then up to a user to define these parameters P_U and P_L based on the trade between runtime and the accuracy of the cardinality estimate.



(a) Expanded partition parameters

(b) No clutter measurements, no birth model

Figure 8. Weight sum testing

Figure 8(b) further simplifies the inputs to the filters, where in this case, the input measurements do not contain clutter and the filters do not have a birth model. Instead, random samples for the two objects are included in the GM prior. This case isolates the vertical lines in the ET-PHD more while also provides an upper bound for the PHD to 6, as predicted by the input measurement rates per object. Further work may consider performing sensitivity analyses to study the interaction between the clutter and birth models. The vertical trends in the ET-PHD outputs in approximately 0.75 hr increments correspond to where the α measurements intersect in Fig. 5(b). At these points, only the ρ measurement is noticeably different between the two objects, which results in the ET-PHD's doubt in maintaining multiple tracks. Because the β measurements between the two objects overlap throughout the simulation, the ET-PHD filter struggles in these intersection times. It is interesting to see that the PHD does not share the same behavior, although this effect may be obscured by this filter's inflated cardinality estimate caused by not having a way to account for multiple returns from each object.

Single Target Test with Event Simulations

Following from the previous single target test without clutter, this section focuses on processing the output measurements from the event simulator into the implemented estimation filters. The object model chosen for this test is a sphere with diameter 10 cm for simplicity, since the primary focus is on translational estimation in far-field rendezvous scenarios. The inclusion of estimating rotational states will be considered for future work, however, the current focus remains on translational states only. For this example, the initial state of the deputy is $\mathbf{x}(t_0) = [0, .75, -350, 0.2, -0.085, 0.00339]^T$, resulting in a rendered image frame like Fig. 9, where the object only occupies several pixels. This particular initial state was chosen because the resulting trajectory generates bearing angles that are constrained to the camera FOV for the entire duration of the simulation. The smaller FOV is limited by the modeled camera focal length and the sensor size. With rendered image frames, the event simulator IEBCS is used to emulate the Prophesee event camera without the clutter model enabled to provide idealized outputs. Table 1 provides a summary of the simulation parameters for this test.

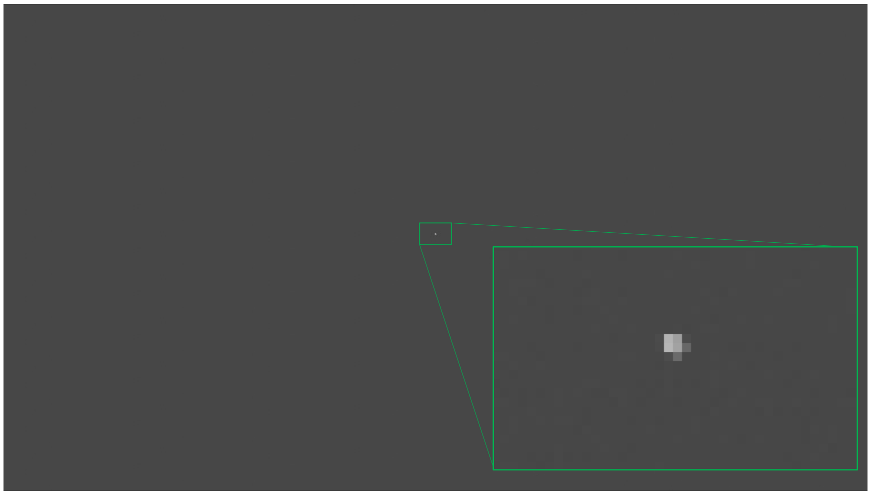


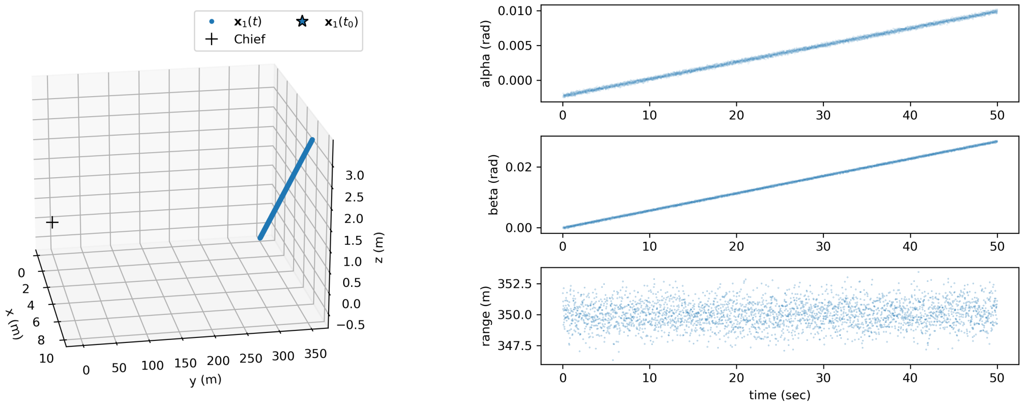
Figure 9. Single target rendered image using Blender

Figure 10(a) provides the deputy's trajectory, with the resulting bearing angles from the event simulator and simulated range measurements in Fig. 10(b). This trajectory input is constrained to only 50 seconds because of the large number of intermediate image frames required to build this test. Event simulators require high image frame rate inputs to emulate the high frequency returns of the asynchronous pixel responses to changes in light. Additional discussion for generating extended

Table 1. Simulation Parameters

Blender Renders		Event Camera		Rangefinder	
Image rate	60 FPS	Focal length	35 mm	Measurement rate	0.1 Hz
Lighting mode	Flat	Sensor size	[6.2208, 3.4992] mm	Noise 3σ	1 m
Object shape	Sphere	Pixel size	4.86 μm	Measurement count	1/timestep
Diameter	10 cm	Measurement rate	1000Hz	Clutter model	None

trajectory tests is included in the next subsection and in future work.



(a) Input trajectory

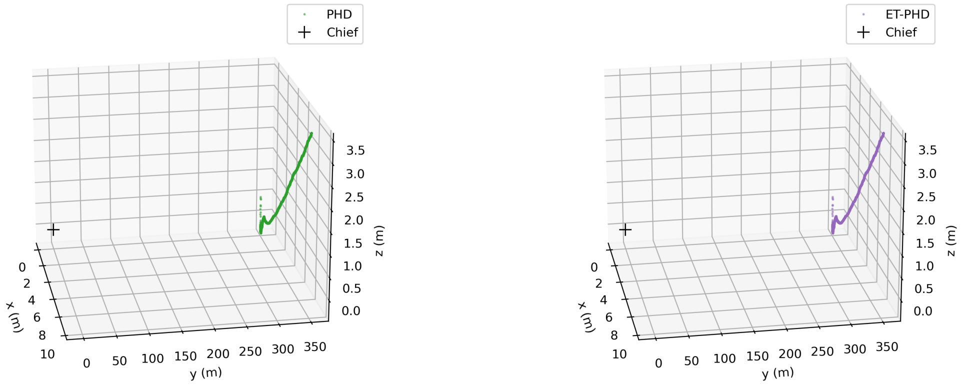
(b) Event measurements with simulated range

Figure 10. Single target PHD and ET-PHD with event simulator inputs

While limited in timespan, the test is meant to provide a proof of concept example for processing bearing angles from events and range measurements with the implemented PHD filters. The filters are initialized with a single component GM with $w = 1$ and a random sample from the multivariate Gaussian with mean $\mathbf{x}(t_0)$ and covariance $\mathbf{P}_0 = \text{blkdiag}\left(100^2\,\mathbf{I}_3, 10^2\,\mathbf{I}_3\right)$, along with parameters $P_D = 1$, $P_S = 1$, $P_L = 0.01$, $P_U = 0.99$, and $\mathbf{R}_{bearing} = (\frac{2}{3} * \pi/180)^2\,\mathbf{I}_2$ radians^2 and $R_{range} = \frac{1}{3}^2$ m^2. The resulting PHD and ET-PHD trajectory estimates are shown in Fig. 11, where the trajectories follow the linear trend in this short simulation timespan. Following the initial corrections to the state, the trajectory estimates are able to approximately follow the true trajectory shown in Fig. 10(a).

Preliminary Work for the Multiple Target Test with Event Simulations

Generating measurements using the event simulator requires long runtimes because of the intermediate, high frame rate input from the image rendering process. Prior to using Blender to generate image frames, the bearing measurements can be checked via the pinhole camera model in Eq. 10 to ensure that the centroid coordinate stays in the camera FOV throughout the trajectory. This is particularly important for the camera model used in the previous section, where the resolution and the 35 mm focal length of the Prophesee camera result in a limited FOV. While an alternative



(a) PHD estimate

(b) ET-PHD estimate

Figure 11. Trajectory estimates of the PHD filters with simulated event measurements

solution to this problem is to model a wide angle lens, using real camera parameters enables the software pipeline to build fidelity instead of relying entirely on user-specified, simulation inputs.

Using the simulation parameters specified in Table 1, two objects with initial conditions are modeled as spherical targets: $\mathbf{x}_1(t_0) = [1, 499.5, 0, -.000283, -.00453, .005]^T$ and $\mathbf{x}_2(t_0) = [1.1, 399.6, 0, -.000226, -004978, .003]^T$. The centroids of these objects were used to generate the predicted measurements, shown in Fig. 12. Generating this plot prior to rendering in Blender allows for verification that the objects will fit within the camera FOV for the duration of the desired timespan of 3 hours. This intermediate measurement prediction step allows the user to quickly adjust the deputy ellipsoid trajectory, instead of waiting for renders to check if or when the objects leave the FOV.

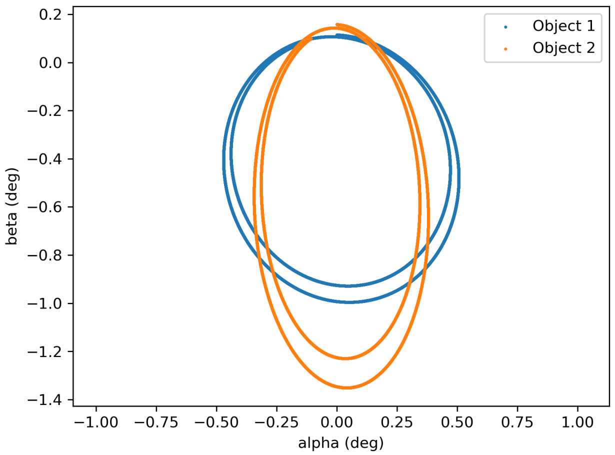


Figure 12. Multiple target input trajectories with simulated bearing measurements

For this test, Fig. 13 shows an example of the Blender-generated image frame that is used as the input to the event simulator for a short duration trajectory. The event simulator was not used to generate a full 3 hour trajectory yet, and the resulting filter tests are left for future work. This is primarily due to the runtime required to generate the events. For a 3 hour simulation and 60 image frames modeled per second, the intermediate Blender process requires 648,000 frames. Following this image generation step, the frame video to event simulation process was hindered by a memory problem likely caused by the large volume of data it needs to handle. Investigating potential methods of resolving this issue are in work, however, this currently prevents further progress on testing the filters given longer trajectories.

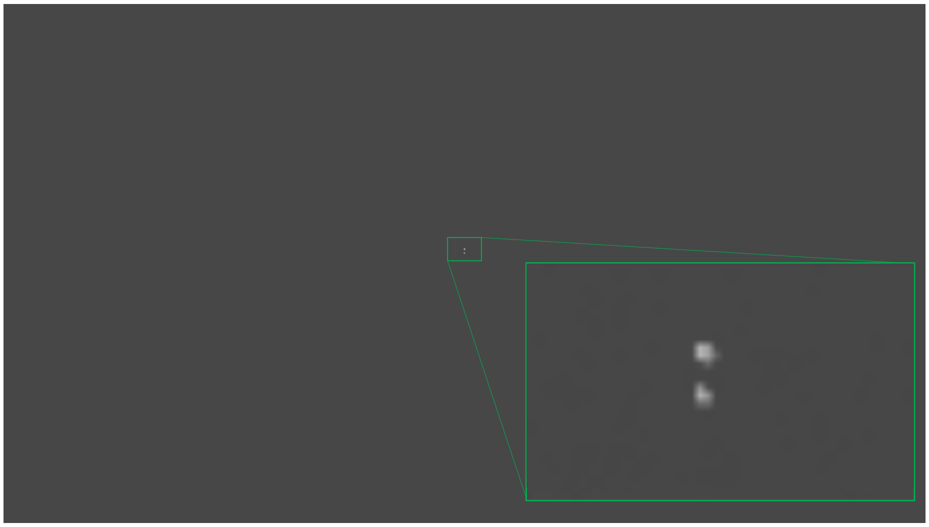


Figure 13. Multiple target rendered image using Blender

The event simulator also includes clutter models that approximate the outputs from a real event sensor, which depend on the ambient lighting conditions. Further testing is planned to include the clutter returns from the event simulator for the long trajectory tests, however, the current runtime and memory issues need to be solved first. Including the clutter measurements will further test the implemented PHD filters by quantifying their response to potentially low signal to noise ratio inputs. Because of the asynchronous pixel response of the event cameras, it is possible that sets of measurements contain all clutter returns. The sensitivity of these conditions may be accounted for by tuning filter parameters P_D, P_L, and P_U, but additional testing will be done to study the filters' response to clutter events.

ONGOING AND FUTURE WORK

With the use of PHD-based filters, knowledge of the measurement and clutter rates are necessary to process measurements and iteratively compute state solutions for the observed targets. In testing, it was imperative to have the measurement simulators (with and without Blender and the event generators in the loop) align with the filter inputs for these measurement rate parameters. Otherwise, the resulting state estimates were observed to diverge, for example, with the ET-PHD weights wrongly including clutter returns as observations of newly birthed targets. An understanding of the sensitivity of these filters' performance to the required measurement rate inputs should be studied because these parameters can be environment-dependent. Modeling the environment's effects may present a challenge, in particular, because the event camera's clutter model depends on the local lighting conditions. While the high dynamic range response is expected to continually track objects in the FOV, the corresponding clutter rates may not be well modeled. In the simulation environment, the generated measurement rates are set by the user, which can then be directly used by the filter test. However, with real camera hardware and environmental conditions that cannot be simply defined by the user, filter robustness to these parameters must be studied. Since this work in the future may consider lab testing with hardware in the loop, it is necessary to study these limitations and propose solutions to enable practical filter implementations.

Some of the challenges in generating tests with Blender and the event simulator in the loop are the large data output from these tools and the runtime this process requires. For this reason, Monte Carlo tests are left to future work to assess filter consistency beyond single run results as presented here. Improved software development and data handling are components that are actively in work for this project, with the goal of reducing the runtime and storage to simulate event camera outputs. Currently, the developed pipeline relies on an intermediary high frame rate video output from Blender to emulate the event camera output well. Instead, other tools can be integrated, like video interpolation methods, to reduce the image outputs stored while maintaining the frame rates required for the event simulator.

CONCLUSION

This paper summarizes the ongoing work towards using event cameras and simulated range measurements for multiple target tracking applications in a relative spacecraft environment. With limits imposed by a camera model's FOV, trajectory constraints are provided to ensure that the modeled objects stay in the camera frame for the duration of the simulation. An implementation of the GM weight update is presented in this work to handle the nonlinear bearing angles and range measurements for the ET-PHD. Through simulated bearing tests with HCW dynamics, single and multiple target tracking examples are provided to iteratively test the filters to assess their response to the measurements. Through position trajectory plots, the resulting state estimates from the PHD and the ET-PHD are shown along with a comparison analysis of their respective cardinality estimates. If allowed more runtime, the ET-PHD's measurement set-based update is able to more accurately track the number of objects in the FOV. As expected, the PHD provides an overestimate for cardinality in cases where multiple measurements are produced by the same object. Preliminary work is included for filter tests integrated with event simulators, however, future work aims to further develop and test methods to handle high clutter rates from the modeled event measurements.

ACKNOWLEDGEMENTS

This work was supported by the NASA Space Technology Graduate Research Opportunities (NSTGRO) Fellowship, Grant Number 80NSSC22K1202.

REFERENCES

[1] R. T. Howard, T. C. Bryan, L. L. Brewster, and J. E. Lee, "Proximity operations and docking sensor development," *2009 IEEE Aerospace conference*, IEEE, 2009, pp. 1–10.

[2] J. D. Mitchell, S. P. Cryan, D. Strack, L. L. Brewster, M. J. Williamson, R. T. Howard, and A. Johnston, "Automated rendezvous and docking sensor testing at the flight robotics laboratory," *2007 IEEE Aerospace Conference*, IEEE, 2007, pp. 1–16.

[3] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 44, No. 1, 2020, pp. 154–180.

[4] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-DOF pose tracking for high-speed maneuvers," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2761–2768, 10.1109/IROS.2014.6942940.

[5] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "Dsec: A stereo event camera dataset for driving scenarios," *IEEE Robotics and Automation Letters*, Vol. 6, No. 3, 2021, pp. 4947–4954.

[6] M. Żołnowski, R. Reszelewski, D. P. Moeys, T. Delbrück, and K. Kamiński, "Observational evaluation of event cameras performance in optical space surveillance," *Proc. 1st NEO and Debris Detection Conference, Darmstadt, Germany*, 2019, pp. 22–24.

[7] M. G. McHarg, R. L. Balthazor, B. J. McReynolds, D. H. Howe, C. J. Maloney, D. O'Keefe, R. Bam, G. Wilson, P. Karki, A. Marcireau, *et al.*, "Falcon Neuro: an Event-Based Sensor on the International Space Station," *Optical Engineering*, Vol. 61, No. 8, 2022, p. 085105.

[8] M. Gehrig and D. Scaramuzza, "Recurrent Vision Transformers for Object Detection with Event Cameras," *arXiv preprint arXiv:2212.05598*, 2022.

[9] L. Azzalini, E. Blazquez, A. Hadjiivanov, G. Meoni, and D. Izzo, "Generating a Synthetic Event-Based Vision Dataset for Navigation and Landing," *12th International Conference on Guidance, Navigation and Control Systems (ESA-GNC)*, Sopot, Poland, 2023.

[10] S. G. Catalan, B. A. Jones, and M. A. Shoemaker, "Applying Event-Based Sensors to Relative Spacecraft State Estimation," *AAS/AIAA Astrodynamics Specialist Conference*, August 2023.

[11] K. Granström, C. Lundquist, and O. Orguner, "Extended target tracking using a Gaussian-mixture PHD filter," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 4, 2012, pp. 3268–3286.

[12] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on signal processing*, Vol. 54, No. 11, 2006, pp. 4091–4104.

[13] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE journal of solid-state circuits*, Vol. 43, No. 2, 2008, pp. 566–576.

[14] Blender Online Community, *Blender - a 3D Modelling and Rendering Package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[15] D. Joubert, A. Marcireau, N. Ralph, A. Jolley, A. v. Schaik, and G. Cohen, "Event camera simulator improvements via characterized parameters," *Frontiers in Neuroscience*, 2021, p. 910.

[16] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, Vol. 4, pp. 398–411. Microcosm Press, Hawthorne, CA, 2013.

[17] Y. Ma, S. Soatto, J. Košecká, and S. Sastry, *An Invitation to 3D Vision: from Images to Geometric Models*, Vol. 26, pp. 44–62. Springer, 2004.

[18] D. C. Woffinden, *Angles-Only Navigation for Autonomous Orbital Rendezvous*. PhD thesis, Utah State University, 2008.

[19] R. Mahler, "PHD Filters for Nonstandard Targets, I: Extended Targets," *2009 12th International Conference on Information Fusion*, IEEE, 2009, pp. 915–921.

[20] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, 2014.