Tracking of Few-Pixel UAVs in Event Data

Jonatan Adolfsson*, Hanna Hamrell*, and David Gustafsson*
*Swedish Defence Research Agency (FOI)
Email: {jonatan.adolfsson, hanna.hamrell, david.gustafsson}@foi.se

***Abstract*—The event camera is a relatively new type of sensor where each pixel asynchronously reports changes in incident light, resulting in low latency and high energy efficiency. This opens for new possibilities within counter Unmanned Aerial Vehicle (UAV) applications, crucial to tackle the threat of this increasingly widespread technology. However, this calls for new data processing methods. As a contribution to both of these challenges, we investigate the possibility of tracking UAVs that cover only a single or few pixels in event camera data. Scene and background activity noise is suppressed using a novel noise filter. Tracking is performed in the image plane using a standard Multi-Hypothesis Tracker (MHT) with parameters optimised for the data. The method is evaluated on several sequences of UAV data collected in an outdoor setting, exhibiting a variety of motion patterns and target sizes. The GOSPA measure and the fraction of observations within the covariance estimate are used to evaluate the results. Excellent tracking performance is achieved for regularly moving targets with a size of 2-15 pixels. Given the UAV and setup used in this study, this corresponds to distances of about 10-30 m when viewing the UAV from the side, or 20-65 m when viewed from below. Tracking is still possible for irregular motion in the same range, whereas rapid movements at close range and single-pixel targets are more challenging. We show that it is perfectly viable to track UAVs of various pixel sizes in event-based data using an MHT algorithm with an appropriate denoising method.**

***Index Terms*—event camera, tracking, counter UAV**

I. Introduction

The widespread use and accessibility of Unmanned Aerial Vehicles (UAVs) have resulted in an inevitable need of protection from said technology. An important aspect of this is to be able to robustly detect and track UAVs to quickly remove them from places where they are not allowed or constitute a threat. In many counter UAV use cases, concealing the position of the tracking sensors helps protecting them from sabotage or tampering, making a passive sensor preferable. Further, a sensor that is cheap, fast, and energy efficient enables multiple sensors to be used to protect a large area without major costs or maintenance, which is especially useful for remote places far from infrastructure. One sensor that meets all these criteria is the event camera. This device generates an asynchronous stream of events based on changes in incident light (see II-A), unlike conventional cameras which produce frames with data in all pixels at fixed time intervals based on the sensor output.

New methods need to be developed for effective processing of event camera data for detection and tracking. In this work, we evaluate the possibilities of tracking few-pixel UAV targets, flying irregularly in a realistic setting. In a first step, we apply a novel noise filtering method adapted to the data. Next, targets are detected using clustering and tracked in the image plane using a standard Multi Hypothesis Tracking (MHT) method. The full tracker is evaluated on several sequences of recorded data, exhibiting different flying behaviours.

This work was funded by the Swedish Defence Research Agency (FOI) and the Swedish Armed Forces R&D programme for Sensors and low observables (FoT SoS, AT.9220423).

II. Background and Previous Work

The event camera is described in Sec. II-A. A significant aspect of tracking in event data is denoising. Therefore, sources of noise and previous work on noise filtering in event data are described in Sec. II-B. Previous work on tracking in event data is summarised in Sec. II-C, whereas various data association algorithms are described in Sec II-D.

A. The Event Camera

Conventional cameras generate frames at discrete time intervals, regardless of changes in the scene. This is very costly energy-wise. A relatively new type of sensor is the event camera, which is a biologically inspired sensor in the sense that each pixel independently generates a so-called *event* each time the accumulated change in incident light pass a predefined threshold. Each event contains information of whether the light intensity has increased or decreased (i.e. polarity), the pixel position, and a time stamp. Since events are only created if the change in incident light has been significant enough, a mostly static scene common in many surveillance use cases results in a very energy efficient sensor. Moreover, the sensor has a very high dynamic range, making it robust against blinding. These properties make it especially suitable for surveillance of areas far from existing infrastructure or as a complement to other surveillance sensors. A well-written overview of the event camera and related signal processing methods and applications is given in [1].

B. Noise Filtering of Event Data

There are several different types of noise in event camera data. Noise is caused by stochastic arrival of events, random variations in the number of events given by the same change in light intensity, as well as changes in light intensity not always triggering an event. Further, there is background activity (BA) noise, i.e. noise caused by pixels that generate events, even if no intensity change has taken place. This is caused by photon shot noise, thermal noise in the circuits, junction leakage noise, and fixed-pattern noise due to variations in the temporal contrast detection thresholds across pixels [2], [3].

Several different methods for denoising event data have been developed. Most of them focus on mitigating BA noise, using the fact that it is temporally uncorrelated with events in its spatial neighbourhood, unlike events caused by activity in the scene. These filters include software based spatio-temporal filters [4], [5], as well as hardware based ones [6], [7]. There are also bio-inspired filters [8]. Other filters focus more on removing unwanted data from the scene. These include a neural network based denoising model [9], a linear comb filter to remove flickering from fluorescent light [10], and the event polarity based anti-flicker algorithm PINK [11].

C. Tracking in Event Data

Existing literature on tracking in event data mostly focuses on tracking features for SLAM [12]–[15], or of tracking unmanned vehicles to aid their navigation and path planning [16], [17]. There is also work done on clustering and multi-target tracking of objects in event data for robot perception [18]. One study demonstrates the use of a Probabilistic Multi-Hypothesis Tracker (PMHT) for tracking low-orbit celestial objects with an event camera [19]. References [12], [14], and [18] use asynchronous tracking methods, whereas the rest of the above studies convert the event data into frames and perform tracking on clusters of events. Another example of asynchronous tracking is used in [20] and [21], where intruders are tracked in mostly static areas using flying UAVs. Here, the data from the event stream is directly converted into feature clusters for the tracker, resulting in an adaptive time step.

D. Tracking Algorithms

A target tracker uses measurements that are processed by an observation model, a filter algorithm for the time update, and a track logic for initialising new tracks, updating existing ones, and determining whether to keep or kill a track (see [22] for details). There are both single- and multi-target trackers, as well as single- and multi-hypothesis algorithms. For a single-hypothesis algorithm (such as Global Nearest Neighbour, GNN), track predictions persist after future measurement updates, making it sensitive to incorrect assignments. Thus, a multi-hypothesis tracker is preferred, although it may have worse time performance. Commonly used multi-hypothesis trackers are the MHT [22], the Joint Probabilistic Data Association (JDPA) filter, the Probability Hypothesis Density (PHD) filter [23], and the Poisson Multi-Bernoulli Mixture filter [24]. A sometimes-used extension to the MHT is the Probabilistic MHT (PMHT), which contrary to the regular MHT has linear time complexity in the number of tracks and hypotheses [25].

The regular MHT treats different tracking hypotheses separately in a hypothesis tree and prunes unlikely hypotheses based on the tracking score. The JDPA instead weighs the different hypotheses into a global probability density, which is used for the association. PHD and PMBM use a different approach, where initially, the full target distribution is tracked instead of creating individual tracks. In a comparison study, PHD and PMBM had better performance than GNN and JDPA in a high-clutter tracking scenario, with PMBM performing the best [26]. On the other hand, the time performance is generally worse for these methods, and in particular PMBM.

Fig. 1: **Left and middle:** The two different sensor setups. **Right:** The UAV of model Anafi from Parrot.

III. Method

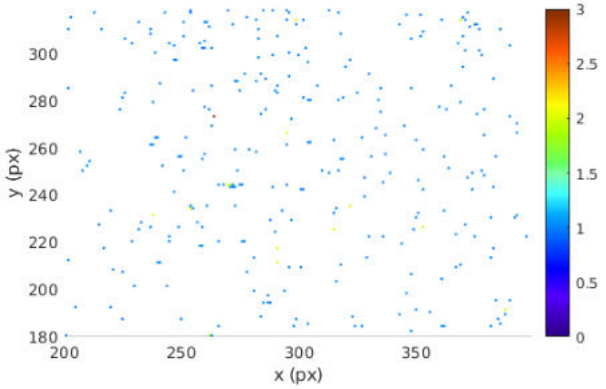
The purpose of this study is to develop a method for tracking UAVs exhibiting various motion patterns, aiming at tracking targets with a size down to only one or a few pixels. Such data was acquired using an event camera and a small UAV, see details in Sec. III-A. Due to the large span of target sizes and large noise levels, it was determined that existing asynchronous tracking methods would not be feasible. Instead, the event data was first converted into frames, see Sec. III-B. We contribute by presenting a new denoising method that works especially well for noisy data with few-pixel targets, described in Sec. III-C. The data was subsequently used as input for tracking, see Sec. III-D. The tracking parameters were optimised on a subset of the data, as described in Sec. III-E. The same parameters were then used for all data. Finally, the evaluation of the tracking is described in Sec. III-F.

A. Data Acquisition

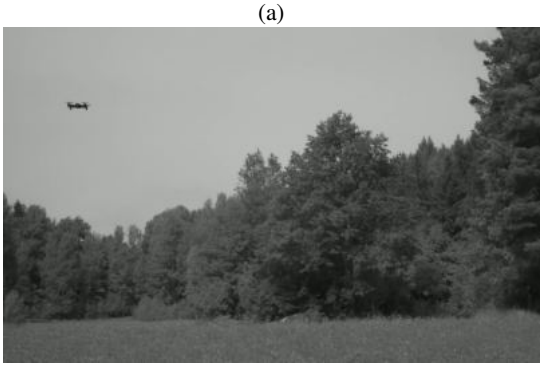
Two sensors were used for collecting data – a DVXplorer event camera from iniVation with a resolution of 640×480 pixels [27] and a visual FLIR BlackFly camera [28] for reference imagery. The sensors were placed next to each other and capturing roughly the same scene, see Fig. 1. The event camera had a 4 mm lens, with a focal length $f = 4\,\mathrm{mm}$.

Two different sensor setups were used. In the first setup ("horizontal mounting"), the sensors were directed towards forestry and sky at the end of a field. In the second setup ("vertical mounting"), they were directed towards the sky. In both setups a Parrot Anafi UAV (size $175 \times 240 \times 65\,\mathrm{mm}$ [29]), see Fig. 1, was flown in front of or above the sensors. During the first setup, the UAV flew in various patterns at different distances from the sensor. During the second setup, the UAV flew in a straight line above the sensors, at varying heights. A summary of all UAV sequences is given in Tab. I. A few examples of the collected data can be seen in Fig. 2.

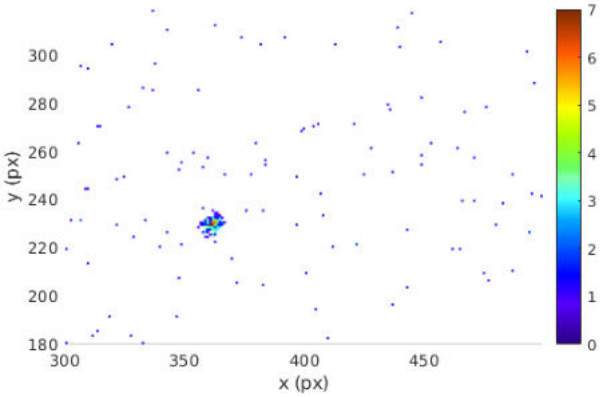
To estimate the sensor noise in the event camera, two measurements were performed with the camera placed inside a dark container. Each recording was approximately one minute, with the first one being executed before the UAV sequences and the other one afterwards. These data sets were used for noise suppression.



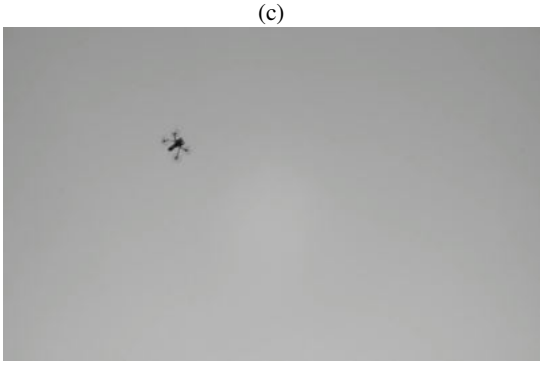
(a)



(b)



(c)



(d)

Fig. 2: Example images from two of the recorded sequences. The event camera images show the number events per pixel accumulated during 50 ms, and are cropped around the UAV. **(a)** Event data from *horizontal_25m*, **(b)** visual frame from the same scene as in **(a)**, **(c)** event data example from *from_below_20m*, **(d)** visual frame from the same scene as in **(c)**.

TABLE I: Description of each measurement scenario.

Sequences	Description
irregular_Xm	The UAV flies irregularly up, down, and sideways, at a distance to the sensor of approximately $X \pm 5\,\mathrm{m}$.
horizontal_Xm	The UAV flies sideways back and forth, keeping a distance of about X metres from the sensor.
circle	The UAV flies in a large circle at a distance between 25 and 35 metres from the sensor.
slanting	The UAV flies diagonally in the image plane, starting 25 m from the sensor and ending at 40 m.
down_up_Xm	The UAV starts at a position near ground and moves upwards at a distance X m from the sensor.
towards_sensor	The UAV flies from a point 40 m away from the sensor straight towards the sensor until it is 5 m from it.
from_below_Xm	The only sequence set using the vertical camera setup. The UAV flies at a height X m above ground. Sequences include distances between 5 and 80 m.

B. Conversion from Event Data to Frames

The event camera data is structured as a stream of events, only recording the time, pixel coordinates, and polarity of each event (positive or negative). To use this for tracking, all events during each time interval of length T were binned into a frame, yielding the number of events $N_\mathrm{f}(x, y, p)$ per pixel with coordinates (x, y) in polarity channel p. The two polarity channels were later combined after background suppression, see Sec. III-C. Ideally, T should be short to catch fast movements, but if chosen too short the data becomes too sparse for the tracker to be able to find the smallest targets. We chose a compromise that worked well for our setup, $T = 50\,\mathrm{ms}$.

C. Background Suppression

The collected data is noisy (cf. Fig. 2a), so the noise needed to be reduced before further processing. At first, there were several hypersensitive pixels that were constantly generating events. To filter those out, and also reduce other intrinsic background events, one of the noise data sets was used. Let T_tot be the full length of this sequence and $N(x, y, p)$ be the total number of events during T_tot. Then the mean number of background events per frame, N_noise, is calculated as

$$N_\mathrm{noise}(x, y, p) = \frac{N(x, y, p) \cdot T}{T_\mathrm{tot}}. \tag{1}$$

This was subtracted from each frame, yielding

$$N^\mathrm{init}_\mathrm{filt}(x, y, p) = N_\mathrm{f}(x, y, p) - N_\mathrm{noise}(x, y, p). \tag{2}$$

Next, only pixels where the measured number of events is significantly above the noise level were kept,

$$N_\mathrm{filt}(x, y, p) = \begin{cases} 0, & N^\mathrm{init}_\mathrm{filt}(x, y, p) < 3\sigma_{N_\mathrm{f}(x,y,p)} \\ N^\mathrm{init}_\mathrm{filt}(x, y, p), & \text{otherwise} \end{cases}, \tag{3}$$

where the Poisson estimate is used for the standard deviation, $\sigma_{N_\mathrm{f}} = \sqrt{N_\mathrm{f}}$.

At last, the final data frames were retrieved by combining the two channels, i.e.

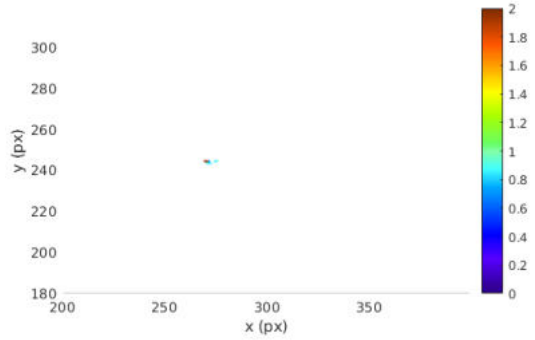


Fig. 3: The same display as in Fig. 2a after noise reduction. Tracking was performed on the centre points of each pixel cluster.

$$N_{\mathrm{f}}^{\mathrm{final}}(x,y) = \sum_{p} N_{\mathrm{filt}}(x,y,p). \tag{4}$$

This did however not suppress all hypersensitive pixels (likely due to different light conditions during the measurements), so 26 pixels were manually removed from the analysis.

The remaining data in each frame was grouped into clusters consisting of all adjacently (horizontally, vertically, or diagonally) connected pixels with events in them. The data still contained significant low-frequency noise, so to reduce this it was observed that moving objects produce events with both positive and negative polarity in close vicinity, whereas e.g. thermal noise only produces events with positive polarity. Thus, only clusters with a negative event observed within d pixels from the centre during the last M frames were used for tracking. We used $d = 5\sqrt{N_{\mathrm{c}}}$, where N_{c} is the number of pixels in the cluster, and $M = 6$. As a final filtering step, only clusters with $N_{\mathrm{f}}^{\mathrm{final}} \geq 1.1$ were included (in practice this means that at least two events are required in most cases). An example of a cleaned event display is shown in Fig. 3.

D. Tracking

The tracking was performed in the image plane, with state parameters

$$\mathbf{X} = (x, y, v_x, v_y, a_x, a_y)^{\mathbf{T}} \equiv (\mathbf{x}, \mathbf{v}, \mathbf{a})^{\mathbf{T}},$$

where x and y are the image coordinates, and v_i and a_i are velocity and acceleration of these parameters, respectively. The motion was described by an Interactive Multiple Model (IMM) filter [30], where three different motion models were mixed to describe all possible scenarios. These were:

1) A fixed-position (FP) model with $\mathbf{v} = \mathbf{a} = \mathbf{0}$,

$$\mathbf{X}_{t+1}^{\mathrm{FP}} = \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{v}_t \\ \mathbf{a}_t \end{pmatrix}. \tag{5}$$

2) A constant-velocity (CV) model with $\mathbf{a} = \mathbf{0}$,

$$\mathbf{X}_{t+1}^{\mathrm{CV}} = \begin{pmatrix} I & T & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{v}_t \\ \mathbf{a}_t \end{pmatrix}, \tag{6}$$

where each block is proportional to the identity matrix.

3) A constant-acceleration (CA) model,

$$\mathbf{X}_{t+1}^{\mathrm{CA}} = \begin{pmatrix} I & T & T^2/2 \\ 0 & I & T \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{v}_t \\ \mathbf{a}_t \end{pmatrix}. \tag{7}$$

These were passed into an Extended Kalman Filter (EKF), where the process noise was modelled as a zero-mean Gaussian with covariance $Q = q^2 \cdot GG^{\mathbf{T}}$, where G is obtained by integrating the motion model (cf. [22, pp. 157-168, 287-288]). Here q was set to a large value to catch sudden turns in the movement. The IMM filter used a transition matrix Π, where direct transitions were allowed between the CA model and the others, but not between the FP and CV models, since this would be unphysical.

Further, the tracker used a linear observation model

$$\mathbf{z} = (x, y),$$

where x and y are image coordinates. The observation model used a zero-mean isotropic Gaussian measurement noise model with standard deviation R.

Tracking was performed using a track based MHT with five hypotheses, based on [22, pp. 325-369, 1069-1111], and using N-best association and elliptic gating. This method was chosen since it was easy to implement given our resources, and serves the purpose of this study. New tracks were assigned a state prior P_0 of $5\,\mathrm{px}$ in position, $300\,\mathrm{px/s}$ in velocity, and $50\,\mathrm{px/s^2}$ in acceleration to be able to catch reasonably fast movements. Tracks were initialised if matching observations were found in two consecutive frames, followed by another two within the next eight frames. A new track was not allowed to be initialised within 30 pixels of an existing one to prevent duplicate track assignments from the same target. The model used detection and false-alarm probabilities $p_{\mathrm{D}} = 0.45$ and $p_{\mathrm{FA}} = 2 \cdot 10^{-5}$, respectively.[1] Tracks were kept until either their likelihood was too low, or there was no matching observation for $2\,\mathrm{s}$ (40 frames).

E. Optimisation of Π and R

The process noise covariance scale factor q was constrained to 50 pixels to reduce jittering (the optimum is rather a bit higher). The transition matrix Π and the measurement noise standard deviation R were determined through optimisation over a sub-sample of the data. The cost function that was minimised was

$$\Gamma + 2000 \cdot \lambda,$$

where Γ is the reduced GOSPA measure, here defined as (original definition in [31])

$$\Gamma = \sqrt{\sum_{\mathrm{truetracks}} \min\left(\|\mathbf{x}_{\mathrm{track}} - \mathbf{x}_{\mathrm{truth}}\|_2, d_{\mathrm{cutoff}}\right)^2}, \tag{8}$$

[1]The parameters p_{D} and p_{FA} depend on multiple parameters, such as target size. For simplification, we used empirical values that worked reasonably well for our measurements.

TABLE II: Summary of the tracking parameters used by the MHT.

Parameter	Value
P_0 $(\mathbf{x}, \mathbf{v}, \mathbf{a})$	$(5\,\text{px}, 300\,\text{px/s}, 50\,\text{px/s}^2)$
q	$50\,\text{px}$
R	$7.5\,\text{px}$
Π (order FP–CV–CA)	$\begin{pmatrix} 0.67 & 0 & 0.33 \\ 0 & 0.84 & 0.16 \\ 0.29 & 0.18 & 0.53 \end{pmatrix}$
p_D	0.45
p_FA	$2 \cdot 10^{-5}$
Minimum track distance	$30\,\text{px}$
Minimum initial speed	$0\,\text{px/s}$
Tentative track logic	$2-2-8$
Maximum tentative length	15
Maximum lost observations	40
Gate threshold	$6\,\sigma$
Number of hypotheses	5
MHT pruning depth	10
Score cap	30
New-track probability	$2 \cdot 10^{-7}$
False-track confirmation probability	0.025
True-track miss probability	0.05

where $\mathbf{x}_\text{truth}$ and $\mathbf{x}_\text{track}$ are the $\mathbf{x}$ values of the ground truth track and best-matched detected track (which in turn depends on the tracking parameters), respectively, if they exist, and $d_\text{cutoff} = 100$ pixels is the cut-off distance. The regularisation parameter λ is the number of distinct tracks building the sequence $\mathbf{x}_\text{track}$. The factor 2000 was determined to give a good balance between minimising the metric and penalising lost tracks. The sum is only taken over the ground truth to maximise tracking performance of the true target(s); thus false tracks are ignored in the measure. The reason is that those probably can be removed in a later step, e.g. by studying the motion parameters. The optimisation was done over the estimate from a single iteration per sample[2] with Brent's method [32], which was useful since the uncertainty of the measure was too large to easily obtain derivatives. The chosen sub-sample was one of the two *irregular_20m* sequences, since this included several motion patterns and the relatively large target size made it easy both to track and to estimate the ground truth.

All tracking parameters (including the results from the optimisation and those not mentioned in the text) are summarised in Tab. II. The parameters that were not included for optimisation were chosen since they were observed to work well and it was not expected that changing them would affect the results very much. A tracking example is shown in Fig. 4.

F. Evaluation

In order to obtain a ground-truth estimate of the data, the target tracks in each sequence were manually annotated in the framed data. Based on this, two measures were evaluated:

1) The reduced GOSPA measure defined by (8).

[2]Ideally many iterations should be used for the optimisation, but this was not done due to time constraints. Moreover, the purpose was rather to obtain a good general-purpose parameter set than optimal settings for the sub-sample used.

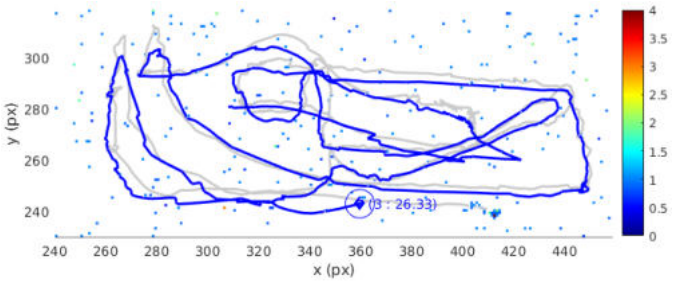


Fig. 4: Tracking example showing the beginning of the sequence *irregular_20m attempt 1*. Grey: ground truth, blue: predicted track. The numbers are the track ID and the log-likelihood of the track score (capped at 30). At this moment, the tracker is close to losing the target.

2) The percentage of the time the true track is within the $1\,\sigma$ and $2\,\sigma$ covariance ellipses, respectively, of the estimated position. Since a Gaussian approximation is used, the true position should fall within these limits 68% and 95% of the time, respectively, in case of ideal tracking.

These measures were first evaluated for each individual sequence. However, since the target distance varies within each sequence, and hence the apparent target size, we have also evaluated the tracking performance as a function of target size. The measure used for the latter is the transversal size $w_\perp$ of the UAV relative to the direction of motion. This was chosen since it should be independent of the target speed, although it depends on the movement direction if the UAV is viewed from the side due to the UAV's dimensions. When viewed from below, the size of the UAV should be independent of the direction of movement, and only depend on the distance to the target. During this measurement, the measures were computed in bins of $w_\perp$ intervals instead of over the full track. All measurements were repeated 10 times to capture uncertainties in the evaluation.

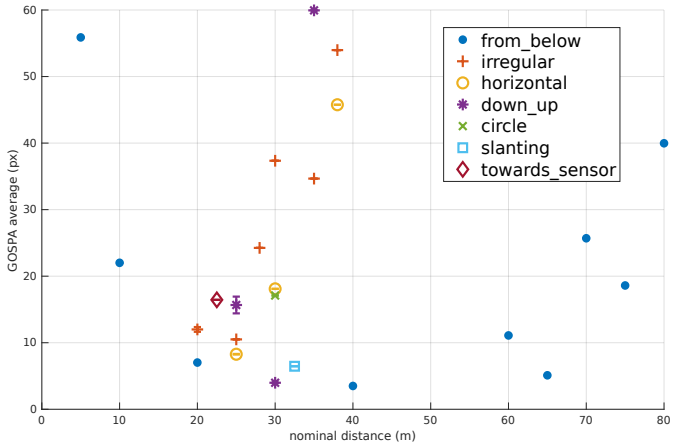
To extract $w_\perp$ at time t, all adjacent cells with at least one event during the time interval $[t-1, t]$ were clustered together. The target was estimated as the cluster with the largest number of events along the trajectory between the annotated data points $\mathbf{x}_{t-1}$ and $\mathbf{x}_{t+1}$ (or the closest cluster if none existed along that path). Then $w_\perp$ was estimated as

$$w_\perp \approx \sqrt{(v_y\ -v_x)\text{Cov}(\mathbf{x},\mathbf{y})(v_y\ -v_x)^\mathbf{T} \cdot 12/\|\mathbf{v}\|^2}, \quad (9)$$

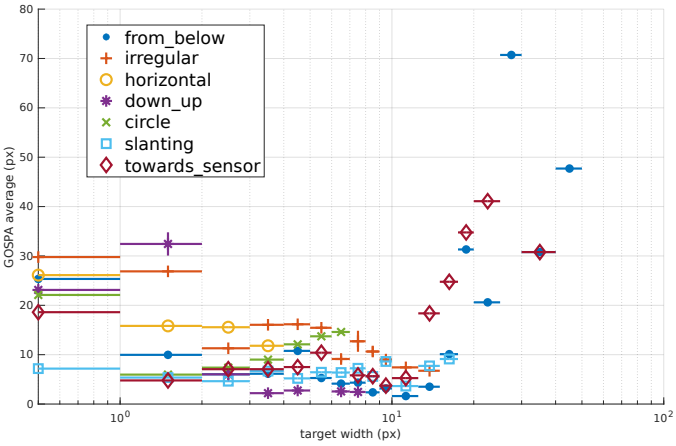
where $(\mathbf{x}, \mathbf{y})$ is the position vector of all events in the cluster and $\mathbf{v} = (v_x, v_y)$ is the velocity of the target (calculated from a smoothed trajectory of the ground truth estimate). The factor 12 ensures this to be correct for a uniform, rectangular target.

IV. Tracking Results

Figure 5 shows the reduced GOSPA values (8) per frame as a function of both the nominal distance of each data sequence (or the average in case of a large distance span) and the transversal target size. In the latter case, all measurements

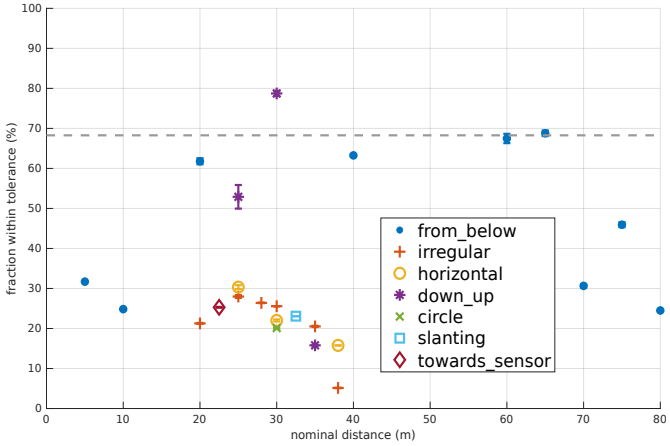


(a)

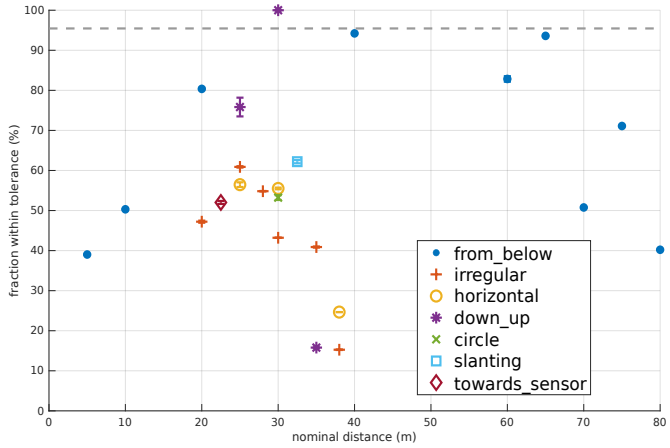


(b)

Fig. 5: Average reduced GOSPA measure (8) per frame for all measurements, averaged over **(a)** each data sequence, and **(b)** each $w_\perp$ bin in each category. The error bars (which are sometimes smaller than the markers) show the statistical error estimated from a small number of tracking repetitions.



(a)



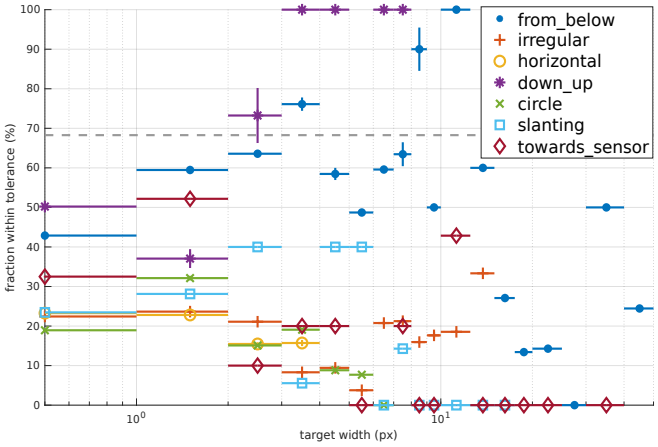
(b)

Fig. 6: Percentage of frames where the ground truth estimate falls within the **(a)** 1σ and **b)** 2σ covariance ellipses, respectively, of the target track, averaged over each data sequence. The dashed lines mark the expected values for ideal tracking, i.e. 68% and 95%, respectively. The error bars (which are sometimes smaller than the markers) show the statistical error estimated from a small number of tracking repetitions.

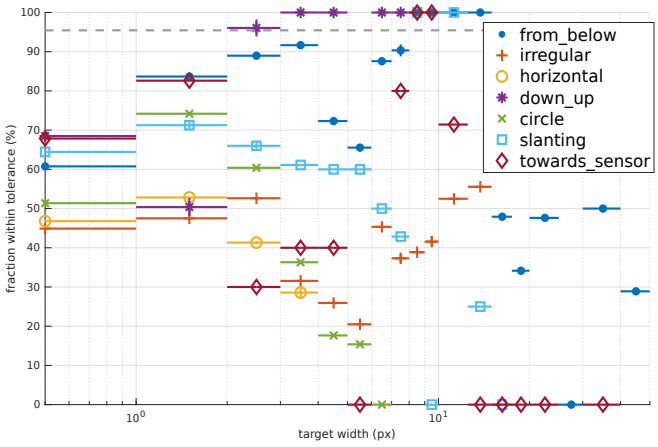
of each sequence category (as defined in Tab. I) have been grouped together, and the average has been taken over each $w_\perp$ bin. For this measure, lower values equals better tracking, with a maximum value at $\overline{\Gamma} = d_{\text{cutoff}} = 100$ pixels. For most categories, there is good tracking at intermediate distances and target sizes, and considerably worse close or far away (large or small targets). In the former case, the UAV was either moving at very high speed in the image (*from_below* measurements) or had rapid turns (*irregular*, *towards_sensor*), making it difficult for the tracker to catch up. In the latter case it was sometimes difficult to find the target.

The range for good tracking is remarkably different between the *from_below* measurements and the horizontal measurements, where the tracking in the former case is best between 20 and 65 m, as opposed to 10 to 30-40 m in the latter case (depending on measurement). Part of the explanation can be seen in the size dependent results, where most measurements have good tracking in the range 2-15 pixels. Therefore the difference in range dependence is largely due to the UAV appearing smaller when seen from the side. However, the *irregular*, *horizontal*, and *circle* categories had generally worse tracking than the other measurements, so this cannot be the full explanation.

Figures 6 and 7 show the fraction of data points falling within 1σ and 2σ, respectively, of the estimated covariance ellipse of the closest track. In case of ideal tracking, these should be 68% and 95%, respectively, which is close to what is observed in some measurements. In particular, this is seen for the intermediate-range *from_below* measurements, but also for the two closest-distance *down_up* measurements, which correlates with the target size. Other than that, the performance is significantly worse for the horizontal measurements than for most *from_below* measurements. For most horizontal measurements with $r \lesssim 35\,\text{m}$, the performance is similar to the



(a)



(b)

Fig. 7: Percentage of frames where the ground truth estimate falls within the **(a)** 1σ and **b)** 2σ covariance ellipses, respectively, of the target track, summed over each category and averaged over each $w_\perp$ bin. The dashed lines mark the expected values for ideal tracking, i.e. 68% and 95%, respectively. The error bars (which are sometimes smaller than the markers) show the statistical error estimated from a small number of tracking repetitions.

from_below_10m measurement. Both for the *from_below_10m* and the *irregular* measurements, it was observed that the tracker frequently fell behind the target, causing it to fall outside the covariance ellipses. For the 38 m measurements, the target was within the estimates only a few percent of the time, meaning that it was mostly lost to the tracker. Similar observations are seen for several of the data points where $w_\perp \gtrsim 20\,\mathrm{px}$, which apart from it being challenging for the tracker to catch up with the target is due to the measurement covariance being much smaller than the target size.

V. Discussion and Conclusions

Our results show that it is possible to use the event camera for UAV tracking in noisy environments, both when the target is clearly visible and when it covers only one or a few pixels. In case of simple target motion (*down_up*, *from_below* scenarios) at intermediate distances, the tracking is close to ideal under a Gaussian approximation, i.e. 68% of the tracks falling within 1σ and 95% within 2σ of the estimated position (this also validates said estimate in these cases). In some cases an even larger fraction is obtained. Here it should be noted that several of the data points were obtained from a very small sample. Even though the tracking was repeated, the data set is still the same, so the presented results may not be representative. Therefore, one should be careful about drawing any conclusions from this.

For some of the other measurements, it is evident that there are limitations with the tested method. The most critical one is that the target easily gets lost at close distances due to its large speed in the image plane, which is not ideal for surveillance purposes. Another limitation is poor tracking of small targets for very irregular movements at large background noise – a common trait of several of the horizontal measurements, and in particular the *irregular* measurements. However, one should keep in mind that this is a challenging tracking scenario. Moreover, the criteria for acceptable tracking performance are application dependent, so even limited tracking may be acceptable in some use cases.

An important reason for losing tracks at close distance is the one-model-fits-all approach that was adopted to be able to track UAVs in all scenarios, so in order to track irregular movements, a large process noise covariance was needed. This is not ideal, since it limits the reliance on the motion model(s) in the tracker, and hence the predictability of future target positions. Therefore, it cannot catch the fast accelerations of closer targets. The tracker also tends to catch smaller target candidates (background or split event clusters) close to the actual target, making the position estimate incorrect. It was noted that the fraction of measurements within the estimated covariance is not an ideal measure for targets larger than the covariance ellipse, simply because it may miss off-centre predictions within the target distribution. A better measure in this case is the GOSPA measure, but since this is given in pixels, a similar issue may be present here as well.

One way to overcome the false-target issue is to include the target size in the observation model, so that only clusters of similar sizes will be matched to the same track. Since the target size $w_\perp$ gives a measure of the distance, an extension to this idea is to insert this distance estimate into a full 3D tracker. This should make it possible to optimise the tracking parameters based on UAV motion parameters, which should reduce the need for finding a compromise that works for all target sizes in the image plane and could potentially improve the tracking. We hope to be able to present an implementation in a future study.

The method presented here is currently not real-time capable, partly due to it being implemented in MATLAB. For real-time applications, the tracker likely needs to be rebuilt in a faster computation environment. It could also be attempted to reduce the number of hypotheses N_{hyp} or the pruning depth

D, since the time complexity (in the worst case) scales as

$$t_{\text{exec}} \sim \mathcal{O}\left(N_{\text{hyp}}^{D}\right).$$

Another approach that could potentially improve performance, both in a tracking and an efficiency perspective, would be to attempt another association method, such as PMBM or PMHT. The former method is likely to worsen tracking speed, but given its success in other noisy tracking applications [26], it may improve the tracking performance also for us. PMHT is more computationally efficient, and has previously been used with promising results in event data tracking, although with more regularly moving targets [19].

This study is based on measurements at a single site without full control of all measurement parameters. Further tests are required to evaluate the method. In particular, greater control of the distance between the target and the sensor would allow for a more careful evaluation of how the distance to the target affects tracking. Tests in other measurement conditions are also needed to study the robustness of the tracker, as well as tests involving targets with other capabilities or multiple targets. It may also be possible to improve the denoising method by using similar light conditions during both calibration and measurements, e.g. by mounting a frosted glass plate in front of the sensor and thus only capturing background noise and not other targets such as insects. This would hopefully reduce the need for manually disabling hot pixels, but is not likely to significantly affect performance.

To conclude, we have shown that with good noise reduction, it is perfectly viable to track small UAV targets in a noisy environment over a large range of distances for a variety of motion scenarios using a single tracking model. However, to be able to cover the closest distances, noisiest sequences, and/or most irregular motions, further tuning or other observation or association models are still needed. If the tracking efficiency could be improved, our method is a promising candidate for tracking UAVs in real time for surveillance purposes.

REFERENCES

[1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 154–180, 2019.

[2] iniVation, "Understanding the performance of neuromorphic event-based vision sensors." [Online]. Available: https://inivation.com/wp-content/uploads/2020/05/White-Paper-May-2020.pdf

[3] iniVation, "Noise filters." [Online]. Available: https://inivation.gitlab.io/dv/dv-docs/docs/noise-filters/

[4] D. Czech and G. Orchard, "Evaluating noise filtering for event-based asynchronous change detection image sensors," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2016, pp. 19–24.

[5] Y. Feng, H. Lv, H. Liu, Y. Zhang, Y. Xiao, and C. Han, "Event density based denoising method for dynamic vision sensor," *Applied Sciences*, vol. 10, no. 6, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/6/2024

[6] A. Khodamoradi and R. Kastner, "$O(N)$-space spatiotemporal filter for reducing noise in neuromorphic vision sensors," *IEEE Transactions on Emerging Topics in Computing*, 2021.

[7] H. Liu, C. Brandli, C. Li, S.-C. Liu, and T. Delbrück, "Design of a spatiotemporal correlation filter for event-based sensors," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS 2015)*, vol. 2, 2015, pp. 722–725.

[8] J. Barrios-Avilés, A. Rosado-Muñoz, L. D. Medus, M. Bataller-Mompeán, and J. F. Guerrero-Martínez, "Less data same information for event-based sensors: A bioinspired filtering and data reduction algorithm," *Sensors (Basel)*, 2018.

[9] R. W. Baldwin, M. Almatrafi, V. Asari, and K. Hirakawa, "Event probability mask (EPM) and event denoising convolutional neural network (EDnCNN) for neuromorphic cameras," in *CVPR 2020*, 2020.

[10] Z. Wang, D. Yuan, Y. Ng, and R. Mahony, "A linear comb filter for event flicker removal," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 398–404.

[11] G. Im, K. Park, J. Kim, B. Son, S. Shin, and H. Lee, "Live demonstration: PINK: polarity-based anti-flicker for event cameras," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023, pp. 3901–3902.

[12] I. Alzugaray, "Event-driven feature detection and tracking for visual SLAM," Ph.D. dissertation, ETH Zürich, April 2022.

[13] A. Zihao Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] R. Hoffmann, D. Weikersdorfer, and J. Conradt, "Autonomous indoor exploration with an event-based visual SLAM system," in *2013 European Conference on Mobile Robots*, 2013, pp. 38–43.

[15] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1433–1450, 2021.

[16] C. Iaboni, H. Patel, D. Lobo, J.-W. Choi, and P. Abichandani, "Event camera based real-time detection and tracking of indoor ground robots," *IEEE Access*, vol. 9, pp. 166 588–166 602, 2021.

[17] C. Iaboni, D. Lobo, J.-W. Choi, and P. Abichandani, "Event-based motion capture system for online multi-quadrotor localization and tracking," *Sensors*, vol. 22, no. 9, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/9/3240

[18] F. Barranco, C. Fermuller, and E. Ros, "Real-time clustering and multi-target tracking using event-based sensors," 2018.

[19] B. Cheung, M. Rutten, S. Davey, and G. Cohen, "Probabilistic multi hypothesis tracker for an event based sensor," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 1–8.

[20] J. Rodríguez-Gomez, A. G. Eguíluz, J. Martínez-de Dios, and A. Ollero, "Asynchronous event-based clustering and tracking for intrusion monitoring in UAS," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8518–8524.

[21] J. P. Rodríguez-Gómez, A. G. Eguíluz, J. R. Martínez-De Dios, and A. Ollero, "Auto-tuned event-based perception scheme for intrusion monitoring with UAS," *IEEE Access*, vol. 9, pp. 44 840–44 854, 2021.

[22] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[23] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

[24] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1883–1901, 2018.

[25] R. L. Streit and T. E. Luginbuhl, "Probabilistic multi-hypothesis tracking," Naval Undersea Warfare Center Division, Newport, Rhode Island, Tech. Rep. NUWC-NPT Technical Report 10,428, February 1995.

[26] J. Smith, F. Particke, M. Hiller, and J. Thielecke, "Systematic analysis of the PMBM, PHD, JPDA and GNN multi-target tracking filters," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–8.

[27] iniVation, "Specifications - current models." [Online]. Available: https://inivation.com/wp-content/uploads/2023/11/2023-11-iniVation-devices-Specifications.pdf

[28] Teledyne FLIR, "BlackFly S USB 3." [Online]. Available: https://www.flir.eu/products/blackfly-s-usb3/?model=BFS-U3-23S3M-C

[29] Parrot, "Anafi white paper v1.4." [Online]. Available: https://www.parrot.com/us/drones/anafi/technical-specifications

[30] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[31] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *2017 20th International Conference on Information Fusion (Fusion)*, 2017, pp. 1–8.

[32] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Prentice-Hall, 1973, pp. 73–75.