Using neuromorphic cameras to track quadcopters

Terrence C. Stewart[1] Marc-Antoine Drouin[1] Michel Picard[1]
[1]National Research Council Canada
Ottawa & Waterloo, Ontario, Canada
FirstName.LastName@nrc-cnrc.gc.ca

Frank Billy Djupkep Dizeu[1] Antony Orth[1] , Guillaume Gagné[2]
[2]Defence Research and Development Canada
Quebec City, Quebec, Canada
guillaume.gagne2@forces.gc.ca

ABSTRACT

In recent work, we have shown that neuromorphic (event-based) cameras are highly efficient at detecting quadcopters. This is done by directly detecting the frequency of the rotating blades. This signal is highly characteristic of quadcopters, in that very few other real-world phenomena generate frequencies that are tightly clustered around a single peak. This makes this detection method highly robust to false positives, and can be generated with very little computational power. However, previous work in this direction has dealt only with detection of the presence of the drone. Here, we show that the same basic computations can also be used to localize the drone within the visual field of the camera. This allows for a system that not only alerts a user that a quadcopter is present, but also provides the extra information of where the drone is located.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; **Object recognition**; • **Hardware** → *Biology-related information processing*; • **Computer systems organization** → *Sensor networks.*

KEYWORDS

neuromorphic cameras, counter drone, quadcopters, Counter Uncrewed Aerial Systems (CUAS), tracking

ACM Reference Format:
Terrence C. Stewart[1] Marc-Antoine Drouin[1] Michel Picard[1] and Frank Billy Djupkep Dizeu[1] Antony Orth[1] , Guillaume Gagné[2] . 2023. Using neuromorphic cameras to track quadcopters. In *International Conference on Neuromorphic Systems (ICONS '23), August 1–3, 2023, Santa Fe, NM, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3589737.3605987

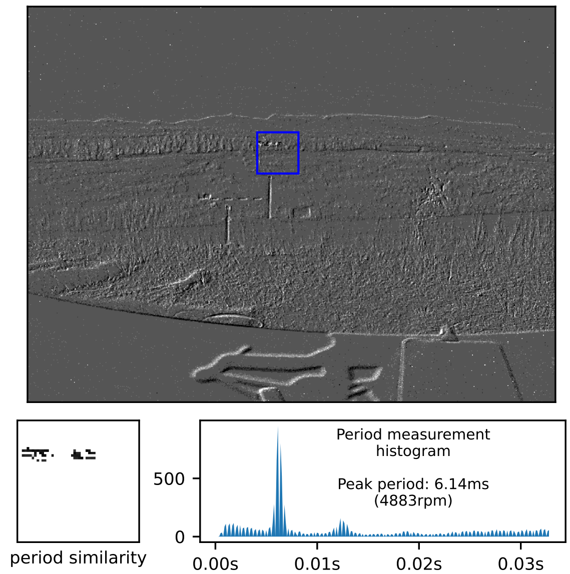

Figure 1: Tracking the physical location of a quadcopter drone, shown as a blue box. Top: Event camera raw data, binned over the previous 0.1 seconds. Left: Period similarity image, indicating how close the current estimate of the period at each pixel is to the target period p_{ideal}. Left: a histogram of period measurements over the entire visual field for the previous 0.5 seconds.

1 INTRODUCTION

Given the recent proliferation of small Uncrewed Aerial Systems (UAS), especially quadcopters, there is a strong desire to be able to detect them. This detection may be for privacy or security reasons, and is important for commercial, government, and military users. It is also of pressing importance given that quadcopters are now being weaponized [4] .

A wide variety of methods exist for detecting quadcopters, including radar, audio, radio frequency, visible-light cameras, and infrared cameras [3]. In our recent work [7][6] we have demonstrated that event cameras can be used to directly measure the high frequency of rotating blades (1000 to 6000 rpm for NATO Class 1 drones). By measuring this spectral signature we are detecting a physical signal that is core to the functionality of quadcopters (rotating blades to generate lift), leading to a robust detection method. Furthermore, by using event cameras to detect these frequencies, the computation can be done extremely efficiently, as compared to similar approaches with high-frame rate standard cameras[2]. In our demonstrator system, we achieved detection in approximately 5 Watts using a Rasperry Pi 4 and a DVXplorer event camera[6].

In this work, we extend our implementation to also detecting the physical location of the quadcopter in the visual field. We do this

ICONS '23, August 1–3, 2023, Santa Fe, NM, USA

ACM ISBN 979-8-4007-0175-7/23/08.
https://doi.org/10.1145/3589737.3605987

by directly applying the existing algorithm used for tracking high-frequency LEDs[5][1]. Since this is related to the algorithm already being used for computing the spectral signature of the quadcopter, this requires minimal additional computational overhead.

2 SPECTRAL SIGNATURE

In [5] and [1], a method was proposed for using an event camera to track the position of a high-frequency LED. In an event camera, data is provided as a series of *events*, where each event is the tuple $< x, y, t, sign >$, giving the x-position, the y-position, a timestamp, and whether that pixel got brighter (ON) or darker (OFF). The core idea to use this for tracking was to note that the events at any given pixel can always be thought of as a block of one or more ON events, followed by one or more OFF events, followed by one or more ON events, and so on. Thus, to get an estimate of the oscillation period at any given pixel, we simply need to subtract the timestamp of the first event in one block of ON events from the timestamp of the first event in the next block of ON events (and the same can be done for OFF events). Now, instead of the raw events $< x, y, t, sign >$, we have a sequence of what we call *measurements* $< x, y, p >$, where p is the period measurement (i.e. the difference between the two timestamps).

In our previous work on drone *detection*, we used these measurements to build up an overall histogram of such measurements over the entire visual field. This explicitly ignores the $< x, y >$ location of the measurements, and instead builds an overall spectral signature of the drone. The bottom-right of Figure 1 shows such a histogram, generated from the previous 0.5 seconds of measurements. The histogram has 256 bins, and each bin is 128 microseconds wide. Any measurements of periods outside these bins (greater than 32,768 microseconds) are discarded.

In the presence of a quadcopter, this histogram shows a direct physical signature of a quadcopter due the presence of the rotating blades. Furthermore, we have found this signature to be very robust to false positives, and developed a simple (100-neuron) neural network to detect the presence of a drone given this histogram as input[6]. An interesting and consistent feature of the histogram is the subharmonic peaks. In Figure 1 the main peak is visible at 6.14ms, but smaller peaks can be seen at 12.28ms and 18.42ms. This is due to the event camera missing some ON → OFF and OFF → ON transitions, and we have used this consistent signature as part of our detection algorithms[7][6].

3 TRACKING

In order to create the histograms shown in the previous section, we collected measurements by taking the difference in time between two ON (or OFF) events. This generated a period measurement at that pixel. In the original work we based our algorithm on[5][1], these measurements were not combined into a histogram; rather, they were used to update an estimated location of a blinking LED with known frequency. The algorithm is as follows (adapted from[5]).

First, we initialize our estimated location $< \tilde{x}, \tilde{y} >$ to the centre of our visual field. We do this in pixel coordinates, and for the DVXplorer this is $< 320, 240 >$.

Next, every time there is a new measurement of the period (i.e. every time we add a value into the histogram), we have the pixel location $< x, y >$ and the measured period ($p = t_{now} - t_{prev}$). This allows us to compute two difference measures: the difference in space (pixels) and the difference in time (period). For the difference in space, we do the Euclidean distance between the current event $< x, y >$ and our current location estimate $< \tilde{x}, \tilde{y} >$.

$$d_s^2 = (x - \tilde{x})^2 + (y - \tilde{y})^2 \tag{1}$$

For the difference in time, we start by noting that the period measured by the event camera exhibits a specific pattern of sub-harmonics. Three such peaks can be in the histogram in Figure 1. As we have discussed in previous work [7], if the actual period is 0.006s, the values measured by our algorithm may be integer multiples of that period (0.0012s, 0.0018s, etc.), as the event camera may miss one or more of the ON → OFF (or OFF → ON) transitions. Thus to compare the measured period to a target ideal period p_{ideal} we can compute the minimum distance for different scaled versions of the measured period. For the purposes of this paper, we only use the first term, but future work will explore the efficiency/accuracy tradeoffs of using more terms, and introducing a scaling factor on w based on which sub-harmonic is used. Furthermore, similarly to [7], this could be further improved by online estimation of the probability of missing ON → OFF transitions, based on the relative heights of the peaks in the histogram.

$$d_p = \min(p - p_{ideal}, \frac{p}{2} - p_{ideal}, \frac{p}{3} - p_{ideal}, ...) \tag{2}$$

Given these two difference measures, we apply two Gaussians to produce a weighting factor w which indicates how similar the measurement is to the measurements we would expect. That is, the tracking algorithm should pay more attention to measurements that are near where the drone is currently believed to be, and are of frequencies similar to that expected.

$$w = w_{max} e^{-d_d^2/\sigma_d^2} e^{-d_p^2/\sigma_p^2} \tag{3}$$

Finally, the tracked position is moved closer to the measurement position based on this weighting w.

$$\begin{aligned} \tilde{x} &\leftarrow \tilde{x} + w(x - \tilde{x}) \\ \tilde{y} &\leftarrow \tilde{y} + w(y - \tilde{y}) \end{aligned} \tag{4}$$

It should be noted that this algorithm is identical to that presented in [5], except we multiply the Gaussians in Eq. 3, rather than adding them. Both approaches should produce similar results, but we have not yet rigorously analyzed the effect of this difference.

In addition, we note that a similar algorithm can be used to compute the estimated period $\tilde{p}$.

$$\tilde{p} \leftarrow \tilde{p} + w(p - \tilde{p}) \tag{5}$$

We initialize $\tilde{p}$ using the value of p_{ideal}. The estimated period could then be used to examine the variation of rotation rate of the propellers, although that is not further explored here.

Furthermore, as slight optimizations, we only perform this computation for measurements that produce periods that are within our histogram (i.e. very long period measurements are ignored). These measurements will produce small w values in any case. Also, if w is very small (< 0.00001), we do not perform the target update (Eq. 4).

The result is a tracking algorithm with four free parameters: σ_d (the width of the Gaussian for space), σ_p (the width of the Gaussian for the period measurements), w_{max} (the maximum amount to update the target position based on a single measurement) and p_{ideal} (the centre of the Gaussian for the period measurements). The value for p_{ideal} can be set based on typical quadcopters, but to set the other values we collected a data set and optimized for algorithm performance, as discussed in the next section.

4 GROUND TRUTH

To optimize the parameters of our tracking system, and to evaluate its overall performance, we need ground-truth data for comparison. Given the complexities involved in calibrating an separate off-the-shelf tracking system to the optics of the event camera and its motion during recording, we chose to manually label the position data for a set of event recordings. These recordings consisted of 200 seconds of footage of quadcopters, including entering/exiting the camera field of view, moving away from the camera until they are not observable, hovering against a forest background while the camera pans, flying over a camera pointed upwards, and being recorded from an event camera mounted on another moving quadcopter.

To hand-label this data set, we developed a software tool that presents two views of the event data. First, we have a relatively standard depiction of the event-camera data: a grayscale image where each pixel has its value increased when an ON event occurs, and decreased when an OFF event occurs. We also apply an exponential decay with a time constant of 0.01 seconds. This is a low-pass-filtered version of the raw event data itself. This approach produces images that are similar to that seen for a more traditional binning approach (i.e. generating an image based on all the events occurring within a particular time window) but is more continuous in time.

To accompany this image, we also produced an image showing how drone-like each pixel in the image is. We compute this by keeping track of the most recent p value (the period measurement) for each pixel. In addition, we track when in time that period measurement occurred (t_p). The current best estimate of the period of a particular pixel is then computed as $p_e = \max(p, t - t_p)$ where t is the current time. That is, if a period of 0.1s was measured 0.05 seconds ago, then 0.1 is still the best estimate for the current period. But, if a period of 0.1s was measured 0.2 seconds ago, then we know the current period is at least 0.2 seconds, since there hasn't been a full ON-OFF-ON cycle in the last 0.2 seconds.

Given these period estimates p_e at each pixel, we then produce a grayscale image where the value is based on the similarity between the period estimate p_e and the target period p_{ideal}. As with the tracking algorithm (Eq. 3), we use a Gaussian $e^{-(p_e - p_{ideal})^2/\sigma^2}$ for this. The result is an image that is mostly blank, except for pixels that have periods similar to that of a drone. With the combination of these two images, we found that the position of the drone was relatively unambiguous to the human eye.

Our software interface shows both images, as well as an initial tracking guess from our tracking algorithm. This can be seen in Figure 1. The user then adjusts this guess by moving a bounding box, including both position and size. We label the data at 0.1 second intervals.

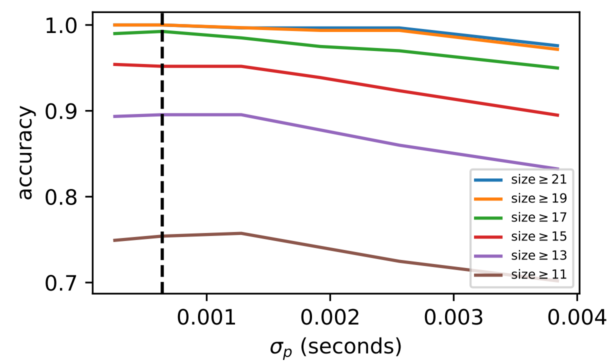

Figure 2: Accuracy of tracking for different size thresholds as σ_p is adjusted. Size is the minimum of the width and height of the labelled bounding box. Tracking is accurate if the Euclidian distance between the tracking algorithm and the true position is less than half the size. Thus, any accurate tracking result is inside the labelled bounding box. Vertical dotted line indicates chosen best parameter setting.

It should be noted that, for the current system, we are not using the fact that we are labelling the size of the drone (width and height) as well as its position. We are recording this data partly for statistics (determining the accuracy of the tracking algorithm at different drone sizes) and for future development.

5 RESULTS

We use two metrics for evaluating our tracking system. First, we measure *accuracy*, defined as whether or not the tracked point $< \tilde{x}, \tilde{y} >$ is inside the hand-labelled bounding box for the quadcopter. Second, we measure the *relative error*, which is the Euclidean distance between the tracked point and the centre of the bounding box, divided by the size of the bounding box, which we define to be the minimum of the width and the height. With this definition, note that any relative error less than 0.5 is guaranteed to be inside the bounding box (i.e. accurate).

We used grid search to find parameter settings for σ_d, σ_p, and w_{max}. Figure 2 provides a sensitivity analysis of σ_p. We measure accuracy at different thresholds of size, as we find accuracy is reduced when drones take up fewer pixels on the camera. For our final parameter settings, the drones larger than 15 pixels (width or height) are tracked to within their bounding box 95.2% of the time, and 99.2% for drones larger than 18 pixels.

A depiction of the performance of the of the tracking is given in Figure 3. Here we see a tight fit between the tracking (solid line) and the ground truth (dashed line) for both x and y values, except when the drone is smaller than 15 pixels (horizontal line at the top of the graph).

Finally, Figure 4 provides a scatterplot of drone size plotted against the relative error. Relative error values below the horizontal line at 0.5 guarantee that the tracked point is inside the drone's

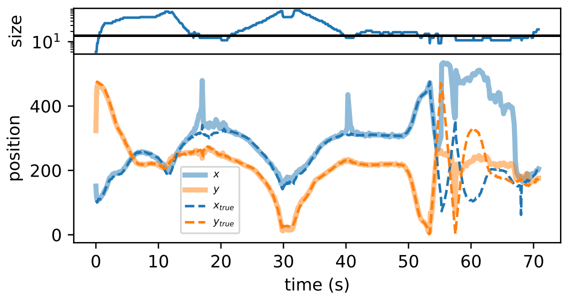

Figure 3: Tracking data for one test sequence, showing only time points when a drone is present. Size of the drone (the smaller of width or height) is shown at the top. Note that tracking is accurate as long as the drone is larger than 15 pixels (horizontal line).

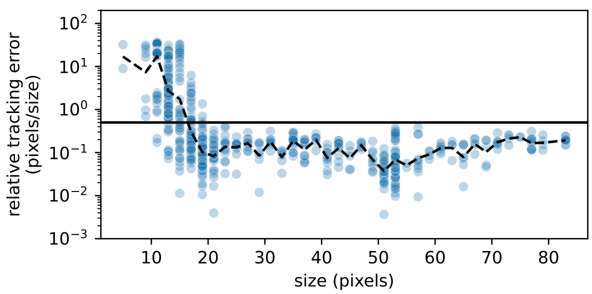

Figure 4: Tracking accuracy at different pixel sizes. Size is the minimum of the width and height of the hand-labelled bounding box. Relative error below 0.5 (horizontal line) indicates the prediction is inside the bounding box. The scatterplot shows individual measurements and the dotted line is the geometric mean for each size.

bounding box. Again, for drones larger than 15 pixels we have consistent high accuracy, but error increases significantly below this point.

In Figure 5, we show two more examples of drone tracking, in addition to the one given in Figure 1. In each case, note that the period similarity plot (bottom left), where darker shading indicates a larger value of $e^{-(p_e - p_{ideal})^2/\sigma^2}$, provides a clear and consistent marker for the presence of a quadcopter across conditions, including being almost directly in front of the sun (last example in Figure 5).

6 CONCLUSIONS AND FUTURE WORK

We applied the computationally efficient tracking algorithm from [5] to the problem of tracking quadcopters using an event camera. On our dataset we were able to track a quadcopter to within its bounding box 95.2% of the time if it was larger than 15 pixels (in width or height), and 99.2% of the time if it was larger than 18 pixels. This forms a useful supplement to our current detection system[6] without incurring significant computational overhead.

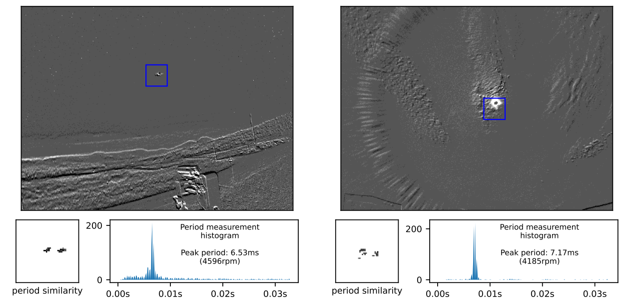

Figure 5: Two examples of drone tracking, shown as a blue box centred at $< \tilde{x}, \tilde{y} >$. For each example, we show the binned event data (top), the similarity of the most recent period estimate p_e at each pixel with the ideal drone period p_{ideal} (left), and the histogram of period measurements over the entire visual field for the previous 0.5 seconds (right).

Our current efforts are to further develop this approach. One primary goal improvement needed is to support multiple quadcopters present in the visual scene. Furthermore, we note that the tracking algorithm itself does have implicit information about the possible presence of the drone, so the state of the tracking algorithm (in particular, the w value) could also be used as input to the drone detection system[6]. Other improvements include using non-Gaussian distributions for the different in period measurement in Eq. 3, as well as exploring different ways of combining the two weighting factors (addition vs multiplication).

Finally, while the emphasis of our approach is to provide highly energy-efficient drone detection and tracking solutions, we are also looking into other approaches for comparison. In particular, we believe that the drone similarity image that we developed for labelling our data could also be used as the input to a YOLO-style object identification and localization system. This method for pre-processing the data may improve performance of such a network over and above the standard image approach. With modern neuromorphic computing hardware it may be possible to build an energy-efficient version of such a network, if it proves to be more accurate than the algorithm we have presented here.

REFERENCES

[1] Andrea Censi, Jonas Strubel, Christian Brandli, Tobi Delbruck, and Davide Scaramuzza. 2013. Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Tokyo, Japan, 891–898. http://purl.org/censi/2013/dvs
[2] Frank Billy Djupkep Dizeu, Michel Picard, Marc-Antoine Drouin, and Guillaume GAGNé. 2022. Extracting Unambiguous Drone Signature Using High-Speed Camera. *IEEE Access* 10 (2022), 45317–45336. https://doi.org/10.1109/ACCESS.2022.3170481
[3] Jimmy Flórez, José Ortega, Andrés Betancourt, Andrés García, Marlon Bedoya, and Juan S. Botero. 2020. A review of algorithms, methods, and techniques for detecting UAVs and UAS using audio, radiofrequency, and video applications. *TecnoLógicas* 23 (2020), 269–285.
[4] T. Gibbons-Neff and N. Yermak. January 7, 2023. Accessed on April 24, 2023. In a Ukraine Workshop, the Quest to Build the Perfect Grenade. *New York Times* (January 7, 2023. Accessed on April 24, 2023). https://www.nytimes.com/2023/01/07/world/europe/ukraine-russia-war-grenade.html
[5] Georg Muller and Jorg Conradt. 2011. A miniature low-power sensor system for real time 2D visual tracking of LED markers. 2429–2434.
[6] Terrence Stewart, Marc-Antoine Drouin, Michel Picard, Frank Billy Djupkep Dizeu, Anthony Orth, and Guillaume Gagné. 2022. A Virtual Fence for Drones: Efficiently

Detecting Propeller Blades with a DVXplorer Event Camera. In *Proceedings of the International Conference on Neuromorphic Systems 2022* (Knoxville, TN, USA) *(ICONS '22)*. Association for Computing Machinery, New York, NY, USA, Article 10, 7 pages. https://doi.org/10.1145/3546790.3546800

[7] Terrence C. Stewart, Marc-Antoine Drouin, Guillaume Gagné, and Guy Godin. 2021. Drone Virtual Fence Using a Neuromorphic Camera. *ICONS 2021: International Conference on Neuromorphic Systems 2021* (2021), 1–9.