2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

A Linear Comb Filter for Event Flicker Removal

Ziwei Wang, Dingran Yuan, Yonhon Ng, and Robert Mahony

Abstract— **Event cameras are bio-inspired sensors that capture per-pixel asynchronous intensity change rather than the synchronous absolute intensity frames captured by a classical camera sensor. Such cameras are ideal for robotics applications since they have high temporal resolution, high dynamic range and low latency. However, due to their high temporal resolution, event cameras are particularly sensitive to flicker such as from fluorescent or LED lights. During every cycle from bright to dark, pixels that image a flickering light source generate many events that provide little or no useful information for a robot, swamping the useful data in the scene. In this paper, we propose a novel linear filter to preprocess event data to remove unwanted flicker events from an event stream. The proposed algorithm achieves over 4.6 times relative improvement in the signal-to-noise ratio when compared to the raw event stream due to the effective removal of flicker from fluorescent lighting. Thus, it is ideally suited to robotics applications that operate in indoor settings or scenes illuminated by flickering light sources.**

Code, Datasets and Video:

https://github.com/ziweiWWANG/EFR

I. INTRODUCTION

Event cameras, such as the Dynamic Vision Sensor (DVS) [1], are bio-inspired vision sensors that record per-pixel brightness changes. In contrast to conventional cameras that accumulate average brightness for every pixel over the exposure time to generate synchronous intensity frames, event cameras output a sparse asynchronous stream of positive and negative event data only when the change in logarithmic brightness at individual pixels exceeds a contrast threshold. The high dynamic range (HDR), fine temporal resolution, low latency (<0.5 ms) and sparse output [2] make event cameras ideal for high-speed robotics vision applications that require low computational cost and operate in challenging lighting conditions, such as feature tracking [3], [4], [5] de-blurring [6], [7], motion estimation [8], autonomous-driving applications [9], 3D reconstruction [10], [11], [12], visual odometry and simultaneous localization and mapping (SLAM) [13], [14], [15]. However, these same properties make event cameras highly susceptible to flicker, particularly those caused by fluorescent and LED light sources, which can swamp the signal of interest in the scene.

Event data de-noising is common in all current event cameras. Current de-noising methods focus primarily on uncorrelated background activity noise [17], [18], [19], [20], [21], [22], [23]. This noise is commonly caused by transistor switch leakage or thermal noise [17], [20] and is present even when there is no real brightness changes [20]. Flicker removal is an entirely different problem since it corresponds to real intensity changes in the image and is only considered 'noise' since flicker is uninteresting and distracting to most robotic systems. However, in a typical indoor or nighttime environment with fluorescent or LED light sources, the events associated with the flicker of the lighting can overwhelm other events in the scene. In the datasets collected in this paper, with only a single fluorescent light source, the unwanted events associated with the flickering light source outnumbered interesting foreground events and occupied 84% of the event data on average. This creates many problems in real-world robotics systems. Excessive events due to flicker can overwhelm the onboard data bus leading to the lost events or requiring the camera to be set at lower sensitivity (leading to fewer events overall), resulting in a bigger quantisation error in the desired foreground information. Modern image reconstruction algorithms often rely on batching events [16], [24], [7] and excessive unwanted flicker can dominate the data in a batch and lead to poor reconstructions of foreground activity. Flicker is also a major issue in optical flow and other motion estimation methods that depend on image consistency. Real-world flickering light sources are complex to model. Rarely are they perfect sinusoidal signals due to the electronics that are used to

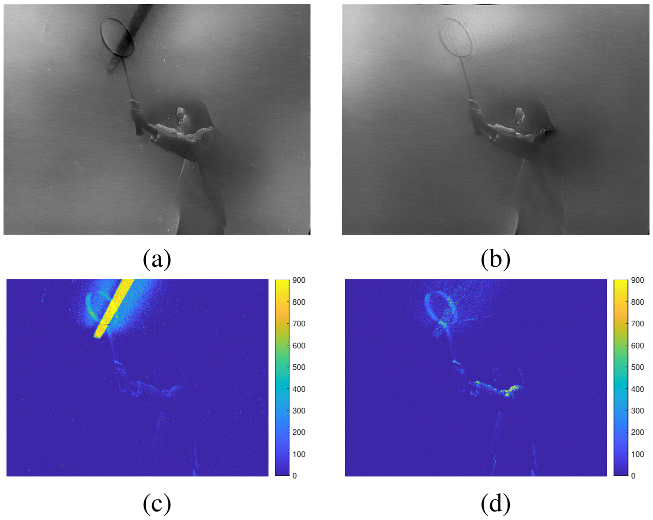
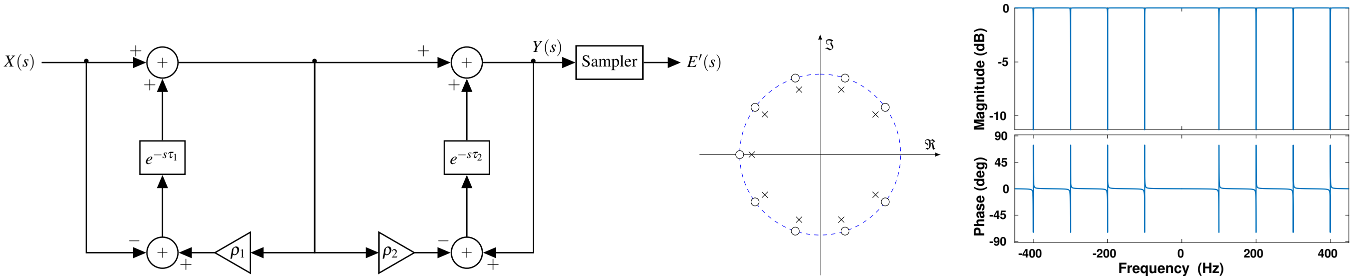


Fig. 1. (a) Image reconstruction E2VID [16] from a high-speed dataset with a flickering (fluorescent) light source in the background. (b) Image reconstruction E2VID [16] after event flicker removal using the proposed algorithm. (c) The heat map of events per second from the unfiltered event stream. (d) The heat map of events per second after event flicker removal.

[1] Ziwei Wang, Dingran Yuan, Yonhon Ng, and Robert Mahony are from Systems Theory and Robotics Group, College of Engineering and Computer Science, The Australian National University. ziwei.wang1@anu.edu.au

978-1-7281-9680-0/22/$31.00 ©2022 IEEE



(a) Block diagram (b) Pole-zero plot (c) Bode diagram

Fig. 2. Our proposed event-based flicker removal algorithm.

generate the driving current and the frequency response usually has many strong harmonics [25].

In this paper, we propose an algorithm for event flicker removal. To the best of the authors' knowledge, this is the first such algorithm that has been published. A key contribution is the recognition that flicker events are highly non-sinusoidal and cannot be removed using a finite-dimensional notch filter. This leads us to consider linear comb filters that attenuate all harmonics of a given frequency at the same time by feeding back a time-delayed version of the original signal. We use both delayed feedback and delayed feed-forward [26], [27] to design a linear filter response that is effective at passing the DC component of a signal while strongly attenuating all harmonics of a desired base frequency. Since the design is based on a linear filter, there is a short transient while the internal filter state converges to attenuate the undesired signal. The resulting algorithm is highly effective in removing flicker from event streams. The algorithm is suitable for integration into a Field-Programmable Gate Array (FPGA) or Application-Specific Integrated Circuit (ASIC), allowing the algorithm to be built into the camera hardware. In the present paper we provide only a demonstration implementation in C++ to demonstrate its potential. In summary,

- We propose a modified linear comb filter for event flicker removal. The algorithm allows parallel pixel-wise computation and is suitable for implementation on low-latency and low-power hardware circuits, such as FPGA and ASIC.
- We demonstrate event flicker removal performance in high frame rate and fast motion video reconstruction in the presence of fluorescent lighting.

II. Related Works

Since event cameras are prone to noise [1] many event data de-noising algorithms have been used in image reconstruction and de-blurring algorithms. For example, noise in event data can be modelled by estimating event data bias and contrast threshold to match event data to reference intensity images [18], [23], [28]. To avoid accumulating event bias over time, the Complementary Filter (CF) [29] exponentially 'forgets' historical event data using a single filter gain parameter. Pan *et al.* [6] estimates event camera contrast threshold by minimising energy function of image sharpness. Some neural networks [16], [24], [30], [7] rely on regularisation and spatial priors to reduce artefacts caused by event data noise.

In addition to estimating event sensitivity and priors, event filtering methods can remove noisy events as a preprocessing step for other algorithms. Delbruck [17] firstly introduced an effective background activity filter that directly removes isolated events with a low spatial-temporal correlation between the nearest neighbouring event data. The algorithm has been implemented in the event-based sensor software jAER [31]. Ieng *et al.* [19] reduced noise by spatial-temporal filtering event data asynchronously and Liu *et al.* [20] implemented a background activity filter in hardware that filters out 98% background noise with 10% loss of real data. Later, Padala *et al.* [22] achieved higher signal to noise ratio and Khodamoradi *et al.* [21] reduced memory complexity of the background activity filter from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. Apart from the correlation with the nearest neighbour, event data noise was also modelled based on differing polarity, refractory period and inter spike interval [32], [28]. Recently, Wang *et al.* [33] bridged event and image data via a noise-robust motion compensation model and proposed a joint filtering method.

To the authors' knowledge, there is no published algorithm that is targeted at attenuation of flicker events in event streams. In contrast, flicker removal is a well-studied field for conventional cameras. For conventional images, flicker is modelled as a dramatic intensity change between two neighbouring frames [34] rather than at a pixel level. Chono *et al.* [35] proposed a quantisation-based method to insert a periodically intra-coded pictures derived from the previous predictive-coded pictures, that suppresses the effect of flickering. Guthier *et al.* [36] proposed an iterative flicker removal algorithm that uses an average intensity change threshold to detect flicker and an iterative intensity scaling to compensate until variation falls below the threshold. Shiau *et al.* [34] designed an adjuster to the video frame, which estimates the atmospheric light of each frame, and assigned a multiplier based on that weight. Many modern video cameras come with 50 or 60Hz de-flicker as a standard functionality.

III. Method

In this section we present the mathematical model of our event-based flicker removal algorithm.

A. Mathematical Model

1) Event Camera Model: As an asynchronous image sensor, the output of an event camera follows the Address Event Representation (AER) [37]. The intensity change of the i^{th} event e_i is encoded in a vector $[t_i, p_i^x, p_i^y, \sigma_i]^T$ with event timestamp t_i, pixel coordinate $p_i = (p_i^x, p_i^y)$, and polarity σ_i for positive or negative direction of the log-intensity change. We model a single event as an impulse signal δ in continuous time

$$e_i(t) = \sigma_i c^{p_i} \delta(t - t_i), \; i \in 1, 2, 3, \ldots \tag{1}$$

where c^{p_i} is the contrast threshold at pixel p_i. An event stream E for asynchronous filter implementations is defined as

$$E(p,t) = \sum_{i=1}^{\infty} e_i(t) \delta_p(p_i) = \sum_{i=1}^{\infty} \sigma_i^p c^p \delta(t - t_i^p) \delta_p(p_i). \tag{2}$$

where p is the pixel location considered and t is the time index and $\delta_p(p_i)$ is the Kronecker delta function that is unity when $p = p_i$ and zero otherwise [38].

2) Flickering and Harmonics: The high temporal resolution property enables event cameras to capture events that encode high-frequency flickering intensity as a periodic waveform. Reproducing the complex waveform of a light source requires a large number of events per second even though the light source appears to be of a constant brightness for a typical frame-based camera and human eye. Fluorescent and LED lights under Alternating Current (AC) power source can be modelled as a periodic signal over time t. Expanding the signal in Fourier series yields

$$f(t) = \sum_{k=0}^{\infty} a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t), \tag{3}$$

for Fourier coefficients a_k and b_k. Due to the electronics in fluorescent and LED lights the higher order coefficients are generally quite large and the signal cannot be attenuated by a simple notch filter at the base frequency. The power spectral density of the signal (3) consists of an infinite sum of harmonics at multiples $k\omega_0$ of the base frequency.

B. Proposed Comb Filters

1) Feed-forward Comb Filter: The basic feed-forward comb filter [39] subtracts a time-delayed version of the signal to itself (see Fig. 3). If the time delay corresponds exactly to the period $\tau_1 = 2\pi/\omega_0$ of the flicker, then the resulting summed signal attenuates the flicker. The corresponding system block diagram is shown in Fig. 3(a). The bode plot shown in Fig. 3(b) displays periodic notches to the input signal corresponding not only to the base frequency $f_0 = \omega_0/2\pi$, but also to the harmonics of the flicker frequency. The infinite harmonic notch filter is characteristic of the time delay action; the cancellation of periodic signals works equally well at multiple cycles of the base period $\tau_1 = \frac{2\pi}{k\omega_0}$ as it does for $k = 1$. This infinite-dimensional nature of the comb filter is ideal for the suppression of harmonic noise characteristic of flicker in event data.

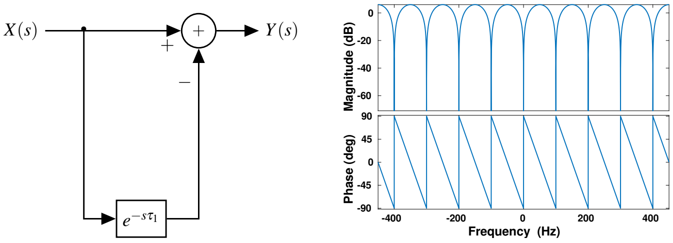
The Laplace transform of the feed-forward comb filter is given by [39]

$$Y(s) = (1 - e^{-s\tau_1}) X(s) \tag{4}$$

The magnitude of the impulse response is computed by substituting complex frequency variable s with $j\omega$

$$\begin{aligned} |H(j\omega)| &= |(1 - \cos(\omega\tau_1) + \sin(\omega\tau_1) j| \\ &= \sqrt{2 - 2\cos(\omega\tau_1)}. \end{aligned} \tag{5}$$

However, as seen from the bode plot in Fig. 3(b), there is a significant magnitude and phase distortion around the notches. The basic feed-forward comb filter also attenuates the DC component of the signal. In the proposed application this will annihilate static scene information and would compromise the performance of the filter.



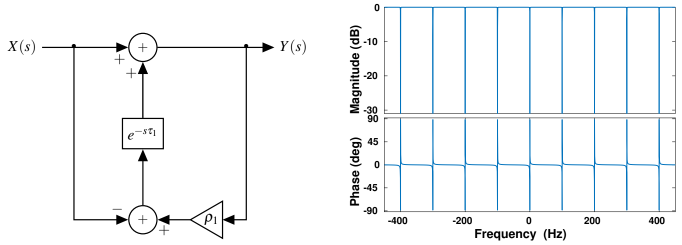
(a) (b)

Fig. 3. (a) Block diagram of the feed-forward comb filter (b) Bode plot of the feed-forward comb filter. The base frequency $f_0 = 1/\tau_1$ is set to 100Hz.

2) Distortion attenuation: To solve the magnitude distortion problem of the feed-forward comb filter [26], we introduce a scaled positive feedback signal in addition to the feed-forward signal as shown in Fig. 4. The feed-forward comb filter can be thought of as a transfer function with an infinite set of zeros corresponding to the expansion of the exponential function. The feedback loop introduces a corresponding infinite set of poles that are frequency aligned but have a slightly lower amplitude. This introduces a near pole-zero cancellation of the filter response localising the frequency effect of the comb filter. The transfer function for the feed-forward/feedback comb filter is

$$Y(s) = \frac{1 - e^{-s\tau_1}}{1 - \rho_1 e^{-s\tau_1}} X(s), \tag{6}$$

where $0 < \rho_1 < 1$ is a tuning parameter. The parameter ρ_1 governs how closely the pole cancels the corresponding zero and determines the depth and width of each notch. A higher ρ_1 value localises the filter effect close to the notch frequency and ensures that other frequencies are not distorted, while a lower ρ_1 value widens both the notch and the associated frequency distortion up to the limit $\rho_1 = 0$, where the basic feed-forward comb filter is recovered. The corresponding bode plot is shown in Fig. 4(b).



(a) (b)

Fig. 4. (a) Block diagram of the feed-forward/feedback comb filter (b) Bode plot of the feed-forward/feedback comb filter. ρ_1 can be set in range $0 < \rho_1 < 1$. We set $\rho_1 = 0.6$ in this paper.

The combination of a feed-forward and feedback filter provides an infinite sequence of notch filters at all harmonics of a base frequency with a tunable factor ρ_1 that allows the notch to be widened at the cost of increased frequency distortion. The remaining difficulty is the fact that the feed-forward loop still cancels the DC response of the base signal.

3) Proposed event-based flicker removal algorithm: The proposed filter (Figure 2) uses a reciprocal comb filter [27] (in resonant mode with gain on the feed-forward path allowing the feedback to dominant and emphasise flicker) at a much higher frequency with a smaller time-constant $\tau_2 << \tau_1$ to introduce a cancellation of the DC pole-zero notch filter. The Laplace transform of the filter is

$$Y(s) = \frac{1-e^{-s\tau_1}}{1-\rho_1 \cdot e^{-s\tau_1}} \cdot \frac{1-\rho_2 e^{-s\tau_2}}{1-e^{-s\tau_2}} X(s), \tag{7}$$

where the gain ρ_2 serves the same purpose as ρ_1 in (6). The time constant τ_2 must be chosen significantly smaller than τ_1, in our work we typically use $\tau_2 = \tau_1/10$ and tune the gain ρ_2 to satisfy $\tau_2(1-\rho_1) = \tau_1(1-\rho_2)$ to ensure that the decade slope of the resonant mode matches the attenuation of the DC notch that needs to be cancelled. The pole-zero plot in Fig. 2 provides a visualisation of the resulting pole zero cancellations. Fig. 2(c) shows the bode plot of our proposed filter with the desired notch filters at all harmonics with the DC component unaffected.

The time-domain solution of (7) can be written as

$$\begin{aligned} y(t) &= x(t) - x(t-\tau_1) - \rho_2 \cdot x(t-\tau_2) + \rho_2 \cdot x(t-(\tau_1+\tau_2)) \\ &\quad + \rho_1 \cdot y(t-\tau_1) + y(t-\tau_2) - \rho_1 \cdot y(t-(\tau_1+\tau_2)). \end{aligned} \tag{8}$$

This is a causal function that only depends on the present and past value of input x and output y. The resulting filter can be implemented by processing incoming events and keeping circular queues of events to implement the time-delays.

An event sampler is applied at the output to regenerate an event stream. This sampler keeps a reference value of intensity and compares the output to the reference generating an event and resetting the reference when the filter output change is above the threshold. The input signal to this sampler is a zero-order hold signal and the threshold need only be checked when an event occurs, either in the input signal or from one of the queues that instantiates the time delay. This can be implemented in digital logic.

IV. Experiment

A. Implementation details

The filter is implemented continuously using C++. It takes around 0.1 seconds to converge for a 50Hz base frequency (see Fig. 7) corresponding to around 3-5 cycles of the flicker. The comb filter parameters are set to: $\rho_1 = 0.6$, $\rho_2 = 1-(1-\rho_1)/10$, $\tau_1 = 1/50$ for 50Hz base frequency and $\tau_2 = \tau_1/10$.

B. Datasets

To the best of the authors' knowledge, there is no existing open-source datasets and algorithm available targeted at the event camera flicker removal problem. To fill the gap, we collected a set of real data under a flickering fluorescent light using a *Prophesee Gen3* event camera (resolution 480 × 640). Our datasets include fast motions of a basketball, table tennis, badminton and Nerf gun. The flickering light source in the datasets is supplied by 50Hz alternating current in Australia, but the flickering primarily occurs at 100Hz because older fluorescent lights with magnetic ballasts flicker at twice the supply frequency [40] and 50Hz flickering mainly occurs at the two edges of the fluorescent light.

C. Main results

1) Video reconstruction and heat map of event rate: We evaluate the raw and filtered event data using the state-of-the-art pure event-based video reconstruction algorithm E2VID [16] in Fig. 5. The flickering light sources are obvious in the E2VID [16] reconstructed images using raw event data in Fig. 5(a). After applying our event-based flicker removal (EFR) algorithm, the flashing lights in reconstructed images are clearly attenuated in Fig. 5(b). Please see video materials for better visualisation of the flicker removal performance.

We also provide heat map analysis of event rate to visualise the distribution of the raw event data and the filter event data in Fig. 5(c)-(d). We accumulate event data during a 0.03-second time window for both raw event data and filtered event data. The event rate is modelled by the ratio of triggered event number to the time period. The event rate from 0 to the maxima is coloured from blue to yellow in Fig. 5. In a raw event stream, the number of flicker events is on average 84% of the total number of events within the image containing a flickering light source. In Fig. 5(c), it is clear that most events are triggered by the flickering fluorescent light, leading to the constantly flickering video reconstructions. The flicker events contain little or no useful information but occupy most data memory and computational resources, while the actual useful events are triggered at a very low event rate in comparison. Using our event-based flicker removal algorithm, most of the flicker events are filtered out in Fig. 5(d), while preserving useful event data. Interestingly, objects across the flickering source have minimal impact on our flicker removal algorithm. For example, when the table tennis ball in Fig. 5(b) and the racket in Fig. 1(b) are moving across the flickering fluorescent light, our filter is able to attenuate flickering events, leading to high-quality video reconstruction performance.

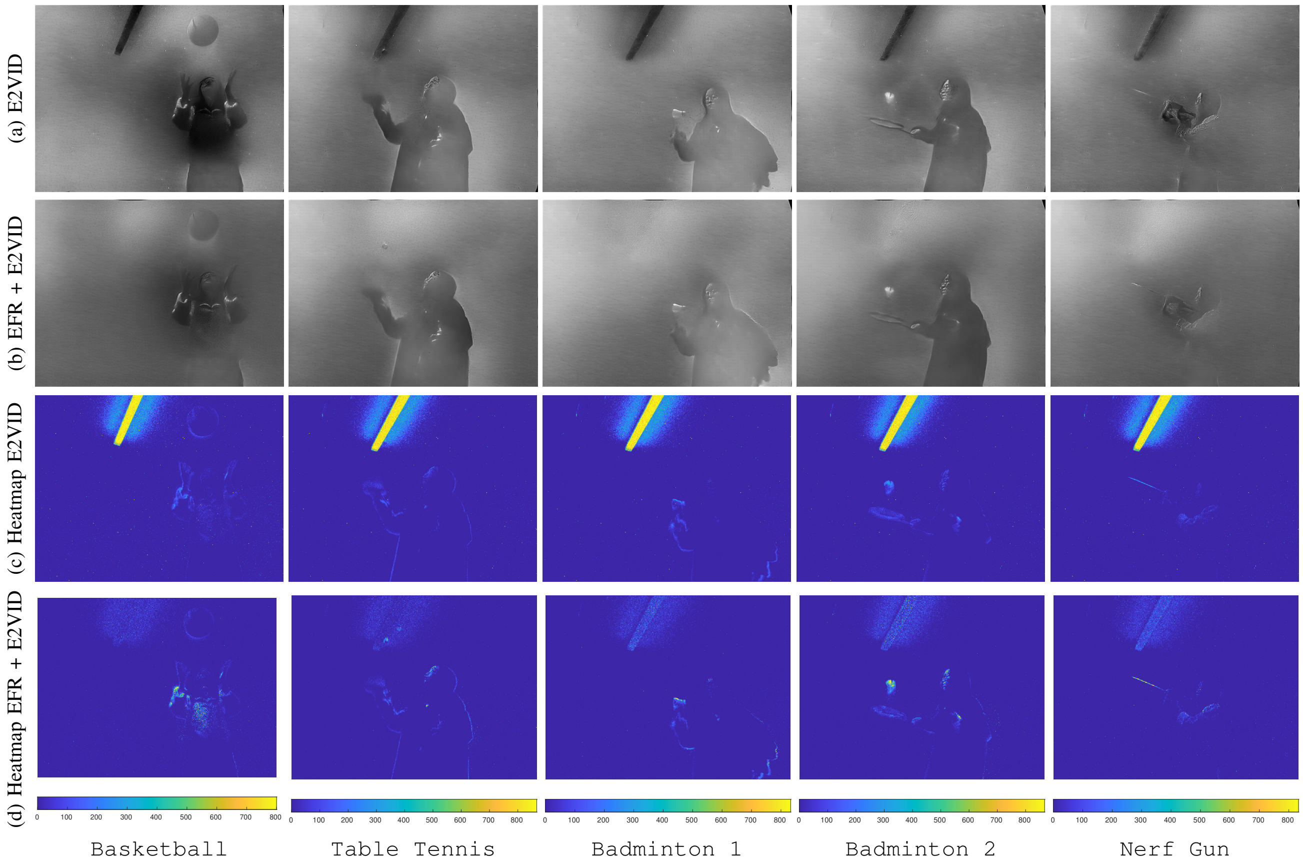
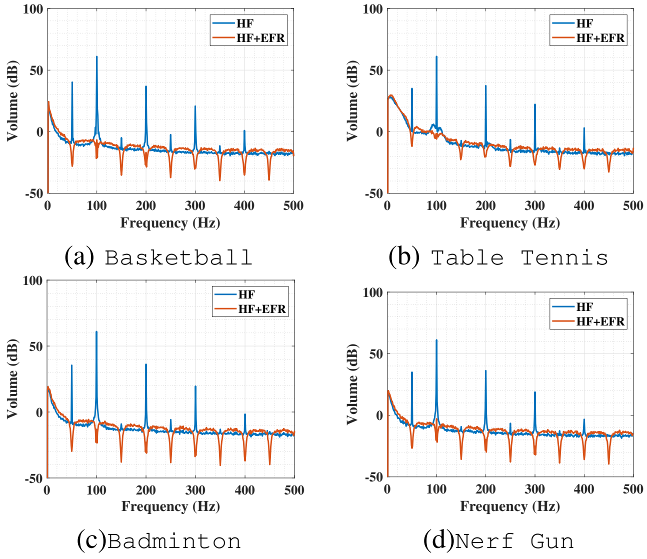


Fig. 5. Qualitatively analysis of the raw and filtered event data from five high-speed datasets with a flickering fluorescent light source. The first row (a) are image reconstruction E2VID [16] using raw event data. The second raw (b) shows image reconstruction E2VID [16] after event flicker removal using our proposed filter (EFR), and the flickering light source is clearly attenuated. The third and fourth row demonstrate heat maps of events per second of the raw and filtered event stream. The events rate from 0 to the maxima is coloured from blue to yellow. Before flicker removal, most events occur around the fluorescent light source. The ratio of events occurring at the flickering region to all events before filtering is 84.0% on average. Our event-based flicker removal algorithm filters out most of the flickering events and mostly preserves the useful event data under the lights. Our filter is not affected by the motion across the flickering source (shown in `Table Tennis`).



(a) `Basketball` (b) `Table Tennis`

(c)`Badminton` (d)`Nerf Gun`

Fig. 6. The power spectral density of high-frequency video reconstruction using HF [29]. The 50Hz, 100Hz flicker and their harmonics are effectively attenuated with our event-based flicker removal algorithm.

We also quantitatively analyse our flicker removal algorithm in Table I, evaluating on the same time period as Fig. 5. We use the signal-to-noise ratio (SNR) `♯foreground-events/♯flicker-events` as a metric for the quality of the data. A higher SNR indicates that most of the events in the dataset are relevant to the robot, whereas a lower SNR indicates that many of the events are associated with uninteresting flicker events. The first row of Table I(a) documents the raw dataset. Note that even though the fluorescent light in this scene is less than 5% of the image, the SNR is an average of 0.19 (or around 84% of the data is flicker). After event flicker removal (Table I(b)) the SNR average is 1.07 indicating that at least 50% of the remaining events are of interest to the robot. The relative improvement in SNR $(b-a)/(a)$ is shown in the final row of Table I. On average, the relative improvement in SNR is a factor of 4.63 times better after event flicker removal. It should be noted that the SNR is highly dependent on the foreground motion. For example, the SNR of the event steam in `Basketball` dataset is much higher than the others (in all three metrics). It is because the moving object in the dataset is bigger, triggering more foreground events than other datasets.

TABLE I

Quantitative analysis of our flicker removal algorithm. (a) Signal-to-noise ratio (SNR = ♯`foreground-events`/♯`flicker-events`) of the unfiltered event stream. (b) SNR of the filtered event stream. (b-a)/(a) Relative improvement in SNR due to event flicker removal.

Metrics	`Basketball`	`Table Tennis`	`Badminton`	`Nerf Gun`	**Avg.**
(a)	0.24	0.17	0.19	0.17	0.19
(b)	2.08	0.92	0.72	0.56	1.07
(b-a)/(a)	7.67	4.41	2.79	2.29	4.63

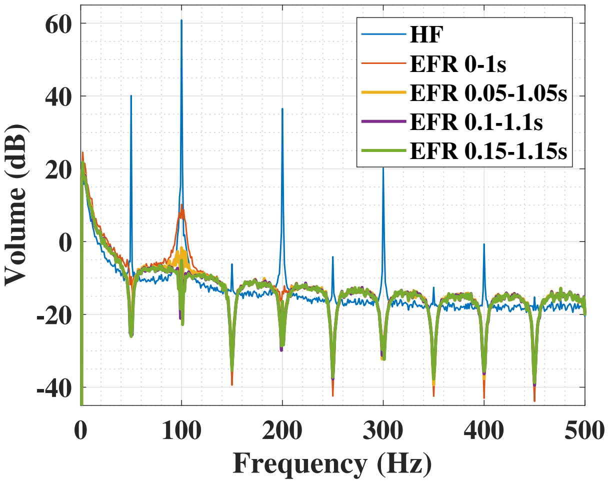


Fig. 7. The power spectral density of the high-pass filter and the filtered reconstructed video of the dataset `Basketball` in the first four 1-second periods. In the time period 0-1 seconds, there is still flicker in the reconstructed video, but once the filter state is converged (0.1 seconds later), the most undesired flicker in the video is removed.

2) Power spectral density analysis: The flicker removal performance is also analysed by the power spectral density (PSD) diagrams. We reconstruct high-frequency video at 1000Hz using a high-pass filter (HF) [29] and analyse the PSD of a patch at the flickering area before and after applying our flicker removal algorithm. The PSD diagrams are shown in Fig. 6. Note that the fluorescent light triggers events at both 50Hz, 100Hz and their harmonics. Therefore, we set the based frequency to 50Hz to attenuate all flickering frequencies. Fig. 6(a)(c)(d) are very similar because there is no object moving across the flickering light source. The table tennis ball in Fig. 6(b) moves across the flickering fluorescent light, leading to more components at low frequencies. It also affects the flickering frequency of raw event data at 100Hz, and the 100Hz notch of our EFR algorithm is not as deep as the other datasets, but most of the flickering events are still removed.

To evaluate the convergence rate of our event-based flicker removal (EFR) algorithm, we also demonstrate the PSD of high-frequency video reconstruction by HF [29] and flicker filtered reconstruction in the first four 1-second sliding window periods (see Fig. 7). After 0.1 seconds, the PSD plot shows that most frequencies are unchanged while the flickering frequencies have been mostly attenuated. It shows that our event-based flicker removal algorithm takes around 0.1 seconds to converge.

V. CONCLUSIONS

In this paper, we introduce the problem that event cameras are sensitive to flicker such as from fluorescent or LED lights. We proposed the first event flicker removal algorithm using a modified linear comb filter and the first event camera deflickering dataset that includes objects with fast motions under a flickering fluorescent light for evaluation. Our proposed filter yields a relative improvement 4.63 in the signal-to-noise ratio of the filtered event stream compared to the raw event stream on average, and the flicker removal performance is visualised using power spectral density. It effectively attenuates flicker events and the associated harmonics, with convergence within 0.1 seconds. Our filter can be used as a front-end or preprocessing algorithm to provide a cleaner and flicker-free event stream for other robotics applications.

REFERENCES

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, “128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor,” *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[2] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis *et al.*, “Event-based vision: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[3] D. Tedaldi, G. Gallego, E. Mueggler, and D. Scaramuzza, “Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS),” in *Int. Conf. Event-Based Control, Comm. Signal Proc. (EBCCSP)*, 2016.

[4] H. Liu, D. P. Moeys, G. Das, D. Neil, S.-C. Liu, and T. Delbruck, “Combined frame- and event-based detection and tracking,” in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2016, pp. 2511–2514.

[5] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, “Asynchronous, photometric feature tracking using events and frames,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 766–781.

[6] L. Pan, R. I. Hartley, C. Scheerlinck, M. Liu, X. Yu, and Y. Dai, “High frame rate video reconstruction based on an event camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[7] Z. Jiang, Y. Zhang, D. Zou, J. Ren, J. Lv, and Y. Liu, “Learning event-based motion deblurring,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020, pp. 3320–3329.

[8] H. Kim and H. J. Kim, “Real-time rotational motion estimation with contrast maximization over globally aligned events,” *IEEE Robotics and Automation Letters*, 2021.

[9] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, “Event-based vision meets deep learning on steering prediction for self-driving cars,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 5419–5427.

[10] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, “Semi-dense 3d reconstruction with a stereo event camera,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 235–251.

[11] Z. Wang, L. Pan, Y. Ng, Z. Zhuang, and R. Mahony, “Stereo hybrid event-frame (shef) cameras for 3d perception,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9758–9764.

[12] D. Gehrig, M. Rüegg, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, “Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2822–2829, 2021.

[13] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, “Low-latency visual odometry using event-based feature tracks,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016, pp. 16–23.

[14] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, “Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[15] D. Liu, A. Parra, and T.-J. Chin, “Spatiotemporal registration for event-based visual odometry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4937–4946.

[16] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, “High speed and high dynamic range video with an event camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[17] T. Delbruck, “Frame-free dynamic digital vision,” in *Proc. Int. Symp. Secure-Life Electron.*, 2008, pp. 21–26.

[18] C. Brandli, L. Muller, and T. Delbruck, “Real-time, high-speed video decompression using a frame- and event-based DAVIS sensor,” in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014, pp. 686–689.

[19] S.-H. Ieng, C. Posch, and R. Benosman, “Asynchronous neuromorphic event-driven image filtering,” *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1485–1499, 2014.

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

[21] A. Khodamoradi and R. Kastner, “$o(n)$ o (n)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors,” *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 15–23, 2018.

[22] V. Padala, A. Basu, and G. Orchard, “A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth,” *Frontiers in neuroscience*, vol. 12, p. 118, 2018.

[23] Z. Wang, Y. Ng, P. van Goor, and R. Mahony, “Event camera calibration of per-pixel biased contrast threshold,” *Australasian Conference on Robotics and Automation, ACRA*, 2019.

[24] T. Stoffregen, C. Scheerlinck, D. Scaramuzza, T. Drummond, N. Barnes, L. Kleeman, and R. Mahony, “Reducing the Sim-to-Real gap for event cameras,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[25] L. P. Frater, “Light flicker and harmonic modelling of electrical lighting,” 2015.

[26] S.-C. Pei and C.-C. Tseng, “A comb filter design using fractional-sample delay,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 5, pp. 649–653, 1998.

[27] G. Jovanovic-Dolecek and S. K. Mitra, “A new two-stage sharpened comb decimator,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 7, pp. 1414–1420, 2005.

[28] Z. Wang, Y. Ng, C. Scheerlinck, and R. Mahony, “An asynchronous kalman filter for hybrid event cameras,” *Int. Conf. Comput. Vis. (ICCV)*, 2021.

[29] C. Scheerlinck, N. Barnes, and R. Mahony, “Continuous-time intensity estimation using event cameras,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 308–324.

[30] S. Lin, J. Zhang, J. Pan, Z. Jiang, D. Zou, Y. Wang, J. Chen, and J. Ren, “Learning event-driven video deblurring and interpolation,” in *Eur. Conf. Comput. Vis. (ECCV)*, vol. 3, 2020.

[31] “jaer open source project,” https://github.com/SensorsINI/jaer, 2007.

[32] D. Czech and G. Orchard, “Evaluating noise filtering for event-based asynchronous change detection image sensors,” in *IEEE Int. Conf. Biomed. Robot. and Biomechatron. (BioRob)*, 2016, pp. 19–24.

[33] Z. W. Wang, P. Duan, O. Cossairt, A. Katsaggelos, T. Huang, and B. Shi, “Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1609–1619.

[34] Y.-H. Shiau, Y.-T. Kuo, P.-Y. Chen, and F.-Y. Hsu, “Vlsi design of an efficient flicker-free video defogging method for real-time applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 238–251, 2019.

[35] K. Chono, Y. Senda, and Y. Miyamoto, “Detented quantization to suppress flicker artifacts in periodically inserted intra-coded pictures in h. 264 video coding,” in *2006 International Conference on Image Processing*, 2006, pp. 1713–1716.

[36] B. Guthier, S. Kopf, M. Eble, and W. Effelsberg, “Flicker reduction in tone mapped high dynamic range video,” in *Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*, vol. 7866. International Society for Optics and Photonics, 2011, p. 78660C.

[37] V. Chan, S.-C. Liu, and A. van Schaik, “Aer ear: A matched silicon cochlea pair with address event representation interface,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 1, pp. 48–59, 2007.

[38] S. Hassani, *Mathematical methods: for students of physics and related fields*. Springer Science & Business Media, 2008, vol. 720.

[39] J. O. Smith, *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K publishing, 2010.

[40] A. Wilkins, J. Veitch, and B. Lehman, “Led lighting flicker and potential health concerns: Ieee standard par1789 update,” in *2010 IEEE Energy Conversion Congress and Exposition*. IEEE, 2010, pp. 171–178.