EBSnoR: Event-Based Snow Removal by Optimal Dwell Time Thresholding

Abigail Wolf, Student Member, IEEE, Shannon Brooks-Lehnert, and Keigo Hirakawa, Senior Member, IEEE

Abstract—We propose an Event-Based Snow Removal algorithm called EBSnoR. We developed a technique to measure the dwell time of snowflakes on a pixel using event-based camera data, which is used to carry out a Neyman-Pearson hypothesis test to partition event stream into snowflake and background events. The effectiveness of the proposed EBSnoR was verified on a new dataset called UDayton22EBSnow, comprised of front-facing event-based camera in a car driving through snow with manually annotated bounding boxes around surrounding vehicles. Qualitatively, EBSnoR correctly identifies events corresponding to snowflakes; and quantitatively, EBSnoR-preprocessed event data improved the performance of event-based car detection algorithms.

Index Terms—Event-based camera, snow removal, Neyman-Pearson hypothesis testing.

1 INTRODUCTION

In automotive imaging applications, environmental noise such as rain, snow, sun flares, etc. are nuisances that significantly deteriorate the performance of driver assistive computer vision algorithms. For instance, frames operating at conventional 30 or 60 frames per second yield snowflake streaks that obstruct the view of the road and the surrounding vehicles that make it difficult for computer vision algorithms to detect them accurately. This can be overcome to an extent by speeding up the frame rate to slow the snowflakes or reduce the frame integration time to freeze the snow. However, it comes at the significant computational burden of the post processing computer unit to accommodate higher frame throughput, or reduced sensitivity to light and increased noise.

We propose an event-based snow removal algorithm called EBSnoR, aimed at partitioning the event stream to snowflakes and background events as shown in Figure 1. EBSnoR exploits spatial-temporal statistical constraints of the snow as it appears on the image formed at the detector. Event-based cameras are better suited to detect snowflakes than the conventional intensity cameras because of their inherent asynchronous readout circuit and the high temporal resolution. Consider Figure 2 where a synthetic frame is rendered by accumulating events at 1ms, 16.66ms, and 33.33ms time intervals. The 1ms frame demonstrates the ability for event-based sensors to resolve high speed phenomenon, as evidenced by the individual snowflake particles observed. Comparing the 1ms frame with the 16.66ms and 33.33ms frames, the latter frames (corresponding to the integration times of 60 and 30 frames per second used by the conventional intensity cameras) suffer from long snowflake streaks stemming from the vehicle moving towards them at high speeds.

Owing to the real-time nature of automotive applications, we focus on simple and effective technique for snow removal. For instance, we rule out the possibility of tracking individual snowflakes whose complexity scales with the density of the snow and becomes unmanageable in heavy precipitation. Instead, the proposed EBSnoR is formulated as a statistically optimal likelihood ratio test performed on the snowflake dwell time, or the duration of the time that a snowflake is observed by a particular pixel. In Section 3 we derive the probability density function (pdf) of the snowflake dwell time. In Section 4 we develop an event-based method to measure the snowflake dwell time and propose a Neyman-Pearson hypothesis test-based snowflake detection via the dwell time thresholding. In Section 5, we demonstrate that the performance of the event-based car detection algorithm improves when performed on the background events only.

Our contributions can be summarized as follows:

• A rigorous statistical analysis of snowflake dwell time and the development of Neyman-Pearson hy-
portional in number to the magnitude of the log-intensity referred to as the edge arrival. The subsequent events of the same polarity are edge encounter, which corresponds to the exact timing of the edge. Event refers to the first of the series of events generated from an inceptive event (IE), trailing events (TE), and noisy events (NE).

2 BACKGROUND AND RELATED WORK

2.1 Event Cameras

Event-based camera is an emerging imaging modality modeled after the human visual system. In contrast to a conventional camera with active pixel sensors (APS) designed to measure pixel intensity synchronously across a frame, pixels in event-based sensors detect intensity changes asynchronously. Specifically, each pixel operates independently of the surrounding pixels to detect a change in log-intensity exceeding a threshold. Change detection is reported as an "event," coupling the X-Y pixel coordinate and the polarity of the event (whether the event is caused by an increase or a decrease in intensity) with the event timestamp.

The advantages of the event cameras over the conventional intensity cameras include the dynamic range exceeding 120dB, temporal resolution in microseconds, 10-100 order millisecond latency, and power consumption in 30 milliwatt range. Owing to its potential to advance the state of real-time high speed image processing and computer vision systems, event-based applications have been considered in fields such as camera tracking [1], high-speed/high-dynamic-range video [2], machine learning [3], image reconstruction [4], and autonomous vehicles [5].

2.2 Inceptive Event Filtering

Multiple events are generated when a pixel encounters an intensity change or an "edge." These events are triggered sequentially rather than simultaneously, introducing ambiguity in timing. Inceptive event filtering is a method of organizing event stream in a way to better reflect the edge properties [6]. Specifically, inceptive event filtering categorizes events into three types: inceptive events (IE), trailing events (TE), and noisy events (NE). Inceptive event refers to the first of the series of events generated from an edge encounter, which corresponds to the exact timing of the edge arrival. The subsequent events of the same polarity are referred to as the trailing events. The trailing events are proportional in number to the magnitude of the log-intensity edge, but the latency associated with them make the event timestamps ambiguous and imprecise. Thus we often count and report the number of trailing events associated with a given inceptive events as "edge magnitude," but not their event timestamps. Inceptive event with no corresponding trailing events are sometimes referred to as noisy events and removed from event stream when not useful.

Baldwin et al. performed the event classification based on the heuristics shown in Table 1 [6]. Here, \( event(n).ts \) represents the timestamp of the event currently being processed, \( event(n-1).ts \) represents the timestamp of the previously processed event, \( event(n+1).ts \) represents timestamp of the next event to be processed, and \( \Delta \) is a predefined threshold. Steps to carry out the inceptive event filtering are summarized in Figure 3(a). The output may be just an event label (IE, TE, or NE), or graph as shown in Figure 3(b).

Example output from inceptive event filtering is shown in Figure 3 showing only inceptive events. Since the inceptive events correspond to the exact timing of the edges, the inceptive events simplify the overall image while maintaining high edge shape fidelity.

2.3 Image Snow Removal Techniques

Techniques aimed at mitigating environmental noise such as rain and snow have been developed for conventional intensity cameras generating frames. Since intensity cameras fully capture background scenery in addition to foreground snow/rain, image-based rain/snow removal methods typically require two major steps: detection of rain/snow-affected pixels, and replacement of those pixels with estimated background intensity values. Existing detection methods typically fall into four main categories: photometric, geometric, temporal, and machine learning.

Photometric methods of rain/snow removal rely on the visual properties of snow, such as brightness, saturation and color. Methods by Chen et al. [7] and Xu et al. [8] use single guided filters to separate high and low spatial frequency components as a way to process rain/snow removal. Zheng et al. [9] also make use of guided filters, however their algorithm uses multi-guided filters in order to separate first the high and low frequency components of the image, and then to separate the rain/snow from the background in the high-frequency data. The method proposed by Manu [10] uses \( L_0 \) gradient minimization to locate and preserve significant edges, treating rain/snow as a form of noise. Method by Santhaseelan and Asari [11] uses the phase congruency of rain to detect potential rain streaks in video. Pei et al. [12] make use of the saturation and visibility properties of rain/snow, developing a network of high pass
TABLE 1
Classification parameters for inceptive event filtering algorithm [6]

| Condition | Description       | Event Type          |
|-----------|-------------------|---------------------|
| event(n+1).ts - event(n).ts ≤ Δ | Trailing Event     |
| event(n+1).ts - event(n).ts > Δ | Inceptive Event    |
| event(n).ts - event(n-1).ts ≤ Δ | Trailing Event     |
| event(n).ts - event(n-1).ts > Δ | Noisy Event        |

(a) Inceptive event filtering state diagram
(b) Output inceptive event graph

Fig. 3. State diagram and output graph structure of inceptive event filtering.

Fig. 4. System diagram and data visualization of the event processing including the proposed event-based snow removal (EBSnoR) algorithm.

filters, orientation filters, and thresholding that will isolate rain/snow streaks. Garg and Nayer [13] developed a rain removal technique leveraging a physics-based model to describe the way that rain blurs an image. A K-means clustering approach proposed by Zhang et al. [14] uses intensity histogram across the entire video sequence to cluster and profile snow/rain profiles separate from the background.

Geometric rain/snow removal methods rely on the patterns and movement properties of rain/snow streaks. Bossu et al. [15] propose a method by which potential rain/snow streaks are identified using a histogram of orientation of streaks. A similar method is used by Brewer and Liu [16], identifying pixels that exhibit a very short-term intensity spike matching the shape properties of rain drops. Another method proposed by Li et al. [17] posits that the rain patterns are repetitive and sparsely scattered, and use these attributes to develop a multiscale convolutional sparse coding model that extract the streak patterns. Ren et al. [18] approach the problem from a matrix decomposition point of view classify intensity fluctuations caused by background, foreground, sparse rain/snow, and dense rain/snow. Barnum et al. [19] develop a physical model of a raindrop/snowflake in order to determine the general shape and brightness of streaks, which is combined with the statistical properties of rain/snow to identify and extract streak patterns.

Temporal rain/snow removal methods inherently make use of multiple video frames. Kim et al. [20] observe that rain/snow particles are too small to affect optical flow of an image, and thus obtain a rain map by subtracting temporally warped frames from the current processing frame. The map is then decomposed into a sparse representation to classify pixels as rain or not rain using support vector machine.

In recent years, removal of rain from intensity camera images/videos has moved to the machine learning realm. Porav et al [21] take a denoising approach, and use a generator model to remove rain/snow noise from images. Zhang and Patel [22] proposed to use a densely connected convolutional neural network architecture to improve the snow removal performance. Jiang et al. [23] use Langrangian shrinkage algorithm to discriminate a priori models of rain-affected and non-affected images.

Separate from the task of detection, the removal of rain/snow requires replacing the affected pixel values. In images, this is commonly accomplished by inferring the proper intensity and color of a rain-affected pixel through the analysis of nearby background objects [7]. In video, this
Rain/snow removal has been studied in automotive settings also. De Charette et al. [24] developed a smart headlight system that will de-illuminate raindrops and snowflakes detected by a camera. This will create the effect of the driver seeing through the rain/snow to observe the illuminated background.

The rain/snow removal task for event-based cameras is profoundly different from the intensity-based camera methods described above. Owing to asynchronous pixel readout circuit with microsecond-order temporal resolution, the generated events track the raindrop and snowflake movements continuously. As shown by events accumulated over 16.66ms and 33.33ms in Figure 2, the snowflakes in a driving scenario can travel considerable distances between the consecutive frames of typical intensity cameras. As such, event-based cameras have a significant advantage in detecting and removing snowflakes.

Another key difference is that in the event-based snowflake removal task, we simply remove events corresponding to snowflakes without estimating the “missing” background events. As is shown by our analysis in Section 3, the dwell time for a snowflake on a particular pixel is in the order of milliseconds. Hence only a handful of background events would actually be blocked by snowflakes in practice.

### 3 Snowflake Dwell Time Modeling

#### 3.1 Analysis of Snowflake Dwell Time

As illustrated in Figure 5, assume that the car is moving along the optical axis $Z$ of the front-facing camera at the speed $V$ far exceeding the snowflake’s velocity (i.e. snowflake velocity is negligible). Then, the snowflake projected onto the two dimensional image of the front-facing camera appears to be moving as the moving vehicle approaches the snowflake. Define $T$ as the “dwell time” or the amount of time that a snowflake intersects a particular pixel. We prove in the Lemma 1 below that the snowflake’s dwell time is independent of the distance $Z$ along the optical axis.

**Lemma 1.** Let $(x, y) \in \mathbb{R}^2$ be the image plane coordinate of the front-facing camera at focal length $\ell$. Then the dwell time of the snowflake when observed by the camera is

$$T = \frac{D \ell}{\sqrt{x^2 + y^2}},$$

where $V$ is the vehicle velocity and $D$ is the diameter of the snowflake.

**Proof.** The apparent velocity $U$ of the snow due to the vehicle motion is a function of the distance $Z$ of the snow along the optical axis, the location $(X, Y, Z)$ of the snowflake on the plane tangential to the optical axis, and the car’s velocity $V$:

$$U = \frac{V \sqrt{X^2 + Y^2}}{Z}.$$  \quad (2)

Thus the dwell time of the snowflake $T$ can be modeled as:

$$T = \frac{D}{U} = \frac{DZ}{\sqrt{X^2 + Y^2}},$$

where $D$ is the diameter of the snowflake. Recall that the camera coordinate $(X, Y, Z)$ can be projected onto image plane $(x, y)$ at the focal length $\ell$ by the following relation:

$$(x, y) = \frac{\ell}{Z}(X, Y).$$

Substituting this into (3) cancels the distance term $Z$ and proves the hypothesis in [1].

Some readers may find Lemma 1 surprising—as the vehicle approaches the snowflake (i.e. $Z$ gets smaller), the apparent pixel velocity $u$ of the snowflake increases:

$$u = \frac{\ell}{Z}U.$$ \quad (5)

Indeed, the closer snowflakes appear to move faster in image plane, as evidenced by longer streak in Figure 2. But the pixel diameter of the snow is also inversely proportional to distance $Z$:

$$d = \frac{D}{Z}.$$ \quad (6)

Hence as the vehicle approaches the snowflake, the increased pixel speed of the snowflake is cancelled out by the increased pixel diameter of the snowflake:

$$T = \frac{d}{u} = \frac{D/Z}{U/Z}.$$ \quad (7)

Thus we conclude that the dwell time is a function of snowflake diameter $D$ and the velocity $V$. One may also reparameterize dwell time in terms of the pixel distance $r = \sqrt{x^2 + y^2}$ from the optical axis (or more precisely, the direction of the vehicle motion), as follows:

$$T = \frac{D \ell}{V r}.$$ \quad (8)

It is easy to see that the snowflakes at the center of the image plane (small $r$) has longer dwell time than the snowflakes in the periphery (large $r$).
We rigorously derive the probability density function (pdf) \( f_T : \mathbb{R} \to \mathbb{R} \) of the dwell time \( T \) for a snowflake with diameter \( D \) and distance \( Z \) while the car travels at velocity \( V \). Although a typical image sensor is rectangle-shaped \((|x| < W/2 \text{ and } |y| < H/2)\) where \( W \) and \( H \) are widths and height, respectively, the math below is considerably simpler if the detector were circular—that is, \( r = \sqrt{x^2 + y^2} < R \) where \( R \) is the detector radius. For this purely mathematical convenience, we derive \( f_T(t) \) in terms of circular detector first, but subsequently draw conclusions about rectangular detector without the loss of generality.

We begin with the reasonable assumption that the snow particles are uniformly distributed. That is,

\[
f_{xy}(x,y) = \begin{cases} \frac{1}{R^2\pi} & \text{if } \sqrt{x^2 + y^2} < R \\ 0 & \text{else} \end{cases}
\]

where \( R \) is the detector radius. Furthermore, differentiating \( P(r \leq r_0) = \frac{r_0^2}{2\pi} \) with respect to \( r_0 \) yields

\[
f_r(r) = \begin{cases} \frac{2r}{R^2} & \text{if } r < R \\ 0 & \text{else} \end{cases}
\]

Treating (8) as a function of random variable, pdf of dwell time \( T \) take the following form:

\[
f_T(T|D,V) = \frac{f_r(D) f_T}{\frac{\partial}{\partial T} f_T} = \begin{cases} \frac{2D^2\ell^2}{R^2 V^2 T^3} & \text{if } T > \frac{D \ell}{V R} \\ 0 & \text{else} \end{cases}
\]

Thus the pdf of the dwell time is proportional to \((D/V)^2\) and inversely proportional to \(T^3\). Figure 3 shows example \( f_T(T|D,V) \) for various vehicle velocities \( V \). Most snowflake diameters are said to range between 0.02 inches (0.508mm) to 0.2 inches (5.088mm) [25]. Even for a large diameter \( D = 5mm \) and moderately slow driving speed \( V=20\text{kmh} \), the dwell time is concentrated below 2ms.

For rectangle sensor model, \( P(r \leq r_0) \) is computed by taking an intersection of the rectangle with the \( r_0 \) radius circle. While not difficult, there are many “case statements” to consider (such as when \( r_0 < H/2 \), \( H/w < r_0 < W \) or \( r_0 > W \) in landscape mode). Yet, even with the rectangular sensor, the main conclusion that the probability density function \( f_T(T|D,V) \) scales proportionally to \((D/V)^2\) and inversely proportional to \(T^3\) does not change.

**4 PROPOSED: EVENT-BASED SNOW REMOVAL**

In this section we present the theory and the implementation of the proposed event-based snow removal (EBSSnoR) algorithm. At the heart of EBSSnoR is the optimal statistical hypothesis testing on snowflake dwell time, which we develop in Section 4.1. We then present a novel technique for measuring the dwell time based on event data in Section 4.2. We make practical considerations in Section 4.3.

**4.1 Optimal Dwell Time Thresholding**

We formalize snowflake rejection in EBSSnoR as a Neyman-Pearson hypothesis testing on the dwell time. That is, we consider the hypotheses:

\[
\begin{align*}
H_0 : \text{ snowflake } \\
H_1 : \text{ background.}
\end{align*}
\]

Define the likelihood ratio function \( L(|V) : \mathbb{R} \to \mathbb{R} \) as:

\[
L(T|V) = \frac{f_T(T|H_1,V)}{f_T(T|H_0,V)},
\]

where \( f_T(T|H_i,V) \) is the conditional likelihood function of the hypothesis \( H_i \) for a given car speed \( V \). Invoking the seminal Neyman-Pearson Lemma [26], thresholding performed on \( L(T|V) \) is a provably optimal binary classification of the hypothesis. That is, the decision rule \( \delta \) takes the form

\[
\delta(T) = \begin{cases} 1 & \text{if } L(T|V) \geq \kappa \\ 0 & \text{if } L(T|V) < \kappa \end{cases}
\]

for some threshold value \( \kappa \). Working with likelihood ratio test is challenging because the likelihood functions \( f_T(T|H_i,V) \) are not explicitly defined, however. Nevertheless, strong theoretical results such as Theorem 2 below can be proven.

**Theorem 2.** Let \( \theta \) be the maximum physically realizable snowflake size in nature. Then the likelihood ratio function \( L(T|V) \) is a monotonically decreasing function for \( T > \frac{\theta}{VR} \).

Proof. Let \( f_D(D|H_0) \) denote pdf of snowflake diameters \( D \). We assume \( f_D(D|H_0) = 0, \forall D > \theta \) because \( D \) cannot exceed \( \theta \). Hence the null likelihood function takes the following form:

\[
f_T(T|H_0,V) = \int_0^\infty f_T(T|D,V) f_D(D|H_0) dD
\]

\[
= \int_0^\theta \frac{2D^2\ell^2}{R^2 V^2 T^3} \phi \left( T - \frac{D \ell}{VR} \right) f_D(D|H_0) dD
\]

\[
= \int_{\min(T,\frac{\theta VR}{D^2\ell^2})}^{\min(T,\frac{\theta VR}{D^2\ell^2})} \frac{2D^2\ell^2}{R^2 V^2 T^3} f_D(D|H_0) dD,
\]

where \( \phi : \mathbb{R} \to \mathbb{R} \) is the unit step function.

Similarly, let \( f_D(D|H_1) \) denote pdf not of snowflake diameters, but of physical dimension of any other details
in background scenery of interest. When the background motion is negligible relative to the vehicle’s velocity (a point we will revisit in Section 4.3), the dwell time pdf in (11) applies to background object details as well. Hence the likelihood function of the alternative hypothesis takes the following form:

\[ f_T(T|H_1, V) = \int_{0}^{T/\tau} \frac{2D^2\ell^2}{R^2V^2T^3} f_D(D|H_1)dD. \]  

(16)

Combining (15) and (16), we arrive at the likelihood ratio for \( T > \frac{\theta}{\tau R} \)

\[ L(T|V) = \frac{f_T(T|H_1, V)}{f_T(T|H_0, V)} = \frac{\int_{0}^{T/\tau} \frac{2D^2\ell^2}{R^2V^2T^3} f_D(D|H_1)dD}{\int_{0}^{T/\tau} \frac{2D^2\ell^2}{R^2V^2T^3} f_D(D|H_0)dD} \]

\[ = \frac{\int_{0}^{T/\tau} D^2 f_D(D|H_1)dD}{\int_{0}^{T/\tau} D^2 f_D(D|H_0)dD}. \]

(17)

The above likelihood ratio is a monotonically increasing function of dwell time \( T \) since \( D^2 \) and \( f_D \) are non-negative values.

It is important to emphasize that the monotonicity of the likelihood ratio in Theorem 2 was rigorously proven without explicitly defining or knowing the snowflake diameter pdf \( f_D(D|H_0) \) nor the background detail pdf \( f_D(D|H_1) \).

As stated earlier, the maximum flake size is reported to be around \( \theta = 5 \text{mm} \) [25].

The major significance of Theorem 2 is that the optimal thresholding is the thresholding of \( T \). That is, the optimal binary classifier in [14] is equivalent to a simple thresholding of the dwell time:

\[ \delta(T) = \begin{cases} 1 & \text{if } T \geq \eta \\ 0 & \text{if } T < \eta. \end{cases} \]

(18)

Thus the proposed EBSnoR carries out snowflake rejection using the simple dwell time thresholding in [18].

Next, we derive the false positive rate (snowflake misclassified as background) and the false negative rate (background misclassified as snowflake) of the hypothesis test in [12] in Corollary 3.

**Corollary 3.** Suppose we set the threshold \( \eta \) as proportional to the critical dwell time of the largest snowflake at velocity \( V \):

\[ \eta = \frac{\theta}{V}. \]

(19)

Then the false positive rate and the false negative rate of the likelihood ratio test in [18] is independent of the car speed \( V \).

**Proof.** The false positive rate in terms of the threshold in (19) can be derived as follows:

\[ P(T > \eta|H_0, V) = \int_{0}^{\infty} \int_{0}^{\eta} \frac{2D^2\ell^2}{R^2V^2T^3} f_D(D|H_0)dTdT \]

\[ = \int_{0}^{\infty} \frac{2D^2\ell^2}{R^2\theta^2V^2} f_D(D|H_0)dD. \]

(20)

The false negative rate is slightly more complicated, as shown below:

\[ P(T \leq \eta|H_1, V) = \int_{0}^{\eta} \int_{0}^{\tau \theta \eta / V} \frac{2D^2\ell^2}{R^2V^2T^3} f_D(D|H_1)dTdT \]

\[ = \int_{0}^{\eta} \left(1 - \frac{D^2\ell^2}{R^2\theta^2V^2} \right) f_D(D|H_1)dD. \]

(21)

We conclude from (20) and (21) that setting the threshold \( \eta \) to be inversely proportional to the vehicle velocity \( V \) makes the false positive and false negative rates independent of \( V \).

One fortunate outcome of Corollary 3 is that a single threshold parameter \( \tau \) would yield a consistent performance across any car speed \( V \) (i.e. the threshold \( \eta \) adjusts for velocity based on \( \tau \)). Neyman-Pearson hypothesis testing comes in two flavors. The first version sets the threshold \( \tau \) to satisfy the false positive rate at the significance level \( \alpha \):

\[ P(T > \eta|H_0, V) = P\left(T > \frac{\theta}{V} \eta \right) < \alpha. \]

(22)

The Neyman-Pearson Lemma provides a theoretical guarantee that the hypothesis test in [18] with threshold \( \tau \) satisfying (22) maximizes the detection at this significance level [26].

The false positive rate itself is tightly upper-bounded:

\[ P(T > \eta|H_0, V) < \frac{\ell^2}{R^2\theta^2}. \]

(23)

(proof: substitute \( \theta \) for \( D^2 \) in (20)). Thus in this work, we choose the threshold \( \tau \) according to the following rule:

\[ \tau = \frac{\ell}{RV}. \]

(24)

The second version of Neyman-Pearson hypothesis testing sets the threshold \( \tau \) to satisfy the false negative rate at the significance level \( \beta \):

\[ P(T < \eta|H_1, V) = P\left(T < \tau \frac{\theta}{V} \right) < \beta. \]

(25)

The threshold determined in this manner guarantees that hypothesis test in [18] maximizes snowflake rejection at this significance level. Unlike the earlier version of Neyman-Pearson hypothesis, this version must set the threshold \( \tau \) empirically because \( f_D(D|H_1) \) cannot be defined explicitly, however. In this work, we handled this by driving a car on an overcast day with no precipitation at a given speed.
V₀ while collecting data with the event camera. With no snowflakes in the scene, the empirical distribution of the dwell time (computed from the event data using the technique described in Section 4.2) is a baseline proxy for the likelihood function fₜ(T|H₁,V₀) at speed V₀. See example in Figure 7(a). Thanks to Corollary 3 setting the threshold τ by exhaustive search to empirically satisfy
\[ P \left( T < \frac{\tau \theta}{V₀}\mid H₁,V₀ \right) = \beta \] (26)
will determine the value of τ that generalizes to other speeds V ≠ V₀. By Neyman-Pearson lemma, threshold τ satisfying (26) guarantees maximum snow detection at the significance level β [26]. For comparison, Figure 7(b) shows the dwell time histogram of snowy day event sequence, although this histogram represents a combination of snowflake and background dwell times. Nevertheless, there is a stark contrast between the two histograms, where Figure 7(b) exhibits the characteristics seen in Figure 6.

4.2 Event-Based Dwell Time Measurement

The proposed EBSnoR is an optimal binary classification based on the decision rule in (18), carried out by thresholding the dwell time T. In this section, we develop a technique to measure the dwell time T from event data. We begin with the underlying assumption that the snow is brighter in relation to the surrounding and background intensity values. A snowflake appears luminous and white because it is made up of tiny translucent ice crystals that transmit or reflect light with minimal absorption [27]. The contrast with background is further exaggerated by the dimmer environmental light available in overcast sky during typical snowfall.

Consider the intensity timing plot in Figure 8(a). Due to the relative brightness of the snowflake compared to the surrounding background radiance, we see a positive step (positive edge) in the intensity at the moment that a snowflake intersects a pixel’s field of view. That intensity will remain high during the snowflake’s dwell time, and the negative intensity step (negative edge) is seen when the snowflake exists this particular pixel.

Thus the dwell time can be inferred from the intensity timing plot by measuring the time between a positive edge followed by a negative edge. The other edge intervals—“positive edge to positive edge,” “negative edge to positive edge,” and “negative edge to negative edge”—can be discarded since they are not consistent with the snowflake model above, and therefore can be attributed to the background. In practice, however, measuring the millisecond-order snowflake dwell time in this manner using conventional framing cameras is difficult because the framerate is too slow.

Using the above principles, the proposed EBSnoR measures the snowflake dwell time from event streams in several stages. The system-level block diagram of EBSnoR is shown in Figure 4. We explain the method in detail by comparing the intensity timing plot in Figure 8(a) to the corresponding event timing plot in Figure 8(b). The microsecond-order resolution of the event timing makes the event-based sensors more attractive for dwell time measurement. Recalling Section 2.2 the arrival of snowflake at a particular pixel is marked by the timestamp of the positive inceptive event. Likewise, the timestamp of the subsequent negative inceptive event coincides with the exact moment of the snowflake exit. Thus, the dwell time is measured between the timestamps of the positive inceptive event and its following negative inceptive event at the same pixel. See Figure 8(b) for the timing of inceptive events (green arrows) corresponding to snowflakes.

Taking real-world event-based sensor hardware into consideration, the dwell time measurement algorithm must account for “missed events.” That is, sometimes the pixels are not sensitive enough to generate events, even when a pixel encounters a large log-intensity edge. For this reason, we search over neighboring pixels to find positive and negative inceptive events in spatial-temporal proximity (temporal threshold of η and spatial window ω). This is illustrated in Figure 8(c), where spatially neighboring inceptive events allows us to recover from missing events to make dwell time measurements.

The algorithm is summarized in the state transition diagrams in Figure 9. For efficient implementation of this algorithm, we process the event based snowflake dwell time measurement as a FIFO (first in, first out) structure arranged in pixels. Referring to the positive event processing in Figure 9(a), when a positive polarity inceptive event at pixel location (x,y) is encountered, the event is stored in the FIFO at the corresponding location, and the state of the pixels moves to “B.” If the negative inceptive event is encountered within spatial window (x,y) ± ω before η = τθ/V, then snowflake is detected. The previous positive inceptive event (as well as the associated trailing events) are marked as snowflake (event(n-1).sf=1), and the state returns to “A.” Similarly, the negative polarity inceptive events are processed according to the state diagram in Figure 9(b). When a positive polarity inceptive event occurs within the spatial neighborhood (x,y) ± ω, its timing is recorded and the state moves to “D.” If the negative inceptive event is encountered at pixel (x,y) before η = τθ/V, then snowflake is detected. This new negative inceptive event and the corresponding trailing events are marked as snowflake (event(n).sf=1) as the state returns to “C.”

Unlike intensity cameras, removing unwanted data from event camera footage is as simple as removing undesired events from the data stream. Thus, once a list of snow events has been obtained, they are deleted from this list of event data points.

4.3 Discussions

Let us address several practical considerations for detecting snowflakes by thresholding dwell time T. First, camera parameters such as focal length and detector size R are fixed and known numbers. The maximum snowflake diameter θ=5mm is reasonable [25]. In automotive applications, vehicle velocity V can be provided by the odometer (although visual odometery techniques can also replace traditional odometry).

In the baseline hypothesis testing in (16), we made an assumption that the background object velocity is insignificant compared to the vehicle motion V. Generalizing to the case that the object velocity is insignificant (e.g. background
Fig. 8. Pixel timing diagram for (a) log-intensity and (b) events. Dwell time can be computed as the timestamp difference between positive and negative inceptive events. (c) Spatial-temporal window used by EBSnoR to threshold dwell time $T$ by the parameter $\eta$. Due to the potential for missing events, we look within the spatial window $\omega$ to find the positive-negative edge pairs.

Fig. 9. State diagram for detecting snow from inceptive events. When an inceptive event is declared as snowflake ($\text{event}(n).sf=1$), we assume that the corresponding trailing events are also labeled as snowflakes.

object is a car) replaces the denominator $V$ with apparent motion relative to the vehicle/camera $V'$. This does not change the main conclusion that the likelihood ratio in [17] is monotonic; and Neyman-Pearson threshold minimizing false positive rate is determined independent of $V'$.

The monotonicity of likelihood ratio in Theorem 2 is only guaranteed for $T > \frac{\theta \ell}{\sqrt{\alpha} VR}$. To understand why this has negligible impact in practice, substitute Neyman-Pearson threshold in (24) into (19):

$$\eta = \frac{\theta \ell}{VR\sqrt{\alpha}}.$$

Since the significance value $\alpha$ is less than 1 (and usually very small), we conclude that the practical operating range of the threshold $\eta$ is far larger than the critical boundary $\frac{\theta \ell}{VR\sqrt{\alpha}}$.

Revisiting (21), consider the false negative rate of the background object with a specific detail size $D_0$ at the Neyman-Pearson significance level of $\alpha$:

$$P(T \leq \eta | H_1, D_0, V) = \int_{\frac{\theta \ell}{VR\sqrt{\alpha}}}^{\eta} \frac{2D_0^2 \ell^2}{R^2 V^2 T^3} dT$$

$$= \max \left( 1 - \frac{D_0^2 \ell^2}{R^2 \theta^2 T^2}, 0 \right)$$

$$= \max \left( 1 - \alpha \frac{D_0^2}{\theta^2}, 0 \right).$$

Since this is a monotonically decreasing function of $D_0$ (at any distance $Z$, thanks to Lemma 1), the background details falsely rejected by EBSnoR affects small spatial details. For instance, at $\alpha=0.05$ significance rate, background details of $D_0=1.58\text{cm}$ will yield 50% false negative rate; false negative rate is 0% for details larger than $D_0=2.23\text{cm}$. Potential loss of spatial details with physical dimensions smaller than 2cm (again, at any distance $Z$) is believed to result in negligible performance loss for computer vision algorithms designed to detect large objects such as vehicles, street signs, and pedestrians.

Fig. 10. Percentage of events removed by EBSnoR, as a function of the EBSnoR threshold. For snowy day sequence (“background + snow”), very high percentage of events fall below the dwell time threshold. By contrast, the baseline sequence (“background”) is far less sensitive to small Neyman-Pearson threshold values.
Lastly, recall that the event-based dwell time measurement described in Section 4.2 measures the time between consecutive positive and negative inceptive events. Thus, the false negative rates in 25 and 28 are exaggerated, in the sense that thresholding does not impact events corresponding to consecutive positive-positive, negative-positive, and negative-negative edges. This fact is reflected in Figure 10 where we plot the percentage of events removed by EBSnoR as a function of threshold $\eta$. It can be seen that the slope of the removed pixels is very steep for snow sequence compared to the “background only” events near $\eta = 0$. This implies that the proposed thresholding effectively discriminates snowflakes from the background.

5 EXPERIMENTS

5.1 UDavton22EBSnow Dataset

For testing the effectiveness of snowflake removal, we collected a new dataset we call UDavton22EBSnow comprised of 9 sequences of driving through snowfall using Prophesee EVK2 HD event-based camera, with 5mm focal-length lens yielding horizontal and vertical fields of view of 63.76 degrees and 38.58 degrees, respectively. The camera was mounted on the dashboard (slightly off-centered towards the passenger side) inside the vehicle front-facing on a moving vehicle navigating through highway, city, and suburb roads. As expected, the highway driving resulted in fast apparent motion of the snowflakes with very short dwell times. City driving sequence at times involve slow ego-motion of the vehicle (recall that the theoretical development in Section 5 assumed that the snowflake speed was negligibly small compared to the vehicle speed). The suburban roads resulted in mostly steady vehicle ego-motion. Captured sequences range between two and five minutes in length. Some sequences contain scenes where the car was stopped at the traffic light. During the stop, the generated events were almost entirely due to snowflake movements and the surrounding vehicles that are still moving.

In three of the sequences, we manually annotated bounding boxes around vehicles—one event sequence from each of highway, city, and suburb drives. Annotations were made at one second intervals, and we also recorded the timestamps of the windshield wiper that sweeps across the camera’s entire field of view. Examples of the labeled cars are shown in Figure 11. The statistics of the event streams are summarized in Table 2. We caution that the number of annotated bounding boxes does not necessarily represent the number of unique cars—a car staying in the field of the camera’s view for more than one second will be annotated multiple times. A portion of the “City” sequence is spent driving through a parking lot, where there is a high number of cars that were annotated. This dataset will be made available to the public upon acceptance of this paper.

5.2 Event-Based Snow Removal Results

Recall Corollary 3 where the Neyman-Pearson threshold $\eta$ scales inversely proportional to ego-motion velocity $V$. Although this formulation has a nice guarantee in terms of false positive and false negative rates, we did not have a technical capability to record vehicle velocity when UDavton22EBSnoR data was being collected. For this reason, we simply fixed the threshold to $\eta = 3000ms$. This may be suboptimal at very fast and very slow ego-motion velocities, but the results were acceptable in most sequences.

Figure 12 shows the result of applying EBSnoR to snow sequence. As evidenced by the detected snowflake tracks, snowflakes are brighter than the surrounding environment—the arrival and the departure of snowflake at a pixel are marked by positive (red) and negative (blue) events, respectively. In the time-space representation like the one in Figure 12, these positive and negative events form parallel tracks, where the time interval between them represents the snowflake’s dwell time.

By contrast, typical background events form two dimensional manifolds stemming from the edges that travel across time. As evidenced by Figure 12, the events forming these manifolds often share the same polarity (at least within some local regions), and has more gentle slopes when compared to snowflakes because of the slower apparent motion on camera’s image plane. Though there are exceptions, by-and-large the we do not commonly see parallel positive and negative manifolds. Even if they do, there is a larger time gap separating them, compared to typical snowflake dwell time. Thresholding by event-based dwell time therefore safely preserves the events corresponding to the background.

EBSnoR applied to event sequences in UDavton22EBSnow are shown in Figures 1 and 13. By visual inspection, we can confirm that most snowflake events are correctly identified by the Neyman-Pearson thresholding. The exceptions (false positive) were very large snowflakes with low apparent speed found near the center of ego-motion, resulting in longest dwell time. The snowflakes did not have enough contrast against the overcast sky to generate events.

The background events are largely preserved as well. Road markings, cars, and structures in the environment are correctly classified as background. In the highway sequence, there seems to be instances where the thin branches are miss-classified as snowflake because of low dwell time. Although it is difficult to determine the level of false negatives (background classified as snowflake) in the textured regions such as leaves in “Suburb” sequence, events classified as snowflake in the leaves region generally form streaks consistent with the behavior of snowflakes.

5.3 Car Detection Results

Because manual annotation of events corresponding to snow is an impractical task, quantitatively evaluating the performance of the event-based snow removal algorithm on this dataset is difficult. Instead, we propose to assess the effectiveness of EBSnoR indirectly by applying an event-based car detection algorithm to snow-removed event stream as a proxy. Specifically, we used convolutional neural network (CNN) based car detection algorithm provided in the Prophesee Metavision Toolkit [28] that was trained on scenes with no snow. We used the network with no modifications—default settings and thresholds, and with no retraining. The network outputs bounding boxes, which we compare to the manually annotated bounding boxes (ground truth).

We use common evaluation metrics for detection tasks, such as recall, precision, the average percent overlap (PO)
Fig. 11. Example of UDayton22EBSnow dataset events and annotated bounding box around vehicles.

Table 2

| Sequence Name | # Events | Total Sequence Duration | Stopped at Traffic Duration | # Bounding Boxes | # Bounding Boxes |
|---------------|----------|-------------------------|----------------------------|------------------|------------------|
| City          | 1.2 billion | 155s                  | 18s                        | 725              | 20               |
| Suburb        | 6.5 billion | 409s                  | 36s                        | 362              | 50               |
| Highway       | 3.3 billion | 176s                  | 0s                         | 109              | 0                |

Fig. 12. EBSnoR partitions event stream into snowflakes (red=positive, blue=negative) and background (yellow=positive, green=negative) events. Applied to UDayton22EBSnow “Highway” sequence. It can be seen that snowflakes form a track of positive event, followed immediately by a negative event track—the space between them represents the snowflake dwell time.

and the average intersection over union (IOU) of bounding boxes. For scoring, we ignored the ground truth and the detected bounding boxes within 0.3ms of the wiper blade sweep. For benchmarking, we repeat the car detection experiment on the entire unprocessed (i.e. not snow-removed) event stream—we refer to this as the “baseline.”

The results are reported in Table 3. As one can see, all metrics suggest that the car detection task improves when pre-processed by EBSnoR compared to the unprocessed event streams. The default threshold on the Prophesee Metavision Toolkit’s CNN seemed to be tuned to yield high precision, erring on the side of lower recall value. Nevertheless, the largest gain was in the recall in the “City” sequence, where number of cars detected was considerably higher with the snow-removed event stream while maintaining precision above 88%. Recalling that the number of bounding boxes in Table 2 does not represent unique cars, most of the misses (opposite of recall) by both EBSnoR and baseline results occurred when the cars first appeared into the scene. Most cars were eventually detected, but EBSnoR-preprocessed car detection tended to recognize the car earlier than the “not snow-removed” event stream—leading to a higher recall percentages. In time-critical applications such as automotive imaging, early detection of cars with EBSnoR-processing is a distinct advantage.

6 Conclusion

In this paper, we proposed a novel event-based snowflake removal algorithm called EBSnoR. Based on the rigorously derived probability density function of the snowflake dwell time and the monotonicity proof of its likelihood ratio, we developed a Neyman-Pearson hypothesis test to detect snowflake by thresholding dwell time. We also develop a method to measure the dwell time from the event stream, based on which we perform the hypothesis test to partition the event stream into snowflakes and background events. The performance of the proposed EBSnoR is assessed using UDayton22EBSnow dataset comprised of city, suburb, and highway drives through snow. By visual inspection, we verified that the algorithm does an excellent job at detecting snow. Although textures (such as leaves) were sometimes misclassified as snowflakes, EBSnoR by and large did an excellent job at detecting the snow. Quantitative evaluation was done by performing event-based car detection algorithm on EBSnoR-preprocessed event stream, which increased recall, precision, the percentage overlap, and the intersection over union of the car bounding boxes.

Acknowledgments

This work was made possible in part by funding from Ford Motor Company University Research Program.

References

[1] G. Gallego, J. E. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, “Event-based, 6-dof camera tracking from photometric depth maps,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 10, pp. 2402–2412, 2018.

[2] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, “High speed and high dynamic range video with an event camera,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 6, pp. 1964–1980, 2021.
Fig. 13. EBSnOR partitions event stream into snowflakes (red) and background (green-positive, blue-negative) events. Applied to UDAYton2EBSnow “Highway” and “Suburb” sequences and played back at 0.3× speed.

| Sequence | Avg. PO | Avg. IOU | Precision | Recall | Avg. PO | Avg. IOU | Precision | Recall |
|----------|---------|----------|-----------|--------|---------|----------|-----------|--------|
| City     | 41.822% | 28.508%  | 88.265%   | 44.716%| 41.317% | 27.200%  | 87.768%   | 42.097%|
| Suburb   | 48.148% | 37.963%  | 77.358%   | 33.148%| 44.716% | 33.282%  | 60.345%   | 37.963%|
| Highway  | 46.875% | 36.804%  | 81.081%   | 27.523%| 42.097% | 33.282%  | 80.555%   | 26.606%|

TABLE 3

Car detection results using annotated UDAYton2EBSnow event sequences. Snow removed events yield higher average percent overlap (PO), average intersection over union (IOU), precision, and recall when compared to the baseline event stream containing both snow and background events. The CNN detection method in [28] was used without modification (i.e., trained without snow in consideration).
Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[23] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, “Fastderain: A novel video rain streak removal method using directional gradient priors,” IEEE Transactions on Image Processing, vol. 28, no. 4, pp. 2089–2102, 2019.

[24] R. D. Charette, R. Tamburo, P. Barnum, A. Rowe, T. Kanade, and S. G. Narasimhan, “Fast reactive control for illumination through rain and snow,” in Proceedings of (ICCP) IEEE International Conference on Computational Photography, April 2012.

[25] T. Skilling, “How big are ‘big’ snowflakes?” The Chicago Tribune, 2006. [Online]. Available: https://www.chicagotribune.com/news/ct-xpm-2006-02-12-0602120724-story.html

[26] J. Neyman and E. S. Pearson, “Ix. on the problem of the most efficient tests of statistical hypotheses,” Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, vol. 231, no. 694-706, pp. 289–337, 1933.

[27] J. J. Koenderink and W. A. Richards, “Why is snow so bright?” J. Opt. Soc. Am. A, vol. 9, no. 5, pp. 643–648, May 1992. [Online]. Available: http://opg.optica.org/josaa/abstract.cfm?URI=josaa-9-5-643

[28] P. de Tournemire, D. Nitti, E. Perot, D. Migliore, and A. Sironi, “A large scale event-based detection dataset for automotive,” arXiv preprint arXiv:2001.08499, 2020.