Turbulence mitigation in imagery including moving objects from a static event camera

Nicolas Boehrer, **Robert P. J. Nieuwenhuizen**, and **Judith Dijk**[*]
TNO Intelligent Imaging, The Hague, The Netherlands

Abstract. Long-range horizontal path imaging through atmospheric turbulence is hampered by spatiotemporally randomly varying shifting and blurring of scene points in recorded imagery. Although existing software-based mitigation strategies can produce sharp and stable imagery of static scenes, it remains highly challenging to mitigate turbulence in scenes with moving objects such that they remain visible as moving objects in the output. In our work, we investigate if and how event (also called neuromorphic) cameras can be used for this challenge. We explore how the high temporal resolution of the event stream can be used to distinguish between the apparent motion due to turbulence and the actual motion of physical objects in the scene. We use this to propose an algorithm to reconstruct output image sequences in which the static background of the scene is mitigated for turbulence, while the moving objects in the scene are preserved. The algorithm is demonstrated on indoor experimental recordings of moving objects imaged through artificially generated turbulence. © *The Authors. Published by SPIE under a Creative Commons Attribution 4.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI.* [DOI: 10.1117/1.OE.60.5.053101]

1 Introduction

Light traveling through the atmosphere encounters turbulent regions that modify the optical path length.[1] As the light propagates, the effects of turbulent regions accumulate, leading to a random phase distortion of the wavefront, which causes time-varying blurs and shifts in the image recorded by a camera. Atmospheric turbulence therefore limits the effective resolution of optical imaging in many long-range observation applications such as surveillance or astronomy.

The effect of turbulence can be mitigated using hardware capable of measuring and correcting for the wavefront distortion while recording (adaptive optics) and/or using software such as image processing techniques.[2] For astronomy, adaptive optics often performs well at correcting the wavefront distortion for the observation of point sources such as stars. For observation of extended areas (in surveillance applications), this technique is often unsuitable, as the wavefront distortion varies across the field of view. This implies that a correction that removes the distortion for one point in the scene does not remove the distortion for regions in other parts of the scene. In such a case, mitigation through image (post)processing is preferred. Most image processing techniques rely on a combination of motion compensation, sharp image region (often called lucky region) identification, multi-frame data fusion, and deconvolution.[3–14] Motion compensation is often accomplished by computing a static reference frame and warping all frames in a sequence to that reference. Typically, the reference frame is obtained by temporal filtering of the intensity values per pixel, such as in the methods of Fishbain et al.[3] and Zhu and Milanfar,[7] or by filtering the estimated pixel motion from frame-to-frame to estimate the true pixel location, such as in the method of Halder et al.[14] Alternatively, a dynamic reference frame can be computed by tracking the frame-to-frame motion, such as in the method of Nieuwenhuizen et al.[8] Sharp image regions are identified using spatial sharpness measures, often based on the local gradients, such as in Aubailly et al.[11] Finally, many proposals have been made for the multi-frame data fusion, often based on temporal low-pass filtering and subsequent sharpening or deconvolution, such as in Zhu and Milanfar.[7] Notable alternative approaches include

---

*Address all correspondence to Judith Dijk, judith.dijk@tno.nl

that of Anantrasirichai et al.,[13] who proposed a recursive image fusion scheme using the dual-tree complex wavelet transform, and Oreifej et al.,[6] who proposed the use of a three-term low-rank matrix decomposition of the spatiotemporal data cube to extract the background estimate.

A limitation of image processing approaches is that the frame rate of classical cameras is typically too low to capture all of the dynamics of the turbulence-induced changes in the images. As a result, local regions in single frames combine instances that are temporarily sharp with instances that are less sharp, and tip/tilt aberrations are averaged to further blur these regions. Moreover, distinguishing moving objects from motion due to turbulence is often problematic at these frame rates. At these frame rates, the frame-to-frame shifts can be substantially larger than a single pixel. Due to this, and due to the small-scale turbulent image distortions, it is difficult to accurately estimate the shifts from the images, which severely complicates the identification of moving objects pixels as shown in Re. 15.

Event cameras, also known as neuromorphic cameras, do not record an entire frame with a shutter but instead output an asynchronous stream of intensity changes, so-called events. Contrary to classical frame recordings that record a lot of redundant data of image regions that stay constant, this new recording technique allows for recording of only the local changes between frames, so the bandwidth and the recording resources are best used to record the local dynamics of the scene or of the camera. When combined with computer vision algorithm like optical flow such as in Ref. 16, visual odometry such as in Ref. 17, 18, and 19, 3D reconstruction such as in Ref. 20, this new paradigm shows advantages for dynamic scene, see Ref. 21 for an extensive overview. With its low latency and high temporal sampling, the event camera is therefore expected to be well suited to record the temporal variations of the atmospheric turbulence otherwise unseen by a conventional camera. This additional information may be used to improve the quality of the restored image as shown in Ref. 22. In this paper, we present our results combining image processing on an intensity image recorded by a camera and event processing to show the enhancement brought by the additional event stream over classical frame-based mitigation.

Because this is a first exploration of this possibility, the scope of the investigation is limited here to applications in which the camera itself is static and moving objects exhibit rigid body motion. This means that different parts of the moving object do not exhibit significant motion relative to each other in the imagery. Section 2 describes the principle of operation of the event camera. Section 3 details how the rapid sampling of the event camera is used to reconstruct a fixed background, separate moving objects from turbulence motion, and reconstruct the appearance of a fast-moving object. These algorithmic building blocks are used to construct the turbulence mitigation pipeline described in Sec. 3. The experimental setup to validate this approach is described in Sec. 4. Section 5 explains the results of the experiment designed to assess the benefit of the event camera and presents results of the turbulence mitigation pipeline compared with state-of-the-art image processing methods. Finally, the conclusions are summarized in Sec. 6.
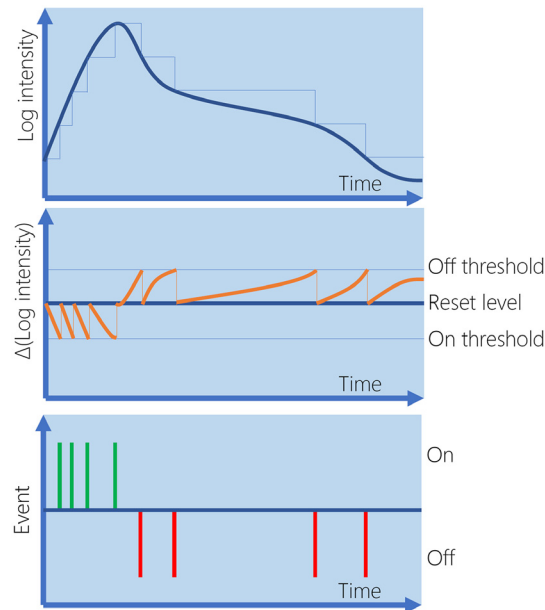
2 Event Camera

Unlike conventional cameras that record entire frames synchronously, event cameras only record logarithmic intensity changes. The event camera encodes the changes as an asynchronous series of spikes called events.

$$e_k = (u_k, t_k, p_k), \tag{1}$$

where the k'th event e_k is a quadruplet that consists of a location $\boldsymbol{u_k} = (x_k, y_k)^T$, a polarity p_k (positive or negative direction), and a time stamp t_k. As shown in Fig. 1, an event is produced when the difference between the memorized log intensity and the current log intensity exceeds a preset threshold S (controlled by the user). With the intensity I and using the notation $L = \log_{10}(I)$, this is written as

$$\Delta L(u_k, t_k) = L(u_k, t_k) - L(u_k, t_k - \Delta t_k) = p_k S. \tag{2}$$

When the threshold is passed, the time stamp, the pixel position, and the polarity are emitted, and the current log intensity value is memorized for the monitoring.

Fig. 1 Pixel operation of an event camera. From top to bottom, the log intensity received by a pixel over time, the corresponding log intensity variations measured by the pixel, and the resulting events emitted by the pixel at instances in which the log intensity variations exceed the threshold values.

3 Event-Based Turbulence Mitigation

Most of the software-based turbulence mitigation approaches aim at recording multiple frames of the same scene. The turbulence will change over time, such that frames will suffer from varying distortions and blur. Turbulence mitigation attempts to distinguish between the temporal consistency of the actual scene and the random distortions. A particularly popular technique relies on the identification and selection of "lucky" space-time regions, regions that show locally the best image quality during a short period of time. With a standard camera, increasing the recording frame rate will then increase the chances of catching the instants for which the image quality is at its best. With its new sensing principle, the event camera has the ability to record short lucky instants without being limited by the integration time.

Here, we investigate how the above-mentioned concept of luckiness can be exploited; it is defined on a local neighborhood of intensities recorded synchronously, while using a camera that records only intensity variations asynchronously. The first step consists of estimating the continuous intensity image from the set of sparsely recorded intensity images and the continuous event stream. Using an iterative backprojection algorithm, we evaluate if the event stream carries information that can indicate if portions of the continuous intensity image stream are lucky.

We now detail our proposed method for how frames are reconstructed from events to be used in an image reconstruction algorithm.

3.1 Image Reconstruction from Events

The main purpose of image reconstruction from events is to transform the asynchronous event stream into classical intensity frames. The created frames benefit from the extended dynamic range of event-based recording and the ability to precisely choose the time at which the frame is reconstructed. This allows for recording information in highly dynamic scenes or information present for a short time, which would not be accessible with classical frame imaging as it has limited dynamic range, blind time, and integration time. However, relying purely on events to recreate an intensity frame also has shortcomings. Indeed, in the presence of flat regions or little motion, very few events may be triggered, resulting in a lack of information for those zones. Also, integrating incremental changes to recreate an intensity image will lead to the unavoidable

integration of error and a drift of the estimated intensity. To overcome these limitations, recent event sensors such as the DAVIS346 from IniVation have an architecture providing two readouts: an asynchronous event stream and an intensity frame resulting from the photocurrent integration during the exposure time.

The combination of regular intensity frames together with the asynchronous stream of events has been researched in different publications. Early approaches for image reconstruction relied on the integration of the contribution of each event.[23] More recently, joint estimation of optical flow and intensities manifold regularization[24] were proposed to address the issues with noise of the early approaches while offering real-time processing. A continuous intensity estimation based on a complementary filter was proposed by Ref. 25. The most recent techniques rely on deep learning with small and large architecture[26,27] providing state-of-the-art performance. The recent high speed, high dynamic range dataset[28] shows the growing interest for this subject's important for automotive applications.

Figure 2 shows the method used to recreate an intensity image $I(t)$ at a point in time t located between the moment at which the camera recorded intensity frames I_j and I_{j+1}. Figure 2(a) shows what the camera recorded: the intensity frames (with their integration marked in blue) and the event stream. For clarity, only a subset of the events is shown on the figure. As events are emitted when the camera records an intensity change, it should be possible to recreate the intensity at a time t by incrementally integrating the intensity changes corresponding to each event up to a specified point in time [Fig. 2(b)]. To work with an event camera, this integration needs to be adapted in two ways:

a. First, events are emitted for log intensity changes, and therefore the intensity frames I_j and I_{j+1} recorded by the camera need to be transformed to the log domain before integrating. To distinguish, we denote with I the frames containing linear intensities and with L the frames containing log intensities (also called log luminance). Frames recorded by the camera are noted with an index such as I_n, and frames reconstructed from events are considered a function of time $I(t)$.

b. The log intensity change corresponding to each event is not known precisely, and the relation with the threshold S [Eq. (2)] set by the user when operating the camera is unknown a priori. We therefore need to estimate the log intensity change corresponding to one event, a quantity that we call a contribution and denote $c(p_k)$. The contribution is independent of the event position in the image and only depends on its polarity [Eq. (3)]:

$$c(p_k) = \begin{cases} c_{\text{on}} & \text{if } p_k > 0 \\ c_{\text{off}} & \text{if } p_k < 0 \end{cases}.$$ (3)

The estimate for the intensity image reconstructed from events $I(t)$ is given by Eq. (4). It consists of the log intensity of the previous frame updated with the sum of the contribution
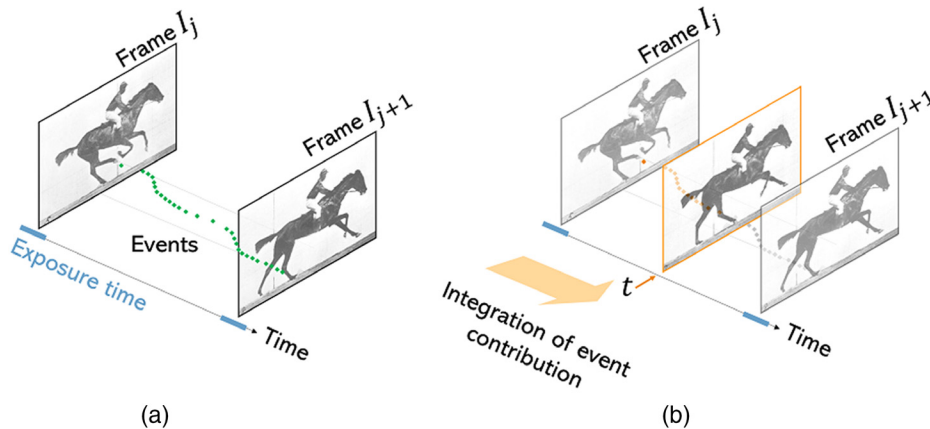
Fig. 2 Image reconstruction from events, (a) the intensity frames and the events and (b) the integration of each event contribution to reconstruct the image at time t.

of the N events occurring between frames I_j (time t_j) and I_{j+1} (time t_{j+1}) and transformed back to linear intensities using the exponential.

$$I(t) = \exp\left(\log_{10}(I_j) + \sum_{k=0}^{N} c(p_k)\right) \quad \text{for } \{e_k | t_j < t < t_{j+1}\}. \tag{4}$$

To estimate the optimal contribution of each event, we use the method proposed by Ref. 23. This method starts from the assumption that in each pixel the difference in log intensity ΔL

$$\Delta L = L_{j+1} - L_j = \log_{10}(I_{j+1}) - \log_{10}(I_j), \tag{5}$$

should correspond to the integrated contribution of all events between the two frames.

$$\Delta L = \sum_{k=0}^{N} c(p_k) \quad \text{for } \{e_k | t_j < t_k < t_{j+1}\}. \tag{6}$$

Assuming independent and normally distributed errors with zero mean, one can estimate a global spatially invariant event contribution (per polarity) by solving

$$\min_{x} \|Ax - b\|_2^2, \tag{7}$$

where the rows of matrix A denote the M pixels in the image for which events occurred between time t_j and time t_{j+1}:

$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ \dots & \dots \\ a_{M0} & a_{M1} \end{bmatrix}, \quad x = \begin{bmatrix} c_{on} \\ c_{off} \end{bmatrix}, \quad b = \begin{bmatrix} b_0 \\ b_1 \\ \dots \\ b_M \end{bmatrix}. \tag{8}$$

With the i'th location $i \in [0, M]$, $u_i = (x_i, y_i)^T$, and for the N events $\{e_k | t_j < t_k < t_{j+1}\}$ with respective position u_k

$$a_{i0} = \sum_{k=0}^{N} \delta(p_{on} - p_k)\delta(u_i - u_k), \tag{9}$$

$$a_{i1} = \sum_{k=0}^{N} \delta(p_{off} - p_k)\delta(u_i - u_k), \tag{10}$$

and

$$b_i = L_{j+1}(u_i) - L_j(u_i) = \log_{10}(I_{j+1}(u_i)) - \log_{10}(I_j(u_i)). \tag{11}$$

3.2 Background Reconstruction

The background reconstruction is shown in Fig. 3, and it implements (in software) an iterative back projection (IBP) scheme. The IBP aims at iteratively updating the estimate of the background B_{n-1} with the residue R_n of each new frame I_n. To compute the residue between the previous background estimate B_{n-1} and a new frame I_n, one needs to correct for the local motion and transform the previous estimate to B_{n-1} to the current recorded frame I_n using the warp $W(B_{n-1}, \Phi_n)$. The warp is a grid transformation derived from $\Phi_n(B_{n-1}, I_n)$ the dense optical flow map, which is updated with each new frame I_n based on Farnebäck.[29] The residue is then projected back to the background reference space using the inverse warp to update the estimate according to the weighting factor α.
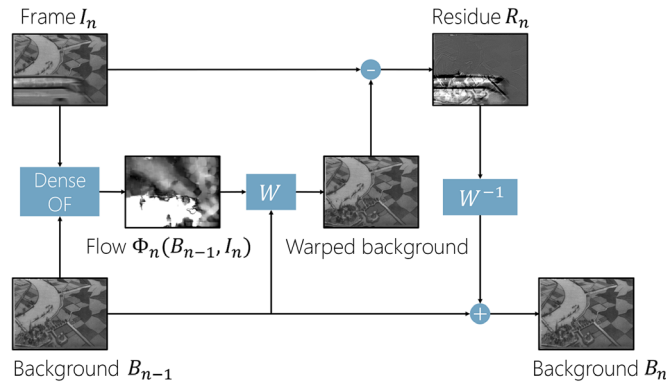
Fig. 3 Overview of the background reconstruction.

$$B_n = B_{n-1} + \alpha W^{-1}(R_n) = B_{n-1} + \alpha W^{-1}(I_n - W(B_{n-1})). \tag{12}$$

To start the process, one needs to pick an initial state. For our test setup, we pick the initial background image as the average of the first eight recorded intensity frames. The frame rate used to record the intensity frames depends on the exposure time chosen by the camera auto exposure function as this provides the best image quality with the given scene illumination (see Table 2 for the intensity frame integration time for each dataset).

As shown in Fig. 4, this setup allows for a comparison of different options to inject events in the process by:

i. Using events to generate frames at a higher temporal sampling than the native recorded frames. The fast temporal variations contained in the event stream and which are for classical frames integrated (if happening during the integration time) or lost (if happening during blind time) can improve the final image. We generate I_n^* from events using a fixed period (1 ms) such that multiple IBP loops are run until the next intensity frame.

$$B_n = B_{n-1} + \alpha W^{-1}(R_n) = B_{n-1} + \alpha W^{-1}(I_n^* - W(B_{n-1})). \tag{13}$$

ii. Using events to directly find lucky zones in an intensity frame. We use the event stream to quantify the luckiness of each pixel in each intensity frame. Assuming no camera motion, a given image location produces events if it contains an edge and if intensity variations induced by turbulence occur. For the image locations that are producing events, the pixel value of the intensity frame for which no event occurred during the frame integration is considered to have a higher chance of being lucky (sharp without motion) than blurred.
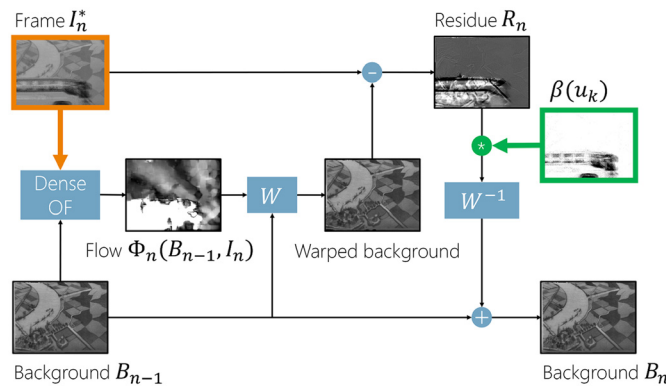
Fig. 4 Two options to integrate the events in the reconstruction process. Top left in orange, using frames recreated from events instead of recorded intensity frames, right in green using events to restrict background updates to zones that did not change (no events were emitted) during the frame integration time.

We expect that a lucky patch has a temporarily flat (or stationary) wavefront distortion, with a temporal first derivative that is also small. To remove the updates of pixels with a likelihood of being disturbed by varying blur or motion, we filter out the updates at the corresponding locations [Eq. (15)]. Therefore the update equation becomes

$$B_n = B_{n-1} + \alpha W^{-1}(\beta(u_k)R_n) = B_{n-1} + \alpha W^{-1}(\beta(u_k)(I_n - W(B_{n-1}))). \tag{14}$$

With the event filter

$$\beta(u_k) = \begin{cases} 0 & \text{if } \sum_{k=0}^{M} \delta(u - u_k) > 0 \text{ for } \{e_k | t_j < t_k < t_j + T\} \\ 1 & \text{otherwise} \end{cases}. \tag{15}$$

3.3 Features for Moving Object Detection

The events produced by the camera are created by the intensity changes caused by image motion of contrasted edges. The field of action recognition from the event stream offers a selection of features that aim at capturing the spatiotemporal behavior of an edge; see Ref. 30. The features are derived from the image of the time tag $T(u_k, t_k)$ (also called image of time surface) of the last event e_k at location $u_k = (x_k, y_k)^T$ with time stamp t_k (and independently from its polarity):

$$T(u_k, t_k) = \max_{j<k}\{t_j | u_j = u_k\}. \tag{16}$$

As shown in Fig. 5(a), for a rigid body motion due to a moving object (or the own camera motion), the edge travels through the field of view producing events at the same rate along the entire edge. The edge contrast does not change over time or space, and we expect few variations in the event production rate spatially (along the edge) and temporally.
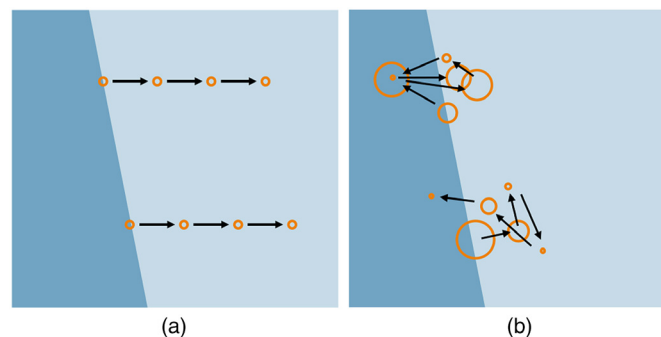
In the case of atmospheric turbulence, in Fig. 5(b), a static edge exhibits motion that is centered around the actual edge position and has a local random direction. The edge contrast varies randomly in space and time due to the variation of the refractive index. We expect bigger variations in the event production rate.

To distinguish between turbulence and rigid body motion, we update for each new event the image of the time tag $T(u_k, t_k)$, and we evaluate two simple features:

- The time difference between the new event and the time tag of the previous event at that location.

$$dt_k = t_k - T(u_k, t_{k-1}). \tag{17}$$

The time difference depends on the speed (component in the edge gradient direction) and the actual contrast of the moving edge, and therefore the distribution of dt_k produced by rigid body

Fig. 5 (a) Rigid body versus (b) turbulent motion, motion in black, blur level in orange circle.
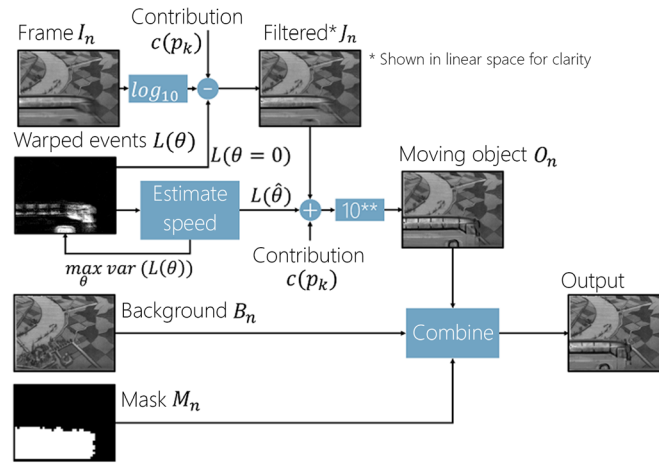
motion is expected to be compact and centered around the dt characterized by the moving object average speed.

i. To take the spatial variations of the motion into account, we also analyze the gradient of the time surface.

$$\nabla T_k = \left(\frac{dT(u_k, t_k)}{dx}, \frac{dT(u_k, t_k)}{dy} \right)^T.$$ (18)

Due to the random local motion orientation in turbulence, the distribution of ∇T_k is expected to be broader for turbulent motion than for rigid body motion.

The ability of those two features to distinguish between turbulence and moving object is investigated in Sec. 6.

3.4 *Moving Object Classification*

As shown in Fig. 6, the distinction between background and a moving object is implemented using a binary mask that indicates the pixels corresponding to the moving object. In our algorithm, a coarse mask is computed by splitting the intensity frame into subblocks (8×8 pixels). For each subblock, the algorithm counts the proportion of pixels in that frame for which the event-based feature [Eq. (17) or Eq. (18)] is below a predefined threshold. However, creating a mask based solely on event statistics would only give information about the edges of the moving object. As no events are triggered by flat surfaces of the moving object, one needs to propagate for each subblock the belief of being part of the moving object. To do that, we also compare the last background estimate with the current frame and compute the block-wise mean-squared-error (MSE) between the two images. The deviation is compared against the distribution of the error between the previous pairs of background and corresponding intensity frame that were affected by turbulent motion only. The algorithm thresholds the block-wise MSE image at N standard deviations (usually between 3 and 5) to create the map of candidate blocks. The mask derived from event statistics and the mask derived from the error between the background and the current frame are combined using a region growing algorithm. Starting from a seed (the event-derived mask), the algorithm iteratively integrates neighboring blocks if they are marked in the error mask. This strategy makes it possible to minimize the amount of false positive (MSE outliers due to strong turbulence incorrectly classified as the moving object) and false negative (flat zones of the moving object incorrectly classified as background).

Fig. 6 Overview of the moving object mask computation.

Fig. 7 Overview of the moving object reconstruction.

3.5 *Moving Object Reconstruction*

As shown in Fig. 7, to reconstruct the appearance, we first estimate the object velocity and subsequently use this to remove the motion blur on the object. The contrast maximization framework is a global approach that was developed for the estimation of the own camera motion. It relies, for a given set of events, on the maximization of contrast of an image of warped events (see Ref. 31 for details). The warping here consists of transforming a set of N events $\{e_k\}$ that were recorded during the frame integration time having the time stamp t_k and position u_k in the image into the set of warped events $\{e'_k\}$ corresponding to a reference time t_{ref} such that $u'_k(\theta) = H(u_k, t_k - t_{\text{ref}}, \theta)$, where H is the warping operator and θ is the velocity parameter. An image of the warped events $L(\theta)$ is created by integrating the polarity p_k of each event at its warped position:

$$L(\theta) = \sum_{k=0}^{N} p_k \delta(u - u'_k(\theta)).$$ (19)

Iterative nonlinear optimization algorithms are then used to solve the contrast maximization problem

$$\max_{\theta} Var(L(\theta)),$$ (20)

and find the optimal velocity parameter $\hat{\theta}$. Here, we apply this algorithm on the set of events $\{e_k\}$ that were classified as being produced by a moving object. The camera is considered to be static and the object motion to be linear in a direction different from the camera optical axis. This reduces the set of motion parameters to a two-dimensional velocity vector $\hat{\theta} = (v_x, v_y)^T$ that we estimate by solving Eq. (20).

To create the moving object image O_n, first, a filtered image J_n is created by subtracting each event contribution from the log intensity frame using the event original location [Eq. (21)].

$$J_n = \log_{10}(I_n) - \sum_{k=0}^{N} c(p_k)\delta(u_k).$$ (21)

Then, with the estimated object velocity $\hat{\theta}$, all events of the moving object are warped to the new location $u'_k(\hat{\theta})$ that corresponds to a reference time chosen during the frame integration (usually the frame mid-exposure time). The moving object image O_n is created by adding each event contribution to the base image J_n using that warped location $u'_k(\hat{\theta})$ and by exponentiating the result to transform back to linear intensities [Eqs. (22) and (23)].

$$O_n = \exp\left(J_n + \sum_{k=0}^{N} c(p_k)\delta(u'_k(\hat{\theta}))\right), \qquad (22)$$

$$O_n = \exp\left(\log_{10}(I_n) - \sum_{k=0}^{N} c(p_k)\delta(u_k) + \sum_{k=0}^{N} c(p_k)\delta(u'_k(\hat{\theta}))\right). \qquad (23)$$

In a final step, the turbulence corrected background and the motion corrected object appearance are then combined using the coarse (binary) mask. The final image replaces the background with the corrected moving object appearance in the zones marked in the mask.

4 Experimental Setup

For the research reported here, we used the DAVIS346 dynamic vision sensor from iniVation.[32]
Table 1 shows the specification of this camera.

To collect test material, we recorded video sequences through man-made, indoor turbulence. The experimental setup of this recordings is shown in Fig. 8. We produced turbulent air flows with a hot plate placed close to the camera and placed a flat test chart with enough contrasted content at a distance of about 4 m. To create footages with moving objects, we used a toy friction bus and a self-propelled train to pass next to the chart and generate motion. The footages were recorded with the Avenir lens 16 to 160 mm $f/2$ at 100 and 160 mm. The lens was focused using a Siemens star target placed in the object plane before recording. The turbulent air-flow pattern produced by the hot plate provides a reasonable approximation of operationally relevant turbulence. The resulting magnitudes of shifts and blurs in the intensity images as well as their correlation length scales appear consistent with previously recorded data in a field trial.[33]

We measured the average and standard deviation of the optical flow magnitude computed between the average intensity frame and each single intensity frame, Table 2 summarizes the results.

Even though we notice some variations in the turbulence magnitude, the variation of the event rate mainly depends on the contrast threshold used for the experiment. To record enough events, we set the contrast threshold as low as possible such that, for the sequences "Escher short" and "Escher train" and "Siemens star," the event rate was above 100 kevt/s while minimizing the

Table 1 Specifications of the DAVIS346 dynamic vision sensor of iniVation.

Resolution	346×260
Time resolution event stream	1 μs
Frames	Gray scale
Dimensions	$40 \times 60 \times 25$ mm^3 ($H \times W \times D$)
Weight	100 g (without lens)
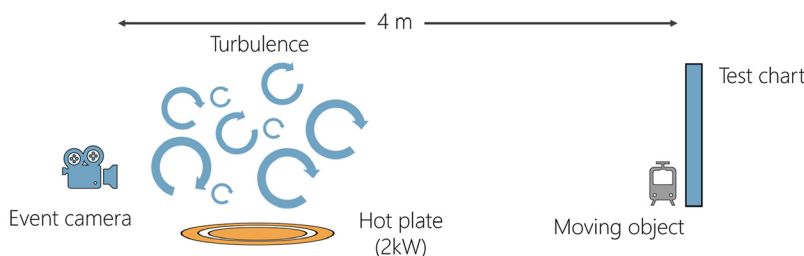
Fig. 8 Experimental setup.

Table 2 Summary of the recorded dataset.

Recording	Duration (s)	Frames	Integration time (ms)	Event rate (kevt/s)	Background motion (pix)	Object motion (pix)
Escher long	38.6	965	13	24	0.80 ± 0.49	—
Escher short	2.9	30	80	554	0.56 ± 0.33	—
Escher bus	4.4	104	24	45	0.96 ± 0.59	2.40 ± 4.79
Siemens star	5.2	44	100	870	0.79 ± 0.65	—
Escher train	6.7	58	97	131	0.51 ± 0.28	1.02 ± 1.91
Airport bus	3.8	79	29	86	0.73 ± 0.39	2.40 ± 6.06

amount of noise (event triggered on the flat surface). The latter could only be assessed visually using the DV Viewer preview and not quantitatively. In a future study, we will investigate the tradeoff between event stream sensitivity and SNR. In the case of a moving object being used in the footage, the event rate and background motion are measured on the frames that do not contain the moving object.

5 Results

5.1 *Background Reconstruction*

In our first set of experiments, we evaluate the quality of the reconstructed static background image. Using the setup described in Sec. 5, we imaged a fixed Siemens star, and no moving objects were used during the recording of the sequence. By switching off the hot plate, we could record the same target without turbulence, and a ground truth image was created from the average of all intensity frames recorded without turbulence. The footage with turbulence was processed using the background reconstruction algorithm variants described in Sec. 4. We used the PSNR between the ground truth and the output of each reconstruction method. We compared four variants:

   a. mean, simply averaging the intensity frame
   b. "ibp_frame," IBP using the recorded intensity frames [Eq. (12)]
   c. "ibp_evt," IBP using intensity frames reconstructed from events at 1 kHz [Eq. (13)]
   d. "ibp_noevt," IBP using the recorded intensity frames, while canceling updates for locations with events during each frame integration Eq. (14).

For each method, we logged the PSNR over time, and Fig. 9 shows its evolution.

The green dashed line shows the PSNR for each intensity frame. It is worth noting that the random quality variation of the raw frames over time. Averaging the raw frames (turquoise cross line) leads to a convergence of the quality after 10 to 12 frames. It prevents suffering from the temporarily significant quality drops but also prevents benefiting from the lucky frames. The reconstruction using IBP (pink dotted, blue diamond, and yellow continuous lines) shows a significant performance improvement for most frames. Surprisingly, as the pink dotted line shows, removing updates ($\alpha = 0$) for locations for which at least one event occurred during integration (which is thus supposed to be degraded) does not show a significant benefit over not using this residue selection (blue diamond line). Finally, creating virtual high-speed frames from the event stream at a high frame rate of 1000 fps (yellow continuous line) exhibits the best performance in all frames.

Figure 10 shows a side-by-side comparison of a central 100×100 pixel crop along with an intensity profile of the resulting image after 16 frames, and Fig. 11 plots the corresponding radial relative contrast result for each method. As expected, a simple averaging produces an image blurred by the accumulation of local shifts. The evolution of the relative contrast also shows
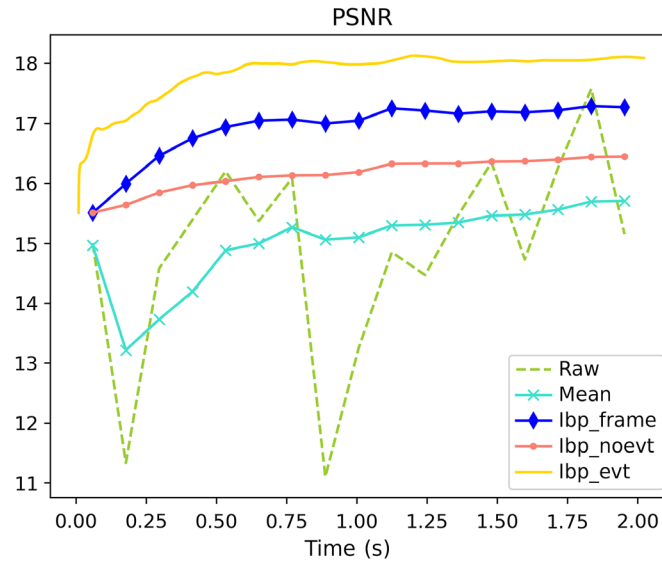
**Fig. 9** Evolution of the PSNR between ground truth and different background reconstruction approaches.
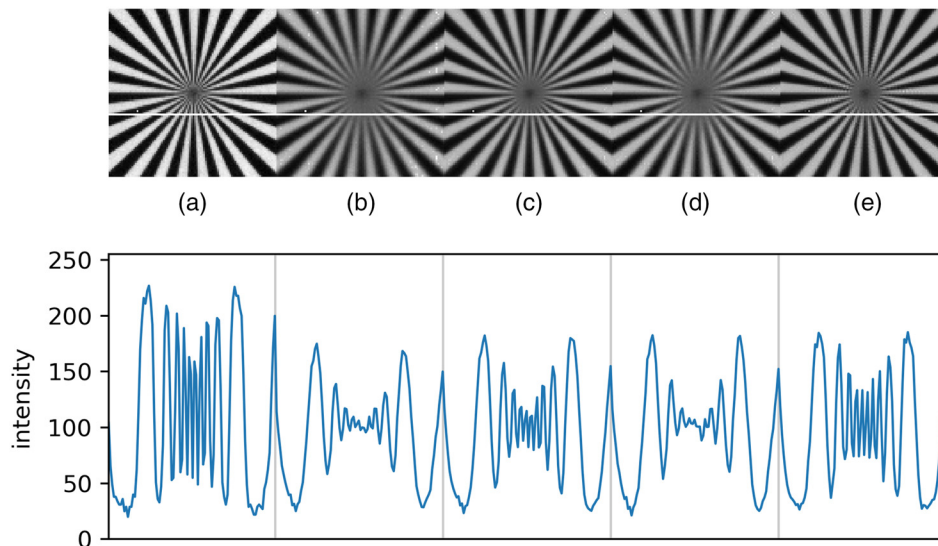
Fig. 10 Reconstruction result after 16 frames [(a) and (e)]: ground truth, mean, ibp_frame, ibp_noevt, ibp_evt.

the strong blur for the finer details. Using IBP with an event filter (not updating zones with events) tends to be counterproductive for a resolution gain. Indeed, regions with finer details produce events more frequently such that few or no updates can be done in the center region and the estimate remains at the initial value.

We also compared the result of the background reconstruction on a textured chart. Figure 12 shows the results for the Escher footage after 16 frames. When comparing the mean [Fig. 12(a)] with ibp_frame [Fig. 12(b)] and ibp_evt, [Fig. 12(c)], we notice again the amount of blur corrected by the registration step. When comparing ibp_frame with ibp evt, we also notice a gain on the local contrast (visible on the windows) and on the resolving power (visible on the field texture).

Using IBP on frames recreated from events instead of using the recorded intensity frames shows the highest performance. Even with a simple approach like the direct event integration
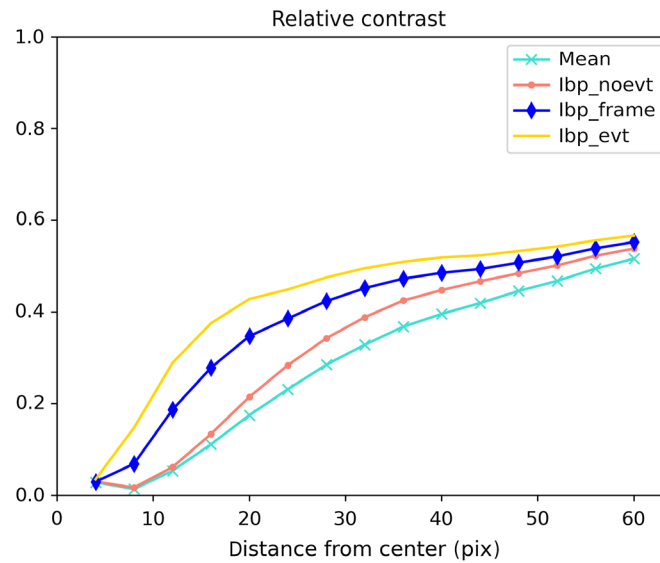
Fig. 11 Siemens star radial relative contrast profile.

(a)　　　　　　　　　(b)　　　　　　　　　(c)
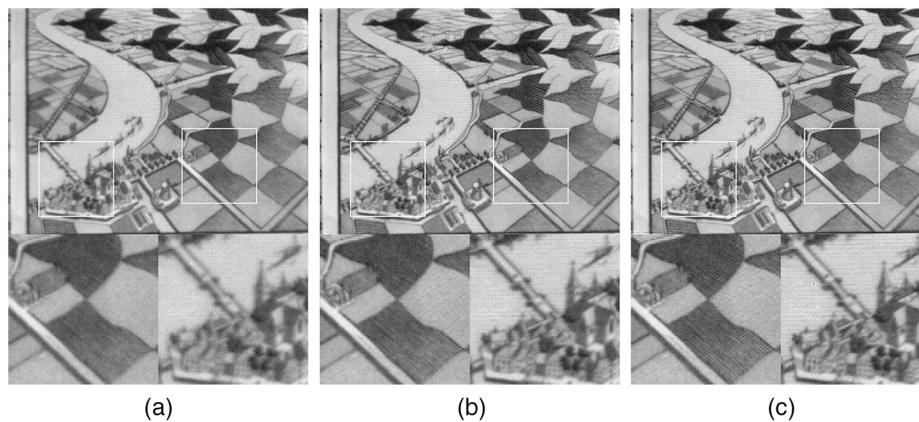
Fig. 12 Background reconstruction result after 16 frames on a textured chart [(a)–(c): mean, ibp_frame, ibp_evt]. For each figure, the white squares mark the position of the detail crops shown at the bottom.

based on a global contribution, we were able to transform the event stream into high-speed frames containing valuable information for the reconstruction.

When measuring on Siemens star, the IBP output shows a better PSNR and a higher resolving power when using the frames created from events. This gain is also confirmed on a target with texture variations. It shows that the information of the short time scale variations contained in the event stream outweighs the disadvantage of the imperfectly localized and noisy contribution.

5.2 *Moving Object Segmentation*

Next, we evaluated the moving object segmentation approach. In particular, we investigate

a. whether the event stream would allow us to distinguish between the moving object and turbulence using event-based temporal features only [Eqs. (17) and (18)]
b. the dependence of the moving object detection accuracy on object velocity.

Using our experimental setup, we recorded different moving objects (Fig. 13) passing in front of the test chart while having the heater produce turbulence.
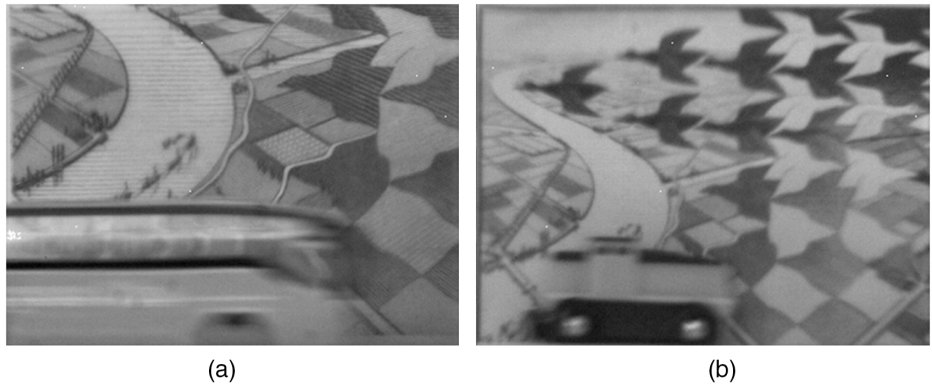
Fig. 13 Intensity frames from two sequences of a (a) fast bus and (b) slow train imaged through turbulence.
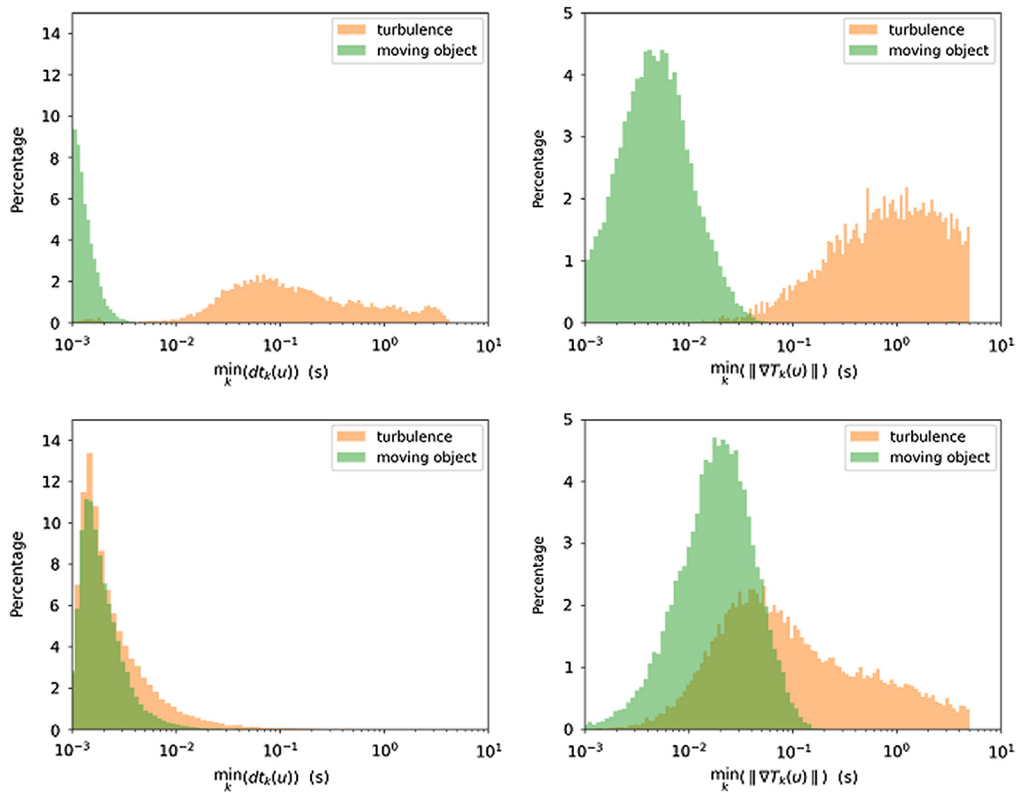
Fig. 14 Comparison of mink(dtk(u)) (left column) and mink(∇Tk(u)) (right column) for sequences with a fast-moving object (top row) and slow-moving object (bottom row).

Over the entire footage, we measured for each pixel location u the minimum value of the time interval between events $dt_k(u)$ and the minimum (non-zero) magnitude of the time surface gradient $\|\nabla T_k(u)\|$. Figure 14 compares the histogram between the bottom part of the image, where a moving object generated events, and the top part, where motion originates only from turbulence.

For the first feature $\min_k(dt_k(u))$, we see (left column) that the fast-moving object (top) generates a compact distribution around 1 ms, whereas turbulence generates a relatively wide distribution between 10 and 100 ms. The corresponding $\min_k(\|\nabla T_k(u)\|)$ (top right) has a wider distribution for the moving object centered around 4 ms, whereas the turbulence generates nearly no events with $\min_k(\|\nabla T_k(u)\|)$ below 10 ms. For a fast-moving object, both features appear

highly discriminative. On the bottom row events produced by a slow-moving object may not be distinguishable from events produced by the turbulence when using only $\min_k(dt_k(u))$. One can notice that the recording has a much higher event rate (partially due to a lower contrast threshold, which tends to be more sensitive and generate events more frequently). As visible in the bottom right part of the figure, $\min_k(\|\nabla T_k(u)\|)$ represents a more reliable metric to distinguish between a moving object and turbulence. The turbulence and moving object distribution overlap significantly. The distribution for the moving object is more compact and centered around 12 ms, whereas the turbulence spreads over a wider range and has a peak near 18 ms.

This experiment shows that both features allow for distinguishing between a moving object and turbulence when the moving object is moving fast relative to the turbulence speed. Nevertheless $\min_k(\|\nabla T_k(u)\|)$ performs better than $\min_k(dt_k(u))$ when the object motion has a similar magnitude as the turbulence motion and confirms that the local random motion orientation is captured by the events and can be used to distinguish between the two types of motion. Therefore we decided to use a moving object classifier based solely on $\min_k(\|\nabla T_k(u)\|)$. The experiment also shows that the latter is still prone to misclassification when the object speed differs less from the turbulent motion. This misclassification can only be avoided by integrating appearance-based features.

5.3 Comparison with State-of-the-Art Methods

Finally, we compared the output of the pipeline with the state-of-the-art methods described in Nieuwenhuizen,[8] Oreifej et al.,[6] Anantrasirichai,[5] and Halder.[14] We processed the intensity frames from our recorded sequences with the frame-based state-of-the-art methods (using the 2× super resolution mode for the approach from Nieuwenhuizen et al.[8]). We then compared the results with the ones produced with the proposed event-based mitigation using the same frames and the event stream. Figure 15 shows the comparison of the outputs on three different sequences with crops (on the bottom of each frame) made at various locations in each image (white box).
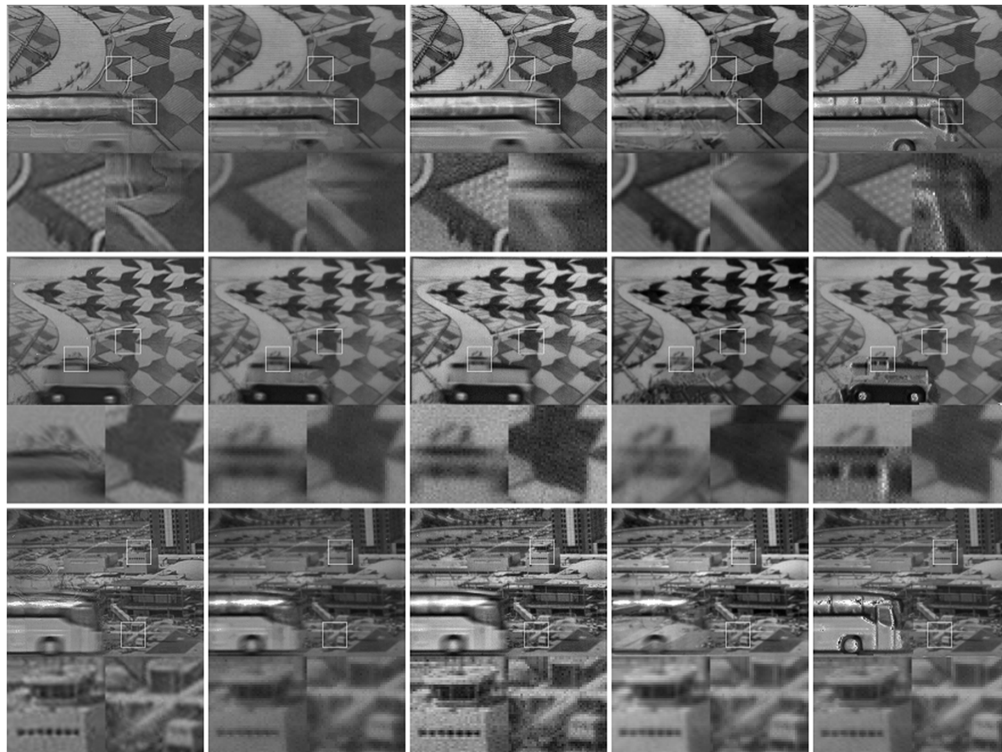
Fig. 15 Comparison with state-of-the-art methods on three recorded image sequences. From left to right, Nieuwenhuizen et al.,[8] Oreifej et al.,[6] Anantrasirichai et al.,[5] Halder et al.,[14] and event-based mitigation (proposed). For each recording, the white squares mark the position of the detail crops shown at the bottom.

The comparison above shows that the approach from Anantrasirichai et al.[5] provides the perceptually sharpest but also noisiest reconstruction of the static background. The proposed event-based processing ranks second in terms of sharpness, but resolves similar levels of detail. This can be observed on the stripe textures on the fields in the background of the first two sequences or on the preservation of the square window shapes on the control tower in the bottom left cutout of the bottom sequence. In most sequences, the described approach thus manages to use the lucky information contained in the event stream to produce an output comparable to the state-of-the-art methods in the static parts of the scene.

For slow-moving objects with little texture such as the Escher train sequence, the incomplete moving object segmentation from the event-based mitigation performs worse than Nieuwenhuizen et al.,[8] Oreifej et al.[6] and Anantrasirichai et al.[5] However, for the fast-moving bus, the event-based mitigation provides a similarly complete segmentation of the bus as Oreifej et al.[6] and Anantrasirichai et al.,[5] whereas the approach from Nieuwenhuizen et al.[8] exhibits mixing of foreground and background on the leading edge of the bus and background deformation above the bus in some cases. When comparing corrected frames, one can also observe the benefit of using events for the moving object reconstruction. By warping the events at a reference time, we were able to correct for the motion blur to refine the object boundaries and to enhance the edges of the moving object, which is visible, for instance, on the top bus sequence.

To quantify the performance of each method, we generated a ground truth image of the background using the footage recorded without turbulence, with the hot plate turned off. The ground truth image is used to compute the peak signal-to-noise ratio (PSNR)[34] and the structural similarity index measure (SSIM)[35] of the registered corrected frames produced by each method. The results are summarized in Table 3, and they show that the proposed method provides a similar quality as Anantrasirichai et al.[5] and Halder et al.[14]

To assess the improvement in resolving power that the turbulence mitigation algorithm attempts to deliver, we also computed the ratio, expressed here as a gain (in dB), between the power spectrum density (PSD) of the frames produced by each method and the PSD of the ground truth image. Figure 16 shows for each sequence the comparison of the frequency-dependent gain of each method when compared with the ground truth image. Table 4 summarizes these results across frames by reporting on the gain at Nyquist frequency (0.5 cycles per

Table 3 Comparison of background image quality with Nieuwenhuizen et al.[8] (N), Oreifej et al.[6] (O), Anantrasirichai et al.[5] (A), Halder et al.[14] (H), and our proposed event-based mitigation.

Sequence	PSNR (dB)					SSIM				
	N	O	A	H	Ours	N	O	A	H	Ours
Escher bus	22.5	20.9	21.5	21.9	**23.0**	0.74	0.65	**0.82**	0.71	0.78
Airport bus	23.9	22.7	22.5	23.5	**24.6**	0.87	0.82	**0.90**	0.84	0.88
Escher train	27.6	29.0	25.3	**29.4**	29.0	0.83	0.85	0.79	**0.86**	0.83

Note: The bold values are the best performing methods for these images.
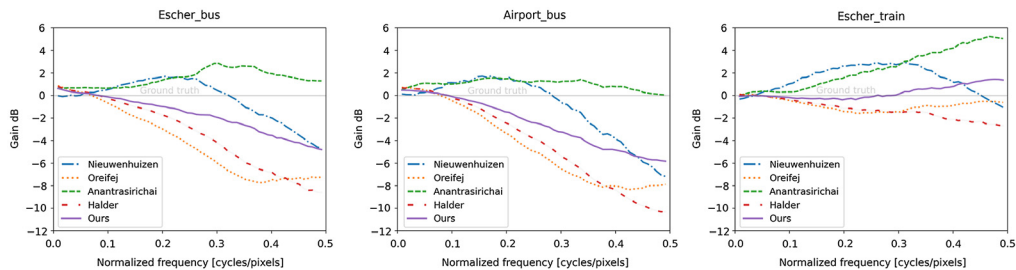
Fig. 16 Frequency gain comparison for the state-of-the-art methods on three recorded image sequences: (a) Escher bus, (b) Airport bus, and (c) Escher train.

Table 4 Gain at Nyquist frequency comparison between the methods of Nieuwenhuizen et al.[8] (N), Oreifej et al.[6] (O), Anantrasirichai et al.[5] (A), Halder et al.[14] (H), and our proposed event-based mitigation.

	Gain at Nyquist frequency (dB)				
Sequence	N	O	A	H	Ours
Escher bus	−4.83	−7.12	**1.29**	−8.52	−4.88
Airport bus	−7.48	−7.77	**−0.06**	−10.16	−5.92
Escher train	−1.05	−0.56	**4.89**	−2.52	1.28

Note: The bold values are the best performing methods for these images.

pixel), which can be seen as a measure sensitive to changes in resolving power. Figure 16 and Table 4 provide confirmation for our qualitative observation that our method has a higher resolving power than Halder[14] and Oreifej et al.[6] The approaches of Nieuwenhuizen et al.[8] and Anantrasirichai et al.[5] achieve higher PSDs. However, their positive gain indicates that the PSD is higher than the ground truth. This implies that they apply excess sharpening and therefore amplify the noise, without necessarily increasing the resolving power.

Even though the experiment shows that the event stream is useful for static background reconstruction, the main advantage resides in the moving object reconstruction. First, it helps in the segmentation of the moving object without being limited by the comparison of two integrated frames such as in an optical flow-based algorithm. Second, it improves the reconstruction of the moving object appearance and the refinement of its boundaries, which can provide important information in an operational situation.

6 Conclusions, Discussion, and Future Work

In this paper, we explored some of the advantages of the event camera for turbulence mitigation. First, we showed that the event stream contains information that may be used to reconstruct a fixed background disturbed by turbulence. When compared with integrated intensity frames, the event stream encodes high frequent variations that allow for a faster convergence of the image reconstruction toward a best estimate having a quality that is comparable to the state-of-the-art method on static scene elements. Then, we used the high temporal resolution of the sensor to build a motion signature and distinguish the fixed background disturbed by turbulence from an object moving in the same scene. The event stream carries a finer description of motion than integrated intensity frames. This allowed us to build accurate moving object masks without computing the optical flow between frames, being only limited by the presence of contrasted moving edges. Finally, we used the event stream to reconstruct the appearance of a moving object from a motion blurred frame. The three different aspects were combined into a processing pipeline. With an indoor experiment, the processing showed improved image reconstruction of moving objects through turbulence when compared with state-of-the-art methods and competitive performance on the static background.

This first study shows the strong potential of event cameras for turbulence mitigation. In future work, we will focus on collecting data with real turbulence and on improving the robustness of the method for variable scenarios. Unlike conventional cameras, which having automatic controls for setting the proper exposure, focus, and white balance, event cameras still lack algorithms to automatically select the best set of settings to record a given scene. We will also search for the best tradeoffs between event collection (and processing) and the final output quality to provide real-time processing. While this paper focused on application scenarios for which the camera was static, in the future we will aim at assessing the potential of the camera for scenarios with the camera moving, possibly under strong motion (vibration and high speed).

Finally, recent research shows increasing interest in deep learning for processing data from event cameras. By learning a richer imaging model, these new methods outperform classical approaches to recreate high-quality video from event streams. This axis of research will be a main topic for the next improvement on turbulence mitigation.

Acknowledgments

References

1. M. C. Roggemann, B. M. Welsh, and B. R. Hunt, *Imaging through Turbulence*, CRC Press, Boca Raton, Florida (2018).
2. A. W. M. Eekeren et al., "Turbulence compensation: an overview," *Proc. SPIE* **8355**, 83550Q (2012).
3. B. Fishbain, L. P. Yaroslavsky, and I. A. Ideses, "Real-time stabilization of long range observation system turbulent video," *J. Real-Time Image Process.* **2**, 11–22 (2007)
4. C. S. Huebner, "Turbulence mitigation of short exposure image data using motion detection and background segmentation," *Proc. SPIE* **8355**, 83550I (2012).
5. N. Anantrasirichai, A. Achim, and D. Bull, "Atmospheric turbulence mitigation for sequences with moving objects using recursive image fusion," in *25th IEEE Int. Conf. Image Process* (2018).
6. O. Oreifej, X. Li, and M. Shah, "Simultaneous video stabilization and moving object detection in turbulence," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 450–462 (2013).
7. X. Zhu and P. Milanfar, "Removing atmospheric turbulence via space-invariant deconvolution," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 157–170 (2013).
8. R. Nieuwenhuizen, J. Dijk, and K. Schutte, "Dynamic turbulence mitigation for long-range imaging in the presence of large moving objects," *EURASIP J. Image Video Process.* **2019**, 2 (2019).
9. J. Gilles, T. Dagobert, and C. De Franchis, "Atmospheric turbulence restoration by diffeomorphic image registration and blind deconvolution," *Lect. Notes Comput. Sci.* **5259**, 400–409 (2008).
10. M. A. Vorontsov and G. W. Carhart, "Anisoplanatic imaging through turbulent media: image recovery by local information fusion from a set of short-exposure images," *J. Opt. Soc. Am.* **A18**, 1312–1324 (2001).
11. M. Aubailly et al., "Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach," *Proc. SPIE* **7463**, 74630C (2009).
12. A. Eekeren et al., "Patch-based local turbulence compensation in anisoplanatic conditions," *Proc. SPIE* **8355**, 83550T (2012).
13. N. Anantrasirichai et al., "Atmospheric turbulence mitigation using complex wavelet-based fusion," *IEEE Trans. Image Process.* **22**, 2398–2408 (2013).
14. K. K. Halder, M. Tahtali, and S. G. Anavatti, "Geometric correction of atmospheric turbulence-degraded video containing moving objects," *Opt. Express* **23**, 5091–5101 (2015).
15. E. Chen, O. Haik, and Y. Yitzhaky, "Detecting and tracking moving objects in long-distance imaging through turbulent medium," *Appl. Opt.* **53**, 1181–1190 (2014).
16. B. Rueckauer and T. Delbruck, "Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor," *Front. Neurosci.* **10**, 176 (2016).
17. E. Mueggler et al., "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Rob.* **34**, 1425–1440 (2018).
18. H. Rebecq et al., "EVO: a geometric approach to event-based 6-DOF parallel tracking and mapping in real-time," *IEEE Rob. Autom. Lett.* **2**, 593–600 (2017).
19. H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," *Lect. Notes Comput. Sci.* **9910**, 349–364 (2016).
20. H. Rebecq et al., "EMVS: event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vision* **126**, 1394–1414 (2018).
21. G. Gallego et al., "Event-based vision: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).

22. N. Boehrer, R. Nieuwenhuizen, and J. Dijk, "Using event cameras for imaging through atmospheric turbulence," in *Commun. and Obs. Atmos. Turbulence* (2019).
23. C. Brandli, L. Muller, and T. Delbruck, "Real-time, high-speed video decompression using a frame- and event-based DAVIS sensor," in *Proc. IEEE Int. Symp. Circuits and Syst.*, pp. 686–689 (2014).
24. G. Munda, C. Reinbacher, and T. Pock, "Real-time intensity-image reconstruction for event cameras using manifold regularisation," *Int. J. Comput. Vision* **126**(12), 1381–1393 (2016).
25. C. Scheerlinck, N. Barnes, and R. Mahony, "Continuous-time intensity estimation using event cameras," in *Asian Conf. Comput. Vision*, Springer, Cham, Switzerland, pp. 308–324 (2018).
26. C. Scheerlinck et al., "Fast image reconstruction with an event camera," in *IEEE Winter Conf. Appl. Comput. Vision*, pp. 156–163 (2020).
27. H. Rebecq et al., "Events-to-video: bringing modern computer vision to event cameras," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 3852–3861 (2019).
28. V. Koltun, D. Scaramuzza, and H. Rebecq, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–26 (2019).
29. G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," *Lect. Notes Comput. Sci.* **2749**, 363–370 (2003).
30. X. Clady, S.-H. Ieng, and R. Benosman, "Asynchronous event-based corner detection and matching," *Neural Network* **66**, 91–106 (2015).
31. G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: loss functions for event-based vision," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.* (2019).
32. Invation, "Invation launches next generation event based dynamic vision sensor," Invation AG, 2018, https://invation.com/invation-launches-next-generation-event-based-dynamic-vision-sensor/ (accessed 24 Febraury 2020).
33. M.-T. Velluet et al., "Data collection and preliminary results on turbulence characterisation and mitigation techniques," *Proc. SPIE* **11159**, 111590Q (2019).
34. G. Padmavathi, P. Subashini, and P. K. Lavanya, "Performance evaluation of the various edge detectors and filters for the noisy IR images," in *2nd Int. Conf. Sens., Signals, Visualization, Imaging, Simul. and Mater.*, Baltimore (2009).
35. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).

Nicolas Boehrer received his MSc degree from the National Institute of Applied Sciences in Strasbourg in 2006. He is currently a research engineer in the Intelligent Imaging group of TNO in the Netherlands. Prior to TNO, he worked as a research engineer on image processing for the airborne imaging industry and for an action camera manufacturer. His research interests include structure from motion, surface reconstruction, and image restoration.

Robert P. J. Nieuwenhuizen is a research scientist at TNO. He received his MSc degrees in applied physics and management of technology from Delft University of Technology, the Netherlands, in 2011. In 2016, he received his PhD from Delft University of Technology on the topic of quantitative analysis for single molecule localization microscopy. Currently, his research interests include electro-optics, imaging through atmospheric turbulence, and electromagnetic signatures.

Judith Dijk received her MSc degree and her PhD in applied physics from Delft University of Technology. The topic of her PhD thesis was the perceptual quality of printed images. She currently works in TNO's Intelligent Imaging Department as a senior research scientist and a program manager. Her research interests include application of imaging technology for defense applications, image enhancement, development in imaging systems, information extraction from imagery, and application of artificial intelligence.