Real-Time Optical Flow for Vehicular Perception with Low- and High-Resolution Event Cameras

Vincent Brebion, Julien Moreau, Franck Davoine

Real-Time Optical Flow for Vehicular Perception with Low- and High-Resolution Event Cameras

Vincent Brebion, Julien Moreau, and Franck Davoine

***Abstract*—Event cameras capture changes of illumination in the observed scene rather than accumulating light to create images. Thus, they allow for applications under high-speed motion and complex lighting conditions, where traditional frame-based sensors show their limits with blur and over- or under-exposed pixels. Thanks to these unique properties, they represent nowadays an highly attractive sensor for ITS-related applications. Event-based optical flow (EBOF) has been studied following the rise in popularity of these neuromorphic cameras. The recent arrival of high-definition neuromorphic sensors, however, challenges the existing approaches, because of the increased resolution of the events pixel array and a much higher throughput. As an answer to these points, we propose an optimized framework for computing optical flow in real-time with both low- and high-resolution event cameras. We formulate a novel dense representation for the sparse events flow, in the form of the "inverse exponential distance surface". It serves as an interim frame, designed for the use of proven, state-of-the-art frame-based optical flow computation methods. We evaluate our approach on both low- and high-resolution driving sequences, and show that it often achieves better results than the current state of the art, while also reaching higher frame rates, 250Hz at 346×260 pixels and 77Hz at 1280×720 pixels.**

***Index Terms*—Machine vision, neuromorphic cameras, optical flow, real-time applications.**

I. Introduction

OVER the last decade, neuromorphic cameras have risen in popularity, due to their unmatched qualities: low latency, high dynamic range, no motion blur, and low energy consumption [1]. Thanks to their asynchronous response and their low latency, event sensors are by nature well suited for dynamic scenes analysis, including optical flow.

Optical flow depicts the per-pixel displacement in an image after a short period (e.g., between consecutive frames). It is usually computed using the brightness constancy constraint [2]: pixels intensities remain constant over short durations. Optical flow is a key enabler for many major applications, such as object detection and tracking [3], [4], motion estimation [5], visual odometry [6], [7], and image segmentation [8].

However, there is no direct translation for frame-based algorithms to event cameras. The sparse and asynchronous nature of their output constitutes a major paradigm shift.

Considering this, several event-based optical flow (henceforth EBOF) methods have been proposed. Former approaches have defined EBOF as a spatiotemporal point cloud matching problem, solved by plane-fitting algorithms [9], [10]. Others have made the choice to accumulate events during short time windows, to create dense image-like representations. They transfer the event-based problem into a frame-based one [11], [12]. The past few years have also seen the rise of neural networks to solve the EBOF problem [13], [14].

Presented work is motivated by the arrival of new, high-resolution event sensors. While low-resolution sensors have been the reference for the past decade, high-resolution neuromorphic sensors now start to be produced [15]. They offer increased visual details, essential for advanced driving applications. Applying the aforementioned EBOF approaches to these new cameras, however, reveals a common flaw: they were all designed with low-resolution event cameras in mind. For high-resolution sensors, they output degraded or incorrect optical flow results, and hardly handle their much higher throughput, resulting in long computation times. This last issue forbids the use of these methods in the real world for intelligent transportation system applications.

As an answer to these issues, we propose a novel optimized framework for computing *real-time*[1] EBOF for both low- and high-resolution event cameras. Our approach was originally inspired by the work of Almatrafi *et al.* [12], who introduce a simple and efficient way of densifying events in successive frame-based representations. In this article, we propose key contributions for making the method faster, more robust, and compatible with high-definition sensors:

- a specific pipeline-based architecture, for computing real-time optical flow using the events from low- or high-resolution neuromorphic sensors;
- the formulation of a novel dense "inverse exponential distance surface", that acts as the frame-based representation computed from the events, able to feed any image-based optical flow method;
- a coherent choice of algorithms and methods together for all the steps up to the fast frame-based state-of-the-art optical flow (with temporal smoothing to fit well with potentially noisy input events);
- we finally build and share a complementary high-definition event-based dataset of indoor sequences with high-speed movements, used as part of our evaluation.

Videos accompanying this article, showing results for both low- and high-resolution data, are available at https://youtube.com/playlist?list=PLLL0eWAd6OXBRXli-tB1NREdhBElAxisD.

Manuscript received...

V. Brebion, J. Moreau, and F. Davoine are with Université de technologie de Compiègne (UTC), CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex, France (e-mail: *firstname.lastname*@hds.utc.fr)

This work was supported in part by the Hauts-de-France Region, and by the SIVALab joint lab (Renault - UTC - CNRS).

[1]Considering a car at 120Km/h, if we tolerate to drive a distance of 1 meter to achieve perception analysis, it means a maximum latency of 30ms.

In complement, our dataset and all the source code linked to this article are available at https://github.com/vbrebion/rt_of_low_high_res_event_cameras.

II. Related Work

A. Optical Flow for Vehicular Perception

Optical flow is of great interest in the field of perception for intelligent vehicles, where multiple dynamics are omnipresent. In [3], Braillon *et al.* compare optical flow for the ground plane seen from a moving vehicle with theorical optical flow in order to detect obstacles. A similar obstacle detection problem is explored in [4], where the authors use their optical flow results to dynamically determine a model of the background motion, and extract obstacles that appear as outliers. Visual odometry using optical flow has also been explored in [6] and [7], where the authors of both articles use an optical flow tracking and feature matching method for estimating the ego motion of the vehicle. An extension of optical flow to 3D, known as "scene flow", has also caught up the attention recently, and approaches such as [16] aim at improving the detection of independently moving objects. Driving benchmarks such as KITTI [17], that includes both optical flow and scene flow, have also constituted major milestones for improving the state of the art.

B. Frame-Based Optical Flow

The problem of optical flow with traditional frame-based sensors has been deeply explored. Historical approaches relied on region-based matching techniques [18], or on the use of spatiotemporal derivatives of the input images [2]. More recent works have proposed extensions to these approaches, by defining pyramidal-based frameworks [19], [20], [21], or by introducing regularization terms to add robustness [22], [23]. The past few years have also seen the rise of neural networks, and their capability of learning from the data to generalize even in presence of noise and inconsistencies. Now, they surpass traditional handcrafted methods for optical flow and currently stand as the state of the art [24], [25], [26].

C. Event-Based Optical Flow (EBOF)

Two main approaches can be distinguished for EBOF.

On one hand, some authors use the events and all their properties, without accumulation into frame-based representations. Such a frameless approach is proposed in [9], where is employed a plane-fitting method on short temporal windows of events, to determine their motion in the visual scene. Other works [27], [28], [29] propose contrast maximization schemes as proxies for computing optical flow, by evaluating the sharpness of motion-compensated images of accumulated events. More recently, authors such as [30], [31] exploit spiking neural networks for a full bio-inspired EBOF estimation.

On the other hand, due to the great advances on optical flow estimation using traditional frames, other authors have proposed to build image-like representations from the event flow, to use them as input for these state-of-the-art methods. In [12], Almatrafi *et al.* accumulate events in short temporal windows using the distance transform, to create stable dense images designed to be used with any frame-based optical flow method. In [11], Zhu *et al.* also create images of accumulated events, but propose to compute a sparse optical flow by extracting visual features through the use of the Harris corner detector [32], and to track them using an expectation maximization algorithm. An alternative approach is proposed in [33], where the authors describe a surface matching approach on short time-shifted images of accumulated events (time surfaces), to evaluate their displacement. Recent works have also adapted proven neural network architectures for a use with images of events; [13], [34] proposed FlowNet-inspired [24] networks for inferring EBOF, while authors of [35] proposed a RAFT-inspired [26] one. Finally, Paredes-Vallés *et al.* [14] designed a light and real-time network, called FireFlowNet.

Apart from that, some methods exploit the capabilities of certain neuromorphic sensors to produce more than events, such as frames or inertial measurements. In [36], the flow of events is employed as a deblurring tool for the frames in highly dynamic scenes, allowing for a better optical flow estimation. In [37], Rueckauer *et al.* used the IMU integrated in the DAVIS240C camera to determine an exact optical flow estimation for pure rotational movements.

Still, none of these methods has considered the issue of computing optical flow with high-resolution event camera. And, very few of them ([37], [10], [14]) have been able to achieve real-time compatibility even for low-resolution inputs.

D. Our Orientation: Densifying Events

Based on the state of the art, methods able to compute high-definition EBOF in real-time are missing. We propose here to treat the event stream through a transformed dense representation[2] for the following reasons.

While the techniques detailed in the first paragraph of Section II-C do treat each event independently, they result in computationally expensive solutions, unable to reach real-time performances on standard CPUs and GPUs, as they perform an update stage for each new incoming event. This design model becomes even less viable for high-resolution neuromorphic cameras, which produce a much larger event throughput [1]. Furthermore, the information brought by each new event independently is very little; condensing them over short time windows allows for richer spatial updates [1]. Building image-like structures from events also facilitates the use of GPUs for fast parallel computations, which compensates the slight latency induced by the short duration of events accumulation. In addition, it allows for the use of state-of-the-art frame-based algorithms, leveraging the decades of research on traditional cameras. Finally, the qualities of event cameras remain. The accumulation time can be accurately adjusted thanks to the high time precision of the events, and the high dynamic range of these sensors allows to operate both for daytime and nighttime [1].

[2]Notice that we do not need and are not considering using image *reconstruction* methods for computing optical flow. Indeed, while approaches as the ones described in [38], [39], [40], [14] do provide interesting reconstruction results, they still show large imperfections, which would degrade the performances of frame-based optical flow methods.

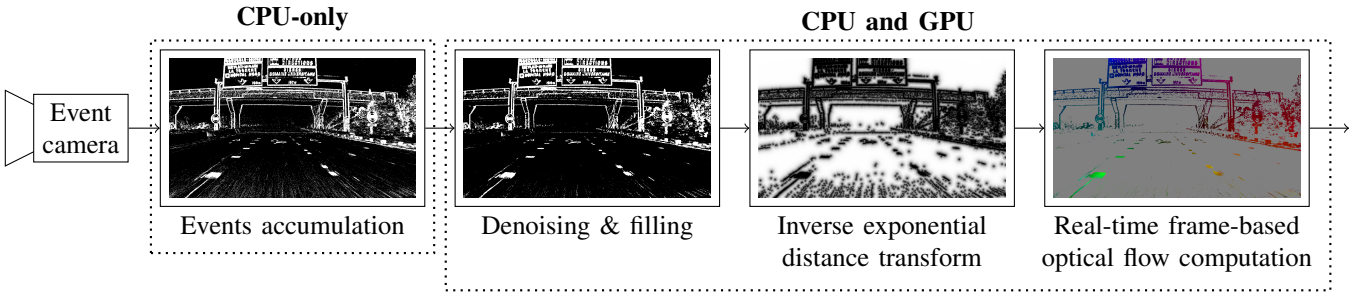

Fig. 1. Our event-based optical flow (EBOF) computation architecture, able to run in real-time with low- and high-resolution event cameras. Due to the pipeline architecture, all four blocks are independent parallel processes. Each block depicts the result it produces, for a sample driving sequence.

At the core of this problem, however, lies the choice of this dense frame-like representation. Accumulating the events during a short time window results only in an image of the scene edges, too simple for adequate optical flow computation. Proposing a densification method for these edge images is therefore a key issue for being able to use frame-based optical flow methods. Moreover, event-based cameras are noisy sensors; a filtering procedure is consequently required for ensuring the stability of these dense frame-like representations.

III. A Flexible Architecture

Following the problem formulation and the reason for the use of a dense representation to reach real-time performances, we detail in this section the novel framework we developed for computing real-time EBOF.

In order to optimize computational time and reach real-time performances, we propose the use of parallelized tasks through a pipeline architecture [41]. An illustration of this framework with example results of each step is available in Fig. 1. The following subsections will therefore detail how each block contributes towards obtaining the real-time EBOF.

A. Accumulation for Edge Images

The first component of our architecture is responsible for receiving and accumulating the events from the camera, in short temporal windows, to form "edge images". These binary matrices indicate whether or not each pixel produced at least one event during the accumulation time ΔT. By doing so, each edge image depicts a binary representation of the main edges of the moving objects in the visual scene, which can be used as a first stable medium for computing optical flow.

These edge images do not take into account the polarity of the events: as argued by Almatrafi *et al.* [12], and as we have experimented, both positive and negative events represent similarly the edges of the objects in the visual scene. The additional computational cost linked to treating polarities separately would be too expensive for little improvement in the final results. The choice of ΔT is also important and linked to the application: taking a too short ΔT will lead to edge images with too few events, resulting in an unstable appearance, while taking a too long ΔT will fail to capture clearly the movement of the objects by introducing blur.

Compared to other dense formulations from the literature (time surface [42], [33], motion-compensated images [27], reconstructed images [40]), our formulation has the benefit of keeping only the information required for frame-based optical flow estimation. Computationally speaking, this makes this solution extremely efficient, as each received event only needs to be placed in a buffer structure. In parallel, a second thread, triggered when the time window has expired, is responsible for collecting all the events from the buffer and creating the edge image, which is then sent for further processing.

Algorithm 1: Denoising

Inputs: An edge image E
The denoising threshold N_d
Output: The denoised edge image E_d
$E_d \leftarrow E$;
foreach *pixel index* $p \in E$ **do**
  if $E[p]$ *is an edge pixel* **then**
    $n_d \leftarrow$ count of edge pixels among the 4 direct neighbour pixels of p in E;
    if $n_d < N_d$ **then**
      $E_d[p] \leftarrow$ not an edge pixel anymore;

Algorithm 2: Filling

Inputs: A denoised edge image E_d
The filling threshold N_f
Output: The denoised and filled edge image E_{df}
$E_{df} \leftarrow E_d$;
foreach *pixel index* $p \in E_d$ **do**
  if $E_d[p]$ *is not an edge pixel* **then**
    $n_f \leftarrow$ count of edge pixels among the 4 direct neighbour pixels of p in E_d;
    if $n_f \geq N_f$ **then**
      $E_{df}[p] \leftarrow$ becomes an edge pixel;

B. Denoising and Filling

Event cameras generate a significant amount of noise, impacting the quality and stability of the edge images, which in return affects the final optical flow computation.

A solution could be to use one of the state-of-the-art denoising solutions of the literature [42], [43] during the accumulation step, that is, before creating the edge image.

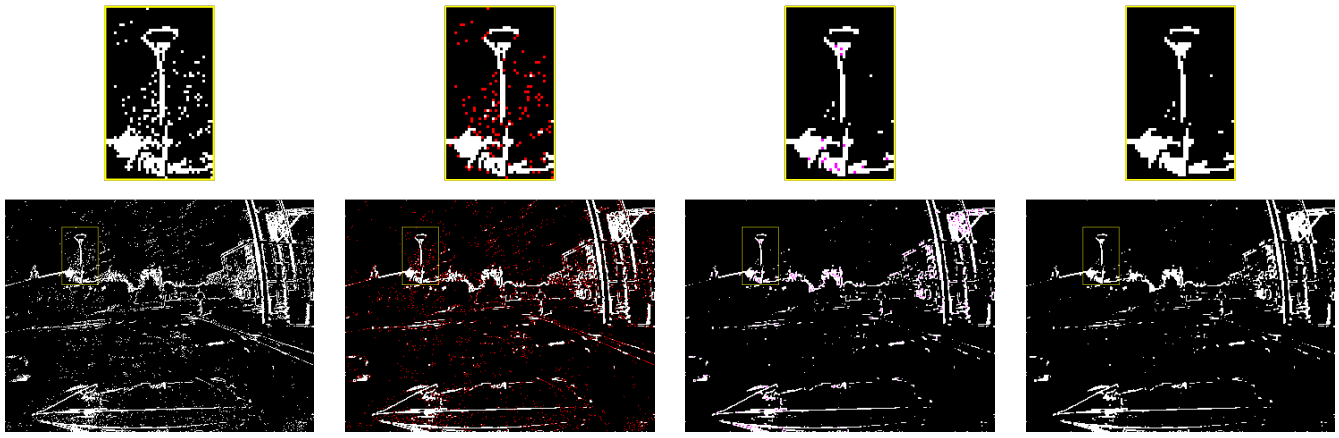

Fig. 2. Steps of the denoising and filling process, for a noisy edge image. From left to right: the original noisy edge image; the same image with edge pixels identified as noise in red ($N_d = 2$); the denoised edge image with the newly added pixels for the filling in pink ($N_f = 3$); the final denoised and filled edge image. A zoomed view of the street lamp (squared in yellow) is also provided for better visibility. Best viewed in color.

Doing so, however, would be computationally expensive, as the sparse and asynchronous nature of the events at that step makes hard to look for neighbour pixels states. Furthermore, many of these solutions were designed for low-resolution sensors, and translate difficultly to higher definition ones.

To circumvent this issue, we propose in this work a novel, fast yet efficient, method for discarding incorrect events. Our approach relies on applying denoising after the edge image creation. Proposed process is then similar to image filtering on the edge map and is computed in two steps: denoising and filling. In the first one, described in Algorithm 1, erroneous edge pixels are sought to be eliminated, by removing isolated events. The second step, on the contrary, aims at filling locations where an edge pixel is missing, but should have been produced by the camera, in order to help stabilizing the edge images. This process is further described in Algorithm 2. An illustration of both these steps is given in Fig. 2.

We underline here the importance of computing denoising and filling separately in this order, to avoid creating inconsistencies. Indeed, if the filling step was processed simultaneously with the denoising, then pixels that would later be discarded as noise could contribute to creating incorrect filling pixels, thus introducing new noise.

Denoising and filling thresholds, respectively N_d and N_f, depend on the event camera configuration, as it may give different noise profile. The aim of the denoising is to discard isolated pixels, that is, pixels with very few neighbours: $N_d = 1$ or 2 appear therefore as the best options. As can be seen in Algorithm 1, setting $N_d = 0$ disables the denoising. Then, the goal of the filling is to slightly stabilize the appearance of the edge images, by adding edge pixels in locations where there are enough neighbouring edge pixels to be confident that an edge pixel should have been produced: values of $N_f = 4$ or 3 are therefore the best compromise to add such pixels. As can be seen in Algorithm 2, setting $N_f = 5$ disables the filling. A general advice is to select $N_d < N_f$. A sensitivity analysis on N_d and N_f is done in Section IV-I.

Finally, while this formulation tends to remove small details from the edge images by considering them as noise (as can be seen for instance for the buildings on the right side of the edge images of Fig. 2), it actually helps obtaining more stable images, by extracting the main edges from the scene, and discarding superfluous textures.

Another advantage of this formulation lies in its simple and parallelizable formulation (the computation for each pixel is independent from the one of its neighbours). We implemented it using the GPU, to exploit its capabilities, and to relieve the CPU, so that it can undertake other complex tasks.

C. The Inverse Exponential Distance Surface

As the edge images are binary matrices (still after denoising and filling), they can hardly be used as is for computing optical flow with traditional frame-based algorithms. In order to make them viable for frame-based optical flow computation, densifying them through the use of the distance transform, as proposed by Almatrafi *et al.* [12], is an interesting baseline.

However, this approach has the main drawback of needing a near-perfect denoising, as a single noisy event can disrupt the appearance of the whole distance surface, as shown in the second row of Fig. 3. The computed distances are not bounded, meaning that the area of influence of each edge pixel can be infinite, and depends on the presence of other close neighbours. An answer to this problem could be to introduce an upper limit to the computed distances, to restrict the influence of an edge pixel to a fixed neighbourhood. This solution, however, would introduce a non-smooth transition in the distance transform function. It can become an issue for the gradient computation on distance surfaces, often used as part of the optical flow estimation.

Another issue of the approach of Almatrafi *et al.* also appears when distinct objects come close to each other: their edges tend to merge together in the resulting distance surface, making the individual objects indistinguishable, such as in the last row of Fig. 3. This phenomenon can lead to incorrect optical flow results, especially when a block-matching or image warping formulation is employed, due to the lack of texture on the produced image. Giving more emphasis to the pixels directly surrounding the edge pixels would help creating

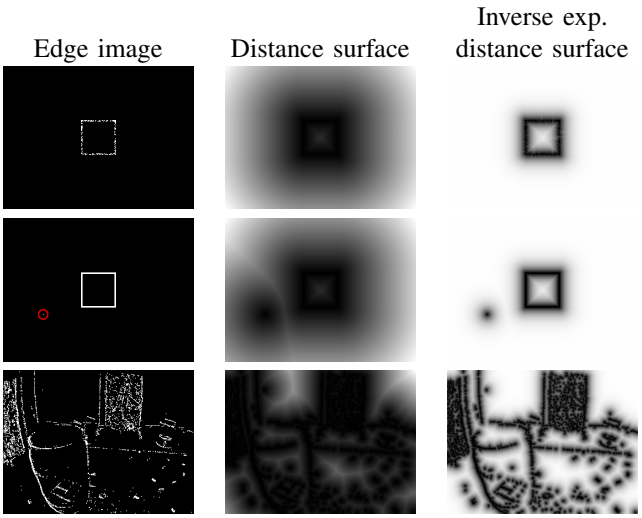

Fig. 3. Comparison between the original distance surface and proposed inverse exponential distance one (with $\alpha = 2$). From top to bottom: a simulated square with 50% of its pixels randomly removed, the same square with an added single pixel of noise (circled in red), and an indoor flying scene from the MVSEC dataset [44].

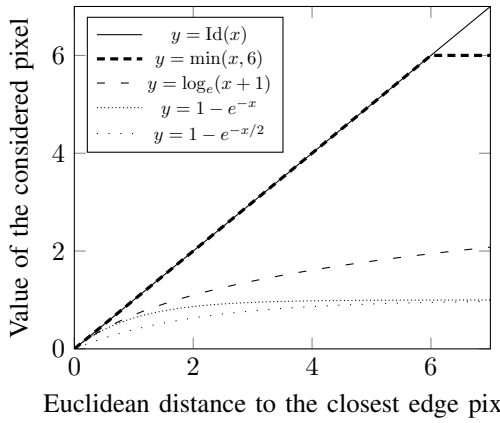

Fig. 4. Values of the distance transform as a function of the distance to the closest edge pixel. Curves represent the original distance transform, the same with an upper bound set to 6px, the natural logarithm version, and our inverse exponential formulation, with α set to 1 and 2 respectively.

distance surfaces with more prominent object edges, limiting this merge issue. A solution could be to employ a function with a logarithmic shape.

To solve jointly both these issues, we propose in this work a novel inverse exponential distance transform formulation:

$$d_{\text{exp}} = 1 - e^{-d_{\text{Euc}}/\alpha}, \tag{1}$$

where d_{exp} is the inverse exponential distance, d_{Euc} the Euclidean distance to the closest edge pixel (i.e., the original distance transform), both expressed in pixels, and α a spreading parameter. Fig. 4 compares the aforementioned functions over the distance to the closest edge pixel. As can be seen through the plot, the main advantage of our inverse exponential formulation is that, while close to a logarithmic formulation, each edge pixel also has a restricted influence area, after which the values saturate to a value of 1.

The spreading parameter α can be used to determine the size of the neighbourhood influenced by each edge pixel. This parameter conditions the appearance of the distance surface, and regulates the remaining imperfections of the edge images. A low value for α restricts the area of influence of each edge pixel to only its close neighbours. It limits the influence of the noise on its appearance, but makes the distance surface less stable and more prone to variations. On the contrary, selecting a higher value for α has the opposite effect: variations of the appearance of the various objects in the scene are well compensated, but noisy events have a more important effect. α can be rewritten from equation (1) as a function of the wanted distance of saturation for d_{Euc}, namely, d_{sat}, in pixels:

$$\alpha = -\frac{d_{\text{sat}}}{\ln(\varepsilon)}, \tag{2}$$

with $\varepsilon = 1 - d_{\text{exp}}$, $\varepsilon > 0$. ε is formulated so as to represent the gap between d_{exp} and the saturation value of 1. Saturation is therefore reached when this gap ε is as small as possible, i.e., $\varepsilon \to 0$. Since we work in the discrete domain, if the inverse exponential distance surface is coded on 8 bits (values ranging from 0 to 1 are represented by values between 0 and 255), then saturation is reached when $\varepsilon = \frac{1}{255}$. Integrating this value in (2) results in the final formulation of α as a function of d_{sat}:

$$\alpha \approx \frac{d_{\text{sat}}}{5.541}. \tag{3}$$

The sensitivity of d_{sat} is studied in the analysis Section IV-I.

The interest of our inverse exponential distance surface formulation is illustrated in Fig. 3, a side-by-side visual comparison with the original distance surface. Inverse exponential formulation compensates the missing pixels similarly to the original distance surface, while limiting the impact of noisy edge pixels. Also, this formulation displays more distinct edges and keeps objects texture, especially visible in the real complex scene represented in the last row (note how the board and the barrel keep a clear appearance using inverse exponential formulation, while they are hardly distinguishable with the original distance surface).

Regarding the implementation of the distance transform, we employed the fast solution described by Coeurjolly *et al.* [45], slightly modified to incorporate our inverse exponential formulation. The choice of this method was especially motivated by its optimized formulation, allowing for large parallelization; this block of our pipeline was therefore implemented on GPU.

D. Selected Frame-Based Optical Flow

The final block of our architecture is the computation of the optical flow itself. Since the previous steps led to dense image-like structures from the flow of events, any state-of-the-art frame-based optical flow method could be used here.

Within the scope of this article, we selected the approach of Adarve *et al.* [21]. Their method is based on an update-prediction architecture, similar to the one of Black [46], which predicts optical flow using an image warping process, and temporally propagates the optical flow estimations using an incremental framework. Multiple update-prediction loops are stacked as a pyramidal structure, enabling the capture of both

large and fine displacements in the images. Their method was designed for a fast and accurate estimation of the optical flow field, and is implemented on GPU.

The choice of this method as our optical flow computation solution was mainly guided by their use of a predictive filter-based formulation, which, beyond enabling real-time compatibility on GPU, allows for a temporal smoothing of the flow. This property brings stability and robustness to the overall optical flow, given its memory effect, which is beneficial given the sometimes unstable nature of events.

Finally, while this method returns a dense optical flow, covering the whole distance surface image, we then restrict it to the edge pixels of the denoised edge image. Indeed, by nature, events encode sparse information, detailing the pixels for which a change in luminosity was observed. The densification produced by the use of the inverse exponential distance transform differs from an inference of missing data. It is only intended for creating texture and smooth transitions, which are necessary to determine the optical flow.

IV. Evaluation

A. Setup

The implementation of our method was made using ROS Noetic, in C++11 and CUDA 11.4, combined with the use of the OpenCV 4.2 library. Both the implementation and the evaluation phases were conducted on a HP ZBook 17 G6 laptop, with an Intel i9-9880H CPU, a NVIDIA Quadro RTX 5000 GPU, 64 GB of RAM, and using Ubuntu 20.04.

Regarding the parameters, two configurations were used, respectively for low- and high-resolution input data.

For the low-resolution data (346×260) of the MVSEC dataset [44], [13], on one hand, we were restricted to use a temporal window of size $\Delta T = 1$ frame[3] for a fair comparison with the other state-of-the-art methods using this time window [13], [10], [31], [34], [35], [14], [39], [47], [48], [49]. For the denoising and filling, we set $N_d = 1$ and $N_f = 4$, due to the high noise in these recordings. The inverse exponential distance transform was configured with $\alpha = 1.08$ (so that $d_{\text{sat}} = 6$px, see (3)). Finally, the optical flow computation library [21] was configured with 3 pyramidal layers, with their regularization weights set respectively to 50.0, 250.0, and 500.0, and with 50, 25, and 5 smooth iterations per layer.

For the high-resolution data (1280×720), on the other hand, a temporal window $\Delta T = 15$ms was used, to better capture the movements. $N_d = 2$ and $N_f = 3$ were empirically chosen, as the best compromise between removing noise and keeping the main edges. $\alpha = 1.08$ ($d_{\text{sat}} = 6$px) also proved to be the more adequate, allowing to keep the scene details, while compensating potential imperfections. The optical flow library was configured with 3 layers, with regularization weights all set to 500.0, and with 20 smooth iterations per layer.

B. Datasets

As part of the evaluation of the proposed methods, four datasets are going to be used in the following subsections. The first one is the low-resolution MVSEC dataset [44], [13], which is currently the only event vision dataset with real data that includes ground truth optical flow. It will serve as the basis for comparison with other state-of-the-art methods. For high-definition data, three complementary datasets are used: the 1 Megapixel Automotive Detection Dataset [50], for a deep evaluation on daily driving sequences; a 20-minute-long driving sequence recorded by Prophesee[4], for visual comparison with the current frame-based state of the art; and a novel high-speed high-definition event-based indoor dataset we recorded as part of this article, to demonstrate the accuracy of our EBOF even under large motions. A summary of these datasets is given in Table I.

C. Evaluation Metrics

In order to evaluate the quality of our optical flow results, three metrics are used as part of this article.

The first two ones, the percentage of outliers and the Average Endpoint Error (AEE), are traditional optical flow metrics, used for instance in the KITTI benchmark [17]. The percentage of outliers reports the number of pixels for which the error is above 3px and 5% of the magnitude of the flow vector. The AEE is a raw error measurement on both orientation and magnitude of the flow, computed as following:

$$AEE = \frac{1}{N} \sum_{i=1}^{N} |v_i - u_i|, \tag{4}$$

where N is the total number of flow vectors, u_i the i^{th} estimated flow vector, and v_i its ground truth equivalent.

However, to this day, no complex high-resolution event-based dataset with a ground truth for optical flow exists. In order to still leverage high-resolution datasets, for instance Prophesee's 1 Megapixel Automotive Detection dataset [50], and to provide a quantitative evaluation of our EBOF results, we adopt the Flow Warping Loss (FWL) metric proposed by Stoffregen *et al.* [39]. The principle is to compensate and accumulate each raw event (considering its polarity and timestamp) by its computed optical flow, in order to recreate an image of compensated events at a reference time t. If the optical flow is accurate, compensated events superimpose in the same pixel position, producing sharp edges. The FWL then evaluates the sharpness of the produced image, compared to the one where events are not compensated:

$$\text{FWL} = \frac{\sigma^2(I_{\text{comp}})}{\sigma^2(I_{\text{uncomp}})}, \tag{5}$$

where σ^2 is the image variance function, I_{comp} the flow-compensated image of events, and I_{uncomp} the original uncompensated image. By doing so, a final FWL value greater than 1 is sought to be obtained, as it indicates that the computed flow is better than the "zero flow" (uncompensated) reference.

Finally, in the EBOF illustrations in the following subsections, and in the videos associated to this article, the pixels where no event was received are colored in medium gray.

[3]In MVSEC, $\Delta T = 1$ frame $\simeq$ 32ms for "Indoor flying" sequences, $\simeq$ 22ms for "Outdoor day" ones, and $\simeq$ 97ms for "Outdoor night" ones.

[4]https://www.prophesee.ai

TABLE I
COMPARISON BETWEEN EVENT-BASED DATASETS USED IN THIS ARTICLE.

Dataset	Resolution	Scenes	Ground truth optical flow	Frames available	Conditions
MVSEC	346×240 (low)	Vehicular, drones	Partial	Yes	Day, night
1 Megapixel Automotive Detection	1280×720 (HD)	Vehicular*	No	No	Day, varying lighting and weather
20-minute-long driving sequence	1280×720 (HD)	Vehicular*	No	Yes	Day, single long sequence
Our high-speed event dataset	1280×720 (HD)	Indoor	No	No	Very fast and erratic motions

*Both datasets contain diverse driving environments (city, highway, suburbs, countryside, villages), which implies various traffic density and the presence of pedestrians or other road users (cyclists, etc). As such, they are particularly representative of daily scenarios a driver may encounter.

D. Ablation Studies

To show the validity of our contributions, we also conducted evaluations with ablations or distance surface alternatives:

Ours$_{NDF}$ – full proposition without denoising and filling;
Ours$_{DS_L}$ – linear distance transform, $y = \mathrm{Id}(x)$;
Ours$_{DS_LB}$ – upper-bound distance transform (set to 6px, equal to the used d_{sat} value with proposed inverse exponential formulation), $y = \min(x, 6)$;
Ours$_{DS_Log}$ – logarithmic distance transform, $y = \log_e(x+1)$.

Fig. 4 illustrates the shape of these variants.

E. Evaluation on the MVSEC Dataset

We evaluated our EBOF method on the low-resolution (346×260) MVSEC dataset proposed by Zhu *et al.* [44], [13]. Despite several shortcomings highlighted by its authors — namely, errors created by moving objects, an approximate synchronization, and the use of default biases — this dataset remains the main reference for evaluating EBOF results on complex real-life sequences. Therefore, we present in Table II our error measurements on this dataset, compared to other reference methods from the literature (both non real-time and real-time capable). We also compare them to a "zero flow" reference, i.e., error measurements when the estimated optical flow is set to a null vector field. Note that, similarly to other authors such as [13], [39], for "Outdoor" sequences, we ignored the pixels where the hood of the car is visible. In the dataset, these pixels contain incorrect ground truth values.

From these results, we obtain AEEs in the order of one pixel, except for nighttime sequences, where the longer accumulation time of $\Delta T \simeq 97$ ms results in greater magnitudes of errors. Our AEE results are remarkably always close to or even better than all the non-real-time state-of-the-art approaches (EV-FlowNet$_{\text{HQF}}$ [39] notably). We display vastly better results than FireFlowNet [14], which is our main comparison point when it comes to fast EBOF methods.

Outlier percentages are also very low, only increasing for the nighttime driving scenes. However, as noted by Ye *et al.* [49], MVSEC ground truth flow is valid only for static world; the moving objects, numerous in the nighttime scenes, could not be kept in the reference, creating errors in the ground truth.

When compared to the ablated versions of our method, it can be seen that the "No denoising" *Ours$_{NDF}$* performs better for the indoor sequences, where the lighting of the scene is controlled, and the noise therefore less prominent. In that case, the denoising and filling step will mostly tend to eliminate small texture details from the scene, which could in reality be kept to improve the stability of its appearance. On the outdoor sequences, on the contrary, our denoising shows its importance, as the noise generated by the environment becomes much more essential to discard to obtain accurate optical flow results. Regarding the distance surface alternatives, they all display worse results than proposed inverse exponential, both for indoor and outdoor sequences; the original linear distance surface *Ours$_{DS_L}$*, notably, displays here the worst results, even worse than the zero flow baseline in some sequences.

Despite the presence of a ground truth in MVSEC dataset, we also computed the FWL metric, in Table III. Our results consistently surpass the value of 1, indicating an optical flow estimation better than the zero flow reference. Most importantly, they surpass those of EV-FlowNet$_{\text{MVSEC}}$ [13] and EV-FlowNet$_{\text{HQF}}$ [39] in most of the sequences. While these results may sometimes slightly contrast with those presented in Table II, from our understanding, they further underline the inconsistencies in the ground truth flow of the MVSEC dataset, but still demonstrate the high accuracy of our approach compared to non-real-time ones.

Regarding the FWL comparison with the alternative methods, proposed denoised inverse exponential formulation displays the best results for all sequences. The *Ours$_{NDF}$* version consistently performs worse, as the small details in the scene, not discarded as noise here, compensate more difficultly than the main edges and lower the results yielded by the FWL metric. Once again, the alternative distance surface formulations all provide worse results than the inverse exponential one. However, it should be underlined here that the "linear-bound" *Ours$_{DS_LB}$* method displays results closer to the inverse exponential distance surface than anticipated.

Finally, we present in Fig. 5 qualitative optical flow results for sequences from the MVSEC dataset. It can be seen that our EBOF is visually close to the reference. Limitations in the ground truth of the dataset can also be observed: the hood of the car is not taken into account in the ground truth of the outdoor sequences (second and third rows), and the moving vehicle at the right of the car in the last row is associated to an incorrectly smoothed ground truth flow.

F. Evaluation on the 1 Mp Automotive Detection Dataset

The arrival of high-resolution neuromorphic cameras means that a more thorough evaluation including these new sensors has to be conducted. In the context of this article, the 1 Megapixel Automotive Detection dataset [50] from Prophesee — while initially intended for automotive object recognition purposes — appears as the most complete, publicly available

TABLE II
RESULTS ON THE MVSEC DATASET. BOLD INDICATES THE BEST RESULTS FOR NON REAL-TIME AND REAL-TIME VERSIONS SEPARATELY.

Sequence	Indoor flying 1		Indoor flying 2		Indoor flying 3		Outdoor day 1		Outdoor day 2		Outdoor night 1		Outdoor night 2		Outdoor night 3	
	AEE	% outliers	AEE	% outliers	AEE	% outliers	AEE	% outliers	AEE	% outliers	AEE	% outliers	AEE	% outliers	AEE	% outliers
Zero flow	1.71	8.9	3.03	40.2	2.53	29.1	1.46	5.1	1.70	13.0	5.41	63.8	6.62	73.7	7.2	77.1
Non Real-Time																
EV-FlowNet$_{MVSEC}$ [13]	1.03	2.2	1.72	15.1	1.53	11.9	*0.49*[*]	*0.2*[*]	-	-	-	-	-	-	-	-
EV-FlowNet$_{MVSEC}$ (updated)[†]	0.85	0.9	1.29	7.5	1.13	5.3	0.56	0.4	-	-	**1.90**	**18.6**	**2.26**	**21.9**	**2.13**	**20.4**
EV-FlowNet$_{EST}$ [47]	0.97	0.9	1.38	8.2	1.43	6.5	-	-	-	-	-	-	-	-	-	-
EV-FlowNet$_{HQF}$ [39]	**0.56**	1.0	**0.66**	**1.0**	**0.59**	**1.0**	0.68	1.0	**0.82**	**1.0**	-	-	-	-	-	-
EV-FlowNet$_{DR}$ [14]	0.79	1.2	1.40	10.9	1.18	7.4	0.92	5.4	-	-	-	-	-	-	-	-
Zhu *et al.* [48]	0.58	**0.0**	1.02	4.0	0.87	3.0	0.32	**0.0**	-	-	-	-	-	-	-	-
Spike-FlowNet [34]	0.84	-	1.28	-	0.87	-	0.49	-	-	-	2.77[‡]	-	-	-	-	-
Nagata et al. [33][§]	*0.28*	-	*0.42*	-	*0.38*	-	*0.26*	-	*0.35*	-	*0.33*	-	*0.36*	-	*0.36*	-
Spiking EV-FlowNet [31]	0.60	0.5	1.17	8.1	0.93	5.6	0.47	0.2	-	-	-	-	-	-	-	-
Gehrig *et al.* [35]	-	-	-	-	-	-	**0.24**	**0.0**	-	-	-	-	-	-	-	-
Real-Time																
ECN$_{masked}$ [49][¶]	-	-	-	-	-	-	*0.30*	*0.0*	-	-	*0.53*	*1.1*	*0.49*	*1.0*	*0.49*	*1.1*
Akolkar *et al.* [10]	1.52	-	1.59	-	1.89	-	2.75	-	-	-	4.47	-	-	-	-	-
FireFlowNet [14]	0.97	2.6	1.67	15.3	1.43	11.0	1.06	6.6	-	-	-	-	-	-	-	-
Ours	0.52	**0.1**	0.98	5.5	0.71	2.1	**0.53**	**0.2**	**0.74**	**1.2**	**2.91**	**30.6**	**3.45**	**39.1**	**3.62**	**39.8**
Ours$_{NDF}$	**0.49**	**0.1**	**0.92**	**4.6**	**0.68**	**1.6**	0.54	0.4	0.75	1.3	2.99	31.8	3.56	40.4	3.70	40.9
Ours$_{DS_L}$	1.81	16.4	2.54	26.4	1.95	18.2	2.12	21.7	1.30	8.7	4.04	45.8	4.78	55.6	5.10	58.7
Ours$_{DS_LB}$	0.62	0.3	1.02	5.6	0.79	1.8	0.64	0.5	0.79	1.3	3.05	32.5	3.68	41.8	3.90	43.4
Ours$_{DS_Log}$	0.70	1.4	1.07	6.5	0.82	2.4	0.69	1.6	0.79	1.9	3.08	33.0	3.70	42.1	3.93	43.8

[*]The authors of EV-FlowNet only evaluated this sequence on a carefully selected 18-second-long extract.
[†]The authors of EV-FlowNet provide an updated version of their model (https://github.com/daniilidis-group/EV-FlowNet/), which we recomputed their results with.
[‡]This result was computed by ourselves, using the code provided by the authors (https://github.com/chan8972/Spike-FlowNet).
[§]The authors used an accumulation window of $\Delta t = 2.5$ms, whereas all the other results presented here used an accumulation window of $\Delta t = 1$ frame (10 to 40 times longer), thus creating an unfair comparison. Therefore, we report their measurements here, but do not consider them for comparison purposes.
[¶]The results of ECN$_{masked}$ were obtained after manually removing from the dataset erroneous ground truth values created by independently moving objects. While allowing for better AEE and outliers results, it creates an unfair comparison baseline. Thus, we report their measurements here, but do not consider them for comparison purposes.

TABLE III
FWL RESULTS ON THE MVSEC DATASET

Sequence	Indoor flying 1 (cut)[*]	Indoor flying 2 (cut)[*]	Indoor flying 3 (cut)[*]	Outdoor day 1 (cut)[*]	Outdoor day 2 (cut)[*]	Outdoor night 1	Outdoor night 2	Outdoor night 3
EV-FlowNet$_{MVSEC}$ [13][†]	1.02	1.13	1.06	1.15	1.21	-	-	-
EV-FlowNet$_{HQF}$ [39]	1.14	1.36	1.23	**1.27**	**1.20**	-	-	-
Ours	**1.21**	**1.41**	**1.37**	1.17	1.17	**1.35**	**1.45**	**1.42**
Ours$_{NDF}$	1.16	1.34	1.29	1.15	1.14	1.28	1.37	1.34
Ours$_{DS_L}$	1.16	1.29	1.27	1.13	1.12	1.21	1.28	1.25
Ours$_{DS_LB}$	1.20	**1.41**	1.35	1.16	1.17	1.34	1.43	1.39
Ours$_{DS_Log}$	**1.21**	1.39	1.36	1.16	1.16	1.32	1.41	1.38

[*]Stoffregen *et al.* [39] selected cuts from these sequences to evaluate their method; we use the same extracts here.
[†]As computed by Stoffregen *et al.* [39].

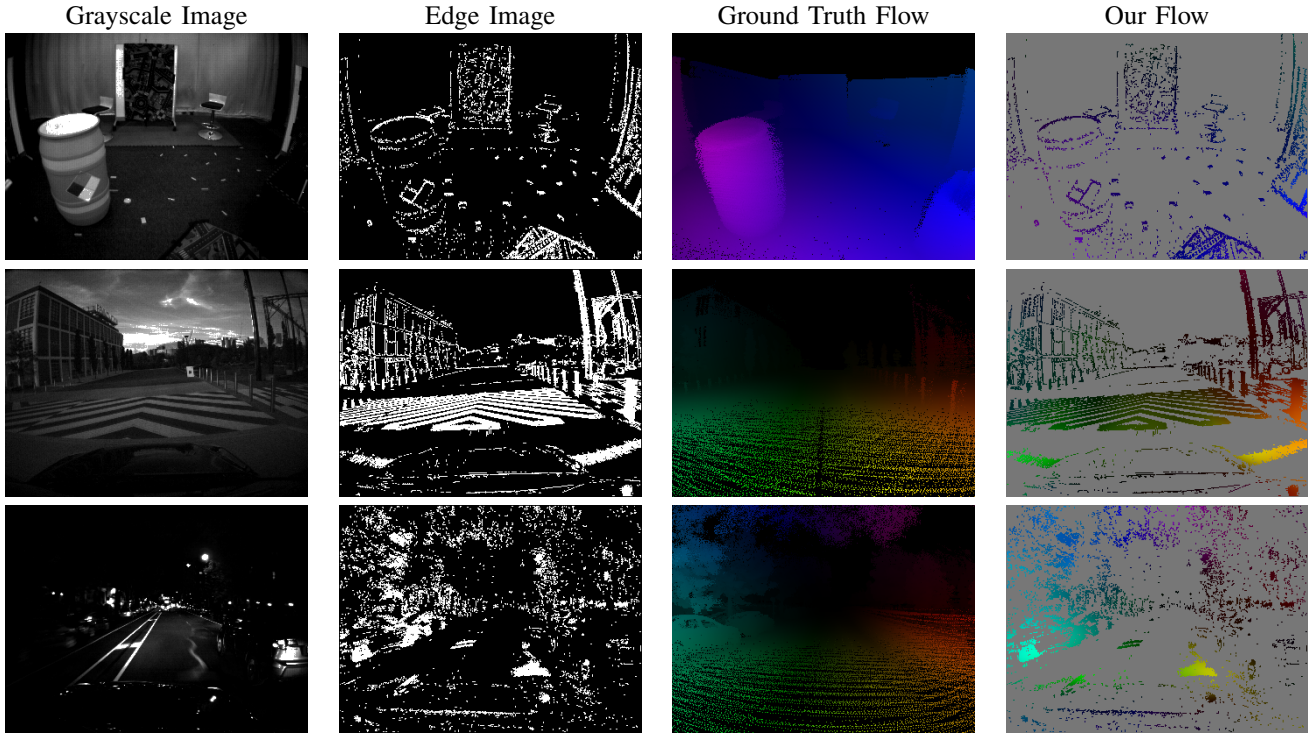

Fig. 5. Qualitative results on MVSEC dataset. Sequences, from top to bottom: Indoor flying 1, Outdoor day 1, and Outdoor Night 1. Best viewed in color.

TABLE IV
FWL RESULTS ON PROPHESEE'S 1 MEGAPIXEL AUTOMOTIVE DETECTION DATASET

Sequence	Jan. 30	Feb. 15	Feb. 18	Feb. 19	Feb. 21	Feb. 22	Apr. 12	Apr. 18	Jun. 11	Jun. 14	Jun. 17	Jun. 19	Jun. 21	Jun. 26	*Global*
Ours	**1.63**	1.47	**1.34**	**1.64**	**1.36**	**1.51**	1.14	1.54	**1.80**	**1.35**	**1.46**	**1.38**	**1.54**	1.56	**1.46**
Ours_{NDF}	1.43	1.37	1.25	1.52	1.28	1.41	1.08	1.43	1.63	1.27	1.35	1.31	1.46	1.46	1.37
$\text{Ours}_{\text{DS\_L}}$	1.44	1.40	1.32	1.41	1.29	1.38	**1.21**	1.27	1.50	1.28	1.27	1.30	1.32	1.42	1.34
$\text{Ours}_{\text{DS\_LB}}$	**1.63**	**1.48**	1.33	1.63	1.35	1.49	1.14	**1.55**	1.78	1.34	1.43	1.37	1.52	**1.57**	1.45
$\text{Ours}_{\text{DS\_Log}}$	1.60	1.47	1.33	1.60	1.34	1.47	1.14	1.54	1.75	1.33	1.41	**1.38**	1.51	1.56	1.43

TABLE V
FWL RESULTS ON THE 20-MINUTE-LONG DRIVING SEQUENCE

Village (0'00 - 4'00)	Side Road (4'00 - 7'00)	Highway (7'00 - 11'00)	Suburban (11'00 - 14'30)	Urban (14'30 - 20'45)	*Full sequence* (0'00 - 20'45)
1.70	1.45	1.67	1.62	1.43	1.56

TABLE VI
FWL RESULTS ON OUR HIGH-SPEED HIGH-DEFINITION EVENT-BASED INDOOR DATASET

Sequence	Checkerboard	Desk	Office	Fan
$\Delta T = 15$ms	1.49	1.71	1.74	1.05
$\Delta T = 5$ms	1.75	1.66	1.84	1.53

high-definition baseline for conducting our evaluation. Given its density (1.2 TB, 14 hours of data), we settled on the use of only its "test" sequences for the evaluation, which account for a total of 2 hours of raw data.

We tried to adapt the codes of the low-resolution EV-FlowNet methods proposed by Zhu *et al.* [13] and Stoffregen *et al.* [39], to provide the same comparisons as on the MVSEC dataset. However, the results it yielded are not representative of the qualities of the methods, as their neural-network-based approaches were not designed nor trained for high-resolution input data. The most apparent and limiting issues are the computation times (the code was not optimized for high-resolution data), and the absence of denoising (it created large artifacts in the optical flow results).

We therefore present our FWL results on this dataset in Table IV, split between each recording day. Similarly to what was observed for low-resolution data, our method always yields FWL values greatly superior to 1, indicating an accurate optical flow. Compared to the ablation alternatives, it can also be observed that our final version displays the best global result, and the best results in all sequences but four ("Feb. 15", "Apr. 12", "Apr. 18", and "Jun. 26", where it is the second best alternative), showing once again the importance of the denoising and of our inverse exponential distance surface.

G. Complementary Evaluation on a 20-minute-long High-Resolution Driving Sequence

While Prophesee's 1 Megapixel Automotive Detection dataset allows for a reproducible evaluation of EBOF methods with the FWL, it contains only event recordings, preventing the comparison with frame-based state-of-the-art methods. In order to complete our evaluation, we make use in this sub-section of a twenty-minute-long driving sequence, containing both the output from high-definition frame-based and event-based cameras. This sequence presents a wide diversity of driving situations (urban/rural roads, highway, roundabouts, many other vehicles, pedestrians, ...). It has been recorded by and graciously shared with us by Prophesee.

The FWL results of this evaluation are presented in Table V, split between each period of the sequence. An overall FWL result on the complete sequence is also presented. It can be observed that our FWL results are greatly satisfying, by remaining quite over the value of 1.

The main advantage of this sequence, however, lies in the possibility to compare our EBOF results to frame-based ones. For this prospect, we used RAFT [26] as frame-based reference, as it currently stands as one of the state-of-the-art optical flow methods. Due to the lack of calibration between the two sensors, only a qualitative evaluation can be presented. Therefore, several visual optical flow results are given in Fig. 6. Rendering on the full sequence can be viewed in video format, at the link given at the beginning of this article. It can be seen that, similarly to when a low-resolution input is used, our optical flow remains visually very close to the reference.

H. Evaluation on our High-Speed High-Definition Event-Based Indoor Dataset

For event-based driving sequences, most of our optical flow results are restricted to a few pixels, due to the low accumulation time of $\Delta T = 15$ms we used throughout this article, in accordance to movements speed. In order to show how our method is able to handle movements of higher magnitudes in various situations, we recorded a high-speed high-definition event-based dataset, using a Prophesee Gen4 camera (1280×720) [15]. This dataset is composed of four indoor sequences taken in office environment (namely, "Checkerboard", "Desk", "Office", and "Fan"). The first three of them were recorded by manually shaking the camera, while for the last one, the camera was fixed in front of a high-speed fan. An illustration of these sequences is given in Fig. 7.

This evaluation relies also on the FWL metric, to compare ourselves to the "zero flow" reference. The results are presented in Table VI. It can be seen here that we always obtain a FWL greater than 1, underlining once again the accuracy of our optical flow results, even under larger apparent motions. However, apart for the "Desk" recording, all recordings display better FWL results when a lower accumulation time of $\Delta T = 5$ms is employed. This is due to the fact that, at such high motion speeds, the edge images become slightly too blurry to provide the best optical flow results when the accumulation time of $\Delta T = 15$ms is employed. Lowering

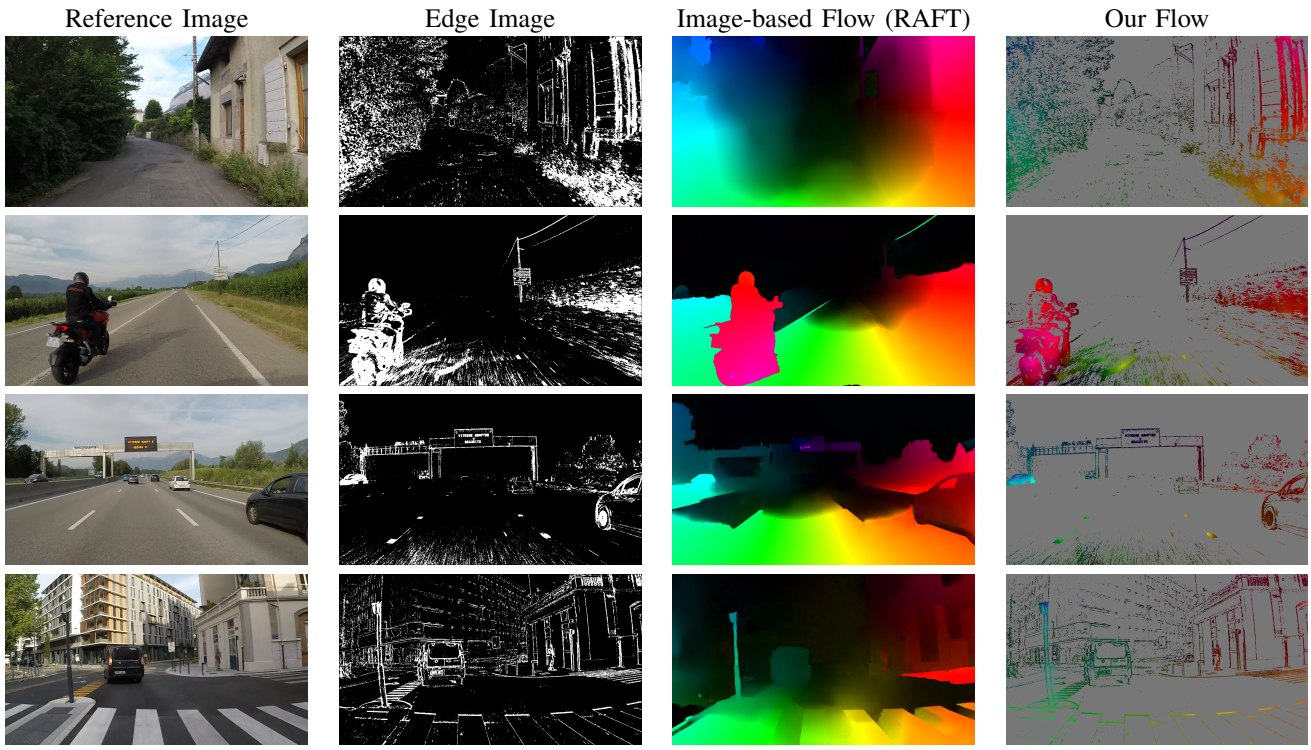

Fig. 6. Qualitative results on the 20-minute-long driving sequence. Extracts used, from top to bottom: a village street, a motorcycle overtaking, a highway, and an intersection. Best viewed in color.

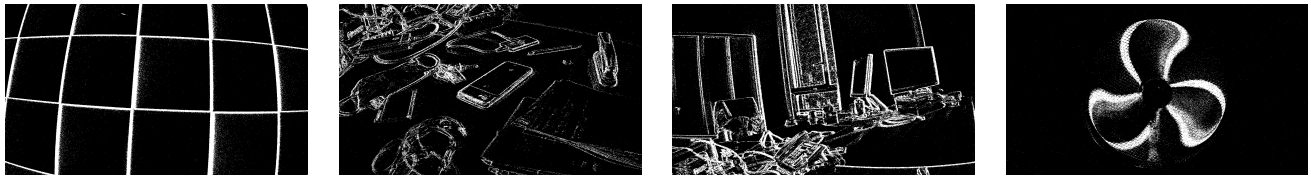

Fig. 7. Edge images of our high-speed high-definition event-based indoor dataset sequences. From left to right: "Checkerboard", "Desk", "Office", "Fan".

Fig. 8. Sample EBOF results for the "Fan" sequence of our high-speed dataset, with $\Delta T = 15$ms (left) and $\Delta T = 5$ms (right). Notice how the blades of the fan are merged together in the first case, while they appear clearly in the second one, leading to improved optical flow results.

accumulation time helps obtaining sharpest edge images, and in return, more accurate optical flow and FWL results. This remark is especially true for the "Fan" sequence, where the very high speed of the blades leads them to appear too blurry for a correct optical flow to be computed when $\Delta T = 15$ms is used; an illustration is given in Fig. 8.

I. Sensitivity Analysis

In this subsection, we analyze the sensitivity of N_d, N_f and d_{sat} parameters on our EBOF results. N_d and N_f define denoising and filling thresholds (see Section III-B), d_{sat} defines the saturation value for the computation of the inverse exponential distance surface (Section III-C). For that purpose, we use the "Outdoor day 1" sequence from the MVSEC dataset with the AEE metric. We display, in Fig. 9, these results for $N_d \in \{0 = \text{"disabled"}, 1, 2, 3, 4\}$, $N_f \in \{1, 2, 3, 4, 5 = \text{"disabled"}\}$, and $d_{\text{sat}} \in \{3, 6, 9, 12\}$ pixels.

From this plot, it can first be noted that the denoising should not be too strong, that is, $N_d \leq 2$. In the same time, the filling threshold should be $N_f \geq 3$. Overall best d_{sat} value is incontestably $d_{\text{sat}} = 6$px, and it is the most sensitive parameter. The best set of parameters is $\{N_d = 1, N_f = 4, d_{\text{sat}} = 6\}$, with the corresponding minimal AEE = 0.53px. The worst set of parameters is $\{N_d = 4, N_f = 1, d_{\text{sat}} = 12\}$, with the corresponding AEE = 0.78px, increasing by 47% compared to the best parameters. This shows the importance of the parameters choice to guarantee adequate optical flow results.

J. Real-Time Compliance

To finally show the real-time compliance of our approach for both low- and high-resolution input data, we used respectively the "Indoor flying 1" sequence from the MVSEC dataset and the `moorea_2019-01-30_000_td_671500000_`

TABLE VII
AVERAGE EXECUTION TIMES AND STANDARD DEVIATION OF EACH STEP OF OUR METHOD FOR LOW- AND HIGH-RESOLUTION INPUTS, IN MILLISECONDS

Version	Edge image (after accumulation)*	Denoising & filling	Inverse exponential distance transform	Optical flow†	Total
Low-resolution (346 × 260)					
CPU-only	0.15±0.03	0.77±0.16	5.07±2.07	*3.59±0.09*	9.57±2.13
CPU & GPU	0.18±0.04	0.50±0.04	1.37±0.09	3.76±0.16	5.82±0.23
High-resolution (1280 × 720)					
CPU-only	0.54±0.06	3.55±0.30	7.43±0.51	*12.21±0.16*	23.73±0.69
CPU & GPU	0.55±0.07	0.69±0.14	3.75±0.46	11.89±0.76	16.88±1.30

*This module uses only the CPU, hence the similar results between the "CPU-only" and "CPU & GPU" lines for this column.
†The optical flow library provided by Adarve *et al.* [21] does not contain a CPU-only version. The CPU-only experiments therefore use the GPU for optical flow computation, hence the similar results between the "CPU-only" and "CPU & GPU" lines for this column.

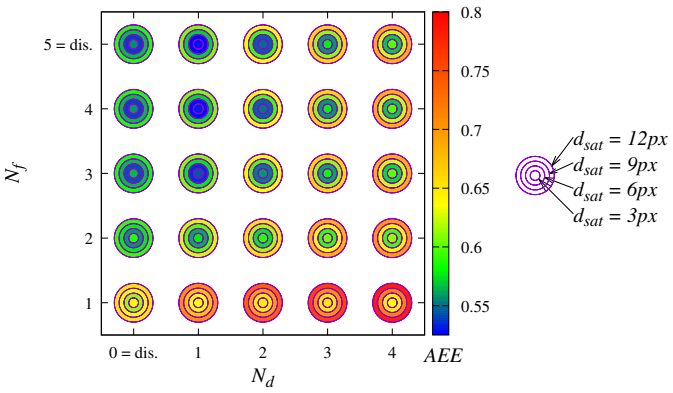

Fig. 9. Sensitivity analysis of our EBOF method, using the AEE on the "Outdoor day 1" sequence from the MVSEC dataset, along the N_d, N_f, and d_{sat} parameters. Lowest AEE (blue) is the best.

`731500000_td` test sequence from Prophesee's 1 Megapixel Automotive Detection dataset. The execution time for each block was not evaluated separately, but simultaneously, as the full pipeline should be able to run on a single computer. In order to also underline the benefits brought by the use of the GPU, we ran the evaluation on both the CPU-only and the CPU+GPU versions of our code.

These results are presented in Table VII. From them, it can be seen on one hand that the GPU-aided version can achieve real-time performances for both low- and high-resolution input data, if their accumulation times are set to at least 4ms and 13ms respectively (the limiting factor being the optical flow module). The values for ΔT used throughout this article, therefore, respect the real-time constraint. The CPU-only version, on the other hand, can achieve real-time performances for both low- and high-resolution data when the accumulation time is of at least 8ms and 13ms respectively (the limiting factor being here the distance transform and optical flow module respectively). As noted in Table VII, however, the optical flow can only be computed on GPU at this time, and transferring this computation to the CPU would likely make these execution times greatly increase.

From the results of Table VII and from the previous conclusions, using our architecture, the best performances that can be reached for a low-resolution input is therefore a 250Hz flow with a latency of 10ms (for the minimal $\Delta T = 4$ms of accumulation time), while, for a high-resolution input, a 77Hz flow with a latency of 30ms can be achieved (for the minimal $\Delta T = 13$ms of accumulation time).

As a fast algorithm, FireFlowNet [14] displays a theoretical inference frequency up to 262Hz for a low-resolution (346 × 260) input, similar to ours. For a higher definition (1280 × 720), however, our approach achieves frame rates 2- to 3-times better than them, as their method can only reach 29Hz. The GPU they use ranks similarly to ours in popular benchmarks. In addition, we ran EV-FlowNet [13] on low- (346 × 260) and on high-resolution (1280 × 720) data. While EV-FlowNet inference achieved a 125Hz flow on low-definition data, it showed its limits on high-resolution input with a 12.5Hz flow output. These frequencies consider only inference process, full latency is unknown and should include events accumulation time and specific dense representation creation.

To compare with a state-of-the-art frame-based optical flow algorithm, we measured a 12.5Hz flow on low-resolution input, and a 2Hz one on high-definition images with RAFT [26].

V. CONCLUSION

In this article, a complete pipeline for real-time computation of event-based optical flow (EBOF) from both low- and high-resolution event cameras has been proposed. It includes optimized algorithmic choices as well as a novel inverse exponential distance surface representation. Several evaluations have been conducted to show the relevance of our contributions. Resulting accuracies surpass or are close to the non-real-time state of the art for low-definition recordings, as well as on novel high-definition sequences. Frame rates of respectively 250Hz and 77Hz for resolutions of 346×260 and 1280×720 were also achieved, making it, to the best of our knowledge, the most accurate EBOF method for low- and especially high-resolution event cameras that could be deployed in the wild.

Deep learning dominates EBOF estimation when looking for accuracy at the cost of real-time ability. The EV-FlowNet architecture is a perfect example of this trend, as shown in Table II. On the contrary, when real-time running is needed, this is currently our proposition, which is not a neural network, that provided the best results. As our method is not based on machine learning, it is independent from a training process involving specific datasets and loss functions. In other words, the results are expected to be similar to the ones presented here for all types of scenes. This can be seen as an advantage,

compared for example to EV-FlowNet which shows very different results according to the way it has been trained.

In hindsight, improvements could be brought to the current method, especially regarding the event accumulation process. Relying on a predetermined fixed accumulation time ΔT, which highly depends on the appearance and dynamics of the visual scene, can indeed lead to instabilities in the appearance of the edge images if not chosen carefully. Introducing an adaptive method to dynamically determine the correct accumulation time to use, as proposed in [51] for instance, could allow for obtaining clear edge images independently from the scene evolution, but would also certainly make the real-time constraint harder to achieve. Our architecture could also benefit from a more optimized implementation on specialized hardware (FPGA for instance, eliminating the need for an energy-intensive GPU), which could also allow for even higher frame rates by lowering the minimum accumulation times. Applying our EBOF to complex automotive-related applications, such as proposed in [52], could also be addressed by future work, for instance for improving the detection of obstacles appearing in the close neighbourhood of the vehicle.

ACKNOWLEDGMENT

The authors thank Renault for lending their Prophesee Gen 4 camera, Prophesee for giving us access to their twenty-minute-long driving sequence, and in particular Davide Migliore for providing thoughtful insight on their high-definition event sensor.

REFERENCES

[1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[2] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[3] C. Braillon, C. Pradalier, J. L. Crowley, and C. Laugier, "Real-time moving obstacle detection using optical flow models," *IEEE Intelligent Vehicles Symposium*, pp. 466–471, 2006.

[4] J. Huang, W. Zou, J. Zhu, and Z. Zhu, "Optical flow based real-time moving object detection in unconstrained scenes," *ArXiv*, 2018.

[5] M. K. Hossen and S. H. Tuli, "A surveillance system based on motion detection and motion estimation using optical flow," *5th ICIEV*, pp. 646–651, 2016.

[6] C. Chuanqi, H. Xiang-yang, Z. Zhen-jie, and Z. Mandan, "Monocular visual odometry based on optical flow and feature matching," *29th Chinese Control And Decision Conference (CCDC)*, pp. 4554–4558, 2017.

[7] C. Tang, X. Zhao, J. Chen, L. Chen, and Y. Zhou, "Fast stereo visual odometry based on LK optical flow and ORB-SLAM2," *Multimedia Systems*, pp. 1–10, 2020.

[8] S. Galic and S. Lončarić, "Spatio-temporal image segmentation using optical flow and clustering algorithm," *Proceedings of the First IWISPA*, pp. 63–68, 2000.

[9] R. Benosman, C. Clercq, X. Lagorce, S. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 407–417, 2014.

[10] H. Akolkar, S. Ieng, and R. Benosman, "See before you see: Real-time high speed motion prediction using fast aperture-robust event-driven visual flow," *IEEE TPAMI*, 2020.

[11] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based feature tracking with probabilistic data association," *ICRA*, pp. 4465–4470, 2017.

[12] M. Almatrafi, R. W. Baldwin, K. Aizawa, and K. Hirakawa, "Distance surface for event-based optical flow," *IEEE TPAMI*, vol. 42, pp. 1547–1556, 2020.

[13] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," in *Proceedings of Robotics: Science and Systems*, 2018.

[14] F. Paredes-Vallés and G. D. Croon, "Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3446–3455, 2021.

[15] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, É. Reynaud, P. Mostafalu, F. T. Brady, L. Chotard, F. LeGoff, H. Takahashi, H. Wakabayashi, Y. Oike, and C. Posch, "A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066GEPS readout, programmable event-rate controller and compressive data-formatting pipeline," *ISSCC*, pp. 112–114, 2020.

[16] P. Jund, C. Sweeney, N. Abdo, Z. Chen, and J. Shlens, "Scalable scene flow from point clouds in the real world," *ArXiv*, 2021.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," *CVPR*, pp. 3354–3361, 2012.

[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th IJCAI*, 1981.

[19] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker," Intel Corporation, Microprocessor Research Labs, Tech. Rep., 1999.

[20] E. Meinhardt, J. Pérez, and D. Kondermann, "Horn-Schunck optical flow with a multi-scale strategy," *Image Processing On Line*, vol. 3, pp. 151–172, 2013.

[21] J. Adarve and R. Mahony, "A filter formulation for computing real time optical flow," *IEEE Robotics and Automation Letters (RA-L)*, vol. 1, pp. 1192–1199, 2016.

[22] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, pp. 75–104, 1996.

[23] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106, pp. 115–137, 2013.

[24] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," *ICCV*, pp. 2758–2766, 2015.

[25] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[26] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *ECCV*, 2020, pp. 402–419.

[27] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," *CVPR*, pp. 3867–3876, 2018.

[28] T. Stoffregen and L. Kleeman, "Simultaneous optical flow and segmentation (SOFAS) using dynamic vision sensor," *ArXiv*, 2018.

[29] D. Liu, Á. P. Bustos, and T.-J. Chin, "Globally optimal contrast maximisation for event-based motion estimation," *CVPR*, pp. 6348–6357, 2020.

[30] F. Paredes-Vallés, K. Y. W. Scheper, and G. de Croon, "Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception," *IEEE TPAMI*, vol. 42, pp. 2051–2064, 2020.

[31] F. Paredes-Vallés, J. J. Hagenaars, and G. D. Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," *ArXiv*, 2021.

[32] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[33] J. Nagata, Y. Sekikawa, and Y. Aoki, "Optical flow estimation by matching time surface with event-based cameras," *Sensors*, vol. 21, 2021.

[34] C. Lee, A. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-FlowNet: Event-based optical flow estimation with energy-efficient hybrid neural networks," in *ECCV*, 2020.

[35] M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza, "Dense optical flow from event cameras," *ArXiv*, 2021.

[36] L. Pan, M. Liu, and R. Hartley, "Single image optical flow estimation with an event camera," *CVPR*, pp. 1669–1678, 2020.

[37] B. Rueckauer and T. Delbruck, "Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor," *Frontiers in Neuroscience*, vol. 10, 2016.

[38] C. Scheerlinck, N. Barnes, and R. Mahony, "Continuous-time intensity estimation using event cameras," *ArXiv*, 2018.

[39] T. Stoffregen, C. Scheerlinck, D. Scaramuzza, T. Drummond, N. Barnes, L. Kleeman, and R. Mahony, "Reducing the sim-to-real gap for event cameras," in *ECCV*, 2020.
[40] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE TPAMI*, vol. 43, pp. 1964–1980, 2021.
[41] C. Ramamoorthy and H. F. Li, "Pipeline architecture," *ACM Computing Surveys*, vol. 9, pp. 61–102, 1977.
[42] T. Delbrück, "Frame-free dynamic digital vision," *Proceedings of the International Symposium on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, pp. 21–26, 2008.
[43] Y. Feng, H. Lv, H. Liu, Y. Zhang, Y. Xiao, and C. Han, "Event density based denoising method for dynamic vision sensor," *Applied Sciences*, vol. 10, p. 2024, 2020.
[44] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3D perception," *IEEE RA-L*, vol. 3, pp. 2032–2039, 2018.
[45] D. Coeurjolly, A. Montanvert, and J. Chassery, "Distances discrètes," in *Géométrie discrète et images numériques*. Hermès Paris, 2007, ch. 5, pp. 123–145.
[46] M. J. Black, "Robust incremental optical flow," Ph.D. dissertation, Yale University, 1992.
[47] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," *ICCV*, pp. 5632–5642, 2019.
[48] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," *CVPR*, pp. 989–997, 2019.
[49] C. Ye, A. Mitrokhin, C. Fermüller, J. Yorke, and Y. Aloimonos, "Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors," *IROS*, pp. 5831–5838, 2020.
[50] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.
[51] M. Liu and T. Delbrück, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," in *BMVC*, 2018.
[52] J. H. Jung and C. G. Park, "Constrained filtering-based fusion of images, events, and inertial measurements for pose estimation," *ICRA*, pp. 644–650, 2020.

Vincent Brebion was born in Blendecques, France, in 1997. He received the engineering degree in computer science and the master's degree in complex systems engineering from the Université de technologie de Compiègne (UTC), France, in 2020. He is currently pursuing the Ph.D. degree in computer vision at the Heudiasyc Lab., UTC, France, and is currently a member of the UTC/CNRS/Renault SIVALab joint laboratory. His research interests include event-based cameras, multi-sensor fusion, and computer vision for autonomous vehicles.

Julien Moreau received the Ph.D. degree in computer vision from the Université de Technologie de Belfort-Montbéliard (UTBM), France, in 2016. He accomplished postdoctoral positions in The French Institute of Science and Technology for Transport Development and Networks (IFSTTAR), in Lille, France, and in the Institute of Information and Communication Technologies Electronics and Applied Mathematics (ICTEAM), at Université Catholique de Louvain, Louvain-la-Neuve, Belgium. Since 2019, he is an associate professor in the Computer Science department of Université de technologie de Compiègne (UTC), France, and is carrying out his research in Heudiasyc UMR 7253, a joint UTC-CNRS research laboratory. From that, he is also a member of SIVALab, a joint laboratory between Renault, UTC and CNRS. His research interests cover stereovision, unconventional cameras, calibration and machine learning applied to perception and localization for mobile robotics.

Franck Davoine received his Ph.D. in 1995 from Grenoble INP - Université Grenoble Alpes in France. He was appointed at Université de technologie de Compiègne (UTC), Heudiasyc Lab., France, in 1997 as an Associate professor and in 2002 as a Researcher at CNRS. From 2007 to 2014, he was on leave at LIAMA Sino-European Lab. in Beijing, P.R. China, as PI of a project with CNRS and Peking University on Multi-sensor based perception and reasoning for intelligent vehicles. In 2015, he was back in Compiegne, PI of a challenge-team within the Laboratory of Excellence of UTC focusing on Collaborative vehicle perception and urban scene understanding for autonomous driving, and member of the CNRS/UTC/Renault SIVALab joint laboratory specializing in localization and perception systems for autonomous vehicles.