Data-set for Event-based Optical Flow Evaluation in Robotics Applications

Mahmoud Z. Khairallah[a], Fabien Bonardi[b], David Roussel[c]and Samia Bouchafa[d]

IBISC, Univ. Evry, Université Paris-Saclay, 91025, Evry, France

Keywords: Event-based Camera, Optical Flow Estimation, Ego-motion Data-sets, Frame Alignment.

Abstract: Event-Based cameras (also known as Dynamic Vision Sensors ”DVS”) have been used extensively in robotics during the last ten years and have proved the ability to solve many problems encountered in this domain. Their technology is very different from conventional cameras which requires rethinking the existing paradigms and reviewing all the classical image processing and computer vision algorithms. We show in this paper how Event-Based cameras are naturally adapted to estimate on the fly scene gradients and hence the visual flow. Our work starts with a complete study of existing event-based optical flow algorithms that are suitable to be integrated into real-time robotics applications. Then, we provide a data-set that includes different scenarios along with a set of visual flow ground-truth. Finally, we propose an evaluation of existing event-based visual flow algorithms using the proposed ground truth data-set.

1 INTRODUCTION

Optical flow is an essential visual cue that is exploited in most of computer vision algorithms for robotic applications. In order to measure the reliability of new proposed algorithms, several synthetic and real data-sets were provided as benchmarks: Middlebury (Baker et al., 2011), KITTI (Menze et al., 2015). The rise of Dynamic Vision Sensors ”DVS” required a total paradigm shift in all computer vision algorithms, optical flow algorithms included, thus implying a need to propose new ground-truth data-sets. (Rueckauer and Delbruck, 2016a) propose synthetic and real DVS data-sets, restricted to camera rotational motions, and use the gyroscope embedded to the DVS sensor to obtain the ground-truth 2D motion also called ”in-plane motion”. (Barranco et al., 2016) use an RGB-D sensor on a pan-tilt rig connected to the camera to create optical flow ground truth knowing 3D motion and depth.

In this paper, our first contribution is to provide an adaptive method to increase the quality of obtained data from Event-based cameras and reject noises. We use a VICON motion capture system. Such a system can provide a 6DOF pose ground truth with high accuracy at frequencies above 100 Hz while being adapted to moving scenes with multiple moving objects. An optical flow ground truth can easily be derived by measuring the relative objects-camera pose to calculate the 2D objects projection on the camera frame. We introduce a calibration method to align the external VICON system with the DVS internal IMU that can be applied easily for any sensors that share roughly the same initial position. We also propose an event-based data-set that can be used as ground truth

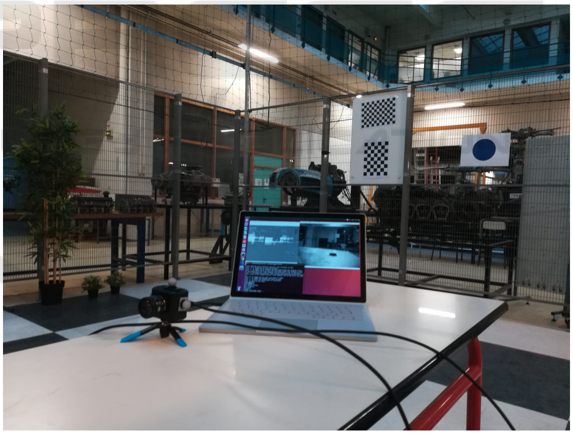

Figure 1: The environment setup of the system: event-based camera, checkerboards, VICON system.

[a] https://orcid.org/0000-0002-0724-8450
[b] https://orcid.org/0000-0002-3555-7306
[c] https://orcid.org/0000-0002-1839-0831
[d] https://orcid.org/0000-0002-2860-8128

Khairallah, M., Bonardi, F., Roussel, D. and Bouchafa, S.
Data-set for Event-based Optical Flow Evaluation in Robotics Applications.
DOI: 10.5220/0010320304800489
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, pages 480-489
ISBN: 978-989-758-488-6

for optical flow algorithms comparison. Finally, we evaluate existing event-based optical flow algorithms that are adapted to robotics applications and previously introduced in (Delbrück, 2008), (Benosman et al., 2012), (Benosman et al., 2013), (Rueckauer and Delbruck, 2016b) and (Mueggler et al., 2015). This paper is organised as follows: in section 2, a brief explanation of the selected optical flow algorithms is presented. In section 3, an illustration of the recorded data-set scenario is explained. In section 4 we present the intrinsic and extrinsic calibration of the systems and in section 5 a comparison between tested algorithms estimated optical flows and the optical flow obtained from VICON is carried out. Results and conclusion are given in the last sections.

2 EVENT-BASED OPTICAL FLOW ALGORITHMS

Several optical flow algorithms have been developed to adapt to the nature of event-based vision sensors in robotics field. The algorithms presented in this study can be grouped in three main categories: variants of Lucas-Kanade (Lucas and Kanade, 1981) optical flow, Local Plane optical flow and Regularised optical flow. We chose one algorithm from each category according to efficiency and real-time robotics applications adaptability criteria. Some modifications are proposed to make these algorithms more adaptive to different dynamic conditions.

2.1 Event-based Representation

The design of DVS cameras is particularly adapted to the visual optical flow nature which is defined as the perceived 2D motion of a 3D moving pattern. The "silicon retina" used in DVS cameras mimics the human eye derivative functionality (the system responsible for motion detection) by sending a signal (event) whenever a change occurs at a specific pixel. The created events can be characterized as a tuple $e = \langle x, y, t, p \rangle$ where x and y are the position of the event in pixel coordinates, t is the timestamp of event's creation and p is the polarity of the event such that $p \in \{-1, +1\}$. The polarity of the event interprets the increasing or decreasing change of illumination that occurs in the environment. An event is created whenever a difference in illumination exceeds a threshold according to the following equation:

$$\Delta L(x_i, y_i, t_i) = L(x_i, y_i, t_i) - L(x_i, y_i, t_i - \Delta t) = p_i \delta_l \tag{1}$$

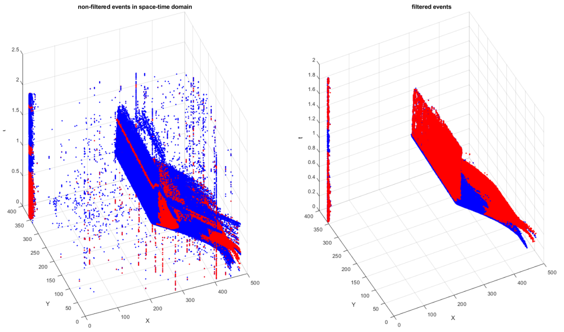

Figure 2: Left: events created due to motion of a beam during 2 seconds without filtering. Right: events after filtering.

where $L(x_i, y_i, t_i)$ is the illumination log intensity of the current time and $L(x_i, y_i, t_i - \Delta t)$ is the illumination log intensity of the event created at this pixel previously at $t_i - \Delta t$, δ_l is the threshold that determines the creation of the event which is namely about $10 : 15\%$. The signed threshold δ_l can be exceeded due to change of luminosity in the environment with no motion or due to motion under constant luminosity or a combination of both cases. In robotics applications, we assume that environment luminosity changes are negligible. Hence, under a brightness constancy constraint, the creation of an event can be approximated by the following equation using a Taylor expansion of the intensity function:

$$\Delta L(x_i, y_i, t_i) \approx \frac{\partial L}{\partial t}(x_i, y_i, t_i)\Delta t = \langle \nabla_u L(x_i, y_i, t_i), \mathbf{\dot{u}}\Delta t \rangle \tag{2}$$

where $\langle ., . \rangle$ refers to a dot product and $\nabla_u L = (\frac{\partial L}{\partial x}, \frac{\partial L}{\partial y})$. The later equation shows that the creation of an event embeds in itself the visual flow of the moving environment. As a consequence, Event-Based cameras provide a quasi-continuous flow of events which facilitates events correlation unlike standard cameras which face brightness discretization issues. According to this, we give, in this study, a selection of Event-Based optical flow algorithms that would be simple to implement, running fast enough for real time applications and then we study their performance according to different criteria. The selected approaches are presented in the following sub-sections.

2.2 Direction Selective Filter

(Delbrück, 2008) propose to augment the information that each event carries with a direction that is determined using a rough optical flow estimation. Velocity magnitude and direction are assigned to each event through three steps: event filtering, direction selection and magnitude estimation.

2.2.1 Event Filtering

Due to the noisy nature and the sensitivity of DVS cameras, it is essential to get rid of uncorrelated events created by background activity or any other source like transistor switch leakage (Lichtsteiner et al., 2008). The author employs what is called an activity filter to reject unwanted events that takes only one parameter T, the "support time", which is the maximum time difference permitted between the current event and events created previously in the same neighbourhood. We use for that the active events surface, which is the buffer that saves only the last existing event at a specific pixel and its timestamp. The support time T decides whether an event can be passed as a true event or as noise. The process is carried out in two steps. First, the event timestamp is stored for the pixel 8-neighbourhood. Second, a check between the timestamp stored in the event's location and the event's timestamp is performed: if an event occurred nearby the current event (within the support time T), the new event is passed. It is discarded otherwise.

Using a constant support time T is not the best solution as events could be rejected or included arbitrarily independently from the environment motion. We propose a modified version of this filter to make it more robust to noise and more adaptive to dynamic environments. We thereby introduce an adaptive parameter T_f that depends on the created events frequencies since they are related to the dynamics of the environment. We use the concept of linear interpolation to estimate T_f so that it can be bounded by T_{min} and T_{max}. Using the frequency directly would lead to be stuck a narrow zone of T_f. For this reason, we use the log inverse function to exploit its saturation property and stretch this zone according to the following equations:

$$\alpha = \frac{k}{\log f_e} \tag{3}$$

$$T_f = \frac{T_{max} - T_{min}}{\alpha_{max} - \alpha_{min}}(\alpha - \alpha_{min}) + T_{min} \tag{4}$$

where k is tuned to get a better logarithmic curve and then the best value of T_f for different frequencies, T_{min} and T_{max} are the minimum and maximum time period that the filter can provide. α_{min} and α_{max} are the values of α which correspond to the lowest and highest values of events frequency f_e. This filter was integrated to all the studied algorithms in this paper to avoid computing incorrect optical flow from noise events and losing time in unnecessary calculations (See Fig. 2).

2.2.2 Direction Selection

A moving edge will tend to create events that are very close to each other in space and correlated in time. The orientation filter treats the ON and OFF events separately. The present event is given its corresponding angle by checking the events that have the maximum correlation with the current event in the neighborhood. Events correlation is defined as the difference in time between the current pixel and the pixels in the neighborhood, where events with maximum correlation (and consequently minimum time difference) are necessarily created by the same edge. According to the location of these events the present event is given a value among 4 possible orientations separated by 45 degrees, one of four orientations is assigned to an event but no positive or negative direction is assigned.

2.2.3 Magnitude Estimation

For the estimation of each event's magnitude, the concept of time of flight, introduced in (Rueckauer and Delbruck, 2016a), is applied by computing the temporal interval of the current event with the recent past events along the direction of the edge. Since the direction of the event is known at this step, it is used to define the two directions perpendicular to the edge. Time intervals of the pixels along that axis are compared. Average of differences of each timestamp is considered as the inverse of the speed on the events (in pixel per second). A positive or negative direction is then assigned according to the time correlation and considering that an edge will only create events before the motion direction and not past it.

2.3 Adapted Lucas-Kanade Optical Flow

The adaptation of Lucas-Kanade optical flow for event-based vision sensors was first introduced by (Benosman et al., 2012). The brightness constancy constraint assumes that there is no change in brightness w.r.t time i.e. $d(I(x(t),y(t),t))/dt = 0$. This assumption leads to the optical flow constraint equation $I_xu + I_yv = \nabla\mathbf{I}^T \cdot \mathbf{U} = -I_t$ where I_x, I_y and I_t are the partial derivatives of image intensity toward x, y and t respectively, u and v are the components of the 2D velocity vector in x and y directions respectively. As the problem of optical flow defined in this scheme is under-determined, an additional constraint was added by Bruce D. Lucas and Takeo Kanade to solve it: it consists of applying a neighbourhood consistency condition. This condition uses the assumption that neighbouring pixels will experience the same 2D velocity. It employs a least-square optimisation scheme to estimate the optical flow over a given neighbourhood.

2.3.1 Backward Finite Difference

The main challenge to estimate an event-based optical flow is to estimate image gradients I_x, I_y and I_t. The creation of an event requires a change of illumination (or gradient in case of illumination constancy) that could be revealed by analyzing gradients values. Thus, the gradient intensity is interpreted as the count of events passed by a pixel during a fixed period of time in the events space $E(x,y,t)$. The events space $E(x,y,t)$ is the space of events that are created due to environmental changes (motion or illumination). (Benosman et al., 2012), propose a backward finite difference scheme to obtain the equivalence of intensity gradients using the following equation:

$$E_x \sim H_e(x_i, y_i, \Delta t) - H_e(x_i - 1, y_i, \Delta t) \quad (5)$$

$$E_y \sim H_e(x_i, y_i, \Delta t) - H_e(x_i, y_i - 1, \Delta t) \quad (6)$$

$$E_t \sim \frac{H_e(x_i, y_i, \Delta t) - H_e(x_i, y_i, \Delta t)}{t_i - t_1} \quad (7)$$

Where the subscripts x, y and t means the partial derivatives in x, y and t directions respectively of events mapping which is similar to image intensity. The gradient is expressed to be similar to the notion of sum of events that would fire at a specific pixel compared to the sum of events fired at the neighboring pixel during an interval Δt. H_e is the histogram of events which is the number of events created in a specific pixel during a certain period of time .

2.3.2 Central Finite Difference

A bias in optical flow estimation is evident in the backward scheme. (Rueckauer and Delbruck, 2016b) introduce the central finite difference method which would yield a symmetric gradient and eliminate the backward bias of the basic Lucas-Kanade method. For the 1st order method the events gradient became:

$$E_x \sim \frac{1}{2}(H_e(x_i + 1, y_i, \Delta t) - H_e(x_i - 1, y_i, \Delta t)) \quad (8)$$

$$E_y \sim \frac{1}{2}(H_e(x_i, y_i + 1, \Delta t) - H_e(x_i, y_i - 1, \Delta t)) \quad (9)$$

while no change has been introduced to the time gradient estimation.

2.3.3 Savitzky-Golay Filter

Since the event-based Lucas-Kanade method is mainly based on rate of event histograms (number of events that occur at a certain pixel during a certain interval of time) in a small neighborhood, the event histogram does not gather a lot of events which would lead to noise sensitivity. (Delbrück, 2008) propose the usage of Savitzky-Golay filter to estimate the image gradient increasing as a consequence signal-to-noise ratio. A low-order polynomial is introduced to fit adjacent points using a least-square scheme, the fitted two-dimensional polynomial function is described as:

$$SG(x,y) = \sum_{p=0}^{n} \sum_{q=0}^{n-p} a_{pq} x^p y^q \quad (10)$$

where n represents the degree of the polynomial. The order of the polynomial is chosen to be linear and symmetric in both dimensions (namely first order), $SG(x,y)$ becomes $a_{00} + a_{01}y + a_{10}x$. The coefficients a_{00}, a_{01} and a_{10} are equivalent to the image gradients E_t, E_y and E_x. Coefficients $\mathbf{a}_{pq}$ are obtained using a least-square fit of data points: **a = Cd d** is a vector that contains timestamp of events to be fitted. **C** is a matrix calculated once for a certain size of neighborhood and equal to $\mathbf{C} = (\mathbf{B^T B})^{-\mathbf{1}} \mathbf{B^T}$ where $\mathbf{B} = x^p y^q$ is the matrix containing the polynomial terms. Hence, the least-square equation becomes $\mathbf{a}_{pq} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{d}$. After estimating the coefficients, the gradients can be used to evaluate the optical flow while increasing the SNR and also expecting a faster computation time.

2.4 Local Plane Fit Optical Flow

By exploiting the almost-continuous nature of the created events from event-based cameras (created events would look like an extruded shape extended in time using edges, See Fig. 3), it helps a lot to estimate image gradients accurately. The local plane fit scheme aims to estimate a vector n_p perpendicular to the local plane around each event. The directional components of this vector enclose the spatial and temporal information of the moving edge that triggered this event. (Benosman et al., 2013) formulated this scheme by using the concept of events mapping.

$$\sum_e : \mathbb{N}^2 \mapsto \mathbb{R} \quad (11)$$

$$(x,y) \mapsto \sum_e (x,y) = t \quad (12)$$

Where events pixel coordinates (x,y) are mapped along the time axis t.

2.4.1 Iterated Fit

Since time is a monotonically increasing function of space we can assert that the partial derivatives of $\sum_e$ are non zero increasing functions, then the usage of

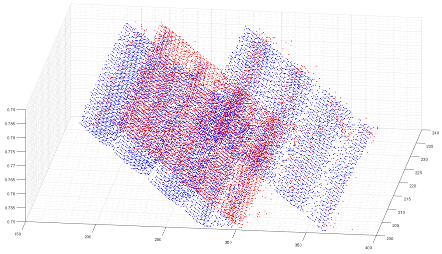

Figure 3: Events Plane-like shapes due to a beam motion (planes correspond to different edges of the beam).

the inverse function theorem around each event is possible as

$$\sum_{ex}(x, y_0) = \frac{d\sum_{e|y_0}}{dx} = \frac{1}{v_x(x, y_0)} \tag{13}$$

$$\sum_{ey}(x_0, y) = \frac{d\sum_{e|x_0}}{dy} = \frac{1}{v_y(x_0, y)} \tag{14}$$

where $\nabla \sum_e = (\frac{1}{v_x}, \frac{1}{v_y})$ is the change gradient vector of time w.r.t space. At each event arrival, a plane is fitted to get the optical flow values using the events packed in a spatio-temporal local neighborhood of $L \times L \times 2\Delta t$. The event under test is used as the center of this neighborhood, accordingly, a plane equation should be satisfied within a σ_1 threshold. Each event fits a plane as in:

$$ax + by + ct + d = 0 \tag{15}$$

where $(a\ b\ c\ d)^T$ are the plane parameters. After fitting a plane for the present event with its neighborhood, a check is carried out to make sure that all events belong to the same surface of the plane within a threshold σ_2. In case an event does not belong to the surface, it is rejected and the same process is repeated until a plane fits within the specified threshold where the optical flow components are $v_x = -c/a$ and $v_y = -c/b$.

2.4.2 Robust Single Fit

We apply the iterations to make sure that all the events belong to the estimated surface impose a strict conditions that would lead to performance deterioration. (Rueckauer and Delbruck, 2016b) propose to use only a single fit while changing the optical flow equations to be

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{1}{|\mathbf{g}|^2}\mathbf{g} = \frac{-c}{a^2 + b^2}\begin{bmatrix} a \\ b \end{bmatrix} \tag{16}$$

where $\mathbf{g}$ $(\frac{-a}{c}\ \frac{-b}{c})^T$ is the gradient to avoid infinity values of optical flow .

2.4.3 Savitzky-Golay Plane Fit

To add a smoothing effect on a noisy events plane surface, a Savitizky-Golay two-dimensional filter is applied to obtain the plane parameters the same way it is used in Lucas-Kanade (see 2.3.3) while some modifications to map a plane are applied. a_{pq} is calculated from the least-square scheme. The plane equation is considered with $c = -1$. The polynomial equation $SG(x, y) = a_{00} + a_{01}y + a_{10}x$ is now mapped to the plane equation $ax + by + ct + d = 0$.

2.4.4 Regularised Plane Fit

To speed up the computation of the local plane fit algorithm and get higher accuracy, (Mueggler et al., 2015) use RANSAC instead of the optimization algorithm while reducing the neighborhood to be $L \times L \times \Delta t$, taking into account only events created prior to the present event. They use a local plane fit and a regularization step to refine events' lifetime estimation, where the lifetime is the time of existence of an event after its creation and defined as: $\sqrt{a^2 + b^2}/c$. The output of the plane fitting algorithm is then used as input of the regularization scheme to smooth the estimated plane parameter by enforcing optimization in the temporal direction. The assumption of constant velocity is used to predict the lifetime of the event in the flow direction $\hat{t}_e$. The error term to be used is defined as a measure of confidence of local plane fit estimation:

$$\Delta t_{err} = |t_e - \hat{t}_e| \tag{17}$$

so the regularized plane vector is

$$\mathbf{n_R} = \arg\min(||A\mathbf{n} - \mathbf{b}||^2 + \lambda(\Delta t_{err})||\mathbf{n} - \hat{\mathbf{n}}_\mathbf{i}||^2) \tag{18}$$

where $\mathbf{n}$ is the estimated plane parameters vector $(\frac{a}{d}\ \frac{b}{d}\ \frac{c}{d})$ and $\hat{\mathbf{n}}_\mathbf{i}^T$ is the predicted one obtained from neighbourhood events. The term $||A\mathbf{n} - \mathbf{b}||$ is the least-square error of the local plane fit where $\mathbf{b}$ is a vector of ones (equation 15 normalized w.r.t d). The regularization term $\lambda(\Delta t_{err})$ is an empirical function of Δt_{err} mainly used to refine events' lifetime. The usage of plane normal drives the regularization to smooth the optical flow estimation as well while focusing the optimization on the time component. This guarantees a smoother but not necessarily an optimal performance for optical flow estimation.

3 DATA-SET SCENARIOS

In order to have a quantitative evaluation of the presented algorithms, an optical flow ground truth should be provided. The paradigm shift influenced by DVS cameras requires new approaches to create optical flow ground-truth while assuring very fine accuracy which is rarely found in the state of the art. For these reasons, we exploit VICON system endowed with high precision and accuracy and employed a new methodology to create the ground truth. The VICON system tracks the PROPHESEE event-based camera that also provides IMU readings to create the needed data-sets. In order to test the reliability of the developed algorithms, datasets featuring in-plane rotation, in-plane translation both at various speeds and also a free-hand motion scenario have been recorded. In all these scenarios, the camera is moving in front of a tracked checkerboard. Checkerboard rigid transform relative to the camera is obtained with the VICON tracking system allowing us to project the checkerboard grid in a camera frame. Reference frames of our experimental setup are explained in Fig. 4. The goal of the various scenarios is to better understand the capabilities and shortcomings of each implemented algorithm. Fig. 1 shows the real environmental setup of our dataset.

4 SENSORS CALIBRATION

The VICON system is calibrated easily using the provided software and it returns the states of the point of origin of the board along with the camera pose. However, our goal is to obtain apparent motion ground truth suitable to optical flow evaluation rather than extrinsic parameters of the camera as used in SLAM or visual-odometry. And since tested algorithms will only provide apparent motion, we propose to make use of the camera's embedded IMU as a middle transformation to estimate between the VICON and the camera frames so that the transformations for the markers on the board and on the camera are (see Fig. 5):

$$
\begin{aligned}
T^{c}_{vic} &= T^{c}_{IMU} \times T^{IMU}_{vic} && (19)\\
T^{c}_{B} &= T^{c}_{IMU} \times T^{IMU}_{vic} \times T^{vic}_{B} && (20)
\end{aligned}
$$

4.1 Intrinsic Calibration

Camera and IMU need to be intrinsically calibrated before implementing the extrinsic calibration.

4.1.1 IMU Intrinsic Calibration

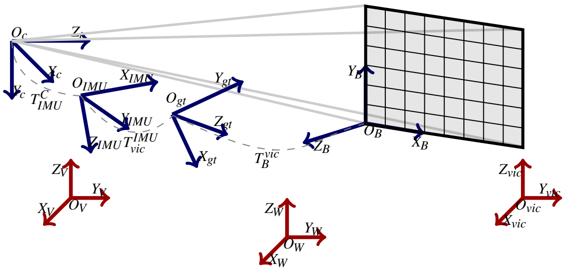

Figure 4: Different reference frames (fixed and moving).

To understand the IMU intrinsic calibration, we need to point out that the IMU parameters to be estimated are divided into two categories, deterministic and random. Deterministic parameters are scale factor, misalignment error and bias offsets. Random errors are the bias residuals and white noise added to the signal, the IMU can be modeled as follows:

$$
\begin{aligned}
\omega_{IMU} &= [I + M_g]\omega + b_g + \delta b_g + \varepsilon_g && (21)\\
a_{IMU} &= [I + M_a]a + b_a + \delta b_a + \varepsilon_a && (22)
\end{aligned}
$$

where ω_{IMU} and a_{IMU} are 3D vectors, rotational velocity and acceleration obtained by the IMU. M_g and M_a are the matrices that contains misalignment errors of the gyroscope and accelerometer respectively. b_g and b_a are the offset bias of the gyroscope and accelerometer. δb_g and δb_a are the bias residual of the gyroscope and accelerometer that changes with very low frequency. ε_g and ε_a are white noise of the gyroscope and accelerometer respectively. Using the six-position calibration (El-Diasty and Pagiatakis, 2010), we obtain the deterministic parameters of the IMU except for the matrix M_g because of the inability to have accurate known excitation source for the gyroscope, which still can be suppressed in a scheme of fusion.

We use Allan variance modeling (El-Sheimy et al., 2007) in order to estimate the parameters that controls the modeling of random IMU parameters.

4.1.2 Camera Intrinsic Calibration

The process of calibrating an event-Based camera is similar to the process for standard cameras, except in the image creation phase. In order to overcome the nature of event-based cameras, we use flashing patterns to adjust the sharpness and lens focus. We generate a flashing checkerboard on a screen and use the resulting images as an input for the embedded MATLAB camera calibrator toolbox to estimate the intrinsic parameters of the camera.

4.2 Extrinsic Calibration

The VICON system and camera need to be calibrated both spatially and temporally. The extrinsic calibration process is divided into temporal synchronization and spatial alignment to make sure the outputs from different systems are adjusted perfectly.

Algorithm 1: IMU-VICON Calibration.

```
Data: $\{\Theta_{IMU}\}_{i=0}^{N-1}$ , $\{\Theta_{vic}\}_{i=0}^{N-1}$
Result: $q$ , $\Delta t$
Initialize: $check = true$ , $itr = 1$ , $\varepsilon = small\ value$
while $check$ = $true$ do
    $\mu_{IMU} = \frac{1}{N}\sum_{i=0}^{N-1}\Theta_{IMU_i}$ , $\mu_{vic} = \frac{1}{N}\sum_{i=0}^{N-1}\Theta_{vic_i}$
    $\Theta_{IMU} = \Theta_{IMU} - \mu_{IMU}$ , $\Theta_{vic} = \Theta_{vic} - \mu_{vic}$
    Calculate: $S_{mn} = \sum_{i=0}^{N-1}\Theta_{IMU_m}\Theta_{vic_n}$          ▷ 9 values
    Construct: $N_{4\times4}$ matrix          ▷ see (Sola, 2017)
    Solve: $\{\lambda, V\} = eig(N)$
    $q_{itr} = V_{\lambda_1}$          ▷ q is the vector corresponds to maximum λ
    project: $\Theta_{MU} = R_q\Theta_{IMU}$
    Solve: $\Delta t_{itr} = xcorr(\Theta_{IMU}, \Theta_{vic})$
    Shift: $\Theta_{IMU} = \Theta_{IMU}(\Delta t : end)$
    $itr{+}{+}$
    if $\frac{1}{N}\sum_{0}^{N-1}|\Theta_{IMU_i} - \Theta_{vic_i}| < \varepsilon$ then
        check = false
    end
end
$q = \prod_{n=1}^{itr} q_n$
$\Delta t = \sum_{n=1}^{itr}\Delta t_n$
```

4.2.1 VICON-IMU Extrinsic Calibration

The correspondence problem of two different frames of reference that observe the same states are widely known as Wahba's problem (Wahba, 1965), where the author defined it, for given two 3D sets of measurements $\{X_i\}_{i=0}^{N-1}$ and $\{Y_i\}_{i=0}^{N-1}$, as the least square minimization of

$$\underset{\{s,\mathbf{R},\mathbf{t}\}}{\operatorname{argmin}} \sum_{i=0}^{N-1} |Y_i - s(\mathbf{R}X_i + \mathbf{t})|^2 \tag{23}$$

where s, $\mathbf{R}$ ant $\mathbf{t}$ are scale, rotation and translation between the two frames respectively. The main two assumptions in order to be able to solve this minimization problem are that noise is suppressed in both measurements and that they are temporally synchronized. Since the IMU measurements suffer from high noise due to integration effect, we choose to use the angles to be injected in the minimization problem. We use an error state Kalman filter (Sola, 2017) to obtain the accurate measurements. To solve the alignment problem, we adopt an iterative scheme since each of the spatial and temporal alignments needs one another as a prerequisite to be fulfilled. We solve for an initial transformation between the two frames using Horn's reformulation of Wahba's problem (Horn, 1986) (see algorithm 1).

$$\underset{\mathbf{q}}{\operatorname{argmin}} \sum_{i=0}^{N-1} (\mathbf{q} \times \Theta_{IMU} \times \mathbf{q}^*)\Theta_{vic} \tag{24}$$

where Θ_{IMU} and Θ_{vic} are the angles measured in IMU and VICON frames and $\mathbf{q}$ is a unit quaternion that represents a rigid body transformation. We project the IMU readings in the VICON frame to apply cross correlation between the two signals to find Δt that would temporally align them. We repeat until the difference between the estimated calibration parameters reaches convergence. It is noted that after two iterations, convergence is fulfilled. Fig. 5 shows IMU and VICON measured angles before and after calibration. To make sure that every sequence is correctly correlated, we calculate the mean absolute difference between the VICON angles and IMU angles after being transformed in VICON frame, results are shown in Table 1.

Table 1: VICON / IMU angles comparison.

sequence	$\phi[°]$	$\theta[°]$	$\psi[°]$
rotate_low	0.7893	0.5988	1.5579
rotate_high	1.2885	0.9427	0.4945
translate_low	1.2347	1.5135	0.4813
translate_high	1.4031	1.6943	0.4838
free_motion	0.3923	0.5544	0.3923

4.2.2 Camera-IMU Calibration

After getting the spatio-temporal calibration between the VICON and IMU we need to do the same between the IMU and the Camera. We used Kalibr toolbox (Furgale et al., 2014) to get the spatial transformation between the camera and the IMU. Since Kalibr provides a temporal difference only for the provided data-set used for calibration (which was totally different from the data-set used for our comparison), we use the concept of cross correlation between the IMU absolute rotational velocity and the events frequency in order to find the best time shift. The choice of events frequency to be correlated with absolute rotational velocity is similar to the temporal synchronization used in (Censi and Scaramuzza, 2014) since with a higher velocity more events would be triggered per second, so the best synchronization will correspond to the best matching these two signals together.

5 GROUND TRUTH CREATION AND COMPARISON

Next step after calibration is to recreate accurate events positions of the checker board to be projected

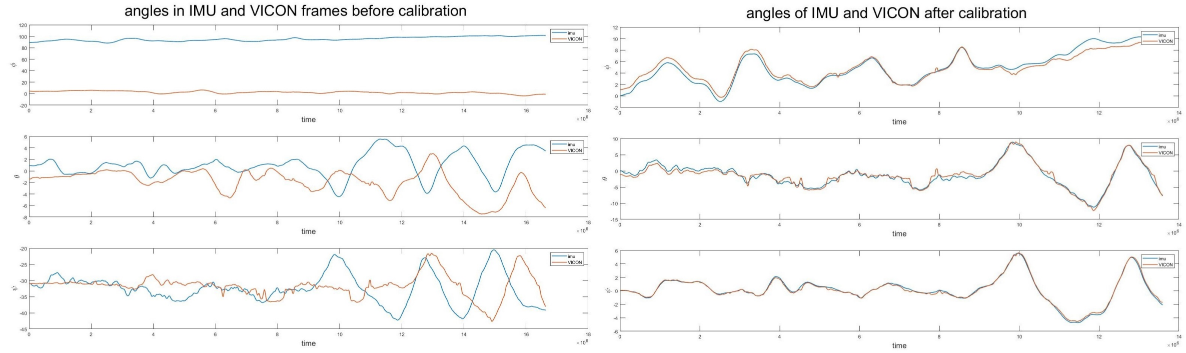

Figure 5: Left: an example of angles (namely the roll angle) in VICON and IMU frames without alignment. Right: the angles after being transformed in the IMU frame is shown to be perfectly aligned.

Table 2: Average angular error of evaluated algorithms in degrees (DS stands for Direction Selective filter, LK for Lucas-Kanade, LP for Local Plane fit and SG for Savitsky-Golay).

Algorithm	rotate_low	rotate_high	translate_low	translate_high	move_free
LK Backward	35.158 ± 27.547	41.585 ± 15.715	32.314 ± 12.642	35.526 ± 12.642	22.975 ± 23.839
LK Central	23.587 ± 19.481	21.135 ± 17.161	19.135 ± 10.362	20.264 ± 9.321	18.625 ± 20.881
LK SG	16.754 ± 9.571	15.487 ± 10.131	14.361 ± 3.752	13.595 ± 8.839	10.576 ± 11.935
DS	78.391 ± 20.507	71.775 ± 21.781	53.651 ± 17.432	57.205 ± 19.381	72.890 ± 68.497
LP Single Fit	12.835 ± 5.718	14.304 ± 5.203	6.914 ± 3.567	6.937 ± 4.893	9.350 ± 7.761
LP SG	8.905 ± 3.107	7.751 ± 4.978	**5.744 ± 2.742**	**5.065 ± 1.347**	**8.678 ± 7.271**
LP Regularized	**5.871 ± 4.178**	**6.170 ± 4.108**	5.863 ± 3.108	4.709 ± 2.938	18.707 ± 15.463

in the camera frame using equation 20. We create 3D points which reside on the edges of the 9×7 checkerboard at time t_0 and then transform these points at each time step in the camera frame. The intrinsic calibration is used to project the transformed checkerboard in the pixels frame while undistorting the scene. To create the optical flow ground truth values, we follow (Heeger and Jepson, 1992) proposition, exploiting the fact that the checkerboard is a rigid plane. If we have $\mathbf{V}$ and $\mathbf{W}$ as translation and rotation speeds of the 3D scene then the optical flow will be approximated by the 2D projection of the 3D motion according to the equation:

$$\mathbf{U}(x,y) = \frac{1}{Z}\mathbf{A}(x,y)\mathbf{V} + \mathbf{B}(x,y)\mathbf{W} \tag{25}$$

where,

$$\mathbf{A}(x,y) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \tag{26}$$

$$\mathbf{B}(x,y) = \begin{bmatrix} (xy)/f & -(f+x^2/f) & y \\ (f+x^2/f) & -(xy)/f & -x \end{bmatrix} \tag{27}$$

With the created ground truth, we adopt the metrics in (Baker et al., 2011) for quantitative comparison. First metric is the average endpoint error ($AEPE$) which is defined as the average value of the vector distance between the estimated motion $\mathbf{u}$ and the ground-truth $\hat{\mathbf{u}}$:

$$AEPE = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{u}_i - \hat{\mathbf{u}}_i|| \tag{28}$$

The second metric is the average angular error (AAE) which is defined as the average angle between the estimated motion $\mathbf{u}$ and the ground truth $\hat{\mathbf{u}}$:

$$AAE = \frac{1}{N}\sum_{i=1}^{N} cos^{-1}\left(\frac{\hat{\mathbf{u}}_i^T \mathbf{u}_i}{||\hat{\mathbf{u}}_i||||\mathbf{u}_i||}\right) \tag{29}$$

To make sure that the created data-set are reliable and that any calibration error is omitted, we randomly select two consequent VICON frames f_1 and f_2 and use the optical flow ground truth to project events triggered between those two frames to the frame f_2 (see Fig. 7). We use the mean of absolute difference between the projected events and the created frame as an error metric. Errors after events projection show acceptable results where the projected events are aligned with the boundaries of the synthetic board and the mean of absolute difference did not exceed the bound of 2 pixel. (see Fig.6)

6 RESULTS

In order to show the comparison between the different algorithms without being biased with high value errors, we use the mean and variance of optical flow values to remove extreme outliers. We apply the comparison w.r.t accuracy and computational power needed. Results are demonstrated in Tables 2 and 3 . Since

Table 3: Relative average end point error of the evaluated algorithms.

Algorithm	rotate_low	rotate_high	translate_low	translate_high	move_free
LK Backward	1.324 ± 0.607	1.215 ± 0.749	0.934 ± 0.691	0.883 ± 0.482	1.196 ± 0.405
LK Central	1.057 ± 0.254	0.951 ± 0.512	0.894 ± 0.533	0.811 ± 0.328	1.163 ± 0.677
LK SG	0.921 ± 0.421	0.812 ± 0.458	0.664 ± 0.582	0.521 ± 0.387	0.994 ± 0.369
DS	1.721 ± 0.227	1.851 ± 0.554	1.322 ± 0.363	1.391 ± 0.452	1.679 ± 1.2161
LP Single Fit	1.125 ± 0.248	1.054 ± 0.187	0.891 ± 0.155	0.803 ± 0.183	1.286 ± 0.320
LP SG	**0.681 ± 0.384**	0.621 ± 0.321	**0.347 ± 0.124**	**0.382 ± 0.168**	**0.691 ± 0.456**
LP Regularized	0.755 ± 0.327	**0.604 ± 0.247**	0.404 ± 0.203	0.410 ± 0.137	0.999 ± 0.010

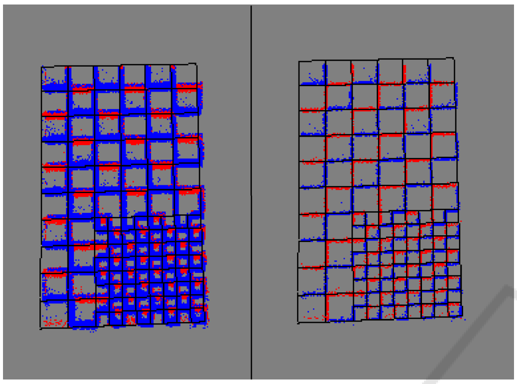

Figure 6: (Left) the events and the synthetic created checkerboard before projecting the events using the optical flow ground truth, (Right) the projected on and off events in the camera frame are sufficiently aligned with the created checkerboard where the mean of absolute difference did not exceed the bound of 2 pixel.

algorithms have been tested using MATLAB, computation time in itself is not a significant metric but relative differences between algorithms indicate the computational power needed.

6.1 Average Angular Error

The obtained results show that the Direction Selection filter has -as expected- the lowest angular error accuracy in all the tested sequences since it provides only a notion of the motion in 8 discrete directions. In the three categories of algorithms, we can conclude that the Local Plane algorithms outperform the other algorithms. The addition of optimal regularization significantly refine the estimation of optical flow's direction but did not provide the best estimation for all sequences because the regularization term optimizes only the temporal component of the plane's normal and not the spatial components. The usage of Savitsky-Golay filter enhances the accuracy of angular error which did not vary much compared to the Regularized plane fit. The performance of the algorithms under test was shown to be always better for scenarios featuring translations while noting that the Regularized plane fit always provide better performance than Savitsky-Golay Local Plane fit because the smoothness effect may not be prominent if the optical flow varies much in neighbourhoods which is the case while rotating.

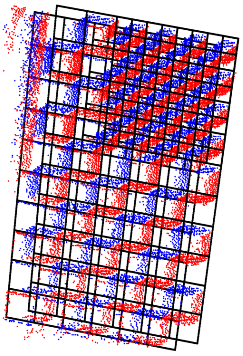

Figure 7: In black: two consequent frames created by the projection of the VICON system. Positive triggered events are shown in red and negative events in blue.

6.2 Average End Point Error

Using the Direction Selective filter feature the least accuracy due to the lack of events used to estimate the optical flow. Savitsky-Golay Local Plane fit is seen to always provide better endpoint error (except for high rotation sequences with minor difference). Regularizing the local plane optical flow helps to enhance the end point error but could not exceed the accuracy of Savitsky-Golay filter for the reason mentioned in the previous section that the regularization enforces only temporal accuracy.

6.3 Computation Time

Although the Direction Selection filter did not provide the best accuracy, it was the fastest algorithm to be performed with significantly shorter computation time, which means that it can be used as indication or as a preliminary step to add direction information for each event. On the other hand, using Regularization -while not being the best w.r.t performance in all the cases- add a significant rise in computation time. Because of this aspect, we question the relevance of

Table 4: Computation times needed for calculations per event.

algorithm	computational time [μs]
DS	**77.465 ± 32.540**
LK Backward	312.974 ± 170.568
LK Central	384.489 ± 184.946
LK Savitzky-Golay	264.913 ± 94.412
LP Single Fit	173.89 ± 120.973
LP Savitsky-Golay	129.749 ± 112.549
LP regularized	536.486 ± 173.914

integrating this algorithm in a more complex scheme. The Lucas-Kanade algorithm provides relatively good error accuracy but is not the best in terms of computational cost. Presenting Savitsky-Golay filter for any algorithm (Lucas-Kanade or Local Plane fit) always refines the accuracy while significantly reducing computation time.

7 CONCLUSION

In this paper, we present a methodology to compare state-of-the-art event-based optical flow algorithms and show their performance in the context of robotic applications. The suggested evaluation led us to propose an event-based optical flow ground-truth dataset using a VICON system. Our study reveals that all the evaluated algorithms need a lot of tuning w.r.t the time interval to calculate optical flow while also tuning many thresholds to get the best optical flow values. Our future work will then focus on proposing adaptive solutions to make these algorithms perform better for various scenes in robotic applications while improving their global performance.

REFERENCES

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31.

Barranco, F., Fermuller, C., Aloimonos, Y., and Delbruck, T. (2016). A dataset for visual navigation with neuromorphic methods. *Frontiers in neuroscience*, 10:49.

Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H., and Bartolozzi, C. (2013). Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2):407–417.

Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C., and Srinivasan, M. (2012). Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37.

Censi, A. and Scaramuzza, D. (2014). Low-latency event-based visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710. IEEE.

Delbrück, T. (2008). Frame-free dynamic digital vision. In *Proceedings of International Symposium on Secure-Life Electronics, Advanced Electronics for Quality Life and Society, Univ. of Tokyo, Mar. 6-7, 2008*, pages 21–26. nternational Symposium on Secure-Life Electronics, Advanced Electronics for . . . .

El-Diasty, M. and Pagiatakis, S. (2010). Calibration and stochastic m odelling of i nertial na vigation s ensor errors. *Positioning (POS) Journal Information*, page 80.

El-Sheimy, N., Hou, H., and Niu, X. (2007). Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on instrumentation and measurement*, 57(1):140–149.

Furgale, P., Maye, J., Rehder, J., and Schneider, T. (2014). Kalibr: A unified camera/imu calibration toolbox.

Heeger, D. J. and Jepson, A. D. (1992). Subspace methods for recovering rigid motion i: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117.

Horn, B. K. P. (1986). Closed-form solution of absolute orientation using uni t quaternions.

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128×128 120 db 15μ s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI*.

Menze, M., Heipke, C., and Geiger, A. (2015). Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*.

Mueggler, E., Forster, C., Baumli, N., Gallego, G., and Scaramuzza, D. (2015). Lifetime estimation of events from dynamic vision sensors. In *2015 IEEE international conference on Robotics and Automation (ICRA)*, pages 4874–4881. IEEE.

Rueckauer, B. and Delbruck, T. (2016a). Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Frontiers in neuroscience*, 10:176.

Rueckauer, B. and Delbruck, T. (2016b). Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Frontiers in neuroscience*, 10:176.

Sola, J. (2017). Quaternion kinematics for the error-state kalman filter.

Wahba, G. (1965). A least squares estimate of satellite attitude. *SIAM review*, 7(3):409–409.