ESA GNC & ICATT Conference 2026
28 Sep. – 2 Oct. 2026, Sevilla, Spain
https://atpi.eventsair.com/gncicatt/

Event-Based Lunar Dataset and Spiking Neural Network for Low Lunar Orbit and Landing Scenarios via Optical Flow Estimation

Elena Pilo[1,*], Paolo Lunghi[1], Paolo Panicucci[1], and Francesco Topputo[1]

[1]*Politecnico di Milano, Via Giuseppe la Masa, 34, 20156, Milan, Italy*
[*]*Corresponding author: elena.pilo@polimi.it*

ABSTRACT

Event-based vision sensors offer high dynamic range, low latency, and sparse data representation, making them promising for autonomous lunar navigation and landing under extreme illumination and motion conditions. Their asynchronous operation also aligns naturally with Spiking Neural Networks (SNNs), enabling low-power, real-time perception. However, progress in event-based perception for space applications is limited by the lack of realistic datasets with accurate motion ground truth and dedicated architectures for event-based data. This work presents two complementary contributions: an event-based lunar vision dataset and a UNet-like SNN for dense optical flow estimation. The dataset includes Low Lunar Orbit and representative landing sequences, providing high-resolution rendered images, event streams, depth maps, and dense optical-flow ground truth. Images are generated using ABRAM, a physics-based radiometrically validated renderer, while event streams are produced with v2e to emulate realistic event-camera behaviour. Depth and optical flow are analytically derived from the simulated geometry and camera motion. The proposed SNN estimates dense optical flow directly from event streams using a multi-scale encoder–decoder architecture. Together, the dataset and model provide a framework for benchmarking event-based perception algorithms for autonomous lunar navigation.

1 Introduction

Neuromorphic vision sensors, or event-based sensors, offer a novel sensing paradigm which attracted growing interest across space applications in recent years. Rather then detecting brightness intensity as standard frame based cameras, in an event-based camera each pixel continuously monitor the logarithm of its brightness and generate a signal whenever the variation in logarithmic brightness exceeds a predefined threshold, yielding an asynchronous sparse stream of events with a temporal resolution in the order of microseconds. Because of their sensing paradigm, their dynamic range (120/140 dB) is much higher than the one of frame-based cameras (60/80 db), and because of their sparse output, their power is typically on the order of tens to a few hundred milliwatts, compared with hundreds of milliwatts to several watts for many frame-based cameras [6]. These characteristics make them particularly attractive for many space applications, particularly for resource constrained platforms. Indeed, although the understanding of their behaviour and performance in the context of space missions is still in its early stages, several works have demonstrated their potential in promising scenarios, including event-based odometry for planetary robots [16], high frequency star tracking [22], spacecraft pose estimation [12, 15] and landing [1, 5, 24]. Lunar landing in particular often demand low-latency perception under extreme and highly variable illumination conditions. In these conditions conventional frame-based cameras may fail, because their fixed exposure and limited dynamic range may lead to motion blur and saturation or under-exposure in high-contrast scenes. Among the perception tasks most relevant to autonomous navigation, optical flow estimation plays a central role, as it provides direct information on relative motion between the sensor and the observed

© *Politecnico di Milano*

scene, which can in turn be used to estimate egomotion during descent [5]. Estimating optical flow directly from the asynchronous, sparse output of event-based sensors, however, calls for processing paradigms that depart from conventional dense, frame-based computation, in order to fully exploit the temporal resolution and sparsity of the event stream. In this respect, Spiking Neural Networks (SNNs) are being increasingly investigated in the machine learning domain, as they process information as discrete, asynchronous spikes rather than dense tensors, and thus naturally align with the event-driven sensing paradigm. Event-based optical flow estimation has been extensively investigated outside the space domain, with several ANN and SNN architectures achieving accurate and efficient flow prediction from event streams [3, 9, 14, 25, 27]. In the space domain, however, event-based optical flow remains largely unexplored, mainly due to the absence of realistic datasets with reliable motion ground truth and of architectures tailored exploit event-based data in space relevant scenarios. A notable work in this direction, which aims to close these gaps was the generation of a synthetic event-based lunar landing dataset [1], later used as the basis for the ELOPE challenge, the first competition dedicated to event-based egomotion estimation from optical flow in a lunar landing scenario [5]. While the results of ELOPE demonstrated the feasibility of the task, they also highlighted the current lack of dedicated, SNN-based solutions capable of fully exploiting the asynchronous nature of event data, as well as the need for larger and more diverse datasets covering different lunar navigation phases. This motivates the two contributions presented in this work: a realistic event-based lunar vision dataset with dense optical flow ground truth, and a dedicated UNet-like SNN architecture for dense optical flow estimation from event streams. The dataset contains sequences of the Moon surface from a Low Lunar Orbit at 100 km altitude, and from the initial part of the descent, up to approximately 40 km altitude, including also event streams, depth maps and dense ground truth optical flow. Images and depth maps are rendered at 100 FPS using ABRAM [19] a physics-based radiometrically validated renderer for celestial bodies. The event streams are instead generated using the v2e software, while optical flow is analytically derived from the simulated geometry and camera motion. After having introducted the considered scenario, section 3 will detail the dataset generation which is then used to train the network presented in section 4. Finally section 5 will present the results and network performance.

2 Scenario

Trajectory Generation

The scenario considered in this work begins from a Low Lunar Orbit (LLO), corresponding to a circular parking orbit at 100 km altitude from which the descent is initiated. From this parking orbit the spacecraft performs a tangential manouvre to lower the orbit perilune, inserting itself in a ballistic elliptical orbit. This first phase of the descent is tipically referred to as the coasting phase, with a pericenter set at 20-15 km altitude for safety reasons. After this phase, a powered descent manoeuvre takes place, initiating the main braking phase. In this work, the dataset is limited to the LLO and coasting phase, as the final part of the descent could not be adequately represented due to the resolution limitations of the available lunar surface maps, as discussed in Section 3.
To simulate the ground truth dynamics, the spacecraft translational motion is propagated according to Eq. (1), where the total acceleration is given by the sum of the three contributions: the lunar gravitational acceleration $\mathbf{g}_M$, Earth's and Sun's third-body perturbations, $\mathbf{g}_E$ and $\mathbf{g}_S$, and the acceleration due to Solar Radiation Pressure $\mathbf{a}_{SRP}$, modelled considering a cannonball model.

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v}, \\ \dot{\mathbf{v}} &= \mathbf{g}_M + \mathbf{g}_E + \mathbf{g}_S + \mathbf{a}_{SRP}. \end{aligned} \tag{1}$$

The lunar gravitational acceleration is computed from a higher-order lunar gravity field, using the LP165P1 Spherical Harmonic model of the Moon's gravity field, truncated at the 10th order. It is obtained as the first derivative of the gravitational potential U according to Eq. (2), as reported in [7]

$$\mathbf{g}_M = \frac{\partial \mathbf{U}}{\partial \mathbf{x}} = \frac{\partial \mathbf{U}}{\partial r}\frac{\partial r}{\partial \mathbf{x}} + \frac{\partial \mathbf{U}}{\partial \varepsilon}\frac{\partial \varepsilon}{\partial \mathbf{x}} + \frac{\partial \mathbf{U}}{\partial B_{nm}}\frac{\partial B_{nm}}{\partial \mathbf{x}} \tag{2}$$

where r is the magnitude of the position vector, ε is the sine of latitude and B_{nm} is a linear combination of the un-normalized cosine and sine gravity coefficients retrieved from the LP165P1 Spherical Harmonic model of

the Moon's gravity field [18]. During both LLO and coasting phases, the camera boresight, coincident with its Z-axis, is aligned with the Nadir direction. The remaining two body axes are fixed by the Sun direction: the X-axis lies orthogonal to the plane defined by the spacecraft, the Sun and the Moon, while the Y-axis completes the right-handed triad. The corresponding angular velocity, required for the motion field computation, is reconstructed via the standard quaternion kinematics relation in Eq. 3.

$$\omega_{BN} = 2\begin{bmatrix} q_0 I_3 + [\mathbf{q}_v\times] \\ -\mathbf{q}_v^T \end{bmatrix}^T \dot{\mathbf{q}}_v\,, \qquad \mathbf{q}_{BN} = \begin{bmatrix} q_0 & \mathbf{q}_v^T \end{bmatrix} \tag{3}$$

2.1 Camera model and ground sample distance

The camera is modelled with the pinhole camera model [10] with intrinsic parameters summarised in Table 1. The resolution, as well as the pixel size, correspond to the Sony IMX6368 event sensor, used in many Prophesee cameras [20]. The focal length is instead chosen as a typical value for other lunar landers [21, 26].

Parameter	Value
Resolution [px]	1280×720
Focal length [mm]	8
Pixel pitch [μm]	4.86
Field of view [deg]	42.5×24.7

Table 1 *Camera intrinsic parameters.*

To then select the maps at the correct resolution, it is fundamental to analyse the Ground Sampling Distance (GSD) along the trajectory. Once the focal length and the pixel pitch are fixed, the GSD is only function of the altitude. Fig. 1 reports this quantity over the altitude range spanned by the coasting phase. The GSD for the LLO is equal to the value at $t = 0$, since it coincides with the de-orbiting manouvre to initiate the coasting arc.

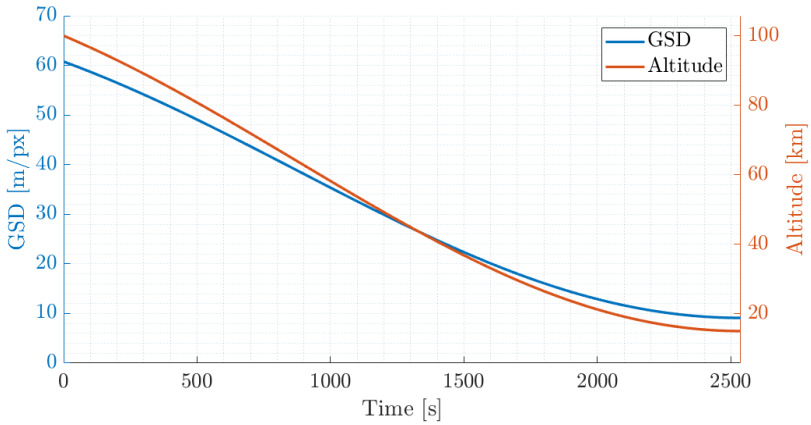


Figure 1 *Camera Ground Sample Distance and altitude as a function of time during coasting. Initial time corresponds to the de-orbiting manoeuvre performed from the LLO.*

2.2 Motion Field Generation

The ground-truth flow field is derived analytically from the relative motion between the camera and the scene, rather than estimated from consecutive images. Optical flow is classically defined as the apparent motion of brightness patterns in an image sequence, and its estimation from real images relies on the brightness constancy assumption, *i.e.* that the observed radiance of a scene point remains constant between consecutive frames. Under

this assumption, the estimated optical flow coincides with the true *motion field*, defined as the projection onto the image plane of the actual 3D velocity of scene points relative to the camera. Since the motion field is a purely geometric quantity, fully determined by the camera motion and scene depth, it can be computed in closed form from the known simulation geometry. This is the standard approach adopted in synthetic optical flow datasets, and is the strategy followed in this work: the ground truth thus corresponds to the ideal motion field, which is used as a proxy for the optical flow.
A world point P_W is first expressed in the camera frame via the rigid-body transform:

$$P_C = [X \;\; Y \;\; Z]^\mathsf{T} = R_{CW}(P_W - C). \tag{4}$$

where C is the camera's position and R_{CW} its attitude. This camera-frame point P_C is then projected onto the image plane through the pinhole camera model, according to Eq. (5) [10].

$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z} \tag{5}$$

where f is the focal length in pixels and (x, y) are image-plane coordinates relative to the principal point.
For a static scene observed by a camera translating with velocity $v = (v_x, v_y, v_z)$ and rotating with angular velocity $\omega = (\omega_x, \omega_y, \omega_z)$ (both expressed in the camera frame), a world point that is fixed, appears to move relative to the camera as in Eq. (6).

$$\dot{P}_C = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = -v - \omega \times P_C = \begin{bmatrix} -v_x - \omega_y Z + \omega_z Y \\ -v_y - \omega_z X + \omega_x Z \\ -v_z - \omega_x Y + \omega_y X \end{bmatrix} \tag{6}$$

The motion field $(u, v) = (\dot{x}, \dot{y})$ is obtained by differentiating the pinhole projection in Eq. (5) with respect to time and then substituting $\dot{X}, \dot{Y}, \dot{Z}$ from Eq. (6) and the inverted projection relations X, Y, Z in Eq. (5).
The final motion field is therefore given by the contribution of the translational and rotational terms as in Eq. (7).

$$\begin{aligned} u(x,y) &= \underbrace{\frac{1}{Z(x,y)}\left(-f\, v_x + x\, v_z\right)}_{u_T} + \underbrace{\frac{xy}{f}\,\omega_x - \left(\frac{x^2}{f} + f\right)\omega_y + y\,\omega_z}_{u_R} \\ v(x,y) &= \underbrace{\frac{1}{Z(x,y)}\left(-f\, v_y + y\, v_z\right)}_{v_T} + \underbrace{\left(\frac{y^2}{f} + f\right)\omega_x - \frac{xy}{f}\,\omega_y - x\,\omega_z}_{v_R} \end{aligned} \tag{7}$$

The translational component (u_T, v_T) depends on the inverse scene depth $1/Z$, whereas the rotational component (u_R, v_R) is independent of depth and results solely from the camera's angular velocity. In this work $Z(x, y)$ is the per-pixel depth map rendered alongside each frame.

3 Dataset

The dataset is primarily intended to train the network presented in Sect. 4.2, and its composition is driven directly by that goal: for the network to generalize beyond the specific conditions it is trained on, it must be exposed during training to a representative range of both the surface appearance and the illumination conditions it will encounter operationally. Surface appearance varies substantially across the lunar surface, and a network trained on a narrow set of terrain types risks learning shortcuts tied to that specific appearance rather than the underlying optical-flow cues. Similarly, illumination conditions are highly variable depending on the phase angle and the area of the Moon and has a direct effect on the visual and event signal. Therefore the dataset must span a wide range of illumination conditions rather than a single lighting regime. For this reason, short segments are extracted from the nominal LLO and coasting trajectories described in Sect. 2 and scattered across different locations on the Moon and illuminated under randomized Sun directions before being rendered and converted into event streams. The dataset is then partitioned into training, validation and test with a ratio of approximately 70%, 20% and 10%. This results in 177 training sequences, 49 for validation and 32 for testing.

3.1 Trajectory scattering

The same scattering procedure is applied independently to the nominal LLO and coasting trajectories, differing only in the number of sequences generated and in the extent of the lunar surface over which they are generated. These constraints are discussed separately for each regime below. In both cases, short windows of 10 s are extracted from the nominal trajectory at random starting epochs and re-projected onto random locations of the lunar surface. For each sampled sequence, the initial camera position relative to the Moon is kept fixed, while the Moon's attitude quaternion is redrawn randomly. Randomizing the Moon's orientation is therefore equivalent to randomizing the ground-track location of the sequence's starting point on the lunar surface. The Moon's attitude is sampled as a uniformly-distributed random quaternion over $SO(3)$, following the method detailed in [13]. This guarantees that the LLO sequences are uniformly distributed on the lunar sphere. Following this quaternion randomization, the longitude theoretical distribution is expected to be uniform, while the latitude follows a cosine density that vanishes at the poles and peaks at the equator. The Sun direction is likewise randomized independently for each sequence. The phase angle is drawn uniformly in $[0°, 90°]$. The Sun direction is obtained by rotating the camera-to-Moon direction about a random axis perpendicular to it. For simplicity, the Moon and Sun orientation are held fixed within a single sequence, neglecting the Moon's rotation and phase angle variation over the short interval.

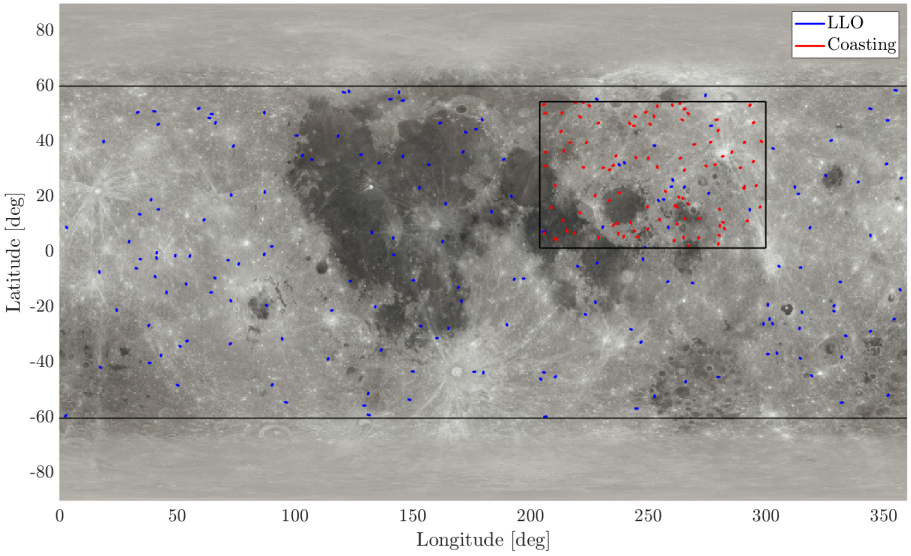


Figure 2 *LLO and Coasting sequences scattered within the selected regions.*

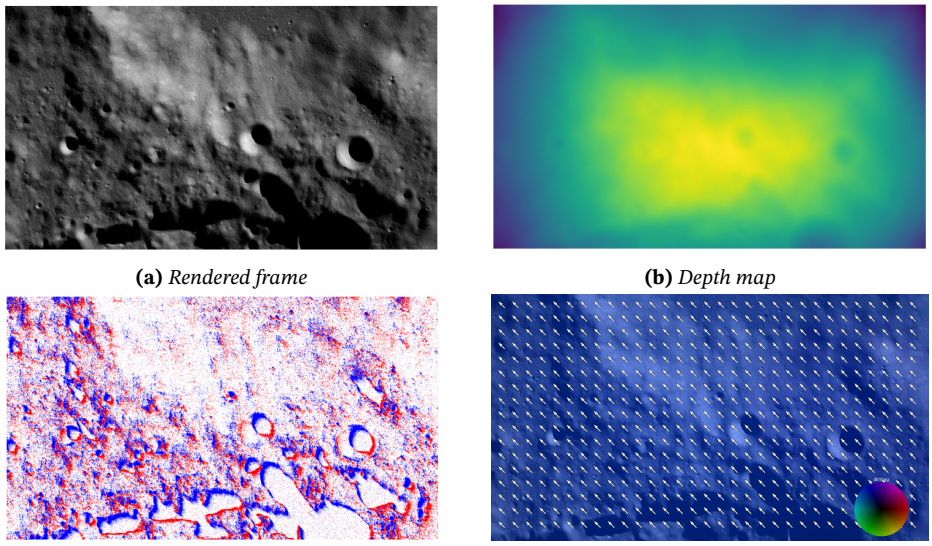
LLO scattering. 200 LLO sequences are generated with the procedure above over the whole sphere. As explained in Sec. 3.2, due to the availability of maps at the required resolution, only the sequences whose starting point falls between latitudes of $\pm 60°$ are then used to produce the synthetic dataset, yielding 164 LLO sequences in total.

Coasting scattering. The coasting phase starts at 100 km altitude but then evolves up to much lower altitudes than LLO, which requires correspondingly higher-resolution surface maps to render geometrically consistent imagery. Only the portion of the coasting arc up to approximately 35 km altitude is used to generate the scattered sequences. Indeed, the last part of the descent, has lower GSD values, as shown in Fig. 1, which would require texture maps with analogous resolution. Because generating synthetic imagery from higher-resolution maps is significantly more time and computationally intensive than for the LLO case, the surface area over which coasting sequences are scattered is restricted to a single region spanning approximately 97° in longitude and 55° in latitude. This computational constraint also results in a smaller number of generated sequences, 94 compared to the LLO case. However, the selected region covers a diverse range of surface features, providing sufficient spatial variation

to generate sequences with different local terrain characteristics, although it does not span the full range of lunar terrain types represented in the LLO dataset. Extending coasting coverage to additional regions is left as future work.

3.2 Synthetic data generation

After having generated the sequences, four data products are generated: rendered images, depth maps, event streams, and per-pixel dense optical flow. Fig. 3 shows an example of the four generated products.



(a) *Rendered frame* **(b)** *Depth map*

(c) *Accumulated events* **(d)** *GT optical flow*

Figure 3 *Example data products for a representative LLO sequence: rendered image, depth map, accumulated event frame, and dense ground-truth optical flow*

Images, depth and optical flow

Synthetic images are generated using ABRAM [19], a physics-based radiometric renderer for celestial bodies, combining albedo maps, Digital Elevation Models (DEM), and normal maps. While albedo and displacement maps are publicly released by scientific teams as open source data, the normal maps are derived from the displacement maps by first constructing a triangular surface mesh from the elevation data and then assigning to each pixel the average normal vector of the neighbouring mesh facets. All sequences are generated with a frame rate of 100 Hz, as it is considered sufficiently high to generate a smooth sequence of images to produce the final stream of events without the need of frame interpolation.

To render the LLO sequences, as the altitude of the orbit remains fixed at 100 km, the 2015 SLDEM at 512 px/deg are used [2, 17]. This dataset, obtained from the co-registered LOLA and Kaguya data, provides a spatial resolution of approximately 60 m/px, between 60°N–60°S. As illustrated in Fig. 1, the GSD at the beginning of the coasting phase, which coincides with the altitude of the LLO, is around 60 m/px. Consequently the 512 px/deg SLDEM provides sufficient resolution to generate images with adequate fidelity.

During the coasting phase, the GSD spans from 60 m/px down to approximately 9 m/px. Therefore higher resolution maps are needed to accurately render this part of the trajectory. To this purpose the Chang'e 2 global DEMs [23] have been used for this phase. This dataset, produced using stereo image data, is available at spatial resolution of 20 and 50 m/px, and is organized in multiple tiles, with a different projection method according to the region, in order to minimize image distortions. For all coasting sequences, the 20 m/px maps have been used.

Camera exposure time is optimized from the electron count of the first rendered frame, so that the sensor operates always below saturation. This exposure is then held fixed for the rest of the sequence. Alongside each image, a corresponding depth map is generated, which is later used as the per-pixel scene depth $Z(x, y)$ required by the motion field equations of Sec. 2. Once a sequence has been rendered, dense optical flow is computed following

the closed-form motion field expressions of Eq. 7, using the per-frame camera translational and angular velocity together with the corresponding depth map $Z(x, y)$ produced during rendering.

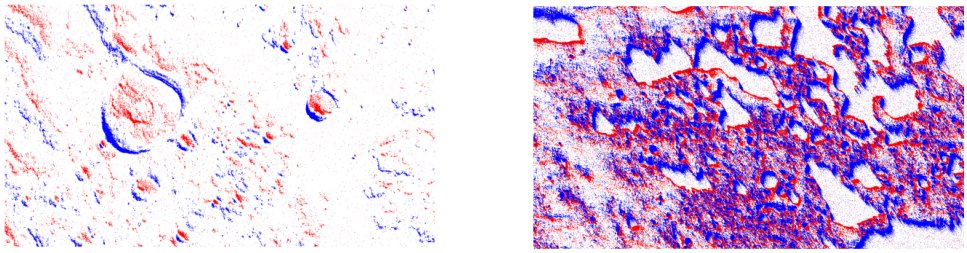
Event stream generation

Event streams are generated from the rendered image sequences using v2e [11], a Dynamic Vision Sensor (DVS) emulator designed to replicate the real response of an event-based camera. Among the available event simulators, v2e is selected because of its accurate modelling of sensor noise and rapid generation of events. No intermediate frame interpolation is applied before processing the sequences with v2e, therefore events are generated with a temporal resolution of 100 Hz. A key driver of event generation quality is scene illumination. Under high phase angles the lunar surface is dominated by strong shadowing and high contrast, producing event streams that are comparatively easy to trigger but prone to noise in low-signal shadow regions. Under low phase angles, illumination is more uniform and contrast is lower, so a more sensitive threshold configuration is needed to capture brightness changes without excessive suppression. This increased sensitivity, however, also makes the sensor more susceptible to noise. For this reason, event generation is performed using two DVS parameter configurations, chosen based on whether the sequence phase angle is higher or lower than 60°. Table 2 specifies the two set of parameters: the positive and negative contrast thresholds, along with their variation, the photoreceptor cut-off frequency, and the leak and shot noise rates. These, as well as the 60° threshold, were tuned empirically against qualitative event video inspection.

Parameter	$\|PA\| > 60°$	$\|PA\| < 60°$
Positive contrast threshold, C_+	0.25	0.15
Negative contrast threshold, C_-	0.25	0.15
Threshold mismatch, σ_C	0.03	0.01
Photoreceptor cutoff frequency [Hz]	10	3
Leak event rate [Hz]	0.1	0.05
Shot noise rate [Hz]	0.5	0.1

Table 2 *v2e simulation parameters. PA: Phase Angle.*

The photoreceptor cut-off frequency was selected jointly with the event thresholds to balance sensitivity to intensity changes against noise suppression. A lower photoreceptor cut-off frequency is paired with the lower thresholds to reduce sensitivity to high-frequency fluctuations, at the cost of a somewhat slower photoreceptor response. Conversely, the larger thresholds used at high phase angles already provide adequate fluctuations rejection, allowing a higher cut-off frequency to be used to preserve a fast response across the sharp, high-contrast transitions produced at shadow boundaries. The leak and shot noise rates are likewise increased for $|PA| > 60°$. This reflects the fact that DVS shot noise is concentrated in low-signal regions of the scene and grows as local illumination decreases [8], so a sequence dominated by deep shadow is expected to exhibit a higher baseline noise-event rate than a more uniformly and moderately illuminated one. Leak events as well were scaled to reflect the higher overall background activity typically observed in high-contrast, high-dynamic-range scenes. Figure 4 shows accumulated event frames from two representative coasting sequences, one below and one above the 60° phase-angle threshold, illustrating the resulting differences in event activity.



(a) **(b)**

Figure 4 *Example event frames accumulated over 0.1 s from coasting sequences with phase angles (a) below* 60° *(57,938 events) and (b) above* 60° *(981,983 events).*

4 Spiking Neural Network

4.1 Input event representation

The event stream generated by an event camera is defined as in Eq. 8, with each event represented by four parameters:

$$e_i = \{(x_i, y_i, t_i, p_i)\}_{i\in[1,N]} \tag{8}$$

where x_i, y_i are the pixel coordinates at which the $i-th$ event occurs, t_i represents its timestamp, and p_i denotes its polarity, which is positive if triggered by an increase in brightness, negative if triggered by a brightness decrease. To provide the network with a suitable input, the raw event stream must first be converted into a structured representation. In this work two approached have been considered: event histograms and voxel grid. The former approach consists in accumulating events over a short time window to create an *event histogram*, where each pixel stores the number of events that occurred at that location within the window, with every event contributing exactly +1 to its corresponding temporal bin. Positive and negative events are accumulated into separate channels, preserving polarity information. Multiple consecutive histograms are then stacked together to form an input sample to the network. The time window used to generate a single event histogram, equivalently expressed through the target accumulation frequency f_{target}, is a hyperparameter to tune. The number of consecutive histograms stacked together is determined by the temporal kernel size and the number of temporal encoding stages. The combination of these parameters is investigated in the design study presented in Sec. 5. The final input tensor has therefore dimension $2 \times T \times H \times W$, as shown in Fig 5, where T is the number of histograms stacked together and $H \times W$ is the camera resolution.

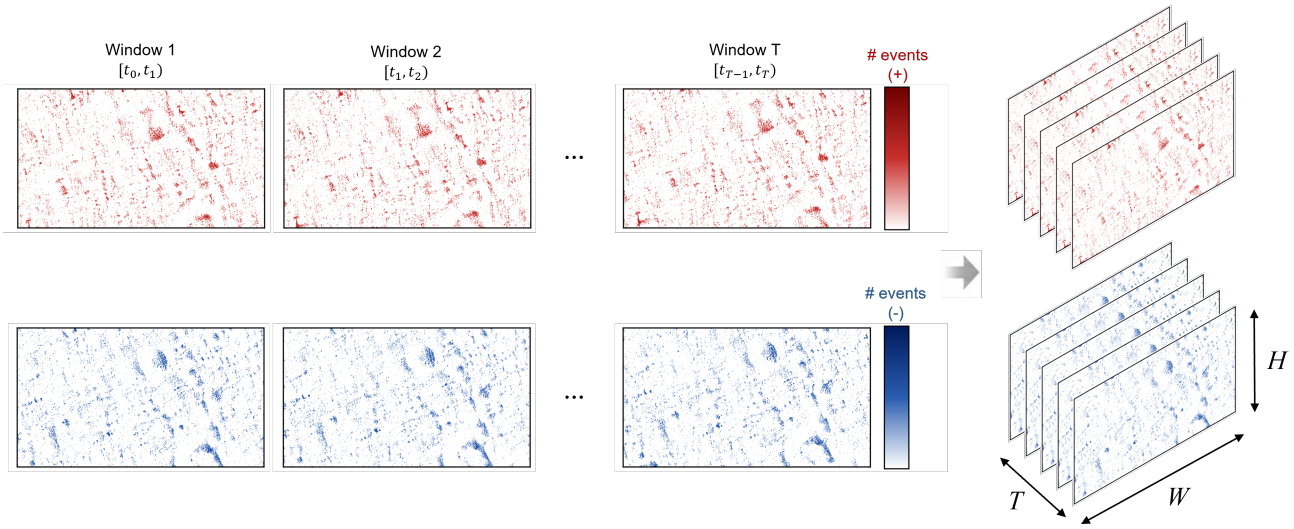


Figure 5 *Schematic illustration of event histogram representation. Positive events are shown in red (top row) and negative events in blue (bottom row).*

The *voxel grid* representation is instead obtained by discretizing the time window into B temporal bins. Each event contributes to its two temporally nearest bins, with its contribution split between them according to proximity, preserving finer temporal information. Eq. 9 shows the general formulation for the voxel construction at location (x, y), where events of both polarities are accumulated jointly and polarity is encoded as the sign of p_i [28].

$$V(x, y, t_k) = \sum_i p_i \max\left(0,\ 1 - \left|t_k - (B-1)\frac{t_i - t_0}{t_N - t_0}\right|\right), \qquad t_k = 0, 1, \ldots, B-1 \tag{9}$$

The resulting tensor has still shape $2 \times T \times H \times W$, with $T = B$. Since polarity is encoded by sign, the signed voxel grid is split into positive and negative parts, mirroring the polarity separation already used for the event histogram. This keeps the two representations interchangeable at the network's input interface, differing only in how each temporal bin's value is computed.

4.2 Network architecture

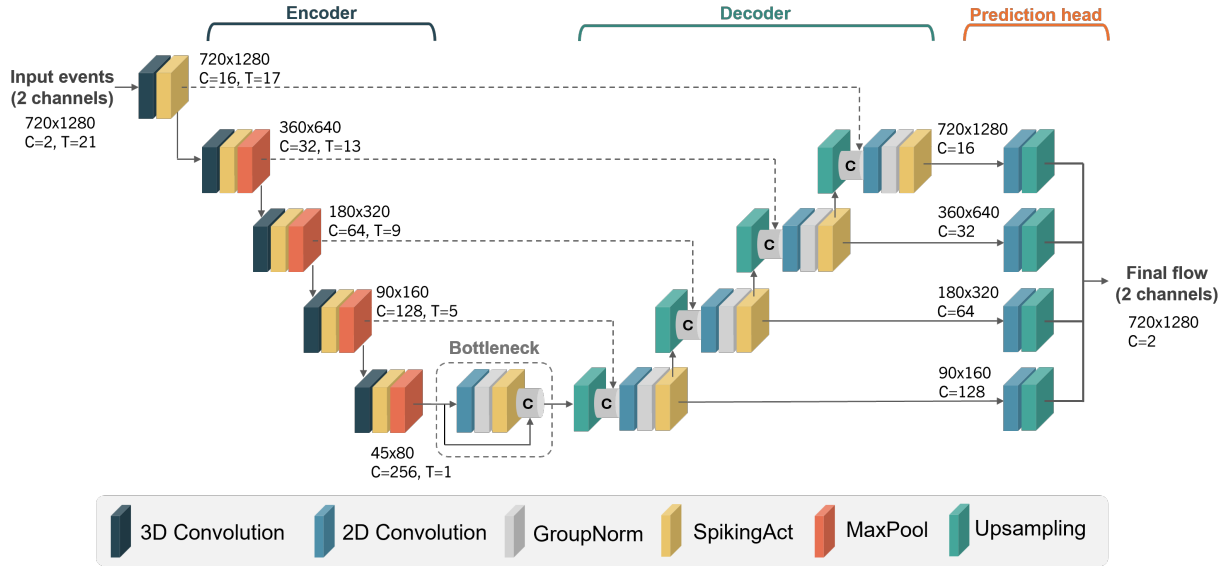


Figure 6 *UNet SNN proposed architecture.*

The proposed model is a UNet-like spiking encoder–decoder that maps the input event tensor of shape $2\times T\times H\times W$ directly to a dense two-channel flow field. The main components are presented below, comprising the spiking neuron model, encoder, bottleneck, decoder and prediction head.

- **Spiking Neuron Model.** All spiking nonlinearities use a stateless Integrate-and-Fire (IF) neuron model. Each neuron accumulates its input into a membrane potential and emits a binary spike when a fixed threshold is crossed, with the potential reset at the start of every forward pass. This avoids maintaining state between samples and is simpler and more hardware-friendly than its leaky counterpart [3]. Temporal context is handled explicitly by the $3D$ convolutions in the encoder, which aggregate information across the stacked input histograms or voxel bins before the spiking nonlinearity is applied. A benefit of this design is that training samples no longer need to be presented sequentially, since no state persists between forward passes.
- **Encoder.** Before the actual encoding stages, the input tensor is processed by a grouped $3D$ convolutional with kernel size $(k_t \times k_s \times k_s)$ and groups equal to 2. This independently expands the positive and negative polarity streams into feature channels, without spatial downsampling. Four encoder stages follow, each implemented again as a grouped $3D$ convolution that doubles the channel count, followed by the spiking activation layer and a $2D$ max-pooling applied independently at each time step. Throughout the encoder, the number of groups equals the number of input channels, enforcing not only fully decoupled polarity tracks, but also no inter-channel mixing. Preliminary experiments showed that this grouping performed better than using one group per polarity. The last temporal slice of each encoder stage is retained as the skip connection for the corresponding decoder stage.
- **Bottleneck and decoder.** After the fourth encoder stage, the temporal dimension is collapsed, yielding to a purely spatial representation. This is first through a bottleneck block and then in the decoder stages. The decoder mirrors the encoder with four purely $2D$ stages. Each stage upsamples its input by a factor of two using nearest-neighbour interpolation, concatenates the corresponding encoder skip connection, and applies a $2D$ convolution, GroupNorm, and spiking activation. GroupNorm is retained here and in the bottleneck because it was found necessary to keep the pre-spike membrane potential near the neuron's firing threshold, which otherwise starves the surrogate gradient during backpropagation.
- **Prediction head.** Each decoder stage produces a two-channel flow estimate. The three coarser predictions are upsampled to the finest resolution, and the final flow field is obtained by averaging the four predictions.

4.3 Training details

Trainings were performed on an NVIDIA RTX 6000 Ada Generation in a supervised manner, using the surrogate gradient approach [4], with the sigmoid function as surrogate gradient model, which performed better than the arctan function. All models were trained for 70 epochs using a constant learning rate of 10^{-3}.
The adopted loss function is composed of two terms: a *modular loss* and an *angular loss*, reported in Eq. 10.

$$
\begin{aligned}
L_{mod} &= \frac{1}{|N_v|} \sum_{i \in N_v} \frac{\sqrt{\left(p_{x_i} - g_{x_i}\right)^2 + \left(p_{y_i} - g_{y_i}\right)^2}}{||\mathbf{g}_i||} \\
L_{ang} &= \frac{1}{|N_v|} \sum_{i \in N_v} \left(1 - \frac{p_{x_i} g_{x_i} + p_{y_i} g_{y_i}}{||\mathbf{p}_i|| \cdot ||\mathbf{g}_i|| + \varepsilon}\right)
\end{aligned}
\tag{10}
$$

The modular loss measures the relative error between the norm of the predicted flow vector and the norm of the ground-truth flow vector, normalized by the ground-truth magnitude. The angular loss measures the cosine distance between the predicted and ground-truth flow directions, penalizing the network independently of the magnitude of either vector. The normalization in L_{mod} allows to express the error as a fraction of the true displacement rather than an absolute pixel offset, keeping L_{mod} in a numerically comparable range to L_{ang}: since the angular loss is a cosine distance and is by construction bounded in $[0, 2]$ irrespective of flow magnitude, an un-normalized endpoint error would scale with the raw magnitude of the flow field and could dominate the combined objective, making the relative weighting between the two terms harder to control. In the angular loss, the ground-truth flow magnitude in the denominator is regularized by a small constant $\varepsilon = 10^{-7}$ to avoid numerical instability. The two terms are then combined into a single training loss as in Eq. 11:

$$
L = w_{mod} \cdot L_{mod} + w_{ang} \cdot L_{ang} \tag{11}
$$

where w_{mod} and w_{ang} balance the relative contribution of the two terms. The weights were respectively set as $w_{mod} = 1$ and $w_{and} = 5$ as they yielded good results and comparable loss contribution. Both input representations described in Sec. 4.1 accumulate events over a fixed time window without any explicit filtering of the raw stream. As a consequence, isolated events caused by sensor noise are accumulated into the input tensor in the same way as events generated by real edges moving across the sensor. At locations where only such spurious events occur, the corresponding input signal carries little to no information about the true motion, yet the network would still be required to regress a valid optical flow value there if these pixels were included in the training loss. To prevent this, rather than filtering events out of the input representation itself, a per-pixel binary mask is constructed alongside each input sample and used to exclude pixels with sporadic events from the loss computation, so that the network is only supervised where the input event signal is trustworthy. Pixels for which fewer than two events were accumulated in either polarity channel over the input window are marked invalid. The resulting mask is applied at loss computation time, so that only valid pixels contribute to the gradient.

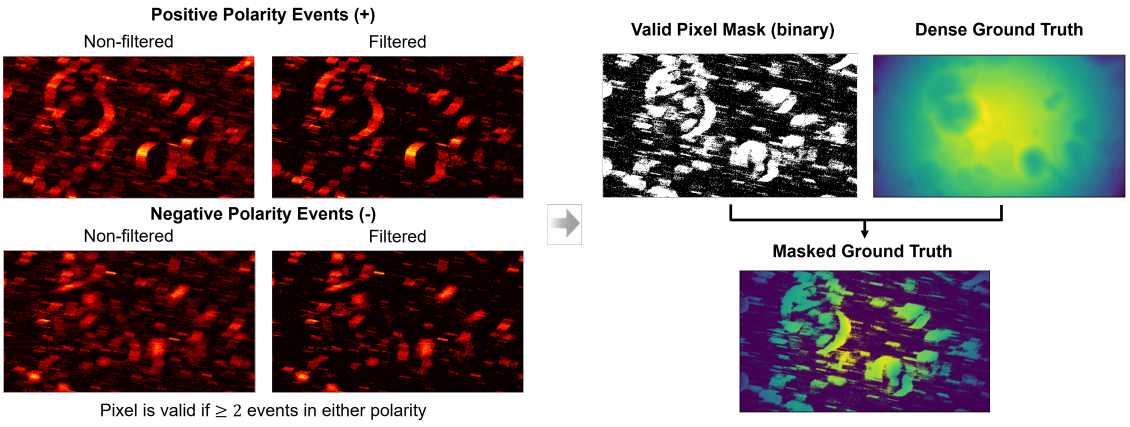


Figure 7 *Valid-pixel masking for event-based optical-flow ground truth for supervision*

5 Results

This section outlines the analyses and results conducted on the network. First the design study is presented, with the aim of chosing the best hyperparameters combination. Then the results split according to the mission regime are presented for the best architecture, with a discussion on model performance and limitations. The performance of the different trainings are evaluated based on the test set, which comprises 32 sequences, 23 LLO and 9 coasting sequences.

5.1 Design study

Firstly the spatial and temporal context of the network are optimized. Table 3 reports the results for the different combinations of spatial kernel k_s, temporal kernel k_t, and the event accumulation frequency f_{target}, defining the rate at which events are accumulated into a single event histogram or voxel bin. All trainings in this table use the event histogram representation detailed in Sec. 4.1. The metrics reported are the Average Endpoint Error (AEE), which is the mean Euclidean distance between the predicted and ground-truth flow vectors, and the Average Angular Error (AAE), that is the mean angle between the two vectors. All metrics are evaluated on the set of pixels corresponding to the masked ground truth detailed in Sec. 4.3. Reducing the spatial kernel from 7×7 to 5×5 gives the single largest improvement in the study: at fixed $k_t = 5$ and $f_{\text{target}} = 10$ Hz, it lowers both mean errors, while nearly halving the parameter count. For $k_s = 7$ and $k_t = 5$, increasing the temporal context at 10 Hz also provides a consistent gain. Increasing the accumulation frequency from 10 to 20 Hz further reduces the errors without increasing the parameter count. The shorter accumulation windows likely produce sharper, more localized edges, yielding to better performance both in magnitude and direction estimation. However, this configuration still has nearly twice the number of parameters of the $k_s = k_t = 5$ configuration and slightly higher mean errors. The configuration with $k_s = 5$, $k_t = 5$, $f_{\text{target}} = 10$ Hz is overall the best configuration and is adopted as the reference one for the remainder of the design study.

Spatial Kernel	Temporal Kernel	Accumulation frequency	Parameters	Mean		Median	
k_s	k_t	f_{target} $[Hz]$	n_p $[\times10^6]$	AEE [px/s]	AAE [°]	AEE [px/s]	AAE [°]
7	7	10	14,046,232	3.84	3.70	1.42	2.24
7	5	10	13,997,624	4.43	5.40	1.94	2.51
7	5	20	13,997,624	3.44	3.35	1.58	1.98
5	7	10	7,167,640	3.77	3.19	1.97	2.65
5	5	10	**7,142,840**	**3.04**	**2.97**	1.67	**1.95**
5	5	20	7,142,840	3.95	4.99	1.88	2.59

Table 3 *Design study on spatial and temporal context.*

A systematic study is then conducted by varying one design choice at a time from the above reference configuration. The results are reported in Table 4. Joint horizontal-flip and polarity-swap augmentation is the only variant that improves on the reference, lowering the mean and median AEE and the mean AAE at unchanged parameter count, with only a marginal increase in median AAE. In particular, the results show that these augmentations are especially beneficial for the more challenging coasting sequences. Replacing the event histogram with the voxel grid of Eq. (9) slightly degrades both metrics. Its finer temporal information, brings no benefit, likely because the flow is nearly uniform over the accumulation window and the histograms already captures the relevant information. The reference configuration uses 16 channels after the first convolution before the encoder (*base channels*). Doubling the width degrades every metric, indicating that the network is not capacity-limited at the reference width, while halving it increases the mean AEE. Finally, training with the modular loss alone degrades the mean AAE as expected but also the mean AEE, which is less intuitive. The angular loss is therefore not just useful to improve direction at the expense of magnitude, a behaviour consistent with what is reported in [3]. The

combined loss is thus retained, together with the augmentation, which is the only modification carried over into the final model.

Design choice	Variation	Parameters n_p	Mean AEE [px/s]	Mean AAE [°]	Median AEE [px/s]	Median AAE [°]
Reference configuration		*7,142,840*	*3.04*	*2.97*	*1.67*	*1.95*
Augmentation	Horizontal flip and polarity swap	7,142,840	**2.69**	**2.78**	**1.55**	**2.04**
Input representation	Voxel grid	7,142,840	3.21	3.06	1.65	2.19
Network width	32 base channels	28,442,472	3.15	3.66	1.81	2.51
	8 base channels	1,801,824	4.73	3.70	1.91	2.26
Loss	Modular loss only	7,142,840	3.42	4.33	1.81	2.60

Table 4 *Comparison of individual design changes against the reference configuration*

5.2 Final model performance

Table 5 reports the accuracy of the proposed network, separated by mission phase. Because the flow magnitude differs substantially between the two regimes a relative endpoint error is also reported. For each regime it is computed by first evaluating per sequence the mean endpoint error over divided by the mean ground-truth flow magnitude, and then averaging all per sequence values, so that every sequence contributes equally.

Regime	n_{seq}	rel-EPE [%]	*AEE* [px/s]				*AAE* [deg]			
			mean	med.	p95	σ	mean	med	p95	σ
LLO	23	6.4	1.66	1.50	2.86	0.60	2.43	2.02	4.92	1.12
Coasting	9	13.0	5.27	4.01	13.27	4.29	3.52	2.25	9.61	3.38

Table 5 *Optical-flow accuracy of the final model divided by mission phase. All metrics are computed at valid event pixels.*

On the LLO sequences the network is both accurate and consistent: the mean and median AEE nearly coincide, the spread across the 23 sequences is small, with a standard deviation of 0.60 px/s. The coasting sequences present instead higher errors, in particular in magnitude. The mean AEE is roughly three times the LLO value and the relative error doubles. The angular accuracy degrades only mildly and the median angular error is essentially unchanged. Therefore the direction of the flow field is still recovered reliably. The dispersion, however, grows for both magnitude and direction. To analyse the coasting performance in more detail, and since these sequences also span a wide range of altitudes, Table 6 reports the errors of all nine of them individually, together with their initial altitude, mean ground-truth flow magnitude and spatial coverage. The latter is defined as the fraction of the sensor array entering the evaluation mask, defined in Sec 4.3. Since all reported metrics are averaged over exactly these pixels, coverage quantifies how much of the image plane actually contributes to the error of a given sequence. The results on the coasting regime show that the model maintains good performance across most of the evaluated sequences, despite the substantially different flow characteristics from the LLO regime. Most sequences exhibit moderate endpoint and angular errors, with several achieving performance comparable to that observed for LLO. The largest errors are concentrated in the two lowest-altitude sequences, which also correspond to the largest ground-truth flow magnitudes in the set. This suggests that the increased aggregate error observed for the coasting regime in Table 5 is driven primarily by these particularly challenging high-flow cases, rather than by a general loss of performance across the coasting trajectories. Sequence *moon C78* attains an angular error comparable to the best sequences while presenting the largest endpoint error of the set, whereas *moon C88* is the only case with a substantially degraded direction estimate. This indicates that the model can retain a reliable estimate of flow direction even when its magnitude estimate becomes inaccurate. Spatial coverage is also reported, to assess whether it is related to the observed estimation error. Sparsest sequences show larger angular errors,

Sequence	Alt. [km]	GT flow [px/s]	Coverage [%]	AEE [px/s]	rel-EPE [%]	AAE [°]
moon_C88	54.2	47.0	56.4	11.46	24.4	12.41
moon_C78	54.7	48.8	1.8	14.48	29.7	2.82
moon_C46	57.8	45.9	42.2	4.35	9.5	1.18
moon_C17	70.6	36.7	63.8	4.01	10.9	1.24
moon_C43	74.6	36.5	75.0	1.89	5.2	1.39
moon_C98	81.1	32.0	0.5	4.53	14.2	5.40
moon_C83	83.3	32.7	27.8	2.97	9.1	1.72
moon_C24	94.7	27.3	0.8	2.23	8.2	3.25
moon_C33	98.6	25.3	92.3	1.55	6.1	2.25

Table 6 *Best model performance on the coasting trajectories.*

likely because a sparsely populated image plane offers fewer independent features to aggregate and leaves the estimate more sensitive to local noise. However, this relationship is not universal, as demonstrated by *moon C88*, which has relatively high coverage but still presents a large angular error. Coverage should therefore be interpreted as one of several factors influencing performance. Most notably, when the two lowest-altitude and highest-flow sequences are excluded, the mean AEE decreases from 5.27 to 3.08 px/s, while the mean AAE decreases from 3.52° to 2.35°. The remaining coasting sequences therefore achieve performance closer to that observed in LLO, demonstrating that the model generalises well to the substantially different motion geometry encountered during coasting over a broad range of altitudes. The results consequently indicate that the main limitation of the current model is not the change in trajectory geometry itself, but its ability to accurately estimate the magnitude of the fastest flow fields.

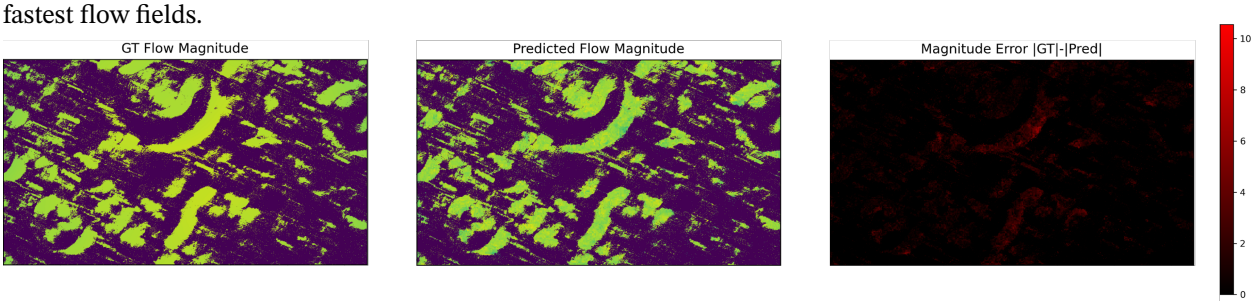


Figure 8 *Ground-truth flow magnitude, predicted flow magnitude and their absolute difference for a representative test sequence. Pixels outside the validity mask are set to zero.*

Fig. 8 compares the predicted and ground-truth flow magnitude for a representative test sequence, together with the map of their absolute difference. In all sequences, the ground truth flow magnitude is not perfectly uniform over the frame but exhibits a mild radial distribution, decreasing from the image centre towards the borders. This is a direct consequence of the observation geometry and is the dominant source of spatial structure in the ground truth, but it is also a weak one: the flow magnitude varies by about 2% in RMS about its own frame mean. Despite the accuracy reported in Table 5, the network struggles to reproduce this radial variation. This behaviour is best read as the network finding the correct optimum of the objective it was given rather than as a failure to converge, and the reason is the weakness of the radial signal relative to everything else in the loss. The radial deviation is a small perturbation on a large constant, so an error in estimating the frame mean flow contributes far more to L_{mod} than an error in the spatial structure carried on top of it. This also represents a limitation of the present study, as the considered sequences predominantly exhibit spatially uniform flow. As future work, the training and evaluation sets should include a wider range of camera orientations and observation geometries. Tilted camera configurations would produce more pronounced variations in flow direction and magnitude, providing a stronger basis for assessing whether the network can recover spatial flow structure beyond the dominant average magnitude.

6 Conclusions

This work presented an exploratory study on event-based perception for lunar navigation, addressing two gaps: the limited availability of realistic event-based lunar datasets with reliable motion ground truth, and the lack of SNN architectures tailored to this scenario. The dataset covers LLO and coasting sequences scattered over the lunar surface under randomized illumination, and provides rendered images, depth maps, event streams and analytically derived dense optical flow. The proposed UNet-like SNN regresses dense flow directly from the event stream, achieving a mean AEE of 1.66 px/s (6.4% relative error) in LLO and 5.27 px/s (13.0%) during coasting, where the residual error is concentrated almost entirely in the lowest-altitude sequences with the fastest flow fields, while the remainder of the coasting set performs close to the LLO regime. Flow direction is recovered reliably in both regimes, indicating that a single, comparatively compact spiking model generalises across different motion geometries. Future work will extend the dataset to lower altitudes and additional regions, as the current dataset stops at approximately 35 km altitude due to the resolution of the adopted maps. Magnitude estimation for fastest flow fields is also an aspect to address next. Finally, richer observation geometries with tilted camera configurations will be introduced to strengthen the spatial component of the learning problem, and integrate the network into a full egomotion estimation pipeline.

References

[1] L. J. Azzalini, E. Blazquez, A. Hadjiivanov, G. Meoni, and D. Izzo. On the generation of a synthetic event-based vision dataset for navigation and landing, 2023. `https://arxiv.org/abs/2308.00394`.

[2] M. K. Barker, E. Mazarico, G. A. Neumann, M. T. Zuber, J. Haruyama, and D. E. Smith. A new lunar digital elevation model from the Lunar Orbiter Laser Altimeter and SELENE Terrain Camera. *Icarus*, 273:346–355, July 2016. https://doi.org/10.1016/j.icarus.2015.07.039.

[3] J. Cuadrado, U. Rançon, B. R. Cottereau, F. Barranco, and T. Masquelier. Optical flow estimation from event-based cameras and spiking neural networks. *Frontiers in Neuroscience*, 17, May 2023. https://doi.org/10.3389/fnins.2023.1160034.

[4] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu. Training Spiking Neural Networks Using Lessons From Deep Learning, Aug. 2023. arXiv:2109.12894.

[5] P. Fanti, L. B. S. Williams, O. Dvořák, M. Märtens, T.-J. Chin, H. Ji, B. Chen, D. Xie, K. Qiao, B. Li, N. Einecke, S. Arumugam, A. R. Padhy, S. K. Rath, S. S. Mahapatra, and D. Izzo. Event-based lunar optical flow egomotion estimation challenge: design and results of the ELOPE competition. *npj Space Exploration*, 2(1):18, Apr. 2026. https://doi.org/10.1038/s44453-026-00033-0.

[6] G. Gallego, T. Delbruck, 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*, 44(1):154–180, Jan. 2022. https://doi.org/10.1109/TPAMI.2020.3008413. arXiv:1904.08405.

[7] R. G. Gottlieb. Fast gravity, gravity partials, normalized gravity, gravity gradient torque and magnetic field: derivation, code and data. Technical report, NASA CR 188243, 1993.

[8] R. Graca and T. Delbruck. Unraveling the paradox of intensity-dependent DVS pixel noise, Sept. 2021. arXiv:2109.08640.

[9] R. Guamán-Rivera, J. Delpiano, and R. Verschae. Event-based optical flow: Method categorisation and review of techniques that leverage deep learning. *Neurocomputing*, 635:129899, June 2025. https://doi.org/10.1016/j.neucom.2025.129899.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2nd edition, 2003. ISBN 978-0521540513.

[11] Y. Hu, S.-C. Liu, and T. Delbruck. v2e: From Video Frames to Realistic DVS Events. 2021.

[12] M. Jawaid, M. Märtens, and T.-J. Chin. Event-RGB fusion for spacecraft pose estimation under harsh lighting. *Aerospace Science and Technology*, 168:111039, Jan. 2026. https://doi.org/10.1016/j.ast.2025.111039.

[13] J. Kuffner. *Effective sampling and distance metrics for 3D rigid body path planning*, volume 2004. Apr. 2004. ISBN 978-0-7803-8232-9. https://doi.org/10.1109/ROBOT.2004.1308895.

[14] C. Lee, A. K. 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 *Computer Vision – ECCV 2020*, pages 366–382, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-58526-6_22.

[15] Z. Liu, B. Guan, Y. Shang, Y. Bian, P. Sun, and Q. Yu. Stereo Event-Based, 6-DOF Pose Tracking for Uncooperative Spacecraft. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–13, 2025. https://doi.org/10.1109/TGRS.2025.3530915.

[16] F. Mahlknecht, D. Gehrig, J. Nash, F. M. Rockenbauer, B. Morrell, J. Delaune, and D. Scaramuzza. Exploring Event Camera-based Odometry for Planetary Robots, July 2022. arXiv:2204.05880.

[17] Massachusetts Institute of Technology. Sldem2015 tiles. `https://imbrium.mit.edu/BROWSE/SLDEM2015/TILES/`, n.d. MIT Imbrium repository. Accessed: 2026-07-28.

[18] NASA. LP165P1, 2000. `https://pds-geosciences.wustl.edu/lunar/lp-l-rss-5-gravity-v1/lp_1001/sha/jgl165p1.lbl`.

[19] A. Pizzetti, P. Panicucci, F. Capolupo, and F. Topputo. Development and validation of a physically based rendering methodology for celestial bodies. *Acta Astronautica*, 245:430–460, 2026. https://doi.org/https://doi.org/10.1016/j.actaastro.2026.03.003.

[20] Prophesee. Event-based sensor imx636 (sony–prophesee), 2024. `https://www.prophesee.ai/event-based-sensor-imx636-sony-prophesee/`.

[21] J. Puritz, G. Molina-Ramos, L. Hong, J. Pelgrift, C. Adam, and T. Crain. Intuitive machines optical camera calibration and tuning. 2026.

[22] A. W. Reed, C. Hashemi, D. Melamed, N. Menon, K. Hirakawa, and S. McCloskey. EBS-EKF: Accurate and High Frequency Event-based Star Tracking, Mar. 2025. arXiv:2503.20101 [cs].

[23] F. Scholten, J. Oberst, K. Matz, T. Roatsch, M. Wählisch, E. J. Speyerer, and M. S. Robinson. GLD100: The near-global lunar 100 m raster DTM from LROC WAC stereo image data. *Journal of Geophysical Research: Planets*, 117(E12):2011JE003926, Dec. 2012. https://doi.org/10.1029/2011JE003926.

[24] O. Sikorski, D. Izzo, and G. Meoni. Event-based spacecraft landing using time-to-contact. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1941–1950, June 2021. https://doi.org/10.1109/CVPRW53098.2021.00222.

[25] H. Sun, J. Wang, W. Cai, D. Chen, Q. Liao, J. He, Y. Cui, D. Yao, and D. Guo. ST-FlowNet: An efficient Spiking Neural Network for event-based optical flow estimation. *Neural Networks*, 190:107730, Oct. 2025. https://doi.org/10.1016/j.neunet.2025.107730.

[26] Wang Fen-Fei, Liu Jian-Jun, Li Chun-Lai, Ren Xin, Mu Ling-Li, Yan Wei, and Zhang Xiao-Xia. A new lunar absolute control point: established by images from the landing camera on Chang'e-3. *Research in Astronomy and Astrophysics*, 14(12):1543, Dec. 2014. https://doi.org/10.1088/1674-4527/14/12/004.

[27] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras. In *Robotics: Science and Systems XIV*, June 2018. https://doi.org/10.15607/RSS.2018.XIV.062. arXiv:1802.06898 [cs].

[28] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Unsupervised Event-based Learning of Optical Flow, Depth, and Egomotion, Dec. 2018. arXiv:1812.08156.