ON THE GENERATION OF SYNTHETIC EVENT-BASED VISION DATASETS FOR NAVIGATION AND LANDING

Loïc J. Azzalini[(1)], **Emmanuel Blazquez**[(2)], **Gabriele Meoni**[(3)], **Dario Izzo**[(4)], **Alexander Hadjiivanov**[(5)]

[(1)]*Advanced Concepts Team, European Space Agency, European Space Research and Technology Centre (ESTEC), Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands, jazzalin@outlook.com*
[(2)]*Advanced Concepts Team, European Space Agency, European Space Research and Technology Centre (ESTEC), Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands*
[(3)]*Distributed Space Systems, Faculty of Aerospace Engineering, TU Delft, Mekelweg 5, 2628 CD Delft, The Netherlands*
[(4)]*Advanced Concepts Team, European Space Agency, European Space Research and Technology Centre (ESTEC), Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands, dario.izzo@esa.int*
[(5)]*Advanced Concepts Team, European Space Agency, European Space Research and Technology Centre (ESTEC), Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands, alexander.hadjiivanov@esa.int*

ABSTRACT

This paper presents a methodology and a software pipeline for generating event-based vision datasets from optimal landing trajectories. The study is motivated by the potential use case for an onboard event-based navigation camera during the approach of a target body and the lack of real-world datasets to support it due to the fact that event-based cameras are relatively new technology. We construct sequences of photorealistic images of the lunar surface with the Planet and Asteroid Natural Scene Generator (PANGU) at various points along optimal descent trajectories with varying boundary conditions. The generated image sequences are then converted into event streams by means of an event-based camera emulator (`v2e`). We demonstrate that the pipeline can generate realistic event-based representations of surface features by constructing a dataset of 500 trajectories, complete with event streams and motion field groundtruth data. We anticipate that novel event-based vision datasets can be generated using this pipeline to support various spacecraft pose reconstruction problems given events as input, and we hope that the proposed methodology would attract the attention of researchers working at the intersection of neuromorphic vision and guidance navigation and control.

Keywords — Neuromorphic vision, event-based sensing, optical flow navigation, motion estimation

1 INTRODUCTION

Neuromorphic vision technology holds great promise for space applications owing to its submillisecond temporal resolution, low power consumption and high dynamic range [1][2]. Dynamic vision sensors, or event-based cameras as they are commonly known, have disrupted research fields across the computer vision landscape, from object tracking to image reconstruction, in robotics to the automotive industry [3]. A dynamic vision sensor is event-driven by design, as independently sensitive image pixels only respond to changes in scene brightness, leading to sparse and asynchronous output streams of data. Only motion between the scene and the camera will generate data, thus avoiding the capture of redundant static information and wasteful usage of onboard resources. Recent demonstrations of dynamic vision for in-orbit space situational awareness (SSA) [4][5] are encouraging, warranting further adoption of event-based hardware in the space sector.

In recent years, there has been a surge in interest in event-based cameras for optimal control, landing and navigation. The most common method for translating events into control relies on estimating optical flow or divergence from events – a bioinspired method used by bees and other insects for smooth landing on surfaces. Most studies on event-based landing consider the ventral landing case, without a lateral motion component. For instance, the use of optical flow reconstruction from events for optimal control of a micro air vehicle (MAV) during ventral landing was investigated in [6]. The method relies on maintaining constant divergence (the ratio of the vertical velocity component to the height) to perform a smooth landing. A time-to-contact (TTC) metric derived from divergence estimated from event streams was used for optimal ventral landing in [7]. Similarly, ventral landing scenarios based on a contrast maximisation technique for estimating divergence in a stream of events were studied in [8]. Recently, optical flow has been demonstrated for dynamic attitude, paving the way for navigation and control for small (insect-scale) robots without using accelerometers, as well as using micro-movements (e.g., vibrations) for attitude control and landing for larger vehicles [9].

Whether tracking space debris or landing on a planetary body or asteroid, events are captured in the image plane of the dynamic vision sensor as a result of the relative dynamics between the onboard camera and the space environment. While event-based vision for guidance navigation and control (GNC) has been considered at a theoretical level in the past [2], onboard use of this technology has yet to be demonstrated. Its limited adoption to date can, in part, be explained by the lack of event-based datasets tailored to vision-based navigation in space and state-of-the-art event processing tools found in other disciplines (e.g., robotics). Our main contribution is a software pipeline for generating event-based datasets from simulated spacecraft trajectories in a photorealistic scene generator for planetary bodies and asteroids. This work considers various landing scenarios (beyond the ventral case) in order to illustrate how the pipeline may be configured to capture event-based representation of surface features from various spacecraft poses.

More specifically, we propose a flexible data pipeline which takes in trajectory specifications as input and outputs streams of events corresponding to the motion of features in the scene. From the initial and final conditions of the spacecraft trajectory and the properties of the target body, we solve an optimal control problem corresponding to a non-ventral, minimum-mass descent trajectory on a lunar-analogue surface. The trajectories are then used to manipulate the viewpoint of a pinhole camera model in the Planet and Asteroid Natural Scene Generator (PANGU)[10], which renders synthetic images of the surface during approach. Finally, a video-to-event converter (`v2e`) [11] is used to generate synthetic events induced by the simulated landings. The sparse and asynchronous events include various sources of noise modelled after the performance of dynamic vision sensors in the challenging lighting conditions in the field. The resulting dataset captures dynamic, event-based representations of common surface features such as craters and the Moon's horizon. Figure 1 provides an overview of the pipeline, from inputs to outputs.

As the main contribution of this work, the pipeline is designed to be modular in order to support various spacecraft pose reconstruction problems given events as input. While the reconstruction of the dynamics from events (represented by a dashed arrow in Figure 1) is not covered in this work, groundtruth data is recorded in anticipation of such inverse problems. Aside from the optimal landing trajectories, the pipeline also saves motion field data in a format that is compatible with optical flow methods [12] which have been successfully applied to event-based data in the past [13], [14].

The following section introduces the dynamics of the lunar lander model (Section 2.1) and the minimum-fuel control problem (Section 2.2) solved to simulate the landing trajectories. The ide-

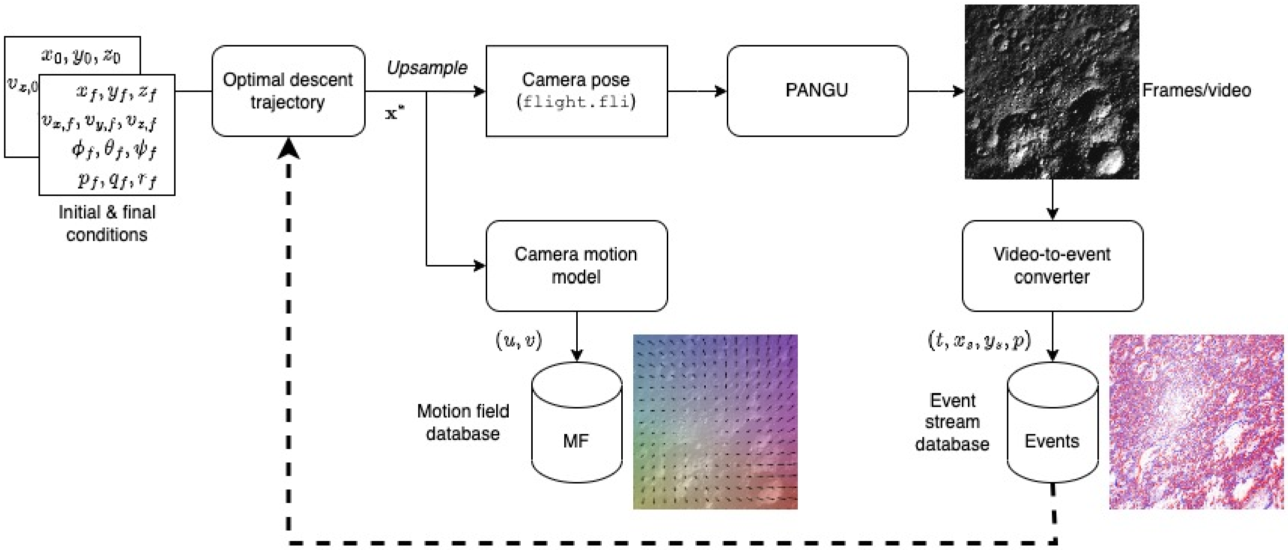

Figure 1: Pipeline to generate event-based vision datasets: input trajectory specifications are processed into groundtruth motion fields, video simulations of the landing sequence and streams of asynchronous events

alised camera model and its equations of motion are detailed in Section 2.3, leading to the derivation of motion field equations for planar and spherical landing surfaces. Section 2.4 introduces the alternative readout of the event-camera and how it can be emulated in software. As this work focuses on the generation of event streams to support navigation and landing, the processing of event-based data will not be discussed.
A dataset of 500 trajectories that share characteristics with typical braking, approach and descent landing profiles is generated. Highlights are presented in Section 3 to demonstrate the data generation capabilities of the pipeline, as hinted at in Figure 1. Specifically, the PANGU scenes of a dimly illuminated lunar surface are compared to the corresponding motion fields and the synthetic event-based representations. The accumulation of events into frames reveal how the generation of events varies with various trajectory profiles and under different lighting conditions at the landing site. Finally, the limitations of the pipeline, particularly in terms of dataset variability, are discussed in Section 4 alongside potential extensions for follow-up work.

2 METHOD

We consider the problem of a lunar lander entering the final stages of a descent towards the surface. In this work, it is assumed that an event-based descent camera is mounted underneath the lunar module and pointed in the direction of motion, offering uninterrupted streams of data to the onboard computer to support vision-based navigation. The geometry of the problem is illustrated in Figure 2 with a simplified trajectory broken down into a braking, approach and (terminal) descent phase, and an idealised camera model capturing surface features in its image plane.

2.1 Equations of motion

The inertial frame of reference $\mathcal{F}_i$ is placed on the lunar surface at the location of the landing site O_i. The attitude of the spacecraft with body frame $\mathcal{F}_{sc}$ relative to the inertial frame is characterised by the roll-pitch-yaw sequence of Euler angles $(\theta_1, \theta_2, \theta_3) = (\phi, \theta, \psi)$. The camera frame $\mathcal{F}_c$ and body frame $\mathcal{F}_{sc}$ are made to be coincident, such that the camera centre O_c (centre of projection) is identical

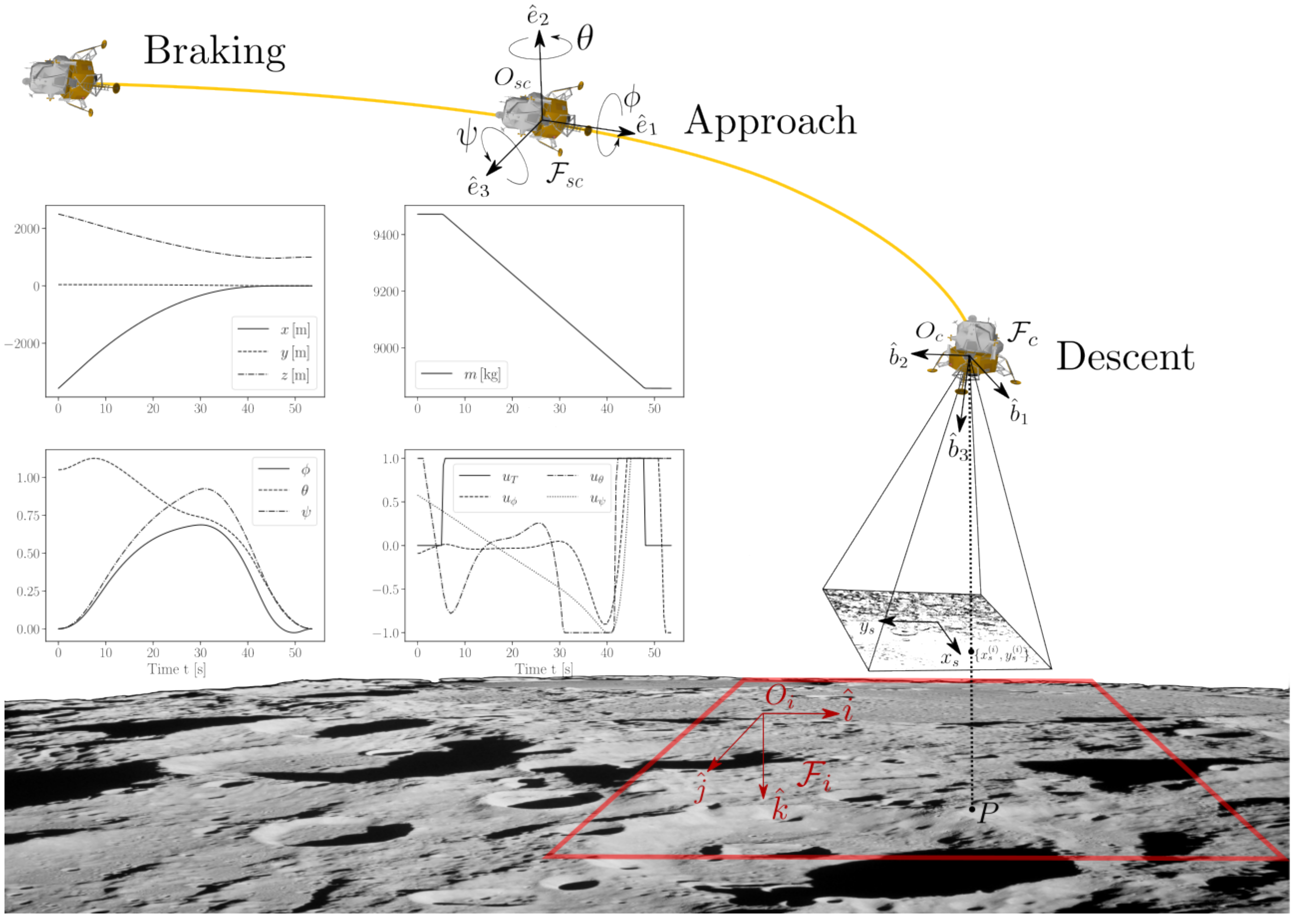

Figure 2: Geometry of the landing scenario and sample minimum-mass trajectory

to O_{sc}. The position, velocity and angular velocity of the spacecraft are respectively defined as

$$\mathbf{r}_{sc} = \mathcal{F}_i^T \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \mathbf{v}_{sc} = \mathcal{F}_i^T \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \qquad \boldsymbol{\omega} = \mathcal{F}_{sc}^T \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{1}$$

and the transformation from $\mathcal{F}_{sc}$ to $\mathcal{F}_i$, i.e. $\mathbf{R} = \mathcal{F}_i \cdot \mathcal{F}_{sc}^T$, is described by the direction cosine matrix

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2}$$

The normalised throttle $u_T \in [0, 1]$ regulates the spacecraft main engine, while the normalised throttles $u_\phi, u_\theta, u_\psi \in [0, 1]$ regulate the attitude control nozzles. These are placed symmetrically at a distance L from the spacecraft centre of mass to generate roll, pitch and yaw moments. Then, we consider the following 12-DOF lander dynamics:

$$\dot{\mathbf{r}} = \mathbf{v} \tag{3}$$

$$\dot{\mathbf{v}} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \frac{1}{m}\mathbf{R}\begin{bmatrix} 0 \\ 0 \\ \bar{F}_a u_T \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5}$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + 2L\bar{F}_b\begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} \tag{6}$$

and the mass-thrust relationship

$$\dot{m} = -\frac{\overline{F}_a u_T + 2\overline{F}_b(u_\phi + u_\theta + u_\psi)}{I_{sp}g_0} \tag{7}$$

where explicit time dependence notation has been dropped to improve readability. The gravitational accelerations on the landing body and Earth are respectively given by g and g_0, I_{sp} is the specific impulse, and $\bar{F}_a$ and $\bar{F}_b$ denote the maximum thrust and maximum secondary thrust for the attitude control. We let $\mathbf{x}(t) = \begin{bmatrix}\mathbf{r}(t)^T & \mathbf{v}(t)^T & \phi(t) & \theta(t) & \psi(t) & \boldsymbol{\omega}(t)^T & m(t)\end{bmatrix}^T$ and $\mathbf{u}(t) = \begin{bmatrix}u_T(t) & u_\phi(t) & u_\theta(t) & u_\psi(t)\end{bmatrix}^T$ be the states and inputs vectors such that the dynamics may be concisely written as $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$.

2.2 Optimal control problem

The dataset trajectory generation is based on a minimum-fuel optimal control problem (OCP). The objective is to minimise the total commanded thrust $\mathbf{u}(t)$ (thereby maximising the rate of change of the mass (7)) and the total time t_f required to bring the spacecraft from a starting pose to a final pose, subject to vehicle dynamics and thrust limits. Formally, this may be written as

$$\begin{array}{rll} \displaystyle\min_{t_f, \mathbf{u}(t)} & \displaystyle\int_0^{t_f} \mathbf{u}(\tau)^T\mathbf{u}(\tau)\,d\tau & \\ \text{subject to} & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), & \text{Dynamic constraints} \\ & \cos\phi \cdot \cos\theta \geq \cos\theta_{max}, & \text{Max thrust tilt} \\ & \mathbf{0} \leq \mathbf{u}(t) \leq \mathbf{1}, & \text{Thrust limits} \\ & \mathbf{x}(0) = \mathbf{x}_0,\ \mathbf{x}(t_f) = \mathbf{x}_f & \text{Boundary conditions} \end{array} \tag{8}$$

where θ_{max} denotes the maximum allowable tilt angle. The initial states $\mathbf{x}_0$ and final states $\mathbf{x}_f$ are to be specified in a manner that approximates the desired braking, approach or descent trajectory characteristics in a free or pinpoint landing. Figure 2 illustrates the solution of an OCP which has been parameterised to simulate an approach phase, i.e. the high starting pitch is regulated in preparation of the terminal descent.

2.3 Camera model and motion field

Before considering the event-based representation of visual features, we review the relationship between a 3D point in the scene and its projection in the image plane. As illustrated in Figure 2, we consider the generic point P on the planetary surface and let

$$\overrightarrow{\boldsymbol{O_c P}} = \mathcal{F}_c^T \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{9}$$

As the spacecraft moves relative to the surface, the apparent velocity of the point P in the camera frame $\mathcal{F}_c$ is opposite to its velocity in $\mathcal{F}_i$:

$$\mathcal{F}_c^T \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = -\mathbf{v}_{O_c} - \boldsymbol{\omega} \times \overrightarrow{\boldsymbol{O_c P}} \tag{10}$$

where $\mathbf{v}_{O_c} = \mathbf{R}^T \mathbf{v}$ is the inertial velocity of O_c in the camera frame since $\mathcal{F}_c$ and $\mathcal{F}_{sc}$ are coincident. Expanding (10) yields

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = - \begin{bmatrix} v_{xc} \\ v_{yc} \\ v_{zc} \end{bmatrix} - \begin{bmatrix} qZ - rY \\ rX - pZ \\ pY - qX \end{bmatrix} \tag{11}$$

Then, given an idealised pinhole camera with focal length f, the projection of P in the sensor array corresponds to

$$x_s = f\frac{X}{Z}, \qquad y_s = f\frac{Y}{Z} \tag{12}$$

By differentiating (12) and substituting in (11), we obtain

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -f & 0 & x_s \\ 0 & -f & y_s \end{bmatrix} \begin{bmatrix} v_{xc} \\ v_{yc} \\ v_{zc} \end{bmatrix} + \begin{bmatrix} \frac{x_s y_s}{f} & -(f + \frac{x_s^2}{f}) & y_s \\ f + \frac{y_s^2}{f} & -\frac{x_s y_s}{f} & -x_s \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{13}$$

where u and v describe the motion field induced in the image plane as a result of the dynamics of the spacecraft relative to the scene. Provided with sufficient information about the target body shape and/or landing region, we can substitute the inverse depth map $h(x_s, y_s) = \frac{1}{Z}$ for an expression that only depends on the spacecraft state. Planar and spherical geometries are considered below.

2.3.1 Planar landing site

If the landing site is assumed to be a perfect plane, the point on the surface P relates to the spacecraft altitude as follows:

$$\begin{aligned} \hat{\mathbf{k}} \cdot \overrightarrow{\boldsymbol{O_c P}} &= \alpha X + \beta Y + \gamma Z = -z = H \\ Z &= \frac{H}{\alpha x_s + \beta y_s + \gamma} = \frac{H}{A} \end{aligned} \tag{14}$$

where $\hat{\mathbf{k}} = \mathcal{F}_c^T \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T$ and the altitude H is equivalent to the negative height from the surface (see the problem geometry in Figure 2). Given that $\mathcal{F}_c$ and $\mathcal{F}_{sc}$ are taken to be coincident in this study, we have from (2)

$$\alpha = -\sin\theta, \qquad \beta = \sin\phi\cos\theta, \qquad \gamma = \cos\phi\cos\theta \tag{15}$$

and hence, letting $A = -x_s \sin\theta + y_s \sin\phi\cos\theta + \cos\phi\cos\theta$,

$$h(x_s, y_s) = \frac{A}{H} \tag{16}$$

2.3.2 Spherical landing site

Alternatively, if the descent camera points towards a perfect spherical surface of radius R centered at O_R, the surface point P relates to the spacecraft altitude H as follows:

$$\begin{aligned}\overrightarrow{\boldsymbol{O_R P}} &= \overrightarrow{\boldsymbol{O_R O_c}} + \overrightarrow{\boldsymbol{O_c P}} \\ &= -(R+H)\hat{\mathbf{k}} + \overrightarrow{\boldsymbol{O_c P}}\end{aligned} \tag{17}$$

Given that $(\overrightarrow{\boldsymbol{O_c P}} - (R+H)\hat{\mathbf{k}}) \cdot (\overrightarrow{\boldsymbol{O_c P}} - (R+H)\hat{\mathbf{k}}) = R^2$,

$$\begin{aligned}(X-\alpha(R+H))^2 + (Y-\beta(R+H))^2 + (Z-\gamma(R+H))^2 = R^2 \\ X^2+Y^2+Z^2 - 2(R+H)(\alpha X + \beta Y + \gamma Z) + (R+H)^2 - R^2 = 0\end{aligned} \tag{18}$$

which, after rearranging (12) and substituting for X and Y, transforms into

$$(x_s^2 + y_s^2 + 1)Z^2 - 2(R+H)(\alpha x_s + \beta y_s + \gamma)Z + H^2 + 2RH = 0 \tag{19}$$

Solving the quadratic equation in Z and inverting yields the inverse depth map

$$h(x_s, y_s) = \frac{x_s^2 + y_s^2 + 1}{(R+H)A - \sqrt{(R+H)^2 A - 2RHB}} \tag{20}$$

2.4 Event-camera emulation

Having described the geometry of an idealised pinhole camera and image sensor, we now introduce an alternative to the standard frame-based readout. While a standard frame camera outputs intensity frames at a fixed rate based on an internal clock, the pixels of a dynamic vision sensor asynchronously report changes in scene brightness (log intensity), resulting in a stream of discrete *events*. The operation principle of a dynamic vision pixel is differential in nature: rather than accumulating light for a certain amount of 'exposure time' like traditional frame-based sensors, they register *changes* in illumination. Since each pixel can detect both positive and negative changes in brightness, there are separate thresholds Θ_{ON} and Θ_{OFF} for positive and negative events. Once the brightness changes beyond one of the thresholds, the pixel outputs an event and resets its baseline to that threshold. A single event in the Address Event Representation format takes the form of (t, x_s, y_s, p), where t is the timestamp of the event, p indicates its polarity (positive or negative), and (x_s, y_s) represents the pixel coordinates. The lower data generation rate compared to a frame sensor allows for readout rates in the range of $2 - 1200\,\mathrm{MHz}$, sub-millisecond temporal resolution and low power consumption [3], whereas the sensitivity and logarithmic scale of the independent pixels afford the sensor a very high dynamic range. In order to consider an event-based camera (a data-driven sensor), for navigation and landing, we need to resort to hardware emulation given the difficulty in sourcing such data *in situ*. Fortunately, toolboxes such as v2e (video-to-event) have been designed to faithfully replicate the output of event cameras in software, including common sources of noise based on the performance of the sensors in various lighting conditions [11].

The v2e pipeline consists of several processing steps designed to convert frames into event sequences. First, the frames of the original RGB video are converted to grayscale based on the brightness Y of the scene at each pixel, and the converted frames are assigned timestamps derived from the frame rate. The next step involves converting the linear frame brightness scale to the logarithmic event brightness scale. This is particularly important due to the limited dynamic range of computer graphics formats (the brightness information content is usually limited to 8 bits, meaning that the brightness is limited to values between 0 and 255). Therefore, v2e linearises the logarithmic scale for low-intensity values ($Y < 20$) in order to reduce quantisation noise in the output. As the final preprocessing step before event generation, the output of the log-linear mapping step is fed into an infinite impulse response (IIR) low-pass filter that models the physical response profile of an analog pixel in low-light conditions, where the bandwidth of the filter increases proportionally to the brightness value Y.

In addition, the pipeline models effects observed in real hardware pixels, such as hot pixels that produce events at a high rate even in the absence of input, leak current resulting in sporadic ON events (typically with a rate of about 0.1Hz) and shot noise observed due to the disproportionate contribution of individual photons to the integration of photon counts in very dark conditions (modelled as a Poisson process).

The emulator exposes several camera settings and noise profile parameters which can be tuned to obtain the desired distribution of events at the output (the values used in this study are given in Table 1). Mid-range parameter values are found to be suitable for the study of lunar landings in low-light conditions.

Table 1: Parameterisation of the emulated event-camera model

Parameter	Value	Unit
Contrast threshold mean (Θ_{ON} / Θ_{OFF})	± 0.25	N/A
Contrast threshold standard deviation (σ_{Θ})	0.03	N/A
Shot noise rate (f_{SN})	5	Hz
Leak rate (f_{LR})	0.1	Hz
IIR low-pass cutoff frequency (f_{LP})	30	Hz

3 RESULTS

An event-based dataset corresponding to optimal trajectories at different stages of a lunar landing is generated to showcase the various features of the pipeline. The selection of a suitable landing site near the lunar south pole combined with low Sun elevations allow for challenging lighting conditions which highlight the perceived benefits of dynamic vision sensing for vision-based navigation. Table 2 summarises the parameter ranges considered in each landing scenario. Section 3.1 covers the implementation and solving of the OCP underlying the minimum-fuel trajectories, while Section 3.2 details the configuration of PANGU to simulate photorealistic low-light conditions around the landing site. The synthetic events generated from the simulated landings in PANGU are presented in Section 3.3.

Table 2: Summary of the landing scenario parameterisation: the target gives the (x, y, z) Moon-centred coordinates of a site near the south pole, and the illumination gives the Sun's (range r_s, azimuth γ_s, altitude α_s)

	Braking	Approach	Descent
Descent range $[m]$	[6500, 4000]	[3500, 1000]	[2500, 100]
Initial pitch range $[\text{deg}]$	[45, 60]	[45, 75]	[-9, 9]
Illumination $[m, \text{deg}, \text{deg}]$	(1.496e11, 105, 7)	(1.496e11, 55, 15)	(1.496e11, 155, 2)
Target $[m]$		(-198974, 49, 1730162)	

3.1 Trajectory generation

By varying the descent range, velocities and attitude of the spacecraft, distinct features are captured in the image plane, simulating what a descent camera would see during braking, approach and descent to the surface. However, in the interest of dataset variability, the descent trajectories are generated from a higher starting altitude than typically observed during the terminal descent stage. Similarly, a return to zero pitch is enforced via the boundary conditions in each pinpoint landing scenario to ensure that sufficient surface features (such as craters) can be used to aid the reconstruction of the pose from events.

For each landing scenario, an initial set of boundary conditions is defined as the base profile. Optimisation of the base trajectory is performed by discretising the corresponding OCP (8) over N nodes via Hermite-Simpson direct collocation; the resulting sequential quadratic program is then passed to the Sparse Nonlinear OPTimizer (SNOPT)[15] for optimization. Additional optimal trajectories can be derived from the base trajectory by sampling new conditions from the preselected parameter ranges (see Table 2) and in turn applying a continuation method.

This approach is used to generate a dataset of 500 optimal trajectories $\{\mathbf{x}^*\}_{i=1}^{500}$. Figure 3 shows the resulting distribution of height from the surface and pitch across the three landing scenarios, where the time $n\delta t$ denotes the nth control interval given that the timestep $\delta t = \frac{t_f}{N}$ varies with each trajectory. The next step in the pipeline consists in upsampling the optimal trajectories from N nodes to the desired number of frames per second (fps). Given the magnitude of the relative velocities of features in the scene, $fps = 100\,\mathrm{Hz}$ is determined to be sufficiently high a frame rate to generate smooth footage of the PANGU simulated landing for the video-to-event conversion. As a result, 1D cubic spline interpolation is applied to each of the 500 trajectories, such that $t_f * fps$ camera poses can be generated for the following step.

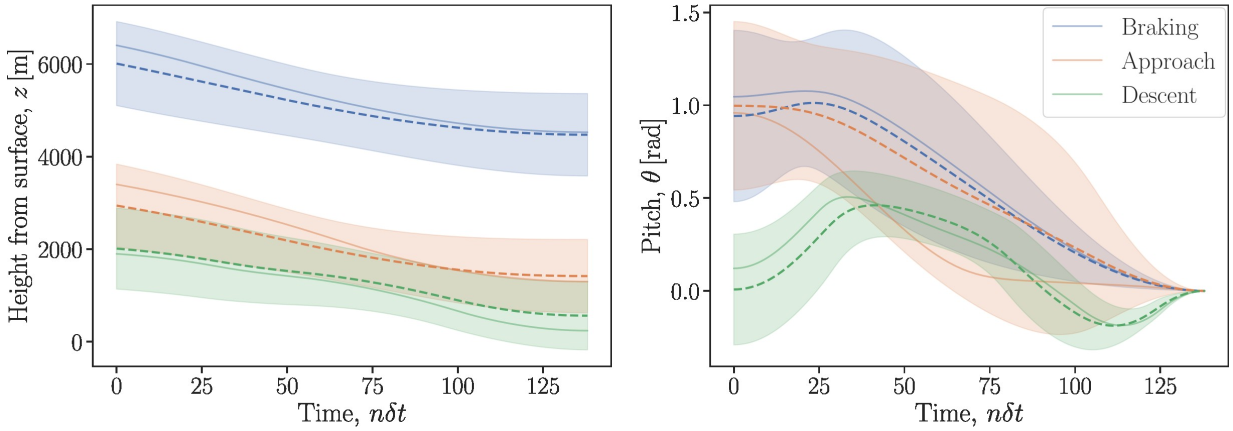

Figure 3: Height and pitch highlights from a sample dataset of 500 optimal trajectories combining braking, approach and descent characteristics (solid: sample trajectory $\mathbf{x}_i^*$; dashed: mean trajectory $\bar{\mathbf{x}^*}$; filled: $\pm 3\sigma_{\mathbf{x}}^*$ standard deviations from the mean)

3.2 Frame rendering

The target body for the simulation of the landing trajectories is a mid-resolution 3D lunar model (available by default in PANGU as `lunar_world_dem`). The whole moon model uses projected digital elevation maps (DEM) to recreate the cratered lunar landscape with high-resolution features concentrated along a strip around the equator to support landing simulations. The landing site is selected to be along this strip at a location with interesting surface features. Additionally, the simulator offers several options for the directional reflectance properties of the body, such as the Lambert and Hapke models, which allow for realistic surface shadow rendering in different lighting conditions. The sequence of camera poses along the interpolated trajectories are passed to the PANGU simulator by means of a `flight.fli` file which controls the camera viewpoint during the simulation. The camera model in PANGU is treated as a pinhole camera with camera matrix

$$\mathbf{K} = \begin{bmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

where, in the idealized case, the aspect ratio $a = 1$, the skew $s = 0$ and the principal point $(c_x, c_y) = (W/2, H/2)$. Drawing inspiration from the characteristics of descent cameras found on previous lunar landers, we set the frame size to $(W, H) = (1024, 1024)$ such that

$$f = \frac{W}{2 \tan \frac{\vartheta}{2}} \tag{22}$$

with the field of view set to $\vartheta = 45°$. These characteristics allow the simulated pinhole camera to capture a wide range of features across the three landing scenarios, including the lunar horizon during high-pitch phases and the surface craters as they enter/leave the field of view. Figure 4 depicts 15 scenes taken from the simulation of the three landing scenarios described in Table 2. In both the braking and approach simulations, the spacecraft pitch decreases over the entire trajectory, with significant roll and yaw motion. The last row describes a near-ventral descent to the surface.

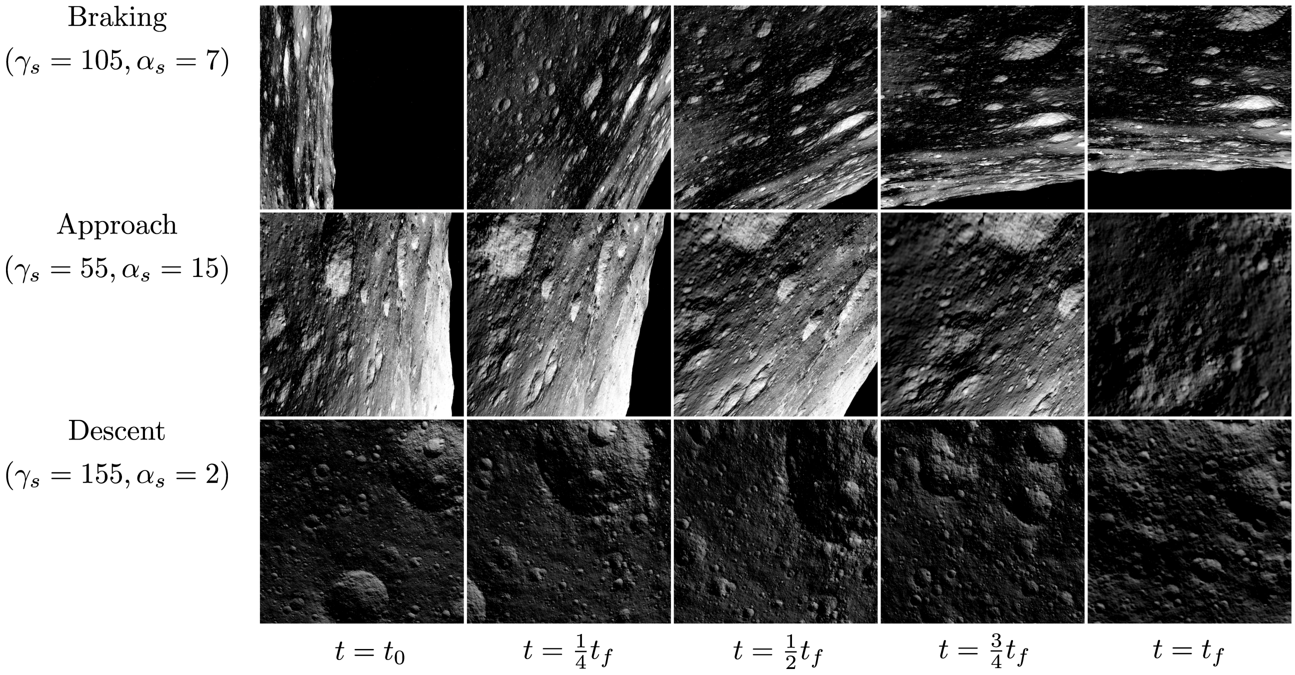

Figure 4: PANGU renders of the lunar surface at different stages of the simulated descent and under the low-light conditions given in the left margin (Hapke reflectance)

The second output of the pipeline is related to the motion field, discussed in Section 2.3. Given the groundtruth trajectories, the motion field induced by the descent can be computed at each frame shown in Figure 4 according to (13) and given the assumption of a spherical landing surface 2.3.2. Despite only being an ideal representation of the motion of 3D points projected into the image plane, the motion field finds value in its relation to the optical flow. Unlike the motion field which is computed

from idealised equations of motion, the optical flow can be estimated from image measurements. By treating the optical flow as an approximation of the groundtruth motion field (the validity of which is application dependent), it is possible to reconstruct the spacecraft motion from measurements.
Figure 5 overlays the motion field computed from the trajectory groundtruth onto the renders of Figure 4. While the motion field is computed at each pixel in the image, a sparser quiver plot is shown for clarity. Alternatively, a directional flow colour code can be used to map the motion field to RGB values, creating smooth colour gradients over the full frame [12]. By including such groundtruth data, it is envisioned that the pipeline can support frame-based motion reconstruction methods, such as those based on convolutional neural networks.

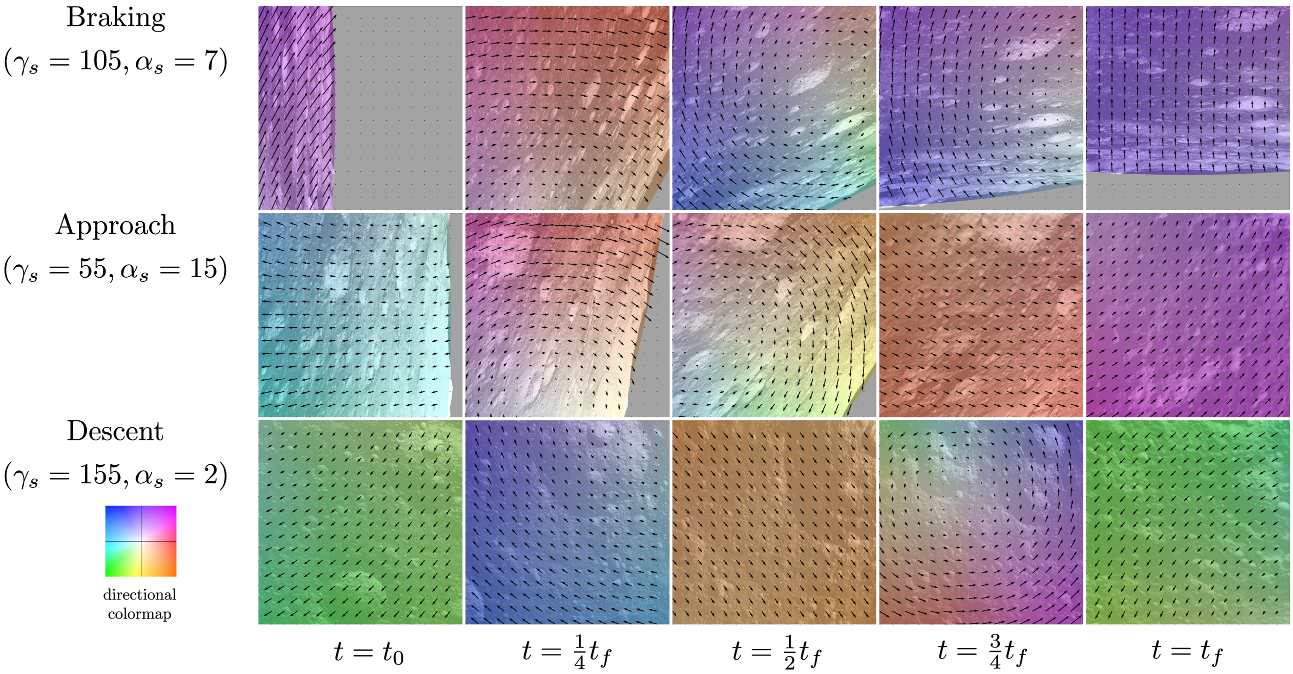

Figure 5: Motion field induced by the relative dynamics of the spacecraft and the scene at different stages of the simulated descent (based on the spherical landing surface assumption 2.3.2)

3.3 Event stream generation

The last step in the pipeline converts the sequences of frames rendered by PANGU (and concatenated into videos at $fps = 100\,\mathrm{Hz}$) into streams of events, thereby emulating the output of a dynamic vision sensor. The `v2e` toolbox, whose tuning parameters are listed in Table 1, is configured using mid-range parameter values that suit the low-light conditions depicted in Figure 4. The parameterisation results in a noisy sensor emulation where, in addition to events generated by moving edges, background pixels fire events in a seemingly random fashion (see the noisy area beyond the horizon in the top left event frame of Figure 6). While more realistic, this will render any reconstruction task from events more challenging.
By design, the pipeline outputs event streams in a spatiotemporal format. However, in order to have a meaningful comparison between the original frames and the spatial distribution of synthetic events around a small temporal neighbourhood, events can be accumulated into *event frames*. The event frames shown in Figure 6 accumulate the events over a short time window of $\Delta t = 2\,\mathrm{ms}$ centred on each of the times $\{t_0, t_f/4, t_f/2, 3t_f/4, t_f\}$.
Salient features in the scene are highlighted with events as a result of the motion of the camera. This leads to near saturation of the middle event frames $(t_f/4, t_f/2, 3t_f/4)$, when the motion of the camera

is most pronounced, and to comparatively quiescent event frames in the final stage of the landing, when the scene is nearly static (also, very little motion is perceived at the start of the near-ventral descent). These results highlight the motion-dependent properties of the output of dynamic vision sensors.

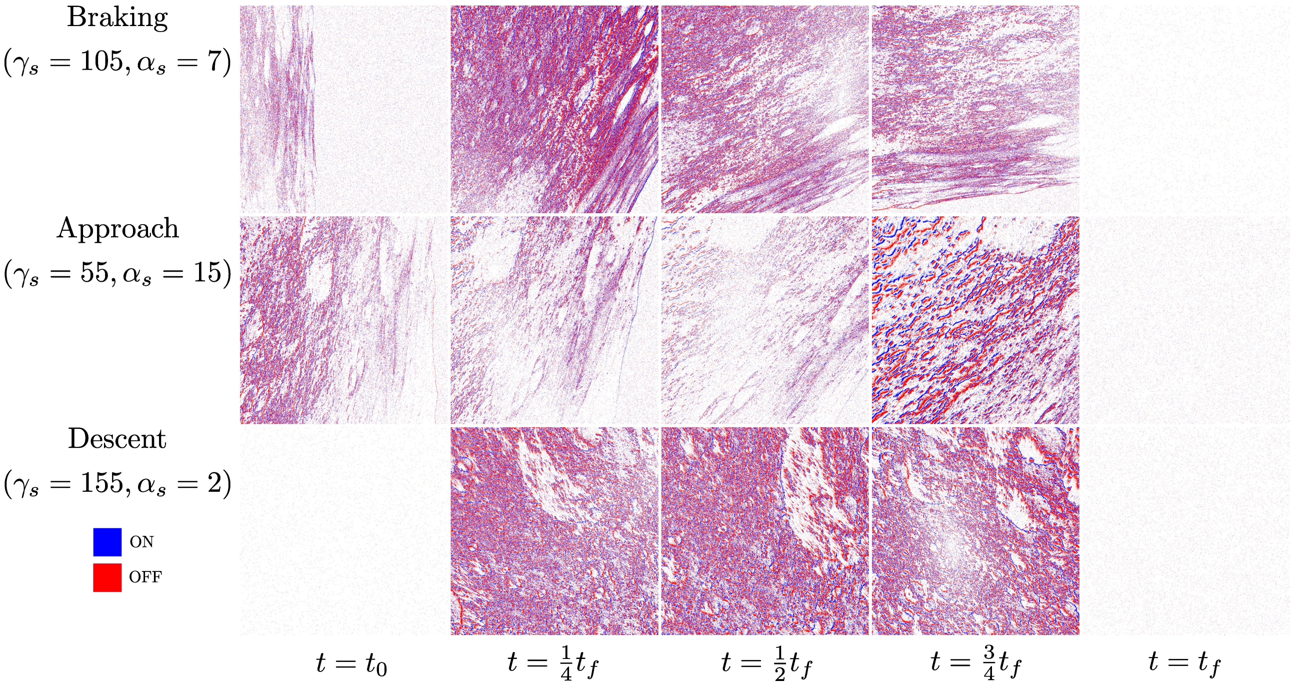

Figure 6: Event frame representation of the lunar surface at different stages of the simulated descent and under the low-light conditions given in the left margin: light blue (dark blue) pixels denote positive/ON (negative/OFF) changes in scene brightness

4 LIMITATIONS

Given the simplified visual navigation geometry (i.e., the coincident camera and spacecraft coordinate frames) and the use of an idealised camera model, the pipeline proposed in this work serves as a baseline for more realistic event-based dataset generation from landing trajectories in the future. An important limitation of the current setup is the size and variability of the generated dataset. The formulation of the landing problem as a minimum-fuel OCP ensures that the generated optimal trajectories remain smooth. This prevents the occurrence of any kind of jerky camera motion or correction manoeuvres which would translate into very distinct event distributions. Moreover, the trajectory generation is currently based on the variation of only a subset of the boundary conditions. Future work may look at expanding the parameter space beyond the ranges proposed in Table 2 in an attempt to improve the resulting distributions of landing profiles. The pipeline is designed to be modular in light of these anticipated additions, and it is straightforward to replace the trajectory optimisation component with a more suitable source of landing trajectories. As such, the pipeline may be adapted to take in camera poses from an external software package (e.g., SPICE kernels).

Dataset heterogeneity can also be achieved by varying the simulated landing environments. While the generation of realistic shadow maps in PANGU can prove computationally expensive, it is a readily available means to alter the distribution of events generated for a given trajectory. Given space constraints, this work associates each landing scenario with a different set of low-light conditions. Repeating the landing simulation with different Sun positions could effectively augment the dataset as the changing shadows would lead to different event streams. This would provide additional material for a comparative analysis of event distributions.

Event-based datasets for navigation and landing can be prepared so as to replicate space environments as faithfully as possible. While low-light conditions are discussed to some extent in this study, additional artefacts such as those induced by high-energy particles are not. Radiation effects are already supported by scene generators like PANGU, and it may be worthwhile to study their impact on the generation of event streams in a follow-up study.

A limitation of the generation and use of groundtruth motion field data for navigation and landing is the assumption of *a priori* knowledge about the target body. While the assumption of a planar or spherical surface can be justified in considering landings on large spherical bodies such as the Moon, this is no longer the case for target bodies with irregular shapes such as asteroids. While a more detailed DEM may be used in these cases, such information may not always be available. Despite using the spherical assumption to overlay the motion field onto the frames in Figure 5, there still remains an offset owing to the use of a DEM on top of a perfect sphere in PANGU. A better understanding of these models is necessary in order to obtain a reliable estimate of the groundtruth motion field.

Finally, we remark that the value of the event-based dataset does not lie in the extent to which the synthetic events approximate what an event-camera would see upon approach of the lunar surface, as this cannot be determined with the current means available. Instead, it provides a representation of surface features over time that is fundamentally different from the frame-based state of the art, and allows practitioners to develop novel tools to extract navigation information from events. For this reason, the accumulation of events into frames, as done above in the interest of qualitative analysis, is often found to be impractical. While Figure 6 accumulates events over a short time window into a single frame, there exist alternative methods based on event counts or the coverage of a fixed image area; the difficulty lies in determining which method best captures the features relevant to the application under study. In doing so, however, one ends up emulating the operation of a frame sensor, which does not take full advantage of the asynchronous and sparse nature of events.

5 CONCLUSION

This work proposes a data pipeline for generating synthetic event-based vision datasets from the point of view of an event-based descent camera during simulated lunar landings. First, a minimum-fuel optimal control problem is solved based on trajectory specifications that approximate the braking, approach and descent stages of a typical lunar landing. Each optimal trajectory is then used as a baseline to solve additional landing profiles, thus augmenting the dataset. Knowing the pose of the camera at each point along the trajectories, a video of the lunar landing can be generated using the PANGU scene generator, where the lighting conditions can be varied to further contribute to dataset variability. In the final step, the videos are passed through a video-to-event converter which emulates the event-camera and outputs spatiotemporal streams of asynchronous events.

An initial dataset of 500 landing trajectories is presented to showcase the various components of the pipeline. By aggregating events over short time windows, an event-based representation of the lunar surface at different stages of the landing can be obtained and compared to the original PANGU scenes and groundtruth data. Additional groundtruth information, such as the motion field is made available as part of the pipeline in anticipation of reconstruction methods based on optical flow.

In follow-up work, we will further extend the dataset by considering additional landing scenarios and low-light conditions. An upcoming open-data challenge will be based on the resulting landing dataset to engage the wider computer vision communities in determining how best to process event-based data for navigation and landing[1]. The success of previous similar initiatives on spacecraft pose estimation [16][17] testifies to the effectiveness of competitive data challenges in garnering interest and solutions to novel optical navigation opportunities. Beyond the data challenge, the event-based landing dataset is envisioned to support investigations into future onboard opportunities for event- and vision-based navigation, including motion estimation based on optical flow, surface feature identification and tracking, and terrain relative navigation. By releasing this pipeline, we hope to promote the creation of new datasets for event-based navigation around other planetary bodies and asteroids, and to support the development of state-of-the-art event processing tools for future space missions.

REFERENCES

[1] S. Roffe, H. Akolkar, A. D. George, B. Linares-Barranco, and R. B. Benosman, "Neutron-induced, single-event effects on neuromorphic event-based vision sensor: A first step and tools to space applications," *IEEE Access*, vol. 9, pp. 85 748–85 763, 2021.

[2] D. Izzo, A. Hadjiivanov, D. Dold, G. Meoni, and E. Blazquez, "Neuromorphic computing and sensing in space," *arXiv preprint arXiv:2212.05236*, 2022.

[3] G. Gallego, T. Delbrück, G. Orchard, *et al.*, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[4] S. Roffe, T. Schwarz, T. Cook, *et al.*, *CASPR: Autonomous Sensor Processing Experiment for STP-H7*, 2020.

[5] M. G. McHarg, R. L. Balthazor, B. J. McReynolds, *et al.*, "Falcon Neuro: an event-based sensor on the International Space Station," *Optical Engineering*, vol. 61, no. 8, p. 085 105, 2022.

[6] B. J. P. Hordijk, K. Y. W. Scheper, and G. C. de Croon, "Vertical landing for micro air vehicles using event-based optical flow," *Journal of Field Robotics*, vol. 35, pp. 69–90, 2017.

[7] 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)*, 2021, pp. 1941–1950.

[8] S. McLeod, G. Meoni, D. Izzo, *et al.*, "Globally optimal event-based divergence estimation for ventral landing," in *Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, 2023, pp. 3–20.

[9] G. C. H. E. de Croon, J. J. G. Dupeyroux, C. De Wagter, A. Chatterjee, D. A. Olejnik, and F. Ruffier, "Accommodating unobservability to control flight attitude with optic flow," *Nature*, vol. 610, no. 7932, pp. 485–490, 7932 Oct. 2022.

[10] I. Martin and M. Dunstan, *Pangu v6: Planet and asteroid natural scene generation utility*, English, Nov. 2021.

[11] Y. Hu, S. C. Liu, and T. Delbruck, "V2e: From video frames to realistic DVS events," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2021.

[12] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. Lewis, and R. Szeliski, "A database and evaluation methodology for optical flow," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[1] The open data challenge will be hosted on the Kelvins data competition platform hosted by the European Space Agency's Advanced Concepts Team: `https://kelvins.esa.int/`

[13] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, “Event-based visual flow,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 407–417, 2014.

[14] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, “Ev-flownet: Self-supervised optical flow estimation for event-based cameras,” in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, 2018.

[15] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

[16] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Martens, and S. D’Amico, “Satellite pose estimation challenge: Dataset, competition design, and results,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 4083–4098, 2020.

[17] T. H. Park, M. Martens, G. Lecuyer, D. Izzo, and S. D’Amico, “SPEED+: Next-generation dataset for spacecraft pose estimation across domain gap,” *2022 IEEE Aerospace Conference (AERO)*, 2022.