SPADES: A Realistic Spacecraft Pose Estimation Dataset using Event Sensing

Arunkumar Rathinam[1], Haytam Qadadri[2] and Djamila Aouada[1]

***Abstract*— In recent years, there has been a growing demand for improved autonomy for in-orbit operations such as rendezvous, docking, and proximity manoeuvres, leading to increased interest in employing Deep Learning-based Spacecraft Pose Estimation techniques. However, due to limited access to real target datasets, algorithms are often trained using synthetic data and applied in the real domain, resulting in a performance drop due to the domain gap. State-of-the-art approaches employ Domain Adaptation techniques to mitigate this issue. In the search for viable solutions, event sensing has been explored in the past and shown to reduce the domain gap between simulations and real-world scenarios. Event sensors have made significant advancements in hardware and software in recent years. Moreover, the characteristics of the event sensor offer several advantages in space applications compared to RGB sensors. To facilitate further training and evaluation of DL-based models, we introduce a new dataset, SPADES, comprising real event data acquired in a controlled laboratory environment and simulated event data using the same camera intrinsics. Furthermore, we introduce an image-based event representation that performs better than existing representations. In addition, we propose an effective data filtering method to improve the quality of training data, thus enhancing model performance. A multifaceted baseline evaluation was conducted using different event representations, event filtering strategies, and algorithmic frameworks, and the results are summarized. The dataset will be made available at `http://cvi2.uni.lu/spades`.**

I. Introduction

State-of-the-art spacecraft pose estimation methods leverage Deep Learning algorithms (DL), especially Convolutional Deep Neural Networks (CNN), to infer the pose of a known non-cooperative spacecraft from a single RGB image [1], [2]. DL-based approaches require abundant labelled data for training, and acquiring such orbital imagery data is expensive and challenging, considering that each target is unique. Hence, current satellite pose estimation models are trained using synthetic images generated using rendering software. However, this leads to the *domain gap* or *Sim2Real* problem [1], i.e., models trained on data collected in one domain (synthetic) show a performance drop when tested on other domains (real) due to overfitting of features specific to the training domain. To close the domain gap, *Domain Adaptation* (DA) methods [3] are adopted to improve the performance of the model in the target domain, using techniques such as adversarial learning and reconstruction approaches.

On the other hand, research has been done to explore the use of different data modalities, and event sensing was proposed [4] as a solution to close the domain gap in spacecraft pose estimation. Indeed, event cameras have gained attention in space applications [5], [6] due to their potential benefits, including high temporal resolution (up to 1μ s), a wide high dynamic range (HDR) (typically up to 140 dB), low latency, and low power consumption [7]. Event sensors capture sparse data, and each pixel is independently activated by changes in light intensity, leading to asynchronous responses. Higher HDR values result in smaller solar exclusion angles, making them well-suited for orbital sensing. The HDR and asynchronous response characteristics of the event sensors help to perceive the target in a way that reduces the sensitivity to drastic illuminations, thus narrowing the domain difference [4].

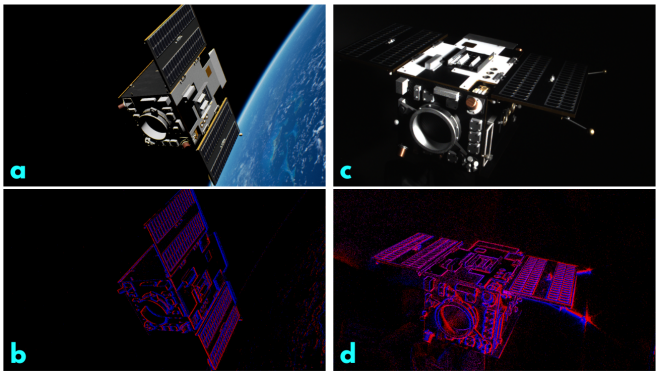

Fig. 1: Samples from the SPADES dataset. (a) RGB image generated using Unreal Engine, (b) Event data generated using the ICNS simulator, (c) Real image acquired in the lab, (d) Real event data acquired in the lab.

The SEENIC dataset [8], introduced in [4], stands out as the first and only event-sensing dataset available for spacecraft pose estimation tasks. However, the dataset lacks diversity in the distribution of pose labels for synthetic data and does not include ground-truth relative pose labels for real data, hindering the efficient validation of algorithms. To better understand the event data under more realistic orbital scenarios while facilitating training and evaluation of the DL models, we introduce a **new event dataset**, called **SPADES** - ***SPA**cecraft Pose Estimation **D**ataset using **E**vent **S**ensing*, as our *primary* contribution. The proposed SPADES dataset uses the Proba-2 satellite of the PROBA-2 mission [9] as the target. This dataset includes simulated and real event data obtained from a realistic satellite model at the SnT Zero-G testbed facility [10], [11]. Our *secondary* contribution focusses on pre-processing techniques for event data. We introduce an image-based event representation with three channels, leveraging 2D CNNs while offering better performance than existing representations. Furthermore, we propose a

* This work was funded by the Luxembourg National Research Fund (FNR) under the project reference C21/IS/15965298/ELITE.
[1] authors are associated with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. `firstname.lastname@uni.lu` [2] author is associated with Télécom Physique, Université de Strasbourg.

mask-based technique for filtering event frames that contain sufficient object details for training while assisting the DL model to learn better and enhance its performance. Finally, to assess baseline performance, we implement existing DL algorithms from the two main spacecraft pose estimation approaches and present the results.

The article is organised as follows: Section II presents related datasets, algorithms, and event representations. Section III presents the proposed SPADES dataset and describes the synthetic data generation pipeline and real data acquisition. Section IV introduces the new event representation and filtering approach along with the evaluation results and finally Section V presents the conclusion.

II. Related Works

A. Datasets for Spacecraft Pose Estimation

The first generation image datasets, SPEED [12] and URSO [13], were oriented toward synthetic data and used simulators to generate realistic renderings of targets in orbit. Recent image datasets, such as SPEED+ [1], SPARK 2022 [14], and SHIRT [15], have included real data from laboratory setups in addition to simulated data. This inclusion serves to validate the DL algorithms' performance in realistic scenarios that simulate space environments.

An event-sensing dataset, SEENIC [8], was introduced in [4] to assess the domain gap in the estimation of the pose of the spacecraft using event data. Despite being the first and only event dataset for spacecraft pose estimation, several limitations are associated with this dataset, summarised below. First, the target model is the Hubble Space Telescope (HST), whose actual dimensions are 13 m in length and 4 m in width. A scaled version of the HST (approximately 1:40 to 50) was used for real data collection, and the precise dimensions of the physical and simulation models were not disclosed. Second, the HST mockup was 3D printed for simplicity, lacking precision and surface texture, which affected the overall quality of the dataset. Third, the real event data lack relative pose labels for the target in the camera reference frame, and the authors rely on measurements between successive poses for their metrics. Finally, the synthetic dataset was generated from a single trajectory, resulting in an imbalance in the pose distribution and also lacking variation in lighting scenarios.

B. Algorithms

The two prominent approaches in DL-based spacecraft pose estimation are the *Direct or End-to-End* approach and the *Hybrid Modular* approach [2], [16]. The direct approach [13], [17] is based on the direct regression of a pose label from an input image. The hybrid pipeline [18] involves a sequence of steps, which includes using an object detection network to detect the target in the image, followed by a keypoint regression network to regress the location of the 2D keypoints, and finally using the Perspective-n-Point (PnP) solver to estimate the pose from 2D-3D correspondences. A brief discussion of existing DL-based satellite pose estimation approaches and datasets is presented in [2].

The baseline evaluation on the SEENIC dataset in [4] employed a *Hybrid* pipeline (without DA techniques) trained with synthetic data and tested with real data. During the performance evaluation on real data, the authors resort to measuring errors between successive poses as a performance metric due to hardware constraints that prevented them from directly obtaining the true relative pose of the object within the camera reference frame. However, it should be noted that such metrics are susceptible to errors that accumulate over time due to drift. Although errors between successive poses may initially appear minor, they can eventually lead to a significant deviation from the actual ground truth.

To mitigate such issues and assess performance using standard pose metrics, the proposed SPADES dataset is supplemented with ground-truth pose labels containing target poses in the camera reference frame for both data modalities.

C. Event Data Processing

An event stream is the sequence of events triggered by the change in light intensity as recorded by individual sensor pixels. Each event readout in the form of a tuple $e = (x, y, p, t)$, where x and y denote the pixel coordinates, p indicates an increase or decrease in intensity (polarity), and t represents the global timestamp of the event in microseconds (μ s) as recorded on the camera timeline. Thus, a sequence of events over a time window of τ can be represented as $E_\tau = \{e_i \mid t < i < (t + \tau)\}$. These accumulated events can be processed and represented in various formats, including images [4], [19], [20], voxels [21], graphs [22], 3D point sets [23], and motion compensated event images [24].

Image-based Representations: The image-based representations convert sparse events into dense frames to take advantage of existing CNN architectures. The event-to-frame (E2F) [4] representation works by accumulating events over a given time window or an event batch, followed by normalisation and exported as an intensity image. The Locally Normalised Event Surfaces (LNES) representation [19] effectively retains temporal and polarity information during the conversion. Within the LNES representation, each event frame consists of two channels, $I \in \Re^{W \times H \times 2}$, distinguished by the polarity of the event. Using individual channels for positive and negative events preserves the polarities and limits event overriding [19]. The Time Surfaces (TS) [20] representation aims to preserve temporal information from the event stream while discarding polarity details. Unlike LNES, TS is generated by applying an exponential decay to the time within the time window using the last set of events recorded in the neighbourhood of the current event $e_i(x, y)$.

III. Dataset

A. Synthetic Data Generation

Trajectory Selection: Generating an event stream involves creating a motion sequence (of images) and can be achieved by moving either the camera or the target while keeping the target within the camera's field of view (FoV). Our data generation pipeline employs a fixed camera and a moving target. The trajectory generation comprises two steps.

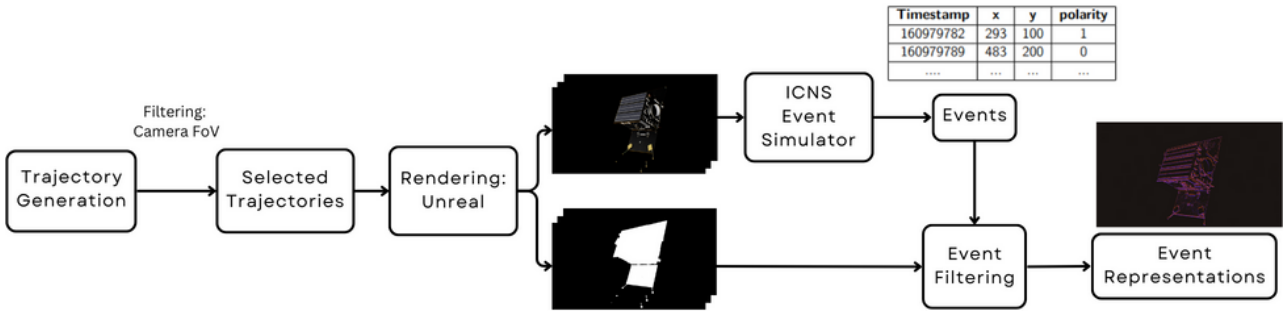

Fig. 2: Overview of the synthetic data generation pipeline.

The *first step* is to initialise the starting and ending poses of the sequence, denoted as $[\mathbf{q_{start}}|\mathbf{t_{start}}]$ and $[\mathbf{q_{end}}|\mathbf{t_{end}}]$, where $\mathbf{q_x}$ represents the orientation as quaternions and $\mathbf{t_x}$ represents the positions as a translation vector. The quaternions were sampled from a uniform distribution. In the translation vector, t_z ranges between 3.5 and 12 m, determined based on factors such as focal length, sensor size, resolution, and target size; t_x and t_y are constrained by the camera's FoV.

The *second step* involves interpolation between the start and the end pose over the n steps: $S = ([\mathbf{q_0}|\mathbf{t_0}], [\mathbf{q_1}|\mathbf{t_1}], ..., [\mathbf{q_n}|\mathbf{t_n}])$. The interpolation methods employed are either Helix or Spline interpolation. After generating each sequence, each pose within the sequence is verified with 2D keypoint projection results, ensuring that all edge keypoints remain within the image. Table I summarises the size of the dataset, the number of trajectories, the interpolation methods and the characteristics of the range. Fig. 2 illustrates the complete data generation pipeline.

RGB data: After generating the ground truth sequence, we render RGB images using a Unreal Engine[1] (UE) simulator. To render these synthetic images, we used the CAD model of the Proba-2 satellite downloaded from the ESA Science Satellite Fleet[2]. Communication with the UE environment is facilitated through the UnrealCV library [25]. The UE environment incorporates 16k Earth texture maps from the Blue Marble collection[3], employs physically-based shading, and includes Rayleigh scattering to simulate atmospheres. Prior to rendering, camera poses are randomly sampled and fixed for each sequence, resulting in diverse backgrounds and lighting scenarios. The target is placed relative to the camera pose using the corresponding ground truth pose, and the images are subsequently rendered.

Event data: The event data stream is generated using the ICNS event simulator [26], which uses Blender[4] to simulate the behaviour of neuromorphic sensors. This simulator offers a more realistic simulation of the sensor output by accurately modelling the sensors' pixel-level behaviour, taking into account factors such as latency, noise, and other relevant characteristics. Samples of generated synthetic event data are depicted in Fig. 4. Aligning the pixel-level behaviour with EVK4 cameras is not considered for this version of synthetic data.

[1] www.unrealengine.com [2] http://scifleet.esa.int [3] visibleearth.nasa.gov [4] www.blender.org

B. Real Data Collection

Testbed: The Zero-G Laboratory facility [10], [11] at the SnT, University of Luxembourg, was used for real data acquisition. The laboratory setup covers a space with dimensions of $5 \times 3 \times 2.3$ m (WxLxH) and is equipped with two UR10 robotic arms mounted on the linear rails installed on the ceiling and side wall. Furthermore, the facility is equipped with an OptiTrack motion capture system (OTS) comprising eight cameras that enable the tracking of a predefined rigid body fitted with either active or passive markers.

Event Camera: The camera utilised in data acquisition is Prophesee Metavision EVK4-HD[27] equipped with the SONY IMX636ES(HD) event vision sensor, representing the latest technology available at the time. This camera has a resolution of 1280×720 pixels on a $1/2.5$" sensor, each pixel measuring 4.86μm. Furthermore, it is equipped with a 6mm fixed focal length lens, providing a horizontal FoV of 54.6°. The maximum read-out throughput is 3 Gevents/s, and the typical power consumption is 0.5 to 1.5W max.

Proba-2 Mockup: The satellite mockup used in the experiments weighs approximately 7 kg, with a scaling ratio of 1:2.5. The dimensions of the physical model, along the X, Y and Z axes, are $0.64 \times 0.24 \times 0.416$m, respectively. The mock-up was manufactured through a third-party vendor, and the materials were carefully selected to minimise deviations from the textures found in the CAD data.

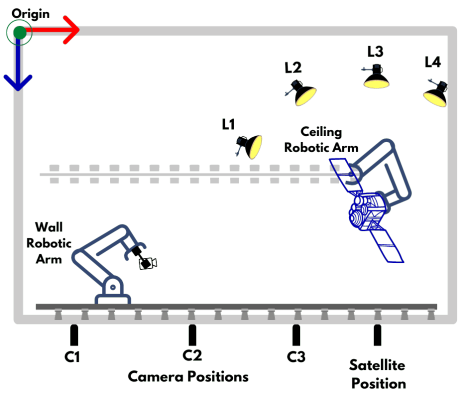

Fig. 3: Schematic of the Zero-G lab setup for real data collection.

Light Setup: The intensity of sunlight in orbit corresponds to the solar irradiance of 1366 W/m^2 or illuminance of $\sim$ 163,000 lux. To emulate orbital lighting, the Aputure LS-600D-PRO LED lamp was used as a light source for data

collection. The lamp can produce 224,200 lux at a distance of 1 m when mounted with a Fresnel F10 lens with a spot angle of 15°. In our setup, the lamp is fixed at a distance of 1.5 m as a trade-off between safety and accuracy, producing 120,000 lux for the colour temperature of 5800K.

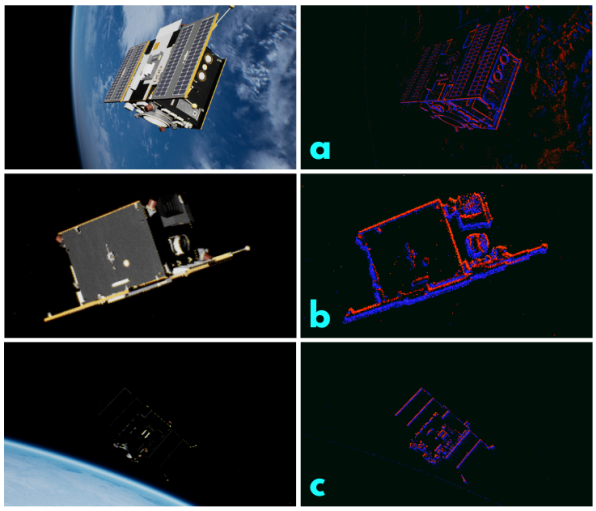

Fig. 4: Synthetic data samples from RGB and Event sensor. (a) images with good lighting and background, (b) images with good lighting and no background, and (c) images with harsh lighting.

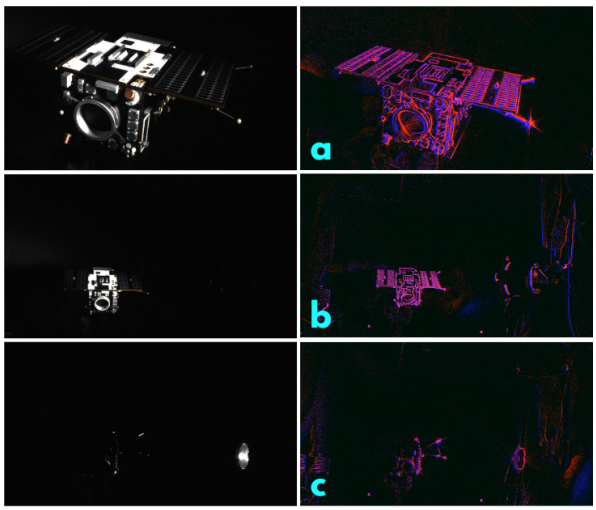

Fig. 5: Real data samples from FLIR RGB camera and Prophesee Event Camera. (a) Close-range with L1, L2 lighting, (b) Far-range with L1, L2 lighting, and (c) Far-range with L3, L4 lighting.

Event camera calibration: To calibrate the event camera, we used greyscale image reconstruction [28], which utilises a neural network-based image reconstruction technique to move from events to grayscale image. The camera is moved around the fixed calibration board to collect the calibration sequence. The event stream is extracted in batches with a fixed time window during processing. Greyscale image reconstruction for each batch of events was achieved using the E2VID model [29], and the corresponding pose labels were extracted from the OTS. The reconstructed images were subsequently processed to compute the camera intrinsics using the MATLAB Camera Calibration toolbox. For extrinsic calibration, the hand-eye calibration approach [30] was used to find the transformation between the actual camera reference frame and the rigid body camera frame defined in OTS. This fixed transformation maps the raw pose label of the rigid body to the actual camera pose in the OTS coordinate system. Similarly, the satellite mockup has a pre-defined marker setup to collect pose labels in the OTS coordinate system. Data between the actual camera reference frame and satellite poses were synchronised on the basis of timestamps, and the transformation was applied to yield ground truth data representing the relative pose information of the object in the camera reference frame.

Data Collection: During real data acquisition, various combinations of lighting conditions (L1, L2, L3, L4) and camera positions (C1, C2, C3) were employed, as illustrated in Fig. 3. Based on camera movement, trajectories are classified into two groups: *static* and *dynamic*. In *static* trajectories, the camera (representing the chaser satellite) maintains a constant distance from the target, observing the target's motion, thereby emulating the observation phase. In *dynamic* trajectories, the camera approaches the target with linear or spiral motion, while the target exhibits stationary or rotational movement along its Y-axis. Further details on the real data set can be found in Table I.

	Synthetic	**Real**
Sensor resolution	1280x720	1280x720
Dataset size	179,400 (no. of poses)	16,930
No. Trajectories	300	32
No. poses/traj	598	529 (avg.)
Interpolation	80% Spline + 20% Helix	-
Range	3.5 - 12 m	3.5 - 9 m
Range dist.	Close, Mid, Far, Limit	Close, Mid, Far
Lighting	Easy, Hard	L1, L2, L3, L4
Rendering	Unreal Engine (RGB)	-
Event Camera	ICNS Emulator	Prop. EVK4HD
Background	Earth	-
Filtering	Bbox/Mask	Min. Event count

TABLE I: SUMMARY OF PROPOSED SPADES DATASET

IV. EXPERIMENTS

This section presents the proposed preprocessing techniques, including the 3-channel event representation and mask-based event frame filtering method, and a baseline comparison of the current spacecraft pose estimation algorithms on the SPADES dataset.

A. Event Representation

The proposed event representation, namely 3-Channel (3C), is a pseudo-frame with three channels to leverage the algorithms designed for RGB images. Improving upon the TS representation [20], the 3C representation uses exponential decay to track temporal information while it splits the actual time window W into three sub-windows of size $W/3$. Events collected within each sub-window are processed independently and organised into channels in chronological order. This approach ensures the segregation of maximal temporal information into separate channels, preserving the inherent asynchronous nature of events by dividing the time window into subwindows. This representation differs

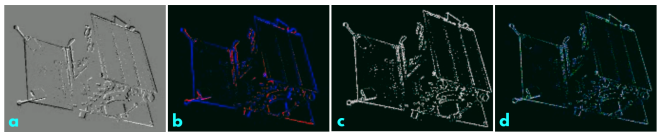

Fig. 6: Event representations (a) E2F, (b) LNES, (c) TS, (d) 3C

from the three-channel approach presented in [31], which uses polarity information and event count. For comparison, different event representations are presented in Fig. 6.

The image-based representations E2F, LNES, TS, and 3C are evaluated on an object detection task. A Faster-RCNN model with Mobilenet-V3-Large [32] backbone was used and initialised using pre-trained weights in the ImageNet dataset [33]. The model was trained with a batch size of 8 for 100 epochs, with early stopping patience set to 20. The time window size for real data is set to 0.1 s. The test dataset comprises 34,790 samples of synthetic test data (after an 80/20 split), while the entire real dataset of 16,930 samples was utilised. This configuration is consistent throughout all experiments detailed in this article. The results are evaluated using the standard metrics, Average Precision (AP) and Average Recall (AR) at varying Intersection-over-Union (IoU) thresholds 0.5, 0.75 and for object bounding box sizes [34]: small (S) [$A_{bbox} \leq 150 \times 150$], medium (M) [$150 \times 150 < A_{bbox} \leq 300 \times 300$] and large (L) [$300 \times 300 < A_{bbox}$], where A_{bbox} denotes the area of the bounding box in sq. pixels. Table II demonstrates that the 3C representation produces the best results for object detection in both synthetic and real data. It should be noted that the 3C surpasses the original TS representation by preventing the loss of information through the use of subwindows.

Rep.	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	AR	AR_S	AP_M	AR_L
Synthetic									
E2F	0.98	0.74	0.60	0.51	0.61	0.66	0.67	0.61	0.63
LNES	0.98	0.73	0.59	0.51	0.63	0.66	0.66	0.62	0.66
TS	0.98	0.74	0.59	0.52	0.65	0.65	0.65	0.63	0.64
3C	**0.99**	**0.95**	**0.84**	**0.82**	**0.79**	**0.89**	**0.89**	**0.90**	**0.83**
Real									
E2F	0.69	0.49	**0.45**	0.46	0.33	0.55	0.54	0.56	0.55
LNES	0.63	0.49	0.43	0.44	0.22	0.54	**0.59**	0.54	0.34
TS	0.63	0.48	0.42	0.44	0.27	0.53	0.57	0.55	0.38
3C	**0.71**	**0.50**	0.40	**0.48**	**0.38**	**0.58**	0.55	**0.59**	**0.57**

TABLE II: EVALUATION OF DIFFERENT EVENT REPRESENTATIONS ON SYNTHETIC (TEST) AND REAL DATASETS.

B. Event Frame Filtering

The initial examination of the event frames from the synthetic dataset indicated that not all frames contain the same amount of information. Since the synthetic events are derived from the RGB images, the optical flow along edges (and sometimes surfaces) can prevent the generation of events. The frames affected in this way have very little information about the object itself, as shown in Fig. 7-b. Our preliminary assessments revealed the necessity of filtering good-quality event data to enhance the model's learning process and improve performance. To address this, we introduce a new method called *mask-based filtering*, which involves screening event frames with sufficient information using the target's segmented mask and computing the distribution of the events with this area. First, we define a discrete uniform distribution,

$$\begin{array}{rccc} \mathbf{p}_{\text{uniform}} & : & \mathbf{M}_{\text{pixel}} & \rightarrow & \mathbb{R} \\ & & (x,y) & \mapsto & \frac{1}{N} \end{array} \quad (1)$$

where N is the number of pixels of the mask and $\mathbf{M}_{\text{pixel}}$ is the set of pixels (x, y) within the mask. Next, a discrete distribution for event data $\mathbf{p}_{\text{event}}$, as follows,

$$\begin{array}{rccc} \mathbf{p}_{\text{event}} & : & \mathbf{M}_{\text{pixel}} & \rightarrow & \mathbb{R} \\ & & (x,y) & \mapsto & \begin{cases} \frac{0.99}{N} & \text{if } (x,y) \in \mathbf{E} \\ \frac{0.01}{N} & \text{if } (x,y) \notin \mathbf{E} \end{cases} \end{array} \quad (2)$$

where $\mathbf{E}$ is the event stream. Subsequently, the KL-divergence is calculated between the two $\mathbf{p}_{\text{event}}$ and $\mathbf{p}_{\text{uniform}}$ to filter out inadequate pose labels using a threshold set across the dataset. For comparison purposes, a simple filtering technique known as *bbox-based filtering* is introduced, which relies on the area of the bounded box of the object. This approach involves calculating the proportion of event occurrences within a specific bounding box relative to the box's area. A predetermined threshold value is then established to eliminate pose labels from the training data.

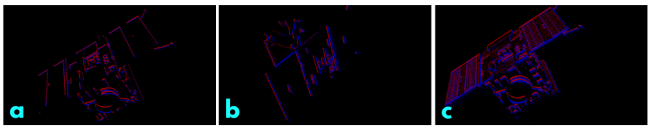

Fig. 7: Mask-filtering samples. (a) and (b) samples removed from training data, (c) samples included in training data.

First, we evaluate the event representations and filtering approaches in object detection tasks. The best-performing representation and filtering technique will be adopted for the baseline evaluation of pose estimation. It is important to note that object detection tasks are also a key component of the *hybrid* pose estimation approach.

Filt.	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	AR	AR_S	AP_M	AR_L
Synthetic									
w/o Filt.	**0.99**	**0.95**	**0.84**	**0.82**	**0.79**	**0.89**	**0.89**	**0.90**	**0.83**
Bbox	0.89	0.58	0.69	0.52	0.67	0.75	0.77	0.68	0.78
Mask	0.98	0.84	0.74	0.59	0.66	0.79	0.80	0.70	0.81
Real									
w/o Filt.	0.69	0.48	0.38	0.49	0.38	0.54	0.52	0.59	0.57
Bbox	0.71	0.50	0.40	0.48	0.38	0.58	0.55	0.59	0.57
Mask	**0.72**	**0.53**	**0.43**	**0.51**	**0.41**	**0.59**	**0.59**	**0.61**	**0.57**

TABLE III: EVALUATION OF FILTERING TECHNIQUES ON SYNTHETIC (TEST) AND REAL DATASETS.

Additional filtered datasets were generated for each technique to determine the most effective filtering method. The original synthetic dataset (without filtering) contained 143,760 pose labels; after mask-based filtering, there were 94,147 labels (65%), and box-based filtering yielded 113,876 labels (79%) for the training set. Three distinct CNN models were trained using synthetic data and evaluated on synthetic and real test data. The test data was filtered solely based on an event count threshold (more than 10,000 events) to identify valid event frames. All filtering experiments utilised

the *3C representation* for both training and evaluation, and the results are summarized in Table III. The result shows that the model tends to overfit on w/o filtering data, as seen in the synthetic test results. After discarding the overfitting results, it is evident that the *mask-based filtering* performs better in both synthetic and real data, even though it was trained with fewer data.

C. Pose Estimation Baseline

The baseline evaluation investigates two main algorithmic approaches, as discussed in Section II-B. In the **Direct** method, the input image is processed through two branches of the network. In the first branch, the image is fed into an object detector, and the resulting Region of Interest (RoI) is then forwarded to a CNN feature extraction backbone. This backbone is followed by a fully connected layer that estimates the rotation parameters using the Fisher Matrix representation [35]. In the second branch, the image is processed through a CNN backbone and a subsequent fully connected layer to predict translation. The CNN feature extraction backbone is Mobilenet-V3-Large, initialised with pre-trained weights from the ImageNet dataset. For the **Hybrid** approach, the network architecture resembles the one proposed in [36], employing Faster-RCNN with a ResNet-50 backbone for object detection, HigherHRNet for keypoint regression and BPnP [37] for PnP optimization during inference. The baseline results are summarised in Table IV using standard pose error metrics similar to those of [3], except for the translation error, which was modified to a relative translation error. The metrics include the relative translation error E_{T} [%], rotation error E_{R} [°], and pose error E_{P} as defined below.

$$E_{\mathrm{T}} = \|\tilde{\mathbf{t}}-\mathbf{t}\|_2/\|\mathbf{t}\|_2; \quad E_{\mathrm{R}} = 2\, acos|\langle \tilde{q}, q\rangle|; \quad E_{\mathrm{P}} = E_{\mathrm{R}}+E_{\mathrm{T}}.$$

The metric *Data*[%] represents the percentage of data for which the algorithm could accurately determine a pose above a threshold. Two confidence thresholds were used: 0.9 for object detection and 0.5 for keypoint regression. Despite the **Hybrid** method showing better performance, it produced results on a smaller portion of the data than the **Direct** method. This was mainly because the confidence score of keypoint predictions often did not meet the threshold, resulting in insufficient numbers for PnP optimisation.

Model	Data [%]	E_{T} [%]	E_{R} [°]	E_{P} [-]	Data [%]	E_{T} [%]	E_{R} [°]	E_{P} [-]
	Synthetic				Real			
Direct	**97.32**	4.29	30.43	0.57	**73.32**	5.13	81.13	1.47
Hybrid	23.98	**3.23**	**6.69**	**0.15**	17.27	**3.34**	**78.98**	**1.41**

TABLE IV: PERFORMANCE OF BASELINE MODELS ON SYNTHETIC (TEST) AND REAL DATASETS

Further examination of the results on the synthetic dataset is provided in Table V, highlights that lighting and background conditions significantly impact the algorithm's performance. The results are constrained to existing methods that do not incorporate domain adaptation techniques. It indicates a notable disparity between the synthetic and real domains in the event data, resulting in a drop in performance.

Model	Data [%]	E_{T} [%]	E_{R} [°]	E_{P} [-]	Data [%]	E_{T} [%]	E_{R} [°]	E_{P} [-]
	No-BG + Easy-LI				BG + Easy-LI			
Direct	**99.89**	2.21	19.99	0.37	**98.78**	**2.93**	29.14	0.54
Hybrid	25.87	**2.12**	**2.47**	**0.06**	24.57	2.98	**3.76**	**0.09**
	No-BG + Hard-LI				BG + Hard-LI			
Direct	**97.48**	4.88	32.45	0.62	**91.74**	5.03	40.14	0.75
Hybrid	22.78	**4.02**	**7.53**	**0.17**	20.64	**4.89**	**13.07**	**0.28**

TABLE V: IMPACT OF BACKGROUND (BG) AND LIGHTING (LI) CONDITIONS ON SYNTHETIC (TEST) DATASET

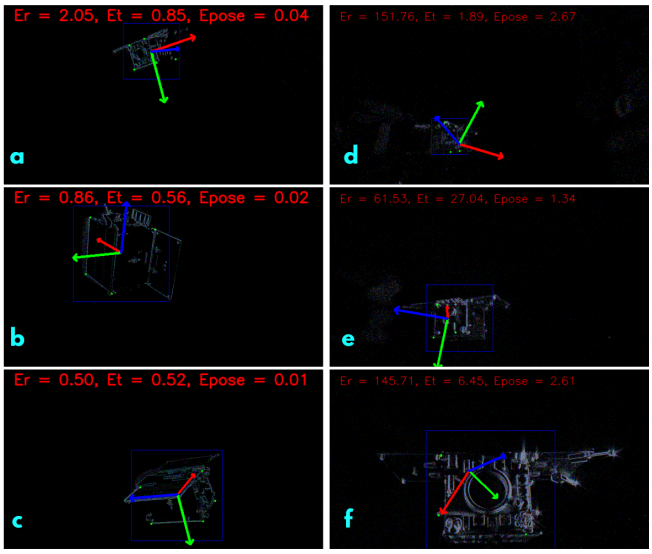

Fig. 8: Real dataset samples with estimated keypoints and poses. (a,b,c) Best performing, (d,e,f) Worst performing.

V. CONCLUSION AND FUTURE WORK

In summary, this study introduced the SPADES dataset, a comprehensive resource containing both synthetic and real event data. Its purpose is to support the training and validation of deep learning (DL) algorithms for spacecraft pose estimation based on events. The proposed 3-Channel event representation showed better performance in object detection tasks compared to existing representations. Furthermore, the use of a mask-based data filtering technique improved the quality of training data, resulting in improved algorithm performance. However, the experimental findings of the baseline models highlighted the persistent gap between the synthetic and real event data. The characteristics of materials and textures significantly influenced the generation of events in synthetic versus real data. Event cameras demonstrated their potential, especially in low-light conditions, where conventional RGB cameras struggled to provide valuable information, which was evident during real data collection. Future efforts will focus on improving the synthetic dataset to bridge this performance gap effectively. Moreover, utilising the asynchronous nature of event data shows promise for advancing pose estimation and tracking methods.

REFERENCES

[1] T. H. Park, M. Martens, G. Lecuyer, D. Izzo, and S. D'Amico, "SPEED+: Next-Generation Dataset for Spacecraft Pose Estimation across Domain Gap," in *2022 IEEE Aerospace Conference (AERO)*. Big Sky, MT, USA: IEEE, 3 2022, pp. 1–15.

[2] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière, and D. Aouada, "A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects," *Acta Astronautica*, vol. 212, pp. 339–360, 2023.

[3] T. H. Park, M. Märtens, M. Jawaid, Z. Wang, B. Chen, T.-J. Chin, D. Izzo, and S. D'Amico, "Satellite pose estimation competition 2021: Results and analyses," *Acta Astronautica*, vol. 204, pp. 640–665, 2023.

[4] M. Jawaid, E. Elms, Y. Latif, and T.-J. Chin, "Towards Bridging the Space Domain Gap for Satellite Pose Estimation using Event Sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, pp. 11 866–11 873.

[5] G. Cohen, S. Afshar, B. Morreale, T. Bessell, A. Wabnitz, M. Rutten, and A. van Schaik, "Event-based sensing for space situational awareness," *The Journal of the Astronautical Sciences*, vol. 66, pp. 125–141, 2019.

[6] S. Afshar, A. P. Nicholson, A. van Schaik, and G. Cohen, "Event-based object detection and tracking for space situational awareness," *IEEE Sensors Journal*, vol. 20, no. 24, pp. 15 117–15 132, 2020.

[7] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-Based Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[8] E. Elms, M. Jawaid, Y. Latif, and T.-J. Chin, "SEENIC: dataset for Spacecraft posE Estimation with NeuromorphIC vision," Oct. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7214231

[9] K. Gantois, F. Teston, O. Montenbruck, P. Vuilleumier, and P. van Braembusche, "Proba-2 mission and new technologies overview," in *Small Satellite Systems and Services - The 4S Symposium*, December 2006. [Online]. Available: https://elib.dlr.de/46830/

[10] M. Olivares-Mendez, M. R. Makhdoomi, B. C. Yalçın, Z. Bokal, V. Muralidharan, M. Ortiz Del Castillo, V. Gaudilliere, L. Pauly, O. Borgue, M. Alandihallaj, J. Thoemel, E. Skrzypczyk, A. Rathinam, K. R. Barad, A. E. R. Shabayek, A. M. Hein, D. Aouada, and C. Martinez, "Zero-g lab: A multi-purpose facility for emulating space operations," *Journal of Space Safety Engineering*, vol. 10, no. 4, pp. 509–521, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468896723000939

[11] L. Pauly, M. L. Jamrozik, M. O. Del Castillo, O. Borgue, I. P. Singh, M. R. Makhdoomi, O.-O. Christidi-Loumpasefski, V. Gaudilliere, C. Martinez, A. Rathinam, *et al.*, "Lessons from a Space Lab–An Image Acquisition Perspective," *International Journal of Aerospace Engineering*, 2023.

[12] S. Sharma, T. H. Park, and S. D'Amico, "Spacecraft Pose Estimation Dataset (SPEED)," 2019. [Online]. Available: https://purl.stanford.edu/dz692fn7184

[13] P. F. Proença and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6007–6013.

[14] A. Rathinam, V. Gaudilliere, M. A. Mohamed Ali, M. Ortiz Del Castillo, L. Pauly, and D. Aouada, "SPARK 2022 Dataset: Spacecraft Detection and Trajectory Estimation," June 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6599762

[15] T. H. Park and S. D'Amico, "SHIRT: Satellite Hardware-In-the-loop Rendezvous Trajectories Dataset," 2022.

[16] A. Rathinam, Z. Hao, and Y. Gao, "Autonomous visual navigation for spacecraft on-orbit operations," in *Space Robotics and Autonomous Systems: Technologies, advances and applications*. Institution of Engineering and Technology, 8 2021, pp. 125–157. [Online]. Available: https://doi.org/10.1049/PBCE131E_ch5

[17] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–12.

[18] B. Chen, J. Cao, A. Parra, and T.-J. Chin, "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[19] V. Rudnev, V. Golyanik, J. Wang, H.-P. Seidel, F. Mueller, M. A. Elgharib, and C. Theobalt, "EventHands: Real-time neural 3D hand pose estimation from an event stream," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12 365–12 375, 2020.

[20] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "Hots: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1346–1359, 2017.

[21] B. Xie, Y. Deng, Z. Shao, H. Liu, and Y. Li, "Vmv-gcn: Volumetric multi-view based graph cnn for event stream classification," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1976–1983, 2022.

[22] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based spatio-temporal feature learning for neuromorphic vision sensing," *IEEE Transactions on Image Processing*, vol. 29, pp. 9084–9098, 2020.

[23] Y. Sekikawa, K. Hara, and H. Saito, "Eventnet: Asynchronous recursive event processing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3887–3896.

[24] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 632–639, 2017.

[25] W. Qiu and A. Yuille, "UnrealCV: Connecting Computer Vision to Unreal Engine," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, vol. 9915, pp. 909–916.

[26] D. Joubert, A. Marcireau, N. Ralph, A. Jolley, A. Van Schaik, and G. Cohen, "Event Camera Simulator Improvements via Characterized Parameters," *Frontiers in Neuroscience*, vol. 15, p. 702765, July 2021.

[27] "Event Camera Evaluation Kit 4 HD IMX636 Prophesee-Sony." [Online]. Available: https://www.prophesee.ai/event-camera-evk4/

[28] M. Muglikar, M. Gehrig, D. Gehrig, and D. Scaramuzza, "How to calibrate your event camera," 2021. [Online]. Available: https://arxiv.org/abs/2105.12362

[29] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI)*, 2019. [Online]. Available: http://rpg.ifi.uzh.ch/docs/TPAMI19_Rebecq.pdf

[30] R. Horaud and F. Dornaika, "Hand-eye calibration," *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, 1995.

[31] W. Bai, Y. Chen, R. Feng, and Y. Zheng, "Accurate and Efficient Frame-based Event Representation for AER Object Recognition," in *2022 International Joint Conference on Neural Networks (IJCNN)*. Padua, Italy: IEEE, July 2022, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9892070/

[32] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[34] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE, 2020, pp. 237–242.

[35] T. Lee, "Bayesian attitude estimation with the matrix fisher distribution on so (3)," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3377–3392, 2018.

[36] A. Rathinam and Y. Gao, "On-Orbit Relative Navigation Near a Known Target Using Monocular Vision and Convolutional Neural Networks for Pose Estimation," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Online, 10 2020.

[37] B. Chen, A. Parra, J. Cao, N. Li, and T.-J. Chin, "End-to-end learnable geometric vision by backpropagating PnP optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8100–8109.