2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Towards Bridging the Space Domain Gap for Satellite Pose Estimation using Event Sensing

Mohsi Jawaid[†,*], Ethan Elms[†,*], Yasir Latif[*] and Tat-Jun Chin[*]

***Abstract*— Deep models trained using synthetic data require domain adaptation to bridge the gap between the simulation and target environments. State-of-the-art domain adaptation methods often demand sufficient amounts of (unlabelled) data from the target domain. However, this need is difficult to fulfil when the target domain is an extreme environment, such as space. In this paper, our target problem is close proximity satellite pose estimation, where it is costly to obtain images of satellites from actual rendezvous missions. We demonstrate that event sensing offers a promising solution to generalise from the simulation to the target domain under stark illumination differences. Our main contribution is an event-based satellite pose estimation technique, trained purely on synthetic event data with basic data augmentation to improve robustness against practical (noisy) event sensors. Underpinning our method is a novel dataset with carefully calibrated ground truth, comprising of real event data obtained by emulating satellite rendezvous scenarios in the lab under drastic lighting conditions. Results on the dataset showed that our event-based satellite pose estimation method, trained only on synthetic data without adaptation, could generalise to the target domain effectively.**

I. Introduction

Object pose estimation is an important capability to enable intelligent robot interactions [1], [2]. Such a capability is also vital in satellite rendezvous, whereby two or more satellites come into close proximity in orbit to achieve formation flight and/or docking [3], which requires accurate estimation of relative poses between the satellites in close range [4].

Robotic vision is a promising approach for object pose estimation. State-of-the-art vision-based pose estimators employ deep learning, whereby a deep neural network (DNN) is trained to predict the pose of the object (or intermediate results such as landmark positions) in an input image [2]. Unsurprisingly, DNN-based visual pose estimation is also being considered actively for satellite rendezvous [5].

However, procuring large amounts of images from actual rendezvous missions to train DNNs is currently a major challenge. Thus, many satellite pose estimation models are trained using computer generated synthetic images [6], [7], [8]; *e.g.*, Fig. 1(a). While such an approach allows methods to be rapidly developed and benchmarked [9], [10], the resulting models cannot transfer to the target domain. To alleviate this problem, visual domain adaptation (VDA) [11], [12] is vital to bridge the simulation-to-real (Sim2Real) gap [13].

Major techniques for VDA nevertheless need target domain images, in addition to the source domain (synthetic) images [11], [12]. In the case of unsupervised VDA, *unlabelled* target domain images are required to transductively learn DNNs that can generalise to the target domain. However, as alluded to above, acquiring images from real rendezvous missions is difficult. Therefore, recent studies on VDA for satellite pose estimation [13] relied on in-laboratory physical emulation to produce target domain images, which arguably still suffers from the Sim2Real gap.

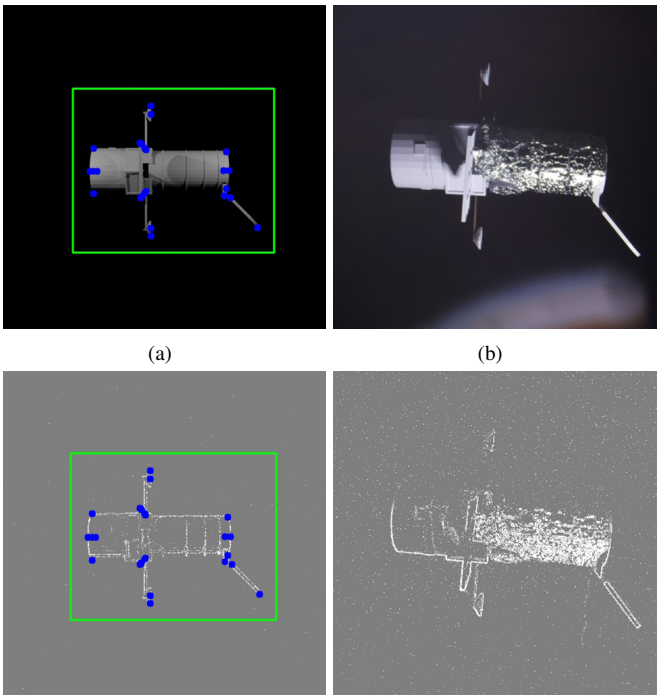

Fig. 1. (a) Rendered frame of a satellite under normal lighting, with ground truth bounding box and set of 24 landmarks. (b) Real image of a 3D model of the satellite under extreme lighting; observe the lens flare and uneven contrast. (c) Event frame corresponding to (a), generated using V2E [14]. (d) Real event frame corresponding to (b). Our premise is that the domain gap between (c) and (d) is lower than that between (a) and (b).

Being an extreme environment, the lighting conditions in space present significant challenges to robotic perception algorithms, *e.g.*, high contrast, camera over-exposure, shadowing and stray light [13]; see Fig. 1(b). Moreover, accurately emulating space lighting conditions in the lab is non-trivial, which compounds the difficulty of Sim2Real. While VDA will remain an important tool, the challenges from space motivate considering other approaches.

In this paper, we explore satellite pose estimation that is inherently robust to extreme lighting. We leverage the high dynamic range and asynchronous change detection of event sensors to perceive satellites in a way that is insensitive to

[†]Mohsi Jawaid and Ethan Elms assert equal contributions to the paper.
[*]All authors are with the Sentient Satellites Laboratory (SSL) at Australian Institute for Machine Learning (AIML), Adelaide, Australia.

drastic illuminations. We then train a satellite pose estimator from *synthetic* event data, processed from 3D rendering of the satellite under mild lighting conditions, with data augmentations to deal with noise expected in real devices. We show that the synthetically trained pose estimator can transfer well to the real domain *without* VDA, since event data is less affected by illumination changes, *cf.* Figs. 1(c) and 1(d). This helps to bridge the Sim2Real gap.

While our satellite pose estimation pipeline is not entirely novel, we are the first to investigate it on event data. Another major contribution is a calibrated event dataset [15] acquired by robotic emulation of satellite rendezvous using a state-of-the-art event camera, under carefully controlled extreme lighting. While our event dataset still suffers from the space domain gap, it nevertheless provides a strong basis to demonstrate Sim2Real transfer by our method across real and simulated domains with very different lighting conditions.

II. RELATED WORK

A. Satellite pose estimation

Estimating the 6DoF pose (position and orientation) of a target object relative to an observing platform is fundamental to robotics [2], [1]. The specific version of interest in this paper is satellite pose estimation during in-space rendezvous and docking, which underpin applications such as on-orbit refuelling, on-orbit maintenance and debris removal [16].

When the target satellite is uncooperative (*e.g.*, a defunct satellite or space debris), pose estimation must be accomplished by the chaser satellite independently. Visual pose estimation using a single camera is attractive due to its simpler design. The importance of the topic is underlined by several datasets [17], [13], [6] and challenges [9], [10], [18]. Following the success of DNN-based object pose estimation in computer vision [2], state-of-the-art monocular satellite pose estimation methods are also based on DNNs [19], [20].

B. Visual domain adaptation

VDA is crucial to enable models trained on image data from a source domain to work in a target domain [11], [12]. The difference between the two domains is called the *domain gap* [21]. Of particular interest is when the source domain is a simulation where it is possible to generate vast quantities of labelled synthetic images. The Sim2Real gap [22], [23] must be overcome using VDA to enable models trained on synthetic data to operate well in the real domain.

Addressing the Sim2Real gap is also crucial in satellite pose estimation, since many methods have been developed based on synthetic data [6], [7], [8], [19], [20]. In addition, the real space environment is affected by extreme lighting conditions [24], which is difficult to simulate virtually. A major recent effort to address Sim2Real for satellite pose estimation is by Park *et al.* [13], who developed a dataset (SPEED+) that contained synthetic training images with ground-truth pose labels as well as unlabelled real testing images captured in a lab with challenging lighting conditions. SPEED+ formed the basis for the recent Kelvins Satellite Pose Estimation Challenge 2021 [10]. However, as alluded to in Sec. I, the dataset still suffers from domain gap since the real images are not from the actual space environment.

C. Event-based vision

In the context of robotic vision, event sensors offer several advantages over RGB sensors such as higher dynamic range, μs latency, mW order power consumption and asynchronous operation while having a similar size and range. Many recent works have exploited event sensors for robotic vision tasks such as object recognition, classification, semantic segmentation, VO and SLAM [25], [26], [27], [28], [29].

Relatively less attention has been paid to pose estimation from event data. Reverter Valeiras *et al.* [30] proposed an asynchronous technique to update a given pose of an object from the event stream. Nguyen *et al.* [31] investigated camera relocalisation from event data [31]. Chen *et al.* [32] explored articulated human pose estimation. Our work differs from the above since we estimate the object pose without prior knowledge of the pose and under strong illumination effects.

In the space domain, applications of event sensors are just emerging [33]. Early works include star tracking [34] and odometry for planetary robots [27] using event sensing.

D. Event datasets

In line with the rapidly growing interest in event sensing for robotic vision, event datasets are increasingly being produced. However, many of the datasets support the tasks of object detection, object classification, VO and SLAM [35], [36], [37], [38], [39], [40]. The event dataset [15] we contribute here is one of the first on satellite pose estimation.

III. SATELLITE POSE WITH EVENT SENSING

This section describes our method of conducting satellite pose estimation using event data. Similar to the setting in [17], [13], [6], [41], [42], [7], we assume that the target satellite is uncooperative, but we have knowledge of the structure of the satellite (*e.g.*, via CAD model or SfM [43]).

Before proceeding, we emphasise that our pipeline will be trained on only synthetic data, and tested on real data without VDA. Details on our dataset and experiments will be provided respectively in Secs. IV and V.

A. From event streams to event frames

An event stream $\mathscr{E}$ has the form $\mathscr{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots\}$, where each event $\mathbf{e}_i = \{x_i, y_i, p_i, t_i\}$ is a tuple with image coordinates (x_i, y_i), polarity $p_i \in \{-1, 1\}$ and timestamp t_i. To leverage existing DNN methods for satellite pose estimation [19], [20], we convert $\mathscr{E}$ into event frames $\mathscr{F} = \{I_1, I_2, \dots\}$ via event-to-frame (E2F) conversion, where each $I_j \in [0, \Gamma]^{M \times N}$ is an image obtained from an event batch $\mathscr{B}_j \subset \mathscr{E}$ within a specific time window W_j, Γ is the highest possible intensity, and $M \times N$ is the spatial resolution of the event sensor.

Our specific E2F method creates for each input batch $\mathscr{B}_j$ a 2D histogram with $M \times N$ cells of the image coordinates (x_i, y_i) of the events in $\mathscr{B}_j$. The values of the histogram are then normalised ($\Gamma = 1$) and exported as an intensity image; see Figs. 1(c) and 1(d). An important parameter in E2F is the duration τ of each $\mathscr{B}_j$; this will be discussed in Sec. V-A.

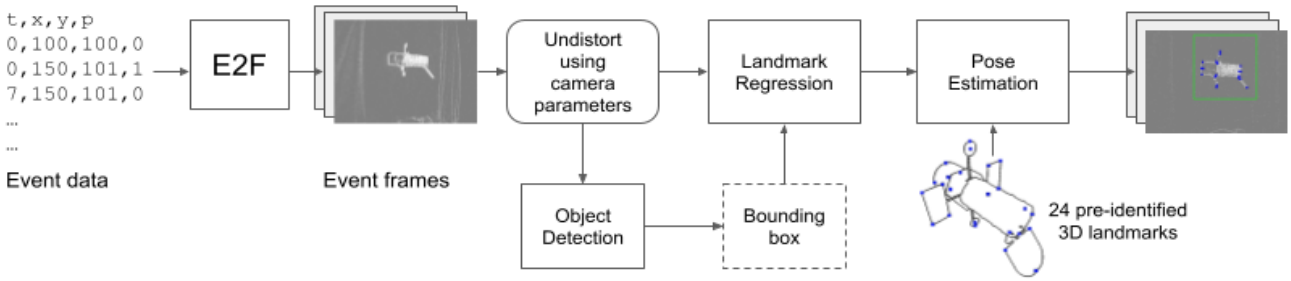

Fig. 2. Proposed pipeline for satellite pose estimation from event frames.

B. Pose estimation from event frames

Let $\mathscr{S} = \{\mathbf{P}_k\}_{k=1}^{K}$ be a 3D point cloud that defines the structure of the target satellite in a canonical reference frame. Given an input event frame I_j that observed the satellite in time window W_j, the projection of $\mathscr{S}$ onto I_j is given by

$$\tilde{\mathbf{p}}_k = \mathbf{K}[\mathbf{R}_j \ \mathbf{t}_j]\tilde{\mathbf{P}}_k, \quad k = 1, \ldots, K,$$

where $\mathbf{K}$ is a 3×3 camera intrinsics matrix [44], and $\tilde{\mathbf{p}}_k$ is $\mathbf{p}_k$ in homogeneous coordinates (similarly for $\tilde{\mathbf{P}}_k$). The rigid transformation $(\mathbf{R}_j, \mathbf{t}_j)$ defines the pose of $\mathscr{S}$ in event frame I_j, or, alternatively, during the time window W_j that generated the event batch $\mathscr{B}_j$.

Why use event frames? Strictly speaking, the notion of a "static" pose $(\mathbf{R}_j, \mathbf{t}_j)$ within a time duration W_j is imprecise due to relative motion of the camera and object in W_j; indeed, the event camera needs to move to generate events. However, W_j is usually small (*e.g.*, < 1 s) relative to the motion speed, hence $(\mathbf{R}_j, \mathbf{t}_j)$ is sufficient to describe the pose. While a fully asynchronous treatment of pose estimation is ideal, our event frame approach is sufficient for our primary aim: *demonstrate Sim2Real transfer for satellite pose estimation*.

C. Proposed method

Fig. 2 shows our pipeline, which was adapted from our winning solution [45] to the Kelvins Satellite Pose Estimation Competition 2021 [10] (`Lightbox` category). The main differences between Fig. 2 and the previous method are

- Using event frames instead of RGB frames; and
- Removing all VDA steps (*e.g.*, adversarial training) that were introduced to bridge the Sim2Real gap in [10].

Details of our method are given below.

1) Landmark selection: A small subset $\mathscr{L} = \{\mathbf{U}_z\}_{z=1}^{Z}$ of $\mathscr{S}$ that represent salient points on the surface of the object was first selected; see Fig. 2. The 2D positions of the landmarks $\mathscr{L}$ in an input event frame will be the target of prediction by the landmark regressor (see below).

2) Object detection: We used Detectron2 [46] with Faster-RCNN [47] + FPN [48] backbone for the object detection task, which was sufficiently accurate for our pipeline (single instance detection on a sparse background).

3) Landmark regression: We use HRNet [49] with 512×512 images and 128×128 heatmaps to predict the 2D positions $\{\mathbf{u}_z\}_{z=1}^{Z}$ of the landmarks $\mathscr{L}$ within the detected bounding box of the target satellite in the input event frame.

4) Pose estimation: The 2D-3D correspondences $\{(\mathbf{u}_z, \mathbf{U}_z)\}_{z=1}^{Z}$ form the input to a perspective-n-points (PnP) solver to estimate the object pose $(\mathbf{R}, \mathbf{t})$ corresponding to the input event frame. To improve robustness and accuracy, we filter the 2D-3D correspondences based on the per-point confidence score output from HRNet such that at least 15 correspondences with a score of more than a 0.95 threshold are kept. If there are not enough such points, we reduce the threshold by 20% until at least 15 points are obtained. The minimum correspondences, threshold and reduction percentage were tuned using cross validation.

5) Training data: Synthetic event frames with ground truth bounding boxes and 2D landmarks (see Sec. IV) were employed to train the object detector and landmark regressor.

6) Training: A single NVIDIA RTX 3090 24GB graphics card was used for training. For object detection we used the SGD optimizer which runs for 10000 epochs with a batch size of 10 images. An initial learning rate of 0.0001 is used with a decay factor of 0.1 after 8000 steps. HRNet was trained for 40 epochs with a batch size of 24. The `adam` [50] optimizer was used with the initial learning rate set to 0.001 with a decay factor of 0.1 after the 25th and 35th epochs.

7) Data augmentation: We subjected the training data to random rotation and random translation data augmentations. Furthermore, we also used our custom augmentations for event-frames called `RandomEventNoise` and `RandomEventLines` (see Fig. 3) to mitigate the effects of background artefacts in our real data capture environment (Sec. IV). All four augmentations were used for both the object detection and landmark regression phases.

8) Testing data: Real event data were procured to evaluate the pipeline (details in Sec. IV).

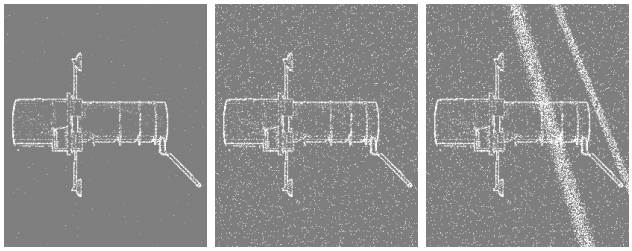

Fig. 3. Our augmentations for event frames. (Left) Event frame generated using V2E. (Middle) With `RandomEventNoise` augmentation. (Right) With `RandomEventNoise` and `RandomEventLines` augmentations.

IV. Event Dataset for Satellite Pose Estimation

To demonstrate Sim2Real transfer for satellite pose estimation using event sensing, we produced synthetic and real event datasets, which will be publicly released [15].

A. Synthetic event data

1) Simulation environment: A textureless 3D model of the Hubble Space Telescope (HST) [51] was rendered in Blender [52]. The HST was specifically chosen due to its complex structure of protruding, non-uniform elements, which creates self occlusions and shadows. The virtual camera follows a smooth trajectory while observing the HST from varying poses. Only *one* lighting condition was used, where we placed 3 point light sources around the satellite at the same height; see Fig. 4. The setup was chosen to evenly illuminate the object with mild shadows; see Fig. 1(a).

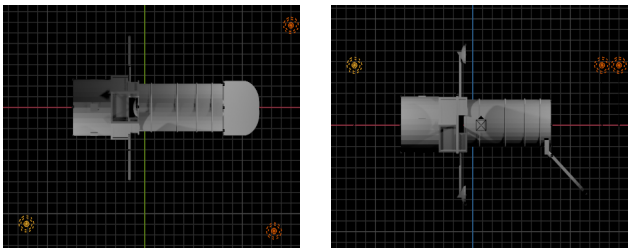

Fig. 4. Blender viewport with the HST 3D model and three point light sources (orange) for illumination. (Left) Top view and (Right) side view.

Following Sec. III, 24 salient points distributed across the 3D surface of the HST were manually chosen as the landmark set $\mathscr{L}$; see Fig. 2. Based on this setup, an RGB video with 10k frames of 640×480 pixels was rendered (the resolution matches the real event camera Inivation DVXplorer employed in the real data collection; details in Sec. IV-B.2). A high frame rate of 100 FPS was used to facilitate synthetic event generation (details in Sec. IV-A.3).

2) Ground truth: The intrinsics and pose of the virtual camera in each RGB frame in the video were exported from Blender. The image positions of the landmarks $\mathscr{L}$ were obtained by projecting their 3D coordinates onto the RGB frames. From the landmark positions, a bounding box with sufficient clearance (10% larger than the tightest bounding box on the landmarks) was defined; see Fig. 1(a).

3) Synthetic frames to events: The rendered RGB video frames were subjected to V2E [14] which employed a method similar to Katz *et al.* [53] to convert linear (0-255) intensity RGB frames to log intensity to simulate event sensors. We used a minimum timestamp resolution of 0.01 s to match the frame rate of the rendering. The high frame-rate obviated the need for additional frame interpolation in V2E.

From the generated event data, we produced event frames following the E2F procedure in Sec. III-A. To increase robustness of the trained model towards different batching durations, we used multiple batching durations τ (0.2, 0.1, 0.05 and 0.01 seconds). For each event frame, we assigned the ground truth pose, landmark positions and bounding box of the RGB frame with the closest timestamp. Fig. 1(c) shows an event frame processed from our synthetic event data.

B. Real event data

A real event dataset was captured using an event camera mounted on a UR5e robot arm to observe a printed 3D model of the HST [51]; see Fig. 5(a). Details are provided below.

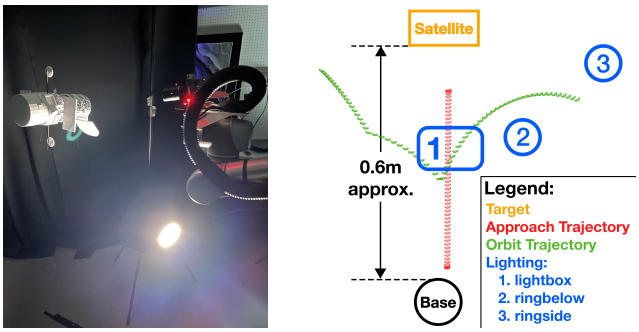

Fig. 5. (Left) Our setup for capturing real event data. Shown here is the `ringbelow` lighting configuration. (Right) Distribution of ground truth event camera poses for the `approach` (red) and `orbit` (green) trajectories with approximate light source positions and directions (not to scale).

1) Printed 3D model: To mimic uneven reflectance and texture on the real HST, we partially wrapped our model with aluminium foil to emulate a thermal blanket; see Fig. 1(b).

2) Event camera: We employed an Inivation DVXplorer event camera (comparable to the DAVIS240C recently launched in space [54]) with a Arducam 4-12mm Varifocal C-Mount lens which has a manually adjustable focal length and aperture. To focus the event camera, it was placed at the initial pose of the approach trajectory with a rotating Siemens Star placed near the satellite. The focus on the lens was then adjusted manually until the pattern was sharp.

3) Event camera calibration: We adapted the intrinsic and extrinsic (eye-in-hand) calibration techniques for RGB cameras [55], [56], as implemented in OpenCV [57]. First, we executed a semi-spherical trajectory on the UR5e arm, whilst carrying the event camera to observe a checkerboard on an LED screen. The observed events and the ground truth robot arm poses (polled at 10 Hz using the manufacturer's API) served as inputs to the calibration. We accumulated the events into event frames using the Inivation DV SDK's accumulation frame module, then ran chessboard corner detection. The results were passed to `calibrateCamera`, which yielded the camera intrinsics. Using the intrinsics, the chessboard to camera transformation was estimated using `solvePnP`, which was then used in `calibrateHandEye` as the `target2cam` input transformation. See [15] for detailed information of the calibration.

4) Rendezvous trajectories: Two trajectories named `approach` and `orbit` were executed; see Fig 5(b). The first one mimics a docking procedure with the satellite, where the camera makes a direct, linear approach towards the satellite; the second mimics an orbit around the satellite. Two different speeds (`slow` and `fast`) were also activated for each trajectory. Specifically, `approach-slow` at 0.0332 m/s, `orbit-slow` at 0.0142 m/s, `approach-fast` at 0.2186 m/s, and `orbit-fast` at 0.3007 m/s.

5) Lighting conditions: 5 lighting configurations were tested: `ambient` was the standard indoor lights of our lab, which illuminate the entire scene evenly. A 9300 Lumens light, combined with a custom light diffuser, was used to create the `lightbox` lighting. The `centre` lighting used a portion of the lab lighting array, with the intention to provide a dim illumination of the entire scene. Lastly, two small ring-shaped lights were employed for the `ringside` and `ringbelow` lightings, where the lights were situated to the right side and below the approach trajectory, respectively.

The location of the light sources relative to the camera motion and target object are shown in Fig. 5(b), while Fig. 6 illustrates the lighting effects via RGB still images.

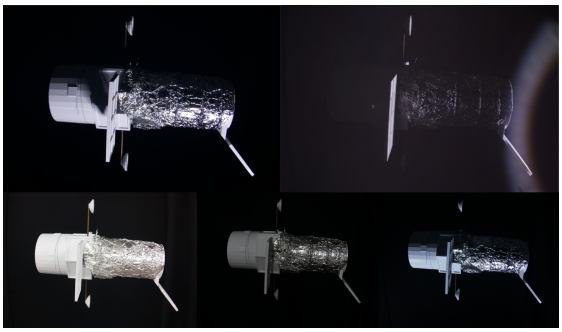

Fig. 6. Lighting effects on satellite model (from top to bottom, left to right: `ringbelow`, `ringside`, `ambient`, `centre`, `lightbox`).

C. Statistics of the real event dataset

All combinations of trajectory type, speed and lighting configuration were enumerated for capture. The capture output for each sequence is an event stream $\mathscr{E}$ and ground truth camera poses $\{(\boldsymbol{\Omega}_\ell, \boldsymbol{\pi}_\ell)\}_{\ell=1}^{L}$ at 10 Hz. Fig. 7 illustrates real event frames from our dataset, while Table I provides an overview of the collected real event data.

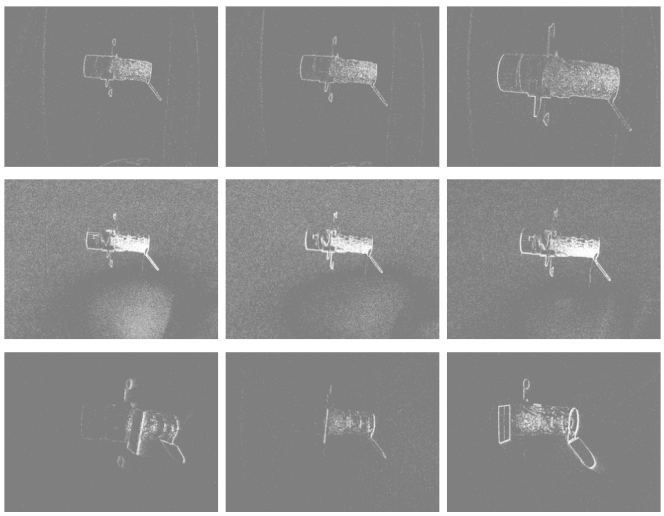

Fig. 7. Sample real event frames from our dataset. `approach-slow-ambient` (top), `approach-fast-ringbelow` (middle) and `orbit-fast-ringside` (bottom). Note that the vast lighting differences did not produce noticeable effects in the data, *cf.* Fig. 6.

Our dataset for **Spacecraft posE Estimation with Neuromorphic vision (SEENIC)**, has been released publicly [15].

Trajectory	Scene	Events	Time (s)	Cam. poses
approach	fast-ambient	1.7 M	2.40	23
	fast-centre	1.5 M	2.40	23
	fast-lightbox	7.5 M	2.41	23
	fast-ringbelow	4.1 M	2.40	23
	fast-ringside	1.6 M	2.40	23
	slow-ambient	6.3 M	14.73	146
	slow-centre	5.8 M	14.73	146
	slow-lightbox	43.3 M	14.73	146
	slow-ringbelow	10.3 M	14.72	146
	slow-ringside	3.9 M	14.73	146
orbit	fast-ambient	6.5 M	4.31	42
	fast-centre	6.5 M	4.31	42
	fast-lightbox	17.1 M	4.31	42
	fast-ringbelow	11.8 M	4.31	42
	fast-ringside	4.7 M	4.31	42
	slow-ambient	39.2 M	87.48	872
	slow-centre	31.9 M	87.46	872
	slow-lightbox	276.7 M	88.74	872
	slow-ringbelow	52.7 M	87.48	872
	slow-ringside	16.3 M	87.48	872

TABLE I

BASIC STATISTICS OF OUR REAL EVENT DATASET (M = MILLION).

V. RESULTS

A. Hyperparameters

The settings of the important hyperparameters of our method were as follows:

- Batch duration $\tau = 0.05$ s for the `approach` scenarios, 0.2 s for the `orbit-slow` scenarios and 0.01 s for the `orbit-fast` scenarios. These values were selected to optimise the signal-to-noise ratio in the event frames.
- $Z = 24$ points on the surface of HST that cover unique structures such as the hatch, solar panels and dishes were selected as the landmark set $\mathscr{L}$; see Fig. 2.

B. Qualitative results

Fig. 8 shows sample outputs of our method on real data (Sec. IV-B), where the method was trained on only synthetic data (Sec. IV-A). Generally the position estimates are better than the orientation estimates; the error in the latter likely due to slight misalignments with the solar panels.

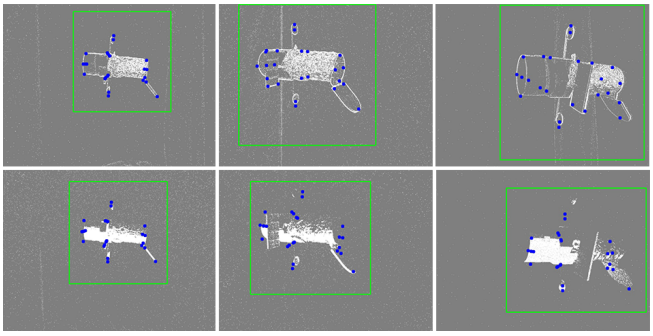

Fig. 8. Sample of real event frames with the predicted object detection bounding box in green and 3D landmarks reprojected using the estimated pose in blue. In each row we show a frame from the `approach-slow`, `orbit-slow` and `orbit-fast` scenes respectively in different lighting. `ambient` (top) `lightbox` (bottom).

C. Quantitative results

Each real event stream $\mathscr{E}$ was converted into event frames $\{I_1, I_2, \dots\}$ following Sec. III-A. Then, each ground truth camera pose $(\boldsymbol{\Omega}_\ell, \boldsymbol{\pi}_\ell)$ for $\mathscr{E}$ was paired with the event frame I_j that was closest in time. The trained model was executed on I_j to estimate the object pose $(\mathbf{R}_j, \mathbf{t}_j)$. The process created L pairs of poses $\{(\boldsymbol{\Omega}_\ell, \boldsymbol{\pi}_\ell)\}_{\ell=1}^{L}$ and $\{(\mathbf{R}_\ell, \mathbf{t}_\ell)\}_{\ell=1}^{L}$.

To measure the accuracy of object pose estimation while mitigating any inaccuracies due to the extrinsic calibration, we computed the relative trans. between successive poses

$$(\mathbf{R}_\ell^{rel}, \mathbf{t}_\ell^{rel}) = \texttt{rel}\left((\mathbf{R}_\ell, \mathbf{t}_\ell), (\mathbf{R}_{\ell+1}, \mathbf{t}_{\ell+1})\right),$$
$$(\boldsymbol{\Omega}_\ell^{rel}, \boldsymbol{\pi}_\ell^{rel}) = \texttt{rel}\left((\boldsymbol{\Omega}_\ell, \boldsymbol{\pi}_\ell), (\boldsymbol{\Omega}_{\ell+1}, \boldsymbol{\pi}_{\ell+1})\right).$$

Following [13], we compute the error of the object pose estimation via the error in the relative transformation, *i.e.*,

$$\phi_\ell = \left\| \mathbf{t}_\ell^{rel} - \boldsymbol{\pi}_\ell^{rel} \right\|_2, \qquad \psi_\ell = 2\arccos\left(\left| < \mathbf{q}_\ell^{rel}, \hat{\mathbf{q}}_\ell^{rel} > \right| \right),$$

which resp. measure translation error (in meters) and rotation error (in degrees), and $\boldsymbol{q}_\ell^{rel}$ and $\hat{\boldsymbol{q}}_\ell^{rel}$ are the quaternion form of $\mathbf{R}_\ell^{rel}$ and $\boldsymbol{\Omega}_\ell^{rel}$. The overall error for $\mathscr{E}$ is

$$\Phi = \sqrt{\frac{1}{L-1}\sum_{\ell=1}^{L-1}\phi_\ell^2}, \qquad \Psi = \frac{1}{L-1}\sum_{\ell=1}^{L-1}\psi_\ell,$$

which are also in meters and degrees, respectively.

Fig. 9 shows the errors as a function of time of the pipeline, trained with augmentations, on 3 sequences with the best, median and worst Φ. Table II shows the overall results, which indicate excellent performance of our method since the errors were maintained at 10's of centimeters and 2-3 degrees, comparable to the top RGB methods [9], [10].

		W/O Aug.		W Aug.	
Trajectory	Scene	Φ (m)	Ψ (°)	Φ (m)	Ψ (°)
approach	fast-ambient	0.167	3.011	**0.157**	**2.090**
	fast-centre	0.166	2.203	**0.130**	**2.082**
	fast-lightbox	0.166	**2.923**	**0.123**	3.156
	fast-ringbelow	0.168	3.054	**0.123**	**2.449**
	fast-ringside	0.153	**3.044**	**0.131**	3.143
	slow-ambient	0.155	2.166	**0.055**	**2.075**
	slow-centre	0.155	**3.135**	**0.075**	3.162
	slow-lightbox	0.154	2.974	**0.082**	**2.086**
	slow-ringbelow	0.155	**2.800**	**0.057**	3.140
	slow-ringside	0.155	2.404	**0.116**	**2.091**
orbit	fast-ambient	0.232	3.289	**0.187**	**2.109**
	fast-centre	**0.189**	2.142	0.200	**2.118**
	fast-lightbox	0.226	3.103	**0.197**	**2.096**
	fast-ringbelow	0.231	2.189	**0.195**	**2.109**
	fast-ringside	0.232	3.258	**0.207**	**2.117**
	slow-ambient	0.177	3.128	**0.113**	**2.098**
	slow-centre	0.222	3.095	**0.119**	**2.096**
	slow-lightbox	0.225	3.074	**0.202**	**2.112**
	slow-ringbelow	0.222	3.019	**0.124**	**2.098**
	slow-ringside	0.223	3.041	**0.131**	**2.100**

TABLE II

ABLATION STUDY SHOWING Φ AND Ψ W & W/O AUGMENTATIONS.

The method was challenged by outlier event frames with low signal-to-noise ratio causing the overall errors in Table. II to be significantly higher than the individual errors

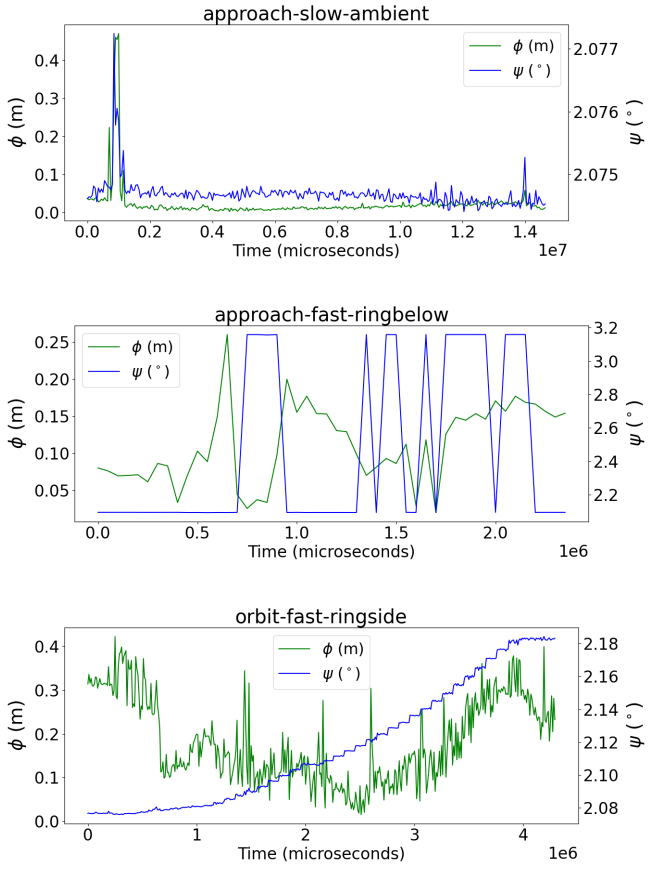

Fig. 9. Relative transformation errors (ϕ_ℓ and ψ_ℓ) over time for several real event sequences from our dataset.

in most event frames. We can see this at the start of `approach-slow-ambient`, when the motion was slow relative to the chosen batching duration τ. Currently τ was manually set and fixed for each $\mathscr{E}$. Automatically adjusting τ will be a fruitful research direction, *e.g.*, [29]. Slight differences in the synthetic and real satellite model such as the bottom dish location also affected the pose estimation.

D. Ablation studies

Our ablation results in Table. II show that the results are generally better in the pipeline trained with augmentations.

VI. CONCLUSIONS

We proposed event sensing to surmount the Sim2Real gap for satellite pose estimation. A novel event dataset with accurate ground truth was constructed to demonstrate the technique. Results show that the idea is promising. Further work will need to be done to increase the scope and breadth of the dataset, as well as raising the robustness of the method towards hyperparameter settings such as batch duration. We hope that this work drives the adoption and use of event sensors for satellite pose estimation.

ACKNOWLEDGEMENTS

Mohsi Jawaid acknowledges support from Poppy@AIML. Ethan Elms acknowledges support from Northrop Grumman. Tat-Jun Chin is SmartSat CRC Chair of Sentient Satellites.

REFERENCES

[1] C. Sahin, G. Garcia-Hernando, J. Sock, and T.-K. Kim, "A review on object pose recovery: from 3d bounding box detectors to full 6d pose estimators," *Image and Vision Computing*, vol. 96, p. 103898, 2020.

[2] Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, and J. He, "Deep learning on monocular object pose detection and tracking: A comprehensive overview," *ACM Computing Surveys (CSUR)*, 2021.

[3] D. Zimpfer, P. Kachmar, and S. Tuohy, "Autonomous rendezvous, capture and in-space assembly: past, present and future," in *1st Space exploration conference: continuing the voyage of discovery*, 2005, p. 2523.

[4] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations," *Progress in Aerospace Sciences*, vol. 93, pp. 53–72, 2017.

[5] 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.

[6] 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.

[7] M. A. Musallam, K. A. Ismaeil, O. Oyedotun, M. D. Perez, M. Poucet, and D. Aouada, "Spark: spacecraft recognition leveraging knowledge of space environment," *arXiv preprint arXiv:2104.05978*, 2021.

[8] H. A. Dung, B. Chen, and T.-J. Chin, "A spacecraft dataset for detection, segmentation and parts recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2012–2019.

[9] "Kelvins satellite pose estimation competition 2019." [Online]. Available: https://kelvins.esa.int/satellite-pose-estimation-challenge/challenge/

[10] "Kelvino satellite pose estimation competition 2021." [Online]. Available: https://kelvins.esa.int/pose-estimation-2021/challenge/

[11] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218306684

[12] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko, "Visda: A synthetic-to-real benchmark for visual domain adaptation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 2102–21 025.

[13] T. H. Park, M. Märtens, G. Lecuyer, D. Izzo, and S. D'Amico, "SPEED+: Next-generation dataset for spacecraft pose estimation across domain gap," *arXiv preprint arXiv:2110.03101*, 2021.

[14] Y. Hu, S.-C. Liu, and T. Delbruck, "v2e: From video frames to realistic dvs events," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1312–1321.

[15] E. Elms, M. Jawaid, Y. Latif, and T.-J. Chin, "SEENIC: Dataset for spacecraft pose estimation with neuromorphic vision," https://github.com/0thane/SEENIC.

[16] M. A. Shoemaker, M. Vavrina, D. E. Gaylor, R. Mcintosh, M. Volle, and J. Jacobsohn, "OSAM-1 decommissioning orbit design," in *AAS/AIAA Astrodynamics Specialis t Conference*, 2020.

[17] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, 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.

[18] "Spark challenge : Spacecraft recognition leveraging knowledge of space environment." [Online]. Available: https://cvi2.uni.lu/spark-2021/

[19] S. Sharma and S. D'Amico, "Pose estimation for non-cooperative rendezvous using neural networks," *arXiv preprint arXiv:1906.09868*, 2019.

[20] 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, pp. 0–0.

[21] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv preprint arXiv:1702.05374*, 2017.

[22] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard, *et al.*, "Sim2real in robotics and automation: Applications and challenges," *IEEE transactions on automation science and engineering*, vol. 18, no. 2, pp. 398–400, 2021.

[23] X. Peng, B. Usman, K. Saito, N. Kaushik, J. Hoffman, and K. Saenko, "Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation," *arXiv preprint arXiv:1806.09755*, 2018.

[24] T. H. Park and S. D'Amico, "Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap," 2022. [Online]. Available: https://arxiv.org/abs/2203.04275

[25] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 639–16 652, 2020.

[26] F. J. Martín Ameneiro, "Evaluation of deep learning-based classification and object detection algorithms for event cameras," Master's thesis, 2021.

[27] F. Mahlknecht, D. Gehrig, J. Nash, F. M. Rockenbauer, B. Morrell, J. Delaune, and D. Scaramuzza, "Exploring event camera-based odometry for planetary robots," *arXiv preprint arXiv:2204.05880*, 2022.

[28] S. Jia, "Event camera survey and extension application to semantic segmentation," in *2022 4th International Conference on Image Processing and Machine Vision (IPMV)*, 2022, pp. 115–121.

[29] K. Xiao, G. Wang, Y. Chen, Y. Xie, H. Li, and S. Li, "Research on event accumulator settings for event-based slam," in *2022 6th International Conference on Robotics, Control and Automation (ICRCA)*. IEEE, 2022, pp. 50–56.

[30] D. Reverter Valeiras, G. Orchard, S.-H. Ieng, and R. B. Benosman, "Neuromorphic event-based 3d pose estimation," *Frontiers in neuroscience*, vol. 9, p. 522, 2016.

[31] A. Nguyen, T.-T. Do, D. G. Caldwell, and N. G. Tsagarakis, "Real-time pose estimation for event cameras with stacked spatial lstm networks," *arXiv preprint arXiv:1708.09011*, vol. 3, 2017.

[32] J. Chen, H. Shi, Y. Ye, K. Yang, L. Sun, and K. Wang, "Efficient human pose estimation via 3d event point cloud," *arXiv preprint arXiv:2206.04511*, 2022.

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

[34] T.-J. Chin, S. Bagchi, A. Eriksson, and A. Van Schaik, "Star tracking using an event camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[35] A. Mitrokhin, C. Fermuller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," *arXiv preprint arXiv:1803.04523*, 2018.

[36] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: histograms of averaged time surfaces for robust event-based object classification," *CoRR*, vol. abs/1803.07913, 2018. [Online]. Available: http://arxiv.org/abs/1803.07913

[37] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, "Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

[38] E. Mueggler, H. Rebecq, G. Gallego, T. Delbrück, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *CoRR*, vol. abs/1610.08336, 2016. [Online]. Available: http://arxiv.org/abs/1610.08336

[39] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3586–3595.

[40] ——, "Video to events: Bringing modern computer vision closer to event cameras," *arXiv Preprint*, vol. 2, 2019.

[41] Y. Hu, S. Speierer, W. Jakob, P. Fua, and M. Salzmann, "Wide-depth-range 6d object pose estimation in space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 870–15 879.

[42] L. Pasqualetto Cassinis, R. Fonod, E. Gill, I. Ahrns, and J. Gil Fernandez, "Cnn-based pose estimation system for close-proximity operations around uncooperative spacecraft," in *AIAA Scitech 2020 Forum*, 2020, p. 1457.

[43] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.

[44] R. Hartley and A. Zisserman, "Camera models," *Multiple View Geometry in Computer Vision*, vol. 2, 2003.

[45] "Kelvins spec2021 lightbox final result (closed)." [Online]. Available: https://kelvins.esa.int/pose-estimation-2021/leaderboard/lightbox-final-result

[46] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[47] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.
[48] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[49] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[51] E. Summers and M. Horn, "Hubble space telescope," https://nasa3d.arc.nasa.gov/detail/HST.
[52] https://www.blender.org/.
[53] M. L. Katz, K. Nikolic, and T. Delbruck, "Live demonstration: Behavioural emulation of event-based vision sensors," in *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2012, pp. 736–740.
[54] Inivation, "World's first neuromorphic sensor launched to space," https://inivation.com/worlds-first-neuromorphic-sensor-launched-to-space/, 2021.
[55] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989.
[56] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, p. 1330–1334, 2000.
[57] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.