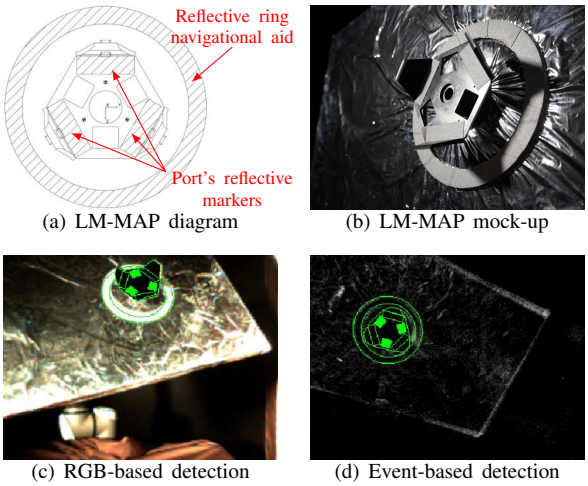
Mixing Data-driven and Geometric Models for Satellite Docking Port State Estimation using an RGB or Event Camera

Cedric Le Gentil[1], Jack Naylor[2], Nuwan Munasinghe[1], Jasprabhjit Mehami[2], Benny Dai[1], Mikhail Asavkin[3], Donald G. Dansereau[2], Teresa Vidal-Calleja[1]

***Abstract*— In-orbit automated servicing is a promising path towards lowering the cost of satellite operations and reducing the amount of orbital debris. For this purpose, we present a pipeline for automated satellite docking port detection and state estimation using monocular vision data from standard RGB sensing or an event camera. Rather than taking snapshots of the environment, an event camera has independent pixels that asynchronously respond to light changes, offering advantages such as high dynamic range, low power consumption and latency, etc. This work focuses on satellite-agnostic operations (only a geometric knowledge of the actual port is required) using the recently released Lockheed Martin Mission Augmentation Port (LM-MAP) as the target. By leveraging shallow data-driven techniques to preprocess the incoming data to highlight the LM-MAP's reflective navigational aids and then using basic geometric models for state estimation, we present a lightweight and data-efficient pipeline that can be used independently with either RGB or event cameras. We demonstrate the soundness of the pipeline and perform a quantitative comparison of the two modalities based on data collected with a photometrically accurate test bench that includes a robotic arm to simulate the target satellite's uncontrolled motion.**



(a) LM-MAP diagram (b) LM-MAP mock-up

(c) RGB-based detection (d) Event-based detection

Fig. 1. The proposed method performs the detection and state estimation of the Lockheed Martin Mission Augmentation Port (LM-MAP) ((a) and (b)) using standard RGB images (c) or event-based data (d).

I. Introduction

Satellite operations support a wide range of infrastructure essential to today's society. First used as scientific, technological, and military demonstrators during the Cold War, satellites are now part of our daily life (e.g., telecommunication, GPS, etc) and are shown to be valuable tools for environment monitoring especially in the effort to fight climate change [1]. Unfortunately, the growing number of satellites in orbit comes with logistic and obsolescence problems given the accumulation of debris and decommissioned satellites. Along with the financial incentive to reduce the number of rocket launches, there is a growing interest for in-orbit maintenance of satellites to extend their lifetime and avoid cluttering orbits. In 2022, Lockheed Martin released the specification of a docking port [2] to standardise and facilitate in-orbit servicing. In this paper, we work towards the adoption of such a standard by addressing the issue of the Lockheed Martin Mission Augmentation Port (LM-MAP) detection and state estimation for autonomous docking operation as illustrated in Fig. 1.

*This work was supported by NSW Space Research Network, grant ID RP220201

[1]Robotics Institute, University of Technology Sydney, Australia.
[2]Australian Centre for Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, Australia.
[3]ANT61, Australia.
Corresponding author: cedric.legentil@uts.edu.au

While some systems are tested in space [3], [4], algorithmic breakthroughs for autonomous docking or in-orbit rendezvous heavily rely on the development of software [5] and physical [6], [7] simulators here on Earth. In [8] the authors focus on camera-based docking port detection and localisation with a large-scale (20m) physical simulator that consists in two industrial robotic arms and a rail system. In previous work [9], we presented our photometrically accurate real-world satellite-docking simulator to allow for the design of novel vision-based perception algorithms. This test bench allows us to compare the performance of the proposed algorithm with both RGB and event cameras.

Event cameras, also called neuromorphic cameras [10], introduced a novel way to acquire "visual" data. Unlike traditional cameras where all the pixels are triggered simultaneously to get a snapshot of the environment, the pixels of an event camera independently trigger events when the level of light changes in that pixel. Accordingly, the output of an event camera consists of an asynchronous stream of events (timestamp, x and y pixel location, and direction of change) that display interesting properties such as low latency, High-Dynamic Range (HDR), absence of motion blur, etc. Naturally, the robotics community has developed an increasing number of algorithms over the past decade to leverage event vision in various applications [11], with examples for keypoint detection [12], tracking [13], [14], [15], odometry [16], [17], and SLAM [18].

While the state of event-based research is not as mature as the standard camera counterpart, there has already been a push toward space-oriented uses of event vision. For example, in [19], [20], [21], and [22] the focus is the detection and tracking of point object/celestial bodies. Closer to the proposed work, [23] proposes an approach to perform satellite state estimation using neural networks to extract keypoints that can later be associated with their corresponding vertices in the Computer-Aided Design (CAD) model. The focus in [23] is closing the domain gap between simulated and real data. Reducing the "sim-to-real" gap allows for network training with large amount of data without the caveat of collecting real-world data.

Outside of event-only spacecraft detection and localisation, numerous works are based on deep learning as discussed in a recent survey [24]. Such methods are specifically trained for particular models of satellite like in [23], where the satellite's CAD model is required during training. Thus, they do not generalise to new targets and require retraining. Also, this might not be compatible with certain servicing or deorbiting missions due to alterations or damages to the satellite during its years of service (potential difference between the CAD model and the real satellite). Most deep learning approaches in [24] rely on network models with millions of parameters possibly making it difficult to deploy for real-time computation on embedded hardware. Another interesting approach is the online supervision in [25] with an adaptive Kalman filter to learn the network parameters while performing the satellite approach manoeuvres. However, this method also requires a prior CAD model of the spacecraft.

In this paper, we propose a LM-MAP detection and monocular state estimation framework that can handle either standard camera images or event data. The core principle of our approach is the combination of data-driven and basic geometric models for data-efficient lightweight estimation. The data-driven component highlights key features of the LM-MAP that are the navigation ring and the port reflectors as seen in Fig. 1. Then, simple geometric models with a RANndom SAmple Consensus (RANSAC) approach allow for 6-Degree of Freedom (DoF) estimation of the port's pose. The contributions of this work are the design and implementation of a satellite-agnostic docking port detection and localisation algorithm, the evaluation of the proposed framework with both event and RGB data, and the open-source release of the datasets used in the evaluation.

II. Method

A. Overview

Let us consider a camera (RGB or event-based) and a LM-MAP moving freely in space. The proposed pipeline aims at detecting the LM-MAP and estimating its 6-DoF pose in the camera reference frame. As illustrated in Fig. 2, it relies on a combination of data-driven and model-based techniques and can handle either event or RGB data sources independently: Convolutional Neural Networks (CNNs) are used to filter the camera data by highlighting key components

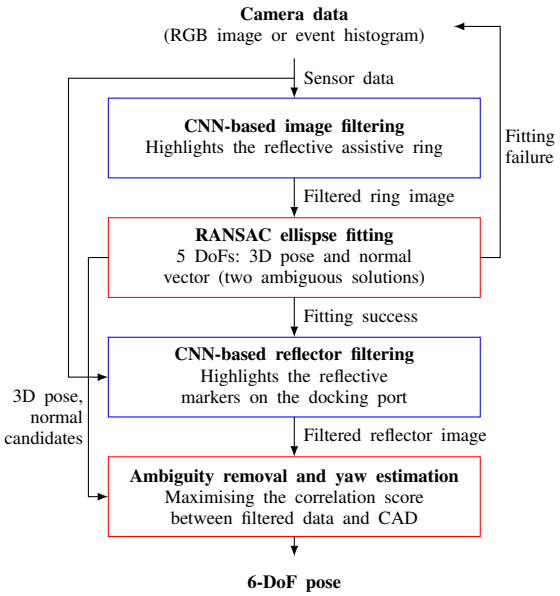


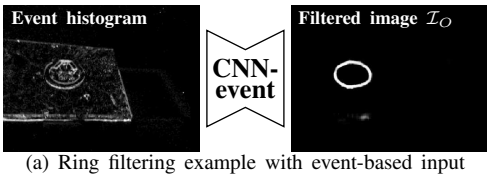
Fig. 2. Diagram overview of the proposed pipeline for satellite docking port detection and state estimation. The blue blocks are built upon data-driven techniques while the red block correspond to geometry-based algorithms.

of the LM-MAP before fitting simple geometric models to perform pose estimation.

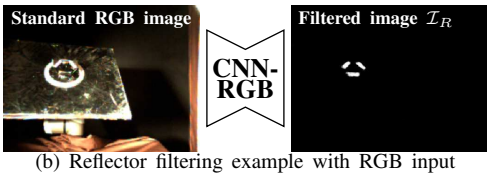
Concretely, images are collected with the camera either as RGB images or as the accumulation of N events in an image-like histogram. The sensor data is passed through a *ring filter* CNN that highlights the reflective assistive ring present around the LM-MAP as shown in Fig. 3(a). By binarising and skeletonising the filtered image, an ellipse is fitted to the pre-processed sensor data. Analysing the geometric characteristics of the ellipse given the actual size of the ring allows for the estimation of 5 DoFs of the LM-MAP pose: the 3D position in the camera frame and the direction of the normal vector of the LM-MAP. Note that the 2 DoFs of the normal vector also present an ambiguity as two different vectors can explain the observed ellipse. To solve for the ambiguity and estimate the last DoF of the LM-MAP's pose, a second CNN is used to highlight the three reflective rectangular markers present on the surface of the LM-MAP. By computing a correlation score between the filtered image and the projection of a simplistic CAD model of the LM-MAP, the optimum "yaw" angle around the normal vector is determined.

B. CNN image filtering

The proposed pipeline can be used either with a single RGB or event camera. While RGB cameras directly provide images, the event data stream is not an image-like representation suitable for a standard CNN input. Accordingly, when using an event camera, the raw stream is converted into a succession of frames that are image-like histograms in which each bin corresponds to a pixel, and is populated with the events that occurred at this location (regardless of their polarity). A total of N consecutive events are used to generate the histograms (N=35k in our implementation).



(a) Ring filtering example with event-based input



(b) Reflector filtering example with RGB input

Fig. 3. Illustration of the ring and reflector CNN-based filters. (a) shows a ring filtering example with event-based input (N = 35k). (b) shows the reflector filtering with RGB data.

Fig. 3(a) provides an example of such a histogram.

The filtering of the assistive navigational ring and the three reflective markers rely on a single CNN architecture. Similarly, the architecture is the same for RGB and event-based sensing except for the first layer which has three channels for RGB data as opposed to only one for event-based data. As per the goal of performing satellite-agnostic LM-MAP detection, the proposed pipeline leverages a U-Net-like [26] CNN to infer mask-like images that highlight the appropriate LM-MAP features (navigation ring or reflectors) as illustrated in Fig. 3. The chosen architecture consists of three consecutive convolution layers with ReLU activation functions and max-pooling, followed by three deconvolution layers with ReLU activation. The one-channel output image has the same size as the input image and is passed through a sigmoid function to provide a mask with values between zero and one. We denote $\mathcal{I}_O$ the output of the ring filter and $\mathcal{I}_R$ the one of the reflector filter.

The lack of fully connected layers implies that only local information is used in the filter output. Accordingly, the proposed network cannot learn the ring position by "recognising" a certain satellite's side panel and infer the LM-MAP position relative to the in-built knowledge of the specific satellite. By only considering local information our approach is truly satellite-agnostic and avoids the risk of overfitting to a closed set of satellites.

C. Ellipse fitting and 5-DoF estimation

As shown in Fig. 3, the proposed pipeline leverages the reflective ring (navigational aid) that is present around the LM-MAP to estimate 5 of the 6 DoFs of the LM-MAP pose. Based on the standard pinhole camera model, the projection of a 3D circle on the image plane almost matches the shape of an ellipse. Thus, we propose to use the simple conic section implicit representation

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \qquad (1)$$

in 2D to approximate the ring projection in the image and subsequently perform the 5-DoF estimation of the LM-MAP's state.

Concretely, given the ring-filtered image $\mathcal{I}_O$ we apply binarisation and skeletonisation [27] to obtain a one-pixel-wide representation of the scene that we denote $\mathcal{I}_S$. The absence of a ring in the current view results in a very low number of active pixels in $\mathcal{I}_S$. Accordingly, with a threshold γ_S, the estimation process is aborted if $\mathcal{I}_S < \gamma_S$. Otherwise, the active pixels are converted into 2D points $\mathbf{X} \in \mathbb{R}^{2\times N}$, based on their image coordinates, and used in a RANSAC-based ellipse fitting algorithm to estimate the parameters $\mathbf{e}_h = \begin{bmatrix} A & B & C & D & E \end{bmatrix}$ in (1). Note that (1) possesses six parameters while an ellipse only has five DoFs. Thus, the value of F is set as a constant equal to one. The RANSAC process consists in selecting five points $\mathbf{X}_h \in \mathbb{R}^{2\times 5}$ and estimating the corresponding hypothetical ellipse parameters $\mathbf{e}_h$ by solving a linear problem. From (1), the conic representation can be converted into a minor/major axis ellipse representation. After a sanity check on the ratio between the minor and major axis of the current hypothesis, the set of inliers from $\mathbf{X}$ is computed. The RANSAC process is repeated H times or until the number of inliers reaches the number of active pixels in $\mathcal{I}_s$. Eventually, the hypothesis associated with the most inliers is refined using all the inliers and selected for the rest of the estimation process.

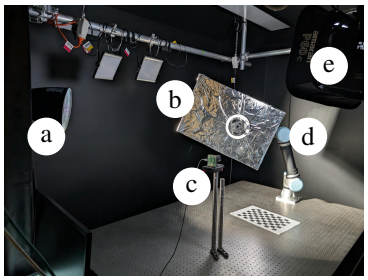
Based on the extremities of the ellipse's major axis $\mathbf{x}_{M1}$ and $\mathbf{x}_{M2}$ in the image, the 3D position $\mathbf{p}$ of the LM-MAP is estimated as the center of the ring

$$\mathbf{p} = \frac{\nu\,(\mathbf{v}_{M1} + \mathbf{v}_{M2})}{\|\mathbf{v}_{M2} - \mathbf{v}_{M1}\|}, \text{ with } \mathbf{v}_\bullet = \frac{\mathbf{K}^{-1}[\,\mathbf{x}_\bullet\,]}{\|\mathbf{K}^{-1}[\,\mathbf{x}_\bullet\,]\|}, \qquad (2)$$

$\mathbf{K}$ the camera intrinsics matrix, and ν the actual ring radius. To estimate the normal vector of the LM-MAP, we first need to find the 3D position $\hat{\mathbf{x}}_{m1}$ and $\hat{\mathbf{x}}_{m2}$ of the ellipse's minor axis extremities. Defining $\hat{\mathbf{x}}_{m\bullet} = d_{m\bullet}\mathbf{v}_{m\bullet}$, with $d_{m\bullet}$ the distance between the camera and $\hat{\mathbf{x}}_{m\bullet}$, allows for the computation of $\hat{\mathbf{x}}_{m\bullet}$ by solving the quadratic problem $\nu^2 = \|d_{m\bullet}\mathbf{v}_{m\bullet} - \mathbf{p}\|^2$. Note that there are two solutions for each of the extremities of the minor axis, one being closer to the camera than $\mathbf{p}$ and one further. However, in practice, the ring configuration using the furthest solution with $\mathbf{x}_{m1}$ or the closest solution with $\mathbf{x}_{m2}$ are very similar. Thus, we only keep the solutions with $\hat{\mathbf{x}}_{m\bullet}$ between the camera position and the ring centre $\mathbf{p}$. Eventually, these correspond to two different normal vector estimates defined as the cross product between the ellipse's major and minor axis in 3D.

D. Yaw estimation

Given the aforementioned 5-DoF estimates, the reflector-filtered image $\mathcal{I}_R$, and a simple CAD model of the LM-MAP, we remove the "two-normal" ambiguity and estimate the last DoF of the LM-MAP's pose by maximising a correlation score between $\mathcal{I}_r$ and the hypothetical projection of the LM-MAP in the camera frame. Formally, assuming an orientation $\mathbf{R}$ (rotation matrix) and previously estimated position $\mathbf{p}$, a mask-like image $\mathbf{I}_M = \pi(\mathbf{R}, \mathbf{p})$ is created by projecting the CAD's reflective markers on the image plane.



a Aputure LS1200D sunshine light
b Satellite mock-up with docking port
c Davis 346 RGB/event camera
d UR5e robotic arm
e Aputure P60C earthshine light

Fig. 4. Photometrically accurate low earth orbit bench for satellite docking.

Maximising the similarity between $\mathcal{I}_R$ and $\mathcal{I}_M$

$$\mathbf{R}^* = \underset{\mathbf{R}}{\operatorname{argmin}} \sum_{u,v} \pi(\mathbf{R}, \mathbf{p})_{[u,v]} \mathcal{I}_{R_{[u,v]}}, \tag{3}$$

provides the final orientation estimate. Thanks to the prior normal vector estimate, the rotation-based maximisation problem is reduced to two 1-DoF problems, where the only unknown is the amount of rotation around the normal vectors. The low dimension of the search space and the LM-MAP axial symmetry allow for the use of a simple grid search between 0 and 120° (increments of 1° in our implementation) to solve (3).

III. Experimental Setup and Data Generation

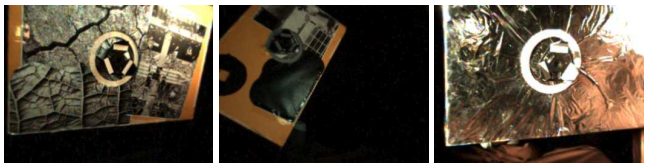
A. Test bench setup

Obtaining data and validating the proposed pipeline on a real satellite in orbit is not practical. To address this issue, we have developed a photometrically accurate physical simulator [9] as shown in Fig. 4. The setup consists of a satellite mock-up mounted on a robotic arm in a dark room with walls painted in black. The room is equipped with various light sources to emulate the illumination conditions of Low Earth Orbit (LEO). One or multiple cameras can be mounted in the environment with an optional colocated light source. The motion of the robotic arm simulates free-floating relative motion between a servicing satellite (camera) and a target satellite (mock-up with scaled-down LM-MAP). After performing camera intrinsic and hand-eye calibration, the test bench provides the ground-truth pose of the mock-up in the camera reference frame. Accordingly, the CAD model of the LM-MAP can be overlaid onto the camera data to generate training data and to compute quantitative metrics in our experiments.

B. Data generation

We have recorded around two hours of data with an iniVation Davis 346 across a collection of sequences that last between 40 seconds and 5 minutes and span a wide range of trajectories, mock-up appearance, and illumination conditions. To conduct various analyses in the following section, we have split the data into four different categories:

1) Augmented textures training: This set contains around one hour and twenty minutes of data with the LM-MAP mounted on non-realistic looking satellite mock-ups (c.f. Fig. 5(a) and (b)). The sequences are collected as a combination of LM-MAP trajectories and illumination conditions varying from no light to full sunlighting with earthshine.



(a) Unrealistic cover (b) Unrealistic cover (c) Realistic mock-up

Fig. 5. Example of images from the data collected with our satellite docking test bench and the Davis 346 RGB/event camera. (a) and (b) are from the *augmented texture* training set. (c) is from the *realistic training* set.

2) Realistic training: Here, around 20 minutes of data are collected using a realistic satellite mock-up as shown in Fig. 4 and 5(c). The mock-up is built with Mylar fixed on an aluminium panel with epoxy to simulate the visual aspect of multi-layer insulation of real satellites. These data also span different trajectories and illumination conditions.

3) Realistic test: This set consists of three sequences of three and a half minutes using the realistic mock-up. Each sequence is based on the same trajectory which differs from the ones in *realistic training*. Only one variable changes from one sequence to another, the level of light: colocated-only, colocated with earthshine, colocated with full sun.

4) Hard cases: Two sequences of 40 seconds have been recorded with the realistic mock-up to show the limits of RGB and event modalities with the proposed pipeline.

C. CNN training

The proposed CNN-based filters are trained in a supervised manner using the known robotic arm pose and satellite mock-up CAD. For each RGB image or event histogram, binary masks are independently generated for the navigation ring and the trio of reflectors. We have trained two sets of ring and reflector filters, one with the *augmented textures* set only, and one with the combination of *augmented textures* and *realistic training* sets. We respectively denote these CNN sets as *high-domain-gap* and *low-domain-gap* models. The training process simply consists of optimising the CNN parameters using the PyTorch implementation of Adam with a binary cross-entropy loss function between the model outputs and the target binary images. The networks possess around 800k parameters and the training procedure takes less than five hours on an Nvidia RTX A500 laptop GPU using an 80/20 cut for training and validation sets. Note that we have not performed any specific hyperparameter training on the proposed network.

IV. Experimental Results

In this section, we aim at demonstrating the soundness of the proposed LM-MAP detection and state estimation pipeline as well as comparing the advantages and drawbacks of both RGB and event modalities for the task of satellite docking. In our quantitative experiments, we filter the outlier state estimates by only considering estimates to be valid when three consecutive predicted poses are close enough to one another. Concretely, if the orientation differs by more than 15° between consecutive estimates of $\mathbf{R}$, the current

pose is rejected. Results shown in this section include or not this simple filtering step as specified.

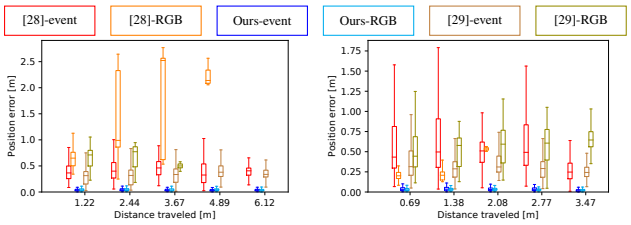
All our experiments are run in real-time at 10Hz on a consumer-grade laptop equipped with an Intel i7-1370p CPU with 32GB of RAM, and an Nvidia RTX A500 (mobile) GPU with 4GB of VRAM. The typical computation load of the proposed pipeline consumes around 25% of the CPU, 5% of the GPU, and 330MB of VRAM. Note that the RANSAC and yaw estimation codes are naive implementations in Python (with the latter code being the main computation bottleneck). Accordingly, an optimised C++ (with or without GPU parallelisation) would greatly lower the computational cost of the proposed pipeline, allowing for its use on low-power embedded systems.

A. Accuracy and generalisation

1) Gap in domain adaptation: This set-up aims to demonstrate the global accuracy of the proposed pipeline as well as pointing out the difference between the modalities in terms of generalisation. Accordingly, we have performed a quantitative analysis of the framework's output over the three sequences of the *realistic testing* set, and for both the *high-domain-gap* and *low-domain-gap* models, with and without the aforementioned outlier rejection mechanism. Looking at the results shown in Table I, one can see that with the *high-domain-gap* models and the outlier rejection, both modalities perform similarly for the low and medium-light sequences. However, with higher illumination, the event-based estimation significantly outperforms the RGB-based one. Given such results, it could seem that the difference in the cameras' HDR capabilities is the main explanation for this empirical observation. To test this hypothesis, we have run the same experiment using the *low-domain-gap* models. As shown on the right of Table I, the RGB and event-based results display similar levels of accuracy over all the testing sequences.[1] This rejects the HDR difference hypothesis as the accuracy does not correlate with the sensing modality when using a higher level of light. Thus, we hypothesise that using event-based histograms offers better generalisation abilities than standard RGB data as they are less sensible to the actual appearance of the mock-up due to their "edge-detection"-like visual aspect. More precisely, with the low-light sequences the RGB images in both the *augmented texture* and *realistic testing* sets are quite similar (mostly dark) but when using high illumination, the appearance between the two sets differs greatly (larger domain gap). Thus the *high-domain-gap* models struggle to generalise for RGB data. This is not the case for the event-based histograms as both the *high-domain-gap* and *low-domain-gap* models perform similarly. The domain gap hypothesis is also favoured by the fact that in these experiments the level of RGB saturation around the LM-MAP does not correlate, with the final accuracy.

Another interesting observation is the impact of the simple outlier rejection mechanism. Regardless of the CNN model or modality, the outlier filtering does effectively improve the RMSE while having a much lower impact on the median as expected (meaning that less outliers are present in the output). An obvious drawback of the outlier filtering is the decrease in detection rate. Note that we also have tested the proposed pipeline on datasets that do not contain any LM-MAP. We have observed only 5 false detections over more than 5000 processed frames.

[1]Additional visualisation at `https://youtu.be/SuDh-xhnaVY`.



(a) *Realistic testing* high light (b) *Realistic training* high light

Fig. 6. Accuracy benchmark with Visual Odometry pipelines over two sequences (a) and (b). The metric is the relative position for various trajectory lengths (based on [30] with *sim3* alignment of the first 40 frames).

2) Visual odometry benchmark: To the best of our knowledge, there is no off-the-shelf open-source estimation pipeline that directly addresses the problem at hand. To provide a benchmark, we chose to compare our method with common Visual Odometry (VO) frameworks that are *EVO* [16] (event-only), *OrbSlam3* [28], and *SVO* [29]. While our set-up differs from standard VO, the background of our test bench is close to being featureless. This is especially true for the event-based tests as no events are generated by the static background. Thus, the static-environment assumption of VO is not violated and the estimates correspond to the relative pose of the camera with respect to the satellite mock-up. As *OrbSlam3* and *SVO* are designed to operate with traditional images, we run both frameworks with RGB images and event frames (histograms of 35k consecutive events) independently. Overall, *EVO* has failed on all the sequences in the *realistic testing* set (tracking failed after a couple of seconds during each test). *OrbSlam3* and *SVO* best performed on the high-light sequence but also suffered numerous loss of tracking. As illustrated in Fig. 6, *OrbSlam3* and *SVO* can occasionally perform at a similar level to our method (lowest boundary of the box plots) but the overall accuracy is not sufficient for satellite docking. For the sake of fairness, it should be noted that the benchmarked methods are not designed for photometrically challenging environments with texture aliasing and rotational symmetries.

3) Theoretical limits of model: In Table I we have demonstrated the soundness of the proposed estimation pipeline with various metrics. However, the normal/rotation accuracy does not seem to be on par with the small position error obtained across all datasets with both RBG and event-based modalities. This observation might be surprising due to the well-known fact that VO pipelines generally estimate accurately the camera's orientation with sub-degree precision thanks to the high sensitivity of the camera measurements with respect to the camera orientation. Unfortunately, for

TABLE I

ACCURACY ANALYSIS OF THE PROPOSED DETECTION AND STATE ESTIMATION PIPELINE USING RGB OR EVENT DATA IN THREE DATASETS WITH DIFFERENT LEVELS OF ILLUMINATION.

		Using *high-domain-gap* training data						Using *low-domain-gap* training data					
		Low light		Medium light		High light		Low light		Medium light		High light	
		RGB	Event	RGB	Event	RGB	Event	RGB	Event	RGB	Event	RGB	Event
With outlier filtering													
Position error [m]	med.	**0.011**	0.013	**0.012**	0.016	0.041	**0.019**	**0.009**	0.014	**0.011**	0.014	0.019	**0.018**
	RMSE	**0.019**	**0.019**	0.023	**0.022**	0.054	**0.027**	**0.015**	0.021	0.022	**0.020**	0.028	**0.024**
Normal error [°]	med.	**4.873**	5.354	**5.616**	5.656	8.084	**5.656**	**4.974**	5.326	**4.979**	5.321	**5.276**	5.682
	RMSE	**5.797**	6.733	**7.761**	8.061	26.066	**7.983**	**5.807**	6.268	6.951	**6.230**	7.734	**7.257**
Rotation error [°]	med.	**5.719**	6.603	**6.751**	6.939	13.095	**7.072**	**5.968**	6.488	**6.063**	6.742	**6.729**	7.096
	RMSE	9.776	**8.498**	**9.280**	10.029	32.440	**9.441**	9.004	**7.513**	8.198	**8.005**	9.588	**9.056**
Detection rate [%]	all	**67.4**	57.1	42.0	**45.0**	22.7	**39.1**	**69.3**	38.3	**54.3**	45.9	**54.4**	31.7
	in FoV	**86.9**	85.5	76.5	**80.3**	26.6	**66.3**	**89.2**	65.4	81.8	**84.1**	**69.9**	55.4
Without outlier filtering													
Position error [m]	med.	**0.011**	0.014	**0.013**	0.018	0.049	**0.021**	**0.010**	0.015	**0.012**	0.015	0.021	**0.020**
	RMSE	**0.025**	0.120	**0.031**	0.098	0.084	**0.064**	**0.022**	0.031	**0.039**	0.098	**0.039**	0.083
Normal error [°]	med.	**4.936**	5.575	5.939	**5.776**	8.862	**6.154**	**5.021**	5.538	**5.250**	5.512	**5.701**	6.435
	RMSE	**10.939**	16.505	**10.587**	19.388	31.570	**19.168**	10.779	**7.851**	17.401	**11.425**	13.196	**13.132**
Rotation error [°]	med.	**6.075**	6.994	**7.230**	7.484	16.887	**8.498**	**6.288**	6.872	**6.360**	7.165	**7.472**	8.410
	RMSE	**15.608**	18.974	**13.206**	22.844	38.952	**23.549**	14.747	**10.491**	19.461	**15.542**	**17.030**	18.213
Detection rate [%]	all	**82.5**	71.9	56.8	**62.2**	58.3	**62.2**	**84.8**	53.8	**69.6**	61.2	**75.1**	56.0
	in FoV	**100**	99.6	98.1	**99.2**	80.1	**92.5**	**100**	90.2	**99.7**	99.4	**94.0**	87.4
Dataset properties													
Saturation of RoI [%]	high	0.0	-	1.7	-	12.1	-	0.0	-	1.7	-	12.1	-
	low	55.3	-	23.2	-	0.1	-	55.3	-	23.2	-	0.1	-
	total	55.3	-	24.9	-	12.2	-	55.3	-	24.9	-	12.2	-

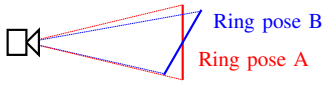
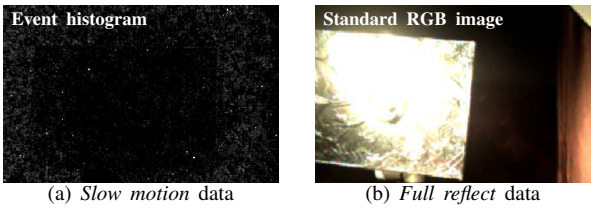


Fig. 7. Illustration of the worst case scenario of measurement sensitivity with respect to the docking port inclination.



(a) *Slow motion* data (b) *Full reflect* data

Fig. 8. RGB and event data samples from the two *hard cases* sequences.

LM-MAP state estimation, we face the opposite situation where the measurements' sensitivity is very low. Depending on the relative pose of the port with respect to the camera, a large difference in the LM-MAP's orientation might lead to a very small variation of measurements. As illustrated in Fig. 7 for the worst case scenario (LM-MAP being fronto-parallel to the camera), a one-pixel difference in the LM-MAP appearance at a typical distance of $0.6\,\mathrm{m}$ can be explained by an $8.9°$ difference of orientation. When the LM-MAP is inclined by $45°$ the same pixel noise corresponds to lower a orientation variation of about $1°$. Accordingly, the proposed pipeline possesses a lower bound on its rotational accuracy which is a function of the distance to the camera and the true orientation. This explains why in Table I the position RMSEs are still correct without outlier filtering while the normal and rotation ones are not.

B. Modality limits

With this setup, we briefly expose some of the limitations of the RGB and event modalities. In Fig. 8 we show samples from the two sequences, *slow motion* and *full reflect*, from the *hard cases* dataset. The first one was recorded with extremely low velocity of the target ($1.5\,\mathrm{mm/s}$) and solely the colocated light switched on (no ambient light), while the second one focuses on a high level of reflection from the sun-analog light source on the area of the LM-MAP. With the *low-domain-gap* models, we obtained a success rate of ellipse detection of 100% with the RGB data and 1.16% with the event-based histograms when using the *slow motion* sequence. For the *full reflect* one, it is the opposite with 0.84% with the RGB data and 79.8% with the events. As illustrated in Fig. 8, the RGB camera is incapacitated in the presence of strong reflection due to sensor saturation, and the event frames suffer from an extremely low signal-to-noise ratio when the target moves very slowly. Both situations are common in the context of satellite docking and maintenance operations. This emphasises the fact that any single modality is not perfectly fit for our application and that complementing events with RGB data represents a promising path of more robust perception in space applications.

V. CONCLUSIONS

We have presented a docking port detection and monocular state estimation framework that mixes data-driven techniques (feature filtering) and geometric models (state estimation). Unlike most existing methods we focused on satellite-agnostic operations that do not require prior knowledge of the spacecraft's CAD model, thus leading to better generalisation abilities without the need to retrain any network. The performance of the framework was evaluated with real data from a physical simulator using both RGB and event cameras. The proposed method despite possessing theoretical limits demonstrated acceptable levels of accuracy. However,

a conclusion from our experiments is that none of the two modalities is individually sufficient to ensure robust estimation across all the environmental conditions encountered during satellite docking operations. Accordingly, future work will explore the seamless integration of the RGB and event data to leverage their complementary strengths. We will also investigate the use of spiking neural networks to better exploit the spatiotemporal nature of event data.

REFERENCES

[1] J. Yang, P. Gong, R. Fu, M. Zhang, J. Chen, S. Liang, B. Xu, J. Shi, and R. Dickinson, "The role of satellite remote sensing in climate change studies," *Nature climate change*, vol. 3, no. 10, pp. 875–883, 2013.

[2] Lockheed Martin, "Lockheed Martin Mission Augmentation Port (MAP) Standard," Lockheed Martin, Tech. Rep., 2022.

[3] C. Samson, C. English, A. Deslauriers, I. Christie, F. Blais, and F. Ferrie, "Neptec 3D laser camera system: From space mission STS-105 to terrestrial applications," *Canadian Aeronautics and Space Journal*, vol. 50, no. 2, pp. 115–123, 2004.

[4] S. Ruel, T. Luu, and A. Berube, "On-orbit testing of target-less TriDAR 3D rendezvous and docking sensor," in *Proc of the International Symposium on Artificial Intelligent, Robotics and Automation in Space (i-SAIRAS 2010)*, 2010.

[5] O. Ma and G. Yang, "Validation of a satellite docking simulator using the soss experimental testbed," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4115–4120.

[6] M. Zebenay, T. Boge, R. Krenn, and D. Choukroun, "Analytical and experimental stability investigation of a hardware-in-the-loop satellite docking simulator," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 229, no. 4, pp. 666–681, 2015. [Online]. Available: https://doi.org/10.1177/0954410014539290

[7] DLR Space Operations and Astronaut Training, "Epos 2.0 rvd simulator," *Journal of large-scale research facilities*, vol. 3, p. A107, 2017. [Online]. Available: http://dx.doi.org/jlsrf-3-155-6

[8] N. W. Oumer, G. Panin, Q. Mülbauer, and A. Tseneklidou, "Vision-based localization for on-orbit servicing of a partially cooperative satellite," *Acta Astronautica*, vol. 117, pp. 19–37, 12 2015.

[9] N. Munasinghe, C. Le Gentil, J. Naylor, M. Asavkin, D. G. Dansereau, and T. Vidal-Calleja, "Towards event-based satellite docking: A photometrically accurate low-earth orbit hardware simulation," in *HERMES2 Workshop at ICRA2024*, 2024.

[10] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[11] 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, 1 2022.

[12] I. Alzugaray and M. Chli, "Asynchronous corner detection and tracking for event cameras in real time," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3177–3184, 2018.

[13] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based feature tracking with probabilistic data association," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4465–4470.

[14] I. Alzugaray and M. Chli, "HASTE: multi-hypothesis asynchronous speeded-up tracking of events," in *31st British Machine Vision Virtual Conference (BMVC)*, 2020.

[15] C. Le Gentil, I. Alzugaray, and T. Vidal-Calleja, "Continuous-time gaussian process motion-compensation for event-vision pattern tracking with distance fields," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 804–812.

[16] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 593–600, 2017.

[17] C. Le Gentil, F. Tschopp, I. Alzugaray, T. Vidal-Calleja, R. Siegwart, and J. Nieto, "Idol: A framework for imu-dvs odometry using lines," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5863–5870.

[18] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 4 2018.

[19] T.-J. Chin, S. Bagchi, A. Eriksson, and A. van Schaik, "Star Tracking Using an Event Camera," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 6 2019, pp. 1646–1655.

[20] S. Bagchi and T.-J. Chin, "Event-based Star Tracking via Multiresolution Progressive Hough Transforms," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 3 2020, pp. 2132–2141.

[21] 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, no. 2, pp. 125–141, 6 2019.

[22] 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, 12 2020.

[23] 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)*. IEEE, 5 2023, pp. 11 866–11 873.

[24] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576523003995

[25] T. H. Park and S. D'Amico, "Online supervised training of spaceborne vision during proximity operations using adaptive kalman filtering," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 744–11 752.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015, pp. 234–241.

[27] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, p. 236–239, mar 1984. [Online]. Available: https://doi.org/10.1145/357994.358023

[28] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[29] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[30] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.