M. Piazza, M. Maestrini, P. Di Lizia
Monocular Relative Pose Estimation Pipeline for Uncooperative Resident Space Objects
Journal of Aerospace Information Systems, published online 25/05/2022
doi:10.2514/1.I011064

Monocular Relative Pose Estimation Pipeline for Uncooperative Resident Space Objects

Massimo Piazza*
Infinite Orbits, 31000 Toulouse, France

Michele Maestrini † and Pierluigi Di Lizia‡
Polytechnic University of Milan, 20156 Milan, Italy

This paper aims to present a deep learning-based pipeline for estimating the pose of an uncooperative target spacecraft, from a single grayscale monocular image. The possibility of enabling autonomous vision-based relative navigation in close proximity to a non-cooperative Resident Space Object (RSO) would be especially appealing for mission scenarios such as on-orbit servicing and active debris removal. The Relative Pose Estimation Pipeline (RPEP) proposed in this work leverages state-of-the-art Convolutional Neural Network (CNN) architectures to detect the features of the target spacecraft using monocular vision. Specifically, the overall pipeline is composed of three main subsystems. The input image is first of all processed using an object detection CNN that localizes the bounding box enclosing our target. This is followed by a second CNN that regresses the location of semantic keypoints of the spacecraft. Eventually, a geometric optimization algorithm exploits the detected keypoint locations to solve for the final relative pose. The proposed pipeline demonstrated centimeter/degree-level pose accuracy on the Spacecraft PosE Estimation Dataset (SPEED), along with considerable robustness to changes in illumination and background conditions. In addition, the architecture showed to generalize well on real images, despite having exclusively exploited synthetic data from SPEED to train the CNNs.

I. Introduction

This work deals with the problem of estimating the relative pose of an uncooperative spacecraft (S/C) from a single grayscale monocular image, in a close-proximity operations scenario. The term "uncooperative" is here referred to as a situation in which the target S/C is not equipped with supportive means (e.g. light-emitting markers) nor is capable of establishing a communication link. The satellite is modeled as a rigid body, which means that its six-dimensional pose space is defined in terms of three translation components and three attitude components, relative to the chaser S/C.

*Lead Spacecraft Navigation Engineer, massimo@infiniteorbits.io.
†PhD Candidate, Department of Aerospace Science and Technology, michele.maestrini@polimi.it.
‡Assistant Professor, Department of Aerospace Science and Technology, AIAA Member, pierluigi.dilizia@polimi.it.

Two main approaches can be considered for estimating the pose of an uncooperative spacecraft relative to another satellite. The motion of a space object might be in principle estimated by means of ground-based tracking. However, such an estimate would be affected by significant uncertainty and its availability would depend on the target's visibility from ground. Even the use of advanced multi-static laser ranging techniques, which allows a significant improvement in the orbit determination uncertainty compared to radar and passive optical tracking, would still result in an unsuitable position uncertainty for an uncooperative object. For instance, bi-static laser ranging in LEO can guarantee an absolute position uncertainty of about 20 m [1], which is excessive for close-proximity operations. The same problem applies to attitude estimation, which may in principle be done using ground observations only, by measuring RCS [2] or light-curve [3] fluctuations. Nonetheless, the uncertainty would be extremely high for the considered scenario.
These limitations render a ground-based tracking approach unsuitable for close-proximity operations between two spacecrafts. The second approach consists in estimating the pose of the target directly onboard the chaser S/C, by exclusively relying on the sensors available on the latter. This currently represents the only strategy that is suitable for this kind of close-proximity operations.

A possible choice to achieve onboard pose estimation may be the use of LiDAR and/or stereo camera sensors, which, nevertheless, can be extremely expensive and represent a substantial contribution to the mass and power budgets of the S/C. In contrast, monocular cameras are characterized by lower complexity and their use for autonomous relative navigation would translate into significant savings in terms of cost, mass, and power requirements. All these benefits come at the expense of very high complexity of the image processing algorithms. Besides, monocular sensors are characterized by weaker robustness to lighting conditions and variable backgrounds, compared to a LiDAR. This aspect is particularly challenging, given the low signal-to-noise ratio that characterizes spaceborne optical images.

Nonetheless, the advantages of using a monocular camera as a navigation sensor make it an appealing possibility, especially within the framework of on-orbit servicing and Active Debris Removal (ADR) missions. Among the missions of this kind, that are slated for launch during the next few years, an example is NASA's *OSAM-1* mission (previously known as *Restore-L* [4]), which is currently scheduled to launch in 2025, along with the commercial servicing programs proposed by companies like Infinite Orbits and Astroscale. Also, the first-ever ADR mission, *ClearSpace-1*,* is expected to launch in 2026.

It is then clear that the ability to accurately estimate the pose of an uncooperative S/C, by relying on hardware with minimal complexity, represents a key enabling technology in all the aforementioned scenarios.

Typically, all the state-of-the-art techniques used for estimating the pose of a spacecraft from a monocular image make use of an image-processing subsystem that identifies the position in the image frame of certain semantic features of the S/C. This is followed by a pose solver consisting in a geometric optimization subsystem, that fits a known 3D model of the target S/C to the features matched in the image. This routine shall then be embedded in a navigation filter,

*https://www.esa.int/Safety_Security/ClearSpace-1 (accessed on March 24th 2022)

to be used in an actual rendezvous scenario, during which the inter-spacecraft distance ranges from tens of meters to a few centimeters.

Depending on the approach adopted for image processing, two main classes of monocular pose estimation methods may be identified: feature-based and deep learning-based pose estimation, which are respectively introduced in Subsections I.A and I.B. In the remainder of this section, the Spacecraft PosE Estimation Dataset (SPEED) dataset and the SLAB/ESA challenge are described. Subsequently, in Section II the architecture proposed in this work is explained in detail. Then, in Section III, various performance metrics of the pipeline are evaluated. Moreover, the most important characteristics of the input image that correlate with estimation accuracy are identified and some pose estimation results are illustrated. Finally, the achieved results are summarized in Section IV, along with suggestions of possible directions for further research in this field.

A. Feature-based pose estimation

Feature-based methods seek for correspondences between edges detected in the image and line segments of the known wireframe model of the spacecraft. In 2014, D'Amico [5] proposed a monocular vision-based navigation system that, for the first time, enabled proximity navigation with respect to a completely uncooperative space object. Indeed, unlike previous work, neither supportive means (e.g. light-emitting markers) nor a priori knowledge of the target's pose are required. The method has been successfully tested on actual flight imagery captured during the PRISMA mission [6]. However, two fundamental limitations are highlighted in [5]: the excessive computational cost, which prevents real-time usage on spaceborne hardware, and the lack of robustness to changes in lighting and background conditions. In 2018, Sharma et al. proposed their Sharma-Ventura-D'Amico (SVD) feature-based method [7]. The method has been tested on actual flight imagery from the PRISMA mission and, compared to previous work, it proved enhanced computational efficiency and superior robustness to changes in the background. The latter is achieved thanks to the fusion of state-of-the-art edge detectors with the Weak Gradient Elimination (WGE) technique, which eliminates gradients where they are weak and highlights those regions where gradients are strong. In addition, an approach that further improves upon SVD has been introduced in 2019 by Capuano et al. [8]. It is similarly based on WGE, however, the resulting segmented image undergoes a different feature extraction process. It is processed by three parallel streams that independently extract segments/corners. The high-level feature synthesis takes into account only the points that are commonly detected by all three streams: in so doing, the different weaknesses of the three algorithms that may lead to false detections are then compensated.

B. Deep learning-based pose estimation

Deep learning-based methods make use of a Convolutional Neural Network (CNN) pipeline whose job, depending on the approach, may either consist in regressing the position in the image frame of predefined keypoints, that later

become the input of a pose solver [9, 10], or in directly estimating the S/C pose, according to one of the following formulations of the pose estimation problem:

- regression problem [11];
- classification problem, which requires a sufficiently dense discretization of the pose space [12, 13];
- hybrid classification-regression problem [14].

Besides spaceborne applications, there exist plenty of research in the field of monocular pose estimation, involving many other use cases, e.g. robotic manipulation, self-driving cars, human pose detection, etc.

Several end-to-end deep learning pipelines have been proposed to directly estimate the pose of a generic object. One of the first successful attempts, that significantly outperformed previous CNN-based work on the `LINEMOD` [15] and `Occluded-LINEMOD` [16] datasets, is presented in [17]. The approach consists in directly regressing the vertices of a 3D bounding box, from which the pose can be efficiently estimated using the EPnP algorithm [18].
The current state-of-the-art for direct pose regression is represented by the `EfficientPose` network [19], which estimates in a single shot both the 2D bounding box and the 6D pose, while also being highly efficient and scalable.
Some other architectures consist in a classification approach, that requires to discretize the pose space [20, 21].

As an alternative, a wide variety of keypoint-based methods has been proposed [19, 22–26]. They all rely on locating the projected keypoint positions in the image frame, which are then used to solve the Perspective-n-Point (PnP) problem, and this eventually yields the 6D pose. Such keypoints are either selected as semantic landmarks lying on the object's surface or they correspond to the 8 vertices of the 3D bounding box.
The pose estimated through end-to-end networks is typically coarser, compered to their keypoint-based counterparts. The main benefit from a direct pose regression/classification is its computational efficiency, which is nearly independent of the number of objects in the image. On the contrary, for keypoint-based solutions, the runtime is practically linearly increasing with the number of objects in the image. In addition, some keypoint-based methods also rely on an initial object detection stage that identifies the Region of Interest (RoI) and crops the full-resolution image, prior to keypoint detection.

Current state-of-the-art techniques for object detection can be classified into two main categories: region proposal methods (such as `R-CNN` [27], `Fast R-CNN` [28] and `Faster R-CNN` [29]) and one-stage methods (such as `YOLO` [30], `SSD` [31] and `EfficientDet` [32]).
For what concerns region proposal networks, the underlying idea that pushed the development of these methods is to avoid wasting computations on regions that are not of interest, i.e. those for which it can be easily excluded that any of the dataset classes are present. This translates into a common feature of all these methods: they always pre-process the image in order to propose candidate regions where to look for objects.
One-stage detectors exhibit superior computational efficiency compared to state-of-the-art region proposal architectures, while maintaining high detection accuracy.

C. Spacecraft Pose Estimation Dataset

The Spacecraft PosE Estimation Dataset (SPEED) is the first and only publicly available Machine Learning dataset for spacecraft pose estimation and has been released in February 2019, with the start of the Pose Estimation Challenge[†] organized by SLAB in collaboration with ESA (Feb-Jul 2019). The same dataset has been used in the present work, both for training CNNs and for evaluating pose estimation performance.

SPEED consists of 15300 grayscale images of the Tango spacecraft, along with the corresponding pose labels. 15000 of these images have been generated synthetically, while the remaining 300 are actual images of a 1:1 mock-up, captured under high-fidelity illumination conditions at the TRON facility. The camera model used for rendering the synthetic images is that of the actual camera employed for capturing the 300 images of the mock-up. The related parameters are reported in Table 1.

Table 1 SPEED camera model

Parameter	Value
Resolution ($N_u \times N_v$)	1920×1200 px
Focal length f	17.6 mm
Pixel pitch ($\rho_u \equiv \rho_v$)	5.86 μm/px
Horizontal FoV	35.452°
Vertical FoV	22.595°

All the photo-realistic renderings of Tango are generated using an OpenGL-based pipeline. In half of these 15000 images, random Earth images are inserted in the background of the satellite. The Earth backgrounds are obtained by cropping 72 real images captured evenly spaced over 12 hours by the geostationary weather satellite Himawari-8.[‡] In all images with Earth background, the illumination conditions used for rendering Tango are consistent with those in the image of the Earth disk.

Besides, Gaussian blurring ($\sigma = 1$) and Gaussian white noise ($\sigma^2 = 0.0022$) are eventually superimposed to all images. The relative position vector for each generated image is obtained by separately sampling the total distance and the bearing angles:

- total distance $\sim \mathcal{N}(\mu = 3, \sigma = 10)$ m (any value either < 3 m or > 50 m is rejected);
- bearing angles $\sim \mathcal{N}(\mu = [u_0, v_0], \sigma = [5u_0, 5v_0])$ px where $u_0 = \frac{N_u}{2}$, $v_0 = \frac{N_v}{2}$ denote the camera principal point

As far as the actual mock-up images are concerned, given the physical constraints of the TRON facility, the distance distribution of real images is very limited compared to synthetic ones and ranges between 2.8 m and 4.7 m. In addition, unlike the synthetic image source (for which pose labels are automatically annotated), the accurate determination of "ground truth" relative poses of the mock-up requires a complex calibrated motion capture system. The facility includes

[†]https://kelvins.esa.int/satellite-pose-estimation-challenge/ (accessed on August 8th 2021)

[‡]https://himawari8.nict.go.jp (accessed on March 24th 2022)

10 Vicon Vero v1.3x cameras that track several infrared reflective markers placed onto Tango's body and in the robotic arm that holds the camera (the one that collects the 300 images in the dataset). High accuracy light sources are present to mimic sunlight and Earth's albedo.

As of August 2021, to maintain the integrity of the post-mortem Pose Estimation Challenge (see Subsection I.D), the ground truth labels of the test set have not been publicly disclosed. Given the purposes of this work, which include a detailed evaluation of both performance and uncertainty of the proposed pose estimation pipeline, it was clearly of paramount importance to be provided with test labels. It was therefore decided to perform a re-partitioning of the original training set (for which pose labels are publicly available) into three new training, validation, and test sets. In particular, the original 12000 training examples were first of all randomly shuffled and then divided into:

- 7680 training images (64%)
- 1920 validation images (16%)
- 4800 test images (24%)

D. SLAB/ESA challenge

The SLAB/ESA Pose Estimation Challenge is based on the evaluation of a single scalar error metric. For convenience, this metric will be referred to as the "SLAB score". Although this metric is separately computed for both the real and synthetic datasets, participants are exclusively ranked based on the performance on synthetic images.

The SLAB score of each image is determined as the sum of a translation error and a rotation error, as defined in Equation (1). The translation error is computed as the norm of the difference between the Ground Truth (GT) relative distance vector $\mathbf{r}$ and the estimated one $\hat{\mathbf{r}}$, normalized with respect to the GT distance. The rotation error is defined as the quaternion error between the GT relative attitude and the corresponding estimate.

$$e^{(i)}_{\text{SLAB}} = \underbrace{\frac{\left\|\mathbf{r}^{(i)} - \hat{\mathbf{r}}^{(i)}\right\|}{\left\|\mathbf{r}^{(i)}\right\|}}_{e^{(i)}_t} + \underbrace{2 \cdot \arccos\left|\mathbf{q}^{(i)} \cdot \hat{\mathbf{q}}^{(i)}\right|}_{E^{(i)}_q} \tag{1}$$

The overall score is then just the average over all the N test images.

$$e_{\text{SLAB}} = \frac{1}{N} \sum_{i=1}^{N} e^{(i)}_{\text{SLAB}} \tag{2}$$

The outcome of the original competition is described in [33] and summarized in Table 2.

Table 2 Leaderboard of the top 10 teams

Team name	Synthetic images score	Real images score	Translation error [m] ($\mu \pm \sigma$)	Quaternion error [deg] ($\mu \pm \sigma$)
1. `UniAdelaide` [10]	**0.0094**	0.3752	0.032 ± 0.095	0.41 ± 1.50
2. `EPFL_cvlab` [34]	0.0215	**0.1140**	0.073 ± 0.587	0.91 ± 1.29
3. `pedro_fairspace` [11]	0.0571	0.1555	0.145 ± 0.239	2.49 ± 3.02
4. `stanford_slab` [9]	0.0626	0.3951	0.209 ± 1.133	2.62 ± 2.90
5. `Team_Platypus`	0.0703	1.7201	0.221 ± 0.530	3.11 ± 4.31
6. `motokimura1`	0.0758	0.6011	0.259 ± 0.598	3.28 ± 3.56
7. `Magpies`	0.1393	1.2659	0.314 ± 0.568	6.25 ± 13.21
8. `GabrielA`	0.2423	2.6209	0.318 ± 0.323	12.03 ± 12.87
9. `stainsby`	0.3711	5.0004	0.714 ± 1.012	17.75 ± 22.01
10. `VSI_Feeney`	0.4658	1.5993	0.734 ± 1.273	23.42 ± 33.57

II. Relative Pose Estimation Pipeline

In this section, the proposed architecture for the Relative Pose Estimation Pipeline (RPEP) is presented. The involved algorithms require the knowledge of camera intrinsics and of the 3D model of the target spacecraft to rendezvous with. Based on this, the architecture that has been developed is capable of estimating the pose of the target spacecraft, from a single monocular grayscale image given as input.

The outline of the architecture is represented in Fig. 1 and it consists of three main subsystems. The first subsystem, called the Spacecraft Localization Network (SLN) and described in Subsection II.A, is responsible for identifying the Region of Interest (RoI) in the image. It is followed in the pipeline by the Landmark Regression Network (LRN), that is detailed in Subsection II.B, whose role is to detect semantic keypoints of the target S/C in the RoI. The third and last subsystem is the pose solver, which, given the landmarks identified by LRN, seeks for the corresponding best pose fit based on the known wireframe model of the target. More specifically, it first runs the Efficient Perspective-n-Point (EPnP) algorithm [18] to obtain an initial estimate of the pose and, in a nominal situation (i.e. if no pose outlier is detected), it successively refines the initial solution using the Levenberg-Marquardt Method.

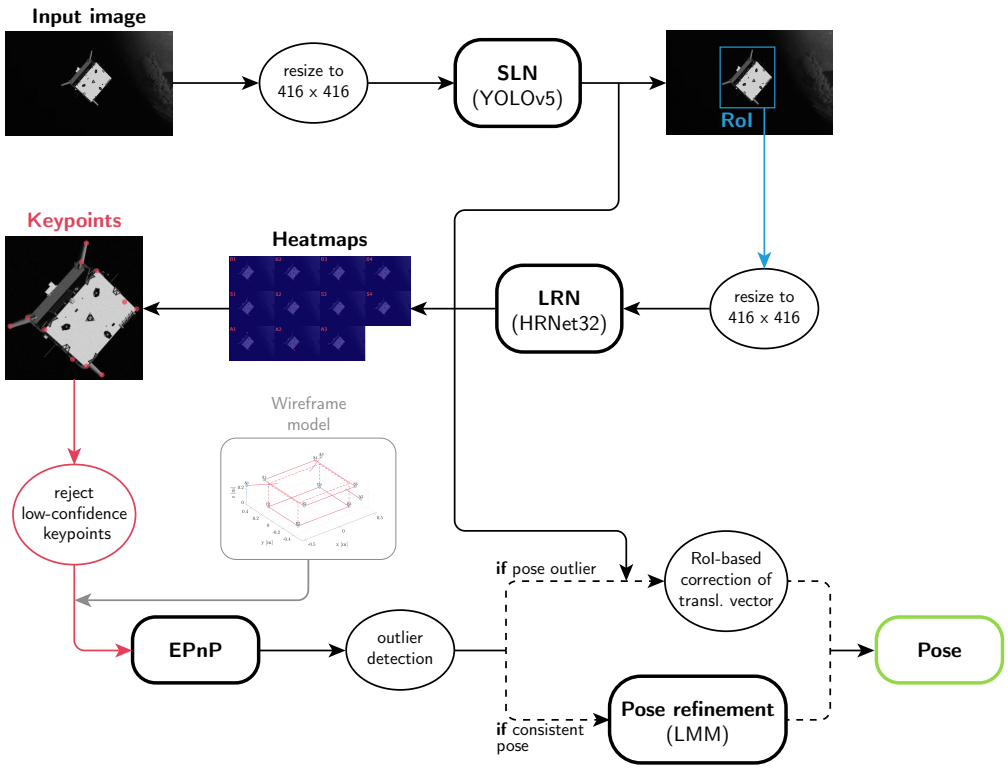


Fig. 1 Architecture of the pose estimation pipeline at inference time

A. Spacecraft Localization Network

The Spacecraft Localization Network (SLN) is the first image processing subsystem of the RPEP proposed in this paper. The You Only Look Once (YOLO) architecture [30], which is a state-of-the-art one-stage method for object detection, has been chosen for this purpose. In particular, one of the most recent iterations of the CNN, `YOLOv5`,[§] was trained to detect the Tango satellite. The SLN receives as input a grayscale image, that is properly resized to match the input size of 416×416 of the `YOLOv5` architecture. This subsystem outputs the so called Region of Interest (RoI), namely the Bounding Box (BB) coordinates associated with the portion of the image containing the S/C. Based on this, further processing of the image will exclusively focus on the identified RoI.

First of all, a one-stage detection approach was chosen over region proposal networks (e.g. [29]) given the clear superiority in terms of computational efficiency of the former class of methods. This is of paramount importance in a spaceborne navigation scenario, where the computing power constraints always make it necessary to opt for efficient yet robust algorithms. In this sense, the smallest model-size version of `YOLOv5`, named `YOLOv5s`, proved particularly interesting for our purposes and has eventually been selected. With 7.5M parameters to train and 191 layers, it appears as an excellent trade-off between speed and accuracy.

[§]https://github.com/ultralytics/yolov5 (accessed on August 8th 2021)

1. Training

The SPEED training labels released by SLAB only include the pose of the Tango spacecraft, provided in terms of translation vector and attitude quaternion for each image. This means that any further label that might be used for intermediate processing steps will have to be annotated.

The minimum rectangle enclosing the S/C in the image frame can be obtained by projecting onto the image plane a simple wireframe model of Tango, based on the known pose. We may at this point annotate the BB of each image by taking the minimum and maximum values of the (P_x, P_y) coordinates of the small amount of points in this simplified 3D model.

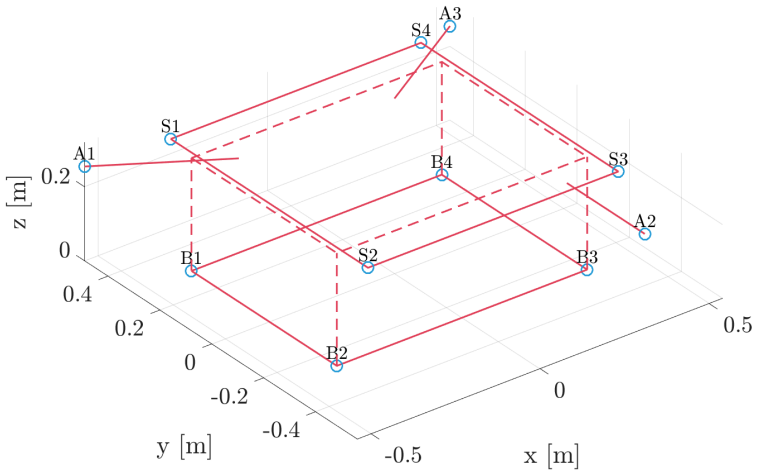


Fig. 2 Wireframe model of the Tango spacecraft

The wireframe model used in our work is depicted in Fig. 2. In particular, the model is composed of 11 semantic keypoints:

- points B1 to B4 are the edges of the bottom surface;
- points S1 to S4 are the edges of the solar panel;
- points A1 to A3 indicate the tips of the Formation Flying Radio Frequency (FFRF) antennas.

These very same keypoints are also used by the successive subsystem, the LRN, which is described in Subsection II.B. The reason behind this choice is that these landmarks represent strong visual features of the spacecraft, and, independently of the pose, most of them will not be occluded by other surfaces.

In order to avoid unintentionally cropping out portions of the S/C from the detected RoI during inference, the minimum rectangle enclosing the projected wireframe model has been slightly relaxed. Specifically, the BB labels are enlarged by the 10% of the average side between width and height of the minimum rectangle. In so doing, the CNN is indeed trained to predict a relaxed bounding box. In Fig. 3, the dashed yellow line indicates the minimum rectangle, while the continuous line is the actual BB label.

The network was trained for 125 epochs using Stochastic Gradient Descent (SGD), with a mini-batch size of 48

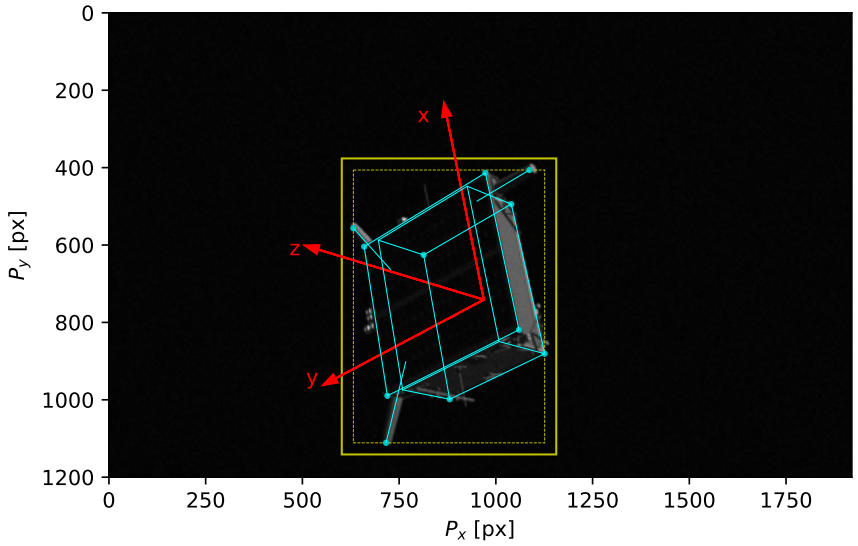


Fig. 3 Bounding box label of the `img001971.jpg` training image

images, learning rate $\alpha = 10^{-3}$, momentum equal to 0.9 and a weight decay of 5×10^{-5}. The binary cross-entropy loss was used during training. In addition, given the assumption of single-class/single-object in the image, a few simplifications in the algorithm were introduced compared to the generic multiple-class/multiple-object framework, for which `YOLO` has been developed. In so doing, we are able to get rid of some unnecessary computation, also making sure that the algorithm outputs one single RoI, provided that the prediction confidence is at least 60%. In other words, we can directly output the prediction with the highest "objectness" score, with no need to process the raw results using the non-max suppression algorithm.

2. Performance evaluation

The performance of SLN has been evaluated using the Average Precision (AP) and Intersection over Union (IoU) metrics.[¶] The AP is defined as the area under the precision-recall curve, that is $\int_0^1 P(R)\,\mathrm{d}R$. The 10 precision-recall curves in correspondence of the IoU thresholds 0.5, 0.55, 0.6, 0.65, . . . , 0.95 are reported in Fig. 4.[‖] These thresholds are used for defining a correct detection (i.e. a True Positive).

The AP_{50}^{95} metric is then simply the average of the AP values computed for each of the 10 curves in Fig. 4. The `YOLOv5` architecture achieves an excellent accuracy, with $\mathrm{AP}_{50}^{95} = 98.51\%$, after only 125 training epochs. Indeed, by comparing the mean and median IoU metrics in Table 3, our spacecraft localization subsystem outperformed at this task both the SLAB baseline and the architecture proposed by the `UniAdelaide` team, which respectively ranked 4th and 1st in the Pose Estimation Challenge.

The prediction results of the SLN subsystem is now illustrated on a few randomly chosen test images. In Fig. 5 the

[¶]https://cocodataset.org/#detection-eval (accessed on August 8th 2021)

[‖]the curves here provided are specifically computed considering the interpolated precision

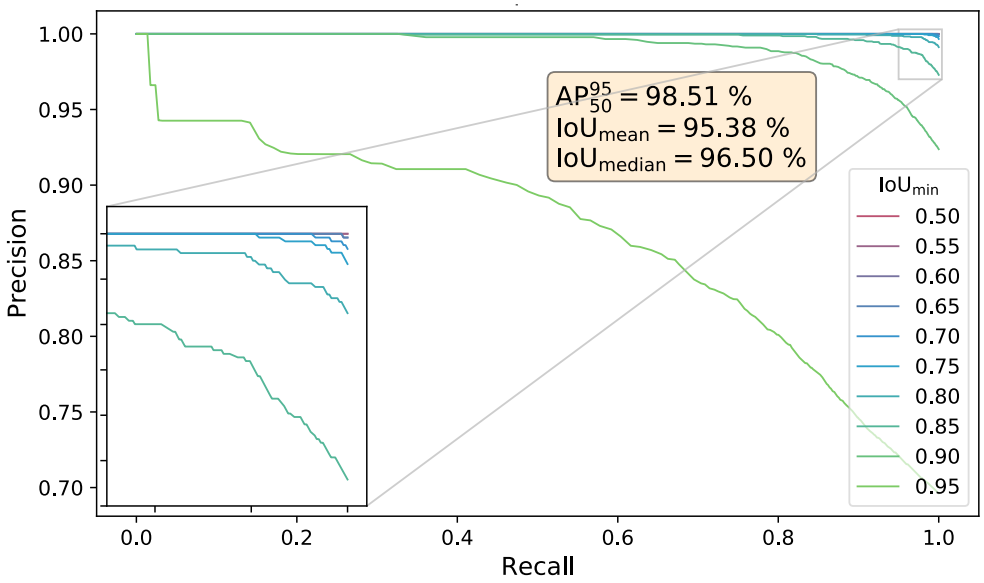


Fig. 4 Precision-recall curves, in correspondence of different IoU thresholds

Table 3 Performance comparison of SLN with other state of the art RoI detection subsystems

	`stanford_slab` [9]	`UniAdelaide` [10]	Our SLN
Mean IoU	91.9%	95.34%	**95.38%**
Median IoU	93.6%	96.34%	**96.50%**

predicted bounding box is drawn in each of the 6 corresponding synthetic test images and the related confidence is also reported. It can be seen that our CNN performs extremely well at identifying a very tight RoI, independently of distance and illumination conditions.

In Fig. 6 inference was run on images characterized by the presence of Earth in the background (a few of these images were rendered in eclipse condition). The excellent robustness to the Earth's presence in the FoV appears evident. This is true in a wide variety of lighting conditions, even in cases in which the S/C is very distant ($\sim$ 30 m) and poorly illuminated. In some cases Tango appears hard to distinguish from the patterns in our planet, also to the human eye.

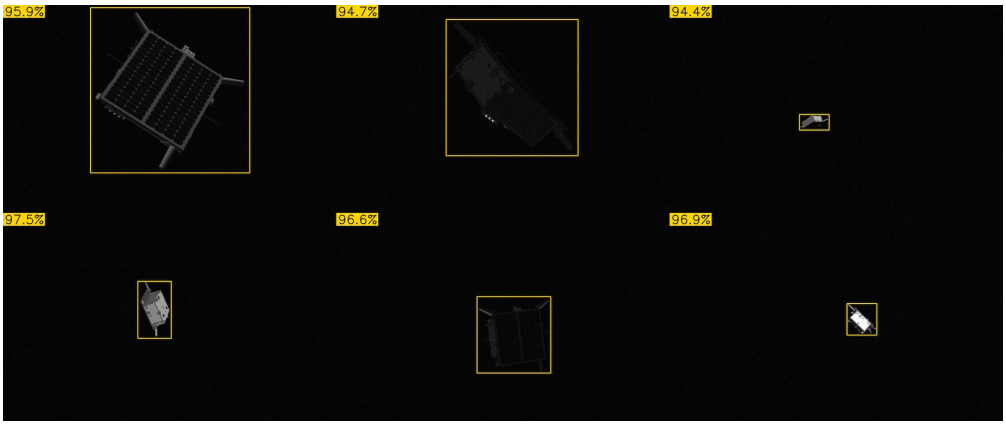


Fig. 5 SLN prediction on 6 test images with black background

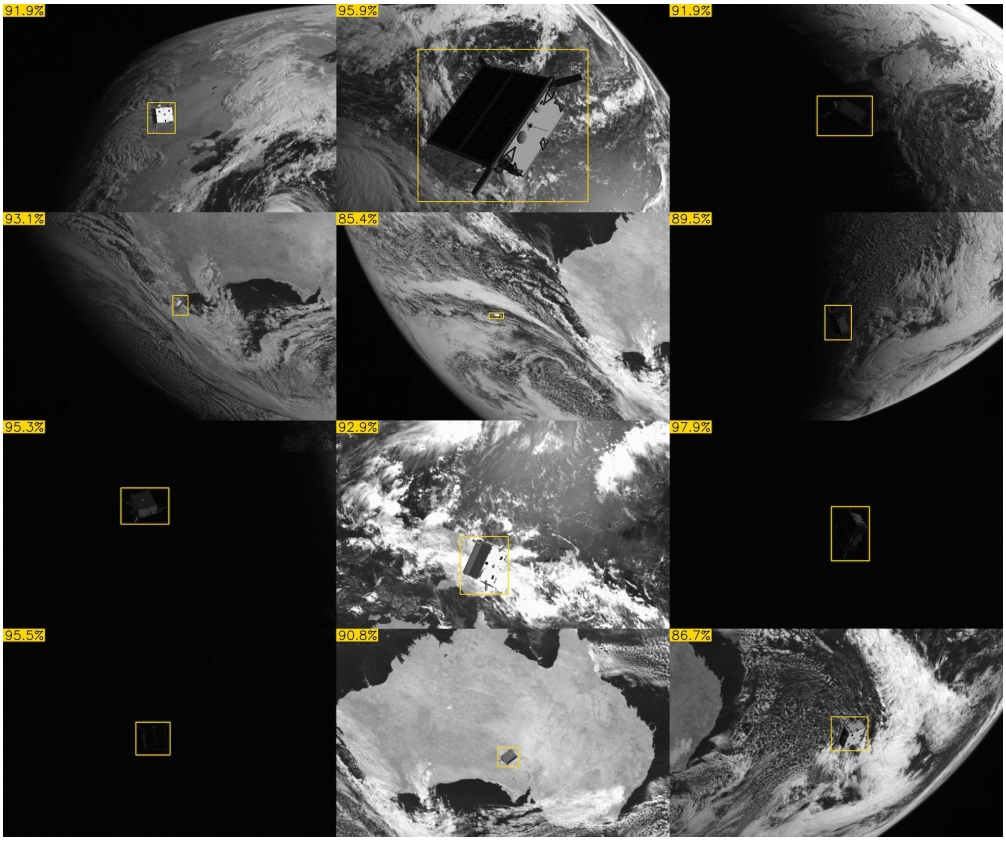


Fig. 6 SLN prediction on 9 test images with Earth background

Although the test labels of real images are not available (except for just five of them), it was decided to run `YOLOv5` on this distribution as well, to figure out from simple visual inspection how well the model generalizes to actual imagery. The experiment was successful and the CNN appeared to identify correctly the RoI in the entire set of images, with very high confidence scores.

B. Landmark Regression Network

The Landmark Regression Network (LRN), which is the second image processing subsystem in the pipeline, receives as input the grayscale RoI detected by SLN. The input size of LRN is again 416×416, which means that RoIs whose largest side is greater than 416 pixels will undergo downscaling. If on the contrary the RoI gets smaller than LRN's input size, then the portion of the image that borders the BB is used to fill the rest of the input window. This is indeed useful in the event of an inaccurate detection where a portion of the S/C would be cropped out.

The unprecedented accuracy demonstrated by the High-Resolution Network (HRNet) architecture [22] in the field of human pose estimation led to the decision of implementing this model in our architecture. This CNN has been trained to regress 11 heatmaps with a size of 416×416, corresponding to the 11 semantic keypoints specified in Fig. 2. The final predicted landmark locations are then obtained as the individual peaks in each heatmap, which will appear as 2D pseudo-Gaussians.

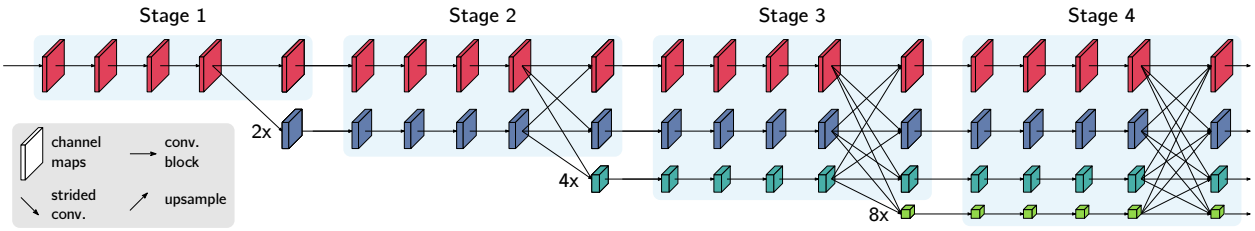


Fig. 7 Main body of the `HRNet` architecture

The strength of `HRNet` lies in two main distinctive aspects.

- Most previous architectures recover high resolution representations by performing an upsampling process downstream of a high-to-low resolution network. In contrast, `HRNet` maintains the initial high-resolution representation throughout the entire network. This clearly eliminates the loss of information associated with traditional approaches, resulting in more accurate heatmaps, which is of paramount importance in a spaceborne relative navigation scenario. In particular, the network starts with a high-resolution subnetwork whose resolution is kept unaltered up to the last layer. As it is depicted in Fig. 7, lower-resolution subnetworks are gradually stacked in parallel as we go deeper in the network.
- Instead of aggregating high- and low-resolution representations, `HRNet` performs repeated multi-scale fusions to boost the low-level representations with the aid of high-level representations, and vice-versa.

In [22] two different versions of the `HRNet` model are presented, which were named `HRNet32` and `HRNet48`. The numbers `32` and `48` indicate versions of the network having respectively 32 and 48 channels in the highest-resolution subnetworks in the last three stages. It was decided to implement the `HRNet32` version, given a performance level quite close to the larger version of the network. The latter appears slightly superior, but this comes at the expense of more than twice the number of FLoating-Point Operations compared to the smaller model [22].

1. Training

The Ground Truth labels have been annotated by projecting onto the image frame the 11 keypoints defined in the 3D wireframe model of Tango, based on the known training poses. The corresponding GT training heatmaps are then set to 2D Gaussians with 1-pixel standard deviation and mean value in correspondence of the projected landmark coordinates.

Despite the use of high-end GPUs on the Google Colab platform, the training of this architecture turned out to be very expensive and has only been carried out for 80 epochs. ADAptive Momentum (ADAM) optimization [35] has been used, with a batch-size of 16 images, $\beta_1 = 0.9$, $\beta_2 = 0.99$, learning rate equal to 10^{-3} and a weight decay of 10^{-4}.

The loss function for the ith image is defined as the mean squared error between the regressed heatmap $\hat{\mathbf{H}}$ and the corresponding Ground Truth $\mathbf{H}$, averaged over all the n landmarks lying inside the image frame:

$$\mathbf{L}_{\text{MSE}}^{(\text{i})} = \frac{1}{n} \sum_{j=1}^{n} v_j^{(\text{i})} \cdot \left[\hat{\mathbf{H}}_j^{(\text{i})} - \mathbf{H}(\mathbf{p}_j^{(\text{i})}) \right]^2 \tag{3}$$

The loss computed for an entire mini-batch is simply the average over all images in the batch, namely $\mathbf{L}_{\text{MSE}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{L}_{\text{MSE}}^{(i)}$.

2. Performance evaluation

The performance of our LRN has been evaluated in terms of AP and Object Keypoint Similarity (OKS). Similarly to the IoU in an object detection framework, the OKS indicates the average degree of overlap between detected keypoints and their actual location.** The 10 precision-recall curves in Fig. 8 are computed in correspondence of the 10 equally spaced OKS thresholds, from 0.5 to 0.95. The Average Precision is then calculated for each of these curves, from which we eventually obtain the global metric $\text{AP}_{50}^{95} = 98.97\%$. This indeed indicates an excellent regression accuracy, obtained after only 80 training epochs.

In Fig. 9, the 11 regressed heatmaps in correspondence each of the two randomly picked input images are reported. All the heatmaps in the figure are characterized by a very sharp peak in correspondence of the landmark location.

It is worth highlighting that the capabilities of the developed CNN are not limited to locating a landmark by identifying in the image the feature associated to it (e.g. a specific edge). Indeed, the network is also able to accurately estimate the position of a keypoint, whenever the related semantic feature is occluded by some other portion of the spacecraft itself: this is the case, for instance, of point B3 in the bottom image or point S2 in the top image of Fig. 9.

**https://cocodataset.org/#keypoints-eval (accessed on August 8th 2021)

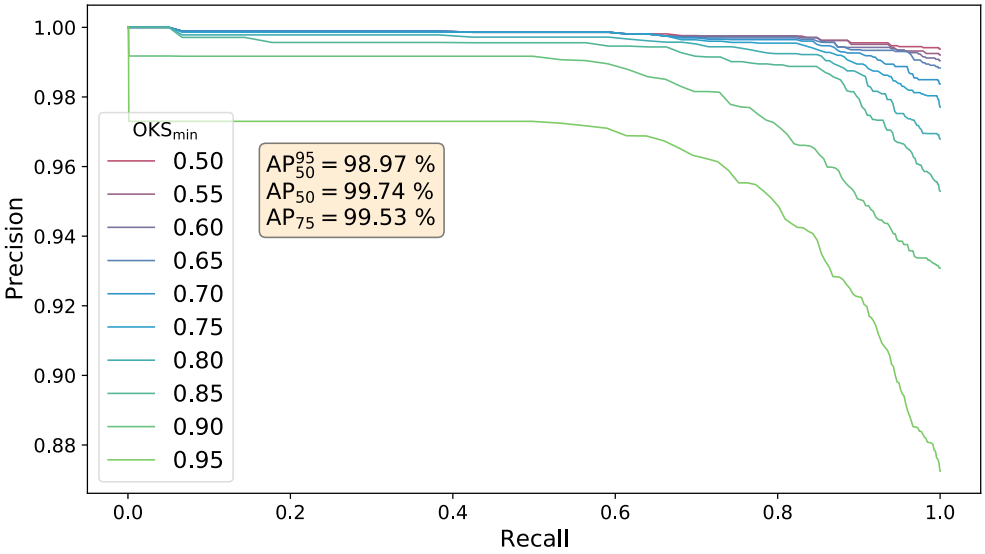


Fig. 8 Precision-recall curves, in correspondence of different OKS thresholds

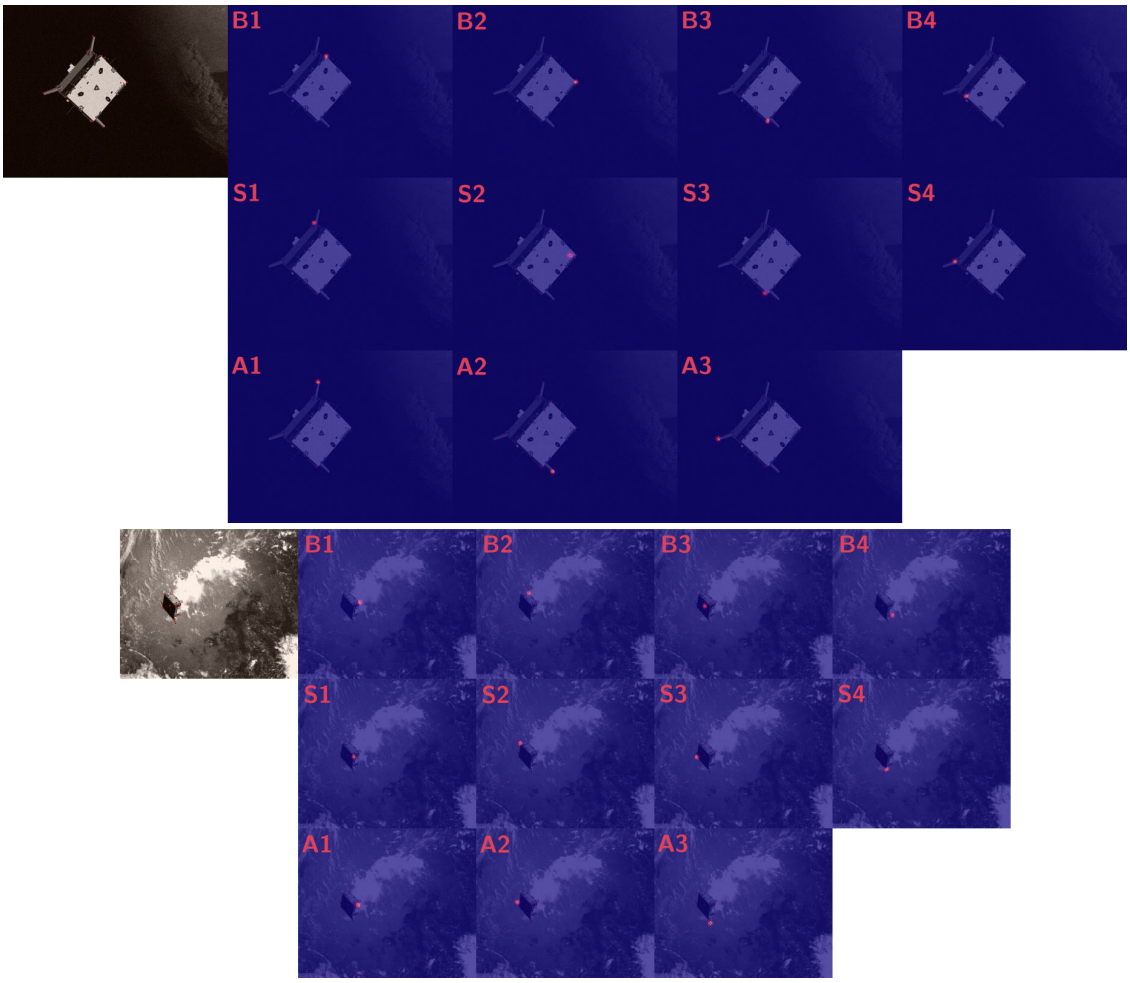


Fig. 9 Examples of regressed heatmaps

C. Pose solver

The pose solver is the third and last subsystem of the proposed RPEP, which identifies the best pose fit based on the keypoints detected by the LRN. The pose solver also leverages BB information (i.e. the output of the SLN) to identify the presence of outliers among the considered keypoints, and partially correct the resulting wrong pose estimate.

1. Keypoint selection

The availability of a heatmap that provides a confidence score for a given detected landmark can be leveraged to filter out potential outliers, which may cause a pose solver to diverge or to output a completely wrong pose. In particular, two hyper-parameters have been tuned, in order to find a good compromise between rejecting potential outliers and retaining a sufficient number of points. Regarding this last goal, it is clearly beneficial in terms of accuracy to over-constrain the 3D model, as long as precise keypoint detections are added. The hyper-parameters that have been consequently selected are:

- # landmarks$_{\text{min}}$: size of the minimal set of landmarks, i.e. the minimum number of the highest-confidence detected landmarks to be always retained, independently of their associated scores
- confidence$_{\text{min}}$: minimum confidence required to retain any landmark in addition to the minimal set

This means that, in general, only a subset of the 11 keypoints will be effectively fed to the pose solver. The optimal tuning of the two above mentioned hyper-parameters will be discussed in Subsection III.B.

2. Initial pose estimation and refinement

After discarding low-confidence landmarks, the remaining ones are fed to the EPnP algorithm [18], which computes a first pose estimate and does not require any initial guess. This method consists in a closed-form solution to the Perspective-n-Point (PnP) problem, having complexity of order $O(n)$. EPnP is characterized by a weak robustness to the presence of outliers among the input keypoints. However, if no outliers are present, the resulting pose estimate turns out to be quite accurate.

At this point, the algorithm checks whether or not the estimated pose is consistent with the BB detected by SLN. Indeed, it was concluded that, after proper training, SLN can be "trusted" more than LRN, just because the former actually performs a simpler task. Thus, whenever an inconsistency is found between the two subsystems, it is reasonable to believe that LRN is to blame. In other words, whenever the projection of Tango's 3D model (based on the initial pose estimate) is inconsistent with the detected BB, this is very likely due to the presence of one or more outliers among the retained landmarks, which translates into a completely wrong pose computed by EPnP.

If no inconsistency is found in the output of EPnP and the reprojection error is acceptable, this initial pose is refined using the Levenberg-Marquardt Method, that iteratively minimizes the reprojection error.

3. Outlier identification & translation correction

If, on the contrary to what previously described, a pose outlier is flagged by the algorithm, the pose is partially corrected by replacing the translation vector output by EPnP with an approximate yet robust estimation. In order to identify a possible pose outlier, an approximate translation vector $\tilde{\mathbf{t}}_{\mathrm{C/B}}$ is first of all computed, by exploiting a RoI-based estimation. This method leverages the knowledge of the characteristic length L_C of the spacecraft, along with the BB's center $(P_x^{\mathrm{BB}}, P_y^{\mathrm{BB}})$ and diagonal length d_{BB} that are detected by SLN. The aforementioned dimensions and coordinates are indicated in Fig. 10.

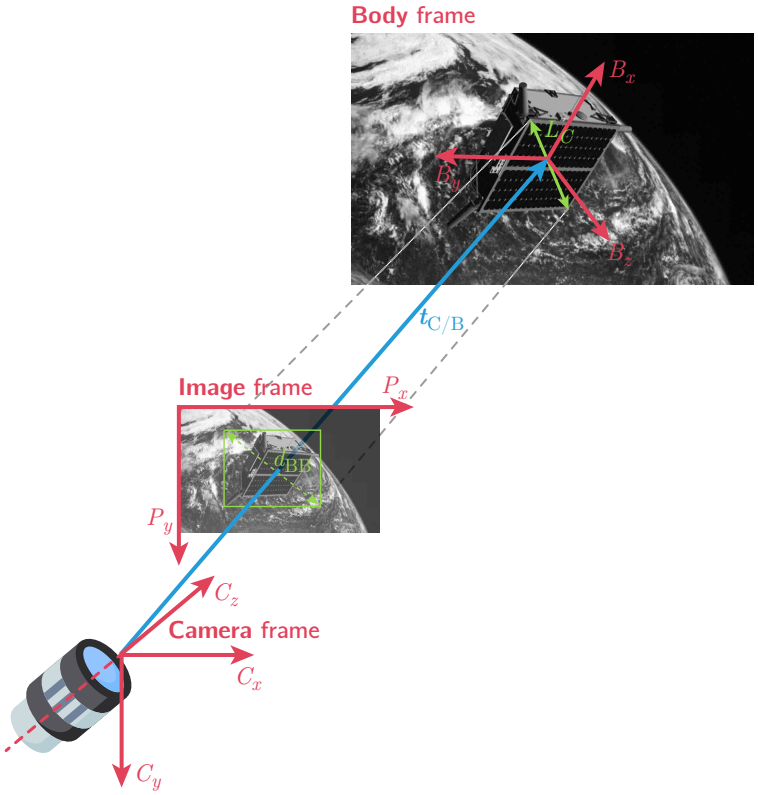


Fig. 10 Reference frames and RoI

Given the camera intrinsics (Table 1), the size of the real-world S/C can be related to the corresponding size in the image frame, hence obtaining the following expression for the distance between the camera-fixed and the body-fixed frames

$$\tilde{t}_{\mathrm{C/B}} = \frac{f/\rho_u + f/\rho_v}{2} \cdot \frac{L_C}{d_{\mathrm{BB}}} \tag{4}$$

The azimuth and elevation angles, α and β, can be similarly computed as

$$\alpha = \arctan\left(\frac{P_x^{\mathrm{BB}} - u_0}{f/\rho_u}\right) \tag{5}$$

$$\beta = \arctan\left(\frac{P_y^{\mathrm{BB}} - v_0}{f/\rho_v}\right) \tag{6}$$

At this point, a coarse estimate of the camera-to-body translation vector may be derived as

$$\tilde{\mathbf{t}}_{\mathrm{C/B}} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ \tilde{t}_{\mathrm{C/B}} \end{Bmatrix} \tag{7}$$

An outlier will be flagged whenever any of the following conditions is encountered.

- The projected geometric center of the S/C, according to the pose estimated by EPnP, has a $> 50\%$ offset[††] from the BB center

$$\frac{|\hat{p}_x^{\mathrm{c}} - P_x^{\mathrm{BB}}|}{w_{\mathrm{BB}}} > 0.5 \quad \text{or} \quad \frac{|\hat{p}_y^{\mathrm{c}} - P_y^{\mathrm{BB}}|}{h_{\mathrm{BB}}} > 0.5 \tag{8}$$

- Large mismatch between the distance estimated by EPnP, $\hat{t}$, and the one obtained from the RoI-based approximation, $\tilde{t}$

$$\left|\frac{\hat{t} - \tilde{t}}{\tilde{t}}\right| > 75\% \tag{9}$$

- Medium distance mismatch and low average confidence of the retained landmarks

$$\left|\frac{\hat{t} - \tilde{t}}{\tilde{t}}\right| > 15\% \quad \text{and} \quad \mathrm{confidence}_{\mathrm{avg}} < 50\% \tag{10}$$

- Medium distance mismatch and high relative reprojection error

$$\left|\frac{\hat{t} - \tilde{t}}{\tilde{t}}\right| > 15\% \quad \text{and} \quad \frac{E_{\mathrm{repr}}}{d_{\mathrm{BB}}} > 10\% \tag{11}$$

Whenever a pose outlier is flagged by the algorithm, the initial estimate of the translation vector will be replaced by the corresponding RoI-based approximation $\tilde{\mathbf{t}}_{\mathrm{C/B}}$.

III. Results

In this section we will first of all define the performance metrics that allow us to evaluate the pose estimation error (Subsection III.A). In Subsection III.B an optimal keypoint rejection process for discarding low confidence predictions from LRN is presented. Thereafter, in Subsection III.C the pose estimation results attained by our pipeline will be presented, while in Subsection III.D we will dwell on the distribution of translation and rotation errors. Subsection III.E highlights the benefits from our pose refinement strategy. In Subsection III.F the runtime breakdown of the three subsystems of the pipeline is illustrated. Eventually, in Subsection III.H several pose estimation results are visualized across a wide range of distances and background conditions.

[††] the pixel offset is normalized with respect to the BB width and height

A. Error metrics

Prior to presenting the results achieved by the proposed RPEP, the error metrics adopted to evaluate the performance of the architecture on the SPEED dataset are defined in terms of mean and median pose error. The median error turned out to be more representative of the accuracy as compared to the mean: the reason is that the presence of only few outliers has been experienced, which are nevertheless characterized by an error that is orders of magnitude larger than nominal detections.

1. Translation error

The absolute translation error for a given image is obtained as

$$E_t = \left\| \hat{\mathbf{t}}_{\mathrm{C/B}} - \mathbf{t}_{\mathrm{C/B}} \right\| \tag{12}$$

which can be easily normalized by dividing it by the GT distance:

$$e_t = \frac{E_t}{\|\mathbf{t}_{\mathrm{C/B}}\|} \tag{13}$$

2. Rotation error

Absolute error The absolute rotation error might be measured in two different ways.

In terms of quaternion error, which represents the overall attitude error with a single scalar metric, it will be computed as

$$E_{\mathrm{q}} = 2 \cdot \arccos |\mathbf{q} \cdot \hat{\mathbf{q}}| \tag{14}$$

In terms of Euler angles, the error will be obtained as the difference between a given estimated Euler angle and the corresponding GT

$$E_{\theta_j} = |\hat{\theta}_j - \theta_j|, \tag{15}$$

Normalized error The main weakness of the SLAB score defined in Equation (1) is that, although it accounts for distance-normalization in its translation component, it does not account for normalization of the rotation error component. This means that the same absolute angular error has the same exact effect upon measured performance, independently of whether that occurs in correspondence of a close-range image or at a distance in which the RoI is just a very small fraction of the entire image area. Consequently, a normalized version of the quaternion error defined in Equation (14) is introduced, which accounts for the angular size of the object relative to the FoV of the camera.

An object's angular size is defined as the angle measured between the two lines of sight corresponding to opposite sides of the object. The angle associated with the diagonal size of each GT Bounding Box is considered in this work.

By resorting to the pinhole camera model and assuming that the lens is set for infinity focus, the diagonal angular size associated with the spacecraft can be computed as

$$\alpha = 2 \cdot \arctan \frac{\rho \cdot d_{\mathrm{BB}}}{2f} \tag{16}$$

where $\rho \equiv \rho_u \equiv \rho_v$ is the pixel pitch [μm/px], d_{BB} is the diagonal length of the BB [px], while f is the focal length [mm].

Note that, in order to normalize the rotation error, it must be divided by a quantity that increases as the attitude gets harder to estimate. The quaternion error defined in Equation (14) is therefore divided by the portion of the diagonal FoV of the camera that is not occupied by the spacecraft, which reads

$$e_q = \frac{E_q}{\mathrm{FoV}_{\mathrm{diag}} - \alpha} \tag{17}$$

where, considering an $N_u \times N_v$ image, the diagonal FoV can be obtained as

$$\mathrm{FoV}_{\mathrm{diag}} = 2 \cdot \arctan \frac{\rho \cdot \sqrt{N_u^2 + N_v^2}}{2f} \tag{18}$$

3. Pose error

The overall pose error is simply measured as the sum of the translation and rotation errors. The SLAB score, which has already been defined in Equation (1), measures the total error as the mean of $\left(e_t^{(\mathrm{i})} + E_q^{(\mathrm{i})}\right)$ computed over all the N test images. After having highlighted the weaknesses the aforementioned metric, we are hereby proposing an alternative performance index that we deem to be more relevant. It has been called the Median Normalized Pose Error (MNPE):

$$e_{\mathrm{MNP}} = \underset{i=1}{\overset{N}{\mathtt{median}}}\left(e_t^{(\mathrm{i})} + e_q^{(\mathrm{i})}\right) \tag{19}$$

where e_t and e_q are defined in Equations (13) and (17), respectively.

B. Optimal keypoint rejection

Grid-search optimization was run to seek for the combination of the two hyper-parameters defined in Subsection II.C.1 that yields the lowest possible MNPE. Recall that these two thresholds determine which of the keypoints detected by the LRN will be discarded and which ones will instead become the input of the pose solver. The obtained results are given in Fig. 11.

As it can be noted from the plot, the minimal set of landmarks shall contain at least 4 points. This is due to the fact that the EPnP algorithm requires $n \geq 4$ points to compute a solution. In addition, although the total number of keypoints

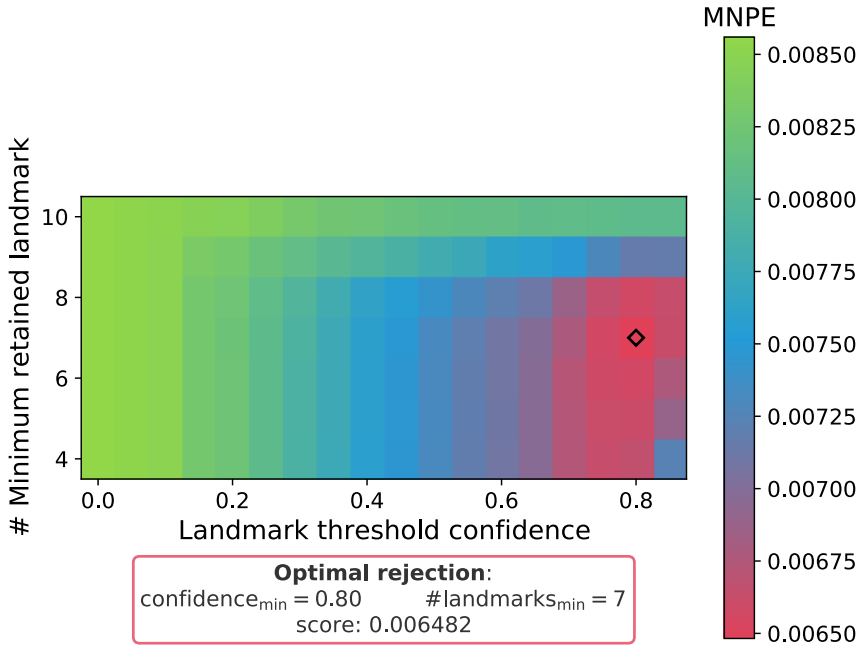


Fig. 11 Optimization of the keypoint rejection process

in the wireframe model amounts to 11, the corresponding hyper-parameter that defines the size of the minimal set of landmarks is only varied between 4 and 10: this is because the condition in which all the 11 landmarks are retained is already included in the case of zero threshold confidence.

The beneficial effect of filtering out low-confidence landmarks is evident from Fig. 11. The threshold confidence alone allows a 24% reduction of the MNPE, compared to retaining all predicted keypoints. In particular, we experienced that accuracy increases monotonically as we increase the threshold confidence up to 0.8, but larger values of such threshold become too restrictive, which causes many precise keypoint detections to be discarded. Introducing a second hyper-parameter that defines the minimal size of the set of retained keypoints grants a further reduction of the error, although the effect is less evident.

All the results presented in the remainder of this discussion have been obtained in correspondence of the optimal values: $\text{confidence}_{\text{min}} = 0.8$ and $\#\text{landmarks}_{\text{min}} = 7$.

C. Performance evaluation

The proposed RPEP achieved a SLAB score of 0.04627 on the test set. This means that, based on the official leaderboard of the SLAB/ESA Pose Estimation Challenge reported in Table 2, the proposed architecture would virtually score 3rd place, hence outperforming the SLAB baseline. Indeed, this performance level has been confirmed by participating in the post-mortem competition. Figure 12 has been printed from the website of the post-mortem

competition[‡‡] and reports the score achieved by the 6 top teams, as of August 8th 2021.

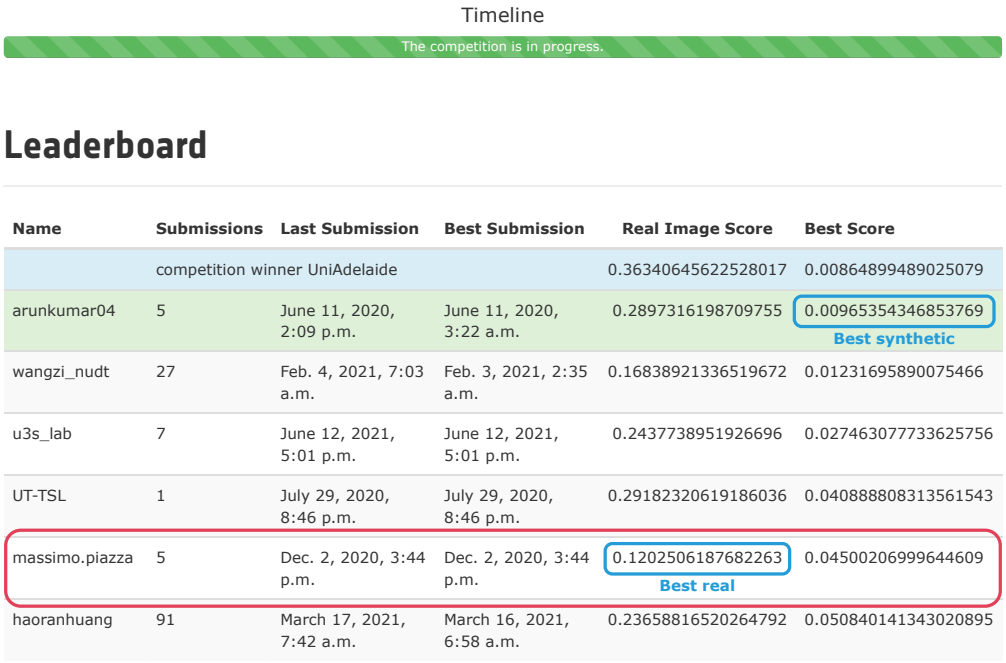


Fig. 12 Top 6 participants of the post-mortem competition

It can be noted that the proposed architecture attained a SLAB score, on the synthetic original test set of SPEED, equal to 0.04500. This corresponds to a performance level that is practically identical to the one estimated on the test set adopted in this work. The score on the synthetic distribution is here labeled as "best score", while the "real image score" indicates the accuracy achieved on the 300 real images of a mockup of the Tango spacecraft.

The most important performance metrics attained by the proposed architecture are reported in Tables 4 and 5.

Table 4 Absolute error of the RPEP

	Absolute error	
	Mean	**Median**
E_t	10.36 cm	3.58 cm
$\mathbf{E}_t$	[0.52 0.56 10.25] cm	[0.24 0.27 3.50] cm
E_q	2.24°	0.81°
$\mathbf{E}_\theta$	[1.57° 0.84° 1.72°]	[0.52° 0.33° 0.34°]
	SLAB score = 0.04627	MNPE = 0.00648

[‡‡]https://kelvins.esa.int/satellite-pose-estimation-challenge/leaderboard/post-mortem-leaderboard (accessed on August 8th 2021)

Table 5 Standard deviation of the RPEP error

	Standard deviation of the error
$\sigma_{\mathbf{E}_t}$	[1.62 1.71 30.44] cm
$\sigma_{\mathbf{E}_\theta}$	[8.92° 5.11° 10.82°]
$\sigma_{\mathbf{e}_t}$	[0.001157 0.001093 0.014890]
$\sigma_{\mathbf{e}_\theta}$	[0.022179 0.012689 0.026854]

It can be immediately noticed that there is a substantial difference between mean and median error. In particular, the latter is typically ~ 3 times smaller, both in terms of translation and rotation errors. This immediately highlights the presence of pose outliers, which are small in number yet with an error that is orders of magnitude larger compared to the extremely accurate detections that nominally take place.

This difference between mean and median error is less pronounced for the t_x, t_y translation components. As illustrated more in detail in the remainder of this section, this behavior has been traced back to the successful RoI-based correction of the translation vector, at least in terms of (x, y) components. The correction of the relative distance component along the boresight direction proved extremely beneficial when completely wrong poses were detected, although there still exist a significant degree of uncertainty due to the fact that this approximate estimation of t_z is based on measuring the size of the detected RoI. At a given distance, the latter may nevertheless vary in a relatively wide range, depending on the S/C's attitude.

For what concerns rotation errors, the estimation of the Euler angle about the y-axis appears to be more accurate compared to the two other components. On the contrary, the θ_z rotation is the one affected by the largest degree of uncertainty and is also the one for which the gap between mean and median performance is most evident. It has been conjectured that this effect is closely related to the distribution of the chosen keypoints, relative to the geometric center of the S/C. Indeed, one may compute a quantity that is analogous to the moment of inertia of a set of points:[§§] to each keypoint, a weight equal to its mean detection confidence across the entire dataset is assigned, whereas the square distance is computed with respect to an (x, y, z) frame having its origin at the geometric center of the S/C. This leads to the definition of the corresponding three quantities, whose values are $I_{xx} = 1.518$ m^2, $I_{yy} = 1.589$ m^2, $I_{zz} = 2.736$ m^2. It can be seen that I_{zz} is almost twice as large as the two other "inertias". In addition note that, in correspondence of the same angular error, the reprojection error (measured in pixels) is higher for keypoints that are farther away from the center of the S/C. Since the pose fit is chosen based on the minimization of the reprojection error, this translates into a pose that is more prone to satisfy a wrong constraint imposed by a wrong keypoint detection that is far from the

[§§] $I_{ii} = \sum_{l=1}^{N_{\text{points}}} m^{(l)} \left[(d_j^{(l)})^2 + (d_k^{(l)})^2 \right]$ where $i, j, k = (1, 2, 3),\ (2, 3, 1),\ (3, 1, 2)$

center. In other words, whenever an outlier is present among the detected keypoints, the estimated Euler angle about the axis associated with the highest keypoint-inertia will be particularly biased towards that outlier. This event may occur whenever a landmark is mistaken for another similar one by the LRN with a high degree of certainty.

1. Estimation uncertainty

A fundamental part of the analysis is the quantification of the uncertainty affecting the estimation, given the ultimate goal of this work of proposing an RPEP that can be embedded in a navigation filter. In Figs. 13 and 14, the distribution of absolute and relative errors over a $[-3\sigma,\ +3\sigma]$ range is plotted.

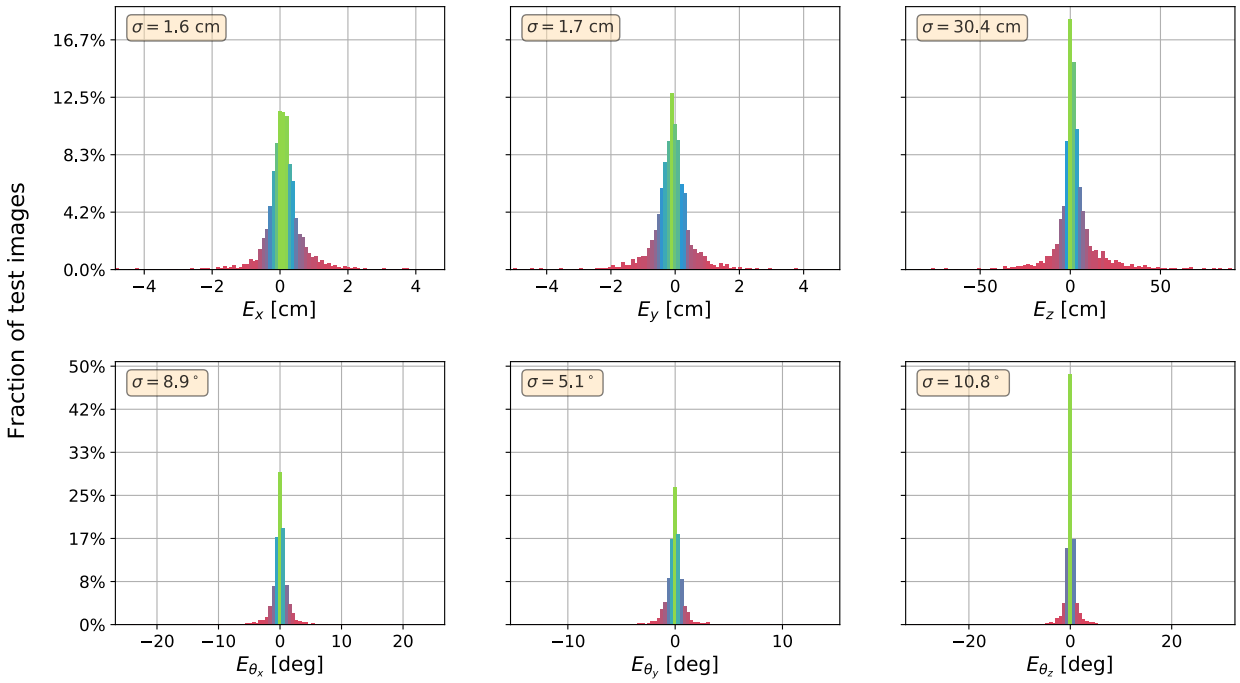


Fig. 13 Absolute error distribution

As one may expect, all 6 pose components are characterized by a normally distributed error with zero mean. In each subplot, the y-axis indicates the fraction of test images associated with a given error bin. All these distributions were plotted using 101 bins. Note that the two distributions of the lateral position errors are practically identical, both in terms of absolute and relative errors. The translation error along the boresight direction is clearly much higher, with an uncertainty that is one order of magnitude larger compared to the two other translation components.

The fact that the distributions of the three rotation error components are similar and characterized by the same order of magnitude indicates an adequate choice of the semantic keypoints. Indeed, their selection should always be aimed at breaking as much as possible the symmetry of the structure, thus avoiding attitude ambiguities, while at the same time being associated with strong visually relevant features.

Note that the error associated with θ_z is affected by a slightly larger uncertainty, which may be explained by the previous

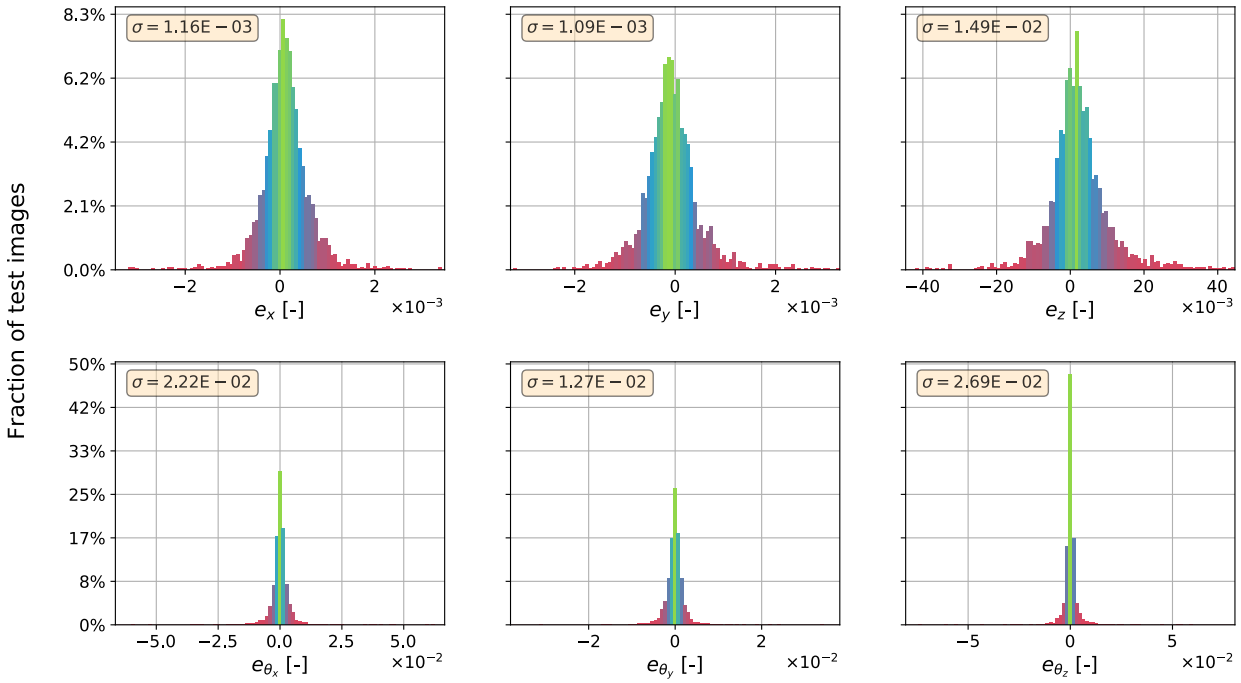


Fig. 14 Relative error distribution

conjecture.

D. Error distribution

The thresholds in terms of the 6 error components are reported in Figs. 15 and 16 as a function of the test set fraction that does not exceed them. In both figures, the top plot provides the error distribution for the entire test set, while in the bottom plot the dataset is truncated at the best 95% fraction, just to zoom-in on the dataset portion that is unaffected by outliers. In Fig. 15 the error components are plotted in semi-logarithmic scale, due to the huge difference between lateral and boresight errors. At a given distance, the latter is one order of magnitude larger than lateral errors.

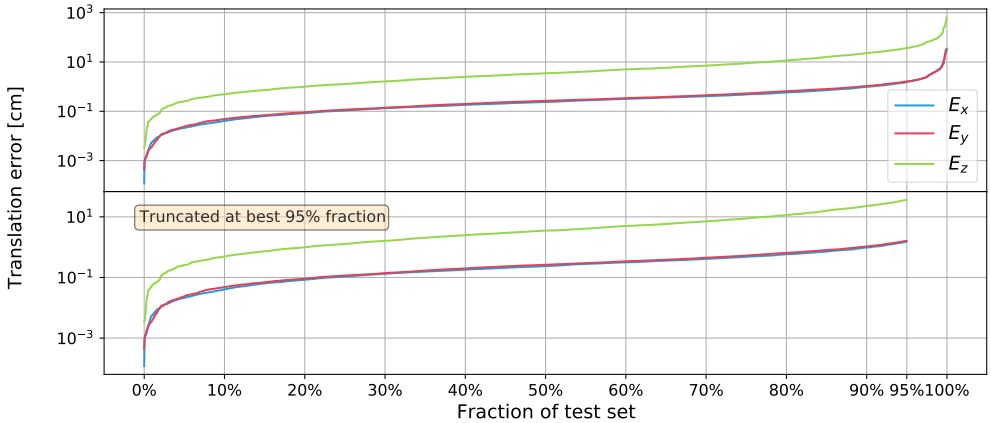


Fig. 15 Translation error distribution across the test set

In Fig. 16 the three Euler angle error components are similarly plotted, but in linear scale. It may be interesting to observe the trend of E_{θ_z}, which up to the best 70% fraction is practically coincident with E_{θ_y}. At that point, E_{θ_z} starts increasing sharply, compared to the two other components, and becomes the largest rotation error component of the 2% worst fraction of the test set. This behavior is clearly linked to the effect of outliers, whose presence, as already explained in Subsection III.C, has a more detrimental repercussion upon the estimation of θ_z.

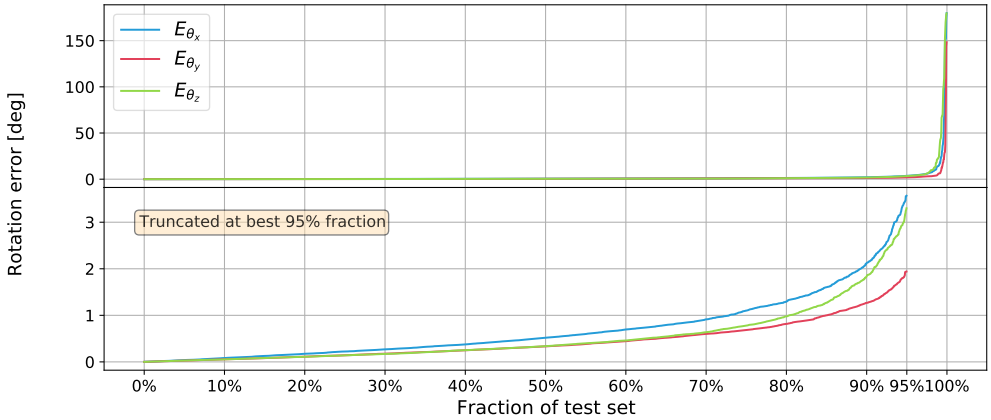


Fig. 16 Euler angle error distribution across the test set

1. Effect of relative distance

At this point, the effect of the inter-spacecraft distance upon the accuracy of the proposed RPEP is investigated. In Figs. 17 to 18 such effect can be visualized in terms of various error metrics. In particular, all test set images were first of all sorted, based on relative distance, and then grouped into 30 batches of 80 images each. For each batch, the corresponding mean performance is plotted against the mean distance. The shaded region indicates the 1σ range uncertainty, i.e. between the 15.87% and 84.13% percentiles.

In Fig. 17, the distance dependency is analyzed for what concerns the absolute translation and quaternion errors. From these two plots it is also evident that the performance of the pipeline remains practically constant ($E_t \sim 3$ cm and $E_q \sim 0.8°$) for all close-range images up to $8 \div 10$ m. This threshold corresponds to a size of the non-resized RoI of about 416 pixels, which means that, for all images taken at lower distances, there is a loss of information implicit in the downscaling to 416×416. In other words, for all images in which Tango is located at a distance $\leq 8 \div 10$ m, the degree of detail in the features that can be detected from the resized digital picture is exactly the same. For what concerns the attitude error, a sudden performance drop-off takes place at separations larger than 25 m. If only the image batches with mean distance ≤ 20 m are considered, the highest batch-errors are $E_t = 22$ cm and $E_q = 4.3°$.

As an example, the flight data from the LIRIS experiment [36, 37] demonstrated a similar pose estimation accuracy,[¶¶] compared to RPEP's nominal performance on synthetic images. The experiment resulted into a successful demonstration

[¶¶] the terminal rendezvous phase (range < 20 m) has been considered

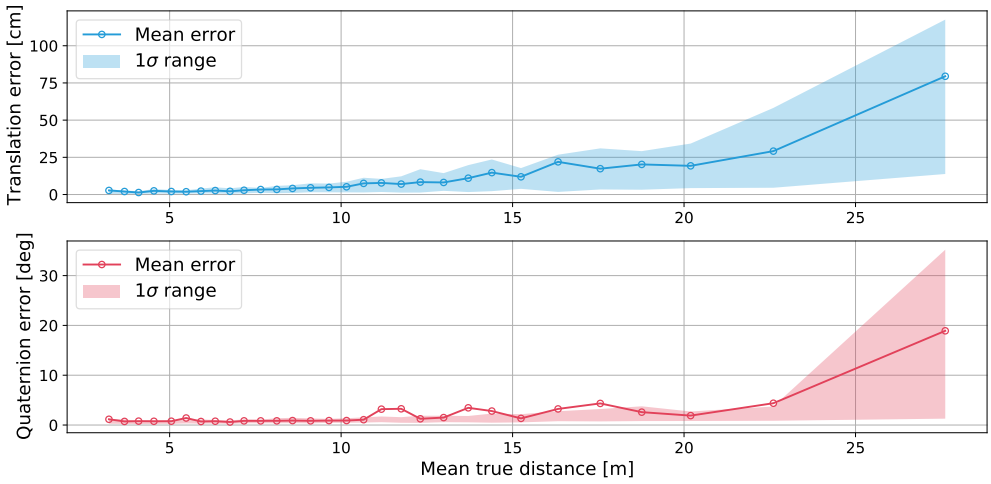


Fig. 17 Effect of inter-spacecraft distance upon absolute errors E_t and E_q

of an uncooperative Rendezvous & Docking sequence between the ATV vehicle and the International Space Station, using a combination of LiDAR and camera sensors.

The distance dependency has been analyzed also in terms of global score and relative errors. The difference between the SLAB score and our MNPE is particularly evident from Fig. 18. For a more direct comparison between the two metrics, a slightly different definition of the Normalized Pose Error compared to the one in Equation (19) has been used in the bottom plot: the mean, instead of the median, over a batch of images has been computed. Indeed, it is immediately visible that in the SLAB score, the non-normalized quaternion error is up to one order of magnitude larger than the relative translation error, which results into a score that is strongly biased towards attitude error. On the contrary, using a fully normalized score reduces this gap. This is therefore supposed to produce a more meaningful and consistent global evaluation metric.

Fig. 18 Effect of inter-spacecraft distance upon SLAB score and Normalized Pose Error

The difference between the orders of magnitude of the two error components that define the SLAB score is further stressed in Fig. 19, in which a scatter plot is provided, representing the results obtained for all the 2400 test images. The color of each dot indicates the GT distance associated with each individual image. The close-up region corresponds to the 2σ rectangle, i.e. whose sides span over the 95.44% of the related error components.

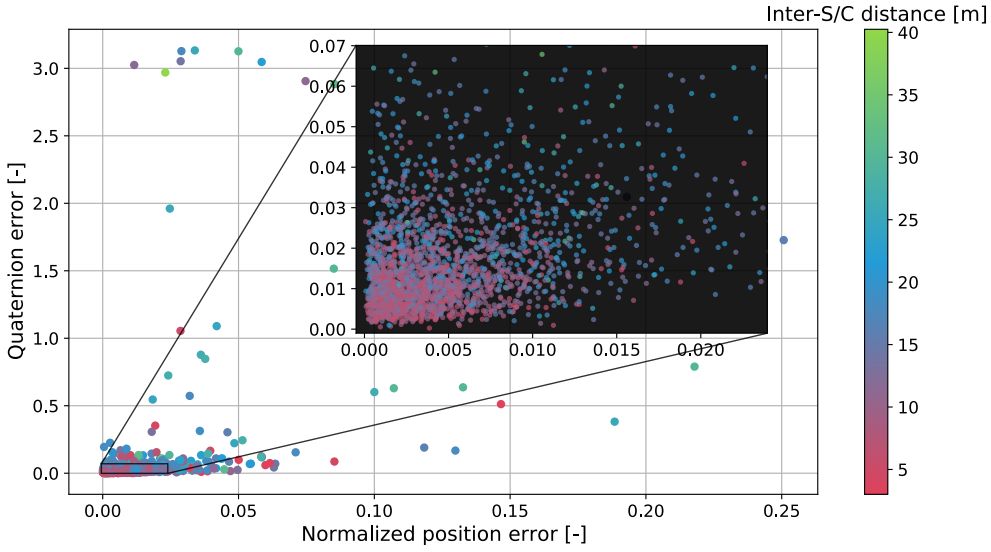


Fig. 19 SLAB error components of all test set images

2. Effect of the image background

Half of the images in our test set were rendered using a black background behind the S/C, while the other half was rendered with the presence of Earth in the image background, either in eclipse condition or not. It is then clear that, despite all the actions taken during training to improve the robustness of the proposed CNNs to a variable background, the presence of Earth in the image may still cause a performance degradation in our pipeline.

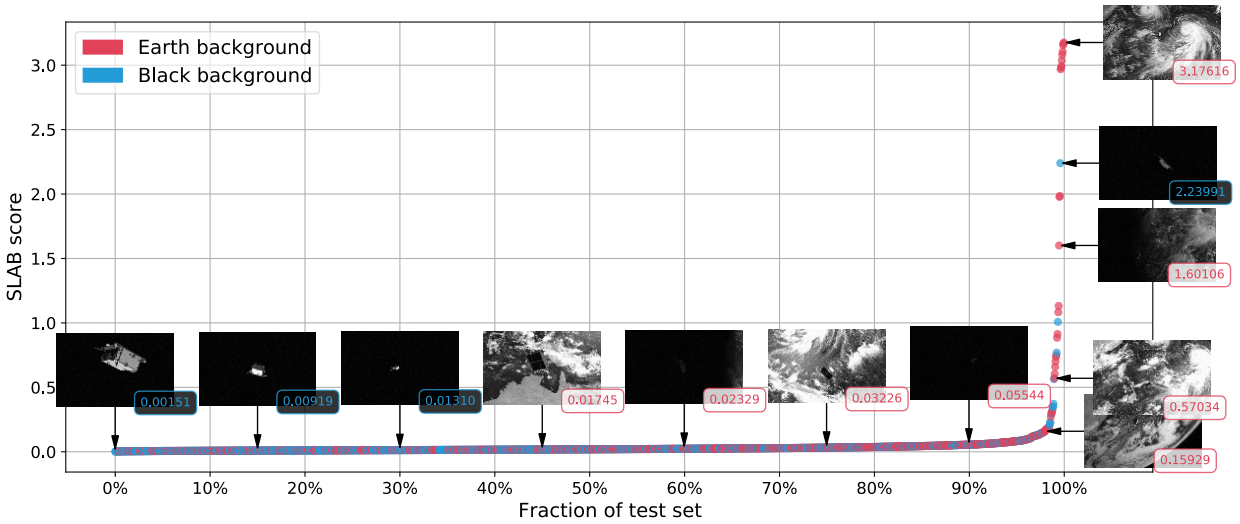


Fig. 20 Effect of the image background upon SLAB score

This is especially true whenever the target is very far from the chaser. Indeed, it can be experienced that in long-range images with the presence of Earth in the background, SLN still works exceptionally well, but LRN may sometimes struggle at properly detecting all semantic keypoints. The aforementioned performance drop is clearly visible in Fig. 20. Here, all test set images were sorted based on their individual SLAB score. Each image is represented as a blue dot if it has a black background, otherwise, if the Earth lies inside the image frame, it is reported as a red dot. It can be observed that most of the images with a very low error have a black background. On the contrary, the right-hand side of this distribution is characterized by the increasing prevalence of images with Earth in the background.

E. Benefit from iterative pose refinement

Unless a pose outlier is detected, the initial pose estimate computed using the EPnP algorithm will always be iteratively refined by employing the Levenberg-Marquardt Method (LMM). It is intuitive that, despite resulting into an improvement of the final estimate, such a strategy will also entail an increased computational cost. It was eventually concluded that the negligible impact in terms of computational burden is certainly justified by the tangible increase in accuracy, compared to exclusively relying on EPnP. For instance, on an Intel Core i7-4870HQ (2.5 GHz) CPU, the runtime associated with the EPnP algorithm only is about 10^{-6} s, while the LMM pose refinement is in the order of

10^{-4} s. In any case, the impact of the pose solver subsystem upon total runtime will remain negligible with respect to the two other subsystems of the pipeline (SLN and LRN).

In Fig. 21, the results obtained with and without LMM refinement are compared by plotting the 2σ distribution of the two normalized error components, across the whole test set. It is immediately visible that the distribution of errors obtained without pose refinement tends to be slightly shifted towards higher errors. It is then worth comparing the

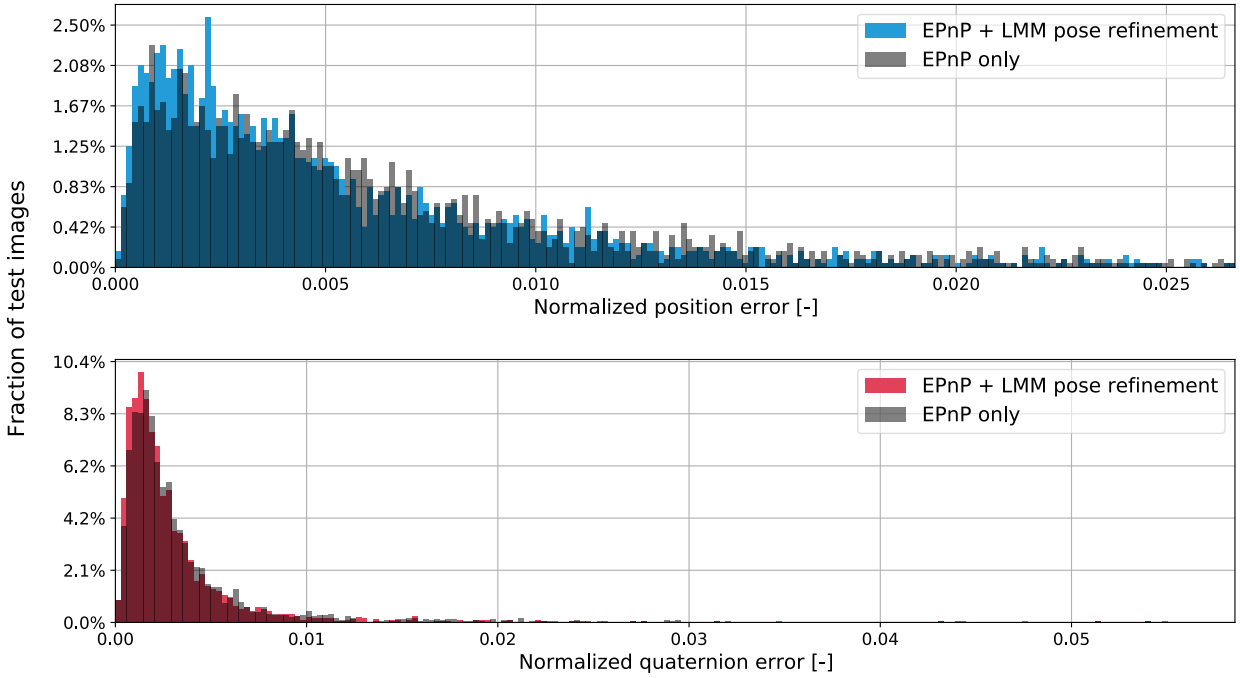


Fig. 21 Effect of pose refinement upon the normalized pose error

resulting global performance metrics, which are reported in Table 6. Note that that pose refinement allows a 12.3% reduction of the Median Normalized Pose Error (MNPE).

Table 6 Main performance metrics of the pipeline, with and without pose refinement

	EPnP + LMM	EPnP only
Mean E_t	10.36 cm	11.14 cm
Median E_t	3.58 cm	4.31 cm
Mean E_q	2.24°	2.39°
Median E_q	0.81°	0.89°
MNPE	0.00648	0.00739
SLAB score	0.04627	0.04966

F. Runtime

The entire pipeline has been tested on an NVIDIA Tesla P4 GPU, in order to evaluate runtime across the whole test set. The resulting execution times for processing each individual image are reported in Fig. 22, from which it is also possible to appreciate the order of magnitude of the computational cost associated with each of the three subsystems. The average total runtime is 0.089 s, which means that our pipeline runs at 11 FPS.

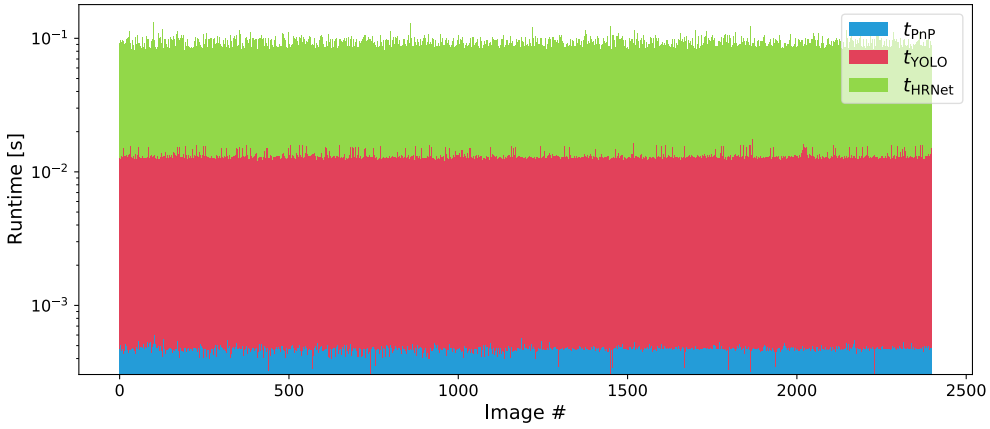


Fig. 22 Runtime breakdown, across the entire dataset

It has to be highlighted that the performance here reported is not meant to be representative of actual spaceborne hardware/software integration. This is basically due to two main reasons:

- SLN and LRN were implemented using the `PyTorch 1.6.0` framework, which allows high-level programming for quick prototyping of an ANN architecture, but cannot be clearly compared, in terms of computational efficiency, with a C/C++ implementation and the possibility of accelerating neural network inference using Field Programmable Gate Arrays (FPGAs)
- a high-end off-the-shelf GPU has been used to evaluate runtime, which clearly outperforms any space-grade hardware

The two aforementioned aspects may somehow compensate in an actual spaceborne scenario, although further investigation is clearly required. Nonetheless, with the obtained results, the computational cost of each subsystem can still be compared in relative terms. The most computationally expensive subsystem is LRN, which represents about 84% of the execution runtime. SLN and the pose solver will require instead an average time of 0.014 s and 0.0005 s, respectively.

G. Comparison with other methods

In Table 7, the S/C pose estimation performance attained by some of the top performing feature-based and deep learning-based architectures available in literature, is compared against the results achieved by our RPEP on the SPEED

dataset.

Table 7 RPEP performance comparison with other methods (SPEED dataset)

	Feature-based		**Deep learning-based**		
	SVD [7]	SVD high-conf. [7]	`UniAdelaide` [10]	`EPFL_cvlab` [34]	Ours
Mean E_t [cm]	146	51	3.2	7.3	10.36
Mean E_q [deg]	38.99	2.76	0.41	0.91	2.24
Solution availability	50%	20%	100%	100%	100%

The superiority with respect to a feature-based approach is here evident, both in terms of accuracy and availability of the solution. However, it is important to underline that the feature-based performance metrics have been obtained on real flight imagery from the PRISMA mission, while the performance of deep-learning methods corresponds to results obtained on SPEED.

The two CNN-based solutions reported in Table 7 correspond to the teams participating in the original Pose Estimation Challenge that have outperformed our solution on synthetic images, while only `EPFL_cvlab` has demonstrated higher accuracy on real test images [33].

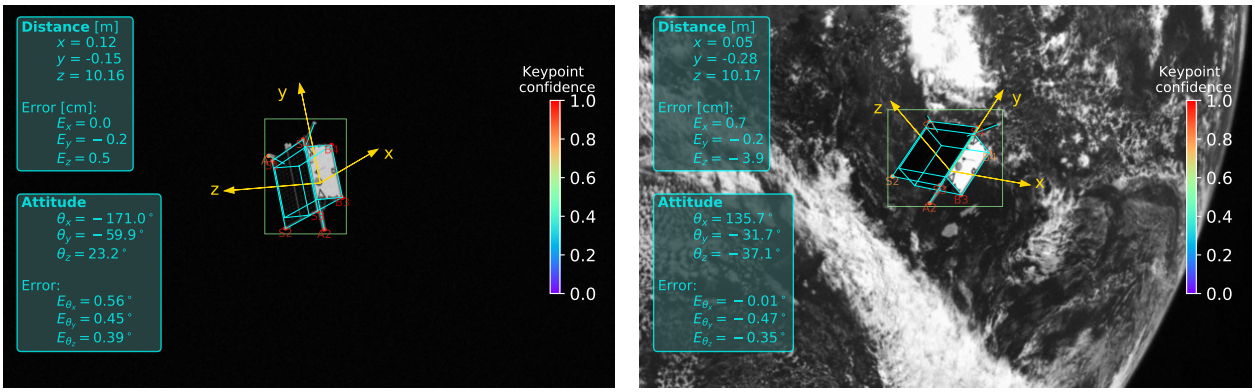
The `UniAdelaide` team proposed an approach that uses `Faster R-CNN` [29] for object detection, followed by a landmark regression stage that employs `HRNet32`. A major difference in the `HRNet` implementation lies in the large and computationally expensive input size of 768 pixels, compared to 416 pixels in our RPEP. In addition, `UniAdelaide` makes use of a Simulated Annealing scheme to progressively remove keypoint outliers while refining the pose using the Levenberg-Marquardt Method to solve the PnP problem.

The architecture designed by the `EPFL_cvlab` team employed a segmentation-driven approach described in [34]. The CNN architecture consists in two streams: one for object segmentation, while the other regresses the 2D locations of 8 landmarks of the Tango S/C. The two streams share a common encoder, which consists in the `Darknet-53` backbone of the `YOLOv3` network [38].

H. Prediction visualization

For an immediate and straightforward visualization of the pose estimation results, an apposite graphical representation has been implemented. In Figs. 23a and 23b the final estimated pose, along with the intermediate results from SLN and LRN, are represented. The two corresponding input images were randomly chosen from mid-range test images and are

somehow representative of median performance, with Earth background (Fig. 23b) and without (Fig. 23a).



(a) Mid-range test image with black background **(b) Mid-range test image with Earth background**

Fig. 23 Prediction visualization examples

In particular, in each image:

- a green rectangle delimits the RoI detected by SLN
- the subset of estimated keypoint positions*** that are fed to the pose solver is properly plotted and labeled; the color of each landmark indicates the confidence score predicted by LRN, according to the color-bar provided on the right side of the image
- a cyan wireframe model is projected onto the image, based on the final pose estimate
- the body-fixed reference frame is plotted in yellow, according to the estimated attitude; the origin of the frame is in correspondence of the center of the bottom surface of Tango
- the two text boxes on the left side of the image report the predicted pose, along with the corresponding errors with respect to the Ground Truth

Some more examples are now visualized, over a wide variety of conditions in terms of distance, illumination and background.

***the rejected low-confidence predictions are not included

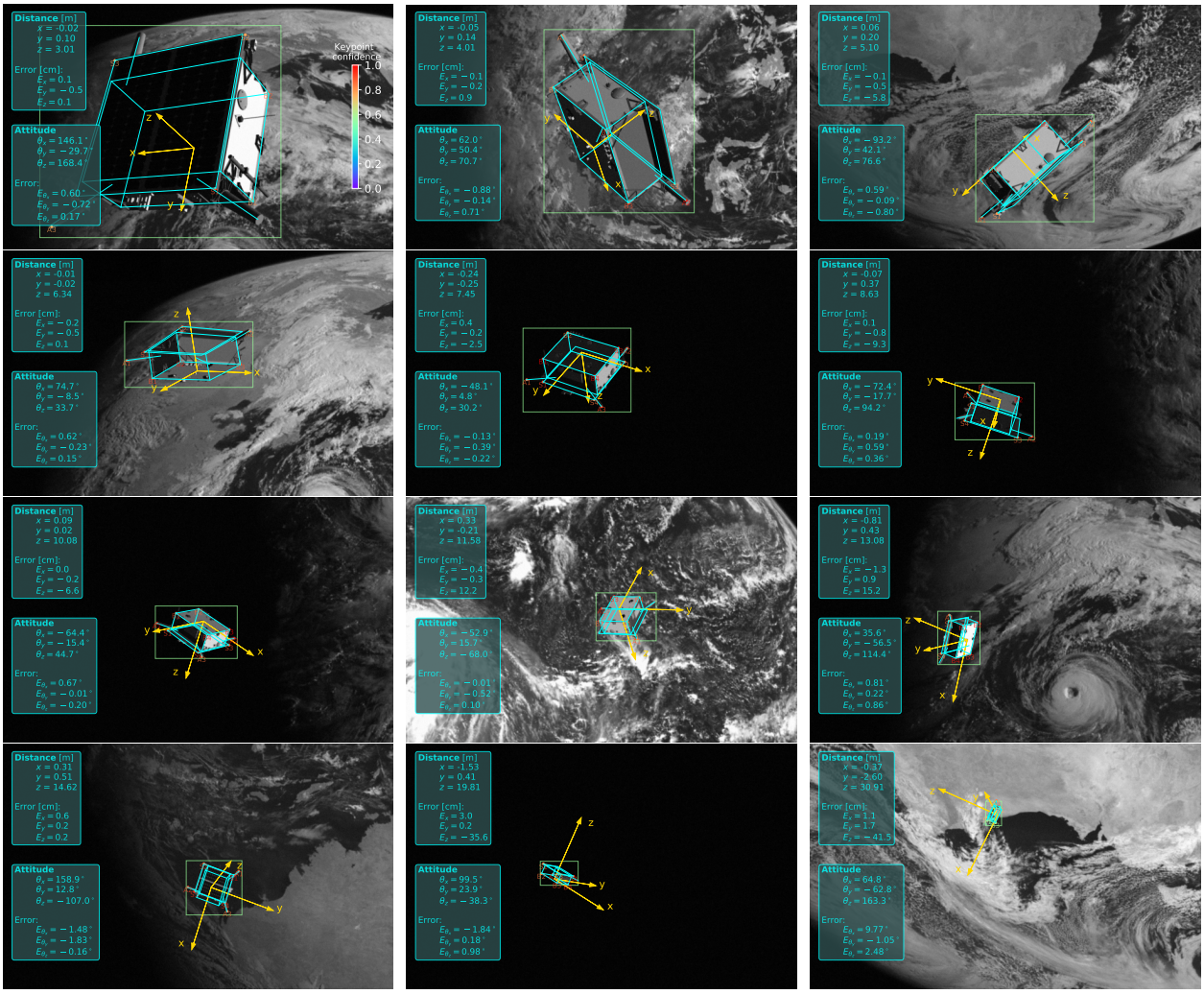


Fig. 24 Prediction visualization mosaic of test images with Earth background and increasing inter-spacecraft distance

In Fig. 24 a total of 12 random test images is displayed with the corresponding inference results. In particular, the random images were sampled from image batches, each with a given range of relative distances, and are here sorted in order of ascending distance. The 12 test images were all rendered with Earth in the background (some of them in eclipse condition). A total of 13 pose outliers out of 2400 test images has been detected and partially corrected (in terms of translation only).

In addition, Fig. 25 shows the inference results obtained on the only 5 real mockup images for which pose labels were given in the dataset. Although these labels were provided for training purposes, it was decided not to include the corresponding images in the training process, so as to use them to quantitatively evaluate the ability of our pipeline to generalize to actual imagery, with no overfitting bias. As expected, a slight performance drop takes place compared to synthetic images at a similar relative distance. Visual inspection of the superimposed wireframe model reveals very similar results in all the unlabeled remaining real images.

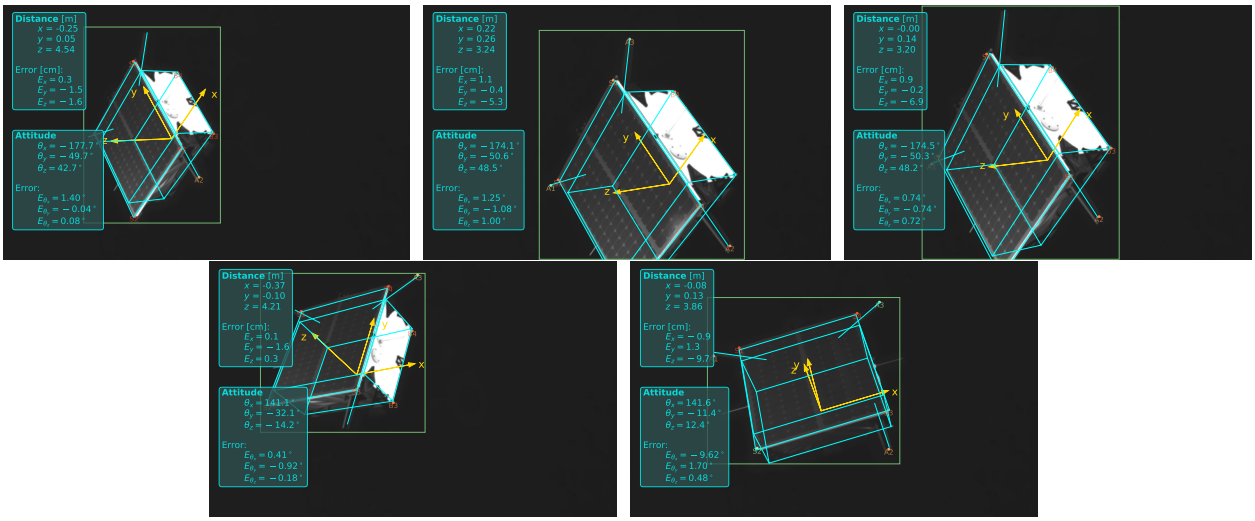


Fig. 25 Prediction visualization of the 5 real images with known pose labels

IV. Conclusions

The main contribution of this work is the development of a deep learning-based pipeline capable of estimating the relative pose of an uncooperative spacecraft from a single monocular image, provided the knowledge of the target's 3D model and with no need of any other a priori information. An RPEP was introduced, which is composed of three main subsystems:

1) Spacecraft Localization Network (SLN), which identifies in the input image the RoI, where the S/C is located. Its measured Average Precision turned out to be $AP_{50}^{95} = 98.51\%$, with a mean IoU of 95.38%.
2) Landmark Regression Network (LRN), which processes the output of the SLN to detect the position in the RoI of pre-defined semantic keypoints of the S/C. Its Average Precision, measured in terms of OKS thresholds, was $AP_{50}^{95} = 98.97\%$.
3) Pose solver, which seeks for the best pose fit of the known 3D wireframe model of the satellite, that minimizes the reprojection error.

The performance of the pipeline was tested on the synthetic images from the SPEED dataset. The architecture demonstrated to outperform the baseline developed by SLAB within the framework of the Pose Estimation Challenge. From the analysis of the results obtained on the test images in SPEED, it was concluded that the accuracy of our estimation strongly correlates with two main factors.

- Inter-spacecraft distance: there will clearly be a progressive drop in performance as the range between chaser and target increases.
- Presence of Earth in the image background: it is intuitive that images with a black background, due to the sharp contrast between the RoI and the rest of the image, will result into features that are easier to detect and hence

higher accuracy of the estimated pose.

The end-to-end performance of the pipeline, evaluated across the entire test set, corresponds to an absolute translation error of 10.36 cm (mean) and 3.58 cm (median), while the quaternion error is 2.24 deg (mean) and 0.81 deg (median). One of the most important aspects that requires further investigation is the problem of outlier correction, with the aim of reducing the gap between mean and median performance.

In a rendezvous scenario, the availability of sequential information can substantially improve outlier correction, compared to our simple RoI-based correction: given that pose outliers are easily identified, a Kalman filter could for instance propagate the previous state estimate while skipping the update step. Such a strategy is expected to work well in the event of isolated outliers, however, a longer sequence may cause the filter to diverge.

The problem of outlier correction may also be addressed statically, that is, by means of an algorithm that detects and rejects keypoint outliers from individual frames, such as the RANdom SAmple Consensus (RANSAC) algorithm [39]. In order to achieve the desired level of robustness, a blend between sequential and static outlier correction will likely be required.

Future work will be devoted to take the necessary steps in the roadmap to spaceborne implementation of a fully vision-based relative navigation system, which will include at least: the performance evaluation in a dynamic rendezvous scenario by feeding the output of the pipeline to a navigation filter; the implementation of an algorithm for identifying individual keypoint outliers; the implementation of data augmentation techniques to randomize the S/C's texture[†††] in the images used for training; the embedded implementation of the RPEP on hardware that is representative of performance attainable on space-grade onboard computers (with use of CNN hardware acceleration), eventually followed by hardware-in-the-loop testing.

References

[1] Bamann, C., and Hugentobler, U., "Accurate orbit determination of space debris with laser tracking," *Proceedings of the 7th European Conf. on Space Debris, edited by Flohrer T. and Schmitz F., ESA Space Debris Office, Darmstadt, Germany*, 2017.

[2] Lyu, J.-T., Zhong, W.-J., Liu, H., Geng, Y., and Ben, D., "Novel approach to determine spinning satellites' attitude by RCS measurements," *Journal of Aerospace Engineering*, Vol. 34, No. 4, 2021, p. 04021023. https://doi.org/10.1061/(ASCE)AS.1943-5525.0001253.

[3] Wetterer, C. J., and Jah, M., "Attitude determination from light curves," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 5, 2009, pp. 1648–1651. https://doi.org/10.2514/1.44254.

[4] Reed, B. B., Smith, R. C., Naasz, B. J., Pellegrino, J. F., and Bacon, C. E., "The restore-L servicing mission," *AIAA space 2016*, 2016, p. 5478. https://doi.org/10.2514/6.2016-5478.

[†††] the underlying idea behind domain randomization [40] is that, by augmenting the training set with images where the S/C texture is randomized in a non-realistic fashion, the CNNs are forced to learn the global shape of the object rather than focusing on local texture, thus making the pipeline more robust to mismatches between reality and synthetic images

[5] D'Amico, S., Benn, M., and Jørgensen, J. L., "Pose estimation of an uncooperative spacecraft from actual space imagery," *International Journal of Space Science and Engineering 5*, Vol. 2, No. 2, 2014, pp. 171–189. https://doi.org/10.1504/IJSPACESE.2014.060600.

[6] Bodin, P., Noteborn, R., Larsson, R., Karlsson, T., D'Amico, S., Ardaens, J. S., Delpech, M., and Berges, J.-C., "The prisma formation flying demonstrator: Overview and conclusions from the nominal mission," *Advances in the Astronautical Sciences*, Vol. 144, No. 2012, 2012, pp. 441–460.

[7] Sharma, S., Ventura, J., and D'Amico, S., "Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous," *Journal of Spacecraft and Rockets*, Vol. 55, No. 6, 2018, pp. 1414–1429. https://doi.org/10.2514/1.A34124.

[8] Capuano, V., Alimo, S. R., Ho, A. Q., and Chung, S.-J., "Robust features extraction for on-board monocular-based spacecraft pose acquisition," *AIAA Scitech 2019 Forum*, 2019, p. 2005. https://doi.org/10.2514/6.2019-2005.

[9] Park, T. H., Sharma, S., and D'Amico, S., "Towards Robust Learning-Based Pose Estimation of Noncooperative Spacecraft," *arXiv preprint arXiv:1909.00392*, 2019. https://doi.org/10.48550/arXiv.1909.00392.

[10] Chen, B., Cao, J., Parra, A., and Chin, T.-J., "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0. https://doi.org/10.1109/ICCVW.2019.00343.

[11] Proença, P. F., and Gao, Y., "Deep learning for spacecraft pose estimation from photorealistic rendering," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 6007–6013. https://doi.org/10.1109/ICRA40945.2020.9197244.

[12] Sharma, S., Beierle, C., and D'Amico, S., "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," *2018 IEEE Aerospace Conference*, IEEE, 2018, pp. 1–12. https://doi.org/10.1109/AERO.2018.8396425.

[13] Sharma, S., Beierle, C., and D'Amico, S., "Towards Pose Determination for Non-Cooperative Spacecraft Using Convolutional Neural Networks," *Proceedings of the 1st IAA Conference on Space Situational Awareness (ICSSA)*, 2017, pp. 1–5.

[14] Sharma, S., and D'Amico, S., "Neural Network-Based Pose Estimation for Noncooperative Spacecraft Rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, 2020. https://doi.org/10.1109/TAES.2020.2999148.

[15] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., and Navab, N., "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," *Asian conference on computer vision*, Springer, 2012, pp. 548–562. https://doi.org/10.1007/978-3-642-37331-2_42.

[16] Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., and Rother, C., "Learning 6d object pose estimation using 3d object coordinates," *European conference on computer vision*, Springer, 2014, pp. 536–551.

[17] Tekin, B., Sinha, S. N., and Fua, P., "Real-Time Seamless Single Shot 6D Object Pose Prediction," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2018, pp. 292–301. https://doi.org/10.1109/CVPR.2018.00038.

[18] Lepetit, V., Moreno-Noguer, F., and Fua, P., "EPnP: An Accurate O(n) Solution to the PnP Problem," *International journal of computer vision*, Vol. 81, No. 2, 2009, p. 155. https://doi.org/10.1007/s11263-008-0152-6.

[19] Bukschat, Y., and Vetter, M., "Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach," *arXiv preprint arXiv:2011.04307*, 2020. https://doi.org/10.48550/arXiv.2011.04307.

[20] Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N., "SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again," *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 1530–1538. https://doi.org/10.1109/ICCV.2017.169.

[21] Sundermeyer, M., Marton, Z.-C., Durner, M., Brucker, M., and Triebel, R., "Implicit 3D Orientation Learning for 6D Object Detection from RGB Images," *arXiv preprint arXiv:1902.01275*, 2019. https://doi.org/10.48550/arXiv.1902.01275.

[22] Sun, K., Xiao, B., Liu, D., and Wang, J., "Deep high-resolution representation learning for human pose estimation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 5693–5703. https://doi.org/10.1109/CVPR.2019.00584.

[23] Chen, B., Parra, Á., Cao, J., Li, N., and Chin, T.-J., "End-to-End Learnable Geometric Vision by Backpropagating PnP Optimization," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020, pp. 8097–8106. https://doi.org/10.1109/CVPR42600.2020.00812.

[24] Peng, S., Liu, Y., Huang, Q., Zhou, X., and Bao, H., "PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019, pp. 4556–4565. https://doi.org/10.1109/CVPR.2019.00469.

[25] Song, C., Song, J., and Huang, Q., "HybridPose: 6D Object Pose Estimation Under Hybrid Representations," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020, pp. 428–437. https://doi.org/10.1109/CVPR42600.2020.00051.

[26] Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., and Birchfield, S., "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018. https://doi.org/10.48550/arXiv.1809.10790.

[27] Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[28] Girshick, R., "Fast r-cnn," *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[29] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, 2015, pp. 91–99.

[30] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.

[31] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "Ssd: Single shot multibox detector," *European conference on computer vision*, Springer, 2016, pp. 21–37. https://doi.org/https://doi.org/10.1007/978-3-319-46448-0_2.

[32] Tan, M., Pang, R., and Le, Q. V., "Efficientdet: Scalable and efficient object detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10781–10790.

[33] Kisantal, M., Sharma, S., Park, T. H., Izzo, D., Märtens, M., and D'Amico, S., "Satellite Pose Estimation Challenge: Dataset, Competition Design and Results," *IEEE Transactions on Aerospace and Electronic Systems*, 2020. https://doi.org/10.1109/TAES.2020.2989063.

[34] Hu, Y., Hugonot, J., Fua, P., and Salzmann, M., "Segmentation-Driven 6D Object Pose Estimation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019, pp. 3380–3389. https://doi.org/10.1109/CVPR.2019.00350.

[35] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. https://doi.org/10.48550/arXiv.1412.6980.

[36] Masson, A., Haskamp, C., Ahrns, I., Brochard, R., Duteis, P., Kanani, K., and Delage, R., "Airbus DS Vision Based Navigation solutions tested on LIRIS experiment data," *ESA 7th Space Debris Conference*, 2017.

[37] Cavrois, B., Vergnol, A., Donnard, A., Casiez, P., and Mongrard, O., "LIRIS demonstrator on ATV5: a step beyond for European non cooperative navigation system," *AIAA guidance, navigation, and control conference*, 2015, p. 0336. https://doi.org/10.2514/6.2015-0336.

[38] Redmon, J., and Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. https://doi.org/10.48550/arXiv.1804.02767.

[39] Fischler, M. A., and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395. https://doi.org/10.1145/358669.358692.

[40] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P., "Domain randomization for transferring deep neural networks from simulation to the real world," *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2017, pp. 23–30. https://doi.org/10.1109/IROS.2017.8202133.