Acta Astronautica 215 (2024) 20–43

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research Paper

Robust spacecraft relative pose estimation via CNN-aided line segments detection in monocular images

Michele Bechini [a,*], Geonmo Gu [b], Paolo Lunghi [a], Michèle Lavagna [a]

[a] Politecnico di Milano, Department of Aerospace Science & Technology, via La Masa 34, 20156 Milano, Italy
[b] NAVER Vision, 6 Buljeong-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, South Korea

ARTICLE INFO

Keywords:
Relative pose estimation
Vision-based navigation
Convolutional neural networks
Synthetic image datasets
Line segment detection
Wireframe

ABSTRACT

Autonomous spacecraft relative navigation via monocular images became a hot topic in the past few years and, recently, received a further push thanks to the constantly growing field of artificial neural networks and the publication of several spaceborne image datasets. Despite the proliferation of spacecraft relative-state initialization algorithms developed, most architectures adopt computationally expensive solutions relying on convolutional neural networks (CNNs) that provide accurate output at the cost of a high computational burden that seems unfeasible for current spaceborne hardware. The paper addresses this issue by proposing a novel pose initialization algorithm based on lightweight CNNs. Inspired by previous state-of-the-art algorithms, the developed architecture leverages a fast and accurate target detection CNN followed by a line segment detection CNN capable of running with low inference time on mobile devices. The line segments and their junctions are grouped into complex geometrical groups, reducing the solution search space, and subsequently, they are adopted to extract the final pose estimate. As a main outcome, the analyses demonstrate that the lightweight architecture developed scores high accuracy in the pose estimation task, with a mean estimation error of less than 10 cm in translation and 2.5° in rotation. The baseline algorithm scores a mean SLAB error of 0.04552 with a standard deviation of 0.22972 in the test dataset. Detailed analyses demonstrate that the uncertainties on the overall pose score are driven mainly by errors in the relative attitude, which gives the highest contribution to the pose error metric adopted. The analyses on the error distributions point out that the uncertainties on the estimated relative position are higher in the camera boresight axis direction. Concerning the relative attitude, the algorithm proposed has higher uncertainties in estimating directions of the target x and y axes due to ambiguities related to the target geometry. Notably, the target detection CNN trained in this work outperforms the previous top scores in the benchmark dataset. The performances of the proposed algorithm have been investigated further by analyzing the effects on the accuracy due to the relative distance and the presence of background in the images. Lastly, the paper delves into the possibility of adopting a sub-portion of the 2D-to-3D match matrix made by the most complex perceptual groups identified that positively affects the overall run-time, pointing out the performances in terms of accuracy of the estimates and providing a comparison of both the baseline and the reduced match matrix versions against state-of-the-art algorithms concerning relative position and attitude errors and solution availability, highlighting the high accuracy and solution availability of the proposed architectures.

1. Introduction

Autonomous spacecraft relative navigation is an enabling technology for incoming space missions, aimed at performing time-critic tasks in a wide range of scenarios with a high level of autonomy required [1]. Key applications where autonomous navigation can be adopted include spacecraft rendezvous and docking [2], proximity operations (active debris removal and on-orbit servicing included) [3–5], and landing [6, 7]. All these scenarios require precise relative navigation to accomplish the mission. The same holds for formation flying where there is the need to maintain the prescribed formation [8] or in case the spacecraft operating in proximity of a space-resident object needs to perform collision avoidance maneuvers ensuring safety [9]. The relative state estimation process must be solved autonomously onboard in all these

* Corresponding author.
E-mail addresses: michele.bechini@polimi.it (M. Bechini), korgm403@gmail.com (G. Gu), paolo.lunghi@polimi.it (P. Lunghi), michelle.lavagna@polimi.it (M. Lavagna).

https://doi.org/10.1016/j.actaastro.2023.11.049
Received 22 June 2023; Received in revised form 23 October 2023; Accepted 27 November 2023
Available online 29 November 2023

scenarios to ensure the reactivity and effectiveness required with robustness in nominal and off-nominal operations. The first step in the relative navigation chain is the initialization of the relative position and attitude (i.e., pose). Spacecraft relative pose initialization refers to the determination of the position and orientation of a spacecraft relative to another in space, typically without any a priori knowledge of the full relative state. The accuracy in this task is needed since the reduction of the error in the initial relative pose estimation brings a drop in the navigation error and also in the entire guidance navigation and control (GNC) chain as a consequence. The most attractive and challenging scenario for relative pose estimation is for a target–chaser system where the target is noncooperative, i.e., there is no communication link between the two spacecraft and the target is not equipped with light-emitting markers [10]. It should be pointed out that in principle, the relative state can also be estimated by using tracking methods from the ground but the success depends on the visibility of the spacecraft from the ground stations and the resolution of their sensors, and the accuracy is strongly affected by uncertainties in the state estimation even for advanced techniques [11,12]. These limitations make the ground-based approach not suited for all the scenarios mentioned before. Accordingly, the relative state estimation process must be solved onboard by relying on the sensors and onboard computers available on the chaser.

Monocular cameras are one of the most adopted imaging sensors to acquire meaningful measurements fed to the GNC chain [10] that processes the acquired images to reconstruct the relative pose between the camera and the target. The widespread use of monocular cameras is due to the low power consumption, cost, and mass requirements of these sensors compared to more complex ones, such as light detection and ranging (LiDAR) [13]. Among the available cameras, the most studied and employed are those operating in the visible spectrum, with applications to both cooperative [14] and uncooperative [15] missions. Despite that, it must be noticed that solutions involving cameras operating also in the thermal infrared (TIR) region of the spectrum are currently under development for applications in low-illumination conditions [16,17].

During the last few years, several architectures have been proposed to solve the relative pose estimation problem via monocular images. This increasing interest is motivated primarily by the publication of the Spacecraft Pose Estimation Dataset (SPEED), one of the first validated spaceborne synthetic image datasets made publicly available, [18], and also by the international Satellite Pose Estimation Competition (SPEC2019) [19]. After that, other spaceborne image datasets were released and a new international competition (SPEC2021) was held [20]. Thanks to this, a wide range of approaches have been studied. Despite that, most of the top-scoring algorithms have been designed disregarding the deployability while maximizing the accuracy in the pose estimation task and resulting in architectures with a high number of parameters (most of which are usually devoted to CNN for features extraction), requiring abundant computational resources [21]. Due to this, most of those algorithms are currently prohibitive for spacecraft on-board computers with constrained resources [21,22].

In contrast, a novel lightweight and robust relative pose estimation pipeline that can run with low inference time on CPUs is proposed here. Our architecture is inspired by a state-of-the-art algorithm, the Sharma–Ventura–D'Amico (SVD) [23]. The baseline SVD algorithm was improved here by a feature detection step that uses a lightweight line segment detection convolutional neural network (CNN), named M-LSD, capable of running in real-time on mobile devices [24] instead of the Hough transform [25] used in the original implementation. The main steps of the proposed pipeline are the target detection, the line segments extraction, and the solution of the Perspective-*n*-Point (PnP) problem to retrieve the relative pose estimate. The performances of the baseline architecture proposed have been tested on the SPEED dataset, used as a benchmark, evaluating both the accuracy and the overall running time. To get more insights into the behavior of the pose initialization scheme developed, the effects of the relative distance and the presence of backgrounds, have been evaluated. Moreover, the baseline architecture is compared against a variant that, in the relative pose estimation task from the detected 2D feature groups and the known 3D feature groups, leverages on 2D-to −3D feature groups correspondences stored in a matrix (named *match matrix* in [23]) with reduced dimensions relying only on most complex feature groups identified in the image being processed. To conclude, the architectures proposed here are compared against other pose initialization algorithms that participated in SPEC2019 in terms of relative translation and attitude errors and solution availability. As the main outcome, the proposed architecture achieves performances comparable with top-scoring architectures in SPEC2019 despite employing lightweight CNNs. It surpasses the SVD algorithm in solution accuracy and availability while reducing computational time. The possibility of adopting the reduced form of the match matrix is demonstrated via dedicated analyses, enabling faster pose retrieval without compromising accuracy. Furthermore, the adopted YOLO (You Only Look Once) target detectors exhibit superior performance with respect to the previous state-of-the-art on SPEED images.

In the remainder of the paper, related work is discussed in detail in Section 2, providing an extensive review of algorithms and methods related to the topics of relative pose initialization and spaceborne image processing. The proposed architecture is outlined in Section 3, providing a comprehensive description of its main components, as well as the training of the involved CNNs, while the results achieved with the selected baseline architecture are provided and discussed in Section 4. In detail, Section 4.1 reports the evaluation metrics adopted, while the results from the participation in the postmortem SPEC2019 are discussed in Section 4.2. The performances of the baseline architecture are analyzed in Section 4.3, and Section 4.4 report the comparison against other algorithms that participated in SPEC2019. Lastly, the conclusions and the hints on possible future developments are given in Section 5.

2. Related works

2.1. Relative pose initialization methods

A general architecture for relative pose estimation can be split into two consecutive steps, i.e., the acquisition step (or initialization) and the tracking phase. During the initialization, there is no a priori knowledge of the target–chaser relative state, while during the tracking phase, the relative state information retrieved at previous instants of time is combined with the current measures extracted from new images of the target. Several classifications of the pose estimation algorithms have been proposed, depending on the approach followed in both steps. From a high-level perspective, the estimation methods can be categorized as non-model-based, model-based, and hybrid [13].

Non-model-based algorithms do not rely on the knowledge of a 3D model of the target. Among those algorithms, the appearance-based ones leverage only the appearance or the textures of the target spacecraft in the acquired images compared with a pre-computed database, without extracting features, as in [26,27]. The approach used in appearance-based methods can also be adopted if the 3D model of the target is available and features are extracted from the 3D model to construct a database used to search for correlations with features from the acquired images [28]. This approach is commonly named template matching in computer vision [28].

Model-based algorithms rely on the knowledge of the 3D geometry model of the target, which is then partially known. The camera mounted on the chaser acquires a 2D image that is processed via Image Processing (IP) algorithms to extract features that subsequently are matched with the corresponding elements of the 3D model available. Once the 2D-to −3D correspondences are known, the relative pose is retrieved by solving the PnP problem. Please notice that an additional

step of pose refinement can be added to the pose estimation pipeline to improve accuracy and computational efficiency [10].

Hybrid algorithms merge model-based and non-model-based approaches, exploiting both the features extracted from the image and the appearance to generate the relative pose estimate, as in [29]. This approach leverages the detected keypoints to generate a first subset of possible correspondences between the known 3D model and the image. A rough relative pose estimate is extracted from the subset and used to verify and iteratively add new 2D-to −3D correspondences to the initial subset up to a threshold value. Finally, the relative pose estimate fed to the navigation filters is retrieved from the last set of correspondences defined during the iterative process. IP algorithms designed to extract features from images have been studied on a large scale over the years and several methods are therefore documented in the literature. The features extracted from 2D images can range from simple points to complex geometrical structures. One of the first applications of visual-based navigation for uncooperative spacecraft, during the Hubble Space Telescope Servicing Mission 4, leveraged two IP algorithms developed to detect both edges and predefined keypoints [30]. The edge detection was performed using a Sobel filter, and the keypoints were extracted from the digital correlation of images [30]. There are several algorithms capable of extracting keypoints and corners from 2D images, ranging from the Harris corner detector [31] to the more recent ORB [32], i.e., Oriented FAST (Features from Accelerated Segment Test [33]) and Rotated BRIEF (Binary Robust Independent Elementary Features [34]). Notably, ORB was proven in [32] to be more computationally efficient than both SURF (Speeded-Up Robust Features) [35] and SIFT (Scale Invariant Feature Transform) [36]. A comprehensive literature review concerning the comparison of keypoints detectors is available in [37–39]. Strong variations of scale and perspective of the target in the images are expected during a close-range approach between spacecraft. Hence, relying on keypoints invariant with respect to these spatial effects [35,36] is well suited for these scenarios. Despite this, keypoint features are sensitive to severe changes in illumination conditions, target partial occlusions, and background objects [13]. In these scenarios, the robustness offered by corners and edge detectors (as Canny edge detector [40] and Hough Transform [25]) is higher [10]. The capabilities of a pose estimation algorithm based on the Canny edge detector and the Hough transform for line extraction were assessed in [41] by testing the pose initialization scheme on actual spaceborne images from the Prototype Research Instruments and Space Mission technology Advancement (PRISMA) mission [42]. The results in [41] show robustness issues in case of adverse illumination conditions and presence of background in images. To overcome the limitations of both keypoints and edge detectors, it was proposed in [43] to couple the Robert Cross Method for edge detection with the Harris corner detector, improving the computational efficiency of the IP chain applied to thermal images only. Concerning visible images, the same concept of fusing edge and corner detector was applied by [44], where a combination of three different detectors (i.e., one corner detector and two line segment detectors) is applied to increase the robustness of the IP presented in [23]. The last step involved in Model-based algorithms is the PnP problem solution. This task is achieved by using the Efficient PnP (EPnP) solver [45], that in [23] was proven to be the less computationally demanding among other solvers, with the highest success rate and accuracy. Hence, the EPnP is particularly suited for applications in which a high number of 2D to 3D correspondences need to be tested. Despite that, the EPnP can lead to sub-optimal solutions of the PnP problem (i.e., local minima) hence, usually, it is coupled with numerical solvers as Newton–Raphson [41] or Levenberg–Marquardt [46] methods to optimize the relative pose estimate. For an extensive review of possible pose estimation schemes, the readers are referred to [10,13] while [47] offers a more comprehensive surveys on PnP solvers and their comparison.

Machine Learning and mostly Convolutional Neural Networks (CNNs) have been applied in the light of monocular-based pose initialization and tracking in the last few years mainly due to their capabilities in image classification [48]. The main advantages of CNNs over more classic feature-based approaches are the increased robustness in low illumination conditions and against low signal-to-noise ratio in images, and the reduced computational complexity [49]. Despite that, it is common knowledge that spaceborne images are affected by higher contrast and lower resolution with respect to terrestrial applications thus, the accuracy of state-of-the-art CNN is expected to decrease if they are applied in a space scenario [13]. At the very beginning, the CNNs were adopted in an *end-to-end* fashion, resulting in a pose estimation pipeline fully demanded to the selected CNN that is trained to learn an implicit mapping function between the 2D image taken as input and the labeled relative pose given as output [50,51]. Some variations of the direct pose regression have been proposed, as in [52], where the CNN solves a classification problem to estimate a coarse pose that is subsequently refined, or in [18] where the relative pose is the output of a CNN that solves a hybrid classification–regression problem. Notably, the CNN proposed in [18] and its improved version reported in [53] were trained in a multi-tasking configuration (i.e. region of interest extraction, heatmaps, pose regression, classification, and segmentation), improving the accuracy achieved by direct end-to-end architectures. Nonetheless, most direct regression CNNs perform worse than classic pose estimation algorithms, especially for the relative attitude [52]. A different approach is to use the CNNs to regress landmarks and keypoints [54], in which CNNs are trained to detect and recognize selected features in images [55,56]. The features extracted from the image by the CNN are matched with those from the available 3D wireframe model and used to solve the PnP problem. It is remarked here that in most of the architectures for CNN-based feature detection, the image is preprocessed with a region of interest (ROI) extraction CNN to improve the accuracy of the feature extraction and, as a consequence, of the entire pose estimation pipeline [57,58].

From the outcomes of both SPEC2019 [19] and SPEC2021 [20] it is evident that the performances in terms of accuracy of the estimated pose ensured by CNN feature detectors coupled with PnP solvers are higher than those given by direct CNN-based pose regression. Notably, we acknowledge here that despite the architectures that won both the competitions [19,59] were based on target localization and landmark regression by CNNs coupled with PnP solvers, the direct regression method based on ResNet [60] proposed in [61] achieved the third best score in SPEC2019.

2.1.1. The Sharma–Ventura–D'Amico (SVD) algorithm

The Sharma–Ventura–D'Amico (SVD) algorithm [23] is the baseline selected and improved with the architecture presented in this work. SVD belongs to the model-based pose initialization algorithms and builds on the scheme proposed in [41]. with improved robustness against backgrounds and efficiency by the introduction of the Weak Gradient Elimination (WGE). The first step of the SVD is to blur the image through a Gaussian filter. Then a two streams approach is followed. In the former, the blurred image is convolved with a Prewitt filter then the WGE is applied to threshold the weak gradient intensities that correspond to the Earth in the background. Subsequently, the Hough transform is applied to the thresholded image to extract line segments corresponding to the edges of the spacecraft. In the other parallel stream, the image is convolved with a Sobel filter and then processed via Hough transform to extract line segments without applying the WGE. The two streams are collected into a single process, and the edge features extracted are merged. The IP part of the SVD concludes with grouping detected line edges into more complex geometrical groups (i.e., proximity and parallel pairs and triads, antennas, and closed polygonal features). The same perceptual grouping applies to the 3D wireframe model of Tango. The poses are then retrieved by building a match matrix for 2D to 3D perceptual groups correspondences and then solving the PnP problem through the Efficient PnP algorithm [45] for all the matches identified. The output poses are ranked by the

lowest reprojection error achieved, and the top-5 estimates are optimized further via Newton–Raphson method. Among these final five pose estimates, the final pose given as output by the SVD is the one that minimizes the reprojection error between detected line segments endpoints and 3D wireframe keypoint reprojected on the image plane. Noticeably, one of the main advantages of the WGE is the automatic extraction of the region of interest (ROI) used for the self-tuning of the hyperparameters needed for the Hough transform and the perceptual grouping of the detected features. Despite the improvements in pose estimation accuracy and background rejection, the SVD still presents significant drawbacks that limit its applications. Concerning the ROI extraction, it has been proven that if the Earth horizon is inside the images, the WGE performs poorly. That leads to a wrong estimation of the ROI and detection of line segments that belong to the horizon and not to the spacecraft [23,44]. Please notice that a correct detection of the ROI is of paramount importance to adapt the hyperparameters of both the Hough transform and perceptual grouping hence a wrong ROI extraction can jeopardize the entire pose estimation pipeline. Other drawbacks pointed out by the authors in [23] concern the presence of duplicate or incomplete edges, and the detection of line segments that do not belong to the 3D wireframe model available. Due to these issues, the test performed on 25 images of Tango [23] shows that the SVD is capable of returning an accurate relative pose estimation only for the 20% of the images [52]. The SVD has been tested on the SPEED dataset in [62] and on fused VIS-TIR images in [63], providing in both cases an accurate solution for less than 10% of the images. Moreover, the works in [62,63] point out that the tuning of the hyperparameters in the Hough transform of the SVD is challenging for a wide dataset of images with highly variable illumination conditions and positions of the target in the images. A first improvement of the SVD features extraction has been proposed in [44]. The pipeline in [44] implemented a ROI extraction method similar to the WGE in [23], based on a Prewitt operator combined with a gradient filter. Notably, the case of the Earth horizon in images is tackled by tuning the gradient filter to more selective values. Moreover, Capuano et al. proposed an improvement of the feature extraction of the SVD by introducing a new IP scheme based on three parallel streams that rely on the Shi–Tommasi corner detector [64], the probabilistic Hough transform [65], and the line segment detector algorithm [66] respectively. The features extracted by the three streams are merged by retaining only those mutual to the three streams, compensating for the different drawbacks of each method. Subsequently, the merged keypoints are exploited to synthesize polylines corresponding to components of the target spacecraft. Despite the demonstrated improvements in feature detection due to the three-streams approach, the computational complexity of the IP coupled with a PnP solver may result to be prohibitive for current onboard computers. An overview of OBC performances is reported in [67], where the tests conducted show that, for current CPU-based OBC, a simple visual-odometry task can last in about 15 s, preventing real-time applications and highlighting the needs of lightweight IP algorithms.

2.2. Target detection and region of interest extraction

Target detection, or object detection, is a computer vision task employed to detect objects of interest in specific locations in images or videos via the definition of a bounding box [68]. The bounding box is the minimum rectangular box defined in image coordinates that completely encloses the target object identified in the image. The region of the image delimited by the bounding box is defined as the Region of Interest (ROI). Nowadays, this task is usually demanded to CNNs due to their high accuracy in detection tasks. In most cases, the CNNs for target detection also include a classification step where the objective is to predict if the image given as input contains at least an object of interest. If the target object is in the image, then the CNNs perform a regression task to predict the pixel coordinates that define the bounding box. The current state-of-the-art CNNs for target detection can be classified into region proposal methods and one-stage methods [58,68].

Region proposal methods rely on a two-stage process where the first step, named region proposal, is responsible for generating "interesting" regions of the image that can contain the target object, while the second step classifies the proposed regions and then performs bounding box regression. One of the first implementations of a region proposal object detector was the Regions with CNN features (R-CNN) [69], where the region proposal task is performed by a selective search algorithm that outputs about 2000 regions. Each region is then processed with a CNN to perform classification and bounding box regression. R-CNN is slow in inference since all the proposed regions must be processed by the CNN one per time. This issue has been addressed with the Fast R-CNN algorithm [70] by adopting a single ConvNet that convert the input image in a single feature map processed by a region proposal algorithm, lowering the inference time. Despite the strong improvements in computational time [70], the Fast R-CNN still relies on an external region proposal algorithm. This issue was addressed with the development of the Faster R-CNN [71], where the region proposal algorithm of the Fast R-CNN was substituted by a convolutional network, named region proposal network, that takes as input the feature map and outputs the proposed regions that are fed to the Fast R-CNN. A target detector based on Faster R-CNN was used in the pose initialization pipeline in [59].

Single-stage object detection CNNs perform the bounding boxes prediction and classification in a single pass through the CNN. One of the most common architectures in this category is the You Only Look Once (YOLO) [72]. The YOLO architecture, inspired by GoogLeNet, consists of stacked convolutional layers interspersed with reduction layers and followed by fully connected layers for prediction. It uses a sliding window approach to split the input image into a fixed-resolution grid. Each grid cell generates bounding boxes and confidence scores, indicating the presence and accuracy of detected items, while conditional class probabilities are computed for each grid cell. The final prediction is made using a fully connected layer and a single-level feature map from the pre-trained convolutional layers. Subsequent versions have undergone significant modifications to improve accuracy and reduce inference time. Comparisons between different YOLO releases are available in [73] and a specific comparison of YOLOv3 (from the initial YOLO series) and YOLOv5 can be found in [74]. A different architecture for single-stage detection is the Single-Shot Detector (SSD) [75]. The SSD is based on a feed-forward CNN that generates a collection of bounding boxes of fixed size. The bounding boxes are auto-scored for the presence of object classes then the non-maximal suppression step generates the final detections. The performances of later versions of YOLO are currently better than the ones scored by SSD both in terms of accuracy and inference time [68].

In this paper it has been decided to adopt YOLOv5[1] as the baseline architecture for target detection since it offers a good trade-off between inference time on CPU and mobile devices, number of parameters, Floating Point Operations (FLOPs), and also the complexity of the training process [73,74]. The YOLOv5s (small version) has been used as a target detector in [58] achieving state-of-the-art performances and outperforming the Intersection-over-Union (IoU) score registered by the Faster R-CNN used in [59].

2.3. Line segments detection algorithms

The topic of line segment detection has been widely studied in computer vision. The available line segment detection algorithms can be classified in three groups [76]: global methods, local methods, and deep learning-based methods. Global methods leverage on edge detection algorithms and then apply voting schemes, as in the Hough

[1] https://github.com/ultralytics/yolov5

Transform, to detect class of shapes, including line segments [77]. These methods entail well-known drawbacks as the fail in detecting edges due to weak gradients or the presence of false positives in regions of the image with a high density of edges. Moreover, the accurate detection of the endpoints of line segments using the Hough transform remains a challenge for complex scenarios like spaceborne images [63], due to the high sensitivity of the line segment detected to the input hyperparameters and the difficulties of tuning such algorithms for highly varying illumination conditions.

Local methods overcome some limitations of the global methods by relying on low-level image cues as strong gradient to detect interesting pixels and then progressively adding neighboring pixels based on gradient information to "build" line segments. Namely, the region growing process proposed by the LSD method [78] allows a better detection in highly populated image regions with an improved efficiency. Other methods improved the scheme proposed in [78] by leveraging on accurate detection of anchor pixel and improved edge drawing schemes [79–81] reaching impressive performances also in case of devices with low computational capacity [76]. Despite that, these hand-crafted methods are sensitive to noise and require careful parameter tuning.

In recent years, deep learning has emerged as a promising approach for edge detection. One notable method is the Holistically-Nested Edge Detection (HED) algorithm [82], which frames the edge detection as a pixel-wise binary classification problem, with proved superior performances compared to traditional methods. Subsequently, a number of edge detection methods have been proposed [83,84]. However, while these methods generate edge maps, they lack explicit geometric information necessary for compact environment representation and accurate localization of line segments. As a result, a post-processing step is required, which can be computationally expensive. Recently, Huang et al. [85] introduced a learning-based approach to line segment detection by proposing a large-scale Wireframe dataset. Their method, called DWP, employs two parallel branches to predict junction maps and line heatmaps, which are then merged to generate line segments. Other methods, such as PPGNet [86] and L-CNN [87], use a point-pair graph representation for line segments, but require an additional classifier to determine if the predicted point pairs correspond to the endpoints of a line segment. Similarly, AFM and HAWP [88,89] utilize attraction field maps from raw images to localize line segments, but still rely on an extra classifier. Although learning-based methods offer advantages over hand-crafted methods, their two-step strategy can limit real-time performance and require heuristic post-processing. To address these limitations, TP-LSD [90] proposed a tri-point-based line representation. However, its use is limited on edge devices due to its large model size based on a stacked hourglass architecture requiring significant computational resources. To overcome the limitations of a high computational burden, the M-LSD [24] has been proposed as a real-time deep learning-based line segment detector specifically designed for edge devices, making it a lightweight and efficient option for resource-constrained environments. Unlike other methods, the network architecture in M-LSD is highly efficient and does not require additional post-processing steps to generate line segment predictions. These characteristics makes the M-LSD a good candidate for spacecraft relative navigation applications, hence it has been adopted as baseline line segments detector in the work here presented.

2.4. Spaceborne image datasets and labels

To improve the accuracy of pose initialization algorithms through CNN-aided methods, labeled image datasets are needed for a proper training phase. The first publicly available dataset was the SPEED dataset [91]. This dataset was adopted for SPEC2019 and includes 15000 synthetic images of Tango rendered with an OpenGL-based tool and 300 mock-up images acquired from the Testbed for Rendezvous and Optical Navigation (TRON) [92]. The images are grayscale with a resolution of 1920 × 1200 pixels and labeled with relative pose only for a subset of 12000 images. The synthetic images have been validated against actual Tango images from the mission PRISMA through histogram comparison [19]. During the SPEC2019 competition also the URSO (Unreal Rendered Spacecraft On-Orbit) dataset [93] was released. The 15000 images in URSO are split in 10000 images of Soyuz and 5000 images of Crew Dragon spacecraft, all rendered by using Unreal Engine 4. The images are RGB with a resolution of 1280 × 960 pixels, with the relative pose labeled for each frame. The URSO dataset has been validated qualitatively only by visually comparing rendered images with actual Soyuz and Crew Dragon pictures [61]. In the context of SPEC2021, an improved version of SPEED was released and named SPEED+ [94]. SPEED+ is made of 59960 synthetic images of Tango rendered with OpenGL, 6740 mock-up images acquired with HIL simulating the Earth albedo (named *lightbox*), and 2791 mock-up images acquired with HIL simulating the direct sun illumination (named *sunlamp*). The images are grayscale with the same resolution as SPEED, while the relative pose labels are available only for the synthetic images [20]. More recently, one of the widest validated, multi-labeled, publicly available image datasets has been released [95]. The dataset comprises 33000 synthetic grayscale images of a simplified Tango model with a resolution of 1024 × 1024 pixels rendered using POV-Ray as the ray-tracing engine. The images are all labeled with relative pose [96], RGB segmentation masks and bounding boxes [97], and reprojected visible line segments from the 3D wireframe model of the target [98]. The images have been successfully validated against SPEED images both qualitatively and quantitatively [95].

During the last years, other image datasets have been published using several rendering software and providing various annotations, but without providing any validation between synthetic images and actual spaceborne images. The dataset in [99] is made of 3771 RGB images (both rendered and real images) with a resolution of 1280 × 720 pixels, representing various spacecraft models labeled with segmentation masks and bounding boxes. Hu et al. [100] rendered 50000 synthetic RGB images of SwissCube CubeSat by using Mitsuba 2, a physically-based ray-tracer, with a resolution of 1024 × 1024 pixels, and labeled with relative pose and segmentation masks annotations. Price et al. [101] rendered 20300 synthetic grayscale images of the rover Minerva-II2 seen from Hayabusa2 by using SolidWorks Photoview 360, with a resolution of 1024 × 1024 pixels, and providing labels for both relative pose and interesting keypoints. Remarkably, the widest dataset available is SPARK (SPAcecraft Recognition leveraging Knowledge of Space Environment) [102]. Despite the drawback of not being validated, it is composed of 150000 RGB images rendered by using Unity3D. It includes 11 different spacecraft models, and all the images are labeled with the relative pose, bounding boxes, segmentation masks, and image depth. Other authors also created image datasets for their works [17,103,104] without making them publicly available. Notably, despite being not public, the dataset in [105] composed of both synthetic and mock-up images of Envisat acquired with HIL in the ARGOS facility at Politecnico di Milano has been validated both qualitatively and quantitatively.

The research presented here employs both SPEED and SPEED+ labeled images within the training phase for the needed CNNs. Moreover, the images from the datasets in [95] have been also included, due to the already available reprojected line segments annotations, increasing the number of the available images for the work here presented to a total of 105 000 labeled images.

3. Pose initialization algorithm

This section describes the architecture developed for M-LSD-based pose initialization. In particular, in Section 3.1 is provided a full overview of the general architecture, focusing on the proposed approach. Section 3.2 thoroughly describes the labeling procedure adopted for SPEED and SPEED+ images, providing information on

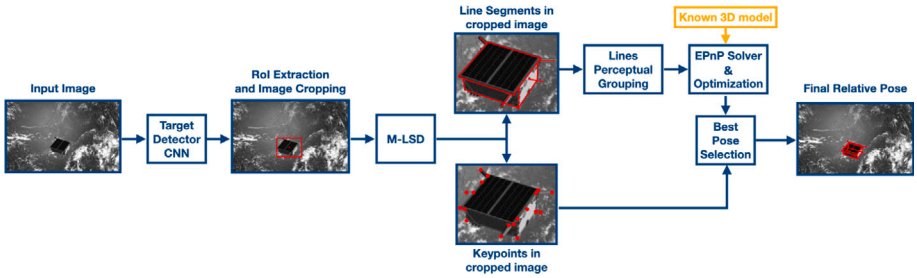

Fig. 1. M-LSD-based pose initialization pipeline high-level overview.

the overall dataset adopted for the work presented here. Sections 3.3 and 3.4 deal with the description of the target detection and line segments detector CNNs respectively, focusing on the modifications introduced with respect to the baselines, the training processes and the performances scored by both the CNNs needed. After detailing the image processing and feature extraction steps, Section 3.5 discusses the matching of 2D to 3D features and the final relative pose estimation step.

3.1. Architecture overview

The general high-level architecture proposed is schematized in Fig. 1. The proposed pipeline involves two tasks fully entrusted to CNNs, i.e. target detection and line segments extraction. Target detection is needed to locate the target in the image, a non-trivial task if the background is present in the image processed. The target detection CNN outputs a bounding box that, if correctly estimated, completely encloses the portion of the image in which the target is present (i.e., the ROI). The full-scale image is cropped to the ROI by using the bounding box estimate provided by the target detector CNN to reduce the size of the image to be processed, reducing the effect of the background and enhancing the accuracy of the feature extraction and of the overall pose estimation pipeline, as already demonstrated in [18,58,59]. The line segment extraction CNN takes as input the cropped image and outputs both the line segments detected and the junctions, i.e., the keypoints. Notice that the junctions come from the junction map given as additional output from the line segment extraction CNN adopted here. This CNN extracts the line segments and the junction maps in separate heads [24], hence, the junctions (i.e., keypoints) extracted do not necessarily coincide with the line segments endpoint. Consequently, a refined ROI can be defined from the keypoints and employed to filter out possible line outliers (e.g., lines not belonging to the target and spurious long lines exiting the refined ROI). Leveraging the refined ROI defined above, the hyperparameters of the line merging and perceptual grouping processes can be scaled based on the ROI diagonal length, adapting them to the different relative distances between camera and target, as in [23,41]. The line segments merging is needed to collect and join possible duplicates and close fragments into a single line segment. The perceptual grouping allows to decrease the search space for the 2D to 3D correspondence problem, lowering the amount of PnP problems to solve and the computational time as a consequence. The perceptual grouping process is applied also to the 3D wireframe model available. Such process is applied only once, on ground before the mission start, and the output 3D perceptual groups are stored and used in the proposed pipeline to build the match matrix i.e., the matrix representing the correspondences between 2D and 3D groups. The match matrix is build here adopting the approach given in [63]. Each entry of the match matrix associates 2D keypoints from 2D perceptual groups to the respective 3D counterpart. All the correspondences in the match matrix are processed with the EPnP algorithm to retrieve the relative pose for each pair. The relative pose estimates are sorted by ranking the computed reprojection error. Only the top-10 poses with the lowest reprojection error are further optimized through a non-linear Levenberg–Marquardt minimization scheme as in [58]. The optimized pose estimates are employed to reproject only the visible 3D wireframe keypoints onto the image plane. The visibility of each 3D keypoint from the estimated camera position and orientation is evaluated using a simplified mesh of the target spacecraft and exploiting the Möller–Trumbore ray-triangle intersection algorithm [106]. The same approach was adopted in [95] to compute visible reprojected lines in images to provide annotations for line segments in view per each image of the synthesized dataset, revealing to be fast and accurate enough. The visible reprojected keypoints are paired via a nearest-neighbor search with the closest feature point from the set of 2D keypoints obtained by combining the endpoints of the merged line segments with the detected keypoints from M-LSD. The final best pose estimated by the proposed pipeline is given by the optimized relative pose that scores the minimum mean reprojection error. The overall architecture is inspired by the SVD algorithm [23]. Nonetheless, the proposed approach introduces substantial updates and modifications to the original SVD that increase the accuracy and the availability of an accurate pose estimate, lowering the overall complexity and computational time also with respect to [44].

3.2. Datasets and labels

From SPEC2019 the SPEED dataset [91] has been used as the benchmark for pose initialization algorithms. For comparison purposes, it is considered the baseline dataset also in this work. The SPEED dataset size has been incremented with images from both SPEED+ [94] and the multilabelled simplified Tango image dataset [95], here named MINIMA (Multilabelled sImplified taNgo IMage dAtaset), to improve the performances of the M-LSD in extracting lines from spaceborne images. For the pipeline presented here, each image needs three different annotations, i.e., the relative pose, the target bounding box, and the reprojected visible line segments from the 3D wireframe model. The images in MINIMA are already labeled with the aforementioned annotations [96–98], while for SPEED and SPEED+ only the relative pose is available and only for a subset of the images (12000 synthetic images in SPEED and 60000 synthetic images in SPEED+) hence the additional required labels have been computed. For sake of completeness, Table 1 shows the intrinsic parameters of the cameras adopted to generate SPEED/SPEED+ and MINIMA datasets.

To retrieve the needed labels the 3D wireframe model of Tango is estimated since it is not publicly available. Reconstructing the 3D wireframe model translates into retrieving the keypoints of the model.

Table 1
Camera parameters of SPEED/SPEED+ and MINIMA datasets.

Parameter	SPEED/SPEED+	MINIMA
Resolution ($N_u \times N_v$)	1920 × 1200 px	1024 × 1024 px
Pixel size ($d_u \equiv d_v$)	5.86 μm/pixel	4.8 μm/pixel
Focal length f	17.6 mm	6.0 mm
Horizontal FOV	35.45 deg	44.54 deg
Vertical FOV	22.59 deg	44.54 deg

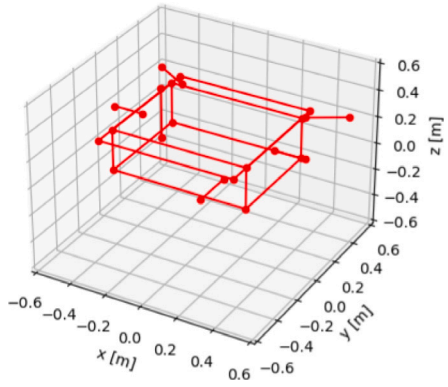

Fig. 2. Reconstructed 3D wireframe model and keypoints for SPEED/SPEED+ images.

This issue is handled by hand-picking the keypoints in 25 images from SPEED and then performing an iterative minimization problem as described in [57], where the objective function to minimize is the sum of the reprojection errors between the 3D keypoints and the associated 2D image features by tuning the 3D coordinates of the keypoints, knowing the relative pose associated with each image. Fig. 2 shows the estimated 3D keypoints and the reconstructed 3D wireframe model. Once the 3D keypoints and a simplified mesh representative of the Tango model used in SPEED and SPEED+ are available, the bounding box and the visible reprojected line segments can be annotated for each image as detailed in [95].

The bounding boxes retrieved by reprojecting the keypoints in the image frame have been enlarged before being annotated as in [58] to avoid any unintentional cut-off of portions of Tango from the ROI. Specifically, the sides of the ROI are widened by the maximum value between the 10% of the half-height and the 10% of the half-width of the originally detected ROI. Notice that by providing enlarged annotated ROI values as ground truth for the target detector CNN during the training phase, the CNN is implicitly forced to learn to predict a relaxed bounding box, limiting the possibilities of cutting out portions of Tango during inference. Fig. 3 shows as a dotted line the minimum ROI computed while, with a solid line, the relaxed ROI used for training the target detector CNN. The bounding box annotations also include the center of the ROI, represented in Fig. 3 as a dot.

Even the annotations of the visible line segments needed to train the M-LSD can be obtained by reprojecting 3D keypoints into the image plane. In detail, the reconstructed 3D wireframe model is divided into lines with associated endpoints (i.e., start and end keypoints). To properly handle possible partial occlusion of the line segments, each defined line is split into sub-segments by adding 100 equally spaced intermediate keypoints along the line segment, starting from the left-most original 3D keypoints. Then, the visibility of each keypoint is evaluated through the Möller–Trumbore ray-triangle intersection algorithm as in [95], given the already labeled relative pose and the reconstructed simplified mesh. The binary visibility score of each keypoint that is inside the FOV of the camera is saved in a vector before reprojecting only the visible keypoints in the image plane. The binary visibility vector is then employed to retrieve which portions of the original line segment are in view and that have to be annotated. If there are more than one visible portions interspersed by non-visible ones, they are annotated as independent line segments. Otherwise, the line segment is annotated as a single line if it is totally in view. The procedure parses each line segment of the 3D wireframe model for each image available, annotating the line segments in view in each image in the standard format of the Wireframe Dataset introduced in [85]. So far, the largest publicly available dataset with wireframe annotation is the one in [98], up to the authors' knowledge. Fig. 4 shows two example images from the SPEED dataset where the annotated reprojected line segments in view computed with the scheme presented above are in red. The algorithm can retrieve both complete and fragmented line segments with a high level of accuracy also for those cases in which only a minor sub-portion of the whole line is visible (e.g., in the back corner of the solar array of Tango highlighted in Fig. 5) or multiple visible lines are close each other (e.g., the lines belonging to the solar panel and the base of the main body of Tango in the right picture), without introducing any unintended gap in correspondence of line junctions.

Table 2 offers a general overview of datasets and labels used in this work.

Table 2
Overview of datasets adopted.

Name	Images	Relative distances	Annotations	
MINIMA	33000	5 m–30 m	*Relative Pose*	Available
			ROI	Available
			Line Segments	Available
SPEED	12000	3 m–40.5 m	*Relative Pose*	Available
			ROI	Computed
			Line Segments	Computed
SPEED+	59960	2.25 m–10 m	*Relative Pose*	Available
			ROI	Computed
			Line Segments	Computed

The images in the MINIMA dataset are noiseless thus the same noise levels of SPEED/SPEED+ images are added before processing the images. Specifically, a Gaussian Blurring kernel with standard deviation $\sigma = 1$ is applied to the noiseless images, then an Additive White Gaussian Noise with variance $\sigma^2 = 0.0022$ and zero mean is superimposed to the blurred image. The intensity value of each pixel in the noised image is clipped to the range [0, 255]. This procedure to add noise to MINIMA images has been already validated in [95]. Although the SPEED/SPEED+ images are already abundant, those from MINIMA have been included because the target model is a simplified version of the Tango adopted for SPEED/SPEED+, with fewer details and slightly different textures for the solar panel. Moreover, both the Earth and the camera models adopted in MINIMA differ from those of SPEED/SPEED+. It has been already proven that introducing some augmentations also in the target texture during training is beneficial to improve performances and slightly fill the domain gap that can be present between the training set and the actual test environment, improving the robustness of the entire pose estimation algorithm [53]. Hence, we introduced the MINIMA to increase the generalization capabilities and the accuracy of the M-LSD and the entire pipeline proposed.

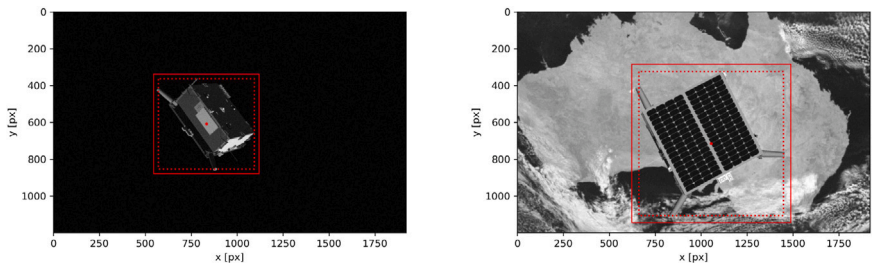

Fig. 3. Examples of computed ROI (dotted lines), enlarged ROI labeled (full lines), and ROI center labeled (dot) for SPEED images.

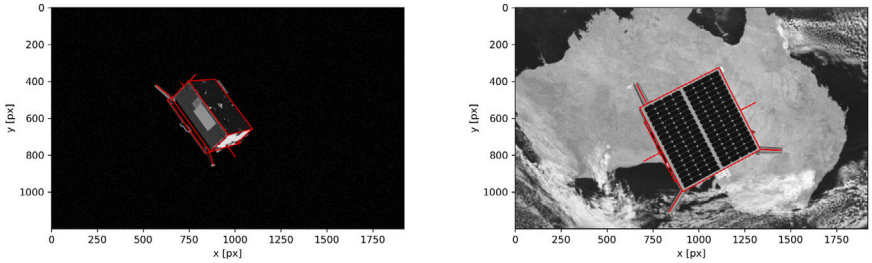

Fig. 4. Examples of computed line segments in view (red lines) for SPEED images.

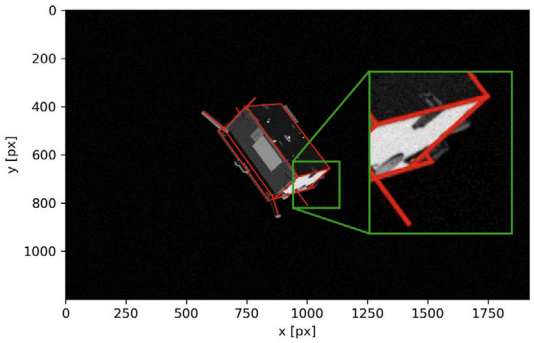

Fig. 5. Examples of correctly interrupted line segment due to visibility.

For training and validations purposes, the SPEED+ and the MINIMA datasets have been split into training and validation sets with a 9:1 ratio, while the 12000 images in the SPEED dataset have been shuffled and randomly split into 7680 training images (64% of SPEED), 1920 validation images (16% of SPEED), and 2400 test images (20% of SPEED). As detailed in the next Sections, the Target Detection CNN has been trained by using only images from SPEED dataset with the aforementioned split, while the M-LSD has been trained by using the training splits from SPEED, SPEED+ and MINIMA and validated with the corresponding validation splits. The images in the test split of SPEED defined above are adopted as labeled test set throughout this paper, and they are never used during the training phases. The images in the labeled test set belong only to the SPEED dataset to provide a labeled testing scenario coherent with the SPEC2019 test set, which, on the contrary, is not labeled, to get more insights on the performances of the pose initialization algorithm developed.

3.3. Target detection CNN

As previously mentioned, the baseline target detector CNN adopted is the YOLOv5 due to its good trade-off between inference time and FLOPs, and since it has already been successfully adopted in its "small" version (YOLOv5s) for spaceborne images in [58]. Specifically, in the selection phase of the baseline target detector CNN, both YOLOv5n and YOLOv5s have been considered. The former is the smallest version of the YOLOv5 series, being composed of 213 layers with a total of 1.76 M parameters and requiring 4.1 GFLOPs during inference. The latter is slightly bigger than the former, and is made of 213 layers with a total of 7.0 M parameters and requires 15.8 GFLOPs during inference. The values reported above hold for an input image having size 512×512 pixels. The input size has been selected from a trade-off between accuracy and inference time. The application scenario is a single-class/single-object, thus a simplified version of the more generic multiple-class/multiple-object framework for which the YOLO CNNs are meant. Hence, the original architecture of YOLOv5n and YOLOv5s has been further simplified by skipping the final non-maximal suppression step and ensuring that the output is a single bounding box with a confidence threshold higher than 0.6 as in [58]. The two-stage methods have been discarded for the implementation proposed here due to the proven high inference time and computational cost [68].

3.3.1. YOLO training and comparison criteria

The training phase of the two YOLO versions is performed on an NVIDIA RTX™ A6000 GPU. Training images are augmented with a probability of 10% during the training phase using the default augmentations of YOLO training (e.g., random brightness variations, blurring, Contrast Limited Adaptive Histogram Equalization (CLAHE), geometrical transformations, etc.) with the addition of random Gaussian and ISO noises. Both the considered variants are trained using the stochastic gradient descent method with a learning rate of 0.01 linearly decaying to 0.0001, a momentum of 0.937, weight decay of 5×10^{-4}, and a minibatch size of 64 images. Binary cross entropy is adopted as loss function. YOLOv5n is trained for 500 epochs, while YOLOv5s training lasts for 800 epochs. The training parameters were maintained as default since the monitored metrics discussed hereafter do not improve by changing the default values. The mini-batch size was selected by maximizing the number of images in each mini-batch without surpassing the available computational resources for the training phase. The number of epochs was defined by noticing during experiments that training longer gives no meaningful improvements, resulting in most of the cases in the early stopping after a few more steps due to no improvements on the monitored metrics discussed hereafter.

The criteria adopted to evaluate the performances of the selected YOLOs on the test set are the Intersection-over-Union (IoU) index and the average precision (AP). The IoU gives the percentage of the overlap between the predicted and ground truth bounding box. The higher the IoU, the more accurate the predicted bounding box. The AP is the area under the precision–recall curve. Namely, the precision is the ratio of the number of correct bounding boxes predicted (i.e., true positives) over the total amount of predicted bounding boxes (i.e., true positives plus false positives), while the recall is the ratio of correct bounding boxes predicted (i.e., true positives) over the total number of ground truth bounding boxes (i.e., true positives plus false negatives). The IoU is thresholded to generate several precision–recall curves for different levels of IoU. The mean average precision is computed by averaging the AP evaluated for each level of IoU. Namely, the AP^{95}_{50} is computed by averaging the AP values evaluated for precision–recall curves obtained by thresholding the IoU to 50% and 95%. The evaluation of the AP is usually performed by computing precision–recall curves for IoU thresholds from 50% to 95% with steps of 5%, leading to ten different curves. Please notice that the higher the IoU threshold, the lower the area under the precision–recall curve hence, in terms of accuracy, the higher the AP^{95}_{50}, the more accurate the ROI detection.

3.3.2. YOLO results and baseline selection

Fig. 6 shows the Precision–Recall curves for different IoU thresholds obtained on the test set for YOLOv5n (left) and YOLOv5s (right) after the training, demonstrating that the YOLOv5s achieves better precision–recall performances than YOLOv5n. Computing the mean IoU in the test set and the average precision in the IoU range from

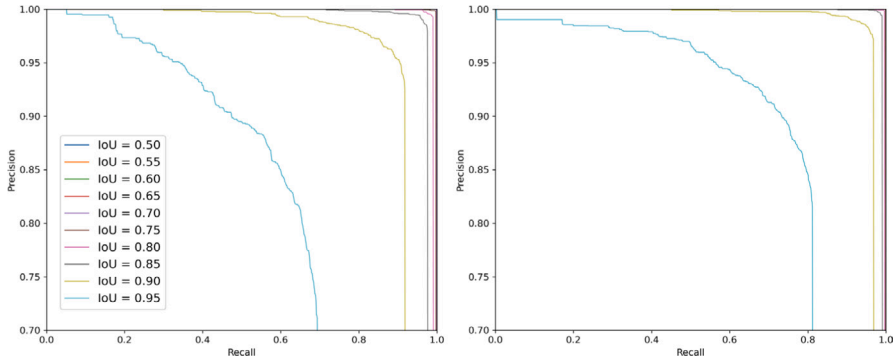

Fig. 6. Precision–Recall curves on test set for YOLOv5n (left) and YOLOv5s (right).

Table 3
Performances of target detection CNNs on SPEED images.

Method	Mean IoU	Mean AP_{50}^{95}
SLAB Baseline [18]	91.9%	N.A.
UniAdelaide [59]	95.34%	N.A.
SLN (YOLOv5s) [58]	95.38%	98.51%
YOLOv5n (our)	95.42%	95.7%
YOLOv5s (our)	96.46%	97.6%

0.50 to 0.95 for both architectures confirms the insights achieved from the plots comparison. The mean IoU values are reported in Table 3 and compared with the top-scoring architecture of SPEC2019 [59], the SLAB baseline [18], and the results achieved by the Spacecraft Localization Network (SLN) in [58] (that scored the most accurate performances in terms of IoU on SPEED using YOLOv5s).

The proposed YOLOv5s outperforms all the previous architectures with an increment in the mean IoU of ∼1.1%, despite the AP_{50}^{95} is slightly lower than the SLN, setting a new highest mean IoU value on SPEED images. Notably, also the proposed YOLOv5n is capable of outperforming all the other architectures listed in Table 3 in terms of IoU, even if the gap in AP_{50}^{95} with respect to the SLN is more severe. Please notice that the AP_{50}^{95} scores for the SLAB Baseline and for the architecture proposed by UniAdelaide are not available in the literature. It is acknowledged that the gap in both IoU and AP_{50}^{95} scores between the SLN and the YOLOv5 model adopted here are due to differences in the training parameters and likely in the YOLOv5 parameters, due to their continuous updated from the developers.

Due to the higher accuracy offered by the YOLOv5s, such architecture has been selected as the baseline to investigate the capabilities of the proposed pose initialization architecture, even with a slightly longer inference time and computational burden with respect to YOLOv5n. The histograms representing the IoU scores of both YOLOv5s and YOLOv5n are reported in Fig. 7 to have a more comprehensive overview of the performances on the SPEED images included in the test set, while examples of YOLOv5s ROI detections against ground truth bounding boxes on test images are shown in Fig. 8 for different levels of IoU.

3.4. Line segment detection CNN

The line segment detector is based on the U-shaped MobileNet architecture from M-LSD [24], which offers the advantage of high detection accuracy with low computational cost, being able of running in real-time on mobile devices. The dataset employed differs from the Wireframe dataset [85] (which is the state-of-the-art benchmark for line segment detection tasks) as spaceborne images result in fewer complex patterns and overlapping situations among various objects. This simplification of the application scenario allows reducing the input resolution from 512 × 512 to 256 × 256 without significant performance degradation, enabling faster computation. The model adopted here comprises 1.5M parameters and requires 6.44 GFLOP for processing each input image.

The M-LSD model is trained using the public datasets and their splits discussed in Section 3.2. Before training, the datasets underwent several preprocessing steps, including resizing to the input size (random cropping was not adopted), affine transformation with shearing ranging from −30° to 30°, rotation ranging from −90° to 90°, and the addition of Gaussian noise. The dataset was further augmented with horizontal or vertical flips, color space distortion including adjustments to brightness, contrast, saturation, and hue, and a specialized augmentation method specifically designed for the training of M-LSD named Segments of Line segment (SoL) [24]. SoL augmentation was designed to enrich the training dataset with a wider range of line segment lengths by dividing the original line segments into multiple overlapping sub-segments, ensuring both continuity between the segments and a broader variety of segment lengths for training, that are used during the training phase.

3.4.1. M-LSD training and evaluation criteria

The training procedure adopted closely follows the one developed for the reference M-LSD model, with minor parameter modifications to account for the differences between the employed datasets and the Wireframe dataset originally adopted in [24]. Namely, the datasets employed predominantly contain long line segments, which require a large receptive field for accurate prediction, concentrated in a small portion of the total image, i.e., the ROI. In contrast, the Wireframe dataset consists of a mixture of short and long line segments, necessitating the detection of both. The input image size has been modified here from 512 to 256 pixels to address this disparity, forecasting to process only the extracted ROI and optimizing the inference time of the model. The number of warm-up epochs has been increased from 5 to 10 for training stability and the matching threshold for the matching loss proposed in [24] was adjusted from 5.0 to 2.5 to follow the reduced input image size. All other parameters remained the same as in the original M-LSD model. The training process was carried out for a total of 300 epochs and reported the performance of the final model adopted. The training employed a mini-batch size of 64 images across 8 NVIDIA V100 GPUs, with each GPU processing a local batch of 8 images. The Adam optimizer was used with default parameters, except for the learning rate, which started at 0.001 and decreased to 0.00001 using cosine decay. These values have been defined starting from the training parameters in [24] and scaling them employing log-scale tuning. The reported values ensure both stability in the training process and optimal performances in the application scenario. The loss function remained identical to the one used in the original M-LSD paper [24].

The primary evaluation metric adopted here is the structural average precision (*sAP*) proposed in [87]. sAP^{ϵ} measures the performance of line segment detection algorithms on vectorized wireframes, as opposed to a heatmap, providing a comprehensive measure of algorithm performance over various thresholds ϵ. Inspired by the mean average

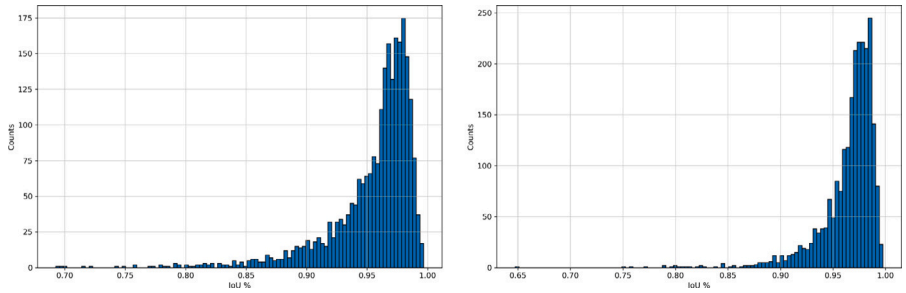

Fig. 7. IoU score on test images for YOLOv5n (left) and YOLOv5s (right).

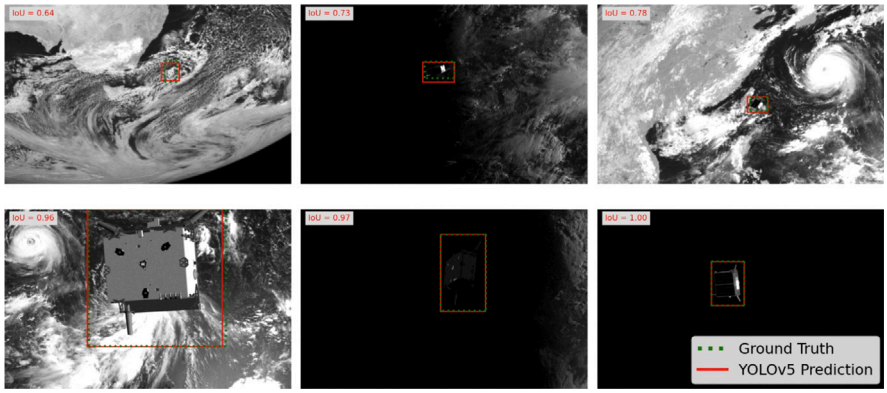

Fig. 8. Ground truth and predicted ROI using YOLOv5s on test images with IoU score.

precision commonly used in object detection and discussed in Section 3.3, *sAP* is computed as the area under the precision–recall curve, where a line segment is marked as a true positive if the sum of the squared error between predicted end-points and their ground truth is less than ϵ. The use of *sAP* in this work reflects the need for a robust and comprehensive measure of line segment detection performance, with the ϵ parameter taken from [87] to ensure a standardized evaluation.

3.4.2. M-LSD training results and baseline selection

As shown in Table 4, the experimental results for two types of M-LSD models (namely, baseline and *tiny* version) are compared by considering both 512 × 512 pixels and 256 × 256 pixels as resolutions for input images. The outcomes in Table 4 point out that there is not a significant difference in terms of model performance when reducing the resolution from 512 to 256 in the same model architecture for the dataset here considered. However, there is a drop of more than four times in terms of computational cost (GFLOP) and a significant increase in terms of frame per second (FPS) measured on an NVIDIA V100 GPU. Similar behavior, in terms of GFLOP and FPS, can be obtained by changing the architecture from the baseline to the *tiny* version and maintaining the same resolution but, in that case, it also leads to a substantial degradation of performances. This aspect holds great importance as it can significantly affect the overall performance of the relative pose estimation task. Consequently, the 256 × 256 M-LSD baseline architecture is adopted as the line detection model in this study, considering the trade-off between performances and computational complexity of the analyzed architectures.

Table 4
Impact of resolution and model structure on M-LSD performance and computational cost.

Setup	sAP^5	sAP^{10}	Params (M)	GFLOP	FPS
512 × 512 M-LSD	63.10	68.92	1.5	25.79	115.4
256 × 256 M-LSD	62.73	68.31	1.5	6.44	151.1
512 × 512 M-LSD tiny	59.31	62.14	0.6	4.22	164.1
256 × 256 M-LSD tiny	58.89	61.88	0.6	1.05	208.3

While training M-LSD, the learning rate was kept fixed at 0.001 during the 10 warm-up epochs, lowered from 0.01 initially proposed in [24] since the originally proposed 5 warm-up epochs at 0.01 did not lead to convergence.

Table 5 presents the results of the ablation study of M-LSD for line segment detection for pose initialization using the aforementioned datasets, while Fig. 9 shows the qualitative results. Baseline (M1 from Table 5) refers to the experimental results using only MINIMA training dataset and all training receipts except for the warm-up and learning rate mentioned before. The performances of the model improved from M1 to M3 since more images from different datasets are included in the training. According to Fig. 9, M1 shows inaccurate results, but with the addition of the SPEED and SPEED+ images, it became possible to obtain results suitable for the relative pose initialization scheme proposed.

Due to the significant emphasis on the performance boost of M-LSD due to the matching loss [24], the model has been tuned by adjusting a matching threshold, which is a hyperparameter used in the matching loss. The matching loss is proposed to make the predicted line segments and the ground truth line segments closer by minimizing the distances between the endpoints of both. If the Euclidean distance between the endpoints is lower than the matching threshold, it is selected for loss computation, and the matching loss is calculated based on the L1 distance between the two endpoints. As the input images are downscaled from 512 × 512 to 256 × 256, using the same threshold value of 5.0 as a default value proposed in M-LSD [24] would result in improperly predicted line segments being excessively included in the matching loss computation, preventing the training of an optimal model. Therefore, the threshold value was reduced from 5.0 to 2.5 in proportion to the decreased resolution. The tuning of the matching threshold brings a huge performance improvement. as pointed out in Table 5 where M4 achieved an sAP^5 of 62.73 on MINIMA, SPEED, and SPEED+ validation datasets combined. As shown in Fig. 9, M4 predicts the junction points and line segments more accurately if compared to other models (M1, M2, and M3), as a consequence M4 was adopted as baseline. Furthermore, Fig. 9 shows that the keypoints returned do not

Table 5
Ablation study for M-LSD. SPEEDs represents the combination of the SPEED and SPEED+ validation sets.

M	Scheme	sAP^5 MINIMA	sAP^5 SPEEDs	sAP^5 all
1	Baseline	56.21	36.28	48.45
2	+ SPEED [91]	55.81	52.89	53.57
3	+ SPEED+ [94]	55.62	54.32	54.81
4	+ make the matching threshold [24] 2.5 from 5.0	60.92	63.77	62.73

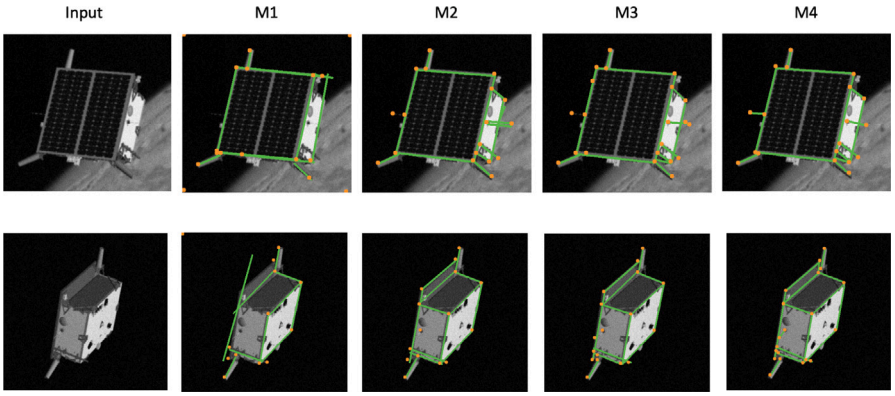

Fig. 9. Qualitative comparison of M-LSD detected line segments and junctions for different models in ablation study.

necessarily coincide with the line segments endpoints since they are independent outputs of the M-LSD, making them suitable to identify a refined ROI to filter possible line segments outliers as outlined in Section 3.1.

3.5. Match matrix and final relative pose estimation

The last part of the proposed pipeline concerns the processing of the detected line segments to retrieve the final pose estimate. The algorithms needed for this task are retrieved from the original SVD implementation in [23]. As already noticed in [23,63,107], the presence of repeated edges and even fragments belonging to the same line segment entirely visible erroneously detected as independent line segments can jeopardize the entire pose estimation algorithm, leading to high estimation errors. The M-LSD was proven during preliminary tests to be robust against these drawbacks that characterize the behavior of the Hough Transform but, to increase the accuracy of the estimated pose and reduce the computation time, a line merging step is also included in the proposed architecture. The (ρ, θ) parametrization, also named polar representation, is adopted to check the similarity of line segments and perform the perceptual grouping. Two line segments (ρ_1, θ_1) and (ρ_2, θ_2) are similar if $|\theta_1 - \theta_2| < \theta_{thresh}$ and $|\rho_1 - \rho_2| < \rho_{thresh}$. Moreover, the Euclidean distance between the farthest endpoints of the two line segments must be lower than a threshold d_{thresh}. Unlike Sharma et al. [23] that set constant threshold values for θ_{thresh} and ρ_{thresh}, the work presented here scales ρ_{thresh} with the diagonal dimension of the detected ROI d_{ROI} as in Eq. (1), improving the correct identification of spacecraft true edges [62], while the θ_{thresh} is maintained constant. d_{thresh} is computed for each image as half the mean length of the detected line segments.

$$\rho_{thresh} = \nu d_{ROI} \tag{1}$$

The interested reader is referred to [23,62,107] for more details on the line segments and parallel streams merging processes. Please notice that the parallel streams merging, discussed in [23] and needed in SVD to properly combine the line segments detected by the two parallel streams, is not included in the architecture here presented since only one stream is present, reducing the overall complexity and computational time required to process a single image.

The perceptual grouping, or feature synthesis, is one of the main innovations introduced with SVD. As already mentioned, it consists in organizing the detected line segments into more complex geometrical groups named "perceptual groups" to reduce the search space for the final relative pose estimation. In a general framework, by assuming that the EPnP is adopted as PnP solver, a minimum of 6 correspondences between the n 2D features and the m 3D points of the available 3D CAD model are needed to uniquely determine the relative pose. Hence, the total amount of correspondences to be tested can be computed as [23]:

$$\binom{m}{6}\binom{n}{6}6! \tag{2}$$

The perceptual grouping reduces the number of correspondences to be tested by introducing the geometrical constraints that characterize each group of lines considered, i.e. the feature detected are not considered as independent, but linked to the other features as in the case of two keypoints belonging to the same line segment [23,41]. The perceptual groups adopted also in the work presented here are the parallel pairs, proximity pairs, parallel triads, proximity triads (or open tetrads), and closed tetrads, while the antennas are treated as a separate group and used to break the symmetry and disambiguate the relative pose. Notice that for Tango the antennas are visible from all the observing conditions, hence they represent a good candidate group to disambiguate images but, due to this role in the implemented algorithm, their correct identification is of the utmost importance. Please notice that the algorithm here presented is tailored to the relative pose estimation using Tango as target, but it can be generalized to other spacecraft provided the presence of identifiable features to break possible symmetries and disambiguate the estimated poses. The simple geometrical constraints defined for SVD, modified by introducing a scaling with respect to d_{ROI} discussed in [62,107] to properly define the aforementioned perceptual groups, have been adopted also in this work. The details of these constraints are not reported here for the sake of brevity but the readers are encouraged to retrieve the needed information in [23,62,107]. The only modification with respect to [62,107] consists in the introduction of more checks to be passed by a line segment before being categorized as an antenna. The original check was only on the length of the line segment, which is classified as "antenna" only if it length is $l \leq \tau d_{ROI}$, where τ is a scaling factor. During some preliminary analysis it has been noticed that for some poses the length of actual antennas in SPEED

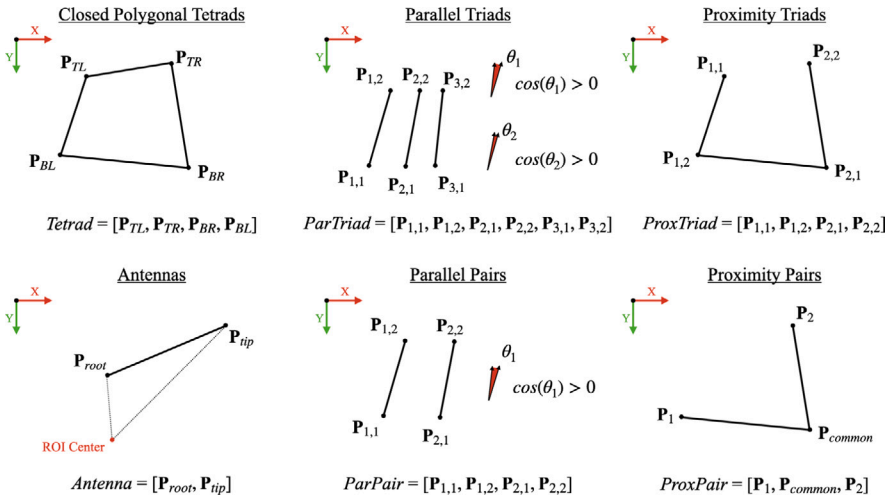

Fig. 10. Schematic of the constrained ordering of the perceptual groups from [63].

images is higher than the length of the side edges that connects the base of Tango with the upper solar panel hence also these edges where classified as antennas. This erroneous classification is first prevented by introducing a check on the radial disposition of the segments that are initially classified as antennas. For the case of Tango all the antennas are directed approximately in radial direction with respect to the center of the ROI. As noticed in [63], retrieving the tip and the root keypoints for a 2D antenna is an easy task provided the accuracy of the ROI detection. Once that this information is available, the angle β between the 2D vectors that represent the position of the root and the tip with respect to the ROI center is evaluated. If $\beta > \beta_{thresh}$ the segment is not an antenna. If $\beta \leq \beta_{thresh}$, then it is verified that the distance between the tip of the segment classified as an antenna and the endpoints of all the other segments is not lower than a threshold value set equal to 20% of the length of the antenna. This second check comes from the fact that all the appendages are directed outside from the envelope of the main body for Tango, hence the tip of all these appendages must be far from all the other line segments detected. If these two additional conditions are not verified, the line segment is not classified as an antenna. Notice that these two additional checks can be implemented with simple operations that must be repeated for few lines since they are applied in cascade, hence their impact on the computational cost is low. The perceptual grouping described here is performed for each image processed to detect 2D perceptual groups. The 3D perceptual groups from the wireframe model are pre-computed only once and stored in memory for subsequent matching with the 2D perceptual groups detected during the runtime. The 3D perceptual groups detected for the Tango wireframe model are 18 parallel pairs, 16 proximity pairs, 12 parallel triads, 12 proximity triads, 2 closed tetrads, and 5 antennas.

Keypoints in 2D groups are assigned to features in 3D groups during the definition of the match matrix. Notice that to achieve a minimum of 6 correspondences, the groups must be combined with at least one antennas. The only exception is the group of parallel triads, that already gives six keypoints per parallel triad detected. The formulation adopted in this work is taken from [63], where it is explained how introducing a constrained ordering in the format used to store all the perceptual groups makes possible to obtain a mathematical formulation to forecast the dimensions of the match matrix, hence the number of correspondences to be tested with the EPnP. A scheme visually representing the constrained ordering from [63] is given in Fig. 10. By defining as $\phi_{a,2D}$ the number of 2D antennas detected in the image, and as $\phi_{a,3D}$ the number of 3D antennas in the 3D CAD model, it is possible to evaluate the number of combinations $N_{comb,ant}$ needed to build the match matrix as a function of k, the number of antennas needed to obtain a complete set of 6 features per each perceptual group, as [63]:

$$N_{comb,ant} = \frac{1}{k!}\frac{\phi_{a,2D}!}{(\phi_{a,2D}-k)!}\frac{\phi_{a,3D}!}{(\phi_{a,3D}-k)!} = \frac{1}{k!}D_{a_{2D},k}D_{a_{3D},k} \tag{3}$$

Where $D_{n,k}$ represents the k-permutations of n elements without repetition. From Eq. (3), by adopting the constrained ordering of the feature groups, it is possible to compute the number of rows of the match matrix as reported in Table 6, where the groups are listed from top to bottom in descending order of complexity.

Notice that only the tip of the detected antennas is adopted to build the match matrix since it is the only portion of the antennas that is always in view, while the root, on the contrary, can be obstructed by other components of Tango. Unlike the work in [63], the number of antennas used in the match matrix is selected here as a function of the detected antennas. This tuning of the number of antennas combined with other feature groups leverages the higher accuracy in classifying line segments as antennas, obtained from the combination of the M-LSD detections and the geometrical check outlined above, to provide more keypoints correspondences to the EPnP. Indeed, using the EPnP solver provided by the OpenCV Python package, a relative pose can be retrieved even for a set of correspondences in input equal to or higher than 4, but the success rate and the accuracy of EPnP increases with the number of input correspondences, as demonstrated in [23]. Building on that, if $\phi_{a,2D} > 4$, then $k = 3$ for the entries in Table 6. This leads to a more accurate pose estimation in the case of the camera z-axis almost aligned with the z-axis in the target reference frame (i.e., Tango seen from the bottom or above). If $2 < \phi_{a,2D} \leq 4$, then $k = 2$ but only non-collinear antennas are combined. If $\phi_{a,2D} = 2$ then $k = 2$ without considering the collinearity of the antennas, while $k = 1$ if $\phi_{a,2D} = 1$. Notice that this last case is underdefined for all the groups, thereby the accuracy of the pose estimate given by EPnP may be poor. The combinations of parallel triads and proximity pairs are handled separately from the other groups. The former are always computed if detected, disregarding the number of 2D antennas available, while the latter are evaluated with $k = 3$ if $\phi_{a,2D} \geq 3$ and $k = 2$ if $\phi_{a,2D} = 2$, to limit the presence of underdefined solutions of the EPnP. A more detailed description of the constrained ordering of the perceptual groups and the computation of the match matrix can be found in [63].

After the definition of the match matrix as reported above, the retrieval of the final pose estimates follows the steps outlined in Section 3.1. Hence, all the 2D to 3D correspondences embedded in the rows of the match matrix are fed to the EPnP solver to define a pose for each of them. The top-10 pose estimates with the lowest reprojection error are refined further through a non-linear Levenberg–Marquardt minimization scheme. These refined pose estimates are employed to

Table 6
Rows for each perceptual group in the match matrix.

Feature group	Points per feature group	Number of 2D feature groups	Number of 3D feature groups	Number of rows in match matrix
Antenna	1	ϕ_a	ϕ'_a	–
Closed Tetrad	4	ϕ_b	ϕ'_b	$8\phi_b\phi'_b \cdot \frac{1}{k!} D_{a_{2D},k} D_{a_{3D},k}$
Proximity Triad	4	ϕ_c	ϕ'_c	$2\phi_c\phi'_c \cdot \frac{1}{k!} D_{a_{2D},k} D_{a_{3D},k}$
Parallel Triad	6	ϕ_d	ϕ'_d	$12\phi_d\phi'_d$
Parallel Pair	4	ϕ_e	ϕ'_e	$4\phi_e\phi'_e \cdot \frac{1}{k!} D_{a_{2D},k} D_{a_{3D},k}$
Proximity Pair	3	ϕ_f	ϕ'_f	$2\phi_f\phi'_f \cdot \frac{1}{k!} D_{a_{2D},k} D_{a_{3D},k}$

reproject only the 3D keypoints visible from the estimated relative pose on the image plane. Subsequently, a nearest-neighbor search matches the reprojected keypoints with the closest feature point from a combination of line segments endpoints and junctions keypoints detected by the M-LSD. The final pose estimate is given by the optimized pose with the lowest reprojection error evaluated between matched pairs of detected and reprojected features. Regarding the Levenberg–Marquardt refinement, the built-in implementation offered by the OpenCV Python package is adopted here by setting as termination criteria a maximum number of iterations equal to 2000 and a minimum tolerance between consecutive solutions of 1×10^{-14}. These values have been selected by noticing during the test phase that no further improvements in the output relative pose are achieved by decreasing the tolerance, while the maximum number of iterations is set to avoid high computational times in case the tolerance termination criterion was never triggered.

Processing all the 2D-to −3D correspondences in the match matrix with the EPnP solver is the most time-consuming step in the entire pipeline outlined since it involves processing each line of the match matrix. Notice that the most complex groups in Table 6 are given by combinations of the most basics ones, thus it could be considered to rely only on a subset of these groups to build a reduced match matrix, lowering the overall computational time. In particular, the reduced match matrix adopted here includes only the two most complex groups available detected in each image coupled with the parallel triads (if detected) to have measures independent from the detection of the antennas. The results in Section 4 provide an evaluation of the effects on both the accuracy and computational time with both full and reduced match matrix.

It is acknowledged that, despite the approach outlined above being general from the formulation point of view, the pipeline has been optimized to work with Tango as a target through checks that account for the specific geometry of this target (e.g., controls on antennas and perceptual groups considered) represents a limitation of the applicability of the proposed approach towards other targets. Hence, the geometry of other spacecraft should be analyzed carefully and included in the pipeline before applying it.

4. Results

The main outcomes related to the proposed pose initialization architecture are detailed and commented in the following subsections. Namely, Section 4.1 describes the general relative pose initialization framework and the metrics adopted to characterize the estimation error. Section 4.2 discusses the outcomes of the participation in the SPEC2019 with the baseline YOLOv5s/M-LSD pose initialization architecture and the verification of the representativeness of a labeled test set extracted from the SPEED images with available labels. The representative test set is then employed to have more insights on the proposed architecture in Section 4.3, including the effect several common sources of errors in the estimation accuracy, and considerations about algorithm latency and possible speedup. Section 4.4 shows a comparison of the proposed algorithm against other approaches to the relative pose initialization problem, highlighting both the benefits and weaknesses.

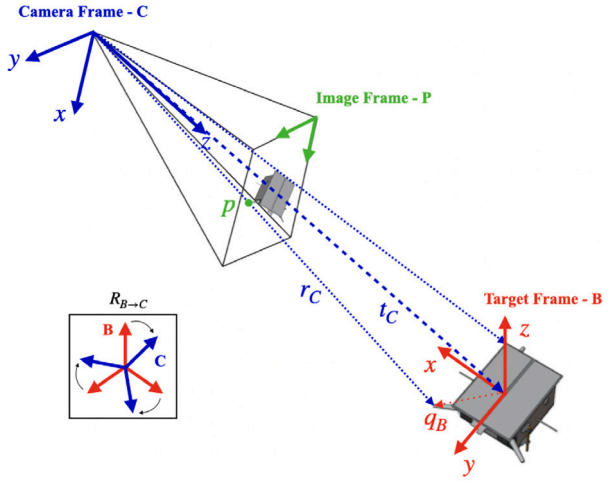

Fig. 11. Reference frames and PnP problem scheme.

4.1. Evaluation metrics

The relative pose estimation problem consists in evaluating the position of the target center of mass in camera reference frame $\mathbf{t}_C$ and the relative attitude expressed as the rotation matrix $\mathbf{R}_{B\to C}$, i.e., the rotation matrix from target to camera reference frame. Taking Fig. 11 as reference, it is possible to map the 3D position of a target feature point expressed in body frame $\mathbf{q}_B$ to its 2D position in the image plane $\mathbf{p}$ by using the projective geometry equations as:

$$\mathbf{r}_C = \left(x_C,\, y_C,\, z_C\right)^T = \mathbf{R}_{B\to C}\mathbf{q}_B + \mathbf{t}_C \tag{4}$$

$$\mathbf{p} = (u,\, v) = \left(\frac{x_C}{z_C} f_x + C_x,\, \frac{y_C}{z_C} f_y + C_y\right) \tag{5}$$

Where f_x and f_y are the focal lengths of the camera expressed in pixels, while (C_x, C_y) are the principal point coordinates of the image in pixels. The relative pose estimation task concerns the estimation of $\mathbf{t}_C$ and $\mathbf{R}_{B\to C}$ given the known location of pre-selected features $\mathbf{q}_B$ and the coordinates in pixel of their reprojection $\mathbf{p}$ in the image plane. Eqs. (4) and (5) are the PnP equations solved by PnP solvers that output the relative pose provided the 2D to 3D matching. Notice that from Eqs. (4) and (5) it is clear that a good relative pose estimate can be performed only if the 2D features are correctly detected and matched with 3D keypoints by IP algorithms.

To evaluate the performances of pose estimation algorithms, SPEC2019 and SPEC2021 adopted a single scalar error metric introduced in [19], defined here as "SLAB error" e_{slab}. The SLAB error is evaluated for each ith image as the sum of the normalized translational error and the rotational error. In detail, the translational error is computed as the norm of the difference between the ground truth relative position $\mathbf{t}_C$ and the estimated one $\hat{\mathbf{t}}_C$ (see Eq. (6)), normalized with respect to the norm of the ground truth relative position, as in Eq. (7).

$$E_t = \|\mathbf{E}_t\| = \|\mathbf{t}_C - \hat{\mathbf{t}}_C\| \tag{6}$$

$$e_t = \frac{E_t}{\|\mathbf{t}_C\|} = \frac{\|\mathbf{t}_C - \hat{\mathbf{t}}_C\|}{\|\mathbf{t}_C\|} \tag{7}$$

The rotation error is evaluated by using the quaternion representation $\mathbf{q}$ of the relative attitude of the camera with respect to the target $\mathbf{R}_{B\to C}$. Specifically, the rotation error is computed by evaluating the quaternion error between the ground truth relative attitude $\mathbf{q}$ and the corresponding estimate $\hat{\mathbf{q}}$ as in Eq. (8).

$$e_q = 2 \cdot \arccos|\mathbf{q} \cdot \hat{\mathbf{q}}| \tag{8}$$

Hence, the SLAB error can be computed per each ith image as:

$$e_{slab}^{(i)} = e_t^{(i)} + e_q^{(i)} = \frac{\|\mathbf{t}_C^{(i)} - \hat{\mathbf{t}}_C^{(i)}\|}{\|\mathbf{t}_C^{(i)}\|} + 2 \cdot \arccos|\mathbf{q}^{(i)} \cdot \hat{\mathbf{q}}^{(i)}| \tag{9}$$

For a dataset of N images, the final overall SLAB score is the mean value of the SLAB error computed for each image:

$$e_{slab} = \frac{1}{N}\sum_{i=1}^{N} e_{slab}^{(i)} \tag{10}$$

The SLAB error is adopted as the main metric also in this work, but to get more insights into the performances of the proposed algorithms (e.g. the error distributions, the effect of the background, etc.), the direct access to the ground truth relative pose of each processed image is needed. Unfortunately, only the images from the official benchmark test set for pose estimation algorithms proposed for SPEC2019 are publicly available, without annotations for the ground truth relative pose associated with each frame. The test set of 2400 images extracted from the actual SPEED dataset discussed in Section 3.2 and not involved in the training phases of the CNNs has been adopted to overcome this limitation and make it possible to have more insights into the performances of the proposed algorithm. The official SLAB score, obtained by participating in the postmortem SPEC2019 competition, was compared against the SLAB score achieved by processing the labeled test set defined for this work, as done in [58], to check the consistency of the labeled test set with the official SPEC2019 benchmark. Please notice that the pipelines adopted to process the labeled test set and to obtain the official score from SPEC2019 are the same with all the hyperparameters frozen, to avoid invalidating the consistency check.

4.2. SPEC2019 score and validation of the test set

The baseline architecture is composed of YOLOv5s as the target detector and M-LSD as line segments and keypoints detector. The hyperparameters of the image processing and line segment grouping steps have been tuned on the test set, while the CNNs have been applied "as they are" after the training phase on the dedicated dataset to avoid overfitting on the test set. The tuning of the hyperparameters (i.e., the threshold values for the line merging and grouping steps) followed a trial-and-error approach aimed at reducing the mean SLAB score computed for the labeled test set extracted from SPEED (please refer to Section 3.2) starting from the values reported in [63] for a re-implementation of the SVD algorithm. The baseline architecture was applied to the SPEC2019 test images to obtain an "official" scoring. Then the achieved SPEC2019 score in terms of SLAB error was compared with the score on the labeled test set to check the representativeness of the latter, allowing us to proceed with a more in-depth analysis of the errors registered. The baseline YOLOv5s/M-LSD architecture is depicted in detail in Fig. 12. The proposed baseline architecture in Fig. 12 (with the complete match matrix evaluation) scored a SLAB error on the test set equal to 0.04552. In the postmortem SPEC2019 competition, a SLAB score of 0.04622 in synthetic images and 0.12546 on real mock-up images were achieved. These results can be retrieved from the official website of SPEC2019 from the postmortem competition leaderboard,[2]. A print of the website page with the official scoring as of 29 November 2022 (i.e., the date of the last submission) is reported in Fig. 13 where the top-10 scoring teams are sorted based on the SLAB score on synthetic images test set (last column in Fig. 13). At the time of the last best submission the YOLOv5s/M-LSD proposed classified 9th in SPEC2019 on the synthetic test set. Together with the synthetic images, the proposed architecture was tested also on 300 images taken from a representative mock-up of Tango, achieving a score of 0.12546 ("Real Image Score" column in Fig. 13). Notably, the achieved result on real images is the 2nd best score among the top-10 architectures, outperforming also the score of UniAdelaide (winner of the official competition). This outcome demonstrates the generalization capabilities of the proposed architecture if applied to a dataset of unseen images with a wide domain gap [20] with respect to the training images. Comparing the score of the proposed architecture with the outcomes of the teams that participated to the official SPEC2019 competition[3] it arises that the YOLOv5s/M-LSD would have been ranked 3rd based on the results on synthetic images, after UniAdelaide [59] and EPFL_cvlab [108], and 2nd on real images, after EPFL_cvlab. Overall, the proposed architecture outperforms on both real and synthetic images most of the architectures, including the SLAB baseline architecture [57] and the fully-CNN-driven method in [61], proposing an implementation with lightweight CNNs for mobile devices. The lightness of adopted CNNs is not a shared feature with the top-performing architectures that rely on heavy architectures that run on GPUs only. Notably, the scores demonstrate that lightweight CNNs can be adopted for relative pose estimation tasks with a level of accuracy comparable with more complex architectures. Examples of ROI and line segments extracted with the YOLOv5s/M-LSD pipeline on synthetic images from the official SPEC2019 test set are reported in Fig. 14 for scenarios spanning from the case of a black background with Tango well-illuminated to almost totally shadowed target with Earth horizon in the image. The YOLOv5s is capable of detecting the spacecraft in all conditions with a highly accurate ROI extraction, also in challenging images with Tango almost non-visible and background entirely filled by the Earth. The same holds for the line segments and keypoints detection. The M-LSD can extract lines and keypoints from all the challenging scenarios, also in case of a high amount of details in the image processed.

Furthermore, the generalization capabilities of lightweight CNNs to real image datasets are impressive, outperforming the scores of most of the other proposed architectures. Fig. 15 shows examples of ROI extraction performed on real mock-up images from SPEC2019, along with the line segments and keypoints regressed by the M-LSD. The performances are slightly degraded with respect to the case of synthetic images, presenting more relaxed bounding boxes and line segment outliers. Nevertheless, the fundamental elements needed to retrieve a correct pose estimate, like the edges of the top solar panel and most of the appendages (e.g., antennas), are correctly detected. Please notice that the line merging and grouping processes, along with the retrieval of the pose through the testing of the correspondences in the match matrix, make the algorithm robust against the presence of outliers in the detected line segments. Despite the lack of ground truth labels for the mock-up images, by reprojecting the 3D wireframe on the image through the estimated pose and performing a visual inspection of the obtained images, it is possible to have some hints on the pose estimation performances. Namely, the worst performances in terms of accuracy of the estimated relative pose are achieved for images where the target is partially out of the image, as shown in Fig. 16, with estimation errors higher than the cases in which there are outliers

[2] Data available in the postmortemleaderboardpage of SPEC2019 [retrieved 29 November 2022].

[3] The official leaderboard can be retrieved from the dedicatedsection in SPEC2019 website

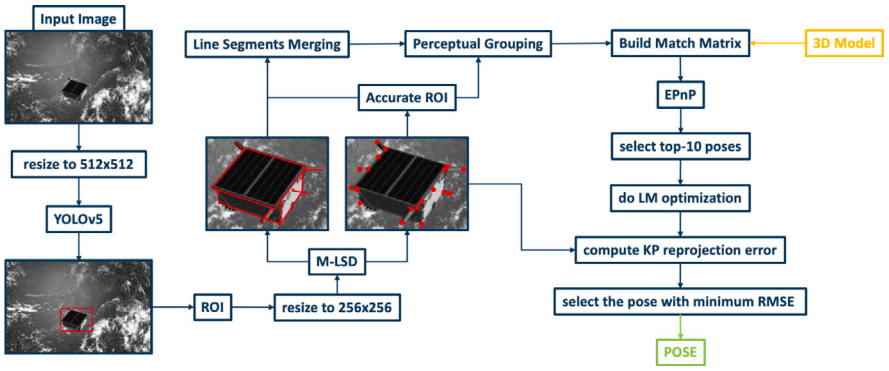

Fig. 12. Detailed scheme of YOLO/M-LSD architecture.

Post Mortem Leaderboard

Name	Submissions	Last Submission	Best Submission	Real Image Score	Best Score
competition winner UniAdelaide				0.36340645622528017	0.00864899489025079
arunkumar04	5	June 11, 2020, 10:09 a.m.	June 10, 2020, 11:22 p.m.	0.2897316198709749	0.009653543468538944
lingge	6	Nov. 21, 2022, 8:59 a.m.	Aug. 2, 2022, 4:38 p.m.	1.0459772894830643	0.011728576028699285
wangzi_nudt	28	Feb. 4, 2021, 5:03 a.m.	Feb. 3, 2021, 12:35 a.m.	0.16838921336519297	0.012316958900752597
lteam	10	March 15, 2022, 7:25 a.m.	March 15, 2022, 7:25 a.m.	0.6295857722564993	0.019300729122915965
u3s_lab	7	June 12, 2021, 1:01 p.m.	June 12, 2021, 1:01 p.m.	0.24377389519267023	0.027463077733630353
UT-TSL	1	July 29, 2020, 4:46 p.m.	July 29, 2020, 4:46 p.m.	0.2918232061918603	0.04088880831356042
massimo.piazza	5	Dec. 2, 2020, 1:44 p.m.	Dec. 2, 2020, 1:44 p.m.	0.12025061876822872	0.04500206999644395
michele.bechini	6	Nov. 29, 2022, 11:13 a.m.	Nov. 29, 2022, 11:13 a.m.	0.12546116819864492	0.0462192325747491
haoranhuang	91	March 17,	March 16,	0.23658816520264792	0.050840141343021505

Created by the Advanced Concepts Team, Copyright © European Space Agency 2021

Fig. 13. Print of the SPEC2019 postmortem leaderboard.

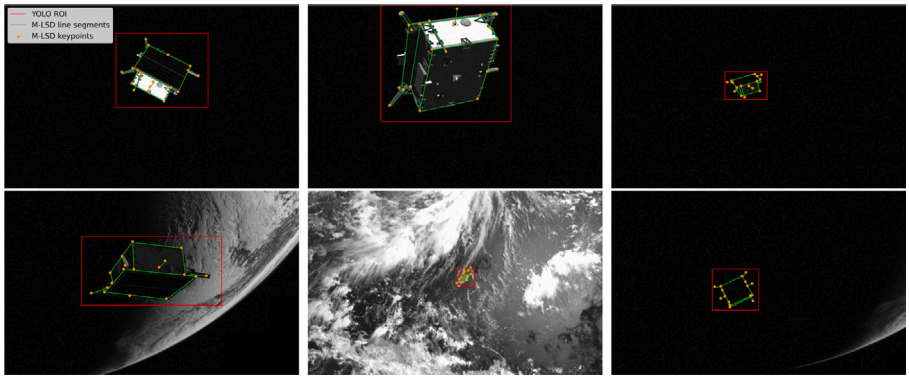

Fig. 14. Examples of extracted ROI, line segments, and keypoints from SPEC2019 test set.

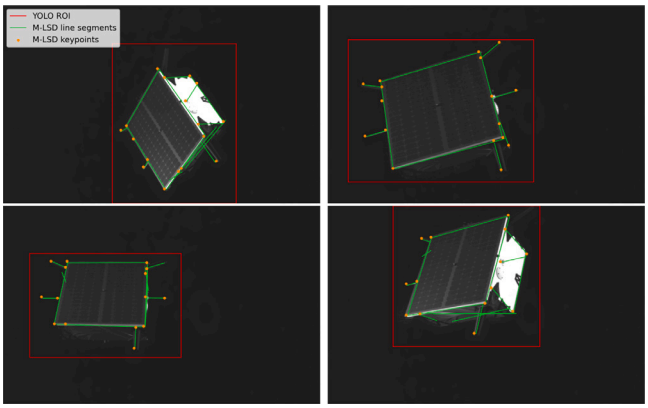

Fig. 15. Examples of extracted ROI, line segments, and keypoints from SPEC2019 real test set.

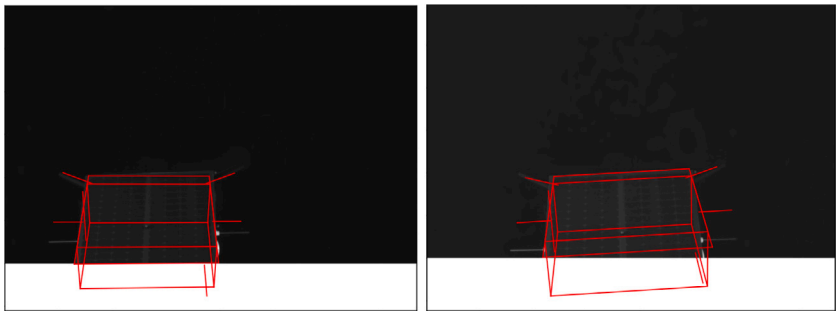

Fig. 16. Examples of low-accuracy reprojected 3D wireframe using estimated relative pose on SPEC2019 real test set.

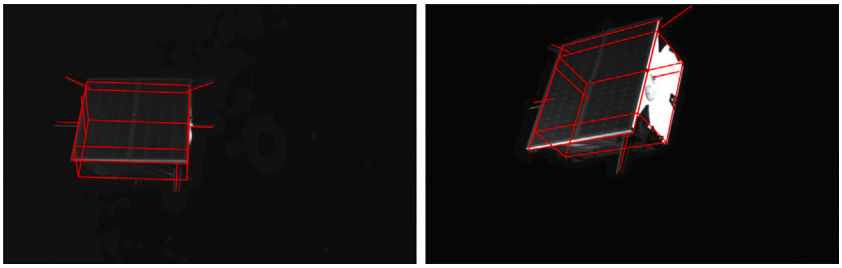

Fig. 17. Examples of high-accuracy reprojected 3D wireframe using estimated relative pose on SPEC2019 real test set.

in the line segments detected. This can be confirmed by comparing the reprojected wireframes in Fig. 16 with those in Fig. 17, obtained from the line segments detected shown in the second row of Fig. 15 (i.e., where the outliers in the line segments detected are evident).

This drawback in the proposed architecture is expected since the pipeline relies on complete line correspondences, as in the SVD, hence if the target is partially outside the FOV of the camera and partial or incomplete lines are detected, they are matched with complete lines from the 3D wireframe models, leading to wrong estimates. Please notice that among the real mock-up images included in the SPEC2019 test set there is a high percentage of pictures in which Tango is partially outside the FOV of the camera, justifying the higher official SLAB error scored by the proposed architecture on mock-up pictures (e_{SLAB} = 0.12546) with respect to the official score achieved on synthetic images (e_{SLAB} = 0.04622).

The insights retrieved without the knowledge of the ground truth relative pose of SPEC2019 images offer an overview of the performances of the proposed pipeline. Despite this, more in-depth analyses are required to pursue the inclusion of the proposed initialization step in a relative navigation chain, which is the final objective of this study. The SLAB error scored on SPEC2019 synthetic test images is comparable with the score on the labeled test set extracted from SPEC2019 training images. As a result, the labeled test set can be considered representative of the SPEC2019 synthetic test set, hence other analyses can be carried out on that labeled set.

4.3. YOLOv5s/M-LSD performances

The YOLOv5s/M-LSD architecture achieved a SLAB score on the 2400 labeled test images from SPEED equal to 0.04552, with a standard deviation of 0.22972. The SLAB score reported has been computed as in Eq. (10). Hence, e_{SLAB} is the mean value of the SLAB error achieved in all the images. Despite the SLAB error being a good scalar metric to compare easily different architectures, it is difficult to get the real performances in terms of errors of estimation in translation and rotations from e_{SLAB} only. Consequently, the estimation errors distributions on the test set have been evaluated to ease the evaluation of the capabilities of the proposed architecture. The estimation errors are expressed as mean value and standard deviation of the quantities employed in the computation of the SLAB error. In addition to the quantities reported in Eq. (6), Eq. (7), and Eq. (8), the angular error $\mathbf{E}_\theta$ in terms of axis misalignment between reference and estimated relative attitude is also considered. These angles are computed as:

$$\theta_i = \arccos\left(\mathbf{r}_{est,i} \cdot \mathbf{r}_{GT,i}\right) \quad \text{with i} = 1, 2, 3 \tag{11}$$

Table 7
YOLOv5s/M-LSD error metrics scores.

Metric	Mean value	Standard deviation
E_t [cm]	8.61	39.77
$\mathbf{E}_t$ [cm]	[−0.26, −0.19, 0.98]	[2.09, 1.77, 40.59]
$\mathbf{e}_t$ [–]	[−0.305, −0.246, 0.680]×10^{-3}	[0.178, 0.154, 2.827]×10^{-2}
e_q [deg]	2.19	12.59
$\mathbf{E}_\theta$ [deg]	[1.86, 1.76, 1.20]	[11.98, 11.07, 6.11]
e_{SLAB} [–]	0.04552	0.22972

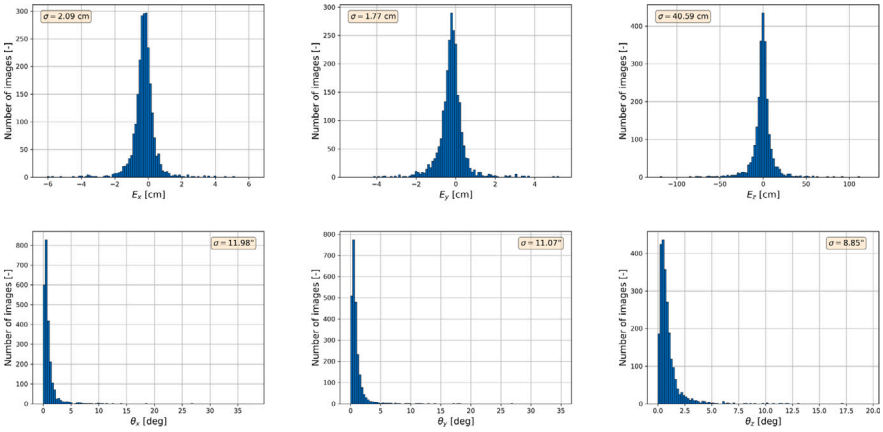

Fig. 18. Errors distributions in 3σ range.

where $\mathbf{r}_{est,i}$ and $\mathbf{r}_{GT,i}$ are the ith columns of the estimated relative attitude $\mathbf{R}_{est} = \mathbf{R}^{est}_{TRG\to CAM}$ and the ground truth $\mathbf{R}_{GT} = \mathbf{R}^{GT}_{TRG\to CAM}$ respectively. The resulting angles θ_x, θ_y, θ_z give the angular estimation error of each axis of the target body reference frame with respect to the camera reference frame.

Table 7 reports the metrics values achieved by the proposed baseline architecture on the labeled test set, while Fig. 18 shows the histograms of the error distributions in the 3σ range.

Concerning the translational error, the proposed architecture can estimate the relative distance with an average error in the order of a few centimeters, albeit the z-axis error is about one order of magnitude higher than these of the x and y axes concerning both the mean value and the standard deviation. This behavior is frequent in literature [18,58,59]. For the case of the architecture proposed here, it can arise from a wrong line segments detection (like in the case of Tango partially out of the FOV or when the target is far from the camera in poor visibility conditions) that causes a drift mostly along the camera boresight (i.e., positive along the camera z-axis) when the relative pose is retrieved from the 2D to 3D correspondences. The drift along the other axes is less pronounced due to the target detection step that bounds the lines extracted by the M-LSD to be inside the ROI. Please notice that drift along the x and y axes in the camera frame mean that the extracted line segments would be shifted from the actual position in the image, resulting to be out of the ROI. The same behavior is pointed out also by the standard deviations computed for each axis that indeed show a σ value for the z component of the translation that is one order of magnitude higher than the other two components.

The analysis of the scores about the rotational errors shows that the mean value is bounded to be in the order of a few degrees. From the evaluation of the angles defined in Eq. (11), it can be noticed that the average error is almost equal for the three axes, with a mean value in the order of about 1.5 degrees. Despite that, the errors about the x and y axes show an uncertainty level higher than the error about the z-axis. This behavior is addressed by looking at the outcomes of the proposed architecture. The geometry of Tango shows almost symmetric features if the spacecraft is observed from positive and negative y direction in the target reference frame (please refer to Fig. 11). Therefore, this can cause ambiguity in the relative pose when the camera boresight axis (i.e., z-axis) is almost aligned with the target y-axis. Intuitively, if the actual relative attitude is such that the camera z-axis is pointing straight towards the -y direction of the target reference frame, but the estimated attitude is such that the camera is pointing in the opposite direction, the angular errors θ_i will be $\mathbf{E}_\theta = [\theta_x, \theta_y, \theta_z] = [180, 180, 0]$ degrees. This ambiguity is handled here by considering all the five principal appendages of Tango (instead of the more classical choice of using only the three larger antennas) and, implicitly, by selecting the best pose estimate among a pool of candidate poses that arise from all the possible combinations of feature groups. Despite that, due to the symmetry previously mentioned and to the fact that some other ambiguities may arise if Tango is observed with the camera pointing axis almost parallel to the x-axis of the target when the front and the back antennas have comparable length and direction in the image frame, some outliers in the attitude estimation are present, driving the dispersion of θ_x and θ_y towards higher values than θ_z, that is not strongly affected by those symmetries. It is acknowledged that the robustness against pose ambiguities can be further improved by relying on some surface features to break the symmetries and disambiguate the estimated pose. For instance, the highly reflective panel on the +y side of the Tango can be employed to improve the robustness against the first symmetry discussed above. Notably, the fact that the distributions of the rotational errors are in the same order of magnitude demonstrates that the features selected and the proposed architecture are adequate to retrieve a relative pose with a small rotational error in most of the cases.

4.3.1. Effect of relative distance

The inter-spacecraft relative distance plays a fundamental role in the goodness of the estimated relative pose since the feature extraction step for a target far away from the camera may be affected by outliers due to the reduced size of the target in the image and, as a consequence, of a lower signal-to-noise ratio in the ROI-cropped image. The relative distance effects on the pose estimations error have been investigated for the YOLOv5s/M-LSD architecture by evaluating both E_t and e_q as a function of the ground truth relative distance. The outcomes of the proposed pose estimation pipeline have been sorted in ascending order of ground truth relative distance and grouped in 30 batches of 80

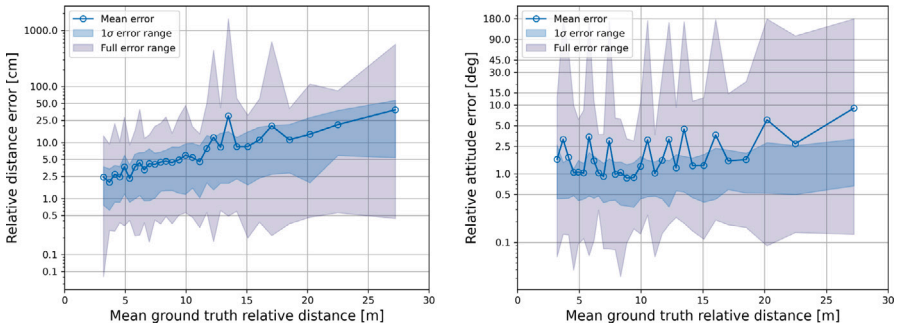

Fig. 19. Effects of relative distance on E_t (left) and e_q (right), y-axis in logarithmic scale.

images as in [58]. The mean of all the values of E_t and e_q are evaluated for each batch and reported in Fig. 19 (y-axis is in logarithmic scale) in correspondence of the mean ground truth distance of the related batch. The blue-shaded regions in Fig. 19 indicate the inter-quartile range of the errors in each batch corresponding to a 1σ range, i.e., the area spanning from the 15.87 to the 84.13 percentiles, being representative of the central trend of the error distribution within which most of the samples fall. The gray-shaded regions in Fig. 19 indicate the full error range, i.e., the region comprised between the maximum and minimum estimation error in each batch of images. The average values of E_t and e_q grow as the ground truth distance increases. The error increment is more pronounced in E_t, while for e_q the increment is slower. Notably, the increment in the uncertainty of the estimates as a function of the ground truth distance, represented by the increase (in the logarithmic-scale plot in Fig. 19) of the blue-shaded region, is higher for E_t and, conversely, it is almost constant for the relative attitude error. Despite this, note that an average translational error of 40 cm compared to a ground truth relative distance of about 27 m corresponds to an estimation error of about 1.48%. The spikes in the error that exit the blue-shaded regions (more evident in the relative attitude error plot) are due to outliers in the pose estimations, also highlighted by an abrupt widening of the gray-shaded region. Namely, when the mean error shows a peak outside the blue-shaded area it means that there is a very small number of largely wrong estimates such that the 16–84 percentile region remains almost constant (i.e., the wrong estimates are actually outliers), but the error associated to these outliers is high enough to strongly affect the mean value in the related batch. The presence of more spikes in the relative attitude plot compared with the single one detected in the relative distance error plot is due to the previously noticed ambiguities in the geometry of Tango that lead, in a few cases, the relative attitude estimates to huge angular errors. It is remarked that the number of unsolved ambiguities is low in number since the uncertainties on the relative attitude estimates are not strongly affected, as demonstrated by the blue shaded region in Fig. 19. To evaluate the contributions of the translational and rotational errors to the SLAB score and to assess the effect of the relative distance on e_{SLAB}, the same procedure adopted to generate Fig. 19 was applied to the SLAB error and its components e_t (normalized relative distance error) and e_q to generate Fig. 20. The blue-shaded region in Fig. 20 (y-axis is in logarithmic scale) spans again from 15.87 to 84.13 percentiles of the SLAB score, while the gray ones show the full error range. The normalized translational error is always below 1.5% except for a peak of close 2.4% to ~14 m of relative distance. From Fig. 20 it is evident that the SLAB score is biased towards attitude errors, as in [58]. This unbalance in the contribution to the final SLAB score is due to the quaternion error e_q that is not normalized thus, e_q is one order of magnitude higher than e_t. The uncertainty of the SLAB score represented by the shaded regions of Fig. 20 starts slightly increasing at about 15 m, where the uncertainty of the translational error begins to grow (see also Fig. 19) but, overall, the trend is dominated by the uncertainties in the attitude error. Please notice that there is a peak

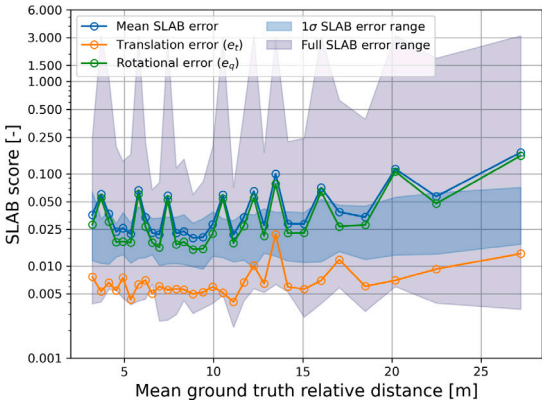

Fig. 20. Effects of relative distance on the SLAB score, y-axis in logarithmic scale.

in the spread of the blue-shaded region in correspondence of the first batch (mean ground truth relative distance of ~3 m). This increment in the uncertainty of the estimate with respect to the neighboring batches is driven again by the uncertainties on the attitude estimation, and it is due to the presence of images with the target partially outside of the FOV that lead to wrong attitude estimations, as outlined in Section 4.2.

4.3.2. *Effect of image background*

The training set of all the CNNs adopted includes images with black background (representing the deep space without visible celestial bodies) and with the Earth in several illumination conditions. Despite that, the presence of the Earth acts as a strong disturbance, especially for feature extraction. To assess the effects of image background on the SLAB score, Fig. 21 shows the SLAB scores achieved by the proposed architecture in the test images sorted in ascending order and differentiated in color depending on the background of the image. Examples of test images with the associated SLAB score are additionally plotted in Fig. 21 to ease the reading of the plot. The plot shows that most images with an associated low SLAB score are with a black background while, as the SLAB score increases, the presence of images with Earth in the background increases. The worst estimates ($e_{SLAB} \approx 3$) were obtained for images with a challenging scenario given by a combination of Earth in the background, particularly challenging illumination and visibility conditions, and high relative distances. For the last portion comprised between 90% and 100% (i.e., 10%) of the dataset where Fig. 21 shows a steep increment of the SLAB score, most of the images show at least one of the characteristics mentioned above that degrades the pose estimation performances. A more in-depth analysis of the failure conditions in the images with the highest SLAB score pointed out that the YOLOv5s is capable of detecting the target with high accuracy, while the M-LSD struggles in detecting accurate line segments and keypoints, leading to high errors in the pose estimate, driven mainly by the estimated relative attitude. Namely, in the worst case overall,

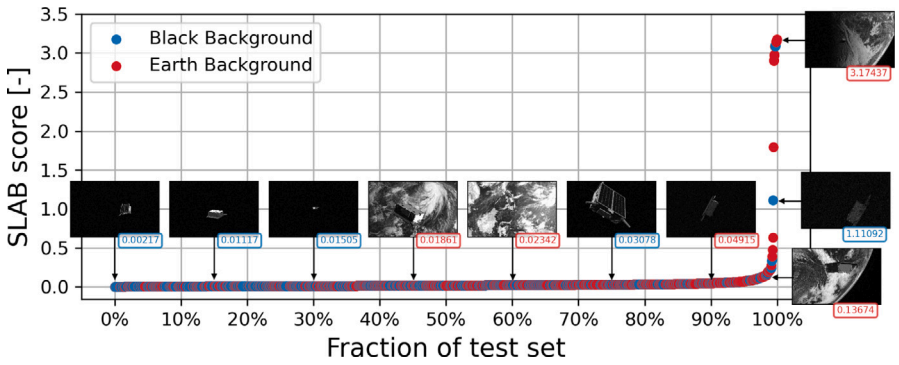

Fig. 21. Effects of image background on the SLAB score.

the YOLOv5s can correctly detect the target, contrarily the M-LSD fails to extract most of the line segments associated with the antennas due to an almost null contrast with respect to the Earth in the background in low illumination conditions. Notably, the fraction of images of the test set with a SLAB error higher than 0.05 is about 10%, while a SLAB error higher than 0.15 is obtained on about 1.5% of the test set.

4.3.3. Effect of reduced match matrix

The results discussed above on the performances of the YOLOv5s/M-LSD architecture derive from the adoption of the complete match matrix for 2D to 3D feature groups correspondences but, as outlined in Section 3.5, it is possible also to use a reduced form of the match matrix. The reduced match matrix can be adopted since the most complex feature groups are given by combinations of the simpler ones. Intuitively, the closed polygonal tetrads derive from combinations of proximity triads defined by merging proximity pairs. Hence, by increasing the "trust level" on the correctness of the line segments detected by the M-LSD and on the feature grouping process, it is possible to discard the simpler groups before the definition of the match matrix, retaining only the most complex ones available for the current image being processed after the definition of all the possible feature groups. The reduced form of the match matrix adopted here uses only the two most complex feature groups available for each image processed selecting them, depending on the current availability, among closed tetrads, proximity triads, parallel pairs, and proximity pairs, with the addition of parallel triads when available. Including the parallel triads reduces the dependencies on the detected antennas and their accuracy, increasing the robustness against line segments wrongly classified as antennas. Notably, this process will lower the computational time required since the EPnP must test fewer correspondences. As mentioned in [23], the most complex feature groups show fewer occurrences than the simpler ones. Moreover, complex feature groups are less prone to erroneous classifications hence the adoption of a reduced match matrix is a viable solution. The performances of the YOLOv5s/M-LSD architecture using the reduced match matrix on the labeled test set of 2400 images are given in Table 8.

Comparing the scores in Table 8 with those achieved by the YOLOv5s/M-LSD with full match matrix (reported in Table 7), it arises that, despite the version with the reduced match matrix performs slightly worse than the baseline, the errors are in the same order of magnitude and all the scores are comparable. The reduction of the match matrix causes an increment of the mean SLAB score of ~0.3% while the mean values of E_t and e_q slightly decrease. Notably, the small increment in the SLAB error is supposed to be related to the presence of more outliers compared to their number in the case of the baseline architecture. This hypothesis is confirmed by the fact that the uncertainties represented by the computed standard deviations increase while, on the contrary, the mean values of almost all the metrics slightly decrease. The increment is more pronounced in the z-component of the translational error and of the attitude error expressed by $\mathbf{E}_\theta$. Hence, the comparison of the performances between the baseline architecture and the YOLOv5s/M-LSD pipeline with the reduced match matrix shows that using a reduced match matrix is feasible despite the relative pose estimates being affected by a slightly higher uncertainty level if compared to those obtained by using the full match matrix. Furthermore, the main advantage of reducing the match matrix size is the expected lowering of the overall computational time since testing the correspondences in the match matrix with the EPnP is the most time-consuming step of the entire pipeline. This aspect is detailed in Section 4.3.4.

Table 8
YOLOv5s/M-LSD (reduced match matrix) error metrics scores.

Metric	Mean value	Standard deviation
E_t [cm]	8.42	40.61
$\mathbf{E}_t$ [cm]	[−0.32, −0.19, 0.73]	[3.41, 1.82, 41.28]
$\mathbf{e}_t$ [–]	[−0.478, −0.251, 0.825]×10^{-3}	[0.862, 0.207, 4.270]×10^{-2}
e_q [deg]	2.18	12.60
$\mathbf{E}_\theta$ [deg]	[1.78, 1.74, 1.36]	[11.36, 10.81, 7.71]
e_{SLAB} [–]	0.04565	0.23984

4.3.4. Runtime evaluation

The running time of the proposed architecture has been tested on an ARM-based processor, the Apple® Silicon™ M1 Pro, using the CPU only. The running time has been evaluated for all the images in the test set by evaluating the time needed for the target detector CNN, the time required by the M-LSD, and the time elapsed from the instant of time when the output of the M-LSD is available to the retrieval of the final pose estimate. Forecasting future tests on representative or actual spaceborne hardware, the CNN models have been exported in TFLite format[4] before conducting all the tests reported through the paper. Notice that the TFLite format is optimized for ARM processors hence it is compatible with the Apple® Silicon™ M1 Pro used here. Fig. 22 shows the execution time registered for the baseline architecture for each image in the test set. The plot is on a logarithmic scale and the processing times of each sub-component of the entire pipeline are stacked vertically to give an overview of the total computational time. On average, the time needed for each step of the proposed pipeline is $t_{YOLO} = 0.085\,\text{s}$, $t_{MLSD} = 0.088\,\text{s}$, and $t_{Pose} = 6.41\,\text{s}$, giving a total mean running time of $\sim 6.58\,\text{s}$. The baseline architecture lowers the overall running time of about 20% with respect to the original SVD algorithms, which on average required 8.22 s to generate a pose solution, as reported in [23]. In agreement with [23], from Fig. 22 it is evident that the most time-consuming phase is the testing of all the correspondences in the match matrix with the EPnP. Notably, the running times of both YOLOv5s and M-LSD are almost

[4] https://www.tensorflow.org/lite

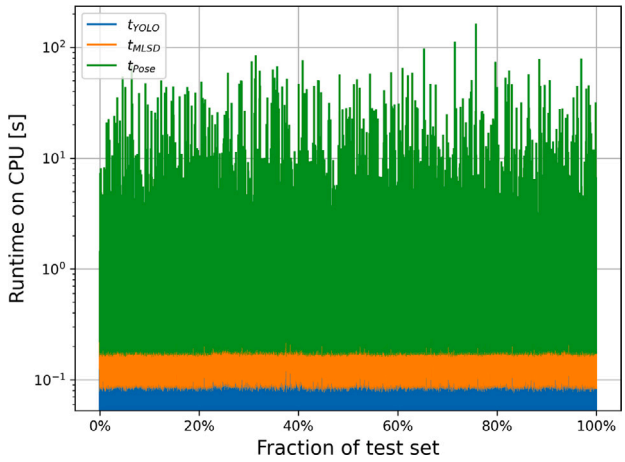

Fig. 22. Runtime of YOLOv5s/M-LSD architecture with full match matrix on the test set.

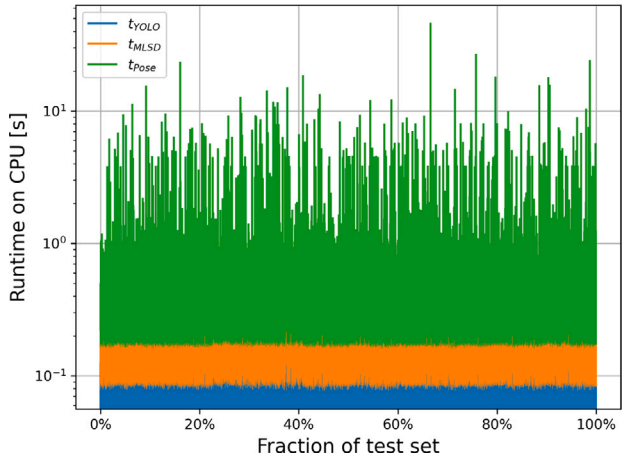

Fig. 23. Runtime of YOLOv5s/M-LSD architecture with reduced match matrix on the test set.

constant across the dataset (proving that the non-maximal suppression of YOLOv5s has been correctly removed), while the time needed for the last phase shows a higher variability. These oscillations in t_{Pose} are due to the fact that the time required scales linearly with the number of correspondences in the match matrix.

Despite the proposed architecture being meant to be used only for relative pose initialization, while the pose-tracking phase can theoretically employ only the outputs of the M-LSD, and although a remarkable improvement with respect to the SVD in terms of computational time has already been achieved, the mean total running time scored by the baseline architecture may be prohibitive for practical applications. Relying on the performances of the YOLOv5s/M-LSD architecture with reduced match matrix in Table 8, which shows comparable performances on the test set if compared to the full match matrix version, and since the most time demanding step is related to the match matrix itself, using the reduced match matrix can bring substantial advantages in terms of computational time without strong drawbacks on the accuracy performances. The runtime breakdown of the reduced match matrix is shown in Fig. 23. The average time needed to retrieve the relative pose from the output of the M-LSD drastically drops by using the reduced match matrix to $t_{Pose} = 1.25\,\text{s}$ with a decrease of about 80.5% with respect to the case with a full match matrix. The total average runtime drops to $\sim 1.42\,\text{s}$, with a decrement of about 82.7% with respect to SVD. A better comparison of the advantages in terms of total computational time offered by the introduction of the reduced

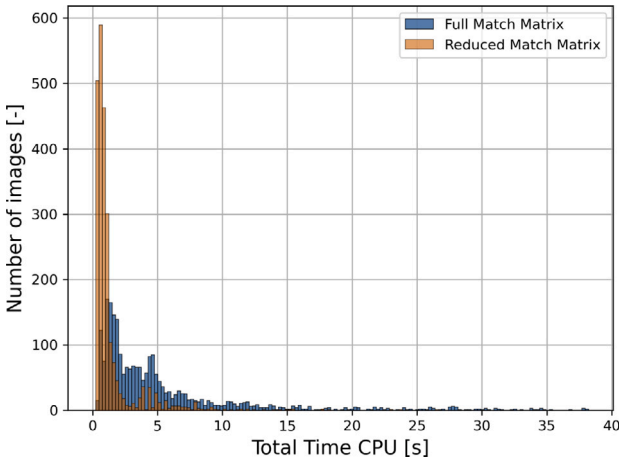

Fig. 24. Histogram comparison of total runtime of YOLOv5s/M-LSD architecture across the test set using reduced and full match matrix.

match matrix with respect to the baseline full match matrix is given in the histogram plot in Fig. 24. The histograms are truncated to a total runtime of 40 s for readability and each bin covers a region of 0.25 s. From Fig. 24 it is evident that for about the 75% of the images the total running time using the reduced match matrix is less or equal to 1.25 s, while using the full match matrix leads to a spread of the total time in the range from 0.25 s to 10 s for most of the images, with some cases in which the running time is higher than 30 s. From the results in terms of accuracy in Section 4.3.3 and from the improvements in terms of total runtime showed in Fig. 24, it can be concluded that the YOLOv5s/M-LSD architecture with reduced match matrix is a good candidate for actual application for autonomous pose initialization, offering a good compromise between accuracy and computational complexity. It is acknowledged that the pipeline is currently coded in *Python 3.8*. Hence, the total running time can benefit from an optimized implementation in suitable languages with specific handling of the CNN models using export formats for embedded computation (e.g., OpenVINO™) or dedicated accelerated inference through field programmable gate arrays (FPGAs). The assessment of accuracy and total runtime on flight-representative hardware is foreseen as future development.

4.4. Comparison with other methods

The results of the tests performed pointed out that the proposed YOLOv5s/M-LSD architecture for pose initialization achieves competitive SLAB error scores compared with other architectures that participated in the SPEC2019.

Table 9 compares the scores of the proposed approach against some top-performing architectures. Moreover, the SVD algorithm is included in the comparison (summarized in Table 9), since it inspired the proposed architecture.

The proposed architecture outperforms the SVD (both standard and high confidence) in all the metrics with a significant increase in accuracy. The most significant improvement has been achieved in terms of the availability of solutions since the proposed architecture is capable of giving an output pose for all the images in the test set, while the SVD produced an output only for a small subset of images. Furthermore, the proposed approach outperforms the SVD also in terms of computational time, as discussed in Section 4.3.4, with a decrement of 20% in the full match matrix version and 82.7% in the reduced match matrix architecture. Notably, the results for the SVD are achieved on real images from the mission PRISMA. Those images are not available for testing, but the scores of the proposed method on mock-up images included in the SPEC2019 test set pointed

Table 9
Performance comparison of YOLOv5s/M-LSD architecture against other methods on SPEED images.

Architecture	Mean E_t [cm]	Mean e_q [deg]	Solution availability
SVD [23]	146	38.99	50%
SVD (high confidence) [23]	51	2.76	20%
UniAdelaide [59]	3.2	0.41	100%
EPFL_cvlab [108]	7.3	0.91	100%
massimo.piazza (RPEP) [58]	10.36	2.24	100%
pedro_fairspace [61]	14.5	2.46	100%
SLAB Baseline [57]	20.9	2.62	100%
Ours (full match matrix)	8.61	2.19	100%
Ours (reduced match matrix)	8.42	2.18	100%

out that the YOLOv5s/M-LSD architecture well generalize to unseen real images, ranking 2nd place among the teams that participated in the actual SPEC2019 and 2nd among the top-10 teams participating in the postmortem SPEC2019. Comparing the proposed architecture with the scores of other participants in SPEC2019 that adopted a CNN-based approach, it arises that only UniAdelaide and EPFL_cvlab achieved higher mean accuracy. Instead, only EPFL_cvlab and the RPEP achieved higher scores on real images. Both our approaches (full and reduced match matrix versions) outperformed the SLAB baseline and the pedro_fairspace methods, showing high-level performances. Concerning the results against the RPEP, the YOLOv5s/M-LSD architectures achieved better accuracy in mean rotational and translational estimation. However, the RPEP achieved a slightly lower SLAB score due to a lower uncertainty on the estimated poses. UniAdelaide and EPFL_cvlab achieved better performances with respect to the pose initialization scheme discussed here, even if the results on mock-up and synthetic images of UniAdelaide may point out that the algorithm adopted by this team is overfitting on synthetic data due to the poor performances on real images. Notably, the CNNs adopted here are lightweight and proven to be suitable for inference in mobile devices, hence they offer a reduced computational complexity that makes them applicable for testing with spaceborne hardware, even if dedicated tests still need to be performed. On the contrary, both UniAdelaide and EPFL_cvlab adopted large and computationally expensive CNN models that may be unsuited for standard spaceborne onboard computers.

5. Conclusions and future works

The research in the field of autonomous spacecraft navigation faced a boost in the last few years with the introduction of CNN-based algorithms, the publication of image datasets, and the organization of international competitions. Most of the effort has been put into the pose initialization from monocular images, i.e., the very first step of autonomous GNC chains. The necessity of developing lightweight CNN-based architecture and proving their accuracy arises by analyzing the outcomes of the competitions and several architectures available in the literature. Most of the algorithms developed for pose initialization with uncooperative targets were designed to achieve top scores in the competitions used as benchmarks, disregarding the computational effort required if compared with available flight hardware, resulting in extremely accurate algorithms but with limited applicability. In this work, the issue is addressed by proposing a relative pose initialization scheme that relies only on lightweight CNNs capable of running with low inference time on mobile devices to retrieve the chaser-to-target relative state provided a single monocular image and the knowledge of the target's 3D wireframe model. The Sharma-Ventura-D'Amico (SVD) algorithm inspired the proposed scheme that leverages three steps: target detection, line segments and keypoint extraction, and the pose solver. The target detection step adopts the YOLOv5s object detection model. The trained architecture scored an average precision of $AP_{50}^{95} = 97.6\%$ and a mean IoU of 96.46%, which is the highest value scored on SPEED images if compared to other publicly available architectures developed for the target detection task. Instead of adopting a two-stream architecture as in the SVD, the proposed approach leverages a single-stream features detection step based on the M-LSD, resulting in a lighter algorithm. The original M-LSD architecture has been modified to output line segments and line junctions (i.e., keypoints), scoring a mean structural average precision on the SPEED/SPEED+ validation set of $sAP = 63.77$. The last step is based on the definition of 2D-to−3D correspondences of line segment endpoints subsequently solved with the EPnP to retrieve the final relative pose estimate. The baseline architecture with the full match matrix achieved a SLAB score in the postmortem SPEC2019 competition of 0.04622 in synthetic images and 0.12546 on mock-up images, entering the top-10 performing architectures but being the only one based on lightweight CNNs, sufficiently fast to be actually considered as a potential candidate to run on flight hardware. From the more in-depth analysis, the baseline architecture scores an absolute mean translation error of about 8.6 cm and a quaternion error of 2.2 degrees, pointing out the high level of accuracy that the YOLOv5s/M-LSD architecture proposed can reach. Regarding the uncertainties in the pose estimated, the analyses revealed that the highest error is achieved in the translation component aligned with the camera boresight axis, with uncertainty levels one order of magnitude higher than the other components, while the uncertainties on the relative attitude are more pronounced on the x and y-axis of the target due to ambiguities on the target geometry, even if the mean values of the angular errors are almost equal for the three axes. Test results showed how achievable accuracy correlates with the inter-spacecraft relative distance and the presence of the Earth in the background combined with the low visibility of the target. Concerning the relative distance, it has been noticed a progressive drop in the accuracy (mainly in the relative translation) related to the higher uncertainties in the line segments detected when the relative distance increase. Also, the presence of the Earth in the background, combined with poor visibility of the target, entails an increment in the estimation error in a few cases. Again, this is related to the accuracy of the line segment detection that deteriorates in the presence of weak contrast between the target and the background. The reduction of the match matrix, leveraging only on the most complex geometrical groups, entails a strong reduction in the computational time, bringing only minor deteriorations in the overall accuracy. The YOLOv5s/M-LSD architecture with reduced match matrix achieved comparable accuracy scores with respect to the full match matrix version, while the computational time drops by a factor 5 on CPU. It is remarked that the proposed scheme is meant only for pose initialization tasks, while the possibility of leveraging keypoints detected by M-LSD to perform pose tracking should be explored in future developments. Additionally, an assessment of the lighting conditions that may limit the applicability of the proposed approach with both synthetic images and frames acquired with hardware-in-the-loop to simulate real camera noises is pointed out as future work. As mentioned, despite the approach adopted being general, the pipeline presented has been tailored to work with Tango as the target. Hence, the geometry of possible new targets should be evaluated carefully, and the proposed pipeline should be adapted accordingly before being applied.

Overall, as the main outcome, the proposed architecture demonstrated performances in the range of top-scoring algorithms in

SPEC2019 leveraging only on lightweight CNNs, with low mean errors both in translation and rotation, strongly improving the results achieved by the original SVD in terms of solution availability, accuracy and, remarkably, computational time (with a reduction of about 82.7% with the reduced match matrix), proving that more computationally efficient CNNs can still achieve high-level performances in relative navigation critical tasks. The achieved performances make this algorithm a promising candidate for testing on flight representative hardware.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] J.A. Starek, B. Açıkmeşe, I.A. Nesnas, M. Pavone, Spacecraft autonomy challenges for next-generation space missions, in: E. Feron (Ed.), Advances in Control System Technology for Aerospace Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 1–48, http://dx.doi.org/10.1007/978-3-662-47694-9_1.

[2] L. Breger, J.P. How, Safe trajectories for autonomous rendezvous of spacecraft, J. Guid. Control Dyn. 31 (5) (2008) 1478–1489, http://dx.doi.org/10.2514/1.29590.

[3] C. Bonnal, J.-M. Ruault, M.-C. Desjean, Active debris removal: Recent progress and current trends, Acta Astronaut. 85 (2013) 51–60, http://dx.doi.org/10.1016/j.actaastro.2012.11.009.

[4] S. Silvestrini, J. Prinetto, G. Zanotti, M. Lavagna, Design of robust passively safe relative trajectories for uncooperative debris imaging in preparation to removal, in: 2020 AAS/AIAA Astrodynamics Specialist Conference, Vol. 175, Univelt, 2020, pp. 4205–4222.

[5] R. Biesbroek, S. Aziz, A. Wolahan, S. Cipolla, M. Richard-Noca, L. Piguet, The clearspace-1 mission: Esa and clearspace team up to remove debris, in: Proc. 8th Eur. Conf. Sp. Debris, 2021, pp. 1–3.

[6] P. Lunghi, M. Ciarambino, M. Lavagna, A multilayer perceptron hazard detector for vision-based autonomous planetary landing, Adv. Space Res. 58 (1) (2016) 131–144, http://dx.doi.org/10.1016/j.asr.2016.04.012.

[7] S. Silvestrini, M. Piccinin, G. Zanotti, A. Brandonisio, I. Bloise, L. Feruglio, P. Lunghi, M. Lavagna, M. Varile, Optical navigation for Lunar landing based on Convolutional Neural Network crater detector, Aerosp. Sci. Technol. 123 (2022) 107503, http://dx.doi.org/10.1016/j.ast.2022.107503.

[8] G. Di Mauro, M. Lawn, R. Bevilacqua, Survey on guidance navigation and control requirements for spacecraft formation-flying missions, J. Guid. Control Dyn. 41 (3) (2018) 581–602, http://dx.doi.org/10.2514/1.G002868.

[9] S. Silvestrini, M. Lavagna, Neural-based predictive control for safe autonomous spacecraft relative maneuvers, J. Guid. Control Dyn. 44 (12) (2021) 2303–2310, http://dx.doi.org/10.2514/1.G005481.

[10] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations, Prog. Aerosp. Sci. 93 (2017) 53–72, http://dx.doi.org/10.1016/j.paerosci.2017.07.001.

[11] C. Bamann, U. Hugentobler, Accurate orbit determination of space debris with laser tracking, in: Proceedings of 7th European Conference on Space Debris, 2017.

[12] E. Cordelli, A. Vananti, T. Schildknecht, Analysis of laser ranges and angular measurements data fusion for space debris orbit determination, Adv. Space Res. 65 (1) (2020) 419–434, http://dx.doi.org/10.1016/j.asr.2019.11.009.

[13] L. Pasqualetto Cassinis, R. Fonod, E. Gill, Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft, Prog. Aerosp. Sci. 110 (2019) 100548, http://dx.doi.org/10.1016/j.paerosci.2019.05.008.

[14] M.R. Leinz, C.T. Chen, M.W. Beaven, T.P. Weismuller, D.L. Caballero, W.B. Gaumer, P.W. Sabasteanski, P.A. Scott, M.A. Lundgren, Orbital express autonomous rendezvous and capture sensor system (ARCSS) flight test results, in: R.T. Howard, P. Motaghedi (Eds.), Sensors and Systems for Space Applications II, in: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6958, 2008, p. 69580A, http://dx.doi.org/10.1117/12.779595.

[15] F. Castellini, D. Antal-Wokes, R.P. de Santayana, K. Vantournhout, Far approach optical navigation and comet photometry for the Rosetta mission, in: Proceedings of 25th International Symposium on Space Flight Dynamics, 25th ISSFD, 2015.

[16] J. Tao, Y. Cao, M. Ding, Z. Zhang, Visible and infrared image fusion-based image quality enhancement with applications to space debris on-orbit surveillance, Int. J. Aerosp. Eng. 2022 (1) (2022) 1–21, http://dx.doi.org/10.1155/2022/6300437.

[17] G.L. Civardi, M. Bechini, M. Quirino, A. Colombo, M. Piccinin, M. Lavagna, Generation of fused visible and thermal-infrared images for uncooperative spacecraft proximity navigation, Adv. Space Res. (2023) http://dx.doi.org/10.1016/j.asr.2023.03.022.

[18] S. Sharma, S. D'Amico, Neural network-based pose estimation for noncooperative spacecraft rendezvous, IEEE Trans. Aerosp. Electron. Syst. 56 (6) (2020) 4638–4658, http://dx.doi.org/10.1109/TAES.2020.2999148.

[19] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Satellite pose estimation challenge: Dataset, competition design, and results, IEEE Trans. Aerosp. Electron. Syst. 56 (5) (2020) 4083–4098, http://dx.doi.org/10.1109/TAES.2020.2989063.

[20] T.H. Park, M. Märtens, M. Jawaid, Z. Wang, B. Chen, T.-J. Chin, D. Izzo, S. D'Amico, Satellite pose estimation competition 2021: Results and analyses, Acta Astronaut. (2023) http://dx.doi.org/10.1016/j.actaastro.2023.01.002.

[21] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière, D. Aouada, A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects, Acta Astronaut. 212 (2023) 339–360, http://dx.doi.org/10.1016/j.actaastro.2023.08.001.

[22] D. Kaidanovic, AI-Aided Optical Navigation about Uncooperative Spacecraft Using Synthetic Imagery Algorithm for Non-Cooperative Spacecrafts (MSc thesis), Politecnico di Milano, 2022.

[23] S. Sharma, J. Ventura, S. D'Amico, Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous, AIAA J. Spacecr. Rocket. 55 (6) (2018) 1414–1429, http://dx.doi.org/10.2514/1.A34124.

[24] G. Gu, B.S. Ko, S.H. Go, S. Lee, J. Lee, M. Shin, Towards real-time and light-weight line segment detection, 2021, CoRR abs/2106.00186 arXiv:2106.00186.

[25] R.O. Duda, P.E. Hart, Use of the hough transformation to detect lines and curves in pictures, Commun. ACM 15 (1) (1972) 11–15, http://dx.doi.org/10.1145/361237.361242.

[26] P. Mittrapiyanumic, G. DeSouza, A. Kak, Calculating the 3d-pose of rigid-objects using active appearance models, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings, Vol. 5, ICRA '04. 2004, 2004, pp. 5147–5152, http://dx.doi.org/10.1109/ROBOT.2004.1302534.

[27] J.-F. Shi, S. Ulrich, S. Ruel, Spacecraft pose estimation using principal component analysis and a monocular camera, in: AIAA Guidance, Navigation, and Control Conference, 2017, pp. 1–24, http://dx.doi.org/10.2514/6.2017-1034.

[28] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, A model-based 3D template matching technique for pose acquisition of an uncooperative space object, Sensors 15 (3) (2015) 6360–6382, http://dx.doi.org/10.3390/s150306360.

[29] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, M. Grassi, Autonomous relative navigation around uncooperative spacecraft based on a single camera, Aerosp. Sci. Technol. 84 (2019) 1070–1080, http://dx.doi.org/10.1016/j.ast.2018.11.042.

[30] B.J. Naasz, R.D. Bums, S.Z. Queen, J. Van Eepoel, J. Hannah, E. Skelton, The HST SM4 relative navigation sensor system: Overview and preliminary testing results from the flight robotics lab, J. Astronaut. Sci. 57 (1–2) (2009) 457–483, http://dx.doi.org/10.1007/BF03321512.

[31] C. Harris, M. Stephens, et al., A combined corner and edge detector, in: Alvey Vision Conference, Vol. 15, 1988, pp. 10–5244.

[32] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: 2011 International Conference on Computer Vision, Vol. 1, 2011, pp. 2564–2571, http://dx.doi.org/10.1109/ICCV.2011.6126544.

[33] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision – ECCV 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 430–443.

[34] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary robust independent elementary features, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision – ECCV 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 778–792.

[35] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Comput. Vis. Image Underst. 110 (3) (2008) 346–359, http://dx.doi.org/10.1016/j.cviu.2007.09.014, Similarity Matching in Computer Vision and Multimedia.

[36] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110, http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[37] E. Karami, S. Prasad, M. Shehata, Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images, 2017, arXiv preprint arXiv:1710.02726.

[38] S.A.K. Tareen, Z. Saleem, A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK, in: 2018 International Conference on Computing, Mathematics and Engineering Technologies, ICoMET, 2018, pp. 1–10, http://dx.doi.org/10.1109/ICOMET.2018.8346440.

[39] D. Bojanić, K. Bartol, T. Pribanić, T. Petković, Y.D. Donoso, J.S. Mas, On the comparison of classic and deep keypoint detector and descriptor methods, in: 2019 11th International Symposium on Image and Signal Processing and Analysis, ISPA, 2019, pp. 64–69, http://dx.doi.org/10.1109/ISPA.2019.8868792.

[40] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6) (1986) 679–698, http://dx.doi.org/10.1109/TPAMI.1986.4767851.

[41] S. D'Amico, M. Benn, J.L. Jørgensen, Pose estimation of an uncooperative spacecraft from actual space imagery, Int. J. Space Sci. Eng. 2 (2) (2014) 171–189, http://dx.doi.org/10.1504/IJSPACESE.2014.060600.

[42] S. D'Amico, P. Bodin, M. Delpech, R. Noteborn, PRISMA, in: M. D'Errico (Ed.), Distributed Space Missions for Earth System Monitoring, Springer Science & Business Media, 2013, pp. 599–637.

[43] J.-F. Shi, S. Ulrich, S. Ruel, M. Anctil, Uncooperative spacecraft pose estimation using an infrared camera during proximity operations, in: AIAA SPACE 2015 Conference and Exposition, Vol. 1, 2015, pp. 1–17, http://dx.doi.org/10.2514/6.2015-4429.

[44] V. Capuano, S.R. Alimo, A.Q. Ho, S.-J. Chung, Robust features extraction for on-board monocular-based spacecraft pose acquisition, in: AIAA Scitech 2019 Forum, Vol. 1, 2019, pp. 1–15, http://dx.doi.org/10.2514/6.2019-2005.

[45] V. Lepetit, F. Moreno-Noguer, P. Fua, EPnP: An accurate O(n) solution to the PnP problem, Int. J. Comput. Vis. 81 (2) (2009) 155–166, http://dx.doi.org/10.1007/s11263-008-0152-6.

[46] R. Opromolla, C. Vela, A. Nocerino, C. Lombardi, Monocular-based pose estimation based on fiducial markers for space robotic capture operations in GEO, Remote Sens. 14 (18) (2022) http://dx.doi.org/10.3390/rs14184483.

[47] S. Sharma, S. D'Amico, Comparative assessment of techniques for initial pose estimation using monocular vision, Acta Astronaut. 123 (2016) 435–445, http://dx.doi.org/10.1016/j.actaastro.2015.12.032.

[48] J.E. Ball, D.T. Anderson, C.S. Chan, Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community, J. Appl. Remote Sens. 11 (4) (2017) 042609.

[49] L. Pasqualetto Cassinis, R. Fonod, E. Gill, I. Ahrns, J. Gil-Fernández, Evaluation of tightly- and loosely-coupled approaches in CNN-based pose estimation systems for uncooperative spacecraft, Acta Astronaut. 182 (2021) 189–202, http://dx.doi.org/10.1016/j.actaastro.2021.01.035.

[50] S. Mahendran, H. Ali, R. Vidal, 3D pose regression using convolutional neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2017, pp. 494–495, http://dx.doi.org/10.1109/CVPRW.2017.73.

[51] S. Sonawani, R. Alimo, R. Detry, D. Jeong, A. Hess, H.B. Amor, Assistive relative pose estimation for on-orbit assembly using convolutional neural networks, 2020, arXiv preprint arXiv:2001.10673.

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

[53] T.H. Park, S. D'Amico, Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap, Adv. Space Res. (2023) http://dx.doi.org/10.1016/j.asr.2023.03.036.

[54] S. Silvestrini, M. Lavagna, Deep learning and artificial neural networks for spacecraft dynamics, navigation and control, Drones 6 (10) (2022) http://dx.doi.org/10.3390/drones6100270.

[55] Z. Zhao, G. Peng, H. Wang, H.-S. Fang, C. Li, C. Lu, Estimating 6d pose from localizing designated surface keypoints, 2018, arXiv preprint arXiv:1812.01387.

[56] S. Peng, Y. Liu, Q. Huang, X. Zhou, H. Bao, Pvnet: Pixel-wise voting network for 6dof pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4561–4570.

[57] T.H. Park, S. Sharma, S. D'Amico, Towards robust learning-based pose estimation of noncooperative spacecraft, 2019, arXiv preprint arXiv:1909.00392.

[58] M. Piazza, M. Maestrini, P. Di Lizia, Monocular relative pose estimation pipeline for uncooperative resident space objects, J. Aerosp. Inf. Syst. 19 (9) (2022) 613–632, http://dx.doi.org/10.2514/1.I011064.

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

[60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

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

[62] P. Fracchiolla, Analysis and Validation of a Vision-Based Pose Initialization Algorithm for Non-Cooperative Spacecrafts (MSc thesis), Università degli Studi di Padova, 2019.

[63] M. Bechini, G.L. Civardi, M. Quirino, A. Colombo, M. Lavagna, Robust monocular pose initialization via visual and thermal image fusion, in: 73rd International Astronautical Congress, IAC 2022, International Astronautical Federation, IAF, Paris, France, 2022, pp. 1–15, https://hdl.handle.net/11311/1221785.

[64] J. Shi, Tomasi, Good features to track, in: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600, http://dx.doi.org/10.1109/CVPR.1994.323794.

[65] J. Matas, C. Galambos, J. Kittler, Robust detection of lines using the progressive probabilistic hough transform, Comput. Vis. Image Underst. 78 (1) (2000) 119–137, http://dx.doi.org/10.1006/cviu.1999.0831.

[66] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, LSD: A fast line segment detector with a false detection control, IEEE Trans. Pattern Anal. Mach. Intell. 32 (4) (2010) 722–732, http://dx.doi.org/10.1109/TPAMI.2008.300.

[67] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, G. Furano, High-performance embedded computing in space: Evaluation of platforms for vision-based navigation, J. Aerosp. Inf. Syst. 15 (4) (2018) 178–192, http://dx.doi.org/10.2514/1.I010555.

[68] A. Tahir, H.S. Munawar, J. Akram, M. Adil, S. Ali, A.Z. Kouzani, M.A.P. Mahmud, Automatic target detection from satellite imagery using machine learning, Sensors 22 (3) (2022) http://dx.doi.org/10.3390/s22031147.

[69] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[70] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[71] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).

[72] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, CVPR, 2016, pp. 779–788, http://dx.doi.org/10.1109/CVPR.2016.91.

[73] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, X. Chu, YOLOv6 v3. 0: A full-scale reloading, 2023, arXiv preprint arXiv:2301.05586.

[74] A. Khalfaoui, A. Badri, I.E. Mourabit, Comparative study of YOLOv3 and YOLOv5's performances for real-time person detection, in: 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology, Vol. 1, IRASET, 2022, pp. 1–5, http://dx.doi.org/10.1109/IRASET52964.2022.9737924.

[75] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot MultiBox detector, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 21–37, http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[76] I. Suárez, J.M. Buenaposada, L. Baumela, ELSED: Enhanced Line SEgment Drawing, Pattern Recognit. 127 (2022) 108619, http://dx.doi.org/10.1016/j.patcog.2022.108619.

[77] Z. Xu, B.-S. Shin, R. Klette, Accurate and robust line segment extraction using minimum entropy with hough transform, IEEE Trans. Image Process. 24 (3) (2015) 813–822, http://dx.doi.org/10.1109/TIP.2014.2387020.

[78] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, LSD: A fast line segment detector with a false detection control, IEEE Trans. Pattern Anal. Mach. Intell. 32 (4) (2010) 722–732, http://dx.doi.org/10.1109/TPAMI.2008.300.

[79] C. Akinlar, C. Topal, Edlines: Real-time line segment detection by edge drawing (ED), in: 2011 18th IEEE International Conference on Image Processing, Vol. 1, 2011, pp. 2837–2840, http://dx.doi.org/10.1109/ICIP.2011.6116138.

[80] X. Lu, J. Yao, K. Li, L. Li, CannyLines: A parameter-free line segment detector, in: 2015 IEEE International Conference on Image Processing, Vol. 1, ICIP, 2015, pp. 507–511, http://dx.doi.org/10.1109/ICIP.2015.7350850.

[81] N.-G. Cho, A. Yuille, S.-W. Lee, A novel linelet-based representation for line segment detection, IEEE Trans. Pattern Anal. Mach. Intell. 40 (5) (2018) 1195–1208, http://dx.doi.org/10.1109/TPAMI.2017.2703841.

[82] S. Xie, Z. Tu, Holistically-nested edge detection, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 1395–1403, http://dx.doi.org/10.1109/ICCV.2015.164.

[83] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, J. Tang, Richer convolutional features for edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 1939–1946, http://dx.doi.org/10.1109/TPAMI.2018.2878849.

[84] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, M. Jagersand, BASNet: Boundary-aware salient object detection, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 7471–7481, http://dx.doi.org/10.1109/CVPR.2019.00766.

[85] K. Huang, Y. Wang, Z. Zhou, T. Ding, S. Gao, Y. Ma, Learning to parse wireframes in images of man-made environments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 626–635.

[86] Z. Zhang, Z. Li, N. Bi, J. Zheng, J. Wang, K. Huang, W. Luo, Y. Xu, S. Gao, Ppgnet: Learning point-pair graph for line segment detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7105–7114.

[87] Y. Zhou, H. Qi, Y. Ma, End-to-end wireframe parsing, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 962–971.
[88] N. Xue, S. Bai, F. Wang, G.-S. Xia, T. Wu, L. Zhang, Learning attraction field representation for robust line segment detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1595–1603.
[89] N. Xue, T. Wu, S. Bai, F. Wang, G.-S. Xia, L. Zhang, P.H. Torr, Holistically-attracted wireframe parsing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2788–2797.
[90] S. Huang, F. Qin, P. Xiong, N. Ding, Y. He, X. Liu, TP-LSD: tri-points based line segment detector, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII, Springer, 2020, pp. 770–785.
[91] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Spacecraft Pose Estimation Dataset (SPEED), Zenodo, 2019, http://dx.doi.org/10.5281/zenodo.6327546.
[92] T.H. Park, J. Bosse, S. D'Amico, Robotic testbed for rendezvous and optical navigation: Multi-source calibration and machine learning use cases, 2021, arXiv preprint arXiv:2108.05529.
[93] P. Proenca, URSO: Unreal Rendered Spacecrafts On-Orbit Datasets, Zenodo, 2019, http://dx.doi.org/10.5281/zenodo.3279632.
[94] T.H. Park, M. Märtens, G. Lecuyer, D. Izzo, S. D'Amico, Next Generation Spacecraft Pose Estimation Dataset (SPEED+), Zenodo, 2021, http://dx.doi.org/10.25740/wv398fc4383.
[95] M. Bechini, M. Lavagna, P. Lunghi, Dataset generation and validation for spacecraft pose estimation via monocular images processing, Acta Astronaut. 204 (2023) 358–369, http://dx.doi.org/10.1016/j.actaastro.2023.01.012.
[96] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Dataset for Monocular Pose Estimation, Zenodo, 2022, http://dx.doi.org/10.5281/zenodo.6499008.
[97] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Dataset for Region of Interest Estimation and Semantic Segmentation, Zenodo, 2022, http://dx.doi.org/10.5281/zenodo.6507863.
[98] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Wireframe Dataset Model for Line Segments Detection, Zenodo, 2022, http://dx.doi.org/10.5281/zenodo.6372848.
[99] H.A. Dung, B. Chen, T.-J. Chin, A spacecraft dataset for detection, segmentation and parts recognition, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2021, pp. 2012–2019, http://dx.doi.org/10.1109/CVPRW53098.2021.00229.
[100] Y. Hu, S. Speierer, W. Jakob, P. Fua, M. Salzmann, Wide-depth-range 6D object pose estimation in space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 15870–15879.
[101] A. Price, K. Yoshida, A monocular pose estimation case study: The Hayabusa2 minerva-II2 deployment, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2021, pp. 1992–2001, http://dx.doi.org/10.1109/CVPRW53098.2021.00227.
[102] M.A. Musallam, V. Gaudilliere, E. Ghorbel, K.A. Ismaeil, M.D. Perez, M. Poucet, D. Aouada, Spacecraft recognition leveraging knowledge of space environment: Simulator, dataset, competition design and analysis, in: 2021 IEEE International Conference on Image Processing Challenges, ICIPC, 2021, pp. 11–15, http://dx.doi.org/10.1109/ICIPC53495.2021.9620184.
[103] L.P. Cassinis, R. Fonod, E. Gill, I. Ahrns, J.G. Fernandez, CNN-based pose estimation system for close-proximity operations around uncooperative spacecraft, in: AIAA Scitech 2020 Forum, 2020, pp. 1–16, http://dx.doi.org/10.2514/6.2020-1457.
[104] K. Black, S. Shankar, D. Fonseka, J. Deutsch, A. Dhir, M.R. Akella, Real-time, flight-ready, non-cooperative spacecraft pose estimation using monocular imagery, 2021, arXiv preprint arXiv:2101.09553.
[105] M. Piccinin, S. Silvestrini, G. Zanotti, A. Brandonisio, P. Lunghi, M. Lavagna, ARGOS: calibrated facility for image based relative navigation technologies on ground verification and testing, in: 72nd International Astronautical Congress, IAC 2021, International Astronautical Federation, IAF, Dubai, United Arab Emirates, 2021, pp. 1–12.
[106] T. Möller, B. Trumbore, Fast, minimum storage ray-triangle intersection, J. Graph. Tools 2 (1) (1997) 21–28, http://dx.doi.org/10.1080/10867651.1997.10487468.
[107] F. Cuzzocrea, Analysis and Validation of Spaceborne Synthetic Imagery Using a Vision-Based Pose Initialization Algorithm for Non-Cooperative Spacecrafts (MSc thesis), Politecnico di Milano, 2020.
[108] Y. Hu, J. Hugonot, P. Fua, M. Salzmann, Segmentation-driven 6d object pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3385–3394.