

Research Paper

IRoLSD: Illumination Robust line segment detection for spacecraft relative navigation

Iason Georgios Velentzas^{ID*}, Panagiotis Tsiotras^{ID}

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, United States of America

ARTICLE INFO

Keywords:

Computer vision
Deep learning
Spacecraft navigation
Line segment detection

ABSTRACT

Line segments are used extensively in computer vision tasks, since they capture robust geometric features from structural information in the scene. In spacecraft relative navigation applications, images often contain prominent geometric structures, rendering line features valuable for downstream tasks such as localization and mapping. However, the performance of both learning-based and traditional line detection algorithms is significantly degraded under the harsh illumination conditions encountered in orbit. In this work, we introduce a novel line detection framework designed to robustly detect line segments in spacecraft imagery. Using images with perturbed lighting conditions and by incorporating depth maps to reconnect fragmented lines, we propose IRoLSD, an algorithm that reliably detects line segments across diverse lighting environments and realistic trajectories. We extensively evaluated the performance of IRoLSD in a series of experiments, proving that the proposed training method outperforms the state-of-the-art line detection algorithms in the especially challenging space domain. We also demonstrate the generalizability of IRoLSD to achieve superior performance across synthetic and real-world images, as well as across different spacecraft models.

1. Introduction

Increasing the robustness and autonomy of Rendezvous and Proximity Operations (RPOs) is critical to ensure safe and efficient spacecraft operation. Over the past several years, missions related to On-Orbit Servicing (OOS) and Active Debris Removal (ADR) have become essential to support sustainable space exploration. Notable examples include e.Deorbit and PROBA-3 by ESA [1], PRISMA by OHB Sweden [2], OAAN [3], and Restore-L by NASA [4]. Since 2020, the number of satellites in orbit has increased sharply with a continuous annual growth. Based on the European Space Agency (ESA) report [5], there has been an increase of approximately 19% in the total number of objects orbiting the Earth during 2024. Advanced Guidance, Navigation, and Control (GNC) capabilities allow for more precise maneuvers with minimal human-in-the-loop requirements.

Vision-based navigation with monocular cameras has emerged as a promising solution. Cameras have already been deployed in missions to demonstrate target detection and formation flying capabilities [1,6]. The rich and compact information they provide in a wide variety of operating ranges, and with low-cost, mass, and power mission requirements, render them an appealing sensor to estimate and track the relative pose between the camera and the typically uncooperative chaser spacecraft. In this context, autonomy means that the servicing spacecraft is able to detect and track the target's relative pose without

human-in-the-loop intervention or reliance on prior knowledge about the target.

Estimating the relative pose of the target object is highly dependent on the robust detection and tracking of prominent visual features. These features play a key role in shape reconstruction and pose estimation, which are essential for safe and effective approach maneuver design. However, spacecraft visual navigation remains a challenging problem due to the unique and harsh illumination conditions encountered in orbit. Unlike terrestrial environments, space lacks atmospheric diffusion and the sole sources of light are direct sunlight and the Earth's albedo. This results in an increasing effect of specular lighting or self-shadowing, which does not allow for a consistent visual appearance of the target. Self-shadowing and occlusions become particularly significant when orbiting around spacecraft with complex structures, hindering the visibility of the features. As a relative trajectory evolves and the lighting conditions change, it is typical that the same features will appear under different pixel intensities, disrupting their local appearance and hindering reliable frame-to-frame association. Previous studies show that point-based visual features are often unreliable in such conditions, leading to frequent loss of tracking [7].

To address the issue of saliency and robustness of visual spacecraft features, we propose to utilize line features that enforce stronger

* Corresponding author.

E-mail address: iason.velentzas@gatech.edu (I.G. Velentzas).

geometric constraints both for feature detection and for landmark optimization. Our previous work [8] demonstrated that line features exhibit more reliable tracking than points and can contribute to the reduction of localization errors. However, line detection algorithms are still predicated on the local appearance of the images and thus are heavily affected by illumination conditions [9].

The main contribution of this paper is a hybrid deep Convolutional Neural Network (CNN) line detector for spacecraft navigation. At the core of our approach is the introduction of **IRoLSD** (Illumination-Robust Line Segment Detector), a novel framework designed to address the unique visual challenges posed by the space environment. Our framework builds upon the DeepLSD architecture [10], which serves as the foundation for IRoLSD. First, we introduce the concept of identifying illumination-robust line segments, which enables consistent and reliable detection of salient features throughout the spacecraft's trajectory, despite drastic changes in lighting conditions. Second, we incorporate depth-assisted line detection, which helps connect fragmented line segments, a common issue in space imagery due to occlusions and self-shadowing caused by complex target geometries. Third, we integrate these techniques into a new supervision strategy that enhances our baseline detector's ability to recognize low-contrast line segments in poorly illuminated regions in the image and ensures repeatable detection across images with significant appearance variation.

This work addresses the problem of unreliable line feature detection in orbit and presents a robust line segment detector that can be used in GNC pipelines for target reconstruction and relative navigation purposes. In addition, we investigate the performance of state-of-the-art line detection techniques in various synthetic and laboratory-generated spacecraft datasets and evaluates the capability of using these line segments to track visual features. Through extensive experiments, we show that our method outperforms existing approaches, both traditional and learning-based — especially under the harsh and variable lighting conditions characteristic of orbital servicing scenarios. We demonstrate that by supervising our network to detect Illumination Robust line segments, IRoLSD can consistently, and more accurately, detect line segments for previously unseen spacecraft models through trajectories with harsh illumination conditions. We finally show that the detected line segments can be effectively used by line matchers to complete the pose estimation task.

The rest of the paper is organized as follows: Section 2 provides the necessary mathematical background of line segment geometry; Section 3 presents current line segment detectors and their application to spacecraft images; Section 4 describes the proposed network architecture and the proposed supervision method; and Section 5 details the experimental procedures and datasets and presents the results. Finally, Section 6 summarizes our key findings and conclusions.

2. Background

In the following subsections, we formulate the 2D line segment distance metrics that will be used to evaluate the performance of line segment detection (Section 2.1). We also describe the processes used for line segment 2D–2D tracking (Section 2.2) and 3D–2D reconstruction and projection (Section 2.3). Finally, Section 2.4 explains the approach to estimate the relative pose using 3D–2D line correspondences.

2.1. 2D line segment: Notation and distances

Let a 2D line segment ℓ be represented as a pair of endpoints $\ell = (p_1, p_2)$, where $p_1, p_2 \in \Omega$ are the pixel coordinates in the image domain $\Omega \subset \mathbb{R}^2$. The length of the line segment is the Euclidean norm of the vector of its endpoints $|\ell| = \|p_2 - p_1\|_2$ and the unit direction of the line segment is defined as $\hat{d}_\ell = (p_2 - p_1) / |\ell|$. The heading of the line segment $\theta_\ell \in [0, \pi)$ is defined as the angle between the direction vector $\hat{d}_\ell$ and the horizontal axis x of the image. Each line segment is associated with its infinite extension, namely a 2D line. A 2D line can

be represented as $L = (a, b, c) \in \mathbb{R}^3$ and is defined as $L = \{(x, y) \in \mathbb{R}^2 \mid ax + by + c = 0\}$. The parameters (a, b, c) of the infinite line can be extracted from the pixel coordinates of the endpoints of the line segment. If we denote the endpoints of the line segment as $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2) \in \mathbb{R}^2$, then the parameters of the associated infinite line are $L = (a, b, c) = (y_1 - y_2, x_1 - x_2, -ax_1 - by_1)$.

Now, let $\ell, \ell' \in \Omega$ be two line segments with endpoints (p_1, p_2) and (p'_1, p'_2) , respectively, and let L, L' be the infinite lines associated with each line segment. We can define different distance metrics between the line segments, each of which focuses on different aspects of the line segment.

The **Structural Distance** metric between two line segments penalizes the deviation of the position of the endpoints. It is defined as the sum of the Euclidean distance between the corresponding endpoints

$$d_S(\ell, \ell') = \min \left(\|p_1 - p'_1\|_2 + \|p_2 - p'_2\|_2, \|p_1 - p'_2\|_2 + \|p_2 - p'_1\|_2 \right). \quad (1)$$

The structural distance metric is symmetric between the two line segments, that is, $d_S(\ell, \ell') = d_S(\ell', \ell)$, and is zero when the two line segments exactly overlap.

The **Orthogonal Distance** metric between two line segments allows for greater variability in the position of the endpoints, by considering the distance of the endpoints of one line segment to the infinite line extension of the second line segment. The shortest distance from a point $p = (x_0, y_0)$ to a line $L = (a, b, c)$ is defined as

$$d_p(p, L) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}. \quad (2)$$

Considering both endpoints of each line segment and averaging their shortest distance, we can define the distance of a line segment $\ell = (p_1, p_2)$ to a line L' as follows

$$d_a(\ell, L') = \frac{d_p(p_1, L') + d_p(p_2, L')}{2}. \quad (3)$$

The orthogonal distance metric is non-symmetric with respect to the two line segments. Hence, for the orthogonal distance of a pair of line segments ℓ, ℓ' with associated infinite lines L, L' , we use

$$d_O(\ell, \ell') = \frac{d_a(\ell, L') + d_a(\ell', L)}{2}. \quad (4)$$

The **Angle Distance** metric between two line segments completely discards the endpoint positions and only considers the angle alignment between the two line segments. We do not utilize this metric to assess whether two line segments are close to each other in the image, but rather to assess how good the alignment is for a pair of line segments that are already deemed to be close to each other. We define the angle distance between two line segments $\ell = (\hat{d}_\ell, \phi_\ell)$ and $\ell' = (\hat{d}_{\ell'}, \phi_{\ell'})$ as the acute angle between the two

$$d_\phi(\ell_1, \ell_2) = \begin{cases} |\phi_\ell - \phi_{\ell'}|, & \text{if } \phi_\ell - \phi_{\ell'} \leq \pi/2, \\ \pi - |\phi_\ell - \phi_{\ell'}|, & \text{if } \phi_\ell - \phi_{\ell'} > \pi/2. \end{cases} \quad (5)$$

The structural and orthogonal distance metrics are graphically shown in Fig. 1.

2.2. Line segment matching

Tracking detected line features across multiple images is crucial for spacecraft navigation, as it enables to establish correspondences between the line segments that are used for tasks such as 3D mapping and relative pose estimation. Let $I : \Omega \rightarrow \mathbb{R}$ and $I' : \Omega \rightarrow \mathbb{R}$ be two images with the same pixel domain $\Omega \subset \mathbb{R}^2$. We associate a detected line segment $\ell = (p_1, p_2)$, where $p_1, p_2 \in \Omega$ with a d -dimensional descriptor vector $\mathbf{d} \in \mathbb{R}^d$, designed to be distinctive and ideally invariant to minor changes in viewpoint or illumination. These descriptors can be handcrafted, such as LBD [11], which typically create a compact representation of the intensity of the pixels surrounding the line segment, or it can be learned, such as SOLD2 [12]. The goal is

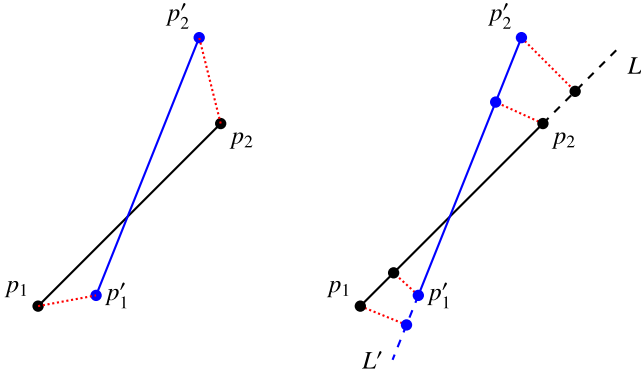


Fig. 1. The structural distance between the endpoints is depicted on the left. On the right, the projections of each endpoint create the orthogonal distances.

to find, for each line segment in I , the most similar line segment in I' based on a descriptor distance metric.

To ensure robust matching, we adopt the Mutual Nearest Neighbor (MNN) [13], which dictates that a line segment ℓ detected in image I with a descriptor vector $\mathbf{d} \in \mathbb{R}^d$ is associated with a line segment ℓ' detected in image I' with a descriptor vector $\mathbf{d}' \in \mathbb{R}^d$, if and only if each one is the nearest neighbor of the other in the descriptor space. To this end, let $r(\cdot, \cdot)$ denote a distance metric, typically the Euclidean distance, or the Hamming distance for binary descriptors, and let $\mathcal{L}(I), \mathcal{L}(I')$ be the sets of line segments in each image. Then, the set of corresponding line segments between images I, I' is

$$\begin{aligned} M(I, I') &:= \{(\ell, \ell') \mid r(d_\ell, d_{\ell'}) < \min_{k \neq \ell} r(d_\ell, d_{\ell'_k}), \quad k, \ell \in \mathcal{L}(I)\} \\ &\cap \{(\ell, \ell') \mid r(d_{\ell'}, d_{\ell'}) < \min_{k' \neq \ell'} r(d_{\ell'}, d_{\ell'_k}), \quad k', \ell' \in \mathcal{L}(I')\}. \end{aligned} \quad (6)$$

For spacecraft navigation it is crucial to avoid creating erroneous matches, especially in the presence of viewpoint and lighting changes. Hence, we employ MNN to match line segments between two images, since MNN provides a robust and symmetric way to establish reliable correspondences.

Tracking line segments throughout sequences of images can be done by applying the MNN concept to each of the sequence pairs. For a sequence of images $(I_1, I_2, \dots, I_N)$, we define the set of tracked line segments as

$$\begin{aligned} \mathcal{T}(I_1, I_2, \dots, I_N) &:= \{(\ell_1, \ell_2, \dots, \ell_M) \mid (\ell_i, \ell_j) \in M(I_i, I_j), \\ &\quad i, j = 1, 2, \dots, N, \quad i \neq j\}. \end{aligned} \quad (7)$$

2.3. 3D line segment reconstruction

In SLAM pipelines, line feature reconstruction in the 3D space is essential for establishing correspondences between 2D image observations and an existing 3D map. This process typically follows the estimation of the relative poses of the camera, enabling accurate localization within the environment. In a SLAM setting, matched line segments between multiple views are triangulated into 3D line primitives using the estimated camera poses. These 3D lines are then used to update the existing map or initialize new map elements. During optimization, only infinite lines associated with the line segments are used, as the endpoints of the 2D line segments are known to be unstable and unreliable [14].

In this work, we employ 3D line reconstruction for evaluation purposes. Matches of 2D line segments are triangulated in the 3D space to yield infinite 3D lines. To evaluate detection accuracy, the reconstructed 3D lines are re-projected onto additional image frames and compared with the corresponding 2D line detections. The comparison is performed in the image plane using distance-based metrics, allowing

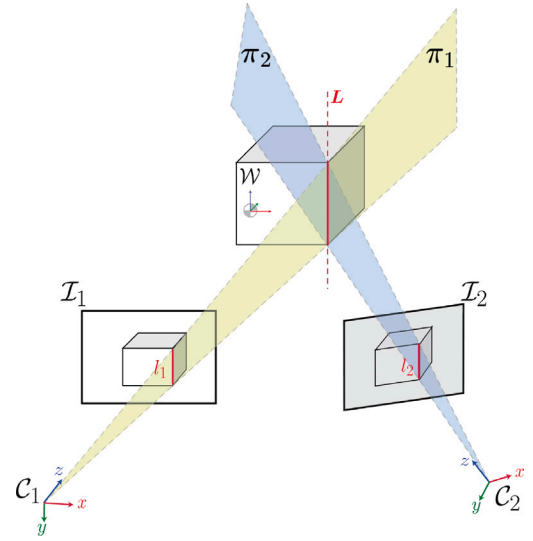


Fig. 2. 3D Line triangulation from 2D line segment correspondence.

for a consistent evaluation across both 2D homographic warpings and 3D homogeneous transformations in realistic trajectory sequences.

Let $\mathcal{W}$ denote the world frame with origin W , which is arbitrarily attached to the target, and let C_1, C_2, C_3 denote the camera frames at three different time steps with origins C_1, C_2, C_3 , respectively. Let $T_{C_i\mathcal{W}}$ be the relative pose of $\mathcal{W}$ with respect to C_i :

$$T_{C_i\mathcal{W}} = \begin{bmatrix} \mathbf{R}_{C_i\mathcal{W}} & \mathbf{r}_{\mathcal{W}C_i}^{C_i} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad i = 1, 2, 3, \quad (8)$$

where $\mathbf{R}_{C_i\mathcal{W}} \in \text{SO}(3)$ and $\mathbf{r}_{\mathcal{W}C_i}^{C_i} \in \mathbb{R}^3$. Let the camera calibration matrix be known and equal to a standard pinhole camera model [15]

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

where $f_x, f_y \in \mathbb{R}$ are the focal lengths in the respective camera frame axes and $c_x, c_y \in \mathbb{R}$ constitute the pixel coordinates of the principal point of the camera. For simplicity, we assume that the camera intrinsic parameters remain the same throughout the trajectory.

Suppose now that we have identified a 2D line segment ℓ_i in the image captured by camera C_i . The plane π_i containing the origin of the camera frame C_i and the two endpoints of the line segment ℓ_i can be expressed mathematically as

$$\pi_i = \ell_i^\top K \begin{bmatrix} \mathbf{R}_{C_i\mathcal{W}} & \mathbf{r}_{\mathcal{W}C_i}^{C_i} \end{bmatrix} \in \mathbb{R}^{1 \times 4}, \quad (10)$$

where $|$ denotes horizontal concatenation of matrices.

In the case we have identified a 2D line segment correspondence ℓ_1, ℓ_2 between two images, we can recover the associated 3D line L expressed in the frame $\mathcal{W}$. Specifically, the Plücker coordinates representation of the 3D line $L = [\mathbf{m}^\top, \mathbf{d}^\top]^\top \in \mathbb{R}^6$, are defined algebraically from intersecting the planes as follows

$$\begin{bmatrix} [\mathbf{d}]_\times & \mathbf{m} \\ -\mathbf{m}^\top & 0 \end{bmatrix} = \pi_1 \pi_2^\top - \pi_2 \pi_1^\top, \quad (11)$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix of a vector. This process is depicted graphically in Fig. 2.

2.4. Pose estimation from lines

In a SLAM pipeline, the recovered 3D line is used to populate the 3D map of the environment and for optimization purposes to estimate the relative pose and refine the 3D map. Assuming that past

2D observations are associated with 3D reconstructed lines, the Line Segment Tracking Process from Section 2.2 also yields correspondences of 3D reconstructed lines to 2D detected line segments.

These correspondences are typically used in an optimization framework, which jointly optimizes the 3D line position and the relative pose between the Target and the Camera. In this work, we utilize a pose optimization scheme to estimate the relative pose, which is the final quantity of interest in the relative navigation problem, assuming that the 3D map is well constructed.

The projection of the 3D infinite line in Plücker coordinates $\mathbf{L} = [\mathbf{m}^\top, \mathbf{d}^\top]^\top$ can be calculated using the camera line projection matrix

$$K_L = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & -f_x f_y \end{bmatrix}, \quad (12)$$

and the camera extrinsic parameters, i.e., the relative pose of the camera to the target T_{CW}

$$\mathbf{L} = K_L \begin{bmatrix} R_{CW} & [\mathbf{r}_{WC}^C]_\times R_{CW} \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \mathbf{d} \end{bmatrix}. \quad (13)$$

The metric we wish to minimize is the error between the detected 2D line segments and the projection of the associated 3D lines into the image plane. Since this is an error between an infinite 2D line and a 2D line segment, we employ the orthogonal distance between the line segment endpoints and the projected infinite line. Then, for N detected 2D line segments ℓ_i ($i = 1, \dots, N$) that are associated with $\mathbf{L}_i = [\mathbf{m}_i^\top, \mathbf{d}_i^\top]^\top$ infinite lines in the 3D space, the optimization problem can be formulated using Eq. (3) as follows

$$\min_{R_{CW}, \mathbf{r}_{WC}^C} \sum_{i=1}^N d_a(\ell_i, K_L \begin{bmatrix} R_{CW} & [\mathbf{r}_{WC}^C]_\times R_{CW} \end{bmatrix} \begin{bmatrix} \mathbf{m}_i \\ \mathbf{d}_i \end{bmatrix}). \quad (14)$$

More details on the optimization parameters in the orthonormal representation, together with the derivation of the Jacobian of the error with respect to the pose, can be found in [8]. Finally, after the pose is estimated, we can calculate the pose estimation error in terms of translation and orientation separately.

Given the ground truth relative translation $\mathbf{r}_{WC}^C$ and the estimated one $\hat{\mathbf{r}}_{WC}^C$, we define the relative translation error as the Euclidean distance in the 3D space as follows

$$e_r = \|\mathbf{r}_{WC}^C - \hat{\mathbf{r}}_{WC}^C\|_2. \quad (15)$$

Given the ground truth relative rotation between the $\mathcal{W}, C$ frames, R_{CW} and the estimated rotation matrix $\hat{R}_{CW}$, we can calculate the orientation error as the magnitude of the rotation that aligns the two frames. We use the quaternion representation of the rotation matrices to calculate this error

$$e_q = 2 \arccos \langle q(R_{CW}), q(\hat{R}_{CW}) \rangle, \quad (16)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product operation of two unit quaternions [16].

3. Related work

3.1. Handcrafted line detectors

Traditional line segment detection methods typically rely on the image gradient to detect high-contrast pixels that are identified as edges. LSD [17] uses the gradient orientation of the pixels to gradually group them into line support regions. The line support regions are approximated by a rectangle, which is iteratively refined to detect the final line segment. EDLines [18] uses an edge drawing algorithm and subsequently connects edge pixels into line segments, only expanding the line segments in the orthogonal direction to the gradient. ELSed [19] builds on the edge drawing technique of EDLines by introducing an enhanced routing algorithm that connects gradient-aligned pixels, while introducing pixel jumps to connect small discontinuities

at the pixel level. AG3line [20] introduces an active grouping strategy by selecting the next pixel using line geometry constraints and reducing line fragmentation. This technique promises more complete line segment extractions using both geometry and gradient alignment, which provides more information than the pixel-level local gradients. MCMLSD uses a different approach [21], which first detects lines and then refines the location of the segment using dynamic programming on a Markov chain model. MCMLSD achieves better localization of line segments by enforcing global lines, but also requires a significant computation overhead.

Handcrafted line segment detectors usually exhibit accurate line detection when the line information can be clearly represented by the image gradient. However, these techniques are highly sensitive to changes in illumination and noise. Since they depend solely on pixel-level cues and specific geometry constraints, they lack the high-level scene understanding to recover line information in dynamic lighting environments. They are also limited by their reliance on manual parameter tuning and lack adaptability to diverse environments.

3.2. Learned line detectors

A variety of deep learning-based methods have been trained to improve the robustness of line detection by using data-driven feature extraction. The representation of line segments is a critical aspect in training, as it influences the network's ability to learn spatial continuity, feature orientation, geometric coherence, and high-level contextual information.

Some approaches tackle the problem of line segment detection by focusing solely on structural lines, with the aim of recovering the underlying wireframe representation of the scene. To this end, HAWP [22], L-CNN [23], and PPGNet [24] detect line segments and junction points in parallel. HAWP introduces a parameterization normalization technique to represent the line segments in the image as a 4-D Attraction Field Map [25], facilitating precise line localization. L-CNN incorporates penalties for incorrect line connectivities and overlapping segments, leading to a clearer and more coherent wireframe output. PPGNet employs a graph neural network, where junctions serve as nodes and line segments as edges, learning the scene's wireframe by modeling its underlying graph structure.

In contrast to wireframe-focused methods, other approaches learn to detect line segments via their endpoints, without explicitly modeling the scene's layout. TP-LSD [26] formulates line segment detection as an end-to-end keypoint regression problem, representing each line segment using a root point and two displacement vectors to its endpoints. In parallel to the triple point extraction branch, which predicts a heatmap of the endpoints and root points, a structural guidance branch uses this heatmap as prior information to fit line segments. M-LSD [27] introduces a real-time line segment detection network with very few parameters and a low inference time. The network regresses multiple feature maps by overparameterizing line segments and utilizes SoL augmentation, which splits a line segment into multiple sub-parts to provide auxiliary data during training. SOLD2 [12] is another endpoint-based detector that jointly performs the detection and description of line segments. To improve robustness against viewpoint changes and occlusions, SOLD2 represents each line segment as a sequence of point descriptors sampled along its length. This joint detection–description strategy captures local appearance information, facilitating feature matching; however, it may lead to degraded performance under harsh and varying illumination conditions. Finally, LETR [28] is a transformer-based architecture that detects line segments without the need for edge or junction features. LETR detects and refines line segments directly through a multiscale encoder–decoder pipeline using a direct endpoint distance loss; LETR offers robustness against occlusions, being able to capture geometric information and high-level scene understanding.

A common limitation of all endpoint-based approaches is their susceptibility to endpoint instability, which is the tendency of predicted endpoints to vary significantly in the presence of occlusions, self-shadowing, or line fragmentation. This instability poses challenges for consistent detection, particularly in spaceborne visual navigation scenarios.

DeepLSD [10] introduces a hybrid line detection scheme that integrates a learned attraction field representation with a handcrafted pseudo-ground truth generation process and a handcrafted final line segment extraction. The method is self-supervised by bootstrapping in the homographic adaptations of LSD detections. The bootstrapping process produces line segments that form the Attraction Field Map (AFM) of the image, which is the partition of the image pixels into regions based on their distance to the closest detected line segment. By encoding the line segment information in an intermediate AFM representation, the network is able to learn more global structures and interconnections between line segments, resulting in cleaner and more coherent representations. Finally, the learned AFM is converted into a surrogate image gradient, and a handcrafted line detector is employed to extract line segments with pixel-level accuracy, preserving the precision of traditional methods while benefiting from learned representations.

3.3. Line detectors for spacecraft navigation

To the best of our knowledge, there have only been a few prior works investigating the application of line segment detection in spacecraft imagery [29,30]. The first two works used line detection to organize line segments and keypoints into higher-level geometric primitives. These combined features were then used to establish 3D-to-2D correspondences with a known 3D model or point cloud, facilitating accurate pose estimation relative to a target spacecraft. The third work has focused on enhancing the images of spacecraft to achieve better line feature extraction.

In [29], Sharma et al. detect line segments using two separate processing streams, both of which rely on the Hough transform. In the first stream, the detection is applied after Weak Gradient Elimination (WGE) to detect shorter lines, whereas in the second stream it is applied to the edges of the image to detect longer lines. To address the issue of manual parameter tuning, the authors propose an adaptive strategy that adjusts the Hough transform parameters based on the size of the detected spacecraft bounding box.

The work of [30] adopted a similar framework but replaced the handcrafted detector with MLSD for line segment detection. Initially, the authors manually selected keypoints in images from the TANGO spacecraft with known relative pose. These keypoints are reconstructed in 3D space and then projected back into the training images to provide supervision. As a result, the network effectively learns to detect the manually annotated line segments associated with the 3D point cloud. A key limitation of this approach is the assumption of perfect pose knowledge of the training images to reconstruct the 3D point cloud and back-project it, as well as its applicability being restricted to the specific spacecraft used during training.

More recently, ST_LSD was proposed to deal with line segment detection specifically for space and complex illumination environments [31]. It involves a handcrafted line extraction method that combines adaptive bilateral filtering with an improved two-dimensional Otsu thresholding process for robust anchor point extraction [32]. The authors of [31] also incorporate a line segment aggregation strategy based on the structural similarity of the line segments. ST_LSD is evaluated on a custom semi-physical dataset of a 3U cubesat that contains seven different spacecraft poses and three different brightness scenarios. The lack of extensive evaluation raises generalizability and scalability concerns in terms of the robustness and speed achieved.

Moreover, as reported in [31], the efficacy of the algorithm degrades significantly in low-illumination scenarios where the contrast

between the target and the background becomes minimal. The anchor extraction strategy struggles to separate the spacecraft from the background, leading to incomplete or missing detections. Finally, the line segment aggregation strategy is predicated on manually selected thresholds that would need to be adapted for different resolution images or different spacecraft-to-image area ratios.

4. Methodology

In visual spacecraft navigation, the robustness of the feature detection step is essential to avoid loss of tracking. Our goal is not to detect as many line segments as possible in the image, but rather to ensure that we detect lines reliably and repeatably despite varying lighting conditions. The presence of many unstable line detections can lead to false tracking and unreliable pose estimates [33]. In addition, the space environment often presents regions of low brightness and contrast, especially when parts of the spacecraft are occluded or illuminated indirectly. It is expected that throughout the frames, parts of the line segment will be occluded or fragmented due to self-shadowing. If a line is detected into subparts, it might lead to erroneous matches when some of the subparts are occluded. Hence, the second goal of our method is to learn to connect fragmented lines, by encouraging the network to connect predicted lines.

4.1. DeepLSD architecture

IRoLSD adopts the DeepLSD network architecture [10], since it combines the robustness of deep learning methods with the pixel-level accuracy of hand-crafted detectors. In the core of the architecture lies the AFM representation [25], a continuous vector field that represents the distance and orientation of each pixel with respect to its closest line segment. This representation allows the detector to only implicitly depend on endpoints and junctions while still being able to learn low-level image details and maintain the geometric structure information. The overall line detection process is organized into the following steps.

Self-supervised data generation: Instead of relying on manual line annotations, DeepLSD proposes to bootstrap the output of a handcrafted line segment detector (LSD) under multiple homographic warpings. Similarly to the homographic adaptation step of SuperPoint [34], every image is augmented with N homographic warpings and the detector is applied both to the image and its warpings. The detections in the warped images are unwrapped to the original image, and all the generated attraction fields are aggregated using the pixel-wise median to suppress noisy measurements.

Attraction Field learning: The Convolutional Neural Network that is trained to estimate the Attraction Field is a U-Net-like architecture [35]. The network has two branches to estimate the distance and angle fields separately. The distance field loss is an L_1 loss between the normalized distance fields, while the angle field loss is an L_2 angular loss between the angle fields.

Line Segments Extraction: The predicted attraction field is transformed into a surrogate image gradient of magnitude and orientation, producing a smoothed representation of the underlying image edges and serving as a refined gradient map for traditional line extraction. In essence, the network is trained to suppress noise and enhance low-contrast lines. The final set of line segments is then obtained using the traditional LSD (Line Segment Detector) [17] applied to this enhanced gradient map.

The hybrid nature of DeepLSD guarantees pixel-level precision in the supervision data generation and the final detection. However, the reliance on handcrafted detectors for the supervision process limits the network's performance. We address this limitation by incorporating the Illumination Robustness (IR) and Depth-Assisted (DA) modules.

4.2. Illumination Robustness (IR)

The DeepLSD ground-truth generation scheme is based on bootstrapping a traditional line detector. The homographic adaptation scheme is a form of making the ground truth lines robust. We want to exploit a similar bootstrapping technique to detect lines that are robust against illumination.

To this end, let $f_{rv}(\cdot)$ denote the image rendering function and let $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ represent the resulting grayscale image defined in the pixel domain Ω .

1. $\mathcal{W}$: the World frame, which is attached arbitrarily on the target spacecraft
2. $\mathcal{C}$: the Chaser frame, which is attached on and aligned with the camera of the chaser spacecraft
3. $\mathcal{E}$: the Earth frame, defined as the Earth-Centered Earth-Fixed (ECEF)
4. $\mathcal{S}$: the Sun frame, which is an arbitrary frame attached on the Sun. The orientation of this frame does not matter since the Sun is spherically symmetrical.

We define $T_{\mathcal{E}\mathcal{W}} \in \text{SE}(3)$ as the relative pose of the Earth with respect to the World frame. The relative position of the Sun to the target, expressed in spherical coordinates in the World frame, is denoted as: $r_S = r_{S\mathcal{W}}^{\mathcal{W}} = [\phi, \theta, \rho]^T$, where ϕ and θ are the azimuth and elevation angles, respectively, and ρ is the radial distance.

In our setup, the components that affect illumination in the scene are the direct sunlight, which is the most intense and dominant light source, and the indirect illumination from the Earth's albedo. Assuming that the 3D models and the texture properties of the target and the Earth are fixed and known, and the camera parameters are also fixed, the observed pixel intensities in the image are dependent only on the relative configuration of the Sun, Earth, camera, and target frames. As a result, we can describe the image rendering function as

$$I = f_{rv}(T_{C\mathcal{W}}, T_{\mathcal{E}\mathcal{W}}, r_S). \quad (17)$$

For the scope of this work, we assume that illumination perturbations are introduced solely by varying the position of the primary lighting source, namely the Sun, while keeping the relative configuration of the other actors fixed. This is a fair assumption, since the effect of the slightly moving Earth has minimal impact on the final image. Since the Sun is constrained to remain outside the camera's field-of-view to avoid direct overexposure, the rendered perturbed images have consistent structural content, with changes only in shading, contrast, and visibility as a result of illumination dynamics.

Let us define the perturbed Sun position as

$$r'_S = r_S + \delta r_S, \quad (18)$$

where each angle of the illumination perturbation $\delta r_S = [\delta\phi, \delta\theta, 0]^T$ is sampled independently and uniformly within the $[-1, 1]$ interval in degrees. We define the perturbed image as

$$I' = f_{rv}(T_{C\mathcal{W}}, T_{\mathcal{E}\mathcal{W}}, r'_S). \quad (19)$$

For each image I , consider N random perturbations of the lighting direction, denoted as $\delta r_S^{(i)}$, and let $f_d(\cdot)$ represent a line segment detection function, such that the set of line segments detected in the original image is given by

$$\mathcal{L} = f_d(I). \quad (20)$$

Then, the set of line segments detected in the perturbed image with lighting direction, denoted as r'_S , is given by

$$\mathcal{L}' = f_d(I'). \quad (21)$$

Illumination Adaptation: An ideal illumination robust line detector would yield $\mathcal{L} = \mathcal{L}'$ for any perturbed image. Notice that equality

between sets refers to empirical correspondence of the line segments, calculated based on structural and angular distance thresholds. In practice, when the target has complex geometry and realistic surface texture properties, even small perturbations will result in noticeable changes in the pixel intensities. These changes disturb the local appearance of the line segments due to differences in brightness, contrast, or self-shadowed pixels. As a result, the line segment detectors are expected to have different outputs even for small perturbations in the lighting direction. Notice that, for large perturbations, the resulting image might be so different that the intersection of $\mathcal{L}$ and $\mathcal{L}'$ becomes the empty set.

To this end, let $\mathcal{L}^{(i)}$ denote the output of the line segment detector on the i th perturbed image, and let $i = 0$ represent the original image, i.e., $\mathcal{L} = \mathcal{L}^{(0)}$. We define the set of illumination robust line segments as the intersection between the line segments detected in the original image and all the perturbed images:

$$\mathcal{L}^* = \bigcap_{i=0}^N \mathcal{L}^{(i)}. \quad (22)$$

As a final consideration, we need to enforce that the inherently noisy line segment detection process results in a set of illumination-robust lines $\mathcal{L}^*$ that produces a smooth supervisory signal for the Attraction Field learning process. The authors of DeepLSD [10] propose to aggregate the detections of homographic adaptation by directly transforming each set of line segments into a distance and angle fields, assigning to each pixel the median value of its distance and angle across all homographic warpings. However, in our case, the impact of illumination perturbation is more significant than a simple homographic transformation. Unlike homographies, changes in lighting direction can lead to partial occlusions or shadowing, causing the original line segments to split into fragments. These effects disrupt the smoothness of the distance and angle fields and degrade the quality of the supervision signal.

To mitigate this, we propose a binary heatmap approach inspired by the way the original SuperPoint paper [34] handles the aggregation of point features. Specifically, let $g(\cdot)$ represent a function that converts a set of detected line segments into a one-hot encoded heatmap, marking pixels that are part of the line segments. The resulting aggregated heatmap over the original and the perturbed images, I^* , is defined as

$$I^* = \frac{1}{N+1} \left(g(\mathcal{L}) + \sum_{i=1}^N g(\mathcal{L}^{(i)}) \right). \quad (23)$$

In this representation, fragmented line segments across perturbations yield weaker intensity values in the heatmap, but these are easily distinguishable from spurious pixel detections. In this way, we are able to connect line segments that are possibly fragmented across images, while avoiding line segments that are not illumination robust. Hence, with a sufficiently large N , this averaging process allows us to distinguish robust line segments from spurious detections and connect lines that are being fragmented in some of the images, extracting a reliable set of illumination-invariant line segments from I^* .

4.3. Depth-assisted line generation

As discussed previously, an illumination-robust set of line segments should yield a smooth and stable attraction field representation. Two key characteristics of realistic spacecraft imagery that pose challenges to this goal are: (i) certain regions of the spacecraft may only be indirectly illuminated, resulting in low brightness and poor contrast, and (ii) self-occlusions caused by complex geometry can cause even prominent structural lines to appear fragmented into multiple shorter segments. To address these challenges, we propose to take advantage of the depth map associated with each image. Depth maps preserve strong gradients along structural boundaries independently from the lighting direction. Since our training data consist of synthetic images, the corresponding relative depth information is readily available and can be used to construct smoother and more precise attraction fields.

It should be mentioned that, despite incorporating depth information during training, we do not assume access to the 3D model of the target. Depth maps are used solely to enhance supervision and are not required during inference, which is one of the benefits of the proposed hybrid detector architecture. Our model is trained and evaluated on different spacecraft models, demonstrating its generalization ability without relying on prior 3D shape knowledge (see Section 5).

Let $f_{rd}(\cdot)$ denote the depth rendering function, and let $D : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ represent the resulting depth map defined over the pixel domain Ω . Note that D does not take into account the lighting direction, and given that we are interested in the spacecraft line segments, we can exclude the Earth from the rendering process. We can describe the depth rendering function as

$$D = f_{rd}(T_{CW}). \quad (24)$$

Applying the same line segment detector f_d on D , we obtain

$$\mathcal{L}_d = f_d(D). \quad (25)$$

Let $\mathcal{L}_d$ contain the structural line segments that can be projected on the image generated from the specific relative pose T_{CW} , without taking into account the illumination conditions.

Although depth-based line segment detection addresses the challenges of low contrast and occlusions, it captures geometric information independently of lighting, potentially identifying line segments that are not perceivable in the original image I . To ensure the reliability of supervision, we wish to exclude depth-detected line segments that are not visible, which means that they are not verified by the intensity gradients in I . Since $\mathcal{L}_d$ detects structural lines in a noise-free manner, we are confident that for a line segment of $\mathcal{L}_d$, even small gradients in the original image correspond to visible line segments, as long as the gradients are consistent throughout the largest portion of the line segment. Conversely, we define a line segment as invisible if the image does not exhibit a strong and consistent gradient along the extent of the segment. Specifically, this occurs when the magnitude of the gradient is uniformly low across the segment or when significant gradients appear only sparsely, covering only a small portion of the line segment. To assess visibility, we introduce a gradient sufficiency criterion based on a manually defined threshold on the gradient magnitude. Notably, because of the existence of depth edges, this means that structurally there exists a line segment, and hence our threshold can be set lower than those typically used in handcrafted line detectors. This eventually allows for more permissive and confident line detection in low-contrast regions.

Formally, we decompose the set of line segments detected on the depth map $\mathcal{L}_d$, into the set of line segments visible in the original image $\mathcal{L}_d^v$ and the set of line segments invisible in the original image $\mathcal{L}_d^{\neg v}$ such that:

$$\mathcal{L}_d^v \cup \mathcal{L}_d^{\neg v} = \mathcal{L}_d, \quad \mathcal{L}_d^v \cap \mathcal{L}_d^{\neg v} = \emptyset. \quad (26)$$

Let $\mathcal{V}(\ell, I)$ be a visibility function that returns 1 if a line segment ℓ is considered visible in the original image I , and 0 otherwise. For each line segment ℓ , we calculate the number of pixels $p \in \ell$, whose gradient is above a predefined threshold τ . If this number is above a threshold α of the length of the total pixel of the line segments $|\ell|$, then the line segment is deemed visible. Let $\mathbf{1}(\cdot)$ denote an indicator function that returns 1 if the input condition is true, and 0 otherwise. We can then formulate the visibility function

$$\mathcal{V}(\ell, I) = \begin{cases} 1, & \text{if } \sum_{p \in \ell} \mathbf{1}(\|\nabla I(p)\| > \tau) \geq \alpha |\ell|, \\ 0, & \text{otherwise.} \end{cases} \quad (27)$$

The set of line segments visible in the original image can then be defined as

$$\mathcal{L}_d^v = \{\ell \in \mathcal{L}_d \mid \mathcal{V}(\ell, I) = 1\}. \quad (28)$$

Using the information provided from the depth map requires encoding $\mathcal{L}_d^v$ in the representation of the aggregated heatmap using the $g(\cdot)$

function. We introduce a weighting parameter $\beta > 0$, (empirically set to 0.3), to preserve the one-hot encoding representation while controlling the influence of the depth-based detections

$$I^* = \beta g(\mathcal{L}_d^v) + \frac{1-\beta}{N+1} \left(g(\mathcal{L}) + \sum_{i=1}^N g(\mathcal{L}^{(i)}) \right). \quad (29)$$

In conclusion, utilizing the information from the depth map offers two key benefits relating to the illumination conditions of the scene:

1. **Connecting fragmenting lines:** Line segments that are detected in multiple smaller parts in the image are more likely to be correctly connected by enforcing geometric continuity from the depth map. In contrast to the ELSESED approach [19] that uses pixel jumps, there is no risk of spurious detections since connectivity is enhanced only from structurally existing lines.
2. **Enhancing low-lit line segments:** Line segments in poorly illuminated regions of the spacecraft that exhibit low gradients in the original image are recovered, ensuring a smoother AFM supervision signal.

5. Experiments

We compared IRO LSD against a selection of both handcrafted and learned line segment detectors. Among traditional approaches, we include LSD [17] and ELSESED [19]. LSD is one of the most widely used detectors in the literature and serves as a strong baseline. ELSESED, on the other hand, incorporates pixel jumps, directly addressing the line fragmentation problem, exhibiting superior performance in man-made environments with structurally longer lines.

For the case of learned detectors, we evaluated two state-of-the-art methods: SOLD2 [12] and DeepLSD [10]. Both methods incorporate a homographic adaptation scheme, which is fundamental for the robustness of the detections. SOLD2 [12] represents line segments via their endpoints and is a state-of-the-art approach for joint detection and description. DeepLSD on the other hand, is reportedly outperforming most state-of-the-art line segment detectors on a variety of tasks and can offer insights on the use of the AFM representation for our specific problem.

5.1. Datasets

Existing datasets do not incorporate varying lighting conditions, which are essential for the Illumination Adaptation step of our network that involves perturbing the direction of the light source. To address this, we use SISIFOS [36], a physically accurate spacecraft simulator that generates synthetic images of spacecraft orbiting Earth, with control over both illumination sources, namely, the Sun and the Earth. A sample of images generated using SISIFOS are shown in Fig. 3. We generate images under randomly sampled scene configurations while enforcing the following constraints to ensure realistic and meaningful training and evaluation. First, we exclude configurations in which either the Earth or the Sun appears directly within the camera field of view. The Earth is excluded because its appearance dominates the image and may interfere with line detection, whereas our network is intended to operate on views of the spacecraft after object detection has been conducted.

Previous research by Chen et al. [37,38] has demonstrated that object detection on spacecraft imagery can be performed with high accuracy. Accordingly, we assume that IRO LSD operates on images where object detection has already been successfully applied. Fig. 4 illustrates the results of applying the pre-trained Depth Anything V2 model [39] to representative examples of different Earth backgrounds. Due to the sensitivity of the V2 model to fine structural details, it can produce reliable relative depth estimates, thereby enabling precise separation of the spacecraft from the surrounding background. Exploiting the clear

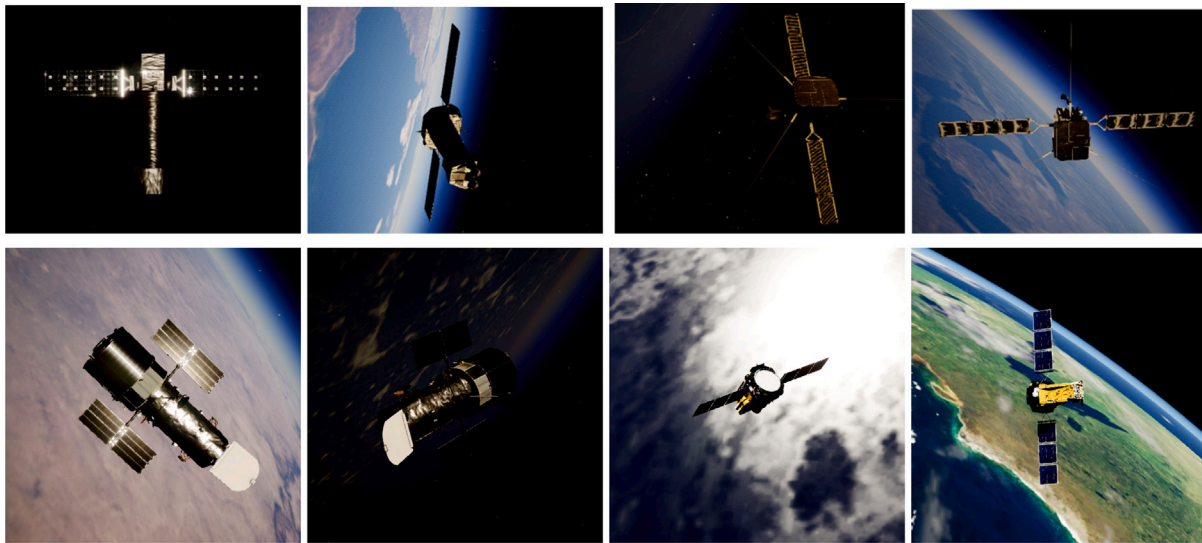


Fig. 3. Gallery of selected images of different spacecraft models under various parameter settings of the SISIFOS simulator [36].

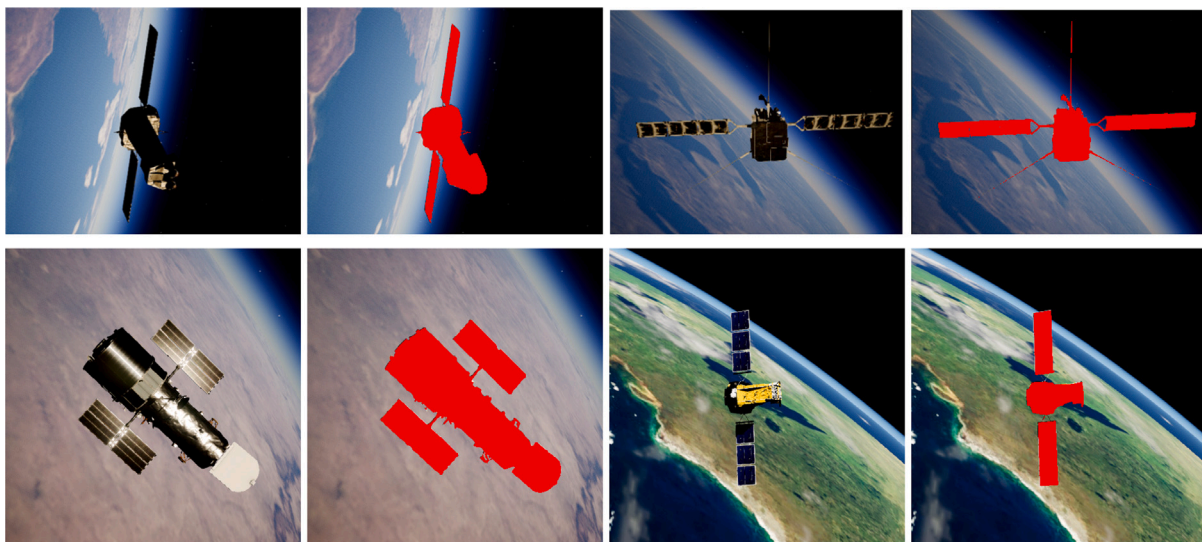


Fig. 4. Object detection results on images of the SISIFOS simulator [36]. Each row contains two image pairs of different spacecraft models, where the left image of each pair depicts the original RGB image and the right image illustrates the mask of the detected target shown in red.

depth distinction between the spacecraft, Earth, and the background, we cluster the depth predictions into three groups and select the cluster corresponding to the smallest depth values, which typically represents the foreground spacecraft. It is observed that the primary detection inaccuracies occur in dimly lit and structurally thin regions of the spacecraft positioned against the dark background of space. These areas are inherently challenging to segment, even for human observers, as their identification relies more on contextual information than on local visual features.

The Sun is excluded to prevent overexposed images that, in real-world scenarios, could saturate or damage the imaging sensor. Furthermore, we discard configurations in which more than 80% of the target spacecraft pixels are rendered almost completely dark. Such configurations are deemed uninformative, as they offer minimal visual content for either training or evaluation of the line detection network. We train IRO LSD on 1,000 synthetic images of the BEPICOLOMBO spacecraft, provided by ESA SciFleet [40]. Fig. 5 shows example images of the training dataset.

For each training image, we additionally generate 20 perturbed variants under perturbed lighting directions. The spacecraft model selected for training contains sufficient geometric structure information, yet is not overcomplicated, allowing IRO LSD to focus on capturing generalizable geometric features common to most spacecraft.

An important aspect of our method is the generalizability of the network. Depth maps are only utilized during the training process and no additional information is assumed during inference except the visual image. Hence, it is important to evaluate the performance of IRO LSD on different spacecraft models than the training one.

We evaluated IRO LSD on the SPEED+ dataset [41], which contains both synthetic and laboratory-generated images of the TANGO spacecraft. We randomly sampled 60 images from the *lightbox* and 60 images from the *sunlamp* domains. These images are distinctively different from the training dataset: TANGO is a simpler spacecraft model, with panels mounted on the body of the spacecraft and not expanded, that exhibit fine and dense line features.

Apart from SPEED, we developed our own evaluation dataset using the SISIFOS simulator [36], which consists of synthetic images of the following ESA spacecraft models: MARS EXPRESS, SMART-1,

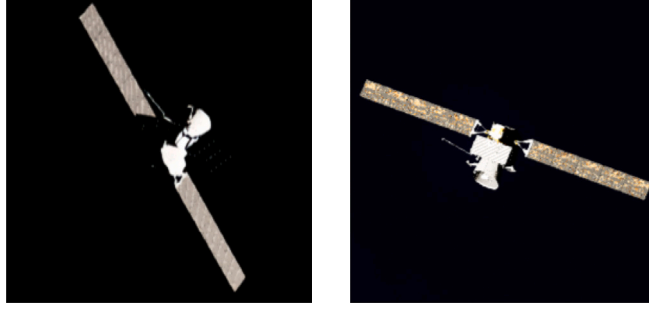


Fig. 5. Example images of BEPICOLOMBO spacecraft from the training dataset.

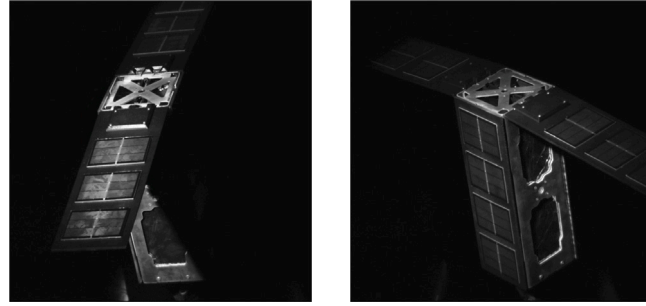


Fig. 6. Example images of the 3U CubeSat model captured in the ASTROS arena.

INTEGRAL, and XMM-NEWTON. For each model, we generated 120 different poses and various images per pose with perturbed lighting directions.

Finally, we developed a realistic image sequence using the ASTROS facility [42] with a custom built 3U CubeSat model equipped with two deformable panels as target (Fig. 6). The CubeSat is mounted on a UR10e robotic arm and is rotated, while illuminated from a collimated light source, mimicking direct sunlight in space. A free-floating robot equipped with a camera captures images while orbiting around the target, introducing realistic effects such as actuation-induced jitter and target tracking inaccuracies. As the spacecraft rotates, different surfaces become illuminated, emulating the natural variation in appearance encountered during an orbital trajectory, as can be observed in the images in Fig. 6.

5.2. Evaluation metrics

An integral part of the evaluation of line detection algorithms is the assessment of whether two line segments in different images represent the same line segment. Typically, this is done using descriptors and MNN sets as described in Section 2.2. However, in the case of pairs of images (I_1, I_2) , where I_2 is generated by applying a known reversible geometric transformation to I_1 , we can use this transformation to map the line segments of image I_2 in the image space of image I_1 . Subsequently, we can use the line segment distance metrics defined in Section 2.1 as similarity measures and employ typical manual thresholding to create the equivalent Mutual Nearest Neighbor set without using descriptors. Depending on the distance metric used to create the MNN set, we have the *structural* or *orthogonal* versions of the following metrics.

Let the sets of detected line segments in I_1 and I_2 have cardinalities N_1 and N_2 , respectively, and let the set of the corresponding line segments $M(I_1, I_2)$ have cardinality equal to M^* . We define the **repeatability** metric, denoted as Rep , as the portion of line segments in I_1 that are also detected in I_2 :

$$Rep = \frac{M^*}{N_1}. \quad (30)$$

To assess the accuracy of each detected line segment, the average distance between repeated line segments is commonly used in previous works, such as SOLD2 [12] and DeepLSD [10]. In this formulation, a set of fragmented short segments with distance e , will produce the same average distance as a single long segment with the same distance e . However, the latter captures the underlying geometric structure more accurately.

We propose to define the **localization error**, denoted as LE , as the length-inverse-weighted average distance between corresponding repeatable line segments

$$LE = \frac{\sum_{i=1}^{M^*} d_i / \|\ell_i\|_2}{\sum_{i=1}^{M^*} 1 / \|\ell_i\|_2}, \quad (31)$$

where (ℓ_i, ℓ'_i) , $i = 1, \dots, M^*$ are the pairs of repeated line segments and d_i , $i = 1, \dots, M^*$ are the distances for each pair. By weighting each segment's error inversely proportional to its length, we favor longer segments, which contribute more meaningfully to structural understanding, while still penalizing localization inaccuracies.

Finally, the **angular error**, denoted as AE , is defined as the average angle distance between repeated line segments in degrees

$$AE = \frac{\sum_{i=1}^{M^*} d_\phi(\ell_i, \ell'_i) \frac{180}{\pi}}{M^*}. \quad (32)$$

In addition to the distance-related metrics, we report some general metrics on the detected line segments. The number of detected line segments, denoted as LPI , is simply the cardinality N_1 of the set of detected line segments in image I_1 . The average length of repeated line segments, denoted as $AvgLL$, is calculated as

$$AvgLL = \frac{\sum_{i=1}^{M^*} \|\ell_i\|_2}{M^*}, \quad (33)$$

where (ℓ_i, ℓ'_i) , $i = 1, \dots, M^*$ are the pairs of repeated line segments. For easier comparison, we provide the total coverage in pixels, denoted as Cov , which is the total number of repeated pixels in the first image:

$$Cov = \sum_{i=1}^{M^*} \|\ell_i\|_2 = M^* \times AvgLL = LPI \times Rep \times AvgLL. \quad (34)$$

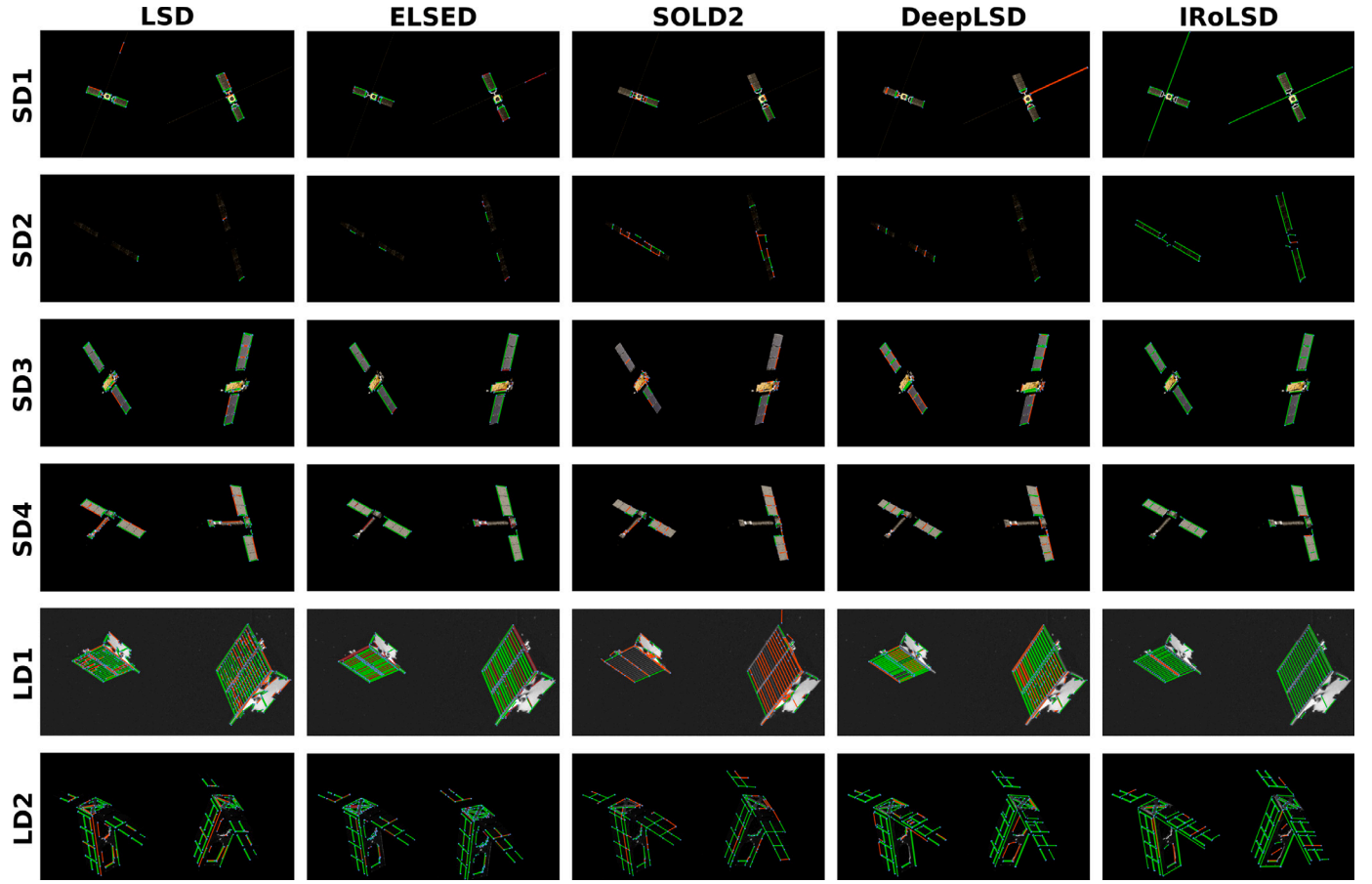


Fig. 7. Qualitative results of homographic adaptation experiment. For each dataset, we demonstrate a representative pair and the performance of IRoLSD and the compared algorithms. For each pair, the left image is the original and the right image is the warped version.

This metric is also reported for the two different types of distances, *structural* or *orthogonal*. Finally, we report the percentage of failure cases, denoted as *Fail%*, as the percentage of test cases where no line correspondences were found as $M^* = 0$ of the total number of test cases. Failure cases are excluded from the error calculations, since the errors are only defined on repeated line segments.

5.3. Experimental scenarios

We evaluated the line detection performance in three scenarios: homographic adaptation, illumination adaptation, and trajectory sequences.

5.3.1. Homographic adaptation

For each image I in the test set, we sample a homographic transformation $H \in \mathbb{R}^{3 \times 3}$ and generate the corresponding pair of images (I, HI) . Given a point in homogeneous coordinates $\bar{p}' \in \mathbb{R}^3$ in image HI , the corresponding point in image I , $\bar{p} \in \mathbb{R}^3$, can be obtained via the inverse homography mapping

$$\bar{p} = H^{-1} \bar{p}'. \quad (35)$$

This transformation is applied to all line segment endpoints to reproject them on the first image. For each image, we repeat this process for five homographic adaptations to limit the effect of favorable homographies in the results.

The results presented in Tables 1 and 2 demonstrate the superior performance of our algorithm compared to both learning-based and traditional methods. IRoLSD consistently exhibits significantly higher repeatability across all datasets, along with lower localization and angular errors. Our algorithm's ability to detect longer lines and achieve

broader pixel coverage indicates that IRoLSD effectively learns geometric patterns under varying lighting conditions and successfully connects fragmented line segments. As illustrated in the visual comparisons of Fig. 7, our algorithm achieves consistently higher repeatability and lower errors on both synthetic and real datasets. The coverage metric further underscores the advantage of detecting longer line segments, as these are more likely to be consistently re-detected. This is particularly evident in the SD2 dataset, where the low-texture and low-contrast surfaces of the SMART1 model result in darker images. In such conditions, all other detectors struggle to find repeatable line segments, leading to high failure rates. In contrast, IRoLSD reliably identifies at least one repeatable line segment, even under challenging lighting conditions. This robustness stems from its training to recognize geometric structures regardless of brightness or contrast variations.

5.3.2. Illumination Robustness

In this experiment, we assess the performance of the line detection algorithm under different levels of brightness differences in the image pair. For each relative pose T_{C_W} in the test set, we create a group of images with the same pose and different lighting directions. We formulate three similarity bins, *high*, *medium*, *low*, based on the maximum and minimum brightness differences of each data set. For each group, we select the brightest image based on the total pixel intensity and split the rest of the images into the three similarity bins. Then, we sample a pair for every similarity bin, where the first image is the brightest image of the group, and the second image is randomly selected from the similarity splits. For the metrics of the average line length and total pixel coverage, we present the mean value of the structural and orthogonal versions of the metric.

Table 1

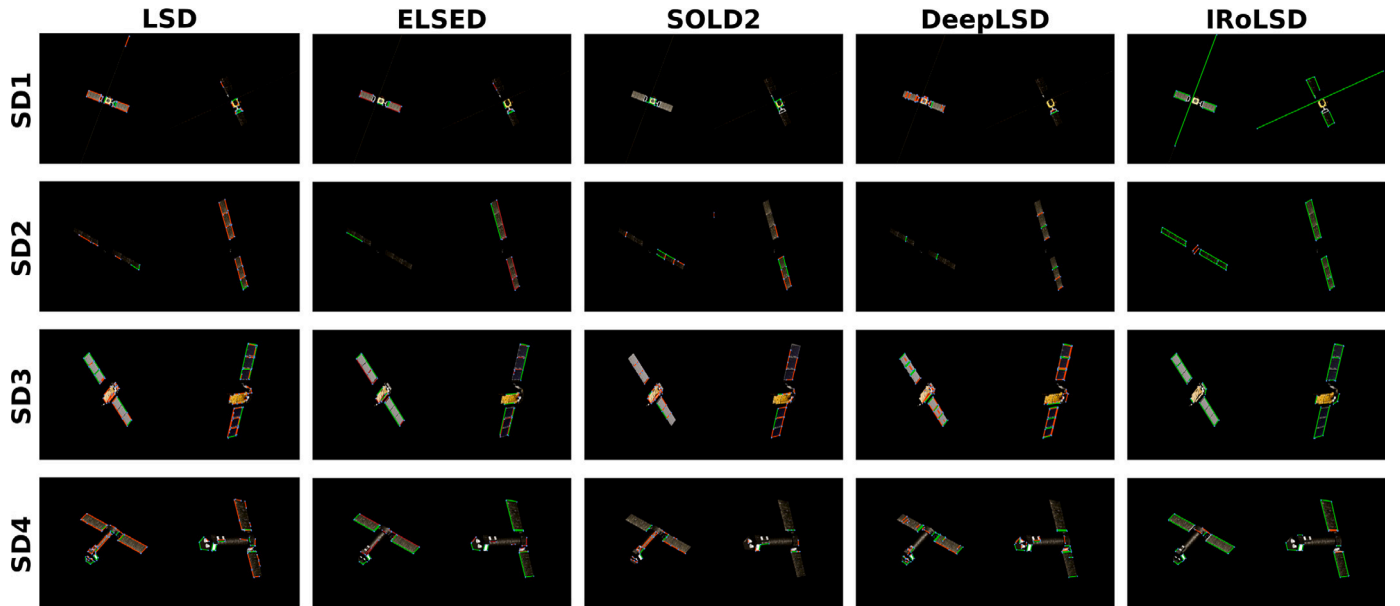
Homographic adaptation results on synthetic datasets: SD1 (MARS EXPRESS), SD2 (SMART1), SD3 (INTEGRAL).

		LSD			ELSEd			SOLD2			DeepLSD			IRoLSD		
		SD1	SD2	SD3	SD1	SD2	SD3	SD1	SD2	SD3	SD1	SD2	SD3	SD1	SD2	SD3
Struct	Rep ↑	0.53	0.53	0.59	0.49	0.58	0.54	0.23	0.25	0.32	0.49	0.28	0.54	0.72	0.78	0.70
	LE ↓	5.53	3.83	4.76	5.86	3.98	5.14	7.57	6.43	6.45	5.68	4.85	4.78	3.91	3.75	4.03
	AE ↓	0.94	0.71	0.88	1.13	0.95	1.01	1.68	1.4	1.43	1.55	1.12	1.15	0.91	0.74	0.82
	AvgLL	19.0	37.3	23.3	26.3	47.1	33.7	15.2	17.12	18.1	19.4	17.5	21.3	65.9	63.2	48.1
	Cov	128	160	305	153	199	334	64	25	134	127	29	262	607	424	525
Orth	Rep ↑	0.62	0.66	0.69	0.64	0.69	0.65	0.30	0.28	0.38	0.49	0.28	0.59	0.73	0.81	0.72
	LE ↓	3.52	2.65	3.22	4.23	2.89	3.27	5.28	4.85	4.71	3.81	4.67	3.43	1.86	1.73	1.72
	AE ↓	1.00	0.95	1.14	1.07	1.02	1.18	1.57	1.12	1.52	1.59	1.05	1.41	0.89	0.81	0.78
	AvgLL	20.3	37.6	23.4	28.1	48.1	35.9	15.4	18.7	21.1	20.7	19.5	25.3	67.4	65.2	50.0
	Cov	160	201	358	214	242	429	85	30	186	135	32	340	630	454	562
LPI		12.7	8.1	22.2	11.9	7.3	18.4	18.3	5.8	23.2	13.3	5.8	22.8	12.8	8.6	15.6
Fail%		5.00	33.3	0.00	5.73	33.3	0.00	8.33	64.2	7.50	5.73	48.2	2.50	0.83	0.83	0.00

Table 2

Homographic adaptation results on synthetic dataset SD4 (XMM NEWTON) and laboratory-generated datasets: LD1 (SPEED), LD2 (ASTROS).

		LSD			ELSEd			SOLD2			DeepLSD			IRoLSD		
		SD4	LD1	LD2	SD4	LD1	LD2	SD4	LD1	LD2	SD4	LD1	LD2	SD4	LD1	LD2
Struct	Rep ↑	0.60	0.56	0.74	0.59	0.58	0.64	0.33	0.54	0.57	0.57	0.57	0.78	0.80	0.66	0.82
	LE ↓	4.32	4.33	3.55	4.21	3.96	3.42	7.76	5.70	5.75	4.63	3.59	3.26	3.93	3.47	2.83
	AE ↓	0.68	0.74	0.68	0.86	0.85	0.72	1.89	1.67	1.20	1.29	0.76	0.62	0.72	0.56	0.65
	AvgLL	29.1	36.7	32.2	32.1	39.4	33.7	19.3	28.1	22.5	24.1	43.2	34.2	41.6	56.4	38.1
	Cov	370	1099	1491	394	1111	1631	138	1159	950	253	1374	2051	702	1384	2218
Orth	Rep ↑	0.73	0.65	0.76	0.68	0.62	0.67	0.39	0.63	0.64	0.59	0.62	0.77	0.80	0.69	0.82
	LE ↓	2.56	2.56	2.16	2.78	3.28	2.76	5.45	4.39	3.58	3.06	1.86	1.94	1.81	1.33	1.65
	AE ↓	0.78	0.70	0.72	0.95	0.75	0.98	1.71	1.63	1.18	1.30	0.82	0.64	0.90	0.56	0.69
	AvgLL	29.2	37.2	33.1	33.8	41.4	34.9	22.6	28.4	22.7	25.2	39.2	35.2	44.8	58.9	41.2
	Cov	452	1294	1574	478	1247	1768	191	1367	1077	274	1356	2084	756	1512	2398
LPI		21.2	53.5	62.6	20.8	48.6	75.6	21.7	76.4	74.1	18.4	55.8	76.9	21.1	37.2	71.0
Fail%		0.00	0.83	0.00	0.00	0.80	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

**Fig. 8.** Qualitative results of Illumination Robustness experiment. For each dataset, we demonstrate a representative pair and the performance of IRoLSD and the compared algorithms. For each pair, the left image is the origin and the right image is the lighting perturbed and homographically warped one.

In the homographic adaptation case, a low-brightness image would affect the detection process in both images, while here it will only affect it in the second image. We expect that having one image of the compared pair with high brightness will result in an increase in the number of detected line segments, which will help us evaluate the performance more accurately under different lighting conditions.

As shown in Table 3, IRoLSD maintains a clear and consistent advantage over other line segment detection algorithms under varying illumination conditions. This robustness highlights the ability of our method to learn generalizable geometric characteristics that are resilient to lighting perturbations. As expected, the highest repeatability

scores are obtained at higher similarity levels, where the line segments in the image pairs are closely aligned. However, the results of the SD2 dataset exhibit an increase in the localization and angle error metrics as the similarity increases. Due to the low texture and contrast of the SMART1 model, line detection is inherently more difficult and the sets of detected line segments are scarce. This also makes the problem of general line detection more difficult, as reported in the homographic adaptation case as well. As a result, higher similarity levels between images leads to more line segments to be detected, or longer line segments to be detected, hence increasing the error metrics.

Table 3

Illumination adaptation results with different levels of brightness similarity on the synthetic datasets: SD1 (MARS EXPRESS), SD2 (SMART1), SD3 (INTEGRAL), SD4 (XMM NEWTON).

	Similarity	LSD			ELSED			SOLD2			DeepLSD			IRoLSD			
		Low	Med	High	Low	Med	High	Low	Med	High	Low	Med	High	Low	Med	High	
SD1	Struct	Rep ↑	0.24	0.34	0.48	0.21	0.28	0.36	0.12	0.15	0.18	0.18	0.24	0.39	0.50	0.60	0.64
		LE ↓	7.96	7.18	5.86	8.41	7.54	6.32	12.22	10.87	9.40	9.32	6.80	6.07	7.74	6.91	5.52
		AE ↓	1.54	1.42	0.93	1.66	1.52	0.98	2.50	1.78	2.02	2.70	2.25	1.96	1.12	1.05	0.84
		AvgLL	18.2	19.3	21.4	21.7	22.0	24.1	15.4	15.1	14.8	17.7	17.8	21.9	59.4	59.6	60.0
		Cov	73.0	121	200	108.1	156	182	41.3	52.8	64.8	49.1	83.1	153	371	554	532
	Orth	Rep ↑	0.33	0.40	0.57	0.36	0.42	0.53	0.19	0.20	0.25	0.28	0.29	0.42	0.52	0.60	0.65
		LE ↓	4.41	4.93	3.78	3.31	3.54	2.99	7.76	7.77	7.45	6.23	5.68	4.20	2.68	2.36	1.81
		AE ↓	1.46	1.61	1.11	1.22	1.28	0.97	2.16	1.91	1.88	2.49	2.44	1.90	1.10	0.99	0.80
		AvgLL	18.4	19.5	21.5	21.9	22.4	24.6	15.6	15.2	14.9	17.9	18.1	22.1	59.7	59.9	60.5
		Cov	92.4	141	242	132	178	226	61.2	67.4	88.2	68.7	94.8	182	412	582	564
	LPI	16.1	17.5	18.8	15.2	16.3	16.9	23.5	24.7	26.6	18.4	25.4	21.8	15.3	19.7	15.7	15.7
	Fail%	11.6	3.34	1.67	10.0	3.34	1.67	13.3	11.6	6.74	10.0	1.67	3.37	1.67	0.00	0.00	0.00
SD2	Struct	Rep ↑	0.36	0.43	0.49	0.32	0.37	0.41	0.31	0.25	0.36	0.34	0.27	0.41	0.53	0.63	0.66
		LE ↓	4.24	4.27	4.67	4.88	5.14	5.63	6.21	6.83	8.81	3.90	4.21	3.82	3.73	3.99	4.21
		AE ↓	0.85	0.94	1.03	0.91	1.02	1.08	1.68	0.96	1.02	1.57	0.95	1.13	0.78	0.75	0.84
		AvgLL	34.8	41.0	38.6	44.0	45.5	48.5	25.1	18.0	22.3	25.9	19.1	21.9	80.0	80.0	82.6
		Cov	124	154	195	203	212	228	71.3	46.9	71.2	83.2	49.4	77.3	394	426	441
	Orth	Rep ↑	0.53	0.57	0.60	0.51	0.56	0.63	0.34	0.31	0.41	0.32	0.34	0.37	0.68	0.73	0.75
		LE ↓	2.22	2.27	2.68	1.97	2.12	2.19	3.90	3.98	3.81	3.88	3.95	3.82	1.89	1.50	2.25
		AE ↓	1.01	0.91	1.13	0.92	0.96	0.99	1.57	1.14	1.05	1.48	1.01	1.04	0.78	0.69	0.91
		AvgLL	35.4	41.2	39.5	44.2	46.2	48.9	25.6	18.7	22.8	26.6	19.3	22.6	80.5	80.4	82.9
		Cov	147	193	215	226	244	276	84.4	53.1	77.8	89.1	57.3	85.6	432	459	483
	LPI	11.0	10.9	12.1	11.2	11.8	10.9	7.3	7.2	7.0	9.7	10.6	10.3	10.6	11.1	9.8	9.8
	Fail%	7.50	5.00	3.33	5.00	3.33	1.67	16.7	12.5	10.0	16.7	12.5	10.0	0.00	0.00	0.00	0.00
SD3	Struct	Rep ↑	0.26	0.39	0.44	0.32	0.36	0.46	0.19	0.21	0.28	0.21	0.32	0.39	0.48	0.59	0.61
		LE ↓	6.64	5.41	5.29	7.12	6.68	5.72	8.42	8.10	7.91	7.22	6.58	6.11	5.82	5.47	4.91
		AE ↓	0.99	0.88	0.82	1.11	1.06	1.02	1.60	1.56	1.35	1.78	1.15	1.25	0.89	0.86	0.79
		AvgLL	27.7	27.4	28.7	31.5	32.5	34.6	18.6	19.4	19.2	29.7	33.3	30.4	48.6	53.2	56.3
		Cov	265	409	472	331	424	487	117	124	168	231	419	457	429	577	623
	Orth	Rep ↑	0.35	0.48	0.55	0.31	0.37	0.42	0.26	0.29	0.36	0.25	0.39	0.48	0.45	0.57	0.60
		LE ↓	5.79	3.69	3.09	6.24	5.53	4.98	7.40	5.89	5.69	5.77	3.72	3.45	3.59	2.72	2.17
		AE ↓	1.24	0.91	0.88	1.16	1.04	0.99	1.52	1.39	1.37	1.96	1.22	1.31	0.76	0.71	0.74
		AvgLL	27.9	27.7	28.9	31.8	32.5	34.9	18.9	19.7	19.4	29.8	33.7	30.8	49.1	53.6	56.6
		Cov	289	427	496	366	457	513	143	147	179	263	434	481	482	614	668
	LPI	36.5	40.6	37.8	31.2	34.6	35.1	35.1	33.4	31.75	40.8	41.8	41.7	24.4	26.4	23.2	23.2
	Fail%	2.33	0.00	0.00	1.67	0.00	0.00	6.33	0.00	0.00	4.67	0.00	0.00	0.00	0.00	0.00	0.00
SD4	Struct	Rep ↑	0.23	0.32	0.43	0.28	0.33	0.38	0.14	0.16	0.26	0.26	0.36	0.49	0.50	0.61	0.71
		LE ↓	6.14	4.96	5.14	7.11	6.23	5.37	8.65	9.38	8.92	5.14	4.55	4.61	6.11	4.77	4.49
		AE ↓	0.97	0.95	0.94	1.03	1.01	0.97	2.19	2.11	1.72	1.67	1.63	1.21	0.96	0.94	0.93
		AvgLL	38.2	44.2	39.5	41.1	46.7	44.6	17.4	17.7	16.4	28.4	32.0	31.5	54.5	53.1	50.5
		Cov	244	367	459	313	404	501	59	67	116	171	251	362	571	649	738
	Orth	Rep ↑	0.35	0.45	0.54	0.36	0.42	0.49	0.23	0.25	0.35	0.29	0.42	0.53	0.49	0.59	0.71
		LE ↓	3.62	2.67	2.83	4.41	3.75	2.91	6.78	6.44	6.05	4.28	3.62	3.09	2.29	2.03	1.93
		AE ↓	1.02	0.98	0.99	1.10	1.05	1.01	1.89	1.79	1.54	1.58	1.51	1.25	0.90	0.87	0.89
		AvgLL	38.3	44.6	39.6	41.3	46.9	44.9	17.5	18.2	16.8	28.8	32.6	31.9	54.8	53.5	50.7
		Cov	287	402	501	321	427	528	77	96	162	197	284	396	596	684	779
	LPI	33.5	31.4	33.0	27.5	26.2	26.5	35.7	35.5	38.6	28.5	25.1	26.5	22.9	23.1	23.69	23.69
	Fail%	0.00	0.00	0.00	0.00	0.00	0.00	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Most notably, IRoLSD demonstrates a significant advantage in terms of coverage, outperforming all other detectors by a large margin. This attribute speaks to the ability of our detector to uncover geometric relationships and patterns in the image and not only relying on local appearance. Fig. 8 demonstrates the above observations. The higher repeatability, coverage, and line length are easily distinguishable in the provided examples.

5.3.3. Trajectory tracking

The main goal of the line detection scheme developed is to be able to detect line segments under harsh and realistic lighting conditions in space to be used for Spacecraft Relative Navigation. We wish to evaluate the ability of the detected line segments to be accurately matched across images and to eventually produce accurate pose estimates in a realistic trajectory scenario. For this purpose, we generated trajectories at a fixed frame rate, ensuring sufficient overlap between the images. For evaluation purposes, we chose portions of the trajectories that maintain a target to camera distance close to 100 m.

To test the ability of the detected line segments to be matched, we used GlueStick [43] with SuperPoint [34] features, except for the SOLD2 [12] algorithm, which has its own line matching module. GlueStick is a deep matching Graph Neural Network (GNN) that sticks detected points and lines together in a graph, encapsulating geometric constraints between them.

For this experiment, we sampled image triads from the generated trajectories, (I_1, I_2, I_3) with known ground truth extrinsic camera parameters $T_{C_1W}, T_{C_2W}, T_{C_3W}$, respectively. We calculated the tracked correspondences of line segments between the three views of Eq. (7). Then, we triangulated the tracked line segments of the first two views into a 3D Line Map using the known camera parameters. This creates a 3D map that contains purely the line detection errors in the first views and no errors in relative pose estimation of the two views.

Subsequently, we projected the 3D line map on the third view and created pairs of matched line segments based on $\mathcal{T}(I_1, I_2, I_3)$. We initialized the pose estimates sufficiently close, allowing the final errors to reflect the errors induced purely by the line detection process in the three views.

For this experiment, we report the average number of tracked line segments, denoted as *Tracks*, which is equal to the cardinality of the set $\mathcal{T}(I_1, I_2, I_3)$ for each triad. The average line length, *AvgLL*, is computed across all views as follows

$$AvgLL = \frac{\sum_{i=1}^{T^*} (\|l_{1i}\|_2 + \|l_{2i}\|_2 + \|l_{3i}\|_2)}{3T^*}, \quad (36)$$

where T^* is the cardinality of $\mathcal{T}(I_1, I_2, I_3)$.

We report the localization error metric, *LE*, but only for the orthogonal distance case, since we do not calculate the endpoints of the projected 3D LineMap. To project the 3D LineMap we use the ground truth relative pose of the third view, so that the error only contains

Table 4

Pose estimation results on synthetic datasets: SD1 (MARS EXPRESS), SD2 (SMART1), SD3 (INTEGRAL), SD4 (XMM NEWTON).

Data	LSD				ELSEd				SOLD2				DeepLSD				IRoLSD			
	SD1	SD2	SD3	SD4	SD1	SD2	SD3	SD4	SD1	SD2	SD3	SD4	SD1	SD2	SD3	SD4	SD1	SD2	SD3	SD4
LE ↓	8.14	12.5	9.57	10.3	7.33	9.24	8.67	9.52	9.58	9.98	12.1	13.9	7.73	7.42	8.76	9.24	4.33	5.21	4.88	4.21
$e_r(m)$ ↓	2.22	7.54	5.24	5.49	2.48	6.13	5.81	4.95	8.32	7.21	8.96	8.41	2.56	5.49	6.12	4.86	0.64	1.86	0.72	0.98
$e_q(^{\circ})$ ↓	16	21	18	19	14	16	22	17	19	26	28	24	12	16	19	15	5	9	7	6
Tracks	4.6	7.3	11.4	12.1	4.1	6.5	9.1	10.5	3.4	4.7	5.6	4.9	5.1	7.4	8.9	10.2	6.2	5.4	8.5	9.7
AvgLL	24.1	17.2	28.5	32.1	38.2	32.1	37.6	41.6	14.7	12.4	19.1	21.9	28.2	31.4	32.7	36.5	74.3	87.4	55.4	67.2
Fail%	26.0	40.0	4.0	8.0	22.0	38.0	4.0	6.0	44.0	68.0	18.0	26.0	22.0	18.0	6.0	8.0	4.0	8.0	0.0	2.0

inaccuracies of the line detection from the three views. This is also the initial reprojection error that guides the pose estimation process.

Finally, we present the pose estimation errors described in Section 2.4, in translation e_r and orientation e_q , in meters and degrees, respectively, along with the metric *Fail%*. The results in Table 4 demonstrate that IRoLSD consistently detects a sufficient number of line segments. In the homographic adaptation and illumination robustness experiments, IRoLSD detects fewer line segments because it avoids regions with high intensity variance due to lighting. However, we notice that the number of tracked line segments is almost the same, indicating that the line segments detected by other algorithms would not survive through a trajectory with evolving light direction and 3D relative rotation. The pose estimation errors of IRoLSD remain consistently 2.5–5 times lower than those of the other algorithms. This means that the inherent inaccuracies of line segment detection in angle misalignment are low enough to only slightly affect the pose optimization results.

6. Conclusion

This work addresses the problem of line detection for spacecraft navigation, where challenging lighting conditions limit the performance of existing algorithms. We proposed IRoLSD, a hybrid self-supervised line detection framework that enhances general-purpose line detection methods to achieve robustness against the intricacies of the orbital environment, such as self-shadowing, sharp specular reflections, and low-contrast regions. Extensive experiments on both synthetic and laboratory-generated imagery demonstrate that IRoLSD consistently outperforms both traditional and learning-based state-of-the-art detectors across diverse conditions. In the Homogeneous Adaptation experiment, IRoLSD is consistently the most repeatable detector with improvements of the order of 20%–30% with respect to the second best performance of any model. In the Illumination Robustness experiment, IRoLSD shows an increase in the repeatability percentage of the order of 80%–100% with respect to the second-best performance of any model. IRoLSD also significantly improves the average line length of the detections by even more than 100% in some cases, which is the result of successfully mitigating the effects of line fragmentation. The superior performance demonstrated in the Homographic Adaptation and Illumination Robustness experiments eventually translates to a significant reduction in pose errors, in the range of 50%–80%. Finally, the failure rate of IRoLSD is 2.5–5 times lower than the failure rate of any model across all spacecraft. As a result, in the domain of visual spacecraft navigation, where robust feature detection is critical to the mission's success, the proposed method offers a viable and promising choice for training a generalizable line detector. Future research directions may focus on evaluating the extension of this framework to other geometric features and the combination of those in a unified SLAM pipeline. It would also be interesting to investigate the online estimation of illumination-robust regions, which is currently performed offline by generating lighting-perturbed images.

CRedit authorship contribution statement

Iason Georgios Velentzas: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Panagiotis Tsiotras:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

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 work was supported by the National Science Foundation FRR 2101250 award and by the US Air Force Office of Scientific Research, United States SURI award FA9550-23-1-0723. Iason Georgios Velentzas would like to acknowledge financial support from the Bodossaki Foundation, Greece.

References

- [1] Lorenzo Tarabini Castellani, Jesús Llorente, José Ibarz, Mercedes Ruiz, Agnes Mestreau-Garreau, Alexander Cropp, Andrea Santovincenzo, PROBA-3 mission, *Int. J. Space Sci. Eng.* 1 (4) (2013) 349–366.
- [2] Simone D'Amico, Jean-Sebastien Ardaens, Sergio De Florio, Autonomous formation flying based on GPS — PRISMA flight results, *Acta Astronaut.* 82 (1) (2013) 69–79.
- [3] Jing Pei, Matt Walsh, Carlos Roithmayr, Chris Karlgaard, Mason Peck, Luke Murchison, Preliminary GNC design for the on-orbit autonomous assembly of nanosatellite demonstration mission, in: AAS/AIAA Astrodynamics Specialist Conference, American Institute of Aeronautics and Astronautics and American Astronautical Society, Stevenson, WA, USA, 2017, NASA Technical Report NF1676L-26932, Document ID: 20170008845.
- [4] Benjamin Reed, Robert Smith, Bo Naasz, Joseph Pellegrino, Charles Bacon, The Restore-L servicing mission, in: AIAA Space Conference, AIAA, Long Beach, CA, USA, 2016.
- [5] European Space Agency, ESA Space Environment Report 2025, Technical report, European Space Agency, 2025.
- [6] Gabriella Gaias, Jean-Sébastien Ardaens, Simone D'Amico, The autonomous vision approach navigation and target identification (AVANTI) experiment: Objectives and design, in: 9th International ESA Conference on Guidance, Navigation, and Control Systems, 2014.
- [7] Mehregan Dor, Panagiotis Tsiotras, ORB-SLAM applied to spacecraft non-cooperative rendezvous, in: Space Flight Mechanics Meeting, Kissimmee, FL, USA, 2018.
- [8] Iason Georgios Velentzas, Mehregan Dor, Panagiotis Tsiotras, A robust monocular line-based relative navigation approach for spacecraft rendezvous and proximity operations, in: AIAA SCITECH 2024 Forum, 2024.
- [9] Lin Chen, Franz Rottensteiner, Christian Heipke, Feature detection and description for image matching: from hand-crafted design to deep learning, *Geo-Spatial Inf. Sci.* 24 (1) (2021) 58–74.
- [10] Rémi Pautrat, Daniel Barath, Viktor Larsson, Martin R. Oswald, Marc Pollefeys, DeepLSD: Line segment detection and refinement with deep image gradients, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Vancouver, Canada, 2023, pp. 17327–17336.
- [11] Lilian Zhang, Reinhard Koch, An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency, *J. Vis. Commun. Image Represent.* 24 (7) (2013) 794–805.
- [12] Rémi Pautrat, Juan-Ting Lin, Viktor Larsson, Martin R. Oswald, Marc Pollefeys, SOLD2: Self-supervised occlusion-aware line description and detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Virtual, 2021, pp. 11363–11373.
- [13] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, Xiaowei Zhou, Efficient LoFTR: Semi-dense local feature matching with sparse-like speed, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Seattle, WA, USA, 2024, pp. 21666–21675.
- [14] Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, Viktor Larsson, 3D line mapping revisited, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Vancouver, Canada, 2023.

- [15] Andrea Fusiello, Computer Vision: Three-dimensional Reconstruction Techniques, first ed., in: Computer Science, Springer Cham, 2024, p. XXIV+338.
- [16] Tae Ha Park, Marcus Mörtens, Mohsi Jawaaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, Simone D'Amico, Satellite pose estimation competition 2021: Results and analyses, *Acta Astronaut.* 204 (2023) 640–665.
- [17] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, Gregory Randall, LSD: a line segment detector, *Image Process. Line 2* (2012) 35–55.
- [18] Cuneyt Akinlar, Cihan Topal, Edlines: Real-time line segment detection by edge drawing (ed), in: 18th IEEE International Conference on Image Processing, Athens, Greece, 2011, pp. 2837–2840.
- [19] Iago Suárez, José M. Buenaposada, Luis Baumela, ELSEd: Enhanced line segment drawing, *Pattern Recognit.* 127 (2022) 108619.
- [20] Yongjun Zhang, Dong Wei, Yansheng Li, AG3line: Active grouping and geometry-gradient combined validation for fast line segment extraction, *Pattern Recognit.* 113 (2021) 107834.
- [21] Emilio J. Almazán, Ron Tal, Yiming Qian, James H. Elder, MCMLSD: A dynamic programming approach to line segment detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, USA, 2017, pp. 5854–5862.
- [22] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Gui-Song Xia, Liangpei Zhang, Philip H.S. Torr, Holistically-attracted wireframe parsing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Virtual, 2020, pp. 2785–2794.
- [23] Yichao Zhou, Haozhi Qi, Yi Ma, End-to-end wireframe parsing, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, Seoul, Korea, 2019.
- [24] Ziheng Zhang, Zhengxin Li, Ning Bi, Jia Zheng, Jinlei Wang, Kun Huang, Weixin Luo, Yanyu Xu, Shenghua Gao, PPGNet: Learning point-pair graph for line segment detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Long Beach, CA, USA, 2019, pp. 7098–7107.
- [25] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, Liangpei Zhang, Learning attraction field representation for robust line segment detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Long Beach, CA, USA, 2019, pp. 1595–1603.
- [26] Shuhan Huang, Feipeng Qin, Peng Xiong, Ning Ding, Yulan He, Xin Liu, TP-LSD: Tri-points based line segment detector, in: Andrea Vedaldi, Horst Bischof, Thomas Brox, Jan-Michael Frahm (Eds.), European Conference on Computer Vision, ECCV, in: Lecture Notes in Computer Science, 12372, Springer, Cham, Glasgow, Scotland, 2020.
- [27] Geonmo Gu, Byungsoo Ko, SeoungHyun Go, Sung-Hyun Lee, Jingeun Lee, Minchul Shin, Towards light-weight and real-time line segment detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2022.
- [28] Yifan Xu, Weijian Xu, David Cheung, Zhuowen Tu, Line segment detection using transformers without edges, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Virtual, 2021, pp. 4255–4264.
- [29] Sumant Sharma, Jacopo Ventura, Simone D'Amico, Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous, *J. Spacecr. Rockets* 55 (6) (2018) 1414–1429.
- [30] Michele Bechini, Geonmo Gu, Paolo Lunghi, Michèle Lavagna, Robust spacecraft relative pose estimation via CNN-aided line segments detection in monocular images, *Acta Astronaut.* 215 (2024) 20–43.
- [31] Xingxing Zhang Changyu, Hu, Hanhan Liu, Ronghua Du, Xiaofeng Zhou, Ling Wang, A line segment detector for space target images robust to complex illumination, *Aerospace* 10 (2) (2023).
- [32] Nobuyuki Otsu, et al., A threshold selection method from gray-level histograms, *Automatica* 11 (285–296) (1975) 23–27.
- [33] Junqiu Wang, Jingjing Cui, Yanjie Gao, Yanxuan Ma, Zhichao Gan, Chao Yang, Efficient pose tracking based on line segment matching, *Am. J. Eng. Appl. Sci.* 12 (2019) 1–9.
- [34] Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich, Superpoint: Self-supervised interest point detection and description, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, Salt Lake City, UT, USA, 2018, pp. 224–236.
- [35] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention, MICCAI, in: LNCS, vol. 9351, Springer, Munich, Germany, 2015, pp. 234–241.
- [36] Iason Georgios Velentzas, Juan-Diego Florez, Noah Bruckner, Mehregan Dor, Panagiotis Tsiotras, SISIFOS: Specialized illumination simulator for orbiting spacecraft, in: AIAA SciTech, Orlando, FL, 2026.
- [37] Hoang Anh Dung, Bo Chen, Tat-Jun Chin, A spacecraft dataset for detection, segmentation and parts recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, Virtual, 2021, pp. 2012–2019.
- [38] Bo Chen, Jiewei Cao, Alvaro Parra, Tat-Jun Chin, Satellite pose estimation with deep landmark regression and nonlinear pose refinement, in: IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, Seoul, Korea, 2019, pp. 2816–2824.
- [39] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, Hengshuang Zhao, Depth Anything V2, in: A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems, vol. 37, Curran Associates, Inc., 2024, pp. 21875–21911.
- [40] European Space Agency, ESA | Science Satellite Fleet 2.0.0, 2025, <https://scifleet.esa.int/>, (Accessed 26 May 2025).
- [41] Tae Ha Park, Marcus Mörtens, Gurvan Lecuyer, Dario Izzo, Simone D'Amico, SPEED+: Next-generation dataset for spacecraft pose estimation across domain gap, in: IEEE Aerospace Conference, AERO, Big Sky, MT, USA, 2022, pp. 1–15.
- [42] Panagiotis Tsiotras, ASTROS: A 5-DOF experimental facility for research in space proximity operations, *Adv. Astronaut. Sci.* 151 (2014) 717–730.
- [43] Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, Viktor Larsson, GlueStick: Robust image matching by sticking points and lines together, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, Paris, France, 2023.