AAS 26-013

TACKLING TARGET MODEL UNCERTAINTIES IN VISION-BASED RELATIVE POSE ESTIMATION

Giacomo Battaglia*, Anthea Comellini†, Francesco Capolupo‡ and Paolo Panicucci§

Close-proximity operations with a non-cooperative target represent a challenging scenario for the Guidance, Navigation, and Control subsystem. Indeed, as a free-drifting and tumbling target does not provide any information about its state, monocular vision-based navigation can only rely on images to determine the relative pose. Despite state-of-the-art assumptions, only a coarse target model can be available onboard as estimated by dedicated observations or by prelaunch shape models. Therefore, the target visual and dynamical properties are uncertain during real operations. This directly impacts image processing and filter designs. In this article, we propose an AI-aided vision-based navigation solution that accounts for these uncertainties in the design process. First, we introduce a novel synthetic dataset featuring a varying-shape target with moving appendages and shape model perturbations. Second, we train a keypoint-regression network with a dedicated visibility-based conditional labeling strategy to tackle shape variations. This approach guides learning toward robust local features, rather than global model matching. Finally, we develop a Schmidt-Kalman filter, including the center of mass and inertia uncertainties as consider parameters. We evaluate the algorithm in a representative orbital scenario and compare multiple configurations within a Monte Carlo framework, showing performance improvements from including those uncertainties in the design.

INTRODUCTION

In Orbit Servicing (IOS), including Active Debris Removal (ADR), refers to the broad concepts of operations performed on orbit between two spacecraft ranging from repairing to refueling, manipulation using robotic arms, capturing and de-orbiting. During these operations, a chaser vehicle approaches a target space object (e.g., an active satellite or a debris) to act in its close proximity. IOS has recently reached a position of prominence, responding to the need for a more sustainable space environment, aggravated by the continuous increase of the space debris population. Indeed, the space debris population is bound to grow over time, potentially rendering entire orbital areas inaccessible in the near future. Therefore, de-orbiting current space debris with dedicated missions and repairing malfunctioning units with in-orbit probes would decrease the number of inactive space objects, increase the lifetime of existing ones, and limit the growth of existing space debris population.

IOS missions usually culminate in a critical close-proximity operations phase. To ensure safety during this phase, the chaser's onboard guidance, navigation and control subsystem must ensure six-degrees-of-freedom relative control with respect to the target. In this context, the role of navigation is to determine the pose (i.e., position and orientation) of the chaser with respect to the target, as long as the pose derivatives when possible. The most challenging scenario foresees a non-cooperative target, where it does not support the chaser with any information about its absolute state. Therefore, navigation must exploit exteroceptor sensors to perceive the external environment and the relative state of the target. Sensor information is processed by dedicated algorithms to extract high-level contextual understanding of changes in the environment. Finally, this contextual belief is processed by onboard smoothing over time with dedicated navigation filtering techniques to

*PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, giacomo.battaglia@polimi.it
†AOCS/GNC Engineer, Thales Alenia Space, anthea.comellini@thalesaleniaspace.com
‡Guidance, Navigation and Control Systems Engineer, European Space Agency, francesco.capolupo@esa.int
§Assistant Professor, Department of Aerospace Science and Technology, Politecnico di Milano, paolo.panicucci@polimi.it

provide a dynamically-consistent pose estimation, along with its derivatives. All of these tasks must run on space-proven processors with limited memory and computational power, imposing a harsh trade-off between performance, robustness, latency, and embeddability. Among all the sensors available on the market to perform close-proximity operations, passive cameras are usually the preferred ones because they barely affect mass, power, and volume budgets during the spacecraft design. The sensor processing algorithm that aims to determine the relative pose from images is known as vision-based navigation.

Despite being a classical assumption in IOS applications (especially for space debris) the non-cooperative target's shape is often not be perfectly known a priori. Observation from ground can provide a coarse model of the target before the mission, and this might be estimated or refined during the inspection phase. In essence, only a coarse shape model is available before close approach, leading to considerable uncertainties on dynamical properties (e.g., center of mass and inertia matrix) exploited by the navigation filter, as well as the shape model used in image processing training. In addition, moving appendages (e.g., solar panels) might be oriented in a different configuration with respect to the reference shape model, directly impacting the shape and associated properties. Recall that even when the shape is certain, variation in mass distribution (e.g., outgassing or sloshing) might directly affect the target barycenter and inertia estimates. All these uncertainties must be considered within the navigation filter design as it might cause navigation failure due to inaccurate filter propagation and update. Similarly to the filtering step, image processing is also affected by shape uncertainty. The image processing is in charge of extracting contextual information (i.e., the pose) from the images. Among different methodologies, AI-aided image processing exploits the abstraction capability of neural networks to improve feature extraction compared to handcrafted-based methods. Their application for spacecraft pose estimation has recently become state of the art[1] due to its higher accuracy and robustness to harsh illumination conditions. Model uncertainties could have a consistent impact on the performance of state-of-the-art AI-based image processing. Indeed, the neural network might be trained on a wrong reference shape model, implying a fatal domain gap when deployed on orbit. To counteract this limitation, uncertainties shall be considered within network training. In conclusion, AI-based image processing shall be trained to generalize to unseen variations in the target shape, while the navigation filter shall account for the target dynamical uncertainties via dedicated modeling in the dynamics and measurement functions.

This work designs a vision-based navigation architecture that accounts for target shape-related dynamical uncertainties throughout the design process. The image processing network is trained on a novel dataset, composed of varying-shape targets with a randomly moving solar panel. The network is trained for keypoint regression with a supervised approach, where conditional labeling accounts for model variations in the ground truth. Furthermore, we choose the network architecture to comply with the strict deployability constraints of space hardware, limiting the size and computational complexity of the design. Finally, we propose an Extended Kalman Filter[2] (EKF), with an additive architecture for the translational state, while we exploit a multiplicative approach (i.e. MEKF[2]) for the rotational one. Uncertainties in the target dynamical properties (i.e., center of mass (CoM) and inertia) are modeled as consider parameters in a Schmidt-Kalman framework.[3] Finally, we test the entire navigation pipeline in a proximity operations scenario with Monte Carlo simulations.

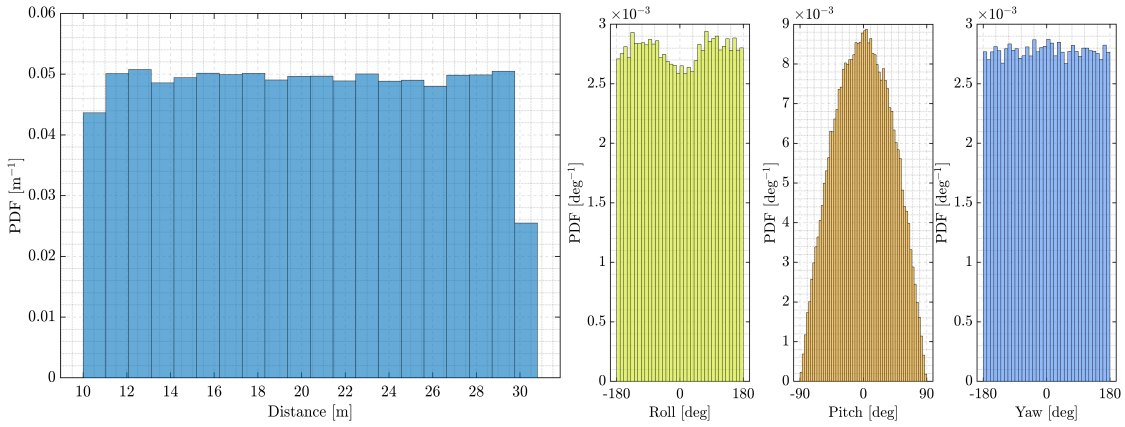
This article is organized as follows. The section "Dataset" describes the novel synthetic dataset, concentrating on our novel shape scattering pipeline. The section "AI-based Image Processing Design" outlines the AI-aided image processing design, introducing a visibility concept for supervised keypoint labeling, and comparing performance of different labeling methods. The section "Navigation filter" provides a brief description of the navigation filter, focusing on uncertainty modeling in the Schmidt-Kalman design. The section "Results" presents preliminary Monte Carlo results of the full navigation chain in a proximity operations scenario. Finally, the section "Conclusion" resumes the results and draws the final considerations from this study.

DATASET

This section describes the design of our novel dataset, featuring a varying-shape target to improve neural network generalization towards unseen perturbations in the target's shape. Crucially, the goal of this design is to increase the robustness of the neural network towards a partially known target, both locally and globally.

Recently, many datasets have emerged to tackle AI training for vision-based navigation in close proximity operations. These datasets are usually designed to train pose estimation networks to specialize in a unique target model, with a model-matching approach. In this context, the network learns to regress a fixed target shape and its projection under challenging illumination conditions. Many examples of these datasets exist. A remarkable one is the SPEED+ dataset,[4] featuring synthetic and lab-taken images of the Tango spacecraft,[5] to assess the robustness of neural networks to the domain gap between real and synthetic images. Many other datasets exist, either fully-synthetic (e.g., the SPEED dataset[6] or Gallet et al.[7]) or with synthetic and lab-taken images (e.g., Lebreton et al.[8]).

This study aims to assess the capability of neural networks to generalize towards variations in the target shape, due to varying configurations of moving appendages or uncertainties in the onboard target model. Therefore, we propose a novel fully synthetic dataset featuring a single target with varying shape and solar panel shape and orientation. This fully synthetic dataset is generated using the SpiCam rendering tool, developed by Thales Alenia Space.[7] SpiCam is a high-fidelity rendering tool specialized in spacecraft image generation. It performs physics-based rendering using energy conservative BRDF (Bidirectional Reflectance Distribution Function), computing multi-spectral radiometric quantities based on sensor efficiency values. SpiCam models all the phases of the detection process, namely the conversion of light fluxes into electrons and then into quantized bits, taking into account all related disturbances. This dataset is composed of 100'000



(a) Distribution in relative distances.

(b) Distribution in relative attitudes.

Figure 1: Pose scattering statistics in the dataset.

images with 1024×1024 resolution. They are split into 80'000 images for training and validation and 20'000 images for testing. It features a base target spacecraft model inspired by Sentinel-3.[9] This shape model has one solar panel free to rotate 360 degrees around its axis. The dataset includes uniform scattering in orbital, pose, environmental parameters, as well as in the target model. Orbital and pose parameters are scattered in the following way:

1. The absolute position is uniformly scattered in an altitude range between approximately 100 km and 950 km, specializing the dataset for a LEO (Low Earth Orbit) application

2. The relative position of the target with respect to the chaser is also uniformly scattered in terms of direction, using uniform rotation sampling,[10] and in terms of relative distance, between 10 and 30 m

3. The relative attitude between the target and the chaser is also scattered using uniform rotations[10]

4. The position of the target in the image plane, positioning the target model origin uniformly in a square of 512×512 size around the image center

Figures 1a and 1b show respectively the scattering statistics of the dataset for relative distance and relative attitude, while Figure 3 provides heatmaps of the appearance of the target on the dataset images. Furthermore, the scattering process controls environmental factors such as illumination conditions and the presence of the Earth and the Moon in the background. In detail:

1. Illumination variability is achieved by uniformly scattering the Sun direction by controlling the Sun phase angle, with a maximum of 75 degrees.

2. Earth is always placed at the inertial frame origin, Earth's attitude and Moon's position are scattered from their ephemeris by date sampling.

Figure 2 shows the occurrence statistics for each class in the dataset. It appears how the Earth is present in the background in approximately half of the images, typical of a LEO case. The main novelty of this dataset

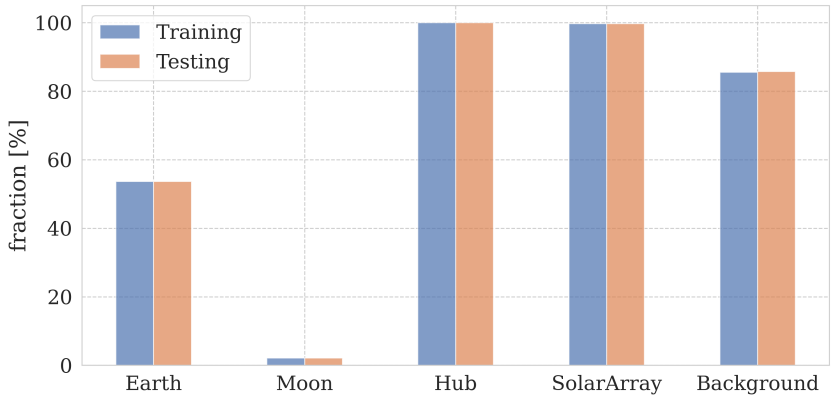
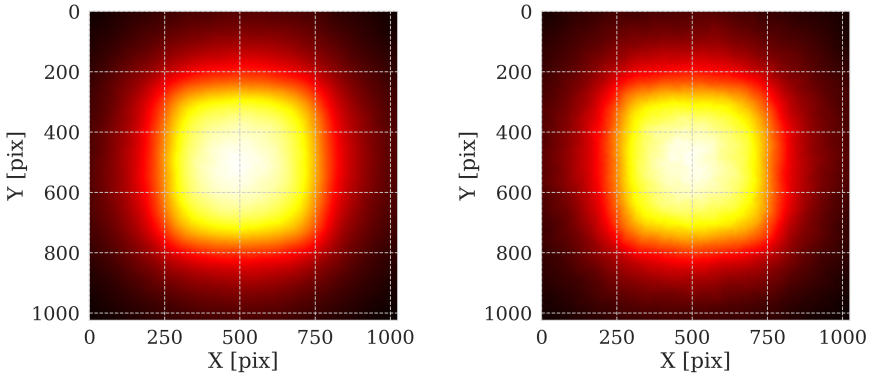


Figure 2: Statistics of per-class appearances in the training/validation and testing splits.



(a) Training/validation split.

(b) Testing split.

Figure 3: Class density distribution heatmap for the training/validation and testing splits on the image plane.

lies in the variability of the target shape. Along with the aforementioned parameters, we scatter not only the optical properties of the target (e.g., albedo and roughness) and its multi-layer insulation textures but also its shape. Our dataset features an articulated part, namely the solar panel randomly rotated around its axis. Also,

Algorithm 1 Remove vertices and corresponding triangles from the mesh.

Require: Mesh with vertices $\mathbf{v} \in V$ and faces $f \in F$, number of vertices to remove n

1: Sample base vertex $\mathbf{v}^* \in V$ in the mesh.
2: Find $\mathbf{v} \in V_{subset}$, i.e., the n closest vertices to $\mathbf{v}^*$ using the euclidean distance: $\|\mathbf{v} - \mathbf{v}^*\|_2$.
3: Remove the vertices $\mathbf{v} \in V_{subset}$ and all faces $f \in F$ s.t. $\mathbf{v} \in V_{subset} \wedge \mathbf{v} \in f$.
4: Remove additional vertices that are no longer connected to any faces.
5: Remap vertex and face indices.

return New perturbed mesh with V' vertices and F' faces.

we introduce perturbations in the actual target shape model, by directly performing scattering on its mesh elements in the rendering phase.

Algorithm 2 Move vertices of the mesh using a spherical harmonic field.

Require: Mesh with vertices $\mathbf{v} \in V$ and faces $f \in F$, number of vertices to remove n, number of terms p of the spherical harmonic field, interval of shape coefficients $[l, u]$, $0 \leq l, u \leq 1$

1: Sample base vertex $\mathbf{v}^*$ in the subset.
2: Find $\mathbf{v} \in V_{subset}$, i.e., the n closest vertices to $\mathbf{v}^*$ using the euclidean distance: $\|\mathbf{v} - \mathbf{v}^*\|_2$.
3: Compute a spherical harmonic field g of order p with random coefficients:

$$g(\theta, \phi) = \Re \left[\sum_{l=0}^{p-1} \sum_{m=-l}^{l} c_{l,m} \, L_l^{(m)}(\phi) \, e^{im\theta} \right] \;,\; c_{l,m} \sim \mathcal{N}(0, 1)$$

where $L_l^{(m)}$ is computed using the Legendre polynomial $P_l(x)$:

$$L_l^{(m)}(\phi) = \begin{cases} (-1)^m \, \Im\big[P_l^{(|m|)}(\cos\phi)\big], & m < 0, \\ P_l^{(0)}(\cos\phi), & m = 0, \\ \Re\big[P_l^{(m)}(\cos\phi)\big], & m > 0. \end{cases}$$

4: Express the vertices $\mathbf{v} \in V$ in spherical coordinates $\mathbf{v} = (r, \theta, \phi) \in V$. Then normalize the scalar field g based on its maximum and minimum values on the mesh, and rescale it inside the l, u boundaries:

$$\bar{g}(\theta', \phi') = u + (l - u) \frac{g(\theta', \phi') - \min_{\theta,\phi \in V} f}{\max_{\theta,\phi \in V} f - \min_{\theta,\phi \in V} f}$$

5: Apply the normalized coefficient to the vertices in V_{subset}:

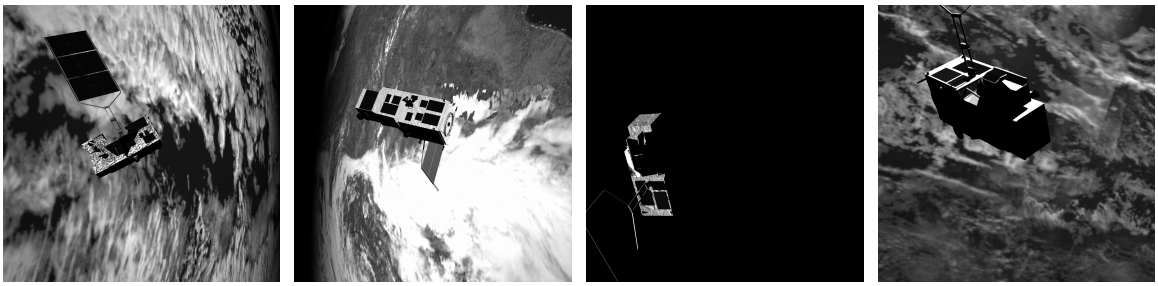
$$\text{if } \mathbf{v} \in V_{subset} \text{ , then } \mathbf{v} \to \mathbf{v}_{new} = \begin{pmatrix} \bar{g}(\theta, \phi) r \\ \theta \\ \phi \end{pmatrix}$$

return Perturbed mesh with V' vertices and F' faces.

To scatter the target shape, we generate a number of perturbed models starting from the base mesh. Perturbations are applied to connected subsets of the mesh triangles, and they are of two types: removal, where the subset is deleted, and deformation, where the subset vertices are moved using a spherical harmonic field. Algorithms 1 and 2 show the two procedures for these two perturbations. These perturbations are applied locally multiple times both on the hub and the solar panel, by sampling the origin point and the size of the subset. The level of "destruction" of the shape is regulated by choosing how many times to apply mesh changes and the number of vertices impacted. In this manner, we generated 100 different perturbed models

for the hub and the solar panel and randomly combined them during image generation. In conclusion, 80 models are used in the training/validation split and 20 models in the testing split.

Finally, to clean up the dataset, we designed an image quality check pipeline to reject outlier images. Indeed, unfortunate illumination conditions might generate images in which the target is not actually visible. Those images represent actual outliers in the dataset, which may pollute the final performance of the trained models. Hence, the image checking pipeline verifies whether an image is compliant with intensity, gradient and entropy thresholds and discards unsatisfying configurations. For completeness, Figure 4 shows some sample images taken from the dataset.



(a) Part of the hub missing, Earth background.

(b) Part of the panel missing, Earth background.

(c) Part of the hub missing, low illumination.

(d) Panel fully out of the image, Earth background.

Figure 4: Some sample images taken from the dataset.

AI-BASED IMAGE PROCESSING DESIGN

The proposed dataset represents a new challenge for AI-based image processing, as state-of-the-art approaches are trained to identify keypoints starting from the whole shape. The variability of the shape model adds new degrees of freedom to the learning problem. On top of this, network size is usually limited by hardware constraints for space applications. In this section, we describe the approach to this neural network design, trying to harmonize the need for a robust solution with low computational resources.

The image processing task is ultimately a pose estimation problem: regressing the relative position and attitude of the target with respect to the camera using in-orbit taken images. Two main approaches exist for this task:[1] direct and indirect methods. Direct methods infer the pose from the image in a single network. They are usually less explainable and more computationally demanding. However, some viable solutions exist for space applications, such as those derived from CenterPose[11] or EfficientPose[12] architectures. The most relevant example is SPN (Spacecraft Pose Network)[13] for IOS. These approaches also see some use in unknown target pose (and shape) estimation.[14] However, accuracy of these methods is usually insufficient for critical close-range navigation, as they tend to be less accurate than geometrical PnP calculations. On the other hand, indirect methods divide the pose estimation task into a set of subtasks, which are simpler and less computationally demanding. First, when distance validity requirements are higher, some works insert a spacecraft detection task, where a dedicated algorithm (e.g. 15, 16) finds a region of interest containing the target in the image. This step may be omitted to tackle the final close-range phases to envision deployability on space-qualified hardware with a lighter solution. The main task of the indirect design is landmark regression. Landmark regression consists of finding a pre-defined set of keypoints, matching them to a known 3D wireframe. This enables determination of 2D-3D correspondences that can feed a Perspective-n-Point (PnP) algorithm, for instance, EPnP.[17] Landmark regression is usually done as heatmap regression, meaning that the network is trained on reproducing a set of activation maps modeled as Gaussian distributions with the mean on the actual keypoint and a certain variance (tuned as a hyperparameter). The loss is usually the mean squared error (MSE). Network size requirements for indirect methods are usually less stringent than for direct ones. Many different architectures are possible. MobileNet-based[18,19] architectures represent the baseline for simplicity and deployability.[20] However, the EfficientNet backbone[21] also shows good performance

on benchmark datasets, especially in combination with a feature pyramid network as BiFPN[22] for multi-scale feature fusion.[23] Other approaches implement transformer-based architectures[24] to achieve global context understanding by the self-attention mechanism. However, transformer models are usually more complex and computationally demanding, with respect to convolutional neural networks, to be robustly embedded in current hardware. In conclusion, although direct methods hold promise, indirect methods are usually more accurate and robust, thanks to tasks separation, better explainability, and the PnP solid mathematical background. However, the performance of these methods depends on the accuracy of the target shape model. This work aims exactly to tackle this point by assessing the robustness of indirect method design to target shape variation. The proposed image processing solution is composed of a heatmap regression network that feeds

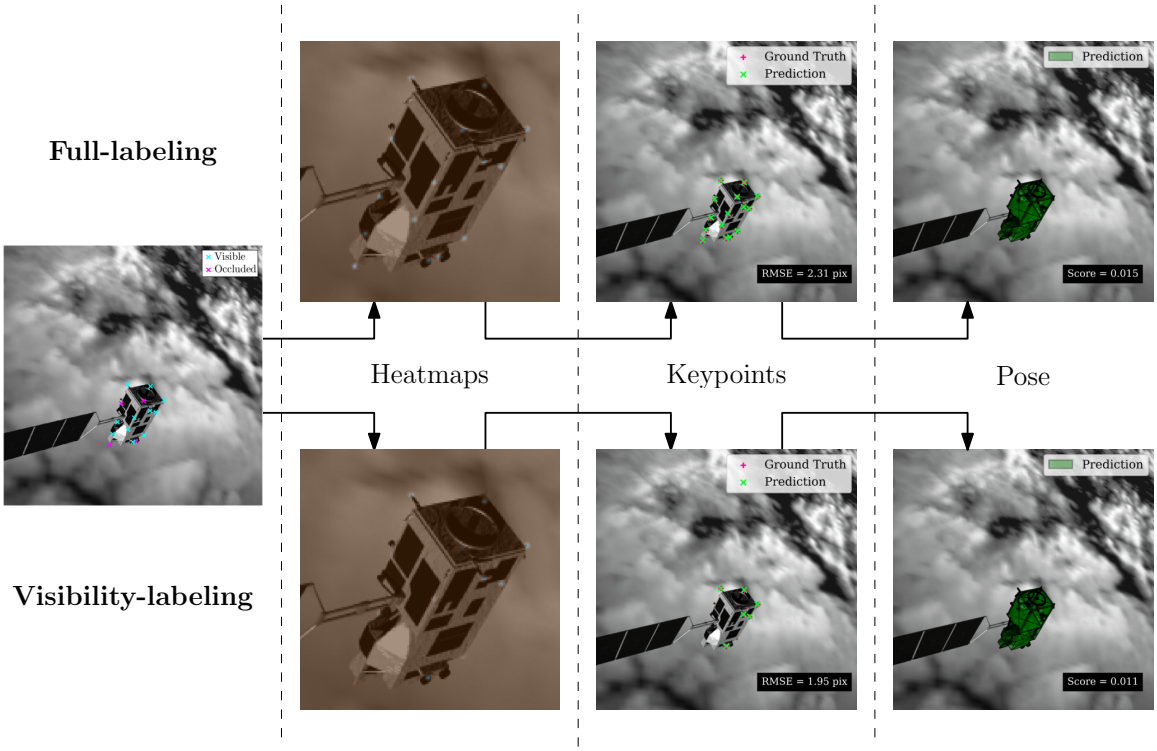


Figure 5: Example of the inference process using a full-labeling and a visibility-labeling network. The pose score is the SPEED score.[6]

a PnP bulked with RANSAC (RANdom SAmple Consensus) for outlier rejection. The network is trained on the above-described dataset, aiming to network generalization with respect to moving appendages and mesh perturbations. However, tiny neural networks as the ones used for onboard space implementations have limited generalization capabilities as they can only learn limited representations. To deal with this tradeoff, it is important to reduce the complexity of the learned features, implying that the chosen keypoints should mostly remain constant with the expected shape model changes. For these reasons, no keypoints are selected on the solar panel, as its free movement coupled with its inherent symmetry would impact the network matching performance. Hence, all keypoints are chosen in the hub which is also perturbed by the mesh modification pipeline. This implies that not all keypoints are present in the modified model with respect to the base one. Moreover, if present, they might be displaced or distorted by the deformation pipeline. Therefore, the network could tackle this problem in a supervised learning framework by tailoring the labeling process. To do this, we introduce the concept of keypoint visibility: each keypoint is associated with a feature blob, comprising a set of vertices and faces of the mesh. Using rendering and pose information, the keypoint is flagged as visible if its corresponding faces on the mesh appear on the image. Otherwise, it is considered cluttered. Note that the keypoint is removed if its corresponding subset was removed in the shape perturbation process. A set of fifteen landmarks is chosen on the model, covering salient features and corners. We implement a semi-

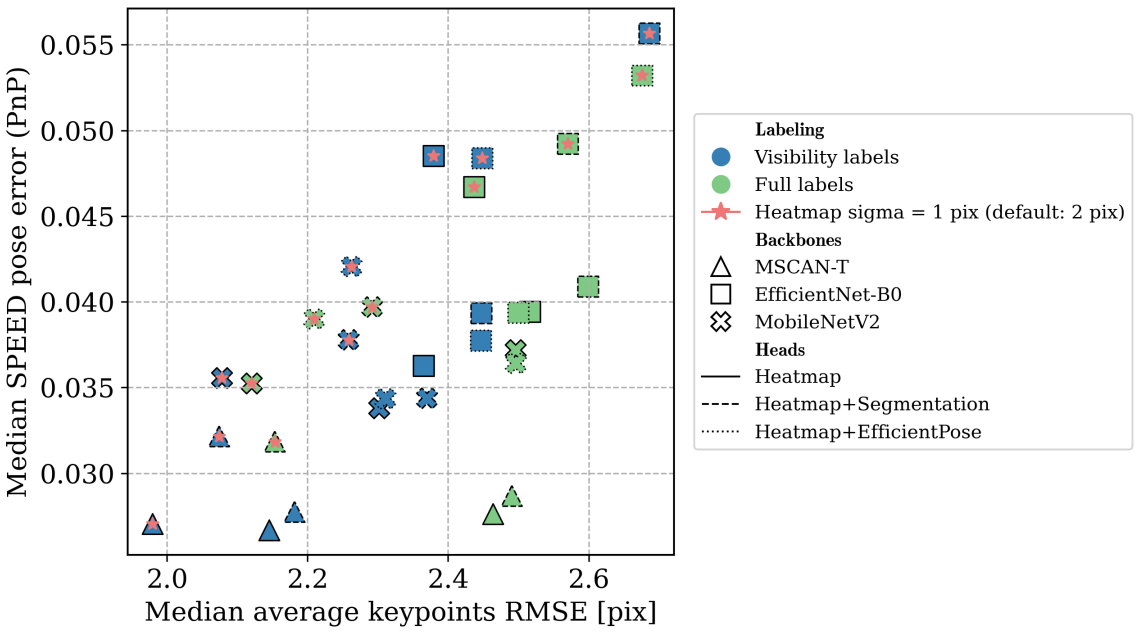


Figure 6: Median values for keypoint root mean squared error and SPEED pose error[6] for the first batch of neural networks.

automated keypoint-choice process. First, given the target mesh, corners are identified to remove redundant vertices. In this context, a corner vertex is defined as a vertex where the relative angle between its linked faces is lower than a certain threshold. Then, a subset of corners is identified using farthest point sampling, to have a uniform distribution of points on the body, starting from an anchor point chosen manually. Finally, those points are then filtered by looking at the activation statistics of the output layer of a pretrained EfficientDet[22] backbone in a small helper fixed-shape image dataset, to obtain the requested number of points.

With these chosen landmarks, we train a number of neural networks with different configurations. First, models are trained from scratch on a slightly simpler version of the dataset presented in this article where randomized models are not randomly coupled but they are sorted to have a tenth of the dataset for each pair (hub and solar array). The aim of this first analysis is to extract a subset of top performing models and configurations to then finetune on the fully randomized dataset. The baseline architecture design follows a typical computer vision approach that stems from the SPNv2[23] archetype. Three different backbones of similar sizes are chosen, such as MobileNetV2,[19] EfficientNet-B0[21] and MSCAN-T.[25] In addition to the two former more classical backbones, MSCAN-T[25] is selected for its light-weight attention-based architecture featuring multi-scale convolutional attention. These backbones are combined with a BiFPN[22] neck for state-of-the-art multi-scale feature fusion. Three types of head configurations are considered, such as: heatmap-only head, heatmap-segmentation head and heatmap-pose head (using the EfficientPose[12] design). All models take 512×512 pixel images, resized from 1024×1024, and output 512×512 heatmap activation maps or segmentation mask. We introduce two types of labeling, using the visibility concept: a less restrictive "full-labeling" approach which keeps all keypoints present in the employed mesh of the image (note: the perturbed mesh might have some landmarks removed) and a stricter "visibility-labeling" approach, which also removes keypoints which are not present on the image because they are occluded by the pose configuration. These two labeling approaches represent two different training goals. The "full-labeling" approach orients training towards model matching, aiming for a global interpolation of the target model in the image. This results in extracting keypoints from images even when the intensity content is consistent with image noise using global shape information. Conversely, the "visibility-labeling" approach tries to move towards local feature understanding, detaching from the global fixed model. In principle, this could help the network to be more

robust to shape variations, as the regression process would be less confident towards a fixed shape and more attached to local contexts. Figure 5 compares the two methods in an example image. Finally, networks are also trained with different heatmap label sizes (i.e., with a standard deviation of 2 pixels or 1 pixel) to understand the performance when changing this hyperparameter. To assess the performance of the entire

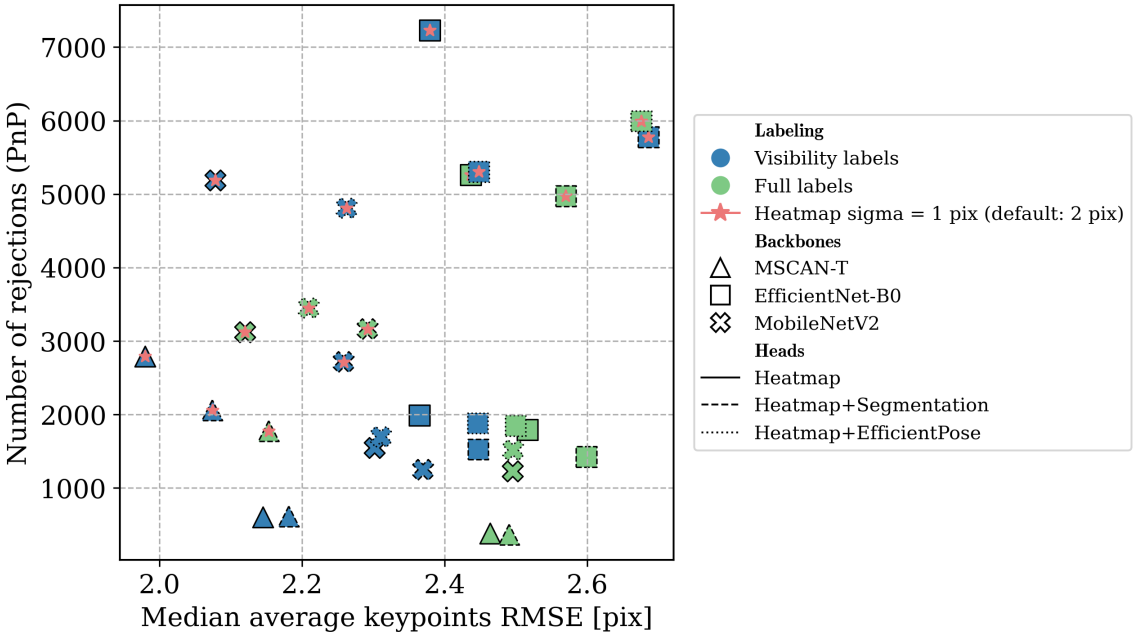


Figure 7: Median values for keypoint root mean squared error and number of rejections for the first batch of neural networks.

set of trained networks obtained by combining the aforementioned architectures, configurations, and training approaches, we consider the mean and median keypoint pixel error, the SPEED pose error,[6] and the number of keypoints detected per image as the main performance indicators. The use of both mean and median aims to understand the skewness of the error distribution, getting an insight on the number of outlier predictions generated by the network. Figure 6 shows the median values for pose error and keypoint root mean square error (RMSE) in the testing set, and Figure 7 also shows the number of rejections for PnP computation (i.e. the number of images where less than five points are available for PnP). From these two types of results, keypoint median accuracies are comparable. However, it can be observed that 1-pixel heatmap labeling typically enables slightly more accurate keypoint inference, but results in a less robust solution due to higher rejection rates, indicating a more challenging task. Moreover, full-labeling and visibility-labeling have comparable median results at this stage, with a slight edge by the latter for precision and the opposite for number of rejections. Finally, concerning network architectures, the MSCAN-T^{25} design shows the best performance in all configurations, while head configurations provide comparable performance in the keypoint regression task.

Secondly, we fine-tune the best-performing models of the first analysis on the final dataset to prune out other candidates. Table 1 reports the mean and median performance in the test set for the best-performing models, providing insight into the differences between the two labeling strategies. The full-labeling networks achieve higher accuracy in end-to-end pose estimation when combined with PnP and RANSAC-based outlier rejection. However, by comparing mean and median values, visibility-labeling designs have a much less skewed keypoint error distribution. In some way, visibility-labeling networks embed a sort of rejection process, as they are less prone to making guesses on keypoint positions unsupported by direct image-related cues. In general, they find fewer keypoints than full-labeling methods, but achieve higher accuracy. This might provide insight on the selection of the training process for the keypoint extractor, also in vision of

the IP-filter interface. In this perspective, we may consider either a loosely or a tightly coupled design.[26] The former defers full pose computation to the image processing, using that as a high-level measurement in the filter. The latter directly provides keypoints as measurements to the filter, to better exploit position and rotation correlations within the estimation process. While full-labeling methods may be more suitable for a loosely coupled design owing to their good synergy with PnP, visibility-labeling might be exploited in a tightly coupled approach thanks to its less skewed keypoint error distribution (thus fulfilling Kalman filtering Gaussianity hypothesis). To investigate this assumption, the networks with both labeling procedures are analyzed in a Monte Carlo campaign with a tightly coupled approach in Section "Results".

Table 1: Neural network performance results after fine-tuning.

Head	Labeling	Heatmap size [pix]	Pretrain heatmap size [pix]	Keypoints RMSE	Rejections (PnP)	Position error [m]	Rotation error [deg]	SPEED score[6]	Segmentation mIoU
Median values									
H	Full	2	2	2.121	28	0.072	0.685	0.0167	
		1	2	2.059	103	0.070	0.654	0.0159	
	Visibility	2	2	2.267	434	0.120	1.098	0.0268	
		1	2	2.105	501	0.108	0.983	0.0241	
		1	1	2.209	975	0.128	1.145	0.0283	
H+S	Full	2	2	2.171	41	0.077	0.721	0.0176	0.907
	Visibility	2	2	2.223	329	0.116	1.043	0.0257	0.910
		1	2	2.211	643	0.123	1.109	0.0271	0.928
		1	1	2.252	1339	0.144	1.340	0.0327	0.928
Mean values									
H	Full	2	2	15.917	28	0.136	1.025	0.0247	
		1	2	15.078	103	0.138	1.049	0.0254	
	Visibility	2	2	2.645	434	0.251	1.955	0.0471	
		1	2	2.454	501	0.283	2.299	0.0553	
		1	1	2.694	975	0.346	2.782	0.0669	
H+S	Full	2	2	15.521	41	0.143	1.051	0.0254	0.894
	Visibility	2	2	2.651	329	0.297	2.097	0.0541	0.896
		1	2	2.554	643	0.357	2.369	0.0626	0.917
		1	1	2.637	1339	0.413	3.242	0.0770	0.916

NAVIGATION FILTER

This section describes the second part of the pose estimation pipeline: the navigation filter. The filtering function is crucial in the VBN design as it enables to refine the estimated pose in time, exploiting dynamics-related priors of the close proximity operations scenario. Moreover, navigation filter complements pose refinement with estimates of the pose derivatives, inaccessible otherwise. The solution we propose is an EKF^2 with an additive architecture for translational dynamics and a multiplicative approach for attitude (i.e., a MEKF^2) to better represent each variable in the correct manifolds. The choice of an Extended filter with respect to an Unscented version[27,28] is mainly driven by its overall lower computational burden, which is a crucial requirement for close-proximity IOS applications. Most importantly, the main contribution of this work lies in the inclusion of target model uncertainty in the navigation filter design through a Schmidt–Kalman approach,[3] ensuring filter consistency under variations in mass, CoM, and inertia.

The general propagation equations between the instants t_k and t_{k+1} for the discrete-time EKF^2 are the

following:

$$\begin{cases} \mathbf{x}_{k+1}^{-} = f_k(\mathbf{x}_k^{+}, \mathbf{u}_k, t_k) \\ \mathbf{P}_{k+1}^{-} = \mathbf{\Phi}\left(t_{k+1}, t_k\right) \mathbf{P}_k^{+} \left(\mathbf{\Phi}\left(t_{k+1}, t_k\right)\right)^T + \mathbf{Q}_k \end{cases} \tag{1}$$

where $\mathbf{x}$ is the state, $\mathbf{P}_k$ is the state covariance at time t_k, $\mathbf{\Phi}$ is the state transition matrix and $\mathbf{Q}_k$ is the discrete-time process noise matrix, which may be computed from the continuous-time matrix using analytical or numerical integration.[2] The update equations at time t_k are the following:

$$\begin{cases} \mathbf{x}_k^{+} = \mathbf{x}_k^{-} + \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_k\right) \\ \hat{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k^{-}) \\ \mathbf{P}_k^{+} = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_k^{-} \\ \mathbf{K}_k = \mathbf{P}_k^{-} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \end{cases} \tag{2}$$

where $\mathbf{z}$ is the measurement, $\hat{\mathbf{z}}$ is the estimated measurement through the measurement model function $\mathbf{h}$, $\mathbf{H}$ is the Jacobian of the measurement function, $\mathbf{K}$ is the Kalman gain, and $\mathbf{R}$ is the measurement noise covariance.

These equations are valid for both the additive and the multiplicative design. However, the latter replaces linear operations with rotation compositions using the chosen representation. For our case, the Modified Rodriguez Parameters (MRP) are used, implying that summation and difference are performed in the MRP space. It is worth noting that the MEKF propagates and updates the MRP error with respect to a reference MRP. Therefore, the MEKF features a reset step, where the MRP error state is set to zero, when updating the reference MRP with the estimated MRP error. This prevents the rotational state from encountering singularities and keeps its value close to zero, thereby easing the validity of linearization.

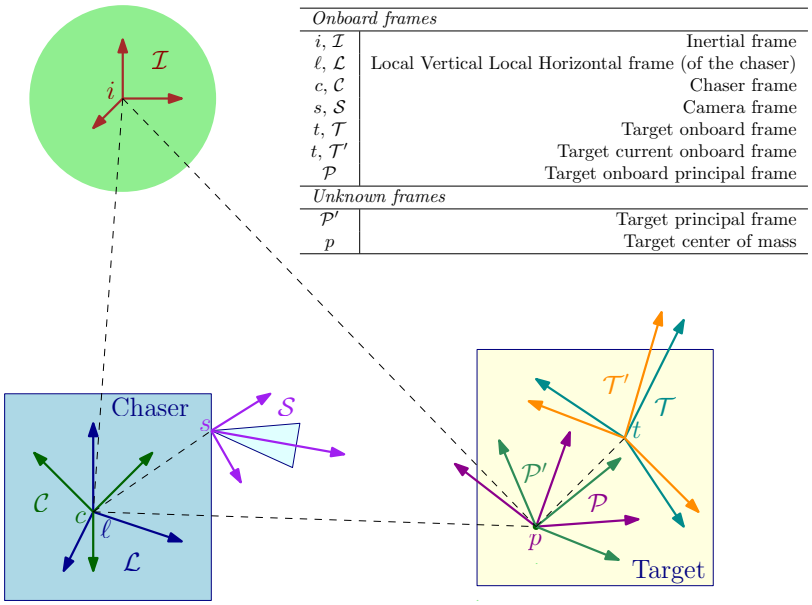
Furthermore, we exploit a Schmidt-Kalman architecture,[3] to include uncertainties in the target shape model (i.e., CoM and inertia matrix). The consider filter uses a set of parameters $\mathbf{c}$ with their corresponding uncertainties to inflate the state covariance based on their correlations with the state. The new discrete-time propagation equations become the following:[3]

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}_k\left(\mathbf{x}_k, \mathbf{c}, t_k\right) \\ \mathbf{P}_{xx,k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_{xx,k}^{+} \mathbf{\Phi}_k^T + \mathbf{\Phi}_k \mathbf{P}_{xc,k}^{+} \mathbf{\Psi}_k^T + \mathbf{\Psi}_k \mathbf{P}_{cx,k}^{+} \mathbf{\Phi}_k^T + \mathbf{\Psi}_k \mathbf{P}_{cc,k}^{+} \mathbf{\Psi}_k^T + \mathbf{Q}_k \\ \mathbf{P}_{xc,k+1}^{-} = \left(\mathbf{P}_{xc,k+1}^{-}\right)^T = \mathbf{\Phi}_k \mathbf{P}_{xc,k}^{+} + \mathbf{\Psi}_k \mathbf{P}_{cc,k}^{+} \\ \mathbf{P}_{cc,k+1}^{-} = \mathbf{P}_{cc,k}^{+} \end{cases} \tag{3}$$

where $\mathbf{\Phi}_k = \mathbf{\Phi}\left(t_{k+1}, t_k\right)$, and $\mathbf{\Psi}_k = \mathbf{\Psi}\left(t_{k+1}, t_k\right)$ is the state transition matrix associated with the consider parameters. The state covariance is sectioned in four submatrices as follows $\mathbf{P} = \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xc} \\ \mathbf{P}_{cx} & \mathbf{P}_{cc} \end{bmatrix}$. Thus, the update equations become:

$$\begin{cases} \mathbf{x}_k^{+} & = \mathbf{x}_k^{-} + \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_k\right) \\ \mathbf{c}^{+} & = \mathbf{c}^{-} \\ \hat{\mathbf{z}}_k & = \mathbf{h}(\mathbf{x}_k^{-}, \mathbf{c}^{-}) \\ \mathbf{P}_{xx,k}^{+} & = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_{x,k}\right) \mathbf{P}_{xx,k}^{-} - \mathbf{K}_k \mathbf{H}_{c,k} \mathbf{P}_{cx,k}^{-} \\ \mathbf{P}_{xc,k}^{+} & = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_{x,k}\right) \mathbf{P}_{xc,k}^{-} - \mathbf{K}_k \mathbf{H}_{c,k} \mathbf{P}_{cc,k}^{-} \\ \mathbf{K}_k & = \left(\mathbf{P}_{xx,k}^{-} \mathbf{H}_{x,k}^T + \mathbf{P}_{xc,k}^{-} \mathbf{H}_{c,k}^T\right) \Big(\mathbf{H}_{x,k} \mathbf{P}_{xx,k}^{-} \mathbf{H}_{x,k}^T + \mathbf{H}_{x,k} \mathbf{P}_{xc,k}^{-} \mathbf{H}_{c,k}^T + \\ & \quad + \mathbf{H}_{c,k} \mathbf{P}_{cx,k}^{-} \mathbf{H}_{x,k}^T + \mathbf{H}_{c,k} \mathbf{P}_{cc,k}^{-} \mathbf{H}_{c,k}^T + \mathbf{R}_k\Big)^{-1} \end{cases} \tag{4}$$

where the measurement Jacobian $\mathbf{H}$ is partitioned similarly to the state covariance $\mathbf{P}$ and the consider parameters $\mathbf{c}$ remain constant. Thanks to this approach, the filter aims to better model covariance evolution during propagation and update, by tuning covariance inflation based on statistic priors about the target center



Onboard frames	
i, $\mathcal{I}$	Inertial frame
ℓ, $\mathcal{L}$	Local Vertical Local Horizontal frame (of the chaser)
c, $\mathcal{C}$	Chaser frame
s, $\mathcal{S}$	Camera frame
t, $\mathcal{T}$	Target onboard frame
t, $\mathcal{T}'$	Target current onboard frame
$\mathcal{P}$	Target onboard principal frame
Unknown frames	
$\mathcal{P}'$	Target principal frame
p	Target center of mass

Figure 8: Frames definitions used in the navigation filter.

of mass and inertia matrix. Stemming from this general formulation, the specialized equations are reported hereunder.

In this section, we use the following notation:

- Points are indicated with lowercase characters, e.g., a, b
- Frames are indicated with uppercase calligraphic characters, e.g., $\mathcal{A}$, $\mathcal{B}$
- ${}^{\mathcal{A}}\mathbf{r}_{a/b}$, ${}^{\mathcal{A}}\mathbf{v}_{a/b}$ respectively denote position and velocity of the point "a" with respect to point "b" expressed in frame $\mathcal{A}$
- $\mathbf{q}_{\mathcal{A}/\mathcal{B}}$, $\sigma_{\mathcal{A}/\mathcal{B}}$ respectively denote quaternion and Modified Rodrigues Parameter (MRP) of the rotation from frame $\mathcal{B}$ to frame $\mathcal{A}$
- $[\mathcal{AB}]$ denotes the direction cosine matrix of the rotation from frame $\mathcal{B}$ to frame $\mathcal{A}$
- ${}^{\mathcal{A}}\omega_{\mathcal{A}/\mathcal{B}}$ angular rate of frame $\mathcal{A}$ with respect to frame $\mathcal{B}$ expressed in frame $\mathcal{A}$
- $\left[{}^{\mathcal{A}}\mathbf{J}\right]$ inertia matrix expressed in frame $\mathcal{A}$

To model all these uncertainties, the filter uses a set of frames as shown in Figure 8. We identify three frames on the chaser:

1. The $\mathcal{C}$ frame (origin: c): the geometric chaser frame, centered on its center of mass for simplicity
2. The $\mathcal{L}$ frame (origin: ℓ), i.e., the Local Vertical Local Horizontal (LVLH) frame for dynamics, centered on the chaser CoM
3. The $\mathcal{S}$ frame (origin: s) centered in the camera location and having its boresight along the z-axis

Moreover, we identify on the target:

1. The $\mathcal{T}$ frame (origin: t) denoting the geometric frame of the target, centered on the onboard center of mass for simplicity

2. The $\mathcal{T}'$ frame (origin: t) denoting the current geometric frame of the Target, as represented by the rotation parameter in the filter

3. The $\mathcal{P}$ frame (origin: p) aligned with the onboard available target principal axes and centered in its real CoM

4. The $\mathcal{P}'$ frame (origin: p) aligned with the real target principal axes and centered in its real CoM.

Finally, the inertial frame is denoted as $\mathcal{I}$ (origin: i).

The navigation filter is parameterized using these frame definitions. The translational state uses the relative Cartesian position and velocity of the target with respect to the chaser LVLH frame. Translation propagation uses the well-known Clohessy-Wiltshire solution of the Hill equations[29] with analytical state transition matrices for state and covariance propagation. The rotational state uses the MRP error from the reference target rotation. Therefore, the navigation state is the following:

$$\mathbf{x} = \begin{pmatrix} {}^{\mathcal{L}}\mathbf{r}_{p/\ell} \\ {}^{\mathcal{L}}\mathbf{v}_{p/\ell} \\ \sigma_{\mathcal{T}/\mathcal{T}'} \\ {}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}} \\ \mathbf{c} \end{pmatrix} \tag{5}$$

where $\sigma_{\mathcal{T}/\mathcal{T}'}$ is the MRP error and $\mathbf{c}$ are the consider parameters (i.e., CoM and inertia). Here, the MRP error undergoes a reset by updating the reference attitude expressed with quaternions at each propagation with the following relationship:[30]

$$\mathbf{q}^{+}_{\mathcal{I}/\mathcal{T}} = \mathbf{q}^{-}_{\mathcal{I}/\mathcal{T}} \otimes \delta\mathbf{q}\left(\sigma_{\mathcal{T}/\mathcal{T}'}\right) \ , \ \ \delta\mathbf{q}\left(\sigma\right) = \frac{1}{16 + ||\sigma||^2} \begin{pmatrix} 16 - ||\sigma||^2 \\ 8\sigma \end{pmatrix} \tag{6}$$

This is standard practice in IOS applications (e.g., see Tweddle et al.,[31] Crassidis et al.,[28] and Park et al.[32]). Attitude is propagated with a Runge-Kutta 4 scheme on the reference quaternion and the angular rate using the following equations:

$$\dot{\mathbf{q}}_{\mathcal{I}/\mathcal{T}} = \frac{1}{2}\mathbf{q}_{\mathcal{I}/\mathcal{T}} \otimes {}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}} \tag{7}$$

$${}^{\mathcal{T}}\dot{\omega}_{\mathcal{T}/\mathcal{I}} = -\left[{}^{\mathcal{T}}\mathbf{J}\right]^{-1}\left({}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}} \times \left[{}^{\mathcal{T}}\mathbf{J}\right]{}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}}\right) \tag{8}$$

where $\left[{}^{\mathcal{T}}\mathbf{J}\right]$ is the inertia matrix of the target in the target frame. We define this dynamics representation approach as a "detailed" approach for rotation, where the filter uses a complete rotational model for smoother propagation. This is opposed to an "agnostic" approach, which omits full dynamics to have a less computationally expensive solution and ignores any dependency with respect to the uncertain target. It is worth noting that agnostic approach is reported in the paper for comparison as a possible filter design under shape-induced uncertainty. The agnostic approach models the dynamics as:

$${}^{\mathcal{T}}\dot{\omega}_{\mathcal{T}/\mathcal{I}} = \mathbf{0}_{3\times 1} \tag{9}$$

As anticipated, consider parameters are used in the detailed approach to account for CoM and inertia errors. We express the consider parameters' vector $\mathbf{c}$ as:

$$\mathbf{c} = \begin{pmatrix} {}^{\mathcal{T}}\mathbf{r}_{p/t} \\ \mathbf{k} \\ \sigma_{\mathcal{P}/\mathcal{P}'} \end{pmatrix} \tag{10}$$

where ${}^{\mathcal{T}}\mathbf{r}_{p/t}$ is the position of the target center of mass in the a priori target frame and $\sigma_{\mathcal{P}/\mathcal{P}'}$ is the MRP that represents the unknown error from the true target principal reference frame to its a priori knowledge. Under the assumption of small angular errors, it can be stated that

$$\begin{cases} [\mathcal{T}\mathcal{P}'] = [\mathcal{T}\mathcal{P}]\,[\mathcal{P}\mathcal{P}'] \approx [\mathcal{T}\mathcal{P}]\left(\mathbf{I}_{3\times 3} - \left[\sigma_{\mathcal{P}/\mathcal{P}'}\right]_{\times}\right) \\ \left[{}^{\mathcal{T}}\mathbf{J}\right] = [\mathcal{T}\mathcal{P}]\left[{}^{\mathcal{P}}\mathbf{J}\right][\mathcal{P}\mathcal{T}] \end{cases} \tag{11}$$

where $[\cdot]_{\times}$ is the skew operator associated with the cross product. Moreover, following Tweddle et al.,[33] we parameterize the normalized diagonal inertia matrix $\left[{}^{\mathcal{P}}\mathbf{J}\right]$ via the the inertia ratios vector $\mathbf{k}$ as follows:

$$\begin{cases} \left[{}^{\mathcal{P}}\mathbf{J}\right] = \begin{bmatrix} e^{k_1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-k_2} \end{bmatrix} \\ \mathbf{k} = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix},\ k_1 = \ln\left(\frac{J_{xx}}{J_{yy}}\right),\ k_2 = \ln\left(\frac{J_{yy}}{J_{zz}}\right) \end{cases} \tag{12}$$

Note that derivatives for the filter can be computed analytically from this formulation, but derivation is here omitted for the sake of brevity.

While the inertia ratios and $\sigma_{\mathcal{P}/\mathcal{P}'}$ affect the state through Euler's equation, the CoM does not directly influence the dynamics, as it appears only in the measurement equations. As we employ a tightly coupled scheme,[26] the filter measurements are the keypoint locations detected by the IP:

$$\mathbf{z} = \left(\mathbf{z}_1^T, ..., \mathbf{z}_m^T\right)^T = (u_1, v_1, ..., u_m, v_m)^T \tag{13}$$

where m is the number of keypoints. The projection equation is the following:

$${}_h\mathbf{z}_j = \mathbf{C}^{\mathcal{S}}\mathbf{r}_{j/s} = \mathbf{C}\left([\mathcal{S}\mathcal{T}]\,{}^{\mathcal{T}}\mathbf{r}_{j/t} + {}^{\mathcal{S}}\mathbf{r}_{t/s}\right) \tag{14}$$

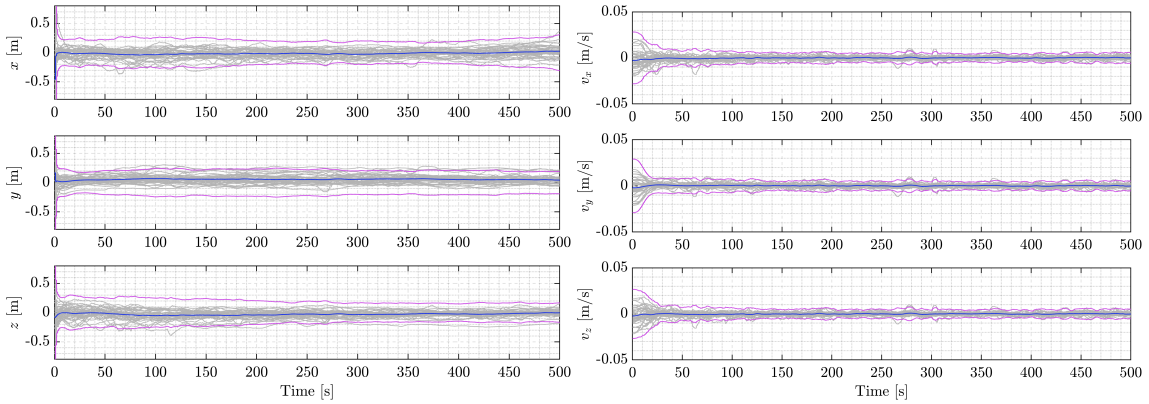
where $\mathbf{C}$ is the intrinsic camera matrix and ${}_h\mathbf{z}_j$ is the keypoint location in homogeneous coordinates. Note that non-homogeneous coordinates can be derived by simply dividing the two first components by the third. This equation may be expressed as a function of the state as follows:

$$\begin{aligned} {}_h\mathbf{z}_j &= \mathbf{C}^{\mathcal{S}}\mathbf{r}_{j/s} = \mathbf{C}\left([\mathcal{S}\mathcal{T}]\,{}^{\mathcal{T}}\mathbf{r}_{j/t} + {}^{\mathcal{S}}\mathbf{r}_{t/s}\right) = \mathbf{C}\left([\mathcal{S}\mathcal{T}]\,{}^{\mathcal{T}}\mathbf{r}_{j/t} - [\mathcal{S}\mathcal{T}]\,{}^{\mathcal{T}}\mathbf{r}_{p/t} + [\mathcal{S}\mathcal{L}]\,{}^{\mathcal{L}}\mathbf{r}_{p/s}\right) \\ &= \mathbf{C}\left([\mathcal{S}\mathcal{C}]\,[\mathcal{C}\mathcal{I}]\,[\mathcal{I}\mathcal{T}]\left({}^{\mathcal{T}}\mathbf{r}_{j/t} - {}^{\mathcal{T}}\mathbf{r}_{p/t}\right) + [\mathcal{S}\mathcal{L}]\,{}^{\mathcal{L}}\mathbf{r}_{p/\ell} + {}^{\mathcal{S}}\mathbf{r}_{\ell/s}\right) \end{aligned} \tag{15}$$

where $\mathbf{r}_{j/t}$ is the position of the j-th landmark in the target wireframe. Recall that the state is present in the rotation $[\mathcal{I}\mathcal{T}]$ and the position ${}^{\mathcal{L}}\mathbf{r}_{p/\ell}$. It should be noted that the CoM position biases the measurements through ${}^{\mathcal{T}}\mathbf{r}_{p/t}$. To contain this bias, we include this parameter and its covariance in the Schmidt-Kalman[3] framework, correlating it to the state through the update equation. Note that the aforementioned agnostic approach does not include this parameter, but we contain this bias through measurement noise tuning. Finally, we perform measurement editing using the Mahalanobis distance[2] in the innovation term with a 5σ confidence value.

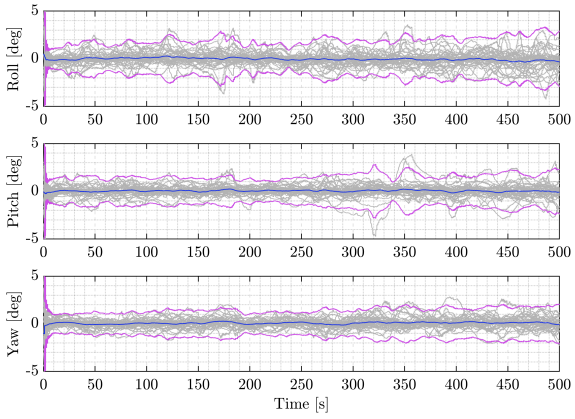
RESULTS

In this section, we present the results with different image processing and filtering configurations in a Monte Carlo campaign. In particular, we analyze two different image processing setups and two different filter setups. Regarding image processing, we either use a full-labeling network followed by a PnP-based RANSAC algorithm or a visibility-labeling network not followed by any RANSAC scheme. This choice is made to support our claim that the visibility-labeling network already performs outlier rejection. It is worth noting that, in operational VBN design, RANSAC could be included. The navigation filters are configured in a detailed or in an agnostic setup, where the second is presented for performance comparison.

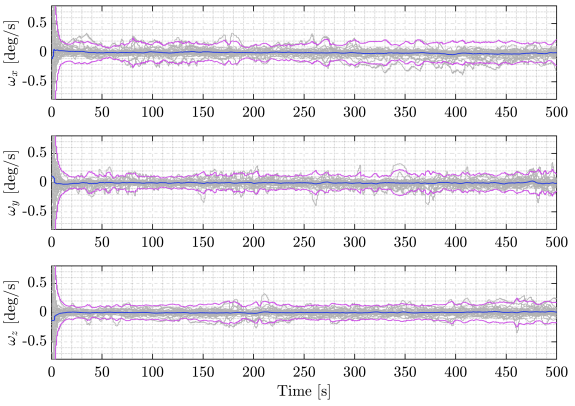


(a) Position error (LVLH frame).

(b) Velocity error (LVLH frame).

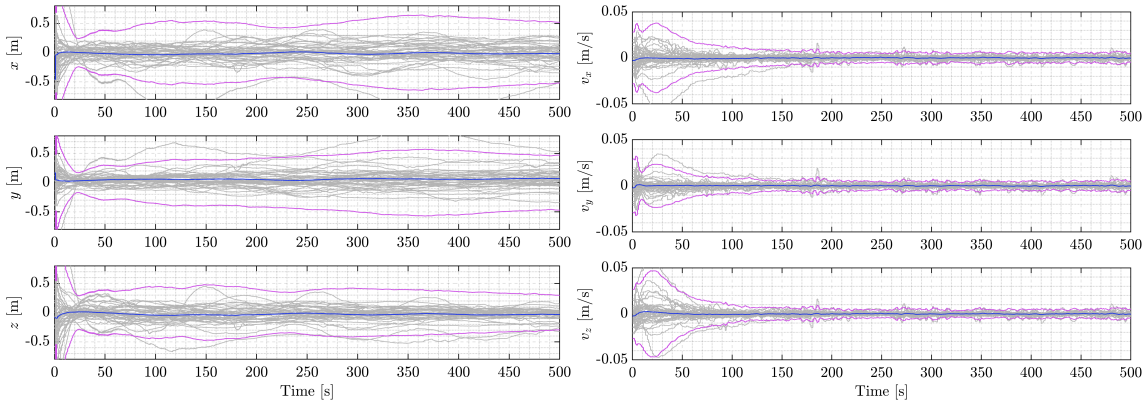


(c) Attitude $[\mathcal{TI}]$ error (Euler ZYX angles).



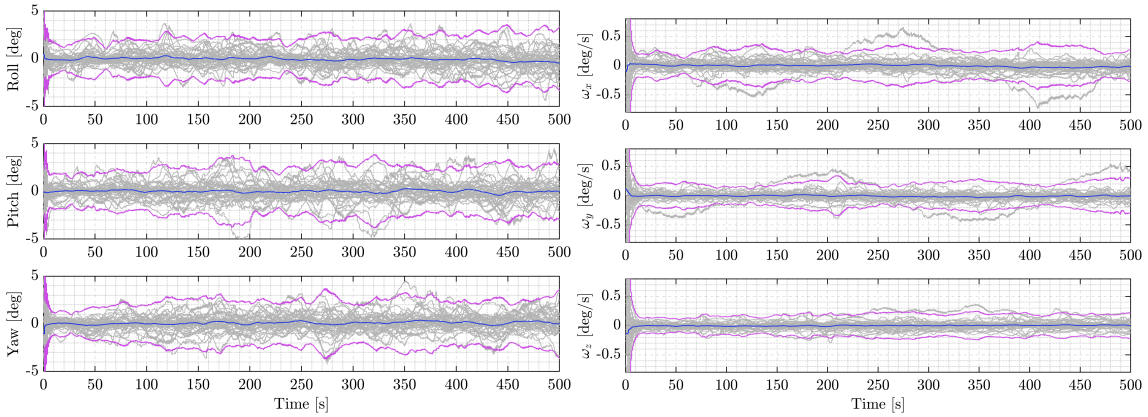
(d) Angular rate ${}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}}$ error.

Figure 9: Monte Carlo results for the detailed pipeline with full-labeling network.



(a) Position error (LVLH frame).

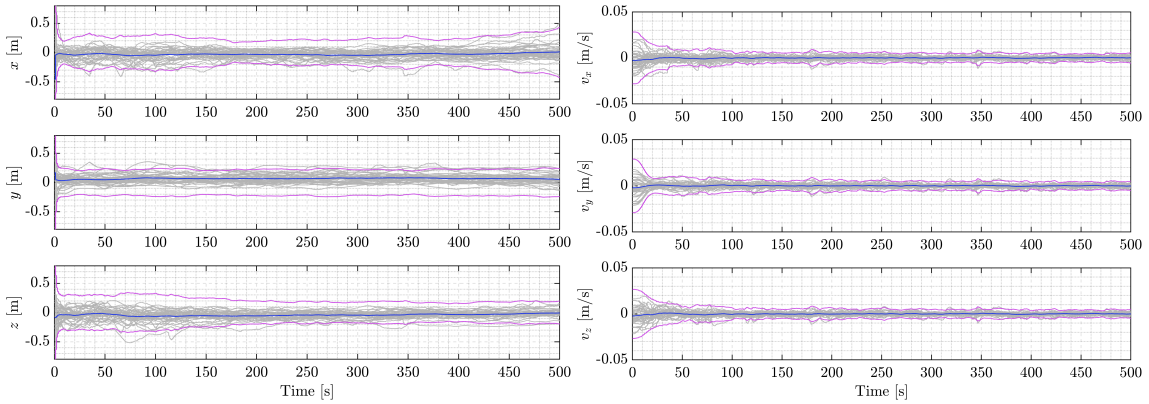
(b) Velocity error (LVLH frame).



(c) Attitude $[\mathcal{TI}]$ error (Euler ZYX angles).

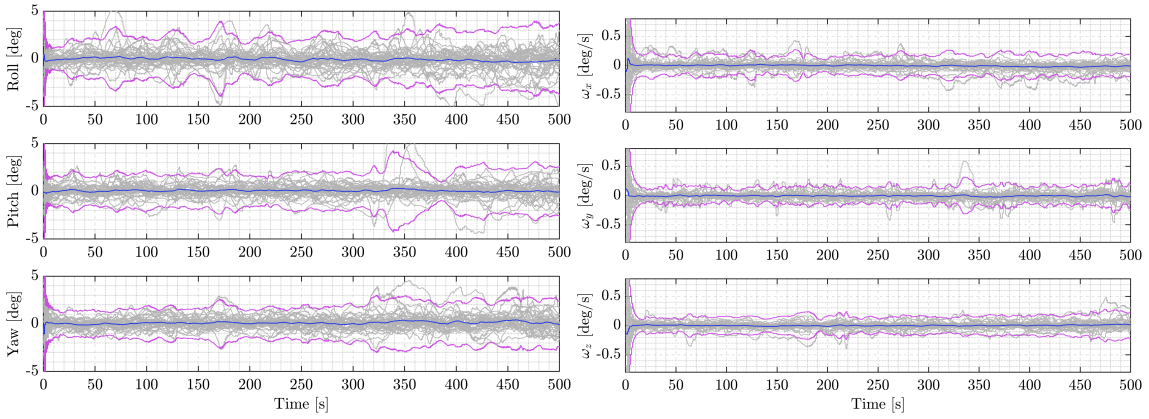
(d) Angular rate ${}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}}$ error.

Figure 10: Monte Carlo results for the agnostic pipeline with full-labeling network.



(a) Position error (LVLH frame).

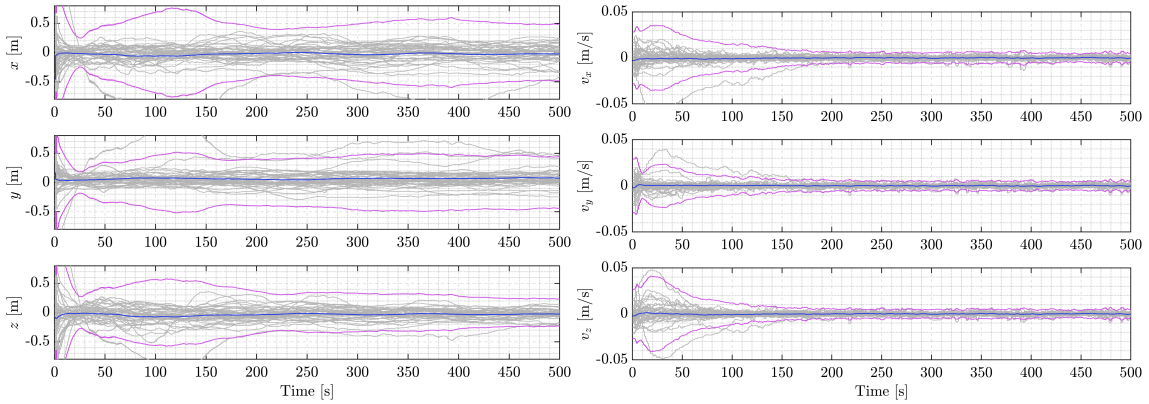
(b) Velocity error (LVLH frame).



(c) Attitude $[\mathcal{TI}]$ error (Euler ZYX angles).

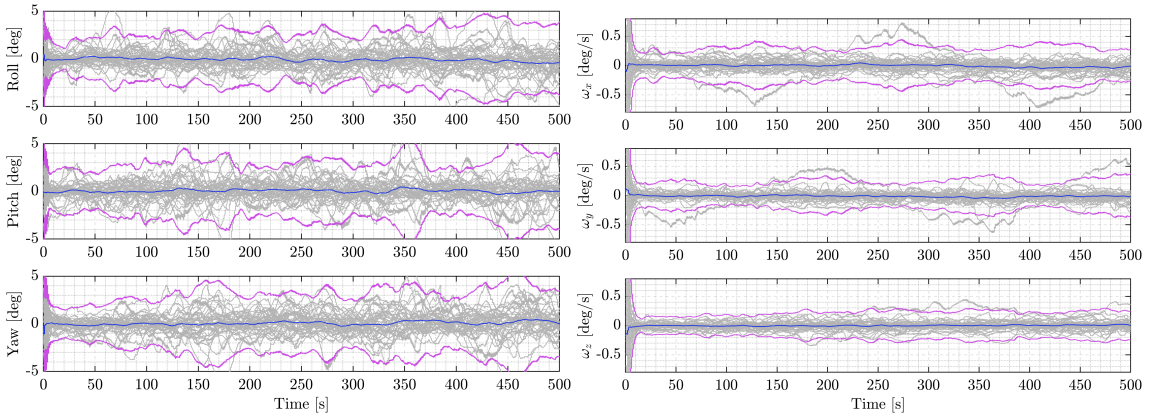
(d) Angular rate ${}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}}$ error.

Figure 11: Monte Carlo results for the detailed pipeline with visibility-labeling network.



(a) Position error (LVLH frame).

(b) Velocity error (LVLH frame).



(c) Attitude $[\mathcal{TI}]$ error (Euler ZYX angles).

(d) Angular rate ${}^{\mathcal{T}}\omega_{\mathcal{T}/\mathcal{I}}$ error.

Figure 12: Monte Carlo results for the agnostic pipeline with visibility-labeling network.

Concerning the simulation scenario, we scatter around a base orbit of 500 km altitude. The chaser and the target are in a baseline relative inspection orbit of relative orbital elements[34] $(0,\ 0,\ 15,\ 0,\ 20,\ 0)^T$ m. The target tumbles without any control at angular rates up to 4 deg/s. The simulation propagates translation and attitude including spherical harmonics, atmospheric drag, solar radiation pressure, Sun-Moon effects, gravity gradient, and Earth magnetic field. The target is scattered in its inertial position, inertial attitude, mass, CoM, and inertia. The Monte Carlo campaign counts 40 samples, each featuring a dedicated sequential dataset generated with SpiCam[7] and a different unseen perturbed target model with a randomly rotated solar panel.

We show the results of this analysis in terms of position and velocity errors for translation, and Euler angles and angular rate errors for rotation in Figures 9, 10, 11, and 12. On the one hand, we compare the two image processing setups. The full-labeling setup is shown in Figures 9 and 10, and the visibility-labeling setup is shown in Figures 11 and 12. It can be observed that the 3σ accuracy is comparable between the two approaches when using the same filter configuration. While the full-labeling approach seems to be more robust when coupled with the RANSAC, the visibility-labeling pipeline still has overall stable performance without introducing any outlier rejection. On the other hand, we assess the differences between the two filter models. Figures 9 and 11 show the results for the detailed approach, while Figures 10 and 12 depict the agnostic approach. The over-time average 3σ errors of the Monte Carlo simulation are shown in the following Table 2. Ultimately, these results show that the agnostic filter proves to be less robust than the detailed one, as

Table 2: Average 3σ error values of the Monte Carlo simulation.

Network labeling	Filter approach	Position error [m]	Attitude error [deg]
Full	Detailed	0.213	1.656
	Agnostic	0.455	2.404
Visibility	Detailed	0.235	2.181
	Agnostic	0.445	3.077

it struggles to contain the biases generated by unmodeled errors due to the uncertain target properties, which impact the dynamics and the measurement uncertainty levels. Finally, concerning image processing designs, the two setups have similar performance, outlining the embedded outlier rejection capability of the visibility-labeling network, which maintains performance without any appended outlier rejection scheme. RANSAC may still be applied to the visibility-labeling image processing to improve performance. Overall, the use of visibility labeling may inform a more outlier-robust design at network level, while increasing the Gaussianity of the keypoints pixel error.

CONCLUSION

This article presented the results of the study of an AI-aided pose estimation pipeline for a close-proximity non-cooperative scenario with a partially known target. This study required a rethinking of the entire algorithm design. First, we created a novel training dataset accounting for shape variability, to help the network generalize to unseen models. Second, we tailored the image processing design by performing a trade-off between a full-labeling and a visibility-based labeling scheme for supervised training. Third, we developed a navigation filter that accounts for target uncertainties in its dynamical modeling. Finally, we tested those approaches in an image-generation-in-the-loop Monte Carlo campaign of an inspection of a tumbling spacecraft. The results underline the trade-off between the various image processing and filtering configurations. The detailed approach shows better smoothing in uncertainty estimation with respect to the agnostic approach, and the two labeling approaches show comparable performance, with the visibility-based one showing promise for its embedded outlier rejection capability and overall keypoint error distribution.

ACKNOWLEDGMENTS

This work was conducted under EISI Agreement No. 4000144147, within the Open Space Innovation Platform (OSIP), through the support of the European Space Agency (ESA) and Thales Alenia Space France.

REFERENCES

[1] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudilliere, and D. Aouada, "A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects," *Acta Astronautica*, Vol. 212, 2023, pp. 339–360.

[2] J. R. Carpenter and C. N. D'Souza, "Navigation filter best practices," tech. rep., 2025.

[3] D. Woodbury and J. Junkins, "On the consider Kalman filter," *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7752.

[4] T. H. Park, M. Märtens, G. Lecuyer, D. Izzo, and S. D'Amico, "SPEED+: Next-Generation Dataset for Spacecraft Pose Estimation across Domain Gap," *2022 IEEE Aerospace Conference (AERO)*, pp. 1–15.

[5] S. D'Amico, P. Bodin, M. Delpech, and R. Noteborn, "Prisma," *Distributed space missions for earth system monitoring*, pp. 599–637, Springer, 2012.

[6] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, and S. D'Amico, "Satellite pose estimation challenge: Dataset, competition design, and results," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 5, 2020, pp. 4083–4098.

[7] F. Gallet, C. Marabotto, and T. Chambon, "Exploring AI-Based Satellite Pose Estimation: from Novel Synthetic Dataset to In-Depth Performance Evaluation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2024, pp. 6770–6778.

[8] J. Lebreton, I. Ahrns, R. Brochard, C. Haskamp, H. Krüger, M. L. Goff, N. Menga, N. Ollagnier, R. Regele, F. Capolupo, *et al.*, "Training Datasets Generation for Machine Learning: Application to Vision Based Navigation," *arXiv preprint arXiv:2409.11383*, 2024.

[9] C. Donlon, B. Berruti, S. Mecklenberg, J. Nieke, H. Rebhan, U. Klein, A. Buongiorno, C. Mavrocordatos, J. Frerick, B. Seitz, *et al.*, "The sentinel-3 mission: Overview and status," *2012 IEEE international geoscience and remote sensing symposium*, IEEE, 2012, pp. 1711–1714.

[10] D. Kirk, *Graphics Gems III (IBM Version): Ibm Version*. Elsevier, 2012.

[11] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Single-stage keypoint-based category-level object pose estimation from an RGB image," *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 1547–1553.

[12] Y. Bukschat and M. Vetter, "EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach," *arXiv preprint arXiv:2011.04307*, 2020.

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

[14] P. F. Huc, E. Bates, and S. D'Amico, "Fast Learning of Non-Cooperative SpaceCraft 3D Models Through Primitive Initialization," *arXiv preprint arXiv:2507.19459*, 2025.

[15] T. Chambon and F. Gallet, "A robust approach merging deep learning and unscented Kalman for vision based space rendez-vous," working paper or preprint, Sept. 2024.

[16] K. Liu and Y. Yu, "Revisiting the Domain Gap Issue in Non-cooperative Spacecraft Pose Tracking," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, pp. 6864–6873.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, Vol. 81, No. 2, 2009, pp. 155–166.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 4510–4520.

[20] S. Kaki, J. Deutsch, K. Black, A. Cura-Portillo, B. A. Jones, and M. R. Akella, "Real-time image-based relative pose estimation and filtering for spacecraft applications," *Journal of Aerospace Information Systems*, Vol. 20, No. 6, 2023, pp. 290–307.

[21] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[22] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 10778–10787.

[23] T. H. Park and S. D'Amico, "Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap," *Advances in Space Research*, Vol. 73, No. 11, 2024, pp. 5726–5740.

[24] A. Lotti, D. Modenini, and P. Tortora, "Investigating vision transformers for bridging domain gap in satellite pose estimation," *International Conference on Applied Intelligence and Informatics*, Springer, 2022, pp. 299–314.

[25] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu, "Segnext: Rethinking convolutional attention design for semantic segmentation," *Advances in neural information processing systems*, Vol. 35, 2022, pp. 1140–1156.

[26] L. P. Cassinis, R. Fonod, E. Gill, I. Ahrns, and J. Gil-Fernández, "Evaluation of tightly-and loosely-coupled approaches in CNN-based pose estimation systems for uncooperative spacecraft," *Acta Astronautica*, Vol. 182, 2021, pp. 189–202.

[27] S. Julier, "The scaled unscented transformation," *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, IEEE, pp. 4555–4559 vol.6.

[28] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *Journal of guidance, control, and dynamics*, Vol. 26, No. 4, 2003, pp. 536–542.

[29] W. Fehse, *Automated rendezvous and docking of spacecraft*. No. 16 in Cambridge aerospace series, Cambridge University Press, 2003.

[30] H. Schaub and J. L. Junkins, *Analytical mechanics of space systems*. AIAA education series, American Institute of Aeronautics and Astronautics, 2. ed ed., 2009.

[31] B. E. Tweddle and A. Saenz-Otero, "Relative computer vision-based navigation for small inspection spacecraft," *Journal of guidance, control, and dynamics*, Vol. 38, No. 5, 2015, pp. 969–978.

[32] T. H. Park and S. D'Amico, "Adaptive neural-network-based unscented kalman filter for robust pose tracking of noncooperative spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 9, 2023, pp. 1671–1688.

[33] B. E. Tweddle, A. Saenz-Otero, J. J. Leonard, and D. W. Miller, "Factor Graph Modeling of Rigid-body Dynamics for Localization, Mapping, and Parameter Estimation of a Spinning Object in Space," *Journal of Field Robotics*, Vol. 32, No. 6, pp. 897–933.

[34] S. D'Amico, "Relative orbital elements as integration constants of Hill's equations," *DLR, TN*, 2005, pp. 05–08.