POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Deep Multi-Task Learning Based Monocular Relative Pose Estimation of Uncooperative Spacecraft

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Francesco Evangelisti

Advisor: Prof. Perluigi Di Lizia

Co-advisors: Francesco Rossi (AIKO srl), Tobia Giani (AIKO srl)

Academic year: 2022-2023

1. Introduction

Close-proximity autonomous navigation about an uncooperative spacecraft is a crucial problem in the modern space industry for in-orbit servicing missions as well as Active Debris Removal (ADR) operations, and estimating the relative pose of the target is a critical task. A *ground-based* approach is unfeasible to achieve the accuracy level needed in close-proximity scenarios, while a *spaceborne solution* implies the use of on-board sensors: using a low Size-Weight-Power-Cost (SWaP-C) sensor like a monocular camera is preferred over heavier and more energy consuming sensors like LiDARs and stereo cameras, but introduces the need of very robust software to perform pose estimation. This dissertation introduces AIKO-NET, a deep Convolutional Neural Network (CNN) capable of estimating the relative pose of an uncooperative spacecraft from a single grayscale monocular image. Starting from SLAB's *Spacecraft Pose Network v2* (SPNv2) [3] as a baseline, the Multi-Task Learning (MTL) feature is expanded by trying to exploit a researched synergy between different yet related tasks. New feature maps and the respective prediction heads are introduced keeping the whole architecture modular and flexible, and a new dataset called *Multi-Feature Spacecraft Pose Estimation Dataset* (MFSPED) is presented and used to provide AIKO-NET with the new feature maps labels. Furthermore, a complete pose estimation pipeline is built: it consists of a relative trajectory generation module, a synthetic images generation process, the pose prediction through AIKO-NET, and the application of an Extended Kalman Filter (EKF) to the position predictions.

This work was developed in collaboration with AIKO, an Italian company pioneering in-orbit servicing and space logistics.

2. Theoretical Background

2.1. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep Artificial Neural Networks (ANNs) that are specifically designed for computer vision applications. ANNs are a fundamental component of the field of Machine Learning (ML), that focuses on creating algorithms that enable computers to learn and improve from experience. In ANNs, inputs are processed through interconnected layers of neurons (nodes), with each

node using a mathematical function to transform the input before passing it on to the next layer. The weights and biases associated with each node are adjusted during training to optimize the network's performance in solving specific tasks. Deep ANNs are characterized by the presence of multiple layers between the input and output layers, called *hidden layers*. In this framework, the Multi-Task Learning (MTL) approach involves solving different tasks in parallel to enhance generalization and performance by utilizing domain information present in the training signals of such different yet related tasks as an inductive bias. The operations characterizing CNNs are *convolution* and *pooling*, which enable the networks to process the input images by extracting what are called *features* used for computer vision tasks. In the next section, EfficientPose [1], a state-of-the-art object detection and pose estimation CNN, will be presented.

2.2. EfficientPose

EfficientPose is the approach used in SPNv2 [3] and in the work presented in this dissertation, and is designed to achieve accurate 6D pose predictions from images.

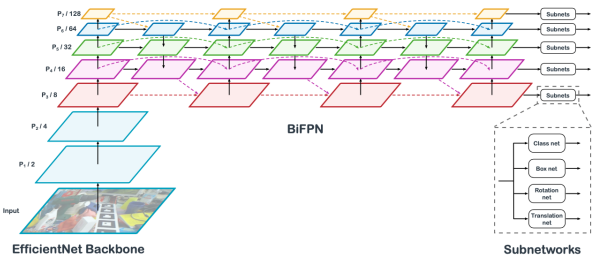


Figure 1: EfficientPose architecture.

This architecture, depicted in Figure 1, makes use of EfficientDet [5] as the backbone, which is an object detection model that derives from EfficientNet [6] the ability to efficiently scale the network (input resolution, width, and depth) in a principled manner. Furthermore, the implementation of the Bidirectional Feature Pyramid Network (BiFPN), which is an efficient multi-scale feature fusion logic, enables the network to learn the same features at different scales.
EfficientPose extends the EfficientDet architecture by exploiting its flexible architecture capable of collecting features at different scales and feeding them to class and box prediction subnetworks: two subnets are added to accurately predict the translation and rotation of the detected objects. The *rotation subnetwork* is responsible for predicting the rotation of an object in 3D space; the *translation subnetwork* is basically the same as the rotation one, but it outputs, for each anchor box (AB), a translation that represents the offset in pixels from the center of the AB to the center point of the corresponding object, exploiting the translational invariance of the input features.

2.3. Perspective-n-Point problem

The Perspective-n-Point problem involves estimating the pose of an object from a set of n 3D points of a known model in its body frame and their corresponding 2D projections in an image taken by a calibrated camera, by mapping the 3D points to their 2D projections.

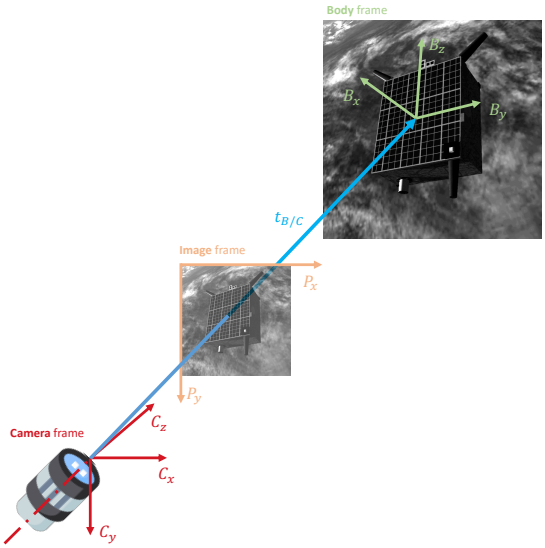


Figure 2: Reference frames involved in the PnP problem.

Taking Figure 2 as a reference, we can apply the problem to the proposed scenario by estimating the pose of the body frame B relative to the camera in the camera frame C. The projection $[u, w]^\top$ of a generic point of the target $\mathbf{r}^B$ on the image frame can be written as:

$$\begin{bmatrix} uw \\ vw \\ w \end{bmatrix} = \mathbf{K} \underbrace{\begin{bmatrix} \mathbf{R}_{B/C} & \mathbf{t}_{B/C} \end{bmatrix}}_{\text{unknown } \mathbf{P}} \begin{bmatrix} \mathbf{r}^B \\ 1 \end{bmatrix}, \quad (1)$$

where $\mathbf{K}$ is the camera intrinsic matrix and depends on the camera parameters, and where $\mathbf{P}$ is the unknown relative pose. The Efficient

Perspective-n-Point (EPnP) solver is a multi-stage analytical approach to the problem that consists in minimizing the reprojection error between the observed 2D image coordinates and the predicted 2D coordinates of the 3D points based on the estimated camera pose. The main idea is to express the n points as the weighted sum of 4 virtual control points and retrieve a formulation that brings 4 possible solutions: among these, the one associated with the lowest reprojection error is selected.

2.4. Relative Orbital Dynamics

The relative orbital dynamics problem is developed in a chief-deputy logic and is addressed by expressing the EOM describing the deputy position evolution in the chief's Hill frame $\{\hat{\mathbf{o}}_r, \hat{\mathbf{o}}_\theta, \hat{\mathbf{o}}_h\}$. Thus, the deputy relative state can be written as:

$$\begin{aligned} \boldsymbol{\rho} &= x\hat{\mathbf{o}}_r + y\hat{\mathbf{o}}_\theta + z\hat{\mathbf{o}}_h \\ \dot{\boldsymbol{\rho}} &= \dot{x}\hat{\mathbf{o}}_r + \dot{y}\hat{\mathbf{o}}_\theta + \dot{z}\hat{\mathbf{o}}_h \end{aligned} \tag{2}$$

The evolution of the deputy state is expressed by the following differential equations [4]:

$$\begin{aligned} \ddot{x} &= 2\dot{\nu}\left(\dot{y} - y\frac{\dot{r}_c}{r_c}\right) + x\dot{\nu}^2 + \frac{\mu}{r_c} \\ &\quad - \frac{\mu(r_c + x)}{((r_c + x)^2 + y^2 + z^2)^{3/2}} \\ \ddot{y} &= -2\dot{\nu}\left(\dot{x} - x\frac{\dot{r}_c}{r_c}\right) + y\dot{\nu}^2 \\ &\quad - \frac{\mu y}{((r_c + x)^2 + y^2 + z^2)^{3/2}} \\ \ddot{z} &= -\frac{\mu z}{((r_c + x)^2 + y^2 + z^2)^{3/2}} \end{aligned} \tag{3}$$

where ν is the chief's orbit true anomaly, μ is the standard gravitational parameter of the main attractor, and r_c is the chief's orbit radius. To complete the formulation, the orbital motion of the chief has to be expressed in terms of true anomaly and orbit radius as:

$$\begin{aligned} \ddot{r}_c &= r_c\dot{\nu}^2 - \frac{\mu}{r_c^2} \\ \ddot{\nu} &= -2\frac{\dot{r}_c}{r_c}\dot{\nu} \end{aligned} \tag{4}$$

2.5. Extended Kalman Filter

In the field of navigation, filtering is used to estimate the state of a system based on measurements. The Kalman filter operates in two stages: prediction and correction. In some cases, the system being estimated may have nonlinear dynamics, or the measurement model may be nonlinear. In such cases, an extended Kalman filter (EKF) can be used. Let us consider a generic nonlinear model for the state $\mathbf{x}$ in Equation 5, where $\mathbf{u}_k$ represents the input at the time step k, $f(\cdot)$ is the state transition function, $h(\cdot)$ is the measurement function which maps the state to the measured quantity $\mathbf{y}$, $\mathbf{w}_k$ and $\mathbf{v}_k$ identify the process and measurement noise.

$$\begin{cases} \mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k\right) \\ \mathbf{y}_k = h\left(\mathbf{x}_k, \mathbf{v}_k\right) \end{cases} \tag{5}$$

To linearize the state transition and the measurement functions, their Jacobian matrices can be computed as:

$$\mathbf{F} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}^+} \qquad \mathbf{H} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}^-} \tag{6}$$

The prediction phase consists in the propagation of both the state $\hat{\mathbf{x}}_k^+$ and the related covariance $\hat{\mathbf{P}}_k^+$ based on their values at the previous time step; the correction phase exploits the incoming measurements and an optimal weighting factor $\mathbf{K}_{k+1}$ called Kalman gain. The full EKF algorithm is reported in Equation 7

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^- &= f\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, 0\right) \\ \mathbf{P}_{k+1}^- &= \mathbf{F}\mathbf{P}_k^+\mathbf{F}^\top + \mathbf{Q}_k \\ \mathbf{K}_{k+1} &\coloneqq \mathbf{P}_{k+1}^-\mathbf{H}^\top\left(\mathbf{H}\mathbf{P}_{k+1}^-\mathbf{H}^\top + \mathbf{R}_{k+1}\right)^{-1} \\ \hat{\mathbf{x}}_{k+1}^+ &= \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1}\left(\mathbf{y}_{k+1} - h\left(\hat{\mathbf{x}}_{k+1}^-, 0\right)\right) \\ \mathbf{P}_{k+1}^+ &= \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}\right)\mathbf{P}_{k+1}^- \end{aligned} \tag{7}$$

Here, $\mathbf{Q}$ and $\mathbf{R}$ represent the covariance matrices associated with process noise and measurement noise, respectively.

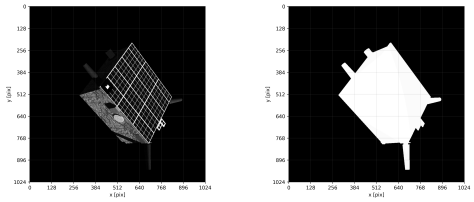
3. AIKO-NET

AIKO-NET is a CNN developed to enhance the current state-of-the-art MTL-based pose estimation of uncooperative spacecrafts through monocular images. The main purposes of AIKO-NET are to demonstrate the reproducibility of the SPNv2 [1] MTL-related improvements and to push this MTL nature of the network to further enhance the estimation accuracy by exploiting a researched synergy between different prediction heads. Thus, the presented architecture is

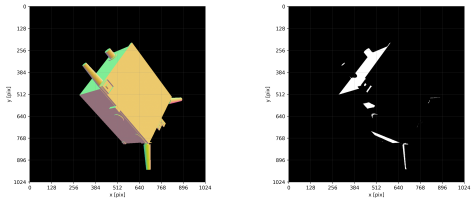
built on top of the SPNv2 which serves as the baseline. Since the prediction of new features is introduced, a completely customized dataset called *Multi-Feature Spacecraft Pose Estimation Dataset* (MFSPED) had to be generated to implement the multi-task approach, and also preprocessing and dataset generation pipelines were developed in order to build an extremely flexible and interconnected framework which finally led to ease of use and modularity of the whole network and training conditions.

3.1. Custom Dataset

Taking inspiration from SLAB's SPEED+ dataset [2], MFSPED is made of synthetic images of the Tango satellite from the PRISMA mission. Building a dataset for multi-task learning purposes involves brainstorming some feature maps that could help the network in solving its main task.



(a) Satellite synthetic image. (b) Depthmap.



(c) Normalmap. (d) Shadowmap.

Figure 3: An example of a synthetic image of Tango and the associated feature maps.

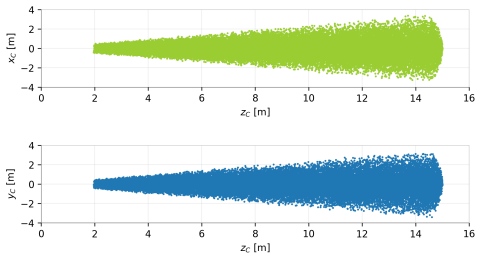
Three additional features, shown in Figure 3, have been thought of:

a) **Depthmap**: a segmentation mask of the satellite with additional information about the distance from the camera. The closer to the camera, the whiter the mask pixels; the further from it, the darkest.

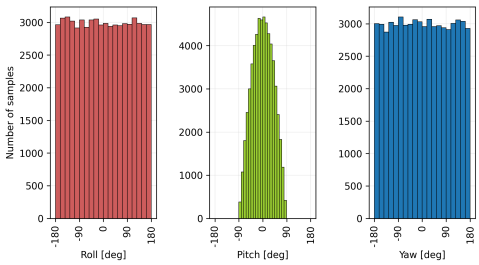
b) **Normalmap**: a colormap related to the normal directions to the satellite surfaces in the target's body frame: the same surface results associated with the same color for every image in the dataset.

c) **Shadowmap**: an image where the shadowed part of the satellite corresponds to white pixels. This feature should help the net to better manage different lighting conditions.

The dataset images are associated with a final feature: the **keypoints Heatmap**. This feature provides 2D heatmaps associated with each of the 18 pre-defined keypoints of the synthetic images.



(a) Position labels distribution.



(b) Orientation labels distribution.

Figure 4: MFSPED labels distribution in the camera frame C.

The satellite images are synthesized in Unity starting from pose data that is distributed as displayed in Figure 4: the distance from the target is considered to be normally distributed on a range between 2 and 15 meters, and a random divergence of the camera is considered to account for an imperfect pointing of the target.
A brief description of the dataset processing pipeline is shown in Figure 5.

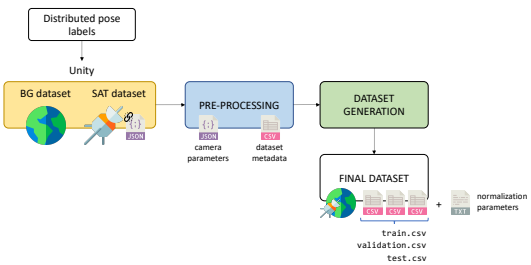


Figure 5: Full dataset pipeline.

3.2. Architecture

The AIKO-NET architecture is an expansion of SPNv2. The features are encoded by EfficientDet and fed to the prediction heads on different scales, from the 3^{rd} to the 7^{th} level of the BiFPN.

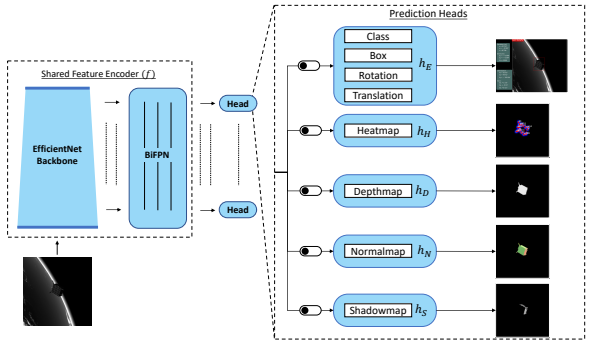


Figure 6: The AIKO-NET architecture.

The key difference with respect to SPNv2 is the addition of several prediction heads that, as depicted in Figure 6, can be switched on or off with ease. This modularity makes AIKO-NET a valid test-bed for experimenting a MTL approach with different head combinations. The following prediction heads are currently implemented:

1. EfficientPose head h_{E}, that is responsible for classification, bounding box prediction, and *direct pose estimation*;
2. Heatmap head h_{H}, which output is used by the EPnP for *indirect pose estimation*;
3. Depthmap head h_{D};
4. Normalmap head h_{N};
5. Shadowmap head h_{S}.

Obviously, one between the EfficientPose and the Heatmap heads has to be on in order to enable the pose prediction. The segmentation head used in the original SPNv2 architecture is replaced by h_{D}, which predicts a similar but slightly more informational feature.

The EfficientPose head loss can be seen as the sum of a:

- *Focal* loss for the classification task;
- *Complete Intersection over Union* (CIoU) loss for the object localization task;
- *SPEED* loss [3] for the pose estimation task.

The SPEED loss is defined in Equation 8, where $\mathbf{t}$ and $\mathbf{q}$ represent the relative translation and rotation, respectively, and where "$\hat{x}$" denotes the predicted quantities x. All the other heads are associated with a pixel-wise Mean Squared Error (MSE) loss. The total loss is a weighted sum of all the losses, and the EfficientPose head loss alone can be seen as the weighted sum of the classification, localization, and pose losses.

$$\begin{aligned} \mathrm{SPEED} &= e_t + Eq \\ &= \frac{\|\hat{\mathbf{t}} - \mathbf{t}\|}{\|\mathbf{t}\|} + 2 \cdot \arccos |\mathbf{q} \cdot \hat{\mathbf{q}}| \end{aligned} \tag{8}$$

4. Relative Pose Estimation Pipeline

The relative pose estimation pipeline consists of four main steps:

1. Trajectory generation by means of relative orbital dynamics.
2. Processing of the trajectory data in Unity to produce a sequence of synthetic images representing the target in the camera frame.
3. Direct and indirect pose estimations through AIKO-NET.
4. Filtering of the estimated position using an EKF.

5. Results

The backbone used is EfficientDet-D3, and the chosen input size for the network is 512×512. The generated dataset is made of 40000 images, and the splits used for the training, validation, and testing phases are presented in Table 1.

Table 1: Generated dataset splits

	`training`	`validation`	`testing`
Split	70%	20%	10%
N_{images}	28000	8000	4000

Exploiting the modularity of the architecture, the different configurations reported in Table 2 are tested.

Table 2: Prediction heads configurations for 9 versions of the network.

	`V0`	`V1`	`V2`	`V3`	`V4`	`V5`	`V6`	`V7`	`V8`
h_E	✓	✓	✓		✓	✓	✓	✓	✓
h_H	✓	✓		✓	✓	✓	✓	✓	✓
h_D	✓	✓				✓			✓
h_N	✓	✓					✓		✓
h_S	✓	✓						✓	✓
BGs		✓	✓	✓	✓	✓	✓	✓	✓

The aim of this process was to find suitable configurations that may over-perform others. The V0 configuration was used for the first tests on a dataset with no backgrounds to assess the performance of AIKO-NET in full configuration and optimal scenario conditions. Configurations from V0 to V7 were trained using loss functions that equally concur to the total loss to identify how much each prediction head enhances the "benchmark performance" associated with V0.
The last version, V8, has the same configuration as V1 but the loss weights are modified based on the previous results.
Each configuration is trained for 50 epochs with a batch size of 10, and validated each 2 epochs. A learning rate of 5e−4 is used, which is scaled by a factor of 1e−1 at the 75% and 90% of the training process.

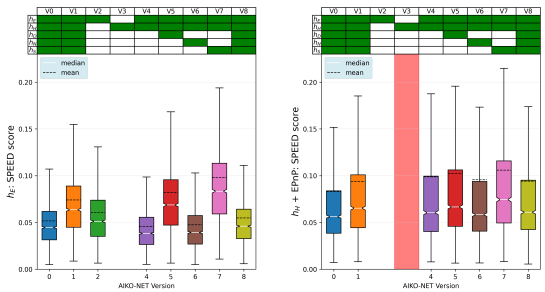


Figure 7: AIKO-NET versions SPEED score comparison. Direct estimation on the left, indirect estimation on the right. Config. V3 errors are replaced by a red shaded region because they are notably higher with respect to the ones from other versions.

Figure 7 depicts the boxplots of the SPEED scores for the different configurations, for both the direct and indirect methods. Comparing the left plot to the right one, we can say that the EfficientPose head has an overall better performance. AIKO-NET V1 is directly comparable to V0 since they share the same configurations: the performance of AIKO-NET V1 is slightly worse than V0, because of the increased difficulty of the problem due to backgrounds. The boxplots for V3 are not included in the plots due to the associated high errors: the architecture is not optimized for keypoint detection only. The indirect estimation method appears to benefit from the tasks performed by the head responsible for the direct estimation, as shown by V4's results. Also EfficientPose benefits from the parallelization with the heatmap estimation task: its predictions in V4 result more accurate than in V2. Despite the fact that the configurations from V5 to V7 exhibit similar levels of performance, it is interesting to note that the worst results are obtained with V7, which involves using the shadowmap feature (the most difficult one) in addition to h_E and h_H. On the other hand, V6 with the normalmap head shows slightly better performance than V4: this could be due to the high informative level of the RGB feature, which seems to aid the network in its primary tasks.
AIKO-NET V8 was trained with different weights associated with the different heads. Let w denote the loss weight associated with the heads denoted by the subscripts E, H, D, N, S. Let w_{E}^{bbox}, w_{E}^{pose} be the losses of EfficientPose tasks. The loss weights used by V8 are summarized in Table 3.

Table 3: Loss weights for AIKO-NET V8.

	w_{E}	w_{E}^{cls}	w_{E}^{bbox}	w_{E}^{pose}	w_{H}	w_{D}	w_{N}	w_{S}
V8	1	0.1	0.5	1	1	0.3	0.5	0.2

The test results reveal that V8's EfficientPose head exhibits improved performance in predicting both the relative position and attitude, as expected when compared to V1. However, there is no significant improvement observed in the relative pose estimation through the indirect method. An example of prediction visualization from AIKO-NET V8 is shown in Figure 8, where the EfficientPose and heatmap heads outputs are displayed in the first row, with the depthmap, normalmap, and shadowmap reported below.

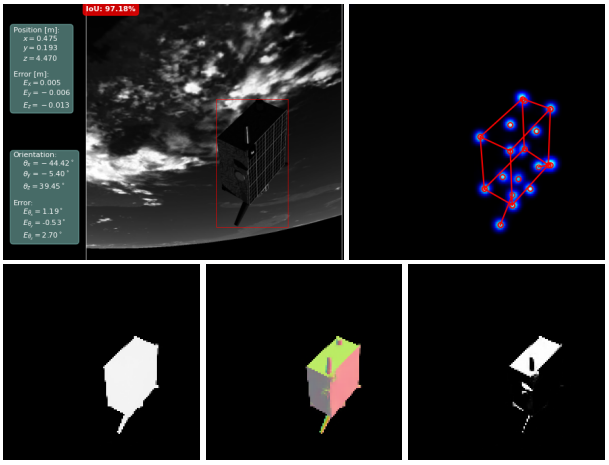


Figure 8: Prediction visualization example for *rgb_36434.png*.

Next, the full estimation pipeline was tested. A relative trajectory was generated, and a sequence of images with no backgrounds was output by Unity and used as input for AIKO-NET `VO`. The predictions were then processed by the EKF.
In the next graphs, showing the absolute position errors, the filtered (blue) and raw (yellow) measurements are overlaid and a logarithmic scale is used. The CNN predictions on a dynamic, simulated scene result consistent: the errors do not present strange behaviors or sudden spikes. This is a promising result as it suggests a first step towards the applicability of AIKO-NET to real-world scenarios. Although the predictions made by the CNN are already very accurate, with errors in the order of centimeters and millimeters, Figure 9 demonstrates the effectiveness of the filtering action. In this specific test, the EKF may not provide significant additional benefits from the error point of view, but its filtering action would be useful in a more complete application for control purposes by removing excessive noise in the measured states.

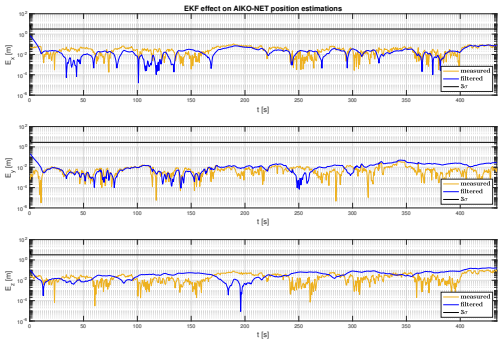


Figure 9: Effect of EKF on AIKO-NET position predictions.

To perform a more rigorous evaluation of the EKF's performance, additional $\sim dm$ noise was deliberately added to the measurements, and the noise measurement matrices were fine-tuned accordingly. With these more inaccurate measurements, the beneficial effect of the EKF is clear in Figure 10: the filtering often manages to lower the errors by an order of magnitude. Thus, the development of such a tool can be useful for lowering measurements errors in scenarios where the predictions are not as accurate as the ones provided by AIKO-NET `VO`: this opens scenarios that may consider pose estimation on satellites beyond the trained range, or the application of a lighter, smaller network for which the quality of the predictions may deteriorate with respect to the tested benchmark configuration.

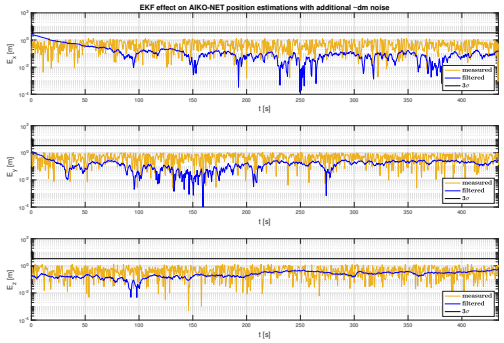


Figure 10: Effect of EKF on AIKO-NET disturbed ($\sim dm$) position predictions.

6. Conclusions and Future work

The thesis investigates the task parallelization benefits in a MTL CNN architecture for relative pose estimation of a known, uncooperative satellite from greyscale, monocular camera images. The study contributes by demonstrating the reproducibility of SLAB's SPNv2 MTL effects and by developing and exploring different configurations of AIKO-NET. AIKO-NET is a MTL based, modular architecture made of a total of 5 prediction heads that can solve multiple tasks simultaneously to improve pose estimation performance using both a direct and an indirect approach.
The study explores nine distinct configurations of AIKO-NET, showcasing state-of-the-art performance and the benefits of a modular architecture that can be trained with a virtually unlimited dataset. AIKO-NET was integrated into a comprehensive pipeline that begins with the custom generation of trajectories and the relative simulated images and concludes with the application of an EKF to the position estimates.
Future work includes addressing the domain gap problem, optimizing the standard deviation used for generating the ground-truth heatmaps for the keypoints prediction, exploring other loss types, applying a dynamically weighted loss function to optimize AIKO-NET, and embedding a filtering process for the relative orientation estimates. Future research directions should pave the way for the deployment of a complete vision-based relative navigation architecture for space systems.

References

[1] Yannick Bukschat. EfficientPose: An efficient, accurate and scalable end-to-end 6D..., 11 2020.

[2] Tae Gwan Park, Marcus Märtens, Gurvan Lecuyer, Dario Izzo, and Simone D'Amico. SPEED+: Next-Generation Dataset for Spacecraft Pose Estimation across Domain Gap. *arXiv (Cornell University)*, 10 2021.

[3] Tae Ha Park. Robust Multi-Task Learning and Online Refinement for Spacecraft..., 3 2022.

[4] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems*. AIAA, 1 2003.

[5] Mingxing Tan. EfficientDet: Scalable and Efficient Object Detection, 11 2019.

[6] Mingxing Tan. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 5 2019.