Monocular 6-DoF Pose Estimation of Spacecrafts Utilizing Self-iterative Optimization and Motion Consistency

Yunfeng Zhang[1,2,3], Linjing You[2,3], Luyu Yang[2,3], Zhiwei Zhang[2,3], Xiangli Nie[2,3*], and Bo Zhang[1,2]
[1]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
zhangyunfeng@amss.ac.cn, {youlinjing2023, yangluyu2023}@ia.ac.cn,
{zhangzhiwei2022, xiangli.nie}@ia.ac.cn, b.zhang@amt.ac.cn

Abstract

Monocular 6-DoF pose estimation is crucial for spacecrafts to achieve precise navigation and positioning, and it has gained increasing attentions in recent years. However, spaceborne imaging quality is heavily influenced by specific factors such as varying illumination conditions, low signal-to-noise ratio and high contrast. In addition, the lack of sufficient labelled space data hampers the performance of deep learning-based pose estimation methods. To overcome these challenges, we propose a novel monocular 6-DoF pose estimation method for spacecrafts utilizing self-iterative optimization and motion consistency. Firstly, we reconstruct an initial 3D spacecraft model using manually annotated 2D keypoints from several images, which can generate the labels of 2D keypoints, heatmaps, and bounding boxes for the entire training set. Subsequently, we train a Multi-task Key-point Prediction Network (MKPNet) model using these label information, and through an iterative optimization process, refine both the 3D model and the performance of MKPNet in predicting 2D keypoints. Additionally, we incorporate temporal information and motion consistency from sequential images to smooth the pseudo-labels of poses predicted by MKPNet during testing. This smoothing process guides the self-training process of the network model, leading to improved generalization and pose estimation accuracy. In the SPARK 2024 Challenge, our method achieves competitive results compared to the state-of-the-art methods and outperforms the baseline regression approaches by a significant margin.

1. Introduction

Monocular 6-DoF pose estimation is crucial for spacecrafts to achieve precise navigation and positioning [10], enabling real-time acquisition of positional and rotational information. This technology allows spacecrafts to autonomously adjust poses in variable space environments, enhancing mission execution efficiency. Moreover, monocular 6-DoF pose estimation provides stable and reliable pose information during space missions, thereby strengthening the robustness and fault tolerance of spacecraft [34].

Recently, various monocular 6-DoF pose estimation methods have been proposed for spacecraft trajectory estimation, which can be broadly classified into two main categories: direct end-to-end approaches and hybrid modular approaches. Direct end-to-end approaches, such as those described in [36], [29], [4], [42] and [7], leverage deep learning techniques to directly estimate the spacecraft's pose from images. Convolutional neural networks (CNNs) and Recurrent neural networks (RNNs) are commonly employed for the end-to-end pose estimation. These approaches enable the learning of complex mappings between images and poses. However, they typically require a substantial amount of annotated training data. Moreover, it is challenging to ensure the generalization ability of these models for spacecraft pose estimation. On the other hand, hybrid modular approaches combine deep learning models with classical computer vision techniques for spacecraft pose estimation. These hybrid algorithms typically consist of three common stages: spacecraft localization, keypoint prediction and pose computation, as discussed in [34]. Keypoint prediction methods [30], [35], [8], [16] involve the detection of 2D keypoints in the monocular images, such as corners. The spacecraft's pose is then estimated by matching these key points across sequential images. Commonly used feature detection and matching algorithms include SIFT [45], SURF [2], and ORB [40], which can effectively extract crucial geometric information from the images. Keypoint prediction techniques usually employ model projection, feature matching, and optimization algo-

*Corresponding author

rithms like RANSAC (RANdom SAmple Consensus) and PnP (Perspective-n-Point) to achieve more accurate pose estimation [24], [14], [19].

To promote the advancement of spacecraft monocular 6-DoF pose estimation, the SPARK2024 Challenge [38] has been organized as part of the AI4Space workshop in conjunction with CVPR 2024. The main objective of SPARK 2024 is to develop data-driven approaches for spacecraft semantic segmentation and trajectory estimation. Spacecraft trajectory estimation seeks to utilize temporal data to estimate the 6-DoF pose of a spacecraft along a given trajectory. However, unlike terrestrial applications, spaceborne imaging quality is heavily influenced by specific factors such as varying illumination conditions, low signal-to-noise ratio and high contrast. In addition, the lack of sufficient labelled space data hampers the performance of deep learning-based pose estimation methods.

To overcome the above challenges, we propose a novel monocular 6-DoF pose estimation method for spacecrafts by utilizing self-iterative optimization and pose smoothing. Firstly, we manually annotate the 2D keypoints of several images to reconstruct a 3D model, which combines with ground truth poses to generate the labels for 2D keypoints, heatmaps, and bounding boxes for the entire training dataset. Then, a multi-task deep network is trained through alternatively iterative optimization with the 3D model to improve the prediction accuracy and refine the 3D keypoints. Finally, the trained model generates pseudo-pose labels on test set, and pose smoothing optimization is employed to refine these labels which guides the model self-training to improve pose estimation accuracy. Our method achieves competitive results compared to the state-of-the-art methods in SPARK2024 Challenge. The main contributions of our method can be summarized as follows:

1) A novel monocular 6-DoF pose estimation method is proposed for spacecrafts utilizing self-iterative optimization and motion consistency, which makes full use of the temporal information of sequential images.

2) The proposed 3D model and MKPNet iterative optimization technique based on the genetic algorithm can continually refine the 3D model and improve the performance of MKPNet in predicting 2D keypoints.

3) The proposed pose smoothing and MKPNet self-training technique can smooth the pose pseudo-labels on test set and perform the self-training process to further improve pose estimation accuracy.

2. Related Work

2.1. Monocular 6-DoF Spacecraft Pose Estimation

The existing methods for monocular 6-DoF pose estimation of spacecraft can be divided into two categories: direct end-to-end approaches and hybrid modular approaches [34]. Direct end-to-end approaches leverage deep learning techniques to directly estimate the spacecraft pose from input data, which can include images, point clouds, or other types of data. In [36], a CNN architecture is proposed to regress the 7D pose vector representing the position and orientation quaternion. In [20], a CNN network is used to classify spacecraft images into discretized pose label classes. On the other hand, hybrid modular approaches focus on predicting key points, which are typically defined based on the spacecraft's CAD model or 3D point cloud, as discussed in [24], [14] and [19]. If CAD model and 3D point clouds are unavailable, techniques such as multi-view triangulation [17], [8], or Structure from Motion [13], can be used to reconstruct a 3D model of the spacecraft that includes 3D keypoints. In some works [30], [35], keypoints are detected from the images, and the spacecraft's poses are then estimated by matching these feature points across different images. Additionally, in [22], the keypoint prediction problem is formulated as a keypoint bounding boxes detection problem. They predict the enclosing bounding boxes over the keypoints along with confidence scores.

2.2. Multi-frame spacecraft pose estimation

Most current spacecraft pose estimation methods primarily rely on individual image frames to estimate the spacecraft's pose, which limits their ability to fully exploit the temporal information present in consecutive images. In contrast, multi-frame spacecraft pose estimation approaches take advantage of the relative motion relationships and temporal information from a sequence of consecutive images to determine the spacecraft's pose, resulting in higher accuracy in pose estimation [34]. In [37], a dataset SPARK2 is introduced to provide pose estimation data as trajectories, which enables the development and evaluation of multi-frame spacecraft pose estimation methods. In [26], a multi-frame pose estimation method is proposed, which aims to achieve smooth and accurate three-dimensional trajectory estimation by enforcing temporal consistency of the estimated 3D positions. To further improve the performance of pose estimation, the temporal convolutional network is introduced in [28], which effectively handles temporal ordering and long-term dependencies in the image sequences. In [39], the ChiNet method is proposed by incorporating Long Short-Term Memory (LSTM) units [15] for modeling sequences of data in spacecraft pose estimation. In summary, multi-frame spacecraft pose estimation approaches make better use of the temporal information, resulting in improved accuracy of pose estimation.

2.3. Self-training

Self-training, also known as self-taught learning, is an approach in semi-supervised learning [11]. It leverages unlabeled data to enhance the performance of a model. The main idea behind self-training is to iteratively train a model

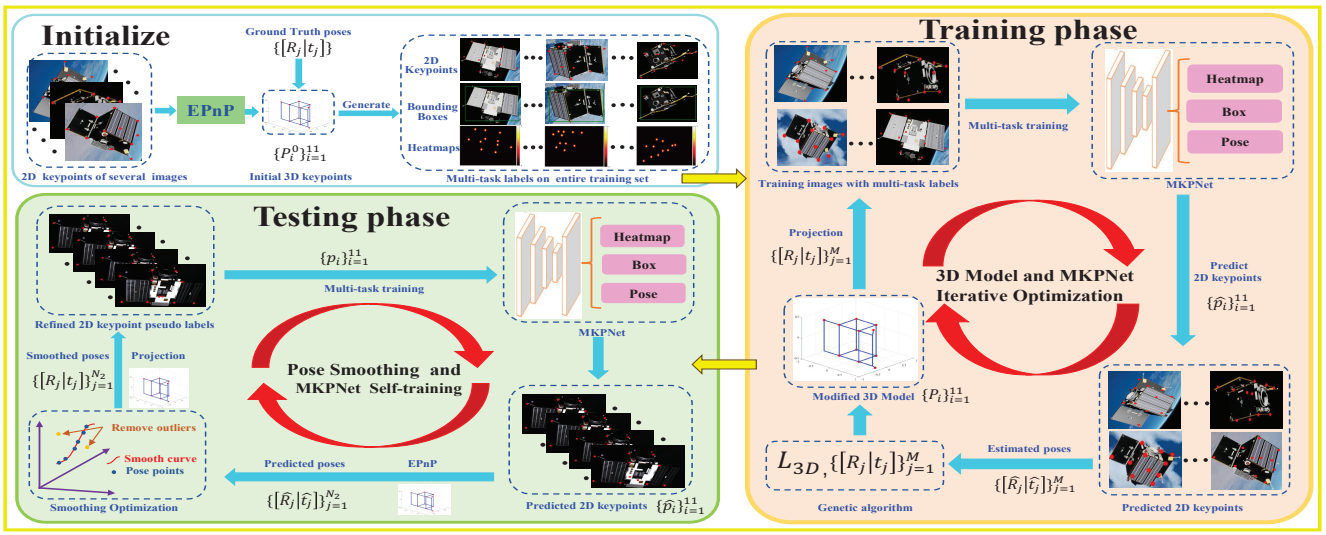


Figure 1. The proposed overall structure of Pose Estimation of Spacecrafts Utilizing Self-iterative Optimization and Motion Consistency.

using a combination of labeled and unlabeled data [1]. In self-training, the model is initially trained with labeled data. It then predicts labels for unlabeled data, treating them as pseudo-labels to expand the labeled dataset. The effectiveness of self-training methods relies on the quality of the generated pseudo-labels. Some approaches have aimed to mitigate the impact of label noise by designing robust loss functions [48] or employing self-label correction techniques [47]. In [35], the label noise is reduced by enforcing consistent pose estimates across the network head. In [32], a CNN is trained to generate pseudo-labels for heatmaps and semantic masks, and the number of inliers from RANSAC is used to eliminate pseudo-labels with low confidence. In [44], segmentation masks are generated from the estimated 3D-mesh model of the spacecraft, and geometric constraints are introduced in the pseudo-label generation process.

3. Method

In this section, we focus on the spacecraft trajectory estimation problem of the SPARK 2024 Challenge. We propose a monocular 6-DoF pose estimation method for spacecrafts by utilizing self-iterative optimization and motion consistency. The overview of the proposed method is presented in Figure 1. Initially, we manually annotate eleven 2D key points in several sequential images and utilize the groundtruth poses of these images and the EPnP algorithm [49] to generate an initial 3D model of spacecraft. Due to annotation errors, we use the entire training dataset to automatically correct the 3D model. During the training process, based on the 3D model and the groundtruth pose information, we can obtain the 2D keypoint label information of the entire training set by employing the EPnP algorithm [49]. This allows us to further generate heatmap and bounding box labels of training set. Then, these labels are used to train the Multi-task Key-point Prediction Network model (MKPNet). Once the MKPNet model is trained, we use it to predict the 2D keypoints and pose information for the entire training set, based on the initial estimation of the 3D model. By comparing the predicted poses with the ground truth poses, we employ genetic algorithm [41] to refine the 3D model. This iterative process is repeated to refine both the 3D model and the MKPNet model, forming a self-iterative optimization process on the training set.

During testing, we utilize the MKPNet network to predict poses for entire test set. To improve the accuracy and enhance the consistency of these predictions, we leverage the motion smoothness technique and temporal information of sequential images to globally smooth the pose data. With the smoothed poses and the optimized 3D model, we generate 2D keypoint pseudo-labels for the test set. Then, MKPNet undergoes the multi-task self-training on test set, which further refines the pose predictions. This self-training process guides the network model's subsequent rounds of self-improvement, allowing it to continuously self-iterative optimize the accuracy of pose estimations.

3.1. 3D Model Initialization and MKPNet Learning

In order to better estimate the pose of spacecraft, inspired by the keypoint-based 3D model reconstruction technique [8, 30], we select several spacecraft images, from multiple trajectory sequences, which capture the spacecraft from various perspectives. We manually annotate eleven 2D keypoints, denoted as $\{p_i\}_{i=1}^{11}$, on the spacecraft in these images. Then, we establish correspondences between the 2D keypoints and 3D keypoints using the known pose information matrix $[R|t] \in R^{3\times4}$ and the camera intrinsic parameter K. By solving the following optimization problem [12],

we obtain the initial 3D model keypoints:

$$P_i = \arg\min_{P_i, s_j} \sum_{i=1}^{11} \sum_{j=1}^{m} \|s_j p_{i_j} - K(R_j P_i + t_j)\|_2. \quad (1)$$

where $i \in \{1, \ldots, 11\}$ represents the i-th keypoint, $P_i \in R^3$ and $p_{i_j} \in R^2$ are the corresponding 3D and 2D coordinates with $j \in \{1, 2, \ldots, m\}$ being the j-th manually annotated 2D image, $[R_j|t_j]$ is the corresponding pose matrix, and $s_j \in R$ represents a scaling factor associated with the projection onto the image plane in each input image. Note that during the process of manual annotation, errors can occur due to the fact that different points in the 3D space of the object may correspond to the same 2D keypoint in the image plane during camera imaging. These errors can lead to inaccuracies in the initial estimation of the 3D model.

To mitigate the bias caused by annotation errors in the initial 3D model estimation, we iteratively optimize the 3D model and the 2D keypoints prediction network in Section 3.2. We leverage the multi-task learning strategy [6] to construct a Multi-task Keypoint Prediction Network model (MKPNet) for enhancing the detection accuracy of spacecraft 2D keypoints. Specifically, these multiple tasks of MKPNet include predicting pre-defined spacecraft keypoints, spacecraft detection and pose estimation. Our deep network architecture consists of a multi-scale sharing feature encoder and multiple prediction heads that perform different tasks. To obtain the label information of 2D keypoints on each training image for supervised learning, we project the 3D model keypoint P_i onto 2D images to get the corresponding 2D keypoint p_i, utilizing the ground truth pose information $[R|t]$ and camera parameter K. The label information satisfies the following relationship:

$$sp_i = K(RP_i + t),\ i = \{1, 2, \ldots, 11\}. \quad (2)$$

For the keypoint detection task, we firstly generate the corresponding heatmap $h_i \in R^{11\times128\times192}$ according to the keypoint p_i. We extract features from the input image x by a feature extraction network F [43]. We utilize a heatmap prediction head H as in [29] to generate the heatmap $\hat{h}_i = H(F(x)) \in h_i \in R^{11\times128\times192}$ for predicting the keypoint $\hat{p}_i$. During training, the loss between the predicted heatmaps and the ground truth heatmaps is based on the mean squared error as:

$$L_H(h, \hat{h}) = \sum_{i=1}^{11} ||h_i - \hat{h}_i||_2. \quad (3)$$

For the detection task, we determine the bounding box b of the spacecraft for each image as follows:

$$[(\min_i(p_i[0]), \min_i(p_i[1])), (\max_i(p_i[0]), \max_i(p_i[1]))]. \quad (4)$$

where $p_i[0]$ and $p_i[1]$ denote the x-coordinates and y-coordinates, respectively. Then, we use a detection head B as in [29], [5] to predict the bounding box $\hat{b} = B(F(x))$. During training, we measure the error between the predicted and ground truth bounding boxes by the IOU [50] loss $L_B(b, \hat{b}) = IoU(b, \hat{b})$. For the pose estimation task, we initially apply a threshold ϵ_1 to filter the predicted heatmap $\hat{h}_i$ of the 2D keypoint $\hat{p}_i$. We select keypoints with high confidence by forming a index set $I = \{i|max(\hat{h}_i) \geq \epsilon_1\}$. To ensure that the EPnP [21] algorithm can accurately estimate the spacecraft's pose, we iteratively adjust the threshold (i.e., $\epsilon_1 = \epsilon_1 - 0.01$) until at least 6 keypoints are retained ($len(I) \geq 6$). The spacecraft poses is estimated by the EPnP algorithm as follows:

$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_{i\in I} \|s\hat{p}_i - K(RP_i + t)\|_2. \quad (5)$$

The loss between the predicted poses and the ground truth poses is defined as:

$$L_{Pose} = 2\arccos(|\langle q, \hat{q}\rangle|) + \frac{\|t - \hat{t}\|_2}{\|t\|_2}. \quad (6)$$

where $\hat{q} = Q(\hat{R})$ with the function $Q(\cdot)$ converting a rotation matrix to a quaternion, q and t are the ground truth quaternion and translation vector and $\langle\cdot\rangle$ denotes the dot product.

The overall loss function for the training process of the MKPNet model is given by:

$$L = L_H + L_B + L_{Pose}. \quad (7)$$

3.2. 3D Model and MKPNet Iterative Optimization

In this section, we propose an iterative optimization strategy of the 3D model and MKPNet, aiming to refine the 3D model and improve the accuracy of the pose estimated by MKPNet. The initial 3D model is reconstructed only based on a limited number of images in training set, which result in significant errors in 3D points estimation. To make full use of the data from the entire training set and reduce the reconstruction error, we incorporate 3D genetic optimization [41]. In this optimization process, we keep the 2D keypoints $\{\hat{p}_i\}_{i=1}^{11}$ obtained from MKPNet fixed and treat the spatial coordinates of the 3D points $\{P_i\}_{i=1}^{11}$ as optimization variables. The objective is to minimize the pose error L_{pose} across the entire training set $X = \{(x_j, y_j)\}_{j=1}^{N_1}$, where y_j represents the ground truth labels of the quaternion q_j and translation t_j.

Specifically, we randomly sample M images from the training set. For each image x_j, we use MKPNet to predict the 2D keypoints $\{\hat{p}_{i_j}\}_{i=1}^{11}$. Then, combining these 2D keypoints with Equation (1), we can obtain the initial estimated 3D keypoints $\{P_i^0\}_{i=1}^{11}$, where $P_i^0 \in R^3$. Then, we use Equation (5) to get the pose $\hat{R}_j$ and $\hat{t}_j$ of the image under 3D keypoints variables. To optimize the 3D keypoints

Algorithm 1: 3D Model and MKPNet Iterative Optimization on Training Set

Input: Training images X_{train}, pose labels Y_{train} and camera parameter K;

Output: MKPNet model $\mathcal{M}(.;\hat{\theta})$ and the optimized 3D Keypoints $\{P_i\}_{i=1}^{11}$;

Initialize: Manually annotate a few images to initialize 3D keypoints $\{P_i^0\}_{i=1}^{11}$ by Equation (1);

for $k = 0$ **to** K_1 **do**

- Compute the 2D keypoints label p_i^k based on 3D keypoint P_i^k and the projection Equation (2);
- Obtain the heatmaps $\{h_i^k\}_{i=1}^{11}$ and bounding box b^k according to $\{p_i^k\}_{i=1}^{11}$;
- Train MKPNet $\mathcal{M}(.;\theta^k)$ based the labels $\{h_i^k\}_{i=1}^{11}, b^k, y_j$;
- Predict the 2D keypoints $\{\{\hat{p}_{i_j}\}_{i=1}^{11}\}_{j=1}^{M}$ by MKPNet $\mathcal{M}(.;\theta^k)$ for randomly selected images $\{(x_j, y_j)\}_{j=1}^{M}$;
- Optimize 3D keypoints $\{P_i\}_{i=1}^{11}$ based on these predicted 2D keypoints $\{\{\hat{p}_{i_j}\}_{i=1}^{11}\}_{j=1}^{M}$ and pose label $\{y_j\}_{j=1}^{M}$ by genetic algorithm to solve the problem (8);
- Let $k = k + 1$;

variables $\{P_i\}_{i=1}^{11}$, we use the pose loss function as the objective function L_{3D} as follows:

$$\min_{\{P_i\}_{i=1}^{11}} \sum_{j=1}^{M} \{2 \arccos\left(|\langle q_j, Q(\hat{R}_j)\rangle|\right) + \frac{\|t_j - \hat{t}_j\|_2}{\|t_j\|_2}\}. \quad (8)$$

We use a genetic algorithm [41] to solve the above optimization problem to obtain the refined 3D model $\{P_i\}_{i=1}^{11}$. Next, using the refined 3D model, we can obtain more accurate 2D keypoints $\{p_i\}_{i=1}^{11}$ of the training images based on the Equation (2). These 2D keypoints can then be utilized to retrain the MKPNet model. As a result, the 3D model and the MKPNet model can undergo alternative iterative optimization, continuously enhancing both the predictive capability of MKPNet for predicted 2D keypoints $\{\hat{p}_i\}_{i=1}^{11}$ and the accuracy of the 3D model $\{P_i\}_{i=1}^{11}$. The above procedure is presented in Algorithm 1.

3.3. Pose Smoothing and MKPNet Self-training

3.3.1 Pose Smoothing Optimization

During the test phase, we utilize the trained MKPNet $\mathcal{M}(.;\theta)$ to infer the positions of 2D keypoints $\{\hat{p}_{i_j}\}_{i=1}^{11}$for each test image x_j in test dataset X_{test}. These predicted 2D keypoints are then combined with the optimized 3D model keypoints $\{P_i\}_{i=1}^{11}$ to estimate the initial pose $[\hat{R}_j|\hat{t}_j]$ of test data using the EPnP algorithm [21].

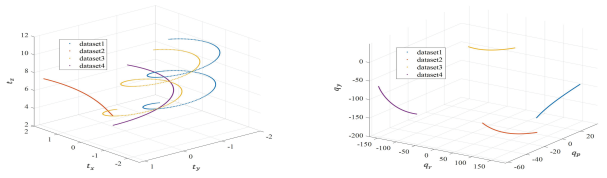


Figure 2. The pose distributions of sequential images in training dataset. Left: Translation variables. Right: Rotation variables.

To further enhance the stability and accuracy of pose estimation, we perform pose smoothing according to the motion consistency of sequential images. Based on the observation that poses of consecutive frames exhibit continuity in pose space, we present the continuous smoothness of pose distributions in Figure 2. In the "translation" space, we visualize the distribution of points formed by the displacement variables t_x, t_y, and t_z. It can be seen that this distribution is represented as a smooth curve in space, which indicates that the translations between consecutive frames are consistent. For rotations, we use the transformation function O to convert the quaternion q of rotations into Euler angles [9] $O(q)$ and visualize the distribution of points in the "rotation" space formed by the variables $Roll$, $Pitch$, and Yaw which abbreviated as q_r, q_p, q_y. Similarly, we observe a continuous smoothness in this rotation distribution for each sequential dataset. Hence, taking into account the temporal information present in the sequential data, we extract pose information $\{t_x, t_y, t_z, q_r, q_p, q_y\}$ from a given sequence of images $X_{test} = \{x_j\}_{j=1}^{N_2}$ in the test set. To analyze and smooth the data, we perform fitting on each translation component of $\hat{t}_j$ using the ordinal variable j. The fitted data pairs correspond to $(j, \hat{t}_{x_j}), (j, \hat{t}_{y_j}), (j, \hat{t}_{z_j})$. Additionally, due to the strong correlation between the rotational components $\hat{q}_{r_j}$, $\hat{q}_{p_j}$, and $\hat{q}_{y_j}$, we fit the rotational data pairs as $(\hat{q}_{r_j}, \hat{q}_{p_j})$ and $(\hat{q}_{r_j}, \hat{q}_{y_j})$. Figure 2 demonstrates that the distribution of each pose component appears to be periodic. Consequently, we select the following smoothing fitting function:

$$u = f(v) = a_0 + \sum_{n=1}^{10} [a_n cos(nvw) + b_n sin(nvw)]. \quad (9)$$

where $\{a_n\}_{n=0}^{10}$, $\{b_n\}_{n=0}^{10}$, w are the parameters to be estimated through the fitting process. With the given definition, we can unify the smoothing optimization process for translation and rotation into a fitting process for the five data pairs, outlined as follows:

(1) **Smooth function Fitting**: For each set of fitted data pair $\{(v_j, u_j)\}_{j=1}^{N_2}$, we employ the least squares method [3] to fit the function f and obtain the smooth function curve $\hat{f}$;

(2) **Outlier Removal and Smooth function Correction**: For each fitted data pair (v_j, u_j), we check if the distance $||u_j - \hat{f}(v_j)||_2$ between the point and the fitted curve exceeds the threshold ϵ, the point (v_j, u_j) is identified as an

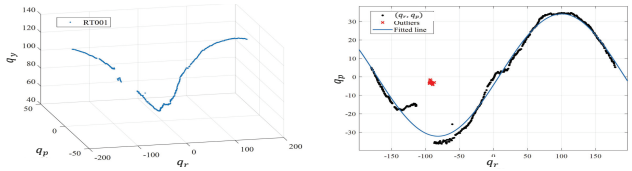


Figure 3. The distribution (left) of rotation and the fitted curve (right) of (q_r, q_p) .

outlier, as shown by the red points in Figure 3. We remove all identified outliers and repeat the previous steps until a smooth fitted curve is obtained without any outliers;

(3) **Application of Smooth Functions**: We apply the obtained smooth fitted curve to all data points (v_j, u_j), replacing u_j with $\hat{f}(v_j)$. This process results in a set of smoothed data pairs.

We visualized pose variables distribution of RT001 in test datasets and the corresponding fitted curve for $(\hat{q}_{r_j}, \hat{q}_{p_j})$ as shown in Figure 3. By following the aforementioned procedure, we obtain five smooth functions for each sequence of images: f_x, f_y, f_z, f_p, f_r. These functions are obtained by fitting the sets of five data pairs: $\{(j, \hat{t}_{x_j})\}_{j=1}^{N_2}$,$\{(j, \hat{t}_{y_j})\}_{j=1}^{N_2}$,$\{(j, \hat{t}_{z_j})\}_{j=1}^{N_2}$,$\{(\hat{q}_{r_j}, \hat{q}_{p_j})\}_{j=1}^{N_2}$, $\{(\hat{q}_{r_j}, \hat{q}_{y_j})\}_{j=1}^{N_2}$. Thus, we can perform smoothing optimization on the initial spacecraft's pose $O(\hat{q}_j)$ and $\hat{t}_j$ to obtain the smoothed pose estimations $O(q_j), t_j$:

$$O(q_j), t_j = Smooth(O(\hat{q}_j), \hat{t}_j) = \begin{cases} t_{x_j} = f_x(j), t_{y_j} = f_y(j), t_{z_j} = f_z(j) \\ q_{r_j} = \hat{q}_{r_j}, q_{p_j} = f_p(\hat{q}_{r_j}), q_{r_j} = f_y(\hat{q}_{r_j}) \end{cases} . \quad (10)$$

Note that the component q_{r_j} will be further corrected during the subsequent self-training process. Through this smoothing optimization approach of pose fitting function, we effectively identify and eliminate outliers, leading to improved accuracy and robustness of the fitting procedure.

3.3.2 MKPNet Self-training

On the test set, after obtaining the smoothed poses q_j and t_j by leveraging temporal information, we project the optimized 3D keypoints $\{P_i\}_{i=1}^{11}$in Algorithm 1 onto each test image using equation (2), resulting in generated 2D keypoints pseudo-labels $\{p_{i_j}\}_{i=1}^{11}$ for each test image x_j. Then, we utilize the 2D keypoint pseudo-labels along with the optimized poses q_j and t_j of the image x_j to generate additional pseudo-labels, including heatmap pseudo-labels $\{h_{i_j}\}_{i=1}^{11}$ and bounding box pseudo-labels b_j. We utilize these pseudo-labels $\{q_j, t_j, p_{i_j}, h_{i_j}, b_j\}$ to retrain the MKPNet model on test set, which aims to make full use of the temporal information of sequential images and enhance the transferability of the model on test data.

In summary, we firstly predict the 2D keypoints of test data using the MKPNet model learned on training set. Then, the 2D keypoints along with the 3D points is utilized to compute the poses. Next, the motion consistency of sequential images is considered to smooth the pose data. Using the refined poses, the multi-task pseudo-labels can be generated to retrain the MKPNet model. This self-training process iterates multiple times, allowing the MKPNet model to self-iterative optimize the accuracy of pose estimation. The above procedure is presented in Algorithm 2.

Algorithm 2: Pose Smoothing and MKPNet Self-training on Test Set

Input: Test images X_{test}, the pre-trained MKPNet $\mathcal{M}(.;\theta)$ and the optimized 3D keypoints ;
Output: Estimated pose $\hat{q}_j^*, \hat{t}_j^*$;

for $k = 0$ **to** K_2 **do**
- Predict 2D keypoints $\{\hat{p}_{ij}^k\}_{i=1}^{11}$ of the test set using the pre-trained MKPNet $\mathcal{M}(.;\theta^k)$;
- Obtain predicted poses $[\hat{q}_j^k, \hat{t}_j^k]$ using the optimized 3D keypoints $\{P_i\}_{i=1}^{11}$ and 2D keypoints $\{\hat{p}_{ij}^k\}_{i=1}^{11}$ by Equation (5);
- Smooth poses $[\hat{q}_j^k, \hat{t}_j^k]$ according to Equation (10) to obtain refined poses $[q_j^k, t_j^k]$;
- Utilize refined poses $[q_j^k, t_j^k]$to generate multi-task pseudo-labels $\{p_{ij}^k, h_{ij}^k, b_j^k\}$;
- Retrain MKPNet $\mathcal{M}(.;\theta^k)$ using these pseudo-labels $\{q_j^k, t_j^k, p_{ij}^k, h_{ij}^k, b_j^k\}$ to update $\theta^k \rightarrow \theta^{k+1}$;

4. Experiments

4.1. Data Set and Metrics

Our experiments utilize the Spacecraft Trajectory Estimation dataset [38] in Stream 2 of SPARK 2024 Challenge. This dataset utilizes data synthetically simulated with a state-of-the-art rendering engine and collected from the Zero-Gravity Laboratory (Zero-G Lab) facility [33] . The synthetic dataset is created using the Unity3D game engine as a simulation environment, capable of generating visually realistic data of the target model. The virtual target was programmed to follow predefined trajectories, while the intrinsic camera parameters remained fixed [27]. The training set of this dataset comprises 100 trajectories, each consisting of 300 RGB images with pose annotations. The test set comprises 4 trajectories and includes 2123 images.

The metrics of SPARK 2024 Challenge are going to sum the relative position error and the geodesic orientation error for each frame, then average these scores over all the frames and trajectories. The metric is largely inspired by the SPEED+ [31].

4.2. Experimental Setup

Our model is implemented using the PyTorch framework. For training, we utilize the AdamW optimizer [25] on the

Methods	ϕ	Translation_error	Orientation_error	Pose_error
EfficientPose-GN	6	0.0671	2.2838	2.3013
EfficientPose-BN	6	0.1039	1.7858	1.8119
EfficientPose-GN	3	0.0981	2.0694	2.0942
EfficientPose-BN	3	**0.0700**	**1.6947**	**1.7132**
Ours-BN	3	**0.0335**	**0.0843**	**0.0934**

Table 1. Comparison with direct regression model.

training dataset. The initial learning rate is set to $1e-3$, which decays by a factor of 0.1 at the 15th and 25th epochs to improve convergence. During the self-training iteration on test set, the learning rate is set to $1e-4$ to fine-tune the model. The hyperparameters ϵ_1, ϵ are set to $5e-1, 1e-1$. All experiments in this paper were conducted on a computer server equipped with two Intel Xeon 6330N @2.2GHz, 256 GB of RAM and four NVIDIA GeForce RTX 4090 GPUs.

4.3. Experimental Results

The objective of Stream 2 in the SPARK 2024 Challenge is to estimate spacecraft trajectory pose by utilizing knowledge of the space environment. we employed EfficientPose [5] as the backbone which is highly accurate, efficient and scalable over a wide range of computational resources, inherited from the EfficientNet [43] which allows us to more effectively scale network depth, width, and resolution parameters. Additionally, in order to enhance the feature extraction and pose estimation capabilities of the network model for 2D spacecraft imagery, we integrate the bidirectional Feature Pyramid Network [23] and multi-task heads, similar to [29].

Comparison with direct regression model with different backbone settings. Initially, we employ the EfficientPose network under different settings to directly regress the spacecraft's pose, and then we select an appropriate backbone setting. Compared Group Normalization (GN) layers [46] with Batch Normalization (BN) layers [18], the GN layers are designed to be batch-agnostic. In the SPARK 2024 Stream 2 Challenge, we compared the performance of networks with BN and GN structures for spacecraft 6DoF pose estimation, as shown in Table 1. It can be seen that the pose estimation network with the BN structure exhibits higher performance than that with GN. Additionally, increasing the model's parameter complexity, as indicated by $\phi = 6$, results in overfitting on the training data, leading to decreased performance in spacecraft pose estimation on the test set. Based on these observations, our MKPNet model uses network structures with Batch Normalization (BN) layers and $\phi = 3$ for spacecraft pose estimation. Furthermore, in Table 1, when comparing with these baseline regression models, we observe a significant improvement in accuracy of pose estimation achieved by our proposed method. This improvement validates the effectiveness of the 3D model iterative optimization and pose smoothing optimization techniques introduced in our method.

Team	Translation_error	Orientation_error	Pose_error
csu_nuaa_pang	0.0252	**0.0187**	**0.0252**
lucca	**0.0243**	0.0448	0.0508
juanqilai (ours)	0.0335	0.0843	0.0934
igodrr	0.0823	0.7214	0.7417
yanlj	0.0335	1.0362	1.0454
dwoiwode	0.0739	1.4460	1.4651
nalixvignola	0.1303	2.1407	2.1741

Table 2. The results on the test set of Spacecraft Trajectory Estimation.

Comparison with SPARK 2024 Challenge Results. Table 2 presents the comparison results in SPARK 2024 Challenge and our proposed method ranks third. For the approaches with test errors above 1.0, we speculate they utilize the direct regression for spacecraft pose estimation. This inference is supported by the fact that their results are comparable to the data direct regression accuracy shown in Table 1. Comparing with these direct regression approaches, our results exhibit a substantial improvement in accuracy on the test set, almost tenfold higher. That is because our method attempts to regress the 2D keypoints and incorporates two components: the 3D model and MKPNet iterative optimization and the pose smoothing and MKPNet self-training operation, on top of the network backbone. This significant enhancement serves as evidence of the effectiveness of our proposed approach. Compared to the top two methods, our approach exhibits comparable translational error but relative lower rotational error. Actually, our method can be further improved by increasing the number of iterations for the 3D model optimization and pose smoothing optimization. In the following ablation study, we will present a more comprehensive analysis of our method.

4.4. Ablation Studies and Analysis

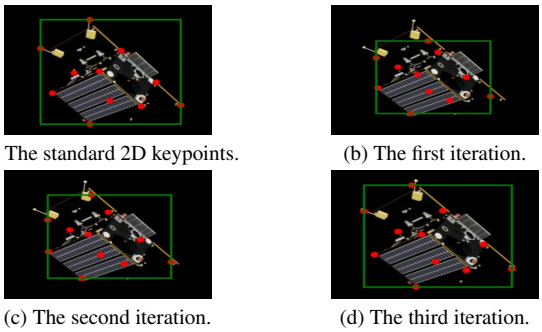
The proposed framework for monocular 6-DoF pose estimation in spacecraft incorporates two main components: 3D Model and MKPNet Iterative Optimization (3MIO), and Pose Smoothing and MKPNet Self-training (PSMS). To assess the effectiveness of the two components, we conduct an ablation study by incorporating 3MIO and PSMS individually into the baseline model. Table 3 presents the quantitative results of the four ablation settings on the SPARK 2024 Stream 2 dataset [38]. It can be seen that both of the two components can improve the accuracy of the pose estimation.

4.4.1 Effectiveness of 3MIO

Table 4 presents the comparison of test errors of the baseline model with 3MIO at different iteration numbers. It can be observed that as the iteration numbers increasing of the 3MIO, the pose error progressively decreases. To visualize

3MIO	PSMS	Translation_error	Orientation_error	Pose_error
✗	✗	0.0700	1.6947	1.7132
✓	✗	0.0567	0.2518	0.2671
✗	✓	0.1254	0.7053	0.7378
✓	✓	**0.0335**	**0.0843**	**0.0934**

Table 3. Ablation studies for 3MIO and PSMS.



(a) The standard 2D keypoints. (b) The first iteration.

(c) The second iteration. (d) The third iteration.

Figure 4. The 2D keypoints and bounding boxes corresponding to the 3D keypoints at different iteration numbers of 3MIO.

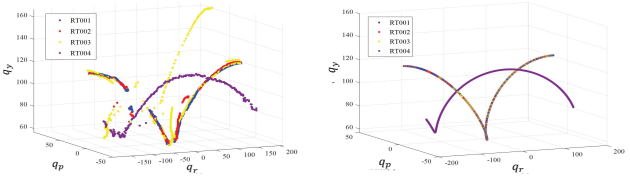
Iterations	Translation_error	Orientation_error	Pose_error
0	0.0700	1.6947	1.7132
1	0.0638	0.8483	0.8647
2	0.1126	0.5643	0.5926
3	**0.0567**	**0.2518**	**0.2671**

Table 4. Comparison of test errors at different iteration numbers of 3MIO.

the effects at different iterations, we project the optimized 3D keypoints onto 2D images using pose labels to obtain the corresponding 2D keypoints positions and spacecraft detection bounding boxes. Figure 4 shows the visual comparison of 2D keypoints and bounding boxes corresponding to the 3D keypoints at different iteration numbers of 3MIO. It can be seen that the 3D keypoints are corrected continually during the 3MIO iterative process. Utilizing the corrected 3D keypoints to generate more accurate 2D keypoints for model training can effectively mitigate the significant bias in pose predictions caused by inaccurate initial 3D model. Compared with the results without correcting the 3D model, the pose error of three 3MIO iterations decreases significantly, which demonstrates the effectiveness of 3MIO.

4.4.2 Effectiveness of PSMS

Figure 5a shows the distribution of the initial predicted rotation on test set , while Figure 5b presents the distribution of rotation after pose smoothing. We can see that there are some noticeable outliers or perturbation points in the rotation distribution in Figure 5a, while these points are smoothed by using PSMS in Figure 5b. Table 5 presents the numerical the test errors of of MKPNet at different it-



(a) Before smoothing. (b) After smoothing.

Figure 5. The distribution of rotation variables before and after smoothing.

Iterations	Translation_error	Orientation_error	Pose_error
1	0.0345	0.3396	0.3488
2	0.0335	0.0918	0.1030
3	**0.0335**	**0.0843**	**0.0934**

Table 5. Comparison of test errors of MKPNet at different iteration numbers of PSMS.

eration numbers of PSMS. It can be observed that in the initial round of iterations, the incorporation of PSMS leads to a significant improvement in accuracy for both translation and rotation. As the iterations progress, the accuracy becomes relatively stable. The PSMS technology results in a remarkable reduction of prediction error compared to the initial predicted poses without PSMS, which provides a strong evidence for the effectiveness of the PSMS component. It is important to note that the effectiveness of PSMS technology in improving pose prediction accuracy is limited by the accuracy of 3D keypoints estimation. While PSMS primarily focuses on optimizing predicted poses by smoothing out noise, it cannot correct the bias introduced by errors in the estimation of 3D keypoints. Therefore, we combine 3MIO with PSMS technology to perform 3D model and pose correction, aiming to achieve a comprehensive optimization of pose prediction accuracy.

5. Conclusion

In this paper, a novel monocular 6-DoF pose estimation method has been proposed for spacecrafts utilizing self-iterative optimization and motion consistency. It includes two main components: 3D model and MKPNet iterative optimization, and pose smoothing and MKPNet self-training. The iterative optimization process continually refines the 3D model and improve the performance of MKPNet in predicting 2D keypoints. The proposed pose smoothing and self-training technique further enhances the generalization capability and pose estimation accuracy of MKPNet model. Experimental results validate the effectiveness of the proposed method and our method performs well on the SPARK 2024 dataset.

Acknowledgement. This work was partly supported by the National Natural Science Foundation of China (NNSFC) under Grant 62076241, and partly sponsored by Beijing Nova Program under Grant 20220484070.

References

[1] Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, Emilie Devijver, and Yury Maximov. Self-training: A survey. *arXiv preprint arXiv:2202.12040*, 2022. 3

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 1

[3] Åke Björck. Least squares methods. *Handbook of numerical analysis*, 1:465–652, 1990. 5

[4] Kevin Black, Shrivu Shankar, Daniel Fonseka, Jacob Deutsch, Abhimanyu Dhir, and Maruthi R Akella. Real-time, flight-ready, non-cooperative spacecraft pose estimation using monocular imagery. *arXiv preprint arXiv:2101.09553*, 2021. 1

[5] Yannick Bukschat and Marcus Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach. *arXiv preprint arXiv:2011.04307*, 2020. 4, 7

[6] Rich Caruana. Multitask learning. *Machine learning*, 28: 41–75, 1997. 4

[7] Lorenzo Pasqualetto Cassinis, Alessandra Menicucci, Eberhard Gill, Ingo Ahrns, and Manuel Sanchez-Gestido. On-ground validation of a cnn-based monocular pose estimation system for uncooperative spacecraft: Bridging domain shift in rendezvous scenarios. *Acta Astronautica*, 196:123–138, 2022. 1

[8] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat-Jun Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1, 2, 3

[9] James Diebel et al. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006. 5

[10] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in aerospace sciences*, 68:1–26, 2014. 1

[11] S Fralick. Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*, 13(1):57–64, 1967. 2

[12] Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming, version 2.1, 2014. 3

[13] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[14] Ying He, Jun Yang, Kejiang Xiao, Chao Sun, and Jia Chen. Pose tracking of spacecraft based on point cloud dca features. *IEEE Sensors Journal*, 22(6):5834–5843, 2022. 2

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[16] Wenxiu Huan, Mingmin Liu, and Qinglei Hu. Pose estimation for non-cooperative spacecraft based on deep learning. In *2020 39th Chinese Control Conference (CCC)*, pages 3339–3343. IEEE, 2020. 1

[17] Yurong Huo, Zhi Li, and Feng Zhang. Fast and accurate spacecraft pose estimation from single shot space imagery using box reliability and keypoints existence judgments. *IEEE Access*, 8:216283–216297, 2020. 2

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 7

[19] Guohua Kang, Qi Zhang, Jiaqi Wu, and Han Zhang. Pose estimation of a non-cooperative spacecraft without the detection and recognition of point cloud features. *Acta Astronautica*, 179:569–580, 2021. 2

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[21] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009. 4, 5

[22] Kecen Li, Haopeng Zhang, and Chenyu Hu. Learning-based pose estimation of non-cooperative spacecrafts with uncertainty prediction. *Aerospace*, 9(10):592, 2022. 2

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 7

[24] Xiang Liu, Hongyuan Wang, Xinlong Chen, Weichun Chen, and Zhengyou Xie. Position awareness network for noncooperative spacecraft pose estimation based on point cloud. *IEEE Transactions on Aerospace and Electronic Systems*, 59 (1):507–518, 2022. 2

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[26] Mohamed Adel Musallam, Miguel Ortiz Del Castillo, Kassem Al Ismaeil, Marcos Damian Perez, and Djamila Aouada. Leveraging temporal information for 3d trajectory estimation of space objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3816–3822, 2021. 2

[27] Mohamed Adel Musallam, Vincent Gaudilliere, Enjie Ghorbel, Kassem Al Ismaeil, Marcos Damian Perez, Michel Poucet, and Djamila Aouada. Spacecraft recognition leveraging knowledge of space environment: simulator, dataset, competition design and analysis. In *2021 IEEE International Conference on Image Processing Challenges (ICIPC)*, pages 11–15. IEEE, 2021. 6

[28] Mohamed Adel Musallam, Arunkumar Rathinam, Vincent Gaudillière, Miguel Ortiz del Castillo, and Djamila Aouada. Cubesat-cdt: a cross-domain dataset for 6-dof trajectory estimation of a symmetric spacecraft. In *European Conference on Computer Vision*, pages 112–126. Springer, 2022. 2

[29] Tae Ha Park and Simone D'Amico. Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap. *Advances in Space Research*, 2023. 1, 4, 7

[30] Tae Ha Park, Sumant Sharma, and Simone D'Amico. Towards robust learning-based pose estimation of noncooperative spacecraft. *arXiv preprint arXiv:1909.00392*, 2019. 1, 2, 3

[31] Tae Ha Park, Marcus Märtens, Gurvan Lecuyer, Dario Izzo, and Simone D'Amico. Speed+: Next-generation dataset for spacecraft pose estimation across domain gap. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–15. IEEE, 2022. 6

[32] Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, and Simone D'Amico. Satellite pose estimation competition 2021: Results and analyses. *Acta Astronautica*, 204:640–665, 2023. 3

[33] Leo Pauly, Michele Lynn Jamrozik, Miguel Ortiz del Castillo, Olivia Borgue, Inder Pal Singh, Mohatashem Reyaz Makhdoomi, Olga-Orsalia Christidi-Loumpasefski, Vincent Gaudilli'ere, Carol Martinez, Arunkumar Rathinam, Andreas Hein, Miguel Olivares-Mendez, and Djamila Aouada. Lessons from a space lab: An image acquisition perspective. *International Journal of Aerospace Engineering*, 2023: 9944614, 2023. 6

[34] Leo Pauly, Wassim Rharbaoui, Carl Shneider, Arunkumar Rathinam, Vincent Gaudillière, and Djamila Aouada. A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects. *Acta Astronautica*, 2023. 1, 2

[35] Juan Ignacio Bravo Pérez-Villar, Álvaro García-Martín, and Jesús Bescós. Spacecraft pose estimation based on unsupervised domain adaptation and on a 3d-guided loss combination. In *European Conference on Computer Vision*, pages 37–52. Springer, 2022. 1, 2, 3

[36] Thaweerath Phisannupawong, Patcharin Kamsing, Peerapong Torteeka, Sittiporn Channumsin, Utane Sawangwit, Warunyu Hematulin, Tanatthep Jarawan, Thanaporn Somjit, Soemsak Yooyen, Daniel Delahaye, et al. Vision-based spacecraft pose estimation via a deep convolutional neural network for noncooperative docking operations. *Aerospace*, 7(9):126, 2020. 1, 2

[37] A Rathinam, V Gaudilliere, MA Mohamed Ali, M Ortiz Del Castillo, L Pauly, and D Aouada. Spark 2022 dataset: Spacecraft detection and trajectory estimation, 2022. 2

[38] Arunkumar Rathinam, Mohamed Adel Mohamed Ali, Vincent Gaudilliere, and Djamila Aouada. SPARK 2024: Datasets for Spacecraft Semantic Segmentation and Spacecraft Trajectory Estimation, 2024. 2, 6, 7

[39] Duarte Rondao, Nabil Aouf, and Mark A Richardson. Chinet: Deep recurrent convolutional learning for multimodal spacecraft pose estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 2022. 2

[40] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 1

[41] Jonathan Shapiro. Genetic algorithms in machine learning. In *Advanced Course on Artificial Intelligence*, pages 146–168. Springer, 1999. 3, 4, 5

[42] Sumant Sharma, Connor Beierle, and Simone D'Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–12. IEEE, 2018. 1

[43] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 4, 7

[44] Zi Wang, Minglin Chen, Yulan Guo, Zhang Li, and Qifeng Yu. Bridging the domain gap in satellite pose estimation: A self-training approach based on geometrical constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 2023. 3

[45] Jian Wu, Zhiming Cui, Victor S Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. A comparative study of sift and its variants. *Measurement science review*, 13(3): 122–131, 2013. 1

[46] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 7

[47] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424, 2021. 3

[48] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 3

[49] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Astrom, and Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2344–2351, 2013. 3

[50] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 international conference on 3D vision (3DV)*, pages 85–94. IEEE, 2019. 4