69^{th} International Astronautical Congress (IAC), Bremen, Germany, 1-5 October 2018.

IAC–18–D1.6.9.x46801

Monocular-Based Pose Determination of Uncooperative Known and Unknown Space Objects

Vincenzo Capuano*, Kyunam Kim, Juliette Hu, Alexei Harvard, and Soon-Jo Chung

Graduate Aerospace Laboratories, California Institute of Technology (GALCIT), Pasadena, California, United States, vcapuano@caltech.edu
* Corresponding Author

Abstract

In order to support spacecraft proximity operations, such as on-orbit servicing and spacecraft formation flying, several vision-based techniques exist to determine the relative pose of an uncooperative orbiting object with respect to the spacecraft. Depending on whether the object is known or unknown, a shape model of the orbiting target object may have to be constructed autonomously by making use of only optical measurements. In this paper, we investigate two vision-based approaches for pose estimation of uncooperative orbiting targets: one that is general and versatile such that it does not require a priori knowledge of any information of the target, and the other one that requires knowledge of the target's shape geometry. The former uses an estimation algorithm of translational and rotational dynamics to sequentially perform simultaneous pose determination and 3D shape reconstruction of the unknown target, while the latter relies on a known 3D model of the target's geometry to provide a point-by-point pose solution. The architecture and implementation of both methods are presented and their achievable performance is evaluated through numerical simulations. In addition, a computer vision processing strategy for feature detection and matching and the Structure from Motion (SfM) algorithm for on-board 3D reconstruction are also discussed and validated by using a dataset of images that are synthetically generated according to a chaser/target relative motion in Geosynchronous Orbit (GEO).

Keywords: Vision-based Navigation, Spacecraft, Pose Determination, 3D Reconstruction

Keywords: Vision-based Navigation, Spacecraft, Pose Determination

1 Introduction

Many of 15,000 known and catalogued orbiting objects are classified as uncooperative objects [1]. In addition to the ones catalogued, there are also a significant number of other orbiting objects too small to be tracked by ground stations whose geometric appearance is unknown. For example, by using radar and optical sensors in the Space Surveillance Network (SSN), the US military can only track objects larger than 1 m in size in GEO (Geosynchronous Earth Orbit) or larger than 5-10 cm in size in LEO (Low Earth Orbit) [1].

In missions of On-Orbit Servicing (OOS), such as manipulation, relocation, active debris removal (ADR), or simply observation of these objects, it is necessary to operate in close proximity and perform relative navigation maneuvers. Communication delays or simply poor coverage make the use of ground-based control commands infeasible. Hence, relative navigation has to be performed autonomously, without control from ground stations. The research study [2] provides a review of the state-of-the-art approaches and algorithms specifically developed for on-board spacecraft pose determination, when processing electro-optical (EO) sensor observations. EO sensors include all devices able to collect radiation in the optical spectrum (0.01 μm to 1000 μm). For spacecraft pose determination, typically either passive monocular/stereo camera or active LIDAR (Light Detection and Ranging) systems are used.

Other works [3, 4, 5, 6, 7, 8] focus on Global Navigation Satellites System (GNSS)–based or more generally Radio Frequency (RF)–based relative navigation, assuming that the orbiting target object is equipped with a GNSS receiver and a communication link or respectively a RF transmitting and receiving antenna, which means it is actively cooperative. As well as for OOS, the RF/GNSS-based approach can actually be very convenient in Formation Flying (FF) missions, where each spacecraft of the forma-

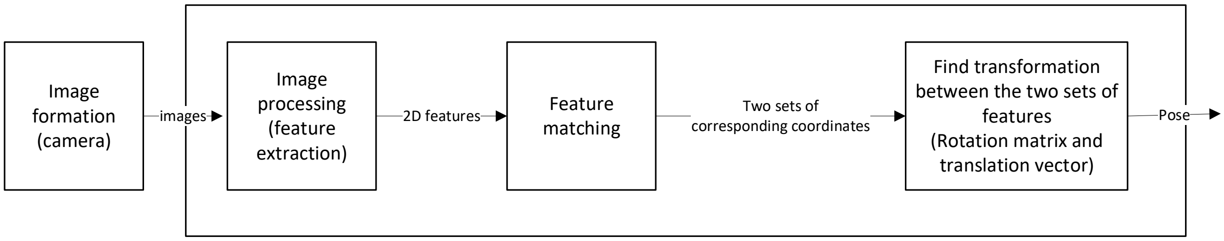
Image formation (camera)
images
Image processing (feature extraction)
2D features
Feature matching
Two sets of corresponding coordinates
Find transformation between the two sets of features (Rotation matrix and translation vector)
Pose

Fig. 1: Vision-based pose determination steps.

tion can interact and exchange information with its neighbor [9, 10].

In FF missions or in any mission where the orbiting target has been already conceived to be in proximity of a chaser, a passively cooperative target equipped with artificial markers (e.g., LEDs, CCRs) might be preferred to an actively cooperative target. Artificial markers and a monocular camera could be used in order to minimize power consumption, mass, and volume of the on-board navigation system or in deep-space missions where GNSS signals are not available. When dealing with uncooperative targets instead, such as space debris, asteroids, comets, or any other object originally not designed to cooperate with the chaser, EO sensors are currently the main technology for pose determination.

A LIDAR measures the distance with the target by illuminating it with a laser source and then processing the backscattered radiation. Such information is used to build a cloud of 3D points (i.e. to perform 3D reconstruction) of the target, allowing its pose estimation. Pose determination using LIDARs instead of stereo/monocular cameras is more robust to poor illumination conditions; however, LIDAR systems have higher hardware complexity and power consumption. Also, stereo cameras can provide 3D points of the target (2D coordinates in the image frame as well as their distance from the target through image rectification and disparity map computation), in some conditions, denser than the ones provided by a LIDAR. However, the ranging accuracy achievable with stereo cameras is significantly limited by the baseline between its two monocular cameras, being acceptable only in close-proximity with the target, e.g. final approach and docking. For far and mid range navigation monocular techniques can be used.

As illustrated in Fig. 1, following the process of image acquisition performed by the on-board camera system, the monocular-vision-based pose estimation process includes mainly three steps [11]:

- Image processing and feature extraction;
- Feature matching;
- Estimation of the pose, as transformation between matched features.

In the image processing step, one or more input images are computationally transformed into an enhanced image in order to extract some useful information from it. For vision-based pose determination, image processing is the foundation for the process of feature extraction. The latter is needed to reduce the amount of data in the images and extract the useful information, the ‘features’. Features can be scalars or vector quantities. As suggested in [11], it is possible to extract region-features, line-features and point-features. In the context of pose determination, correspondence or feature matching is the problem of matching the features extracted in one image of the target with corresponding features of another image, or identifiable points in a model when available. A set of corresponding features can then be used to determine the pose of the target with respect to the camera and the host space vehicle.

In this paper, we focus on monocular-vision-based pose determination as a suitable approach for nanosatellites with low mass, low volume, and low power consumption requirements, orbiting and maneuvering in a wide range of distances from the target. In particular, we investigate the case of unknown and known uncooperative targets.

1.1 *Related Work*

Different mono-vision-based approaches have been proposed in the literature which depend on whether the target geometry is known or unknown. If a target geometry is needed, a model can be built before the mission and stored or it has to be built on board. Many studies can be found on model-based spacecraft pose determination, which assume a target uncooperative but *known*. In Section 4 of [2], a complete review of the most relevant ones is provided. A fewer works exist in the literature, similar to the one here proposed, on vision- based pose determination of uncooperative *unknown* orbiting objects. In [12] an approach to estimate simultaneously kinematic state, geometry, and mass information of an unknown target was proposed , however it relies on a set of perfectly synchronized and cooperating 3D sensors uniformly distributed around the target. A more recent study, described in [13], proposes a feature-based SLAM approach focusing only on the filtering part and assuming already detected and tracked 3D features. Moreover, the presented architecture, based on the on-board acquisition of stereo-images, relies on a relative dynamics linear model (between the chaser and the target) which is only valid for circular orbits and, in particular, does not include the estimation of the unknown inertia matrix. An algorithm for real-time pose estimation based on monocular vision-only SLAM/SfM is presented in [14], where a Bayesian filter is adopted for the relative rotational dynamics with a simplified process model that assumes constant rotational velocity. The inertia matrix of the target is not estimated and the initial target pose is assumed to be known. In [15], the RANSAC algorithm is proposed in combination with an Extended Kalman Filter (EKF)-SLAM filter, for segmenting the resulting point cloud and reconstructing the target's structure. Anyhow, the computing time for the shape reconstruction of a few seconds does not allow the use of the algorithm in real time. The iSAM (incremental Smoothing and Mapping) method [16] is proposed in [17] to estimate pose and inertia matrix (up to a scale in absence of external torques) of a spinning orbiting target, but only for offline implementations. The use of an Iterated EKF algorithm is proposed in [18] to estimate the relative kinematic state and the ratios of the inertia matrix components. Finally in [19], an EKF-SLAM-based method for real-time relative state estimation of uncooperative unknown spinning targets using stereo vision is proposed. Although the use of the Kalman filter and its variations is popular among the SLAM and relative navigation field, there are some prior works looking into deriving a nonlinear observer for improved filter stability and robustness with nonlinear dynamics and measurement models [20, 21].

1.2 *Research Objective*

The goal of this study is to explore the main approaches for pose determination of uncooperative orbiting targets. Looking for a versatile and flexible solution which could be adopted in different mission scenarios (from OOS to FF, ADR, and exploration of asteroids), we first investigate a monocular pose estimation approach that does not require any knowledge of the target. For such a goal, the traditional EKF-based SLAM (Simultaneous Localization and Mapping) architecture and algorithms (see [22, 23, 24, 25]) were modified to perform Simultaneous Estimation of Pose and Shape, hereafter *"SEPS"*. This method, online, is suitable for real-time applications and valid for any type of target, also completely unknown. In addition, in the second part of the paper, we investigate an image-to-model approach that requires the knowledge of a simplified geometric model of the target. This could be built before the mission if the target's geometry is known already. Alternatively, if it is possible to safely orbit in proximity of the target and acquire a number of images, enough textured and representative of its complete geometry, the model could be reconstructed by processing those images directly on board. The OpenSfM implementation is considered for this goal. A preliminary implementation of both approaches validated with numerical simulations is described in this paper for a given chaser/target relative trajectory in GEO.

The paper is organized as follows. Following the introduction, Section 2 details the architecture, the image processing and the estimation steps derived to perform SEPS. Section 3 presents the image-to-model approach investigated as well as an on-board offline 3D reconstruction method that makes use of the Structure from Motion (SfM) implementation. Section 5 reports preliminary simulated performance of some of the described algorithms. Finally, in Section 6, the conclusion is drawn.

2 Simultaneous Estimation of Pose and Shape of an Unknown Target

The 3D reconstruction or mapping can be performed simultaneously with the pose estimation. This approach, typically adopted to locate a vehicle with respect to an unknown stationary world, while mapping it, is known as SLAM (Simultaneous Localization and Mapping). In case of pose determination of an orbiting target with respect to an orbiting chaser (hosting the camera), the problem becomes

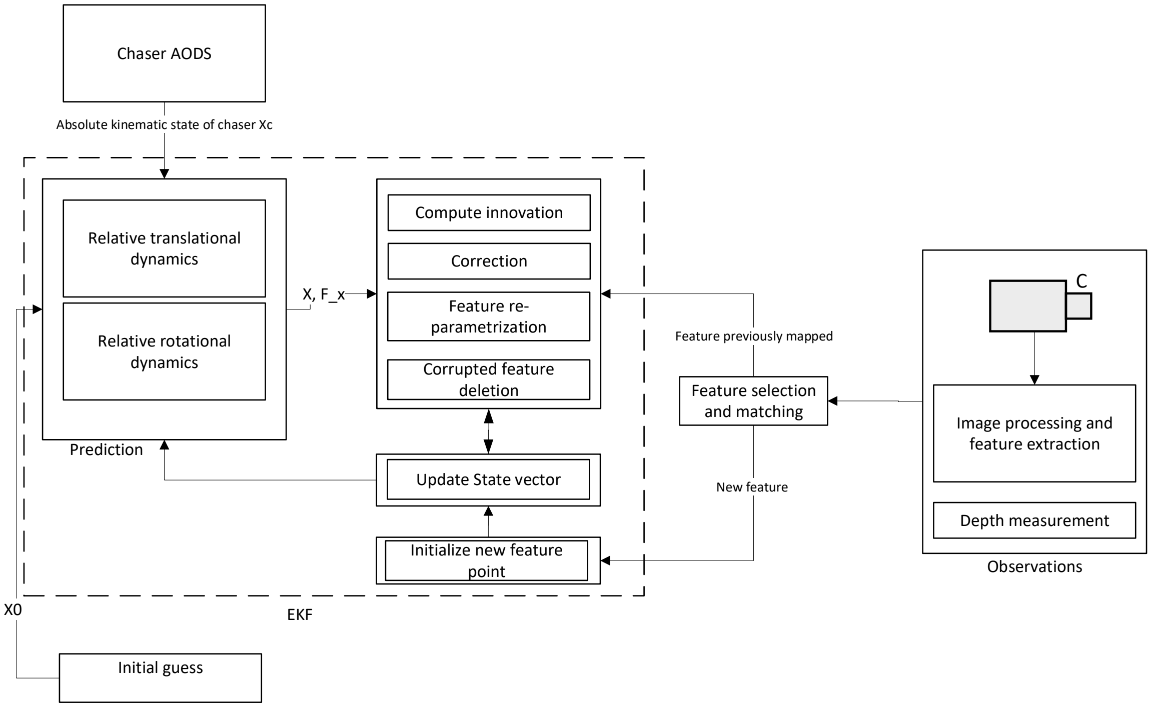
Chaser AODS
Absolute kinematic state of chaser Xc
Relative translational dynamics
Relative rotational dynamics
Prediction
X, F_x
Compute innovation
Correction
Feature re-parametrization
Corrupted feature deletion
Update State vector
Initialize new feature point
EKF
X0
Initial guess
Feature previously mapped
Feature selection and matching
New feature
C
Image processing and feature extraction
Depth measurement
Observations

Fig. 2: Monocular-vision-based pose determination of an unknown orbiting target.

more challenging since the target is not stationary with respect to the world frame. In the following sections, we detail the architecture and implementation of the SEPS filter for unknown targets.

2.1 *Architecture*

Figure 2 illustrates the architecture of the proposed SEPS filter. Different functional blocks can be identified.

Feature points are extracted from the images of the target, acquired on board the chaser by a monocular camera system. For only one of the feature points, also the depth measurement is provided for example by a sinlge-segment LIDAR. The same depth measurement is also used, with larger uncertainty, as depth measurement of the other extracted feature points. These are the observations of an EKF–based system, adopted as a sequential estimator, suitable for real time estimation, at relatively high rates. The relative translational and rotational dynamics models are used together with the absolute kinematic state of the chaser (provided by its Attitude and Orbital Determination System (AODS) to predict the observed feature points of the orbiting target. The observations of features already mapped and their predictions are fused to provide a better estimate of the relative pose between chaser and target, and at the same time, to refine the current 3D reconstruction. The observations of new features are initialized and added into the 3D model and to the state vector. The following sections characterize the image acquisition and processing and the nonlinear filtering implementation.

2.2 *Image Acquisition and Processing*

At each time step, images are acquired sequentially from an on-board monocular camera and converted to grayscale if colored. For spacecraft applications images typically have high contrast, high resolution, and low signal-to-noise ratio, which presents the separate challenge of processing them for meaningful data.

With monocular images, image processing subsystems may employ methods, such as image segmentation or feature detection, to break down an image into information that can be analyzed and manipulated. In feature detection methods, the most significant geometric features of the target object are extracted. The most reliable features of the target visible in space imagery are edges, corners, and blobs, which can be detected more reliably than points, over large distances between the chaser and the target, variable illumination conditions, and degradation of objects from prolonged space exposure [2, 26].

A well-known edge and corner detection algorithm is the Harris corner detector [27]. Several other feature detection algorithms are gradient-based techniques, such as the Hough transform [28] and Canny edge detector [29]. Applying these gradient-based methods directly to real images may not be successful since these methods are indiscriminate towards background and foreground, so edges from other objects or areas of high frequency variation also may be selected. Weak gradient elimination is introduced in [30], as a technique that distinguishes the target from the background so that a gradient-based method can select the real edges of the spacecraft.

Computer vision techniques using features with descriptors, such as SIFT [31], have also been introduced in spacecraft applications. Both [2] and [26] question SIFT's viability for on-board computation, noting heavy computational burden, ambiguity in feature detection and matching against symmetrical spacecraft, and conditions of high input image quality and low image noise for optimal performance. To that end, there have been other feature descriptors, such as SURF and ORB [32]), developed to lower computational burden.

For this paper, synthetic images illustrated in Fig. 3, were generated with the open source 3D suite Blender [33] using an existing model of the Aura spacecraft [34]. The camera was positioned at the simulated chaser positions while the model was rotated with the predetermined dynamics, according to the relative trajectories defined in Section 4.1. As the goal of this paper was not to propose new robust feature detection and tracking algorithms that are able to deal with actual space imagery (characterized by variable and unfavorable illumination conditions), a parallel light source was used to simulate solar illumination and the final images were rendered using ray-tracing.

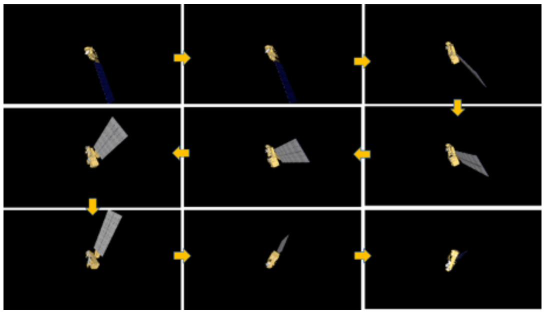

Fig. 3: Part of the synthetic images sequence generated with the free and open source 3D creation suite Blender [33] from the 3D model of the spacecraft [34].

In the block diagram shown in Fig. 2, observations lead to feature selection and matching. At the very first time step, there is only detection and initialization of unmapped features. Thereafter, features from an image can either be unmapped, or previously mapped and are used in the correction step of the EKF. Detection of previously unmapped features is implemented according to the following procedure adapted from [35].

The 960×540 sized image is partitioned into 100 subimages by defining a 10×10 cell grid, such that each subimage corresponds to a grid cell. A grid cell is randomly selected from the set of all unoccupied grid cells (unoccupied meaning that no feature has been detected from the subimage corresponding to that cell). The subimage corresponding to the selected grid cell is extracted from the original image. Using the Harris algorithm [27], the strongest Harris point is retrieved from the subimage. If the Harris score satisfies a threshold value, then the 2D coordinates of the point are stored as a measurement, and a 15×15 patch around the Harris point is stored as the feature's appearance. Feature matching is described in Section 2.3 under the correction step.

Figure 6 shows an example of both newly detected features and matched features after processing and extracting Harris points from one of the synthetic images. Extracted features are passed as inputs to the filter.

2.3 *Estimator Implementation*

In our implementation, the state vector includes relative states of the chaser with respect to the target, positions of the features of the target, and the target's principal moments of inertia up to a scale:

$$\boldsymbol{x} = \left[\boldsymbol{c}^\top, \boldsymbol{s}^\top, \boldsymbol{k}^\top\right]^\top . \quad (1)$$

Here, $\boldsymbol{c}$ is a vector of relative states of the chaser:

$$\boldsymbol{c} = \left[(\boldsymbol{r}^{\mathcal{T}}_{C/T})^\top,\ (\boldsymbol{v}^{\mathcal{T}}_{C/T})^\top,\ \boldsymbol{q}^\top\ (\boldsymbol{\omega}^{\mathcal{T}}_{C/T})^\top\right]^\top , \quad (2)$$

where $\boldsymbol{r}^{\mathcal{T}}_{C/T}$ and $\boldsymbol{v}^{\mathcal{T}}_{C/T}$ are the relative position and velocity vectors of the chaser with respect to the target expressed in the target frame, $\boldsymbol{q}$ is a quaternion representing the relative attitude of the chaser with respect to the target, and $\boldsymbol{\omega}^{\mathcal{T}}_{C/T}$ is the relative angular velocity of the chaser with respect to the target expressed in the target frame. We used the subscripts C and T to denote the chaser or target and the superscript $\mathcal{T}$ to represent the target frame. Similarly, we will use $\mathcal{C}$ and $\mathcal{I}$ to denote the chaser and the Earth-centered inertial (ECI) frames, respectively, hereafter. The dimension of $\boldsymbol{c}$ is 13.

Also, $\boldsymbol{s} = \left[\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_n\right]^\top$ is the target shape state with n reconstructed features (or "landmarks"), which includes the 3D coordinates of the reconstructed target feature points, expressed in target frame.

Finally, the vector $\boldsymbol{k} = \left[k_1, k_2\right]^\top$ includes two inertia ratios of the target, used to parametrize its inertia matrix, which also has to be estimated to propagate the relative rotational dynamics. As proved in [17], the inertia matrix is not fully observable if a torque free motion is assumed and only two of three degrees of freedom can be estimated. However, as done also in [18], the following two components of the inertia matrix, scaled with any scale factor, are sufficient to propagate the rotational motion. We set

$$k_1 = \ln\left(\frac{J_x}{J_y}\right) \quad \text{and} \quad k_2 = \ln\left(\frac{J_y}{J_z}\right). \tag{3}$$

By adopting this parameterization, the normalized target inertia matrix is

$$\boldsymbol{J}_T = \begin{bmatrix} e^{k_1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{k_2} \end{bmatrix}. \tag{4}$$

Therefore, when estimating the parametrized inertia matrix, $\boldsymbol{x}$ is a $15 + 3n$ element vector. The states are estimated by means of a nonlinear filter (e.g., EKF) through the prediction and correction processes, as well as the target's features initialization process, where new features of the target are detected and added to the current 3D reconstructed model. The discrete time prediction and update steps of an EKF can be expressed as follows [36]:

$$\hat{\boldsymbol{x}}_k^- = \hat{\boldsymbol{x}}_{k-1}^+ + \int_{k-1}^{k} \boldsymbol{f}(\boldsymbol{x}, t)dt, \tag{5}$$

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{\Phi}_{k-1}^\top + \boldsymbol{Q}_{k-1}, \tag{6}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^\top(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^\top + \boldsymbol{R}_k)^{-1}, \tag{7}$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)) = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\boldsymbol{z}_k^-, \tag{8}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-. \tag{9}$$

Where

$\hat{\boldsymbol{x}}_k^-$ is the a priori state estimate at a time step k,

$\hat{\boldsymbol{x}}_{k-1}^+$ is the a posteriori state estimate at time step $k-1$,

$\boldsymbol{\Phi}_{k-1}$ is the state transition matrix at a time step $k-1$,

$\boldsymbol{P}_k^-$ is the a priori estimate error covariance at a time step k,

$\boldsymbol{P}_{k-1}^+$ is a posteriori estimate error covariance at a time step $k-1$,

$\boldsymbol{Q}_{k-1}$ is the discrete process noise covariance a time step $k-1$,

$\boldsymbol{R}_k$ is the discrete measurement noise covariance at a time step k,

$\boldsymbol{H}_k$ is the measurement matrix at a time step k,

$\boldsymbol{K}_k$ is the Kalman gain at a time step k,

$\boldsymbol{y}_k$ is the measurement vector at a time step k,

$\boldsymbol{z}_k^-$ is the innovation measurement vector at a time step k,

$\boldsymbol{I}$ is a unit matrix.

2.3.1 *Prediction*

The prediction step is performed by integrating over time the relative dynamics between the chaser and target. In general, a continuous-time system is given as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}_s), \tag{10}$$

where $\boldsymbol{x}$ is a vector of system states, $\boldsymbol{f}$ is a dynamics model (nonlinear in general), $\mathbf{u}$ is a control signal, and $\boldsymbol{w}_s$ is the system noise vector, typically considered as Gaussian with a certain covariance. In our case, we assume $\boldsymbol{u} = 0$ because there is no control involved. The discrete time predicted state vector is given by Eq. (5).

Among several relative translational and rotational spacecraft dynamics models proposed in the literature, we use the ones proposed in [37, 38] for propagation of $\boldsymbol{c}$. However, both formulations are developed in a chaser-centric manner, whereas in a typical SLAM framework, the estimation is done with respect to the world frame where visual landmarks are located, which, in our case, corresponds to the target frame. For this reason, the dynamics in [38] are rederived from the target frame perspective. Also, we note that the dot notation in Eq. (11) implies time differentiation with respect to the target frame,

$$\dot{\boldsymbol{x}} = \left.\frac{d\boldsymbol{x}}{dt}\right|_{\mathcal{T}} = \frac{d}{dt}\begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{s} \\ \boldsymbol{k} \end{bmatrix}_{|\mathcal{T}}. \tag{11}$$

Note that $\dot{\boldsymbol{s}} = \dot{\boldsymbol{k}} = \boldsymbol{0}$. Therefore, we only need to know $\dot{\boldsymbol{c}}$ in order to propagate the relative dynamics.

The relative translational dynamics, when expressed in the chaser frame with respect to the chaser, is independent of the relative attitude dynamics [38]. However, this is not true when the relative translational dynamics is expressed in the target frame with respect to the target. This is because available information for the relative dynamics filter is absolute states of the chaser given from its external absolute navigation system and relative measurements of the target seen from the chaser. Therefore, in order to describe the relative dynamics in the target frame, the knowledge of the chaser states and measurements needs to be transformed into the target frame, which requires the knowledge of the relative attitude dynamics. For this reason, we describe the relative rotational dynamics first, and then present the relative translational dynamics.

Let $\boldsymbol{q} = [q_x, q_y, q_z, q_w]^\top$ be a quaternion with $[q_x, q_y, q_z]^\top$ and q_w being its vector and scalar components, respectively, whose equivalent rotation matrix transforms a vector in the chaser frame to the target frame, hence denoted as $\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}} = \boldsymbol{R}(\boldsymbol{q})$. Let $\boldsymbol{\omega}^{\mathcal{C}}_{C}$ and $\boldsymbol{\omega}^{\mathcal{T}}_{T}$ denote angular velocities of the chaser and target, respectively, as seen from the ECI frame and described in their respective frames. The relative angular velocity is defined as $\boldsymbol{\omega}^{\mathcal{T}}_{C/T} = \boldsymbol{\omega}^{\mathcal{T}}_{C} - \boldsymbol{\omega}^{\mathcal{T}}_{T} = \boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{\omega}^{\mathcal{C}}_{C} - \boldsymbol{\omega}^{\mathcal{T}}_{T}$. In our formulation, we assume $\boldsymbol{\omega}^{\mathcal{C}}_{C}$ is known from an external absolute navigation system of the chaser, but $\boldsymbol{\omega}^{\mathcal{T}}_{T}$ is not known because the target is unknown. The filter estimates $\boldsymbol{q}$ and $\boldsymbol{\omega}^{\mathcal{T}}_{C/T}$, and $\boldsymbol{\omega}^{\mathcal{T}}_{T}$ is estimated from these filtered states as

$$\boldsymbol{\omega}^{\mathcal{T}}_{T} = \boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{\omega}^{\mathcal{C}}_{C} - \boldsymbol{\omega}^{\mathcal{T}}_{C/T}. \tag{12}$$

The quaternion kinematics is given as follows:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{Q}(\boldsymbol{q})\boldsymbol{\omega}^{\mathcal{T}}_{T/C}, \tag{13}$$

where $\boldsymbol{\omega}^{\mathcal{T}}_{T/C} = -\boldsymbol{\omega}^{\mathcal{T}}_{C/T}$ and,

$$\boldsymbol{Q}(\boldsymbol{q}) = \begin{bmatrix} -q_w & -q_z & q_y \\ q_z & -q_w & -q_x \\ -q_y & q_x & -q_w \\ q_x & q_y & q_z \end{bmatrix}. \tag{14}$$

In order to obtain the relative attitude dynamics written in the target frame, consider the following angular acceleration $\boldsymbol{\alpha}^{\mathcal{T}}_{C/T}$:

$$\begin{aligned} \boldsymbol{\alpha}^{\mathcal{T}}_{C/T} &= \left(\frac{d\boldsymbol{\omega}_{C/T}}{dt}\bigg|_{\mathcal{T}} \right)^{\mathcal{T}} \\ &= \boldsymbol{R}^{\mathcal{T}}_{\mathcal{I}} \left(\frac{d\boldsymbol{\omega}_{C/T}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{I}} - \boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{\omega}^{\mathcal{T}}_{C/T}, \end{aligned} \tag{15}$$

where the subscripts after vertical bars represent in which reference frames differentiation is done. Notice that

$$\left(\frac{d\boldsymbol{\omega}_{C/T}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{I}} = \left(\frac{d\boldsymbol{\omega}_{C}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{I}} - \left(\frac{d\boldsymbol{\omega}_{T}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{I}}, \tag{16}$$

$$\begin{aligned} \boldsymbol{R}^{\mathcal{T}}_{\mathcal{I}} \left(\frac{d\boldsymbol{\omega}_{T}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{I}} &= \left(\frac{d\boldsymbol{\omega}_{T}}{dt}\bigg|_{\mathcal{T}} \right)^{\mathcal{T}} \\ &= \boldsymbol{J}_T^{-1}\left(\boldsymbol{m}_T - \boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{J}_T\boldsymbol{\omega}^{\mathcal{T}}_{T}\right), \end{aligned} \tag{17}$$

$$\begin{aligned} \boldsymbol{R}^{\mathcal{T}}_{\mathcal{I}} \left(\frac{d\boldsymbol{\omega}_{C}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{I}} &= \boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}} \left(\frac{d\boldsymbol{\omega}_{C}}{dt}\bigg|_{\mathcal{C}} \right)^{\mathcal{C}} \\ &= \boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\left[\boldsymbol{J}_C^{-1}\left(\boldsymbol{m}_C - \boldsymbol{\omega}^{\mathcal{C}}_{C} \times \boldsymbol{J}_C\boldsymbol{\omega}^{\mathcal{C}}_{C}\right)\right], \end{aligned} \tag{18}$$

where $\boldsymbol{J}_C$ and $\boldsymbol{J}_T$ are inertias of the chaser and target, and $\boldsymbol{m}_C$ and $\boldsymbol{m}_T$ are external moments applied to the chaser and target, respectively. By substituting Eqs. (16), (17), and (18) into Eq. (15), we obtain the following desired result:

$$\begin{aligned} \boldsymbol{\alpha}^{\mathcal{T}}_{C/T} &= \boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\left[\boldsymbol{J}_C^{-1}\left(\boldsymbol{m}_C - \boldsymbol{\omega}^{\mathcal{C}}_{C} \times \boldsymbol{J}_C\boldsymbol{\omega}^{\mathcal{C}}_{C}\right)\right] \\ &\quad - \boldsymbol{J}_T^{-1}\left(\boldsymbol{m}_T - \boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{J}_T\boldsymbol{\omega}^{\mathcal{T}}_{T}\right) - \boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{\omega}^{\mathcal{T}}_{C/T}, \end{aligned} \tag{19}$$

where Eq. (12) can be used to fully expand the expression.

Let us now consider the relative translational dynamics. Let $\boldsymbol{r}^{\mathcal{C}}_{C}$ and $\boldsymbol{r}^{\mathcal{T}}_{T}$ be positions of the chaser and target with respect to the origin of the ECI frame described in their respective frames. The relative position is defined as $\boldsymbol{r}^{\mathcal{T}}_{C/T} = \boldsymbol{r}^{\mathcal{T}}_{C} - \boldsymbol{r}^{\mathcal{T}}_{T} = \boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{r}^{\mathcal{C}}_{C} - \boldsymbol{r}^{\mathcal{T}}_{T}$. We assume $\boldsymbol{r}^{\mathcal{C}}_{C} = [r_C, 0, 0]^\top$ is given from an external absolute navigation system of the chaser with $r_C = \frac{a_C(1-e_C^2)}{1+e_C\cos\theta_C}$, where a_C is the semimajor axis, e_C is the eccentricity, and θ_C is the true anomaly of the chaser. However, $\boldsymbol{r}^{\mathcal{T}}_{T}$ is not known and it is estimated from the knowledge of $\boldsymbol{r}^{\mathcal{C}}_{C}$ and estimations of $\boldsymbol{r}^{\mathcal{T}}_{C/T}$ and $\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}$. The relative velocity $\boldsymbol{v}^{\mathcal{T}}_{C/T}$ in the target frame is defined as

$$\begin{aligned} \boldsymbol{v}^{\mathcal{T}}_{C/T} &= \left(\frac{d\boldsymbol{r}_{C/T}}{dt}\bigg|_{\mathcal{T}} \right)^{\mathcal{T}} = \left(\frac{d\boldsymbol{r}_{C/T}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{r}^{\mathcal{T}}_{C/T} \\ &= \boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{R}^{\mathcal{C}}_{\mathcal{I}} \left(\frac{d\boldsymbol{r}_{C/T}}{dt}\bigg|_{\mathcal{I}} \right)^{\mathcal{I}} - (\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{\omega}^{\mathcal{C}}_{C} - \boldsymbol{\omega}^{\mathcal{T}}_{C/T}) \times \boldsymbol{r}^{\mathcal{T}}_{C/T}, \end{aligned} \tag{20}$$

where we used Eq. (12) in the last equality. The relative acceleration $\boldsymbol{a}^{\mathcal{T}}_{C/T}$ in the target frame is computed as

$$\begin{aligned}\boldsymbol{a}^{\mathcal{T}}_{C/T} &= \left(\frac{d^2\boldsymbol{r}_{C/T}}{dt^2}\bigg|_{\mathcal{T}} \right)^{\mathcal{T}} \\ &= \left(\frac{d^2\boldsymbol{r}_{C/T}}{dt^2}\bigg|_{\mathcal{I}} \right)^{\mathcal{T}} - 2\boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{v}^{\mathcal{T}}_{C/T} \\ &+ \left(\frac{d\boldsymbol{\omega}_{T}}{dt}\bigg|_{\mathcal{T}} \right)^{\mathcal{T}} \times \boldsymbol{r}^{\mathcal{T}}_{C/T} + \boldsymbol{\omega}^{\mathcal{T}}_{T} \times (\boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{r}^{\mathcal{T}}_{C/T}).\end{aligned} \tag{21}$$

Note that

$$\left(\frac{d^2\boldsymbol{r}_{C/T}}{dt^2}\bigg|_{\mathcal{I}} \right)^{\mathcal{T}} = \left(\frac{d^2\boldsymbol{r}_{C}}{dt^2}\bigg|_{\mathcal{I}} \right)^{\mathcal{T}} - \left(\frac{d^2\boldsymbol{r}_{T}}{dt^2}\bigg|_{\mathcal{I}} \right)^{\mathcal{T}}, \tag{22}$$

and also that the absolute accelerations of the chaser and target with respect to the ECI frame are

$$\left(\frac{d^2\boldsymbol{r}_{C}}{dt^2}\bigg|_{\mathcal{I}} \right)^{\mathcal{T}} = -\frac{\mu}{r_C^3}\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{r}^{\mathcal{C}}_{C}, \tag{23}$$

$$\left(\frac{d^2\boldsymbol{r}_{T}}{dt^2}\bigg|_{\mathcal{I}} \right)^{\mathcal{T}} = -\frac{\mu}{\|\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{r}^{\mathcal{C}}_{C} - \boldsymbol{r}^{\mathcal{T}}_{C/T}\|^3}(\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{r}^{\mathcal{C}}_{C} - \boldsymbol{r}^{\mathcal{T}}_{C/T}). \tag{24}$$

By substituting Eqs. (22), (23), and (24) into Eq. (21) and using Eq. (17), we obtain the desired relative translational dynamics expressed in the target frame as

$$\begin{aligned}\boldsymbol{a}^{\mathcal{T}}_{C/T} &= -\frac{\mu}{r_C^3}\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{r}^{\mathcal{C}}_{C} \\ &+ \frac{\mu}{\|\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{r}^{\mathcal{C}}_{C} - \boldsymbol{r}^{\mathcal{T}}_{C/T}\|^3}(\boldsymbol{R}^{\mathcal{T}}_{\mathcal{C}}\boldsymbol{r}^{\mathcal{C}}_{C} - \boldsymbol{r}^{\mathcal{T}}_{C/T}) \\ &+ \boldsymbol{J}_T^{-1}\left(\boldsymbol{m}_T - \boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{J}_T\boldsymbol{\omega}^{\mathcal{T}}_{T}\right) \times \boldsymbol{r}^{\mathcal{T}}_{C/T} \\ &- 2\boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{v}^{\mathcal{T}}_{C/T} + \boldsymbol{\omega}^{\mathcal{T}}_{T} \times \left(\boldsymbol{\omega}^{\mathcal{T}}_{T} \times \boldsymbol{r}^{\mathcal{T}}_{C/T}\right),\end{aligned} \tag{25}$$

where Eq. (12) can be used to fully expand the expression. In this work, both $\boldsymbol{m}_C$ and $\boldsymbol{m}_T$ are zero.

To summarize, $\boldsymbol{c}$ propagates as follows:

$$\dot{\boldsymbol{c}} = \frac{d\boldsymbol{c}}{dt}\bigg|_{\mathcal{T}} = \frac{d}{dt}\begin{bmatrix}\boldsymbol{r}^{\mathcal{T}}_{C/T} \\ \boldsymbol{v}^{\mathcal{T}}_{C/T} \\ \boldsymbol{q} \\ \boldsymbol{\omega}^{\mathcal{T}}_{C/T}\end{bmatrix}_{|\mathcal{T}} = \begin{bmatrix}\boldsymbol{v}^{\mathcal{T}}_{C/T} \\ \boldsymbol{a}^{\mathcal{T}}_{C/T} \\ \frac{1}{2}\boldsymbol{Q}(\boldsymbol{q})\boldsymbol{\omega}^{\mathcal{T}}_{T/C} \\ \boldsymbol{\alpha}^{\mathcal{T}}_{C/T}\end{bmatrix}. \tag{26}$$

2.3.2 *Correction*

The correction step includes the following substeps. The implementation of the correction is modified from the methods proposed by Joan Solà in [35].

i. Selection of feature observations to correct

ii. Feature matching

iii. Nonlinear filtering execution

iv. Correction of feature parameters out of the filter

v. Feature re-parameterization and feature deletion in case of corruption

Step i. In the filter structure, the feature observations are processed in the EKF filter one by one. In order to select the 3D feature points to correct, the ones estimated previously have to be projected onto the 2D camera measurement space according to the predicted pose.

First, each feature point $\boldsymbol{p}^{\mathcal{T}}$ in the target frame is expressed in chaser frame, by means of the predicted pose from the previous pose estimate $(\boldsymbol{R}^{\mathcal{C}}_{\mathcal{T}}, \boldsymbol{r}^{\mathcal{T}}_{T/C})$.

$$\boldsymbol{p}^{\mathcal{C}} = \boldsymbol{R}^{\mathcal{C}}_{\mathcal{T}}(\boldsymbol{p}^{\mathcal{T}} - \boldsymbol{r}^{\mathcal{T}}_{C/T}) \tag{27}$$

Then, it is projected onto the image plane.

A perspective monocular camera associates points in 3D space, $\boldsymbol{p}^{\mathcal{C}} = \begin{bmatrix}x_c & y_c & z_c\end{bmatrix}^\top$, with points in the 2D image plane, $\boldsymbol{P} = \begin{bmatrix}X & Y\end{bmatrix}^\top$, by means of the projection process. As illustrated in Fig. 4, a simple pin-hole camera model is adopted here, characterized by an optical center, O, and optical axis, and an image plane, perpendicular to the optical axis, situated at a distance f from the optical center, named focal length. The intersection of the optical axis in the image plane is the principal point. We assume the optical axis to be aligned with the local z axis, the origin in the principal point and the other two axis.

The projection of the object point $\boldsymbol{p}^{\mathcal{C}}$ in the 3D space corresponds to the intersection of the line $\overline{O\boldsymbol{p}^{\mathcal{C}}}$ with the image plane. The image point $\boldsymbol{P}$ is obtained as follows:

$$\frac{X}{f} = \frac{x_c}{z_c}, \qquad \frac{Y}{f} = \frac{y_c}{z_c} \tag{28}$$

Therefore, the transformation from 3D space to the 2D local coordinate frame of the camera is

$$\boldsymbol{P} = \begin{bmatrix}X \\ Y\end{bmatrix} = \begin{bmatrix}x_c \\ y_c\end{bmatrix} \cdot \frac{f}{z_c}, \tag{29}$$

where z_c is the depth of the object point.

The point $\boldsymbol{P}$ can be expressed in pixel units [pix] instead of metric units [m], as follows:

$$u = u_0 + s_u X, \quad v = v_0 + s_v Y \tag{30}$$

where $\begin{bmatrix}s_u & s_v\end{bmatrix}^\top$ are the horizontal and vertical pixel density in [pix/m] and $\begin{bmatrix}u_0 & v_0\end{bmatrix}^\top$ the pixel coordinates of the principal point in [m]. Among all the

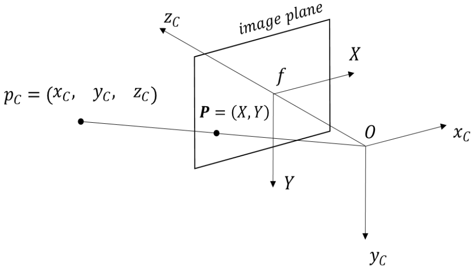

Fig. 4: Pin-hole camera model.

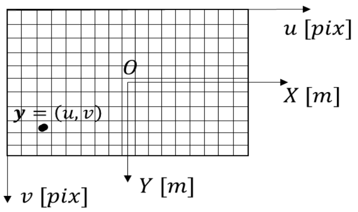

Fig. 5: Projected point into the image plane in metric and pixel units.

projected points, only the ones visible (i.e. in the field of view of the camera) are considered.

The continuous time measurement model can be written as

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{w}_m \tag{31}$$

where $\boldsymbol{y}$ is the noisy measurement, $\boldsymbol{x}$ is the state vector, $\boldsymbol{h}(\cdot)$ is a non-linear function, and $\boldsymbol{w}_m$ is the sensor's noise, usually considered white Gaussian. The sensor camera measurements can be expressed in the chaser frame with a known rigid transformation $(\boldsymbol{R}^{\mathcal{C}}_{\mathcal{S}}, \boldsymbol{r}^{\mathcal{C}}_{S/C})$, where $\mathcal{S}$ is used to denote the sensor camera frame.

An observation $\boldsymbol{y}_i$ is the measure of the i-th target feature in the camera frame. When actual images are used, as described in Section 2.2, the features are extracted from images of the target acquired by the camera system and then are processed. Two components of $\boldsymbol{y}_i$ are the geometric parametrization of the i-th feature in the camera measurement space, in pixel (u_i, v_i). In our case, the monocular camera system is a bearing only sensor and does not provide the depth of the extracted feature point. In this study for simplicity, we assume that the depth d is measured by another sensor, e.g. a single-segment LIDAR sensor. Assuming that the LIDAR frame coincides with the camera frame, then $\boldsymbol{y}_i$ also includes the depth of the i-th feature as third component.

Once the positions of all previously mapped feature points are predicted in the camera frame, they are sorted based on the expectation covariance matrix (i.e. the uncertainty of the measurements), and only the first N are selected.

Step ii. The selected 2D points in the image plane (which are the selected predicted observed features) are matched to the observed ones.

For feature detection and matching, a Harris-based template matching approach is used. The predicted appearance of a visible landmark (or feature) from time $k-1$ is given as a 15×15 pixel patch around the original 2D coordinates of that landmark when first detected and initialized. This patch is slid across a grey-scale image acquired by the monocular camera at time step k, and a zero-normalized cross correlation (ZNCC) score [39] is computed between the two 15×15 patches to quantify the similarity in the subimages.

The patch in the new image yielding the highest ZNCC score is considered the most optimistic potential match. If the ZNCC score exceeds a threshold value, then a landmark at time $k-1$ is successfully matched to a feature in the image at time k, and the pixel center coordinates of the best patch are stored.

Note that at each time step, a new image is processed and extracted feature points are matched to the predicted observed features of previously mapped points. Points that are not matched are initialized as new features. Figure 6 illustrates a set of points that can be extracted from an image in one time step, either by template matching or Harris point extraction.

Step iii. In this step, the innovation is computed, on the basis of the difference between the new measurement and the prediction.

At time k, the discrete-time innovation's $\boldsymbol{z}^-_k$ and innovation's covariance matrix $\boldsymbol{Z}^-_k$ are the following:

$$\boldsymbol{z}^-_k = \boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}^-_k) \tag{32}$$

$$\boldsymbol{Z}^-_k = \boldsymbol{H}_k \boldsymbol{P}^-_k \boldsymbol{H}^\top_k + \boldsymbol{R}_k \tag{33}$$

where the observation matrix (at time step k) $\boldsymbol{H}_k$ is the Jacobian of the observation functions, given by the equation

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}^-_k} \tag{34}$$

Then, the Kalman gain can be calculated according to Eq. (7), while the state vector and the covariance matrix can be updated, respectively according to Eqs. (8) and (9).

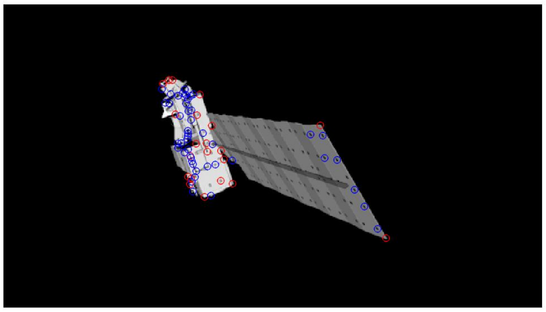

Fig. 6: Multiple Harris points extracted from an image (generated with the free and open source 3D creation suite Blender [33]) of the 3D model of the Aura spacecraft [34]. Features matched to previously mapped features are in red, and features previously unmapped and newly initialized are in blue.

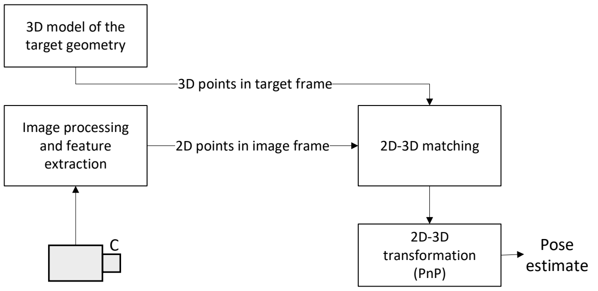
3D model of the target geometry
3D points in target frame
Image processing and feature extraction
2D points in image frame
2D-3D matching
C
2D-3D transformation (PnP)
Pose estimate

Fig. 7: Monocular model-based pose determination of known orbiting target.

Step iv. The EKF correction is performed, according to Eqs. (5) and (6).

Step v. The feature points are corrected and reparametrized. Then, they are deleted if either the ratio between the matches or the inliers and the searches is smaller than a threshold and if the number of searches is higher than a threshold.

2.3.3 *Initialization of New Features*

Any extracted feature not yet part of the state vector (of the current 3D reconstructed model) has to be initialized and added in the state vector. This operation results in an increase of the state vector's size from $15 + 3n$ to $15 + 3(n + 1)$. For this goal, the new feature points are re-projected onto the 3D camera space, according to the last pose estimate, their 2D coordinates in the image plane and their depth measurement or estimate. This is done by inverting the observation function $\boldsymbol{h}(\cdot)$ to compute $\boldsymbol{p}^{\mathcal{C}}$. The point $\boldsymbol{p}^{\mathcal{C}}$ in camera frame is transformed to the corresponding point in target frame $\boldsymbol{\zeta}$ and then added to the current 3D model $\boldsymbol{s}$, part of the state vector.

3 Pose Estimation of a Known Target

If the target geometry is known, a detailed or simplified model of it (e.g. see Fig. 8) can be built offline and stored on board and model-based algorithms can be used for pose estimation. In this case, as illustrated in Fig. 7, adopting an "image-to-model" approach and feature extraction and matching algorithms, the pose can be estimated by matching the geometrical natural features extracted from the acquired images with the corresponding ones of the model. Also in this case, several types of features can be extracted, such as corners, curves, lines, depending on the characteristics of the geometry and of the acquired images. In pose acquisition (or initialization) no information about the prior target pose is available, while in pose tracking the pose is updated based on the previous pose estimate and the new acquired image. When using monocular camera systems, feature-based algorithms can make use of PnP (Perspective-n-Point) solvers.

Given a set of n 3D homogeneous points, $\boldsymbol{\pi}_i = \begin{bmatrix} x_i, & y_i, & z_i, & 1 \end{bmatrix}^\top$ with $i = 1, 2, \ldots, n$ in the target frame, their corresponding projected n 2D points $\boldsymbol{\Pi}_i = \begin{bmatrix} u_i, & v_i, & 1 \end{bmatrix}^\top$ in the camera image frame, and the intrinsic camera parameters matrix $\boldsymbol{K}$, solving the PnP problem corresponds to computing the rotation matrix from target frame to camera frame, $\boldsymbol{R}_{\mathcal{T}}^{\mathcal{C}}$ and the translation vector from the origin of the target frame to the origin of the camera frame $\boldsymbol{r}_{T/C}^{\mathcal{C}}$, as follows:

$$s_i \boldsymbol{\Pi}_i = \boldsymbol{K} \left[\boldsymbol{R}_{\mathcal{T}}^{\mathcal{C}} \mid \boldsymbol{r}_{T/C}^{\mathcal{C}} \right] \boldsymbol{\pi}_i, \tag{35}$$

where s_i is a scale factor for the image point. The camera matrix is

$$\boldsymbol{K} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{36}$$

where f_x and f_y are the scaled focal lengths, γ is the skew parameter, and u_0 and v_0 are the coordinates of the principal point.

A comparative assessment of the most commonly used PnP solvers is provided in [26] and [40]. In this study, we adopt the PosIt PnP solver [41] for both pose acquisition and pose tracking processes, as it has lower computational burden than the accurate Newton-Raphson Method, but it can deal better than the EPnP method with long distances along the optical axis and pixel location noise [26]. PosIt can estimate the pose between a set of at least four-non-coplanar 3D points of the model and their corresponding 2D points in the image. It uses a Scaled Orthographic Projection (SOP) to approximate the true perspective projection for a first coarse pose estimation, then iteratively refined until convergence. The SOP approximation linearizes Eq. (35) and provides a coarse $\boldsymbol{R}^{\mathcal{C}}_{\mathcal{T}}$ and $\boldsymbol{r}^{\mathcal{C}}_{T/C}$ without the need of a starting pose.

For the i-th correspondence, a scaled value s_i is introduced and updated at the end of each iteration. Initially the scaled value is set to 1 ($s_i = 1$). Scaling each term of Eq. (35) by $\frac{1}{\boldsymbol{r}^{\mathcal{C}}_{T/C}(3)}$ and initializing with the SOP assumption $\frac{s_i}{\boldsymbol{r}^{\mathcal{C}}_{T/C}(3)} = 1$, the first two rows become a linear system of equations with eight unknowns, which can be solved with $n = 4$. Then, $\frac{s_i}{\boldsymbol{r}^{\mathcal{C}}_{T/C}(3)}$ can be computed from the estimated $\boldsymbol{R}^{\mathcal{C}}_{\mathcal{T}}$ and $\boldsymbol{r}^{\mathcal{C}}_{T/C}$. The computation of $\boldsymbol{R}^{\mathcal{C}}_{\mathcal{T}}$ and $\boldsymbol{r}^{\mathcal{C}}_{T/C}$, given the updated s_i, can be iterated until s_i is smaller than a threshold or until a maximum number of iterations has been reached.

Instead, to solve the correspondence problem (the matching between the set of 3D points in the model and the 2D points in the image frame), as proposed in [42], we used the RANSAC algorithm [43] for pose acquisition, and the Soft-Assign strategy [44] for pose tracking (adopting the SoftPosit implementation of [45]).

3.1 *3D Model of Known Geometry*

Fig. 8 illustrates a simplified geometrical model of the target we used for model-based pose determination.

3.1.1 *Model Built before the Mission*

The model in Fig. 8 can be built before the mission (if possible) and stored on board the chaser. Clearly, the more complex and rich of details is the model, the higher the number of points that can be matched (and potentially the achievable accuracy in pose determination). However, the larger is the set of 3D points, the higher is the computational burden of the 2D to 3D matching process.

In particular, the model illustrated in Fig. 8, was constructed by manually selecting representative points of the AURA spacecraft 3D model [34], illustrated in Fig. 9. A more effective approach might be building the model by using the same feature detector that will be used on board, in order to minimize and maximize respectively the number of outliers and of inliers.

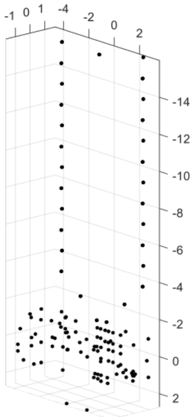

Fig. 8: Simplified geometrical model of the target spacecraft illustrated in Fig. 9.

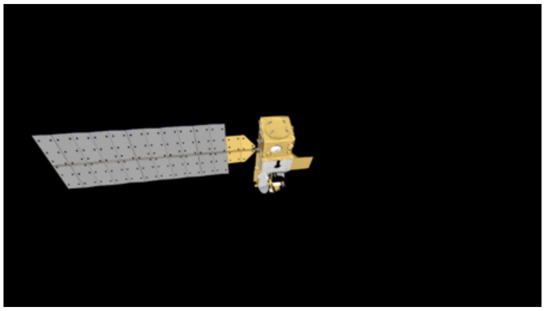

Fig. 9: A synthetic image of the Aura spacecraft, input of the simulations carried out in Sec. 4, generated with Blender [33], from the 3D model of the spacecraft [34].

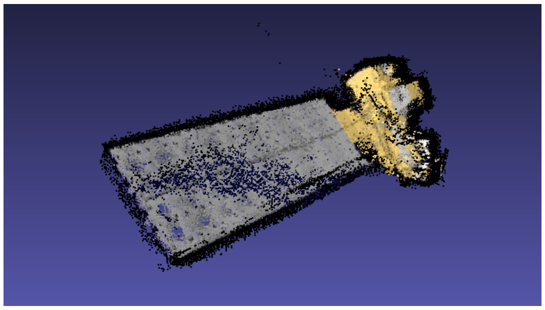

Fig. 10: Reconstructed point cloud with OpenSfM using 150 images.

3.1.2 *Model Built On Board*

With enough time and processing power, it is possible to reconstruct the 3D model of the object on board. Rather than doing frame-to-frame object tracking, the camera can capture a set of discrete images and use offline processing. These algorithms are usually exhaustive in searches for image correspondences to minimize chances of failure and increase accuracy. Time consuming nonlinear optimizations are used in iterative reconstructions. To get a dense 3D model, costly matching algorithms are run on the resulting relative camera positions.

A typical implementation of SfM starts by processing and matching each pair of images to find keypoint correspondences, which are triangulated into maps of 3D points. If the target is rotating relative to the chaser, it will generate images with large relative view angles that allow for accurate point triangulation. Once these correspondences are established a bundle adjustment algorithm is run, taking in the camera poses and 3D point locations as parameters to minimize the reprojected point errors. The bundle adjustment is run iteratively, adding images incrementally to the full reconstruction. After the camera positions are recovered a template matching algorithm such as ZNCC or NCC can then be run on multiple views simultaneously to increase accuracy and point density.

Note that the keypoint identification algorithm used for live tracking may also be used with SfM. We can then use the resulting map of tracked features as a 3D model without needing to do a full dense reconstruction.

Figure 10 shows the resulting point cloud from running OpenSfM on the 150 images of the simulated data. It took approximately 2 hours on an i7-3770K CPU configured to run off of 8 threads using the HAHOG feature and descriptor combination. There are some dense outliers, but overall the reconstruction is accurate up to scale.

4 Simulated Performance

The approaches described in Sections 2 and 3 were partially validated through numerical simulations.

4.1 *Simulation Models and Assumptions*

For both approaches we assumed the chaser and target are orbiting in GEO (on the same orbit of IntelSat [46]) with an initial separation of 15 m along the boresight of the camera and an initial relative rotation of 1 deg/s. The assumed camera intrinsic parameters are the following. $f \cdot m_x = f \cdot m_y = 300$ where f is the focal length and m_x and m_y are the scale factors relating image pixels to distance. The image center is $(480, 270)$ and the image size is $(960, 540)$.

The SEPS approach was tested, following a coarse tuning of the covariance matrices. As already highlighted in the previous sections, the image processing strategy was validated using a sequence of images synthetically generated according to the predetermined relative trajectory.

4.2 *Pose Estimation and 3D Reconstruction of an Unknown Target*

The following figures illustrate the target pose (position and attitude) estimation error when adopting the SEPS approach described in Section 2, with the assumptions of Section 4.1. In particular, Fig. 11 shows the norm of the relative 3D position estimation error. Figure 12 displays the relative attitude estimation error, as the angle describing the magnitude of the rotation error in an axis angle representation. Figures 13 and 14 show the norm of the relative 3D velocity estimation error and the norm of the relative 3D angular velocity estimation error. All the filtered state components converge. However, while the attitude, velocity, and angular velocity estimations converge to the true relative state, with an error smaller than 1 deg, 0.3 m/s and 0.005 deg/s respectively, the position estimation after convergence is affected by a bias of approximately 2.5 m, due to the depth estimation error.

4.3 *Pose Estimation of a Known Target*

Figures 15 and 16 display the pose estimation error (respectively the position and the attitude estimation error) when adopting the model-based approach described in Section 3, under the assumptions described in Section 4.1. The knowledge of the target geometry is an advantage in the process of pose estimation, enabling the use of model-based algorithms, without the need to also simultaneously perform 3D reconstruction. Thanks to the knowledge of the geometry, both position and attitude are estimated with higher accuracy than with the SEPS strategy.

5 Conclusion

In this paper, we investigated two main approaches for on-board pose estimation of uncooperative orbiting targets: one general method suitable also for unknown targets, which performs Simultaneous 3D Reconstruction and Pose Estimation (*SEPS*), and a second one that can be adopted only when a model of the target's geometry is available on board.

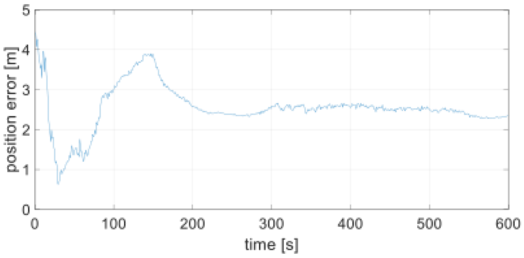

Fig. 11: Norm of the relative 3D position estimation error, with SEPS approach.

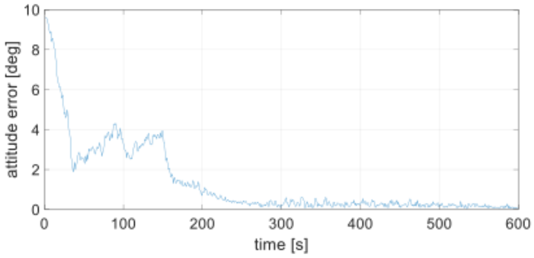

Fig. 12: Relative attitude estimation, with SEPS approach.

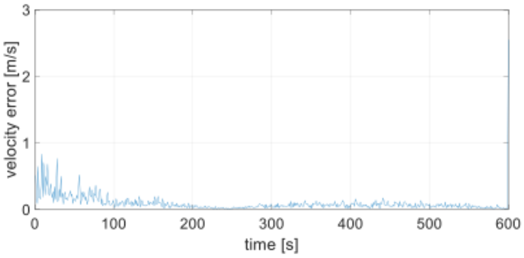

Fig. 13: Norm of the relative 3D velocity estimation error, with SEPS approach.

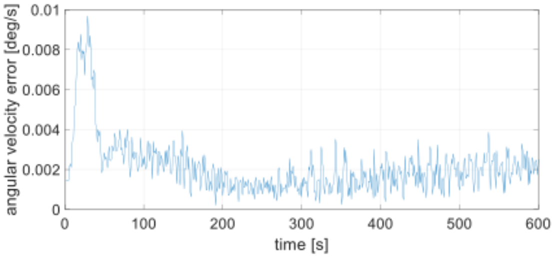

Fig. 14: Norm of the relative 3D angular velocity estimation error, with SEPS approach.

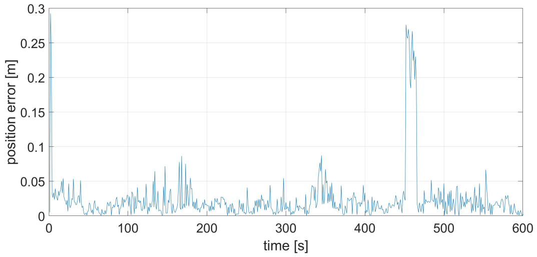

Fig. 15: Norm of the relative 3D position estimation error, with model-based approach.

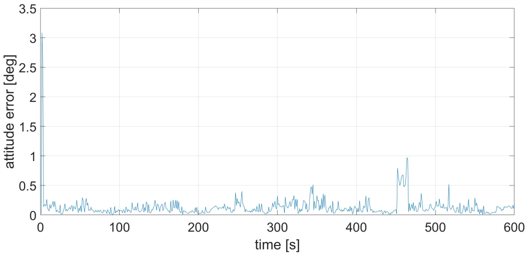

Fig. 16: Relative attitude estimation error, with model-based approach.

For the first approach, a detailed description of the architecture and implementation was provided, as the proposed strategy aims at solving a more complex and challenging problem still at a research stage and (to the best of the authors' knowledge) not yet experimentally demonstrated. Indeed, the pose estimation of an unknown (and uncooperative) target, also requires the simultaneous estimation of its inertia matrix. Moreover, the level of complexity of the architecture increases if the estimation process relies on monocular instead of stereo images.

Then we described the second approach which in our architecture adopts the well-known PosIt PnP solver and the RANSAC and Soft Assign algorithms for the 2D-3D feature matching. The possibility of building a geometrical model of the target directly on board from a batch of acquired images, rather than before the mission, was also discussed. The described approaches and algorithms were all tested by means of numerical simulations, using the same dataset of images, synthetically generated according to a realistic chaser/target relative trajectory in GEO.

For future work, we plan to further investigate the problem of feature extraction and tracking from actual space images, characterized by low carrier-to-noise ratios and high image contrast. Also, we will fully validate the proposed SEPS method using a spacecraft simulator testbed at Caltech.

Acknowledgement

The first author was supported by the Swiss National Science Foundation (SNSF). This work was also supported in part by the Jet Propulsion Laboratory (JPL). Government sponsorship is acknowledged. The authors thank F. Y. Hadaegh, A. Stoica, M. Wolf, S. R. Alimo, and M. Quadrelli.

References

[1] D. Wright, "The Current Space Debris Situation," in *2010 Beijing Orbital Debris Mitigation Workshop*, 2010.

[2] R. Opromolla, G. Fasano, G. Rufino and M. Grassi, "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations," *Progress in Aerospace*, vol. 93, pp. 53-72, 2017.

[3] P. Buist, P. Teunissen and P. Joosten, "GNSS-Guided Relative Positioning and Attitude Determination for Missions with Multiple Spacecraft," in *GPS/GNSS Symposium*, Tokyo, 2013.

[4] S. D'Amico and O. Montenbruck, "Differential GPS: An enabling technology for formation flying satellites," in *7th IAA Symp. Small Sat. Earth Observ.*, Berlin, Germany, 2009.

[5] R. Kroes, "Precise Relative Positioning of Formation Flying Spacecraft using GPS," NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission, Delft, The Netherlands, 2006.

[6] M. Psiaki and S. Mohiuddin, "Relative Navigation of High-Altitude Spacecraft Using Dual-Frequency Civilian CDGPS," in *ION GNSS*, Long Beach, California, 2005.

[7] J. Mitchell, B. Barbee and P. &. L. R. Baldwin, "Expanding Hardware-in-the-loop Formation Navigation and Control with Radio Frequency Crosslink Ranging," 2007.

[8] B. Sheard, K. Heinzel, K. S. D. A. Danzmann, K. W. M. and W. Folkner, "Intersatellite laser ranging instrument for the GRACE follow-on mission," *J. Geodesy*, vol. 86, no. 12, pp. 1083-1095, 2012.

[9] S. Bandyopadhyay, R. Foust, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Review of Formation Flying and Constellation Missions Using Nanosatellites," *J. Spacecraft Rockets*, vol. 53, no. 3, pp. 567-578, 2016.

[10] F. Y. Hadaegh, S.-J. Chung, and H. M. Manohara, "On Development of 100-Gram-Class Spacecraft for Swarm Applications," *IEEE Systems J.*, vol. 10, no. 2, pp. 673-684, June 2016.

[11] P. Corke, "Robotics, Vision and Control," Springer, 2017.

[12] M. Litchter and S. Dubowsky, "State, shape, and parameter estimation of space objects from range images," in *IEEE lnt. Conf. Robotics & Autom.* , New Orleans, LA, 2004.

[13] A. Sonnenburg, M. Tkocz and K. Janschek, "EKF-SLAM based Approach for Spacecraft Rendezvous Navigation with Unknown Target Spacecraft," in *18th IFAC Symp. Autom. Control in Aerospace*, Nara, Japan, 2010.

[14] S. Augenstein and S. M. Rock, "Improved Frame-to-Frame Pose Tracking during Vision-Only SLAM/SFM with a Tumbling Target," in *2011 IEEE Int. Conf. Robotics Autom.*, Shanghai, China, 2011.

[15] F. Schnitzer, K. Janschek and G. Willich, "Experimental Results for Image-based Geometrical Reconstruction for Spacecraft Rendezvous Navigation with Unknown and Uncooperative Target Spacecraft," in *2012 IEEE/RSJ Int. Conf. Intell Robots Sys.*, Vilamoura, Algarve, Portugal, 2012.

[16] M. Kaess, A. Ranganathan and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365-1378, 2008.

[17] B. E. Tweddle, "Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations," MIT, 2013.

[18] V. Pesce, M. Lavagna and R. Bevilacqua, "Stereovision-based pose and inertia estimation of unknown and uncooperative space objects," *Advances Space Res.*, vol. 59, no. 1, pp. 236-251, 2017.

[19] Y. Li, Y. Xie, "Relative State Estimation of Model-Unknown Spinning Noncooperative Target Using Stereo EKF-SLAM," Proceedings of the 36th Chinese Control Conference, 2017.

[20] A. P. Dani, S.-J. Chung, and S. Hutchinson, "Observer Design for Stochastic Nonlinear Systems via Contraction-based Incremental Stability," *IEEE Trans. Automatic Control*, vol. 60, no. 3, pp. 700-714, 2015.

[21] A. Dani, G.Panahandeh, S.-J. Chung, and S. Hutchinson, "Image Moments for Higher-Level Feature Based Navigation," *IEEE/RSJ Int. Conf. Intell. Robots Sys (IROS)*, Tokyo, Japan, pp. 602-609, 2013.

[22] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robotics Autom.*, vol. 17, no. 3, pp. 229-241, 2001.

[23] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Autom. Mag.*, vol. 13, no. 2, pp. 99-110, 2006.

[24] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *IEEE Int. Conf. on Computer Vision*, 2003.

[25] J. Civera and A. M. J. Davison, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 932-945, 2008.

[26] S. Sharma and S. D'Amico, "Comparative Assessment of Techniques for Initial Pose Estimation using Monocular Vision," *Acta Astronautica*, vol. 123, pp. 435-445, 2016.

[27] C. Harris, and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Alvey Vision Conf.*, 1988, pp. 147–151.

[28] R.O. Duda, P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," Commun. ACM 15 (1) (1972) 11–15.

[29] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-8, No. 6, 1986, pp. 679–698.

[30] S. Sharma and J. Ventura, "Robust Model-Based Monocular Pose Estimation for Noncooperative Spacecraft Rendezvous," *J. Spacecraft Rockets*, 2017, pp. 1–35.

[31] D.G. Lowe, "Distinctive image features from scale-invariant key- points," Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[32] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in: *Proc. IEEE Int. Conf. Comp. Vision*, Nov. 2011.

[33] "Blender," [Online]. Available: https://www.blender.org.

[34] C. M. Garcia, "NASA 3D resources," 2018. [Online]. Available: https://nasa3d.arc.nasa.gov/detail/aura-eoe3d. [Accessed 24 July 2018].

[35] J. Sola, D. Marquez, J. Codol, and T. Vidal-Calleja. An EKF-SLAM toolbox for MATLAB. Available online: https://github.com/joansola/slamtb.

[36] P. D. Groves, "Principles of GNSS, Inertial, and Multisensor Integrated Navigation systems," Artech House, 2013.

[37] P. Gurfil and K. Kholshevnikiv, "Manifolds and metrics in the relative spacecraft motion problem," *J. Guid. Control Dyn.*, vol. 29, 2006.

[38] S. Segal and P. Grufil, "Effect of Kinematic Rotation-Translation Coupling on Relative Spacecraft Translational Dynamics," *J. Guid. Control Dyn.*, vol. 32, pp. 1045-1050, 2009.

[39] W. Krattenthaler, K.J. Mayer, M. Zeiler, 1994. Point correlation: A reduced-cost template matching technique. In: *1st IEEE Int. Conf. on Image Processing (ICIP)*, vol. I, September, 1994, Austin, Texas, USA, pp. 208-212, 1994.

[40] D. Grest, T. Petersen and V. Kuger, "A Comparison of Iterative 2D-3D Pose Estimation Methods for Real-Time Applications," in *Scandinavian Conf. Image Anal.*, 2009.

[41] D. F. Dementhon and L. S. Davis, "Model-Based Object Pose in 25 Lines of Code," *Int. J. Computer Vision*, vol. 15, pp. 123-141, 1995.

[42] V. Capuano, G. Cuciniello, V. Pesce, R. Opromolla, et al., "VINAG: A Highly Integrated System for Autonomous On-Board Absolute and Relative Spacecraft Navigation," The 4S Symposium 2018, Sorrento, Italy, 2018.

[43] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[44] A. Gold, A. Rangarajan, C.P. Lu, S. Pappu and E. Mjoisness, "New algorithms for 2D and 3D point matching: pose estimation and correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019-1031, 1998.

[45] D. De Menthon and P. David, "SoftPOSIT Demonstration Code," University of Maryland, College Park, MD, 2003.

[46] "Satbeams," [Online]. Available: https://www.satbeams.com/satellites?id=2223. [Accessed 24 August 2016].