Monocular-Based Pose Determination of Uncooperative Space Objects

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

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

Abstract

Vision-based methods to determine the relative pose of an uncooperative orbiting object are investigated in applications to spacecraft proximity operations, such as on-orbit servicing, spacecraft formation flying, and small bodies exploration. Depending on whether the object is known or unknown, a shape model of the orbiting target object may have to be constructed autonomously in real-time by making use of only optical measurements. The Simultaneous Estimation of Pose and Shape (SEPS) algorithm that does not require a priori knowledge of the pose and shape of the target is presented. This makes use of a novel measurement equation and filter that can efficiently use optical flow information along with a star tracker to estimate the target's angular rotational and translational relative velocity as well as its center of gravity. Depending on the mission constraints, SEPS can be augmented by a more accurate offline, on-board 3D reconstruction of the target shape, which allows for the estimation of the pose as a known target. The use of Structure from Motion (SfM) for this purpose is discussed. A model-based approach for pose estimation of known targets is also presented. The architecture and implementation of both the proposed approaches are elucidated and their performance metrics are evaluated through numerical simulations 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

1 Introduction

Many of more than 22,000 known and catalogued orbiting objects are uncooperative [1]. In addition, there are also a significant number of other orbiting objects whose geometric appearance is unknown. In missions of On-Orbit Servicing (OOS), such as manipulation, relocation, active debris removal (ADR), or simply observation of these objects, as well as in missions of small body exploration, it is necessary to operate in close proximity and perform relative navigation maneuvers. Communication delays or simply poor coverage do not permit the use of ground-based control commands. Hence, relative navigation has to be performed autonomously, without control from ground stations. The recent study [2] provides a review of the state-of-the-art approaches and algorithms specifically developed for on-board spacecraft pose determination using electro-optical (EO) sensor measurements. EO sensors include all devices that are 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 prior 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*. The RF/GNSS-based approach can actually be very convenient in Formation Flying (FF) missions, where each spacecraft of the formation can interact and exchange information with its neighbor [10, 11]. 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 and CCRs) might be preferred to an actively cooperative target. Artificial markers and a monocular camera could be used in order to minimize the power consumption, mass, and volume of an on-board navigation system or to support deep-space missions where GNSS signals are not available. In addition, in order to overcome individual limitations and to provide a more accurate and robust navigation solution, GNSS/RF-based systems and Vision-based systems can be synergistically integrated as proposed in our recent [9].

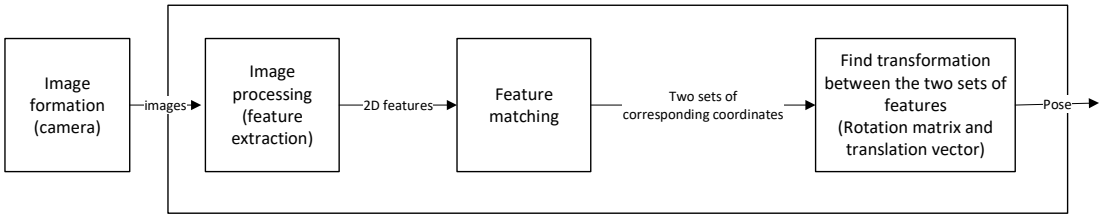

Fig. 1: Vision-based pose determination steps.

When dealing with *uncooperative* targets instead, such as space debris, asteroids, comets, or any other objects 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 shown 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 [13]: (a) image processing and feature extraction; (b) feature matching; and (c) 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. It is possible to extract region features [25], line or curve features [12], and point features [13]. In the context of pose determination, correspondence or feature matching entails 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 of uncooperative orbiting objects as a suitable approach for small satellites 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 consider both the cases of unknown and known uncooperative targets.

1.1 *Related Work*

Various monocular-vision-based approaches exist in the literature, and key distinctions can be made as to whether the target geometry is known or unknown. Many studies can be found on model-based spacecraft pose determination, which assume a target to be uncooperative with *known* model parameters. In Section 4 of [2], a complete review of the most related work is provided, in conjunction with proposed approaches.

Fewer prior studies on vision-based pose determination of uncooperative *unknown* orbiting objects, similar to the one here proposed in Section 2, exist in the literature. In [14], an approach to simultaneously estimate the 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 [15] proposes a feature-based SLAM (Simultaneous Localization and

Mapping) 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 linear relative dynamics 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 or Structure from Motion (SfM) is presented in [16], 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 [17], the RANSAC algorithm is proposed in combination with an Extended Kalman Filter-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 [18] is proposed in [19] 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 [20] to estimate the relative kinematic state and the ratios of the inertia matrix components. Finally in [21], 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 [24, 25].

1.2 *Contribution*

This paper is an updated and extended version of our previous work [22]. The objective 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 small bodies), we first investigate a monocular pose estimation approach that does not require any knowledge of the target.

For such a goal, the filter-based SLAM architectures and algorithms (see [26, 27, 28, 29]) were modified to perform robust Simultaneous Estimation of Pose and Shape, hereafter *"SEPS."* This method is suitable for real-time applications and for targets completely unknown.

Being a monocular camera a bearing-only sensor, it is generally not possible to measure the distance of the extracted features and therefore recover the scale of the reconstructed map. To solve this limitation, in addition to monocular images, a single beam LIDAR measurement is also assumed. Unlike the other techniques proposed in the literature, SEPS does not propagate the attitude dynamics of the target but only its kinematics, thereby estimating its angular velocity from optical flow. As a consequence, the estimation of the inertia matrix of the target is no longer required, eliminating the risk of divergence due to the additional uncertainty introduced in the estimation process.

In addition, we derive a novel measurement equation and filter that can efficiently use optical flow information along with a star tracker to estimate the target's relative angular and translational velocity vectors, and its center of gravity.

Pose and shape of the target may or may not be estimated simultaneously; if it is possible to safely orbit in proximity of the target and acquire a number of adequately textured images representative of the target's complete geometry, the shape could be reconstructed from scratch or refined after SEPS by processing those images offline directly on board. A SfM implementation is considered for this goal. Once the shape of the target has been constructed, the pose can be estimated with a model-based approach. Such an approach is adopted for the pose estimation of a known target as it requires the knowledge of a simplified model of its geometry.

Both approaches were implemented and tested with numerical simulations using the same dataset of images in input, synthetically generated for a given chaser/target relative trajectory in GEO and a given target spacecraft.

The paper is organized as follows. Following the introduction, Section 2 elucidates the architecture, the image processing, and the estimation steps derived to perform pose estimation of an unknown target. Section 3 presents the image-to-model approach investigated for pose estimation of known targets. Section 4 reports the simulated performance of the described algorithms. Section 5 discusses the use of the described pose determination methods in different phases of a rendezvous mission. Finally, in Section 6, the conclusion is given.

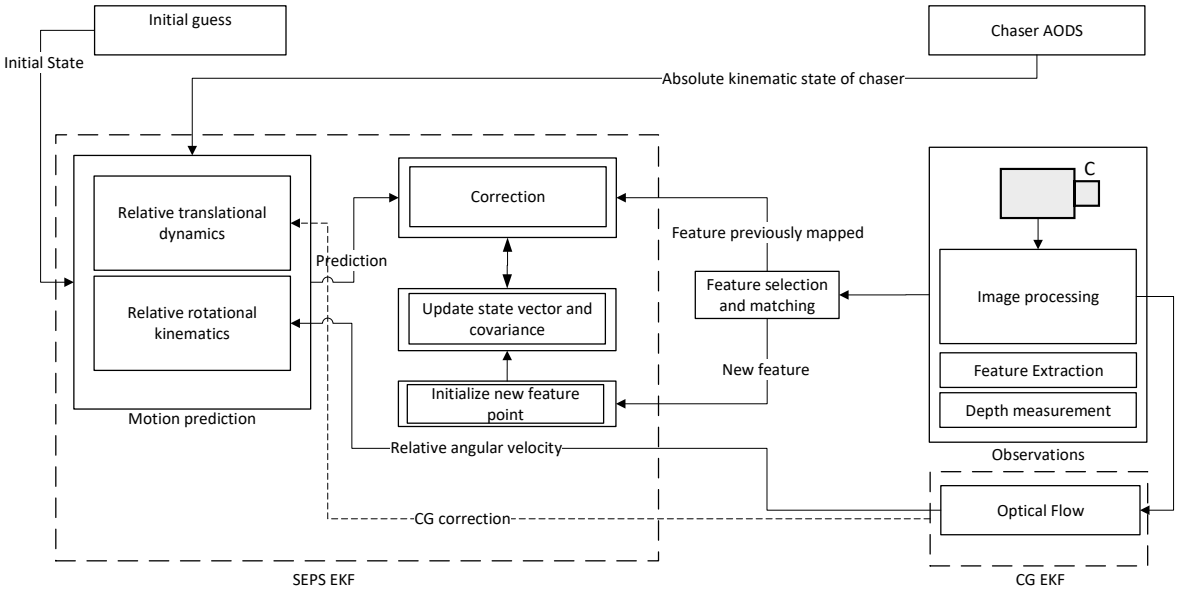

Fig. 2: Monocular-vision-based pose determination of an unknown orbiting target.

2 Pose and Shape Estimation of an Unknown Target

2.1 *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 Simultaneous Localization and Mapping (SLAM). In case of pose determination of an orbiting target with respect to an orbiting chaser (hosting the camera), the problem becomes more challenging since the target is not stationary with respect to the world frame. In the following sections, we detail an architecture and implementation of the SEPS filter for unknown targets, developed for this study.

2.1.1 *Architecture*

Figure 2 illustrates the architecture of the 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, as 2D coordinates in pixels. These serve as the observations in the correction step of our nonlinear SEPS filter, adopted as a sequential estimator, suitable for real time estimation, at relatively high rates. Although the architecture relies mainly on a monocular camera, the use of an additional low power single-segment LIDAR is assumed at each time step to determine the distance of one of the feature points, tracked over consecutive images. This depth measurement is then used to recover the scale of the reconstructed map. The same depth measurement is also used, with a larger uncertainty, as initial depth guess (prior) of the new extracted feature points, which are not part of the reconstructed map. The relative translational and rotational motion 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.

2.1.2 *Image Acquisition and Processing*

For this paper, synthetic images, shown in Fig. 3, were generated with the open source 3D suite Blender [43] using an existing model of the Aura spacecraft [44], according to the relative trajectories defined in Section 4.1. Since the main goal of this study is to investigate different pose estimation approaches (given a number of detected and matched

features), in order to simplify the problem of feature detection, a single parallel light source was used to simulate only the solar illumination. The final images were rendered using ray-tracing.

At each time step, images are acquired sequentially and converted to grayscale. In the block diagram shown in Fig. 2, observations lead to feature detection and matching. The image processing subsystem has to break down the images into information that can be analyzed and manipulated. In feature detection methods, the most significant geometric features of the target object are extracted. For simplicity, in the present paper, we adopt the well-known algorithm Harris corner detector [36]. Figure 4 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.

It is important to note that actual spacecraft images can have high contrasts, low resolution, and low signal-to-noise ratio, which presents a separate challenge of processing them for meaningful data. In that case, using the simple Harris corner detector may not be effective.

Instead of a corner detector or in addition to it, other image processing algorithms can be used for an effective feature extraction. Some of these are gradient-based for example, such as the Hough transform [37] and Canny edge detector [38].

Computer vision techniques using features with descriptors, such as SIFT [40], have also been introduced in spacecraft applications. The existing work [2, 35] questions the viability of SIFT for onboard computation, noting its heavy computational burden, ambiguity in feature detection and matching against symmetric spacecraft, requirements of high input image quality and low image noise for optimal performance. Hence, there have been other feature descriptors, such as SURF, AKAZE, and ORB [41, 42] developed to lower the computational burden.

However, in general, applying only one of these methods directly to real images may not be successful since these methods are indiscriminate towards background and foreground. As suggested in our prior work [59], a combination of different methods should be used.

In [59], we proposed a new robust feature detection algorithm able to deal with actual space imagery, characterized by variable and unfavorable illumination conditions. The proposed strategy is based on a gradient based filter for background elimination, on the use of three multiple processing streams and on the synthesis of polylines to reduce the number of outliers.

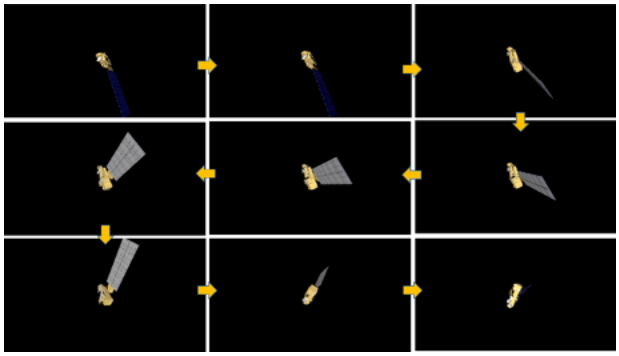

Fig. 3: Part of the synthetic image sequence generated with the open source 3D creation suite Blender [43] from the 3D model of the Aura spacecraft [44].

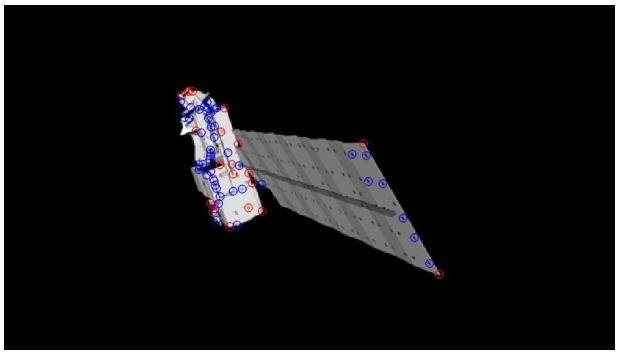

Fig. 4: Multiple Harris points extracted from an image.

2.1.3 *Estimator Implementation*

In our implementation, the state vector $\boldsymbol{x}$ includes relative position, relative velocity, and relative attitude of the chaser with respect to the target as well as 3D positions of the extracted features of the target. Unlike in the state-of-the-art approaches, here all components of the state are expressed in the target frame. In addition (also in contrast to the implementation described in our previous work [22]), in the proposed formulation, we do not estimate the relative angular velocity between chaser and target as part of the filter state, but instead we determine it from optical flow [23], as explained in Section 2.2. The advantage, with respect to the state-of-the-art, is that in this case there is no need to estimate the target's principal moments of inertia (parametrized and up to a scale [19]), whose uncertainty significantly affects the robustness of the filter estimate. Another benefit of this approach is that the size of the fil-

ter state is smaller with a consequent reduction in computational burden. Therefore, the state vector is defined as follows.

$$\boldsymbol{x} = \left[\boldsymbol{c}^\top, \boldsymbol{s}^\top\right]^\top. \tag{1}$$

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

$$\boldsymbol{c} = \left[(\boldsymbol{r}^T_{C/T})^\top, (\boldsymbol{v}^T_{C/T})^\top, (\boldsymbol{q}^T_C)^\top\right]^\top, \tag{2}$$

where $\boldsymbol{r}^T_{C/T}$ and $\boldsymbol{v}^T_{C/T}$ are respectively the relative position and velocity vectors of the chaser with respect to the target expressed in the target frame.

In general, we will use the following rule for the notation of vectors and frames throughout the work: $\mathbf{v}^C_{A/B}$ represents a vector describing a physical quantity $\mathbf{v}$ of the frame A measured with respect to the frame B and expressed in the frame C.

In Eq. (2), we used the subscripts C and T to denote the chaser or target and the superscript T to represent the target frame. Similarly, we will use I to denote the Earth-centered inertial (ECI) frame.

$\boldsymbol{q}^T_C$ is a quaternion representing a frame rotation from the target to the chaser; equivalently, the rotation matrix $\boldsymbol{R}^T_C = \boldsymbol{R}(\boldsymbol{q}^T_C)$ associated with this quaternion acts as a vector coordinate transformation from the chaser frame to the target frame, e.g., $\mathbf{v}^T = \boldsymbol{R}^T_C\mathbf{v}^C$. With the latter interpretation in mind, the subscript and superscript of quaternions and rotation matrices will represent where the vector is rotated from and where it is rotated to, respectively, throughout the work.

The dimension of $\boldsymbol{c}$ is 10. In our previous implementation described in [22], $\boldsymbol{c}$ also included $\boldsymbol{\omega}^T_{C/T}$, the relative angular velocity of the chaser with respect to the target, and two inertia ratios of the target used to parameterize its inertia matrix (only observable up to a scale in case of torque free motion).

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

Therefore, when not estimating the relative angular velocity $\boldsymbol{\omega}^T_{C/T}$ and the parameterized inertia matrix, $\boldsymbol{x}$ is a $10 + 3n$ element vector instead of a $15 + 3n$. The states are estimated by means of a nonlinear filter 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. In this paper, we use the well-known Extended Kalman Filter (EKF), however any nonlinear filter might be used.

Suppose we have a continuous-time system

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

where $\boldsymbol{x}$ is a system state vector defined in Eq. (1), $\boldsymbol{f}$ is a dynamics model (nonlinear in general), $\boldsymbol{u}$ 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 $\boldsymbol{u} = \mathbf{0}$ because there is no control involved.

The discrete time prediction and update steps of an EKF can be expressed as follows:

$$\hat{\boldsymbol{x}}^-_k = \hat{\boldsymbol{x}}^+_{k-1} + \int_{k-1}^{k} \boldsymbol{f}(\boldsymbol{x}, t)dt, \tag{3}$$

$$\boldsymbol{P}^-_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}^+_{k-1}\boldsymbol{\Phi}^\top_{k-1} + \boldsymbol{Q}_{k-1}, \tag{4}$$

$$\boldsymbol{K}_k = \boldsymbol{P}^-_k\boldsymbol{H}^\top_k(\boldsymbol{H}_k\boldsymbol{P}^-_k\boldsymbol{H}^\top_k + \boldsymbol{R}_k)^{-1}, \tag{5}$$

$$\hat{\boldsymbol{x}}^+_k = \hat{\boldsymbol{x}}^-_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}^-_k)) = \hat{\boldsymbol{x}}^-_k + \boldsymbol{K}_k\boldsymbol{z}^-_k, \tag{6}$$

$$\boldsymbol{P}^+_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}^-_k. \tag{7}$$

where subscript k indicates the k-th time step. Also, superscripts $+$ and $-$ respectively denote a priori and a posteriori estimates or error covariance matrices. The $\boldsymbol{P}_k$, $\boldsymbol{Q}_k$, $\boldsymbol{R}_k$, $\boldsymbol{\Phi}_k$, and $\boldsymbol{H}_k$ matrices denote the estimate, process noise, and sensor noise convariance matrices, the state transition matrix, and the measurement matrix, respectively. Finally, $\boldsymbol{K}_k$ is the Kalman gain while $\boldsymbol{y}_k$ is the measurement vector and $\boldsymbol{z}^-_k$ is the innovation measurement vector.

2.1.4 *Initialization*

At the very first time step, there is only detection of unmapped features. The filter has to be initialized with a certain pose, as well as with the coordinates of the first extracted features and of the center of gravity (CG) of the target to be used as the origin of the target frame.

By extracting and tracking at least 5 feature points between subsequent frames, we determine their relative pose by solving a 2D to 2D matching problem, as well as their 3D coordinates up to a scale. A detailed description and formulation of this initialization approach is provided in the previous study of some of the authors [33], where the 5-point Stewenius algorithm [34] is used with random sample consensus (RANSAC) to determine inliers, followed by least-square optimization, and use of star tracker data to determine the relative positions in the ECI frame.

If the target has a principal axis rotation, only one axis will be determined and the filter can be initialized with the CG of the target anywhere on that axis. In case the target has a non-principal axis rotation, then a different axis can be determined in a subsequent frame and intersected with the first solution to determine the CG. The initialized CG can be then corrected with the filter described in Section 2.2.2.

It is important to observe that in case of high rotational dynamics of the target, the image acquisition rate must be high enough to match at least 5 distinct features in two consecutive images. Depending on the on-board computational resources, the minimum time between two consecutive images with at least 5 common landmarks might be too short for performing on board feature extraction and matching. As a consequence, for a successful initialization, it is fundamental to maximize the on-board computational resources and minimize the computational load.

2.1.5 *Prediction Step*

The prediction step is performed by integrating over time the relative dynamics between the chaser and target as in Eq. (3). Note that the dot notation in this section implies time differentiation with respect to the target frame,

$$= \frac{d\boldsymbol{x}}{dt}\Big|_T = \frac{d}{dt}\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{s}\end{bmatrix}_{|_T}, \quad (8)$$

where the subscript T after vertical bars represent the reference frame in which differentiation is done. Note that $\dot{\boldsymbol{s}} = \boldsymbol{0}$ because the landmarks are stationary in the target frame. Therefore, we only need to know $\dot{\boldsymbol{c}}$ in order to propagate the relative dynamics.

Among several relative translational and rotational spacecraft dynamics models proposed in the literature, our model is based on the ones proposed in [46, 47] for propagation of $\boldsymbol{c}$. In [47], relative dynamics between a leader and follower were developed, and the propagation of translational dynamics required the knowledge of several orbital parameters of the leader, which is not available from an uncooperative target. Hence, we chose the chaser to serve as the leader and the target as the follower and assumed the required orbital parameters of the leader (or chaser) is provided by an external absolute navigation system of the chaser.

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. On the other hand, the states propagated in the derived equations of [47] are all expressed in the leader frame, which is the chaser frame in our case, hence they are not directly applicable to our formulation. For this reason, we re-derive the relative dynamics equations from the follower-centric perspective, which corresponds to the target-centric perspective in our case, such that they can be used in the prediction step of the EKF.

The relative translational dynamics, when expressed in the chaser frame with respect to the chaser, is independent of the relative attitude dynamics [47]. 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 $\boldsymbol{q}_C^T = [q_x, q_y, q_z, q_w]^\top$ denote a quaternion vector 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 $\boldsymbol{R}_C^T = \boldsymbol{R}(\boldsymbol{q}_C^T)$.

In our formulation, we assume $\boldsymbol{\omega}_{C/T}^T$ is determined from optical flow as described in Section 2.2.3. Then the filter estimates $\boldsymbol{q}_C^T$ by making use of the quaternion kinematics, given as follows:

$$\dot{\boldsymbol{q}}_C^T = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{T/C}^T)\boldsymbol{q}_C^T, \quad (9)$$

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

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y & -\omega_x \\ -\omega_z & 0 & \omega_x & -\omega_y \\ \omega_y & -\omega_x & 0 & -\omega_z \\ \omega_x & \omega_y & \omega_z & 0 \end{bmatrix} \quad (10)$$

for $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$.

Let us now consider the relative translational dynamics. Let $\boldsymbol{r}_{C/I}^C$ and $\boldsymbol{r}_{T/I}^T$ 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 $\boldsymbol{r}_{C/T}^T = \boldsymbol{r}_{C/I}^T - \boldsymbol{r}_{T/I}^T = \boldsymbol{R}_C^T\boldsymbol{r}_{C/I}^C - \boldsymbol{r}_{T/I}^T$. We assume $\boldsymbol{r}_{C/I}^C = [r_C, 0, 0]^\top$ 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, $\boldsymbol{r}_{T/I}^T$ of an uncooperative target is not known and it is estimated from the knowledge of $\boldsymbol{r}_{C/I}^C$ and estimations of $\boldsymbol{r}_{C/T}^T$ and $\boldsymbol{R}_C^T$. The relative velocity $\boldsymbol{v}_{C/T}^T$ in the target frame is defined as

$$\begin{aligned}\boldsymbol{v}_{C/T}^T &= \left(\left.\frac{d\boldsymbol{r}_{C/T}}{dt}\right|_T\right)^T = \left(\left.\frac{d\boldsymbol{r}_{C/T}}{dt}\right|_I\right)^T - \boldsymbol{\omega}_{T/I}^T \times \boldsymbol{r}_{C/T}^T \\ &= \boldsymbol{R}_C^T\boldsymbol{R}_I^C\left(\left.\frac{d\boldsymbol{r}_{C/T}}{dt}\right|_I\right)^I - (\boldsymbol{R}_C^T\boldsymbol{\omega}_{C/I}^C - \boldsymbol{\omega}_{C/T}^T)\times\boldsymbol{r}_{C/T}^T,\end{aligned} \tag{11}$$

where the angular velocity of the uncooperative target can be estimated by $\boldsymbol{\omega}_{T/I}^T = \boldsymbol{R}_C^T\boldsymbol{\omega}_{C/I}^C - \boldsymbol{\omega}_{C/T}^T$. In the above, $(\frac{d\boldsymbol{r}}{dt}|_A)^B$ refers to a time differentiation of $\boldsymbol{r}$ with respect to the frame A, whose components are expressed in the frame B. The relative acceleration $\boldsymbol{a}_{C/T}^T$ in the target frame is computed as

$$\begin{aligned}\boldsymbol{a}_{C/T}^T &= \left(\left.\frac{d^2\boldsymbol{r}_{C/T}}{dt^2}\right|_T\right)^T = \left(\left.\frac{d^2\boldsymbol{r}_{C/T}}{dt^2}\right|_I\right)^T - 2\boldsymbol{\omega}_{T/I}^T \times \boldsymbol{v}_{C/T}^T \\ &- \left(\left.\frac{d\boldsymbol{\omega}_{T/I}}{dt}\right|_T\right)^T \times \boldsymbol{r}_{C/T}^T - \boldsymbol{\omega}_{T/I}^T \times (\boldsymbol{\omega}_{T/I}^T \times \boldsymbol{r}_{C/T}^T).\end{aligned} \tag{12}$$

Note that

$$\left(\left.\frac{d^2\boldsymbol{r}_{C/T}}{dt^2}\right|_I\right)^T = \left(\left.\frac{d^2\boldsymbol{r}_{C/I}}{dt^2}\right|_I\right)^T - \left(\left.\frac{d^2\boldsymbol{r}_{T/I}}{dt^2}\right|_I\right)^T, \tag{13}$$

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

$$\left(\left.\frac{d^2\boldsymbol{r}_{C/I}}{dt^2}\right|_I\right)^T = -\frac{\mu}{r_C^3}\boldsymbol{R}_C^T\boldsymbol{r}_{C/I}^C, \tag{14}$$

$$\left(\left.\frac{d^2\boldsymbol{r}_{T/I}}{dt^2}\right|_I\right)^T = -\mu\frac{\boldsymbol{R}_C^T\boldsymbol{r}_{C/I}^C - \boldsymbol{r}_{C/T}^T}{\|\boldsymbol{R}_C^T\boldsymbol{r}_{C/I}^C - \boldsymbol{r}_{C/T}^T\|^3}. \tag{15}$$

By substituting Eqs. (13)–(15) into Eq. (12), we obtain the desired relative translational dynamics expressed in the target frame as

$$\begin{aligned}\boldsymbol{a}_{C/T}^T = &-\frac{\mu}{r_C^3}\boldsymbol{R}_C^T\boldsymbol{r}_{C/I}^C + \mu\frac{\boldsymbol{R}_C^T\boldsymbol{r}_{C/I}^C - \boldsymbol{r}_{C/T}^T}{\|\boldsymbol{R}_C^T\boldsymbol{r}_{C/I}^C - \boldsymbol{r}_{C/T}^T\|^3} \\ &- 2\boldsymbol{\omega}_{T/I}^T \times \boldsymbol{v}_{C/T}^T - \left(\left.\frac{d\boldsymbol{\omega}_{T/I}}{dt}\right|_T\right)^T \times \boldsymbol{r}_{C/T}^T \\ &- \boldsymbol{\omega}_{T/I}^T \times \left(\boldsymbol{\omega}_{T/I}^T \times \boldsymbol{r}_{C/T}^T\right).\end{aligned} \tag{16}$$

In the above, all of the quantities to compute $\boldsymbol{a}_{C/T}^T$ are available either from the filter states or the absolute navigation system of the chaser except $\left(\frac{d\boldsymbol{\omega}_{T/I}}{dt}|_T\right)^T$ that can be computed by numerically differentiating $\boldsymbol{\omega}_{T/I}^T$.

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

$$\dot{\boldsymbol{c}} = \left.\frac{d\boldsymbol{c}}{dt}\right|_T = \frac{d}{dt}\begin{bmatrix}\boldsymbol{r}_{C/T}^T \\ \boldsymbol{v}_{C/T}^T \\ \boldsymbol{q}_C^T\end{bmatrix}_{|T} = \begin{bmatrix}\boldsymbol{v}_{C/T}^T \\ \boldsymbol{a}_{C/T}^T \\ \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{T/C}^T)\boldsymbol{q}_C^T\end{bmatrix}. \tag{17}$$

2.1.6 *Correction Step*

The correction step includes the following substeps. The implementation of the correction is modified from the methods proposed by Joan Solà in [45].

i. Selection of predicted feature observations to correct

ii. Feature matching with observations

iii. EKF state and covariance correction

iv. Feature deletion in case of corruption

Step i. In the filter structure, the feature observations are processed in the EKF 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 $\boldsymbol{p}^T$ in the target frame is expressed in chaser frame, by means of the predicted pose from the previous pose estimate $(\boldsymbol{R}_T^C, \boldsymbol{r}_{T/C}^T)$.

$$\boldsymbol{p}^C = \boldsymbol{R}_T^C(\boldsymbol{p}^T - \boldsymbol{r}_{C/T}^T) \tag{18}$$

Then, it is projected onto the image plane.

A perspective monocular camera associates points in 3D space, $\boldsymbol{p}^C = \begin{bmatrix}x_c & y_c & z_c\end{bmatrix}^\top$, with points in the 2D image plane, $\boldsymbol{P}(\boldsymbol{p}^C) = \begin{bmatrix}X & Y\end{bmatrix}^\top$, by means of the projection process. As illustrated in Fig. 5, 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 (focal length) from the optical center. 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 $\boldsymbol{p}^C$ in the 3D space corresponds to the intersection of the line $\overline{O\boldsymbol{p}^C}$

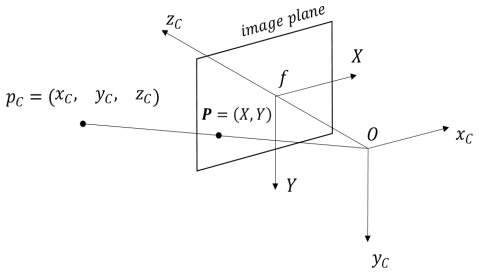

Fig. 5: Pin-hole camera model.

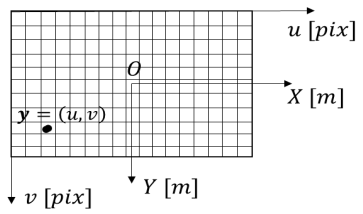

Fig. 6: Projection of points onto the image plane in metric and pixel units.

with the image plane. The image point $\boldsymbol{P}$ is obtained as follows:

$$\frac{X}{f} = \frac{x_c}{z_c}, \quad \frac{Y}{f} = \frac{y_c}{z_c} \tag{19}$$

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

$$\boldsymbol{P}(\boldsymbol{p}^C) = \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} \cdot \frac{f}{z_c}, \tag{20}$$

where z_c is the depth of the object point.

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

$$u = u_0 + s_u X, \quad v = v_0 + s_v Y \tag{21}$$

where $\begin{bmatrix} s_u & s_v \end{bmatrix}^\top$ are the horizontal and vertical pixel density in [pix/m] and $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^\top$ are the pixel coordinates of the principal point in [m] (Fig. 6). Among all the projected points, only the visible ones (i.e. in the field of view of the camera) are considered.

Once the positions of all previously mapped feature points are predicted in the camera frame, they are sorted based on the determinant of expectation covariance matrix (proportional to 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.

The continuous time measurement model can be written as

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{w}_m \tag{22}$$

where $\boldsymbol{y}$ is the noisy measurement, $\boldsymbol{x}$ is the state vector, $\boldsymbol{h}(\cdot)$ is a nonlinear 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 $(\boldsymbol{R}_S^C, \boldsymbol{r}_{S/C}^C)$, where S is used to denote the sensor camera frame.

An observation $\boldsymbol{y}_i$ is the measurement of the i-th target feature in the camera frame. The features are extracted from images of the target acquired by the camera system and then are processed. The components of $\boldsymbol{y}_i$ are the geometric parameterization of the i-th feature in the camera measurement space, in pixel (u_i, v_i).

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 [48] 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 4 illustrates a set of extracted points and a set of matched points.. Features matched to previously mapped features are in red, and features previously unmapped and newly initialized are in blue.

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 $\boldsymbol{z}_k^-$ and

innovation covariance matrix $\boldsymbol{Z}_k^-$ are the following:

$$\boldsymbol{z}_k^- = \boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-) \tag{23}$$

$$\boldsymbol{Z}_k^- = \boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^\top + \boldsymbol{R}_k \tag{24}$$

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

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^-} \tag{25}$$

Then, the Kalman gain can be calculated according to Eq. (5), while the state vector and the covariance matrix can be corrected, respectively according to Eqs. (6) and (7).

Step iv. The feature points 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.

2.1.7 *Initialization of New Features*

Features detected in an image can either be unmapped, or previously mapped and are used in the correction step of the filter. Detection of previously unmapped features was implemented according to the following procedure adapted from [45]. The 960×540 sized image is partitioned into smaller subimages of 100 pixels 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 from the subimage corresponding to that cell no feature has been matched to the predicted features). The subimage corresponding to the selected grid cell is extracted from the original image. Using the Harris algorithm [36], 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 2.1.6 under Step ii.

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 $10 + 3n$ to $10 + 3(n + 1)$.

For this goal, the new feature points are re-projected onto the 3D camera space, according to the last pose estimate, their 2D coordinates on the image plane and an initial guess in depth used as prior provided by the single beam LIDAR depth measurement. The re-projection onto the 3D space is done by inverting the observation function $\boldsymbol{h}(\cdot)$ to compute $\boldsymbol{p}^C$. The point $\boldsymbol{p}^C$ in camera frame is transformed to the corresponding point in target frame $\boldsymbol{p}^T$ and then added to the current 3D shape $\boldsymbol{s}$, part of the state vector, with higher covariance.

2.1.8 *Scale Recovery*

At each time step a single beam LIDAR provide one depth measurement. As long as relative position of the LIDAR and camera is calibrated, the depth provided can be used to determine where on the image the measured feature was. This can then be added as a custom feature, with a good depth measurement, to be tracked frame-to-frame over time. These custom features are enough to recover the scale of the constructed map since their relative 3D position covariance is much lower.

2.2 *Estimation of Angular Velocity and Center of Gravity*

For a 3D landmark moving in the world, the optical flow provides the corresponding pixel velocity vector in the image. By using the velocity vectors of the target's rigid body points, it is possible to obtain measurements of the relative rotational and translational velocities of the camera and target in the camera frame. In this section we present novel measurement equations and filter that can efficiently use flow information along with a star tracker to estimate the target's rotation, relative velocity, and CG.

2.2.1 *Measurements from Optical Flow*

Previous implementations that use optical flow to measure linear and rotational velocities combine all relative motion between the target and camera into a single measurement that assumes either a fixed camera or fixed world [13]. In this implementation, we break up the measurement into different inertial components that allow us to separate out the camera's own motion from the target's rotation and relative velocity.

Consider a camera on board the chaser moving relative to the target in the ECI frame with velocity $\boldsymbol{v}_{C/T}^I$ and the target rotating at a rate of $\boldsymbol{\omega}_{T/I}^I$. If the chaser is controlled by reaction wheels it will not have perfect pointing accuracy, and so have some drift in the attitude between subsequent frames. This will cause some movement in the image and so the flow measurements will have a bias. We assume that at time t the attitude of the camera, as measured by the star tracker, is $\boldsymbol{R}_C^I(t)$. Then the instantaneous rotational velocity of the chaser

camera in ECI at $t = 0$ can be either be approximated from two successive attitude measurements by $\boldsymbol{\omega}^I_{C/I} t = \log(\boldsymbol{R}^C_I(t)\boldsymbol{R}^I_C(0))$, where log is the standard $SO(3)$ logarithm, or by a combination of a gyroscope ($\boldsymbol{\omega}^C_{C/I}$) and attitude measurement such as from a star tracker via $\boldsymbol{\omega}^I_{C/I} = \boldsymbol{R}^I_C \boldsymbol{\omega}^C_{C/I}$. This gives the approximation $\boldsymbol{R}^C_I(t) = e^{\boldsymbol{\omega}^I_{C/I} t}\boldsymbol{R}^C_I(0)$.

Then a point on the target $\boldsymbol{p}^T$ can be measured in the normalized image coordinates as

$$\begin{aligned}\begin{bmatrix} u \\ v \end{bmatrix} &= \boldsymbol{P}(\boldsymbol{p}^C)(t) \\ &= \boldsymbol{P}(\boldsymbol{R}^C_T(\boldsymbol{p}^T - \boldsymbol{r}^T_{C/T}))(t) \\ &= \boldsymbol{P}(\boldsymbol{R}^C_I(\boldsymbol{R}^I_T \boldsymbol{p}^T - \boldsymbol{r}^I_{C/T}))(t)\end{aligned} \tag{26}$$

Taking the derivative and evaluating at $t = 0$ gives

$$\begin{aligned}\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} =& \frac{\partial \boldsymbol{P}(\boldsymbol{p}^C)}{\partial \boldsymbol{p}}(\boldsymbol{\omega}^I_{C/I} \times (\boldsymbol{R}^C_I(\boldsymbol{R}^I_T \boldsymbol{p}^T - \boldsymbol{r}^I_{C/T}) + \\ & \boldsymbol{R}^C_I(\boldsymbol{\omega}^I_{T/I} \times \boldsymbol{R}^I_T \boldsymbol{p}^T - \boldsymbol{v}^I_{C/T}))\end{aligned} \tag{27}$$

$$\begin{aligned}\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}_{t=0} =& \left.\frac{\partial \boldsymbol{P}(\boldsymbol{p}^C)}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}^C} (\boldsymbol{\omega}^I_{C/I} \times (\boldsymbol{R}^C_I(\boldsymbol{R}^I_T \boldsymbol{p}^T - \boldsymbol{r}^T_{C/T}) + \\ & \boldsymbol{R}^C_I(\boldsymbol{\omega}^I_{T/I} \times \boldsymbol{R}^I_T \boldsymbol{p}^T - \boldsymbol{v}^I_{C/T}))(0) \\ =& \left.\frac{\partial \boldsymbol{P}(\boldsymbol{p}^C)}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}^C} (\boldsymbol{\omega}^I_{C/I} \times \boldsymbol{p}^C + \boldsymbol{\omega}^C_{T/I} \times \boldsymbol{p}^C \\ & - \boldsymbol{\omega}^C_{T/I} \times \boldsymbol{r}^C_{T/C} - \boldsymbol{v}^C_{C/T}))(0)\end{aligned} \tag{28}$$

So the flow of $\boldsymbol{p}^C$ is determined by the rotation of the camera ($\boldsymbol{\omega}^I_{C/I}$), the inertial rotation of the target in the camera frame ($\boldsymbol{\omega}^C_{T/I}$), the position of the center of mass of the target in the camera frame ($\boldsymbol{r}^C_{T/C}$), and the relative velocity in the camera frame ($\boldsymbol{v}^C_{C/T}$). Since $\boldsymbol{\omega}^I_{C/I}$ is known, we have a linear equation with the 6 unknowns being $\boldsymbol{\omega}^C_{T/I}$ and the combined quantity $\boldsymbol{R}^I_C(\boldsymbol{\omega}^C_{T/I} \times \boldsymbol{r}^C_{T/C} + \boldsymbol{v}^C_{C/T}) = \boldsymbol{\omega}^I_{T/I} \times \boldsymbol{r}^I_{T/C} + \boldsymbol{v}^I_{C/T}$. As long as there is no degeneracy, 3 points will give us 6 equations, allowing us measurements of these unknowns. As long as there are dynamics in the rotation, the varying $\boldsymbol{\omega}^I_{T/I}$ will make $\boldsymbol{r}^I_{T/C}$ observable after multiple measurements over time. Otherwise the CG is observable up to its position along a fixed rotation axis.

2.2.2 *Estimation of Center of Gravity*

The CG and translational velocity of the target in the ECI frame are estimated with a second filter, independent from the main one described in Section 2.2. This filter relying on the measured optical flow is characterized by the following state vector $\boldsymbol{x}$ and measurement vector $\boldsymbol{y}$.

$$\boldsymbol{x} = \left[(\boldsymbol{r}^C_{T/C})^\top, (\boldsymbol{v}^C_{T/C})^\top\right]^\top \tag{29}$$

$$\boldsymbol{y} = \left[\boldsymbol{\omega}^C_{T/I} \times \boldsymbol{r}^C_{T/C} + \boldsymbol{v}^C_{C/T}\right] \tag{30}$$

In our implementation, an EKF with the same formulation described in Section 2.1.3 was adopted, using the linear model proposed in [49] for the translational dynamics required to propagate the state. The state is initialized by using the translational velocity observed from optical flow and the CG location as described in Section 2.1.4. Fig. 7 shows the architecture of the filter.

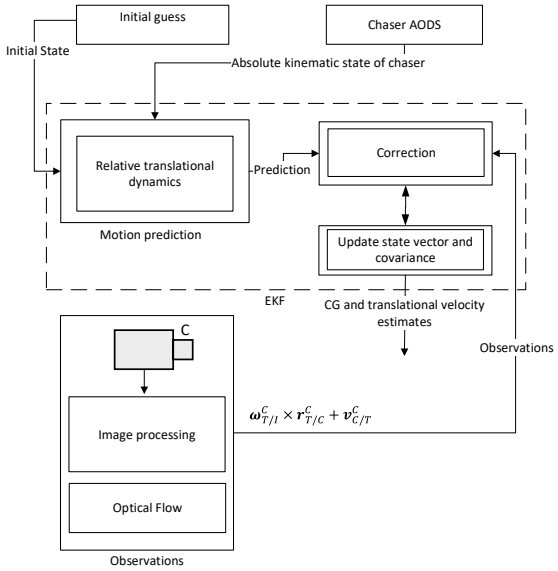

Fig. 7: Filter architecture for estimation of CG (and relative velocity).

2.2.3 *Integrating with SEPS*

After filtering, we can compute the rotation required in Eq. (9) as well as the CG location of the target with respect to the camera in the target frame as $\boldsymbol{R}^T_C \boldsymbol{R}^C_I(\boldsymbol{\omega}^I_{C/I} - \boldsymbol{\omega}^I_{T/I}) = \boldsymbol{\omega}^T_{C/T}$ and $\boldsymbol{R}^T_C \boldsymbol{r}^C_{T/C} = \boldsymbol{r}^T_{T/C}$, respectively.

Therefore, the rotation rate $\boldsymbol{\omega}^T_{C/T}$ estimated from optical flow can be used to propagate the attitude in the SEPS filter of Section 2.1.3 (see the output of optical flow in Fig. 2). In addition, considering that $\boldsymbol{r}^T_{C/T} = -\boldsymbol{r}^T_{T/C}$, the target CG location $\boldsymbol{r}^T_{T/C}$ (and velocity $\boldsymbol{v}^T_{T/C}$) estimated from optical flow can

be used to correct the CG of the target in the SEPS filter (see the dashed connection line in Fig. 2) .

Note that without using the filter of 2.2.2 ("CG EKF" in Fig. 2), in the SEPS filter, the CG does not necessarily correspond to the origin of the target frame where the feature points $\boldsymbol{s} = \left[\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_n\right]^\top$ are defined. Indeed, when initializing the SEPS filter as described in Section 2.1.4, the CG location could be not observable or estimated with a large error due to a slow rotation rate. This can be corrected with the CG location estimate of the second independent filter of Section 2.2.2. Since the feature points in the SEPS filter are expressed in the target frame, when updating the CG location in the camera frame their positions do not need to be adjusted, which otherwise could introduce instability.

Another advantage of having a separate and independent estimation of the CG location and of the relative angular velocity, is that these can be estimated at a higher rate than the output rate of the SEPS filter. In fact the state estimation of the SEPS filter is definitely more expensive computationally, considering the dimension of the state. A higher output rate might be necessary to compute the optical flow in case of higher relative rotational dynamics between chaser and target.

2.3 *On-Board Offline 3D Reconstruction*

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 template 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

$$\sum_{\boldsymbol{y}} \left\| \boldsymbol{P}(\boldsymbol{R}^C_{T,k}(\boldsymbol{p}^T - \boldsymbol{r}^T_{C/T,k})) - \boldsymbol{y} \right\|^2_\Omega, \qquad (31)$$

summed over the landmark observations $\boldsymbol{y}$, where $\boldsymbol{y}$ is an image feature position of landmark $\boldsymbol{p}^T$ that was observed at time step k and Ω the expected covariance of the image feature measurements. To solve the minimization problem a typical gradient descent algorithm, such as Levenberg-Marquardt, is run until a convergence criterion. Between convergences images are added incrementally to the full reconstruction and landmarks are retriangulated to detect and remove outliers.

After the camera positions are recovered a template matching algorithm such as ZNCC or NCC (Normalized Cross Correlation) 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 use the resulting map of tracked features as a 3D model without needing to do a full dense reconstruction.

Figure 8 shows the resulting point cloud from running OpenSfM [32] on the 150 images of the simulated data (see Fig. 9). It took approximately 2 hours on an i7-3770K CPU configured to run off of 8 threads using the HaHOG (Hessian Affine feature point detector and HOG descriptor) combination. There are some dense outliers, but overall the reconstruction is accurate up to scale.

Once an accurate 3D model of the target has been estimated, in theory the pose can be determined with a model-based approach as the one described in Section 3, exactly as in the case of a known target. However, for such a goal the 3D model created via SfM or after several iterations of the SEPS filter should be simple enough and consist of a limited number of points to enable a fast 2D to 3D matching. One way to ensure that, proposed in our previous work [59], would be by robustly extracting only the points that belong to high level geometric structures identifiable in the target geometry, as polygons, or in general polylines.

3 Pose Estimation of a Known Target

3.1 *Pose Determination*

If the target geometry is known, a detailed or simplified model of it (see Fig. 11) 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. 10, adopting an "image-to-model" approach and feature extraction and matching algorithms, the pose can be estimated by matching the geometric natural features extracted from the acquired images with the

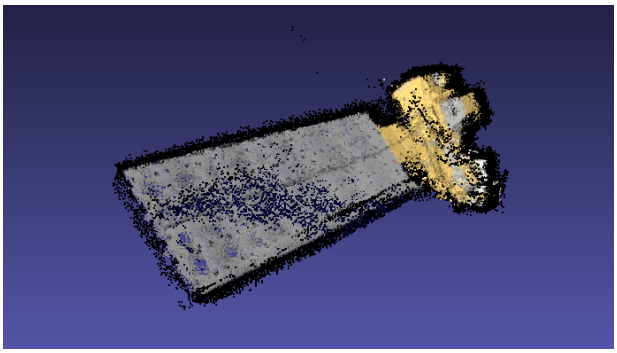

Fig. 8: Reconstructed point cloud with OpenSfM using 150 images.

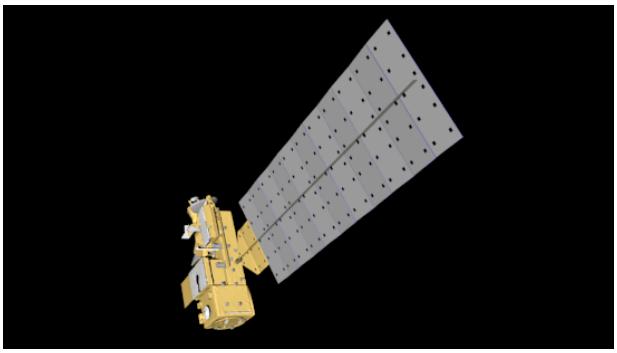

Fig. 9: A synthetic image of the Aura spacecraft, input of the simulations carried out in Section 6, generated with Blender [43], from the 3D model of the spacecraft [44].

corresponding ones of the model. Also in this case, several types of features can be extracted, such as corners, curves, or 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 Perspective-n-Point (PnP) 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, $\boldsymbol{R}_T^C$ and the translation vector from the origin of the target frame to the origin of the camera frame $\boldsymbol{r}_{C/T}^C$, as follows:

$$s_i \boldsymbol{\Pi}_i = \boldsymbol{K} \left[\boldsymbol{R}_T^C \mid \boldsymbol{r}_{C/T}^C \right] \boldsymbol{\pi}_i, \tag{32}$$

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

$$\boldsymbol{K} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{33}$$

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 [35, 50]. In this study, we adopt the PosIt PnP solver [51] 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 [35]. 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. (32) and provides a coarse $\boldsymbol{R}_T^C$ and $\boldsymbol{r}_{C/T}^C$ 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. (32) by $\frac{1}{\boldsymbol{r}_{C/T}^C(3)}$ and initializing with the SOP assumption $\frac{s_i}{\boldsymbol{r}_{C/T}^C(3)} = 1$, the first two rows become a linear system of equations with eight unknowns, which can be solved with $n = 4$. Then, $\frac{s_i}{\boldsymbol{r}_{C/T}^C(3)}$ can be computed from the estimated $\boldsymbol{R}_T^C$ and $\boldsymbol{r}_{C/T}^C$. The computation of $\boldsymbol{R}_T^C$ and $\boldsymbol{r}_{C/T}^C$, 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 [54], we used the RANSAC algorithm [55] for pose acquisition, and the Soft-Assign strategy [56] for pose tracking (adopting the SoftPosit implementation of [57]).

Note that, although not done in this study, it is possible to refine the pose estimate by filtering it with a translational and rotational model. A simple approach (loose integration) is fusing the vision-based

pose (relative position and attitude) estimated with a model-based approach with the corresponding pose predicted using the dynamics models, through a non-linear filter for example.

A more complex but more effective solution instead is filtering directly the visual feature extracted with the corresponding ones predicted by a dynamics model (tight integration), which is essentially equivalent to the filter formulation proposed in Section 2.1, assuming a known 3D model of the target and therefore not including the reconstructed features $\boldsymbol{s}$ in the state vector $\boldsymbol{x}$ (in Eq. (1)) but only the pose state $\boldsymbol{c}$.

It is important to highlight that the success of the feature matching, most of all the one in acquisition strongly depends on the illumination condition and on the geometry of the target and its on-board model.

The problem of unfavorable lighting condition is extensively discussed in our previous work [59]. Moreover, symmetry in the target geometry can lead to a significant bias in pose acquisition (more specifically in the attitude estimate), due to the existence of more than one pose solution, given a certain set of matched 2D to 3D features. If for example the target has two opposite faces almost identical, the pose estimation could be affected by an error of 180 deg in attitude.

For this reason to avoid any risk of ambiguity, it is important to design an appropriate feature extraction method (customized for the target geometry) able to detect features (or more complex structures, combinations of features) that exist only once in the geometry and in the on-board model of the target, (e.g. an antenna, an appendix, a nozzle, or a specific polygonal structure, etc). In our previous work [59], we propose an advanced and robust feature extraction strategy to detect polylines, seen as higher level features, combination of lower level features as segments and points. In another work of the first author [52], a Circular Hough transform is used to detect specific target components.

It is also important to mention that alternative completely different approaches not deterministic but learning-based [53] have been proposed in the literature very recently that make use of a Convolutional Neural Network to solve the problem of pose initialization.

3.2 *3D Model*

Figs. 11 and 12 illustrate the points of a simplified geometric model of the target we used for model-based pose determination. Such a model 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 during the 2D to 3D matching process.

In particular, the model illustrated in Fig. 11 and 12, was constructed by manually selecting representative points of the AURA spacecraft 3D model [44], 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.

Another effective and complementary strategy to reduce ambiguity in matching and to increase the chance of successful feature extraction and matching is to build the 3D model to be used on board based on the feature extractor method that will be adopted during the mission, which essentially means building a 3D model of the features that most likely will be extracted on board during the mission.

As highlighted in the previous section, a model can be built directly on board, with several iterations of SEPS or via SfM (as described respectively in Sections 2.1 and 2.3).

4 Simulated Performance

The performance of the two approaches described respectively in Sections 2 and 3 were simulated given the same data set in input.

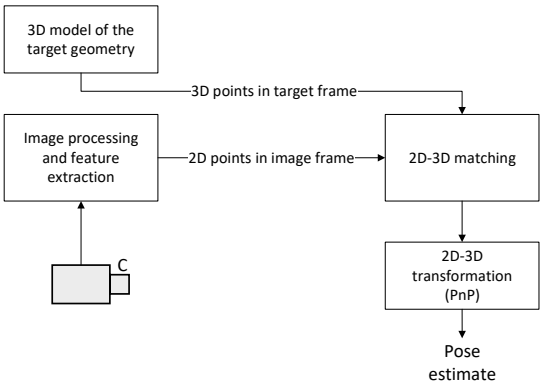

Fig. 10: Monocular model-based pose determination of known orbiting target.

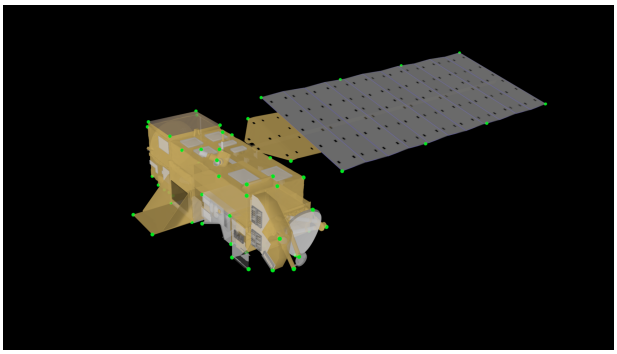

Fig. 11: 3D points representing a simplified geometrical model of the target spacecraft illustrated in Fig. 9.

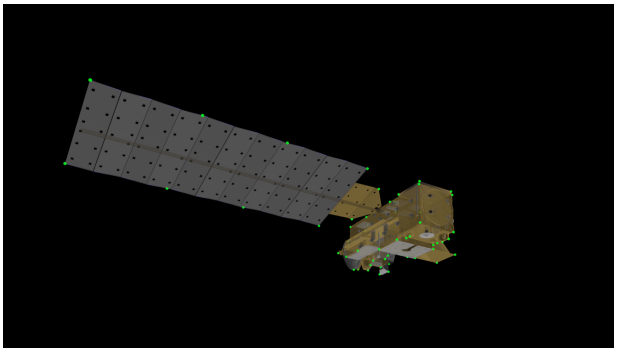

Fig. 12: 3D points representing a simplified geometrical model of the target spacecraft illustrated in Fig. 9.

Chaser Orbit	GEO (IntelSat)
Target initial separation	15 m along chaser velocity axis
Target initial relative rotation	1 deg/s around one principal axis
Camera boresight	aligned with chaser velocity axis
Initial relative position and velocity estimation errors	5 m and 0.1 m/s
Initial relative attitude estimation error	10 deg

Table 1: Simulation assumptions.

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 [58]) 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: $f \cdot s_u = f \cdot s_v = 300$ where f is the focal length and s_u and s_v are the scale factors relating image pixels to distance. The image center is $(480, 270)$ and the image size is $(960, 540)$. Additional details are reported in Table 1. As already highlighted in the previous sections, as for input of the image processing, we used a sequence of images synthetically generated according to the predetermined relative trajectory.

4.2 *Pose Estimation of an Unknown Target*

The SEPS approach was tested, following a coarse tuning of the covariance matrices. 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. 13 shows the norm of the relative 3D position estimation error. Figure 14 shows the relative attitude estimation error, as the angle describing the magnitude of the rotation error in an axis-angle representation. Figure. 15 shows the norm of the relative 3D velocity estimation error.

The estimation convergence is proved for all kinematic components of the state. The attitude and velocity estimates converge to the true relative state, with an error smaller than 1 deg and 0.01 m/s respectively, whereas the position estimate after convergence is affected by a small bias of approximately 0.3 m.

As shown in Fig. 16 and Fig. 17 respectively, from optical flow the angular velocity is estimated with an error of approximately 0.01 deg/s, while the CG with an error smaller than 0.3 m.

In SEPS, the average run time of one estimation step, in MATLAB on a 1.8 GHz Intel Core i7-8550U processor, for an average number of 147 extracted points, is 0.3 s.

4.3 *Pose Estimation of a Known Target*

Figures 18 and 19 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 in the SEPS strategy.

Note that velocity and angular velocity are not a

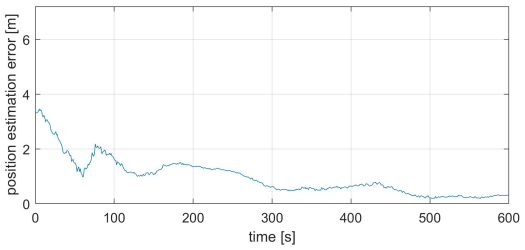

Fig. 13: Norm of the relative 3D position estimation error, with SEPS approach.

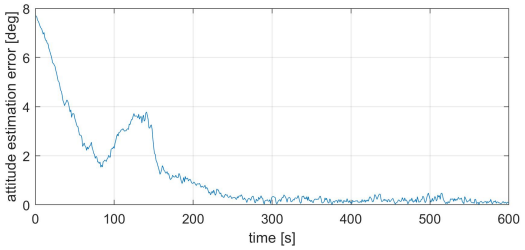

Fig. 14: Relative attitude estimation, with SEPS approach.

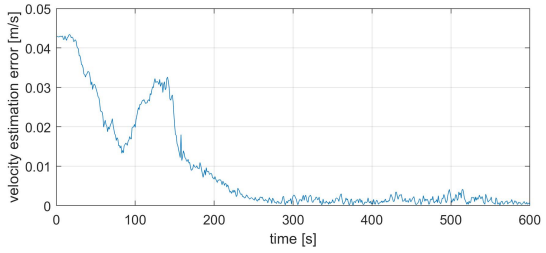

Fig. 15: Norm of the relative 3D velocity estimation error, with SEPS approach.

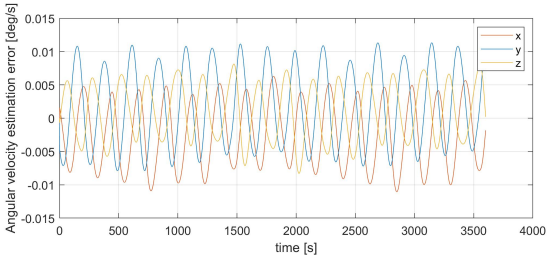

Fig. 16: Norm of the relative 3D angular velocity estimation error when using optical flow.

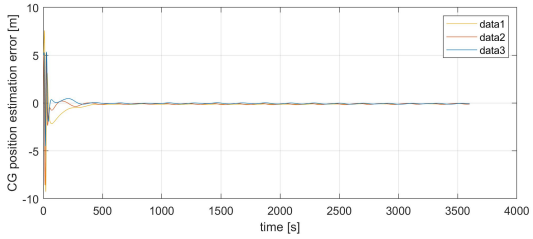

Fig. 17: CG position estimation error using the filter described in Section 2.2.2

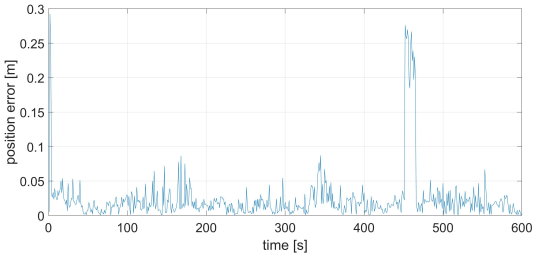

Fig. 18: Norm of the relative 3D position estimation error, with model-based approach.

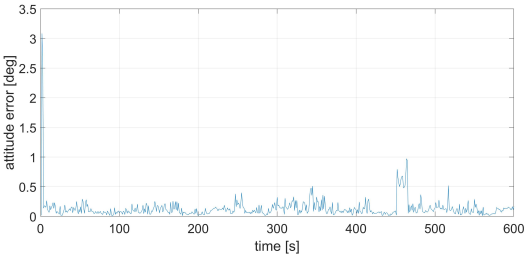

Fig. 19: Relative attitude estimation error, with model-based approach.

direct output of the PnP solver, which only estimates the 3D to 2D transformation from target to camera and does not include any information about how the pose changes over time.

The average run time of one estimation step of PoSit, in MATLAB on a 1.8 GHz Intel Core i7-8550U processor, for an average number of 147 extracted points, is 0.05 s. However, at the very beginning of the pose estimation process, when no knowledge of the target previous pose is available, to match the 2D points in the image plane with the 3D points of the model, the RANSAC algorithm takes a relatively long time, in our numerical simulations, about 200 s.

Moreover, it is important to highlight that when extracting features from actual space images with unfavorable illumination conditions, the ratio outliers/inliers might be significantly larger, resulting in a longer convergence time for the PnP solver in both acquisition and tracking.

5 Discussion

5.1 *Space Rendezvous*

Space rendezvous (and docking) is usually seen as a proximity operation, where the chaser spacecraft navigates in close proximity of typically passive and uncooperative targets. In general the problem includes multiple stages.

As described in [30], the rendezvous mission can be divided into different phases, depending on the

type of activity, relative separation with the target, and type of navigation data and hardware available on board chaser and target as well.

At the very beginning of the mission (Phase I as called in [30]), it is likely that the chaser is located in a different orbit than that of the target, following a launch phase or arriving from another planet. In this case, the target is out of sight and out of contact and in Phase II, the chaser has to transfer and drift toward the orbit of the target (typically below and behind it in a slightly lower orbit).

In Phase III, the target is in sight and/or in contact and the chaser can get closer to it, possibly by simply drifting or performing small maneuvers. This drift phase may end into a first relative parking orbit, at a certain safe distance from the target (100 m to 1 km), such that there is no risk of accidental collision.

In Phase IV, the chaser can start to approach the target, now in proximity, typically horizontally along the spacecraft velocity vector (V bar) or vertically along the radial vector (R bar), making use of relative navigation sensors and very small thruster firings. This phase of 1 to 5 orbits duration ends into a second parking orbit at smaller distance from the target (100 m to 10 m).

Then different phases are possible, depending on if the final goal of the rendezvous is docking with the target, inspection, or the relative station keeping for formation flying missions [30]. Table 2 reports the different navigation approaches for each of the first five phases and the corresponding chaser-target initial separations (where $\lambda_{\max}$ is the maximum Earth central angle from the spacecraft altitude).

In the first two phases the target is neither in sight nor in contact and the chaser has to rely on a precise or approximate knowledge of its orbit and that of the target, i.e., performing Absolute Navigation (Abs Nav). As soon as the target is close enough to not be obstructed by the Earth (or any other planet) horizon, it is possible to start a coarse vision-based Relative Navigation (Rel Nav), where the chaser can see the target moving as a point of light. According to [30], accurate relative navigation begins in Phase IV on the way to a relative parking orbit, followed by a closer range rendezvous in Phase V with low thrust profile.

Phase	Initial Separation	Navigation
I	>10,000 km	Abs Nav
II	$> 2\lambda_{\max}$	Abs Nav
III	$2\lambda_{\max}$ to 1 km	Abs Nav or Coarse Rel Nav
IV	1 km to 100 m	SEPS+BA or Model-based Pose Acquisition
V	100 m to 10 m	Model-based Pose Tracking

Table 2: Rendezvous phases and navigation approaches.

5.2 *Monocular-based Relative Navigation*

The approaches investigated in this paper on monocular-based pose determination are essentially relevant in Phases IV and V. If the target is unknown, the SEPS approach described in Section 2 could be adopted in Phase IV to safely reach the desired relative parking orbit. While in this orbit, the 3D model of the target reconstructed with SEPS could be refined via Bundle Adjustment (BA) as described in Section 2.3. Making use of such refined 3D model of the target geometry in Phase V and in the following phases, monocular pose estimation could be performed with the more precise and less computational expensive model-based pose tracking approach, as described in Section 3.

With respect to the classification of different mission phases in Table 2, we can see that the pose and pose rate estimation errors of SEPS (shown in Figs. 13–16) are small enough when the chaser is in Phase IV on its way to a relative parking orbit without risk of collision.

During Phase V, the accuracy achievable with a model based approach (Figs. 18 and 19) also appears to be high enough for performing an inspection.

The accuracy required in case of docking depends on the adopted docking mechanism, but in most of the cases it is still lower than the one reported in Figs. 18 and 19. For example using a magnetic docking for CubeSats, the accuracy required ranges from few centimeters to tens of centimeters for relative positioning and is of a few degrees for relative attitude estimation [31].

The accuracy in model-based pose estimation can be further improved by filtering the output of the PnP solver (relative position and attitude) with a translational and rotational dynamics model, as briefly mentioned in Section 3.1 and as done in [54].

6 Conclusion

In this paper, we investigated the main approaches for on-board monocular-based pose estimation of uncooperative orbiting targets. In particular, we described the algorithmic details of a filter proposed for Simultaneous 3D Reconstruction and Pose Estimation (*SEPS*) of unknown targets. Such a formulation relies on monocular images, a single beam LIDAR measurement and on the estimation of the angular velocity and of the CG of the target, both computed from the optical flow of the same acquired images.

A novel derivation of the optical flow was proposed that assumes that both world and camera are moving. A second independent filter that estimates CG and translational velocity based on optical flow measurements was also proposed. We also outlined a strategy based on SfM to refine the 3D model of the target reconstructed with the SEPS algorithm, or to directly reconstruct it from scratch on board and offline. In addition, we described a model-based approach that can be used when the target's geometry is known (or has been already reconstructed), that estimates only its pose and relies on the use of a simplified shape model of it.

These 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. The accuracy achieved in simulations with SEPS is less than a meter in relative positioning and less than 1 deg in attitude estimation, while with the model-based approach, is of a few centimeters and of less than 1 deg, respectively.

Such levels of accuracy satisfy the navigation requirements in different phases of a rendezvous mission. However experimental validation will be needed to confirm the performance obtained in simulation. Moreover, it is important to highlight that the achievable performance might worsen when processing real images of an orbiting target in unfavorable illumination conditions.

For future work, we plan to further focus on the described strategies and rigorously validate our own robust and original solutions to the problem of spacecraft pose estimation. These will be done using an advanced spacecraft simulator testbed available 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. Rahmani, and S. R. Alimo.

References

[1] D. Wright, "Space debris by the numbers," `https://www.esa.int/Our_Activities/Space_Safety/Space_Debris/Space_debris_by_the_numbers`, accessed on June 5, 2018.

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

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

[9] V. Capuano, A. Harvard, Y. Lin, S.-J. Chung, "DGNSS-Vision Integration for Robust and Accurate Spacecraft Navigation," in *ION GNSS+* 2019, Sept. 16-20, 2019, Miami, Florida.

[10] 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.

[11] 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.

[12] K. Meier, S.-J. Chung, and S. Hutchinson, "Visual-Inertial Curve Simultaneous Localization and Mapping: Creating a Sparse Structured World without Feature Points," J. Field Robotics, vol. 35, no. 4, pp. 516-544, 2018.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

[22] V. Capuano, K. Kim, J. Hu, A. Harvard, S.-J. Chung, "Monocular-Based Pose Determination of Uncooperative Known and Unknown Space Objects", 69th International Astronautical Congress (IAC), Bremen, Germany, October 2018.

[23] D. J. Fleet and Y. Weiss, "Optical flow estimation", Handbook of Mathematical Models in Computer Vision, 237-257, 2006.

[24] 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.

[25] 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.

[26] 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.

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

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

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

[30] J. R. Wertz and R. Bell, "Autonomous Rendezvous and Docking Technologies — Status and Prospects," *SPIE AeroSense Symposium*, Orlando, FL. April 23–25, 2003.

[31] C. Pirat, F. Ankersen, r. Walker, V. Gass, "Vision Based Navigation for Autonomous Cooperative Docking of CubeSats", Acta Astronautica, 146 (2018) 418-428

[32] Djurdjani and D. Laksono. "Open source stack for Structure from Motion 3D reconstruction: A geometric overview," *2016 6th International Annual Engineering Seminar (InAES)*, pp. 196-201, 2016.

[33] F. Baldini, A. Harvard, S. Chung, I. Nesnas, and S. Bhaskaran. "Autonomous Small Body Mapping and Spacecraft Navigation", *69th International Astronautical Congress*, 2018.

[34] H. Li ; R. Hartley. "Five-Point Motion Estimation Made Easy", *18th International Conference on Pattern Recognition*, 2006.

[35] 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.

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

[37] 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.

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

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

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

[41] 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.

[42] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, 2018, pp. 1-10. doi: 10.1109/ICOMET.2018.8346440

[43] "Blender," [Online]. Available at: `https://www.blender.org`.

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

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

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

[47] 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.

[48] 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.

[49] K. Yamanaka and F. Ankersen. "New state transition matrix for relative motion on an arbitrary elliptical orbit," *Journal of guidance, control, and dynamics*, vol. 25, pp.60-66, 2002.

[50] 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.

[51] 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.

[52] V. Capuano, R. Opromolla, G. Cuciniello, et al., "A Highly Integrated Navigation Unit for On-Orbit Servicing Missions", 69th International Astronautical Congress, Bremen, Germany 2018.

[53] S. Sharma, C. Beierle, S. D'Amico, "Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Convolutional Neural Networks", 2018 IEEE Aerospace Conference, Big Sky, MT.

[54] 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.

[55] 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.

[56] 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.

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

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

[59] V. Capuano, S. R. Alimo, A. Q. Ho, S.-J. Chung, "Robust Feature Extraction for On-board Monocular-based Spacecraft Pose Acquisition," AIAA SciTech Forum 2019, San Diego, CA, US.