Robust Navigation Solution for Vision-Based Autonomous Rendezvous

Anthea Comellini, Florent Maye, Vincent Dubanchet, Davide Casu, Emmanuel Zenou, Christine Espinosa

Robust Navigation Solution for Vision-Based Autonomous Rendezvous

Anthea Comellini
ISAE SUPAERO & Thales Alenia Space
Toulouse, France, 31400
anthea.comellini@isae-supaero.fr

Florent Mayé, Vincent Dubanchet, Davide Casu
Thales Alenia Space
Cannes, France, 06150
florent.mayé, vincent.dubanchet, davide.casu@thalesaleniaspace.com

Emmanuel Zenou
ISAE SUPAERO
Toulouse, France, 31400
emmanuel.zenou@isae-supaero.fr

Christine Espinosa
ISAE SUPAERO & Institut Clement Ader
Toulouse, France, 31400
christine.epinosa@isae-supaero.fr

Abstract— **This paper proposes Thales Alenia Space vision-based navigation solution for close proximity operations in autonomous space rendezvous with non-cooperative targets. The proposed solution covers all the phases of the navigation. First, a neural network robustly extracts the target silhouette from complex background. Then, the binary silhouette is used to retrieve the initial relative pose using a detection algorithm. We propose an innovative approach to retrieve the object's pose using a precomputed set of invariants and geometric moments. The observation is extended over a set of consecutive frames in order to allow the rejection of outlying measurements and to obtain a robust pose initialization. Once an initial estimate of the pose is acquired, a recursive tracking algorithm based on the extraction and matching of the observed silhouette contours with the 3D geometric model of the target is initialized. The detection algorithm is run in parallel to the tracker in order to correct the tracking in case of diverging measurements. The measurements are then integrated into a dynamic filter, increasing the robustness of target pose estimation, allowing the estimation of target translational velocity and rotation rate, and implementing a computationally efficient delay management technique that allows merging delayed and infrequent measurements. The overall Navigation solution has a low computational load, which makes it compatible with space-qualified microprocessors. The solution is tested and validated in different close proximity scenarios using synthetic images generated with Thales Alenia Space rendering engine SpiCam.**

TABLE OF CONTENTS

1. INTRODUCTION

Autonomous rendezvous (RDV) is a key capability to answer main challenges in space engineering, such as Active Debris Removal (ADR) and On-Orbit-Servicing (OOS). ADR aims at removing the space debris, in low-Earth-orbit (LEO) protected region, that are more likely to lead to future collision and feed the Kessler syndrome [1], thus increasing the risk for operative spacecraft (S/C). OOS includes inspection, maintenance, repair, assembly, refuelling and life extension services to orbiting S/C or structures [2]. The two actors in a rendezvous are referred to as the *chaser* and the *target*. The chaser vehicle is a spacecraft which has both attitude and translational control capability, and it actively navigates to the target vehicle in the rendezvous process [3]. Rendezvous scenarios are many and varied, and depend mainly on the nature of the target. In this paper, we propose to classify targets as *cooperative* or *non-cooperative* and *prepared* or *non-prepared*. According to the definition provided by the Consortium for Execution of Rendezvous and Servicing Operations (CONFERS), when the target does not assist the chaser in acquisition, track and rendezvous operations, it is referred to as non-cooperative [4], meaning that the chaser has to estimate autonomously on board the target state. The notion of cooperative or non-cooperative therefore relates to target's behaviour. On the other hand, the notion of prepared or non-prepared concerns target's design. It indicates whether the satellite was originally conceived to participate in a RDV, and therefore whether it is supplied or not with equipment to help both the tracking and the servicing. Because space debris objects were not conceived to participate in a RDV and are now inoperative, ADR operations will target non-cooperative and non-prepared S/C. On the other hand, OOS ideally targets cooperative and prepared S/C. However, a generation of prepared satellite is not yet in orbit, and a cooperative target can become non-cooperative in case of a system failure. A RDV with a non-cooperative non-prepared target remains the most challenging scenario for chaser's GNC (Guidance, Navigation & Control) system. For this reason, the proposed study is focused on non-cooperative and non-prepared targets. Autonomous rendezvous navigation algorithms require accurate, up-to-date measurements of the relative pose (i.e., position and attitude) of the target. Inexpensive camera sensors have a small form factor -so that they are easily integrated to the S/C without affecting its design- and a low power budget (unlike LIDARs [5], [6],[7]). For this reason, the coupling of camera sensors with image processing (IP) and computer vision (CV) algorithms can provide a cost effective solution. This paper proposes a vision based navigation algorithm that allows the 6-degrees-of-freedom (DOF) pose estimation of a non-cooperative non-prepared target using a single visible monocular camera. Indeed, the use of monocular vision provides advantages with respect to stereo-camera configurations, because single cameras have a lower complexity and a much larger operational range which is not limited by the size of the satellite platform.

Related Work

Tracking methods can be classified as *frame-by-frame tracking* or *tracking-by-detection* algorithms [8]. In frame-by-frame tracking, the object's pose retrieved from a frame is

used as a prior for a local search of the pose in the following frame. This recursive approach makes image features identification and matching relatively easy [8]. However, these methods require initialization and can diverge in the presence of local minima. For this reason, a recursive tracking algorithm must be complemented with a tracking-by-detection algorithm to enable initial pose acquisition and fault detection. In a tracking-by-detection algorithm, the pose is retrieved by exploiting a-priori information on the geometry and appearance of the tracked object, but with no knowledge of the pose at previous instants.

Frame-by-frame tracking can be classified into *non-model-based* and *model-based* techniques [5]. Non-model-based techniques do not assume any *a priori* knowledge of the tracked object's shape, texture and other visual attributes. These methods rely on visual features (e.g., SURF, SIFT, ORB) which are ideally recognisable and distinguishable from one image to another, regardless of the pose of the camera, the lighting or the spectral conditions in the image. For space applications, methods that depend upon visual features detection are not robust since phenomena such as occlusions, harsh lighting, and reflective materials can make reliable detection and correspondence impossible [9]. On the other hand, model-based techniques take advantage of *a priori* knowledge of the object whose pose is to be estimated. This knowledge can be in the form of "fiducials" (or "markers"), or in the form of a 3D geometrical description of the object. Fiducials are features expressly designed in such a way that they are easily detected and identified with an *ad hoc* method [8] (e.g., ArUco, ARToolKit, AprilTags). The use of fiducials implies that the target is originally designed to be easily trackable, and therefore it is prepared. On the other hand, in a rendezvous with a non-prepared target, the pose estimation algorithm can still rely on the knowledge of the 3D geometric model of the spacecraft. 3D model-based tracking algorithms are therefore the best candidates for this study.

On the other side, monocular pose estimation by detection is by definition model-based, and it can be performed using geometric methods or by template matching. In geometric methods, the observed 2D features in the input image are matched with a database of features computed offline. Then the pose is retrieved solving the Perspective-n-Point problem [10]. Geometric approaches based on local feature are proposed for S/C pose acquisition in [11], [12], [13]. However, methods relying on features detection lack robustness in the adverse illumination conditions encountered by spaceborne systems. In template matching approaches, a *training* set of views of the object is acquired offline to generate a database of templates that are compared at run-time with the input image. Classical template matching approaches compare the pixels intensities of the templates and the input image according to similarity measures or alignment functions [14]. However, these methods are computationally expensive and lack robustness to illumination changes [14]. Other approaches rely on templates of local features, such as image gradient orientations (e.g., [15], [16]) or binary templates of the extracted edges. An edge-based template matching approach relying on a similarity measure derived from *Chamfer Matching* [17] and on an unsupervised clustering technique is proposed for S/C pose estimation in [14]. The template matching approaches described are computationally complex because of the evaluation of a large number of possible pose hypotheses and real-time may not be achievable on space qualified processors. They might be appropriate for pose initialization, but cannot be used as a backup algorithm to help detecting divergence in the recursive tracking algorithm. For this reason, recent work has focused on using Convolutional Neural Networks (CNNs) for pose estimation directly from greyscale images in a end-to-end fashion [18], [19]. However, the reported accuracy is currently still lower than that of geometric methods [11]. Hybrid approaches, where CNNs are used to extract keypoints (i.e., local features) from the image while the PnP solver is used to compute the pose, have recently shown very good accuracy with synthetic images [20]. However, this still raises the issue of relying on local features detection.

The approach proposed in this paper is to couple an edge-based recursive tracking algorithm with a detection algorithm relying on template matching with global features instead of local ones. The rest of the paper is structured as follows. In Sec.2 the background subtraction algorithm based on convolutional neural network (CNN) is detailed. In Sec.3 we detail the pose estimation by detection algorithm based on Zernike invariants and geometric moments. In Sec.4 the frame-by-frame tracking algorithm and its coupling with the dynamic navigation filter are presented. Sec.5 presents the integrated navigation solution designed for the particular camera sensor and target model selected for this study. In Sec.6 the navigation solution is tested in different realistic scenarios and the computational load of the navigation function is discussed. In Sec.7 the conclusion are drawn and perspective for future studies are given.

Problem Statement

Figure 2 shows a schematic representation of the pose estimation problem. Let P^{tg} be the coordinates of a point P expressed in the target reference frame (RF) tg, which is centered at the target center of mass (COM). The coordinates of P, expressed in the camera RF cam, are:

$$P^{cam} = [x^{cam},\, y^{cam},\, z^{cam}]^T = R_{cam-tg}\, P^{tg} + t^{cam}_{O_{cam}-O_{tg}}\,. \tag{1}$$

The rotation matrix R_{cam-tg} is constructed using the classical Euler representation, with $\varphi \in]-\pi, \pi]$ the *roll* angle, $\vartheta \in]-\pi/2, \pi/2]$ the *pitch* angle and $\psi \in]-\pi, \pi]$ the *yaw* angle. In this paper we will also use quaternions to describe rotations, according to the Hamilton convention [21] (i.e., quaternions are written $q = q_0 + q_1 i + q_2 j + q_3 k$, where q0 denotes the scalar part, q^* denotes the quaternion conjugate $q^* = q_0 - (q_1 i + q_2 j + q_3 k)$ and $\otimes$ denotes the quaternion product):

$$P^{cam} = [x^{cam},\, y^{cam},\, z^{cam}]^T = q_{cam-tg} \otimes P^{tg} \otimes q^*_{cam-tg} + t^{cam}_{O_{cam}-O_{tg}} \tag{2}$$

With the notation $\mathrm{q}([\varphi,\, \vartheta, \psi])$, we indicate the quaternion corresponding to the Euler angles $[\varphi,\, \vartheta, \psi]$. The vector $t^{cam}_{O_{cam}-O_{tg}}$ corresponds to the translation from the origin O_{cam} of the camera RF to the origin O_{tg} of the target RF, expressed in the camera RF. For simplicity, we will refer to $t^{cam}_{O_{cam}-O_{tg}}$ as $t^{cam}_{cam-tg} = [x^{cam}_{cam-tg},\, y^{cam}_{cam-tg},\, z^{cam}_{cam-tg}]$, the translation vector from the camera to the target COM. According to the classical pinhole camera model, the point P^{cam} is projected into the image plane as follows:

$$p = [u_p,\, v_p] = \left[\frac{x^{cam}}{z^{cam}} f + C_x,\, \frac{y^{cam}}{z^{cam}} f + C_y\right]. \tag{3}$$

where f denotes the focal length of the camera and (C_x, C_y) denotes the principal point of the image (see Fig.2).

For this particular study, a camera having a field of view (FOV) of 30 deg and a sensor size of 1024×1024 has been selected (i.e., on Fig. 2 we have $C_x = C_y = 512$ and $f = C_x / \tan(\mathrm{FOV}/2) = 1911$.). The geometry of the target

S/C used in this study is inspired from the structure of the Iridium-NEXT satellites and shown in Fig. 2. The target has a size of $4 \times 3 \times 1$ meters and an inertia matrix equal to $diag([700; 500; 1000])$ kg m^2. The target dimensions, translated in pixels, correspond to a projected size of $76 \times 53 \times 19$ pixels at 100 m, $191 \times 143 \times 48$ pixels at 40 m, $255 \times 191 \times 64$ pixels at 30 m, $382 \times 287 \times 96$ pixels at 20 m, and $764 \times 573 \times 191$ pixels at 10 m. The main structure of the S/C, composed of the central body and the lateral solar arrays, has two symmetry planes, (O_{tg}, x_{tg}, z_{tg}) and (O_{tg}, y_{tg}, z_{tg}), see Fig. 2. Some elements on the central body such as antennas and a docking fixture break the symmetry but are relatively small and visible only for a restricted range of attitudes (i.e., $|\varphi| \in [80, 100]$ deg). For all the other attitudes, to each set of Euler angles $[\varphi, \vartheta, \psi]$, corresponds a symmetric solution $[-\varphi, -\vartheta, \psi - \pi]$. The presence of two solution branches is an issue that is discussed in Sec.5. All the images used in this study are generated with Thales Alenia Space rendering engine SpiCam.

2. BACKGROUND SUBTRACTION

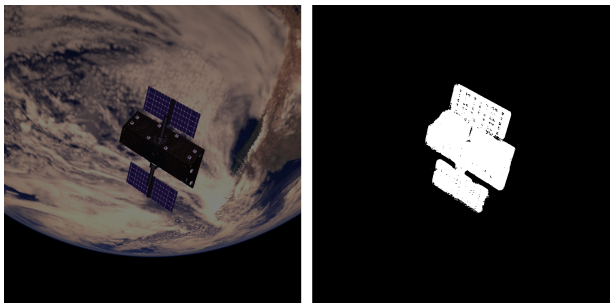


Figure 1. Example of input and output images of the segmentation algorithm

In order to robustly apply the pose estimation algorithms, it is necessary to isolate the target spacecraft from any possible complex background present on the captured image. In this section we will present the use of neural networks applied to space image segmentation. Recently major advances have been achieved in the domain of computer vision thanks to neural networks, especially convolutional neural networks. This allows reaching in many applications a precision that was complicated to achieve until now, and image segmentation strongly benefits from these new techniques. The use of neural networks for image segmentation has already been explored on classic cases, with famous network architectures like Mask-R CNN [22], SegNet [23], or U-NET [24]. In the case considered in this study, the need is to precisely extract the mask of a satellite in space environment, with complex llumination conditions and backgrounds such as the Earth. The solution must be embeddable, meaning that the network must have a low computational load. Fully convolutional neural networks with transposed convolutions layers applied to a classical encoder-decoder architecture have been selected for this study. The convolution layers offer a higher level of representation at each step, while the encoder architecture decreases the feature space to force the categorization of pixel into relevant classes. Finally, the transpose convolution layers reconstruct the segmented image. This approach seems to be well fitted to our case, with acceptable resource consumption with a network composed by 6 convolutional layers and 13500 parameters to be trained. A meta-optimisation operation was performed to find the best fitted meta-parameter of the network (e.g., depth of the encoding/decoding, number of convolution kernels). The output of the neural network is a grey-scale image where the intensity of the pixel corresponds to the probability the pixel has of belonging to the spacecraft silhouette. The segmentation has been tested with two independent dataset generated with two different synthetic image generators, one of them being highly representative. The network was previously trained on 9000 labeled images randomly generated by the image generator, and then tested on an independent dataset generated by the same generator on a given test case. The network was able to extract the satellite in almost every case, even when it was barely visible due to the distance or the presence of shadows. The mask borders are precise with a low noise of only few pixels. Some outputs showed the issue of misdetection of some spacecraft parts, either on the inside of the mask -especially for very short range detections-, or on thin parts like the solar array connections. The generated mask can present some false detection with secondary blobs, but these blobs are always smaller and with a lower pixel intensity than the satellite blob. A simple post processing algorithm is applied to extract the main blob: a 20% downscaled image exploration is performed to detect the blob, which is then extracted using a watershed algorithm. The final output is a binary image, as shown in Fig.1. For some images, this post-processing resulted in a separation of the solar arrays from the satellite core. This issue has been solved applying morphological filtering on the full scale image before the extraction of the main blob. This post-processing technique can also help filling missing S/C parts that are inside the detected silhouette. With the filtering, the algorithm has a complexity of $\mathcal{O}(n)$, with n the size of the image. While no precise metric was measured at this stage due to the difficulty to find a relevant one, it was assessed that only the 0.2% of the images was presenting defaults making them non exploitable for pose estimation. The impact of the proposed segmentation method on the pose estimation algorithms will be shown in Sec.6, where the full navigation solution is tested.

3. DETECTION ALGORITHM

We detail in this section a method allowing to estimate the pose of a known object by *detection*, i.e., it requires no prior information about the pose of the observed object, making it suitable for initial pose acquisition and the monitoring of faults in other on-board estimators. The approach allows to fully retrieve the object's pose using its observed binary silhouette and a pre-computed set of global features (i.e., features that are computed using the whole image). Global features provide a low-dimensional representation of the target's silhouette on a binary image. The interest in using global features such as Fourier descriptors [25], [26], [27], [28] or image moments [29], [26], [28], [30] is that these features can be made invariant to translation, to scaling, and most importantly to rotation. If a bi-dimensional shape is described by such invariant features, the value of the features will not depend on the position of the shape centroid (translation invariance), on the shape dimension (scaling invariance), and on the rotation of the shape in the image plane (rotation invariance). The shape of the projected silhouette does depend both on the relative attitude and the relative position of the observed object, and the contribution of the translation is coupled with the contribution of the attitude. However, under the hypothesis of weak-perspective model, those effects can be decoupled. The weak perspective model can be assumed when the depth of the object along the line of sight is small compared to the distance from the camera [14], and the FOV is small or the object is placed near to the center

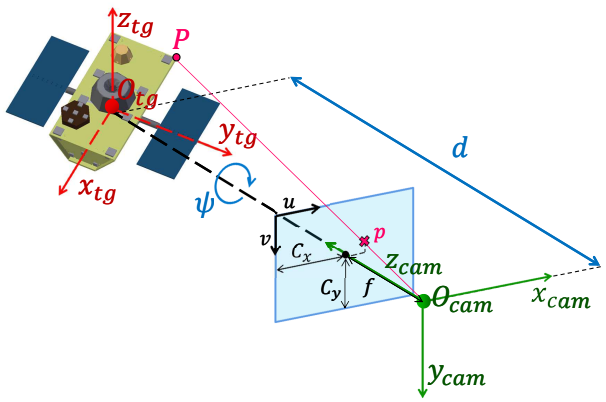


Figure 2. Schematic representation of the pose estimation problem using a monocular image. In this particular case, the camera optical axis is pointed toward the target COM, as done for the generation of the images of the database used in the detection algorithm detailed in Sec.3

of the FOV (i.e., $x^{cam}_{cam-tg} \sim 0$ and $y^{cam}_{cam-tg} \sim 0$). These are indeed conditions that are met during the pose acquisition phase. In the weak perspective model it is assumed that all points on a 3D object are at the same distance z from the camera without significant errors in the projection (compared to the full pinhole perspective model). Under this hypotheses the distance z^{cam}_{cam-tg} affects only the scale of the projected silhouette, while the components x^{cam}_{cam-tg} and y^{cam}_{cam-tg} affect only the position of the silhouette centroid. Thus, if global invariant features are used do describe the S/C silhouette at a given pose, the value of the features will depend only on the roll and pitch angles φ and ϑ. In fact, ψ only affects the rotation of the projected shape in the plane image, as visible in Fig.2. The principle of the pose estimation algorithm that we propose is the following. During an off-line process, a set of synthetic views of the target, referred to as *training* images in the following, is generated for a sufficiently large number of views (referred to as N_w) selecting randomly discrete values of the pairs $(\varphi, \vartheta) \in]-\pi, \pi] \times]-\pi/2, \pi/2]$. When generating the database, the camera is pointing to the target COM (i.e., $x^{cam}_{cam-tg} = y^{cam}_{cam-tg} = 0$). The yaw angle ψ and the camera-target distance $d = z^{cam}_{cam-tg}$ remain fixed, with $\psi = \psi_{train}$ set to zero. The value of $d = d_{train}$ should be chosen close to the value of the relative range at which the pose must be acquired, in order to maximize the performance of the algorithm [31]. Once the images corresponding to the N_w viewpoints are generated, the global invariant descriptors are computed for each view. For this study, we have selected Zernike moment (ZM) invariants [32], which provide better performance than other invariants such as Complex moments or Fourier descriptors [31]. The methodology used to compute the rotation invariants is detailed in [31] and [32]. The size of the resulting database is $N_w \times N_f$, with N_f the dimension of the feature vector. At run-time, when the camera acquires a new image of the target, the algorithm computes the descriptor vector associated to the resulting view and finds in the database the pair $(\varphi_{meas}, \vartheta_{meas})$ with the closest descriptor vector (minimizing the Euclidean distance), i.e., performs a nearest neighbor search. The remaining degrees of freedom, i.e., the yaw angle ψ and the relative camera-target position t^{cam}_{cam-tg}, are estimated using geometric moments (GMs), according to the procedure described in the following. From a given image, the yaw angle ψ of R_{cam-tg} can be retrieved using the second order central moments μ_{ij} $(i, j \leq 2, i + j = 2)$. First, the *inclination* $\tilde{\psi}_0 \in]-\pi/2, \pi/2]$ of the major axis of inertia of the silhouette on the image is given by $\tilde{\psi}_0 = 1/2 \operatorname{atan}[2\mu_{11}/(\mu_{20} - \mu_{02})]$ [32]. However, to recover the full in-plane angle of rotation $\psi_0 \in]-\pi, \pi]$ of the silhouette, we still need to determine a specific direction along the axis of inertia, i.e., distinguish between $\psi_0 = \tilde{\psi}_0$ or $\psi_0 = \tilde{\psi}_0 - \pi$. For this, the third order central GMs can be used, since they change sign under a rotation of π [33]. The sign of moment μ'_{03}, computed on the image frame rotated by an angle $\tilde{\psi}_0$ defines the in-plane rotation ψ_0:

$$\psi_0 = \begin{cases} \tilde{\psi}_0 & \text{if } \mu'_{30} > 0\,, \\ \tilde{\psi}_0 - \pi & \text{if } \mu'_{30} < 0\,. \end{cases} \tag{4}$$

The ambiguity can be resolved only if the silhouette is not rotationally symmetric. In the case of a silhouette having an N-fold rotation symmetry (N-FRS) (i.e., if it repeats itself after rotation around its centroid by $2\pi j/N$, for all $j = 1, ..., N$), N solutions are possible. In such case, only an observation of the target pose on a set of consecutive frames

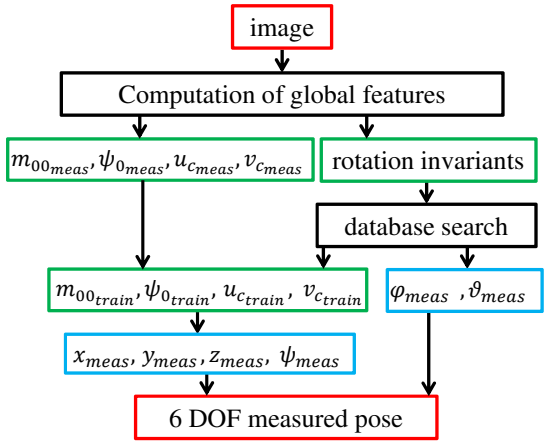


Figure 3. Structure of the global feature based detection algorithm.

could help solve the ambiguity, as done in Sec.5. Once $\psi_{0_{test}}$ is determined for the current image, we can deduce the measured yaw angle ψ_{meas} of the camera frame by

$$\psi_{meas} = \psi_{0_{test}} + (\psi_{train} - \psi_{0_{train}}). \tag{5}$$

where $\psi_{0_{train}}$ is the major axis of inertia (precumputed and stored in the database) of the training view that best matches the current view, and $\psi_{train} = 0$ by construction. Finally, we obtain a measure of the relative camera-target position using zeroth and first order moments. The zeroth order geometric moment m_{00} is the area of the silhouette, while $m_{10}/m_{00} = u_c$ and $m_{01}/m_{00} = v_c$ are the x and y coordinates of the silhouette's centroid in the image frame. The values of $m_{00_{train}}$, $u_{c_{train}}$, and $v_{c_{train}}$ must be precomputed for each view and stored in the database. The relative distance camera-target can be retrieved according to:

$$d_{meas} = \sqrt{m_{00_{train}}/m_{00_{meas}}} \cdot d_{train} \tag{6}$$

where $m_{00_{train}}$ is the zeroth order moment for the best matching training view. Recalling (3) and exploiting the fact that $d = \sqrt{(x^{cam}_{cam-tg})^2 + (y^{cam}_{cam-tg})^2 + (z^{cam}_{cam-tg})^2} \sim$

z^{cam}_{cam-tg}, the components x^{cam}_{cam-tg} and y^{cam}_{cam-tg} can be approximated by:

$$\begin{cases} x^{cam}_{cam-tg} &= \dfrac{d}{f}\left(u_{c_{meas}} - C_x\right) \\ y^{cam}_{cam-tg} &= \dfrac{d}{f}\left(v_{c_{meas}} - C_y\right) \end{cases} \tag{7}$$

where ($u_{c_{meas}}$, $v_{c_{meas}}$) are the coordinates of the observed silhouette centroid. The identities (7) can be used to ensure camera pointing even before the target's full pose has been acquired. Fig. 3 summarizes the structure of pose estimation by detection algorithm.

4. FRAME-BY-FRAME TRACKING ALGORITHM

Within the many 3D model-based pose estimation techniques for monocular vision, the more suitable for space applications are the ones relying on edge extraction and tracking, since edges are strong features easily detectable in correspondence of high image gradients. These methods are computationally efficient and naturally stable to lighting changes even for specular materials [8]. A well-known model-based algorithm relying on edge tracking is RAPiD (Real-time Attitude and Position Determination) algorithm. RAPiD was first theorised in 1990 by C. Harris and C. Stennett in [34] and it was one of the first monocular 3D tracker to successfully run in real-time due to its low computational complexity. At instant t, the 3D *a priori* model is projected in the image frame using the pose parameters estimated at instant $t-1$. Visible edges are selected and sampled in order to determine a set of "control points" that will be used in the optimisation process. At the same time, edges are extracted on the greyscale image captured at the instant t, resulting in a binary image. Then the control points are associated to the observed points on the image. The matching is carried out by searching along the vector normal to the edge that contains the control point. This mono-directional search reduces the matching search-space from bi-dimensional to one-dimensional, thus allowing fast tracking. To compute the pose correction, RAPiD method relies on the fact that, at first order, small changes in the object pose will cause a displacement of the control points in the image frame which is linear in the pose parameters. This linearity enables to determine the variation of pose through the solution of a simple linear least square problem. However, for high relative motion of the target in camera frame, or for low acquisition rate, the displacement of the observed points in the image frame may be to high to enable linearization. For this reason, we propose a non-linear version of RAPID method. The application of a non-linear version of RAPiD in space rendezvous has been assessed also in [35], [36], [37], where a graphic process units (GPU) is used in order to render not only geometrical edges but also texture discontinuities of the *a priori* model. Other RAPID-like methods have been proposed in Refs.[38], [39]. All the cited works integrate, to the pose estimation algorithm, a linear Kalman filter (KF) which propagates a simple kinematic model. The method that we propose does not need any GPU augmentation in order to be compatible with typical space processing capabilities. Moreover, it is coupled with a dynamic navigation filter enabling the estimation of the full target rotational and translational state exploiting only relative pose measurement. In fact, in the case of high rotation rates typical of a tumbling object, a simple kinematic filter does not allow to estimate the rotation rate of the target, which needs to be known to perform some RDV operations requiring the synchronization of chaser motion with target motion. The proposed filter implements a method allowing to merge multi-rate and delayed measurements, being particularly suitable for the RDV application, as tracking algorithms have relatively high latency.

Contours-based tracker

The first step of the algorithm is the extraction of the edges from the image captured at instant t using the Canny edge detector. However applying the Canny edge extractor on the greyscale images leads to the major drawback of detecting not only geometrical edges, but also texture discontinuities. Moreover, the direction of the light could make invisible some geometrical edge due to the particular reflective texture of the MLI (Multi Layer Insulation), or generate projected shadows. All these conditions can lead to possible false matching of the control points with a non-geometrical edge. As this work does not consider the option of using GPU acceleration to render also texture discontinuities and shadows, the most robust solution to reduce the number of false matching is to rely only on the external contour of the target, using therefore the silhouette of the target extracted after the segmentation process (Sec.2). In the meantime, a *masking algorithm* is run in order to select the control points. The a priori 3D model is projected into the image frame using the estimated pose at the instant $t-1$ and the camera intrinsic calibration matrix. Then each surface is meshed monodimensionally along its perimeter obtaining a set of possible control points. All the points whose projection in the plane $u-v$ is inside the projected perimeter of any other surface are discarded, since they certainly do not belong to the external projected perimeter of the target.
At the end of the iteration, the output of the masking algorithm is a set of 2D visible points (i.e., the control points) belonging to the edges. In order to enable the matching procedure, for each control point the two-component normal vector (orthogonal to the edge containing the point) is computed. For each control point, the corresponding 3D coordinates in target frame are stored because they will be used in the optimisation process. Finally, the matching of a control point with an observed point in the binary contour image is found by moving from the projected control point along the projected normal vector -in both directions- until a value equal to 1 is found. In order to reduce the number of false matching, we have decided to introduce the value d_{toll}, a maximal acceptable distance from the control point to the corresponding matched point. If no match is found at a distance lower than d_{toll}, the control point is discarded. This value of tolerance depends on many factors such as the time span between two time step, the relative rotational and translational target rate, the relative distance camera-target, and the image size.
Once that the control points have been matched to the measured points, the cost function can be built. Given a fitting non linear function $\hat{y}(\rho)$ of a vector of n parameters ρ, and a set of m data points y_i, the parameters can be estimated minimising the sum of the weighted squares of the errors between the measured data y_i and the fitting function $\hat{y}_i(\rho)$. The resulting scalar cost function is:

$$\chi^2(\rho) \quad = \sum_i^m \left(y_i - \hat{y}_i(\rho)\right)^2 = (Y - \hat{Y}(\rho))^T W (Y - \hat{Y}(\rho)) \tag{8}$$

where the weight matrix W is equal to diag$(1/\sigma_i^2)$, with σ_i the standard deviation of the error associated to measurement y_i. The set of measurement data Y will be a vector composed by the projections of $[u_i, v_i]^T$ (i.e., the measured coordinates

in image frame of the matched point corresponding to P_i^{tg}, the 3D point in target RF) along the projected normal $\mathbf{n_i} = [n_{\hat{u}_i}, n_{\hat{v}_i}]^T$.

$$y_i = n_{\hat{u}_i} u_i + n_{\hat{v}_i} v_i, \quad Y = [y_1, y_2, ..., y_i, ..., y_m]^T \tag{9}$$

The weight matrix W is set to be equal to the identity matrix of size $m \times m$, since there is no way to know the error related to each matched image point $[u_i,\ v_i]$.
In the pose parameter ρ, the rotations will be described using quaternions (i.e., Eq.(2)), as quaternions have the lowest dimensionality possible for a globally non-singular representation of the rotational group $SO(3)$ [40]. The navigation filter described in the following paragraph also relies on quaternions for the attitude estimation problem. The set of parameters to be estimated will be $\rho = [t_x, t_y, t_z, q_0, q_1, q_2, q_3]$, which has size $n = 7$, with $[t_x, t_y, t_z] = [x_{cam-tg}^{cam},\ y_{cam-tg}^{cam},\ z_{cam-tg}^{cam}]$ and $q_{cam-tg} = [q_0,\ q_1\ , q_2\ , q_3]$. For each matched control point P_i^{tg} in target RF, the corresponding point $p_i = [\hat{u}_i(P_i^{tg}, \rho),\ \hat{v}_i(P_i^{tg}, \rho)]$ in the image frame is given by (2) and (3). Then, the non-linear function $\hat{Y}(\rho) = [\hat{y}_1(\rho), \hat{y}_2(\rho), ..., \hat{y}_i(\rho), ..., \hat{y}_m(\rho)]^T$ can be built, with $\hat{y}_i(\rho) = n_{u_i}\hat{u}_i + n_{v_i}\hat{v}_i$. The non-linear optimisation is solved using the Levenberg-Marquardt (LM) algorithm. The Jacobian of the nonlinear functions $\hat{Y}(\rho)$ has to be derived in order to compute at each iteration the update of parameter ρ. The computation of the elements of the Jacobian is provided in [41]. The obtained Jacobian is used at each iteration to analytically compute, according to the version of the LM algorithm proposed in [42], the direction of the parameter increment, starting from the initial guess ρ_0 which is taken to be equal to the estimated pose at instant $t-1$. In this optimisation process, some approximations are introduced: at each update of the pose parameter within the LM algorithm, a new set of control points and normal vectors should be computed by the masking algorithm, and new matches with the binary image should be found. Nevertheless, such a process would dramatically increase the computational load of the algorithm. For this reason, the set of control points, matched points and normal vectors will be kept constant for all the loops within each LM run. The final parameter ρ_{est} estimated by the LM algorithm will be the *measured pose* of the tracking algorithm.
One of the main problem related to the IP-CV algorithms is to have a good characterization of the measurement noise covariance matrix R, which is needed to integrate the measurements into the Kalman filter. The covariance of the estimated parameter ρ_{est}, and therefore of the measured pose, can be computed from the Jacobian and the weight matrix, according to : $R = [J^T W J]^{-1}$. However, the value of W is not known. For these reason, the value of the χ^2 evaluated at the estimated parameter ρ_{est} divided by the degree of freedom of the problem $\nu = m - n + 1$ is used as an indicator of the level of confidence of the measurement. This value is also used as one of the convergence criteria in the LM algoritm [42].

$$r = \frac{\chi^2(\rho_{est})}{\nu} \tag{10}$$

The value of r is a sort of averaged squared reprojection error, and it will be referred to as *residual* in the following sections. For a given target geometry, camera parameters and relative camera-target distance, the residual can be correlated to the noise covariance matrix of the associated measurements (i.e., a very low residual indicates that the reprojection error is minimal and therefore that the measurement is reliable). This correlation is obtained experimentally during offline calibration. During these test, a value of r_{max} (function of the range) must be set in order to determine whether a measurement is acceptable or not. In Sec.5 we will provide more details about the way this eventuality is managed in the navigation function.

Integration of the measurements in the navigation filter

The tracking algorithm discussed in the previous paragraph provides a measurement of the relative pose camera-target, but no measurement of the relative translational velocity and rotational rate. As anticipated, certain close proximity operations require the knowledge of the complete relative state of the target. The coupling of the IP-CV algorithm with a dynamic KF enables the estimation of both translational and rotational velocity of the target. Moreover, the measurements computed by the IP-CV tracking algorithm can be affected by a consistent delay, due to the computational load associated to operation such as the image segmentation, the 3D model projection, and the non-linear optimisation. There will be therefore a delay between the time of acquisition (i.e., the capture of the image by the camera), and the time in which the measurement becomes available. In addition, due to the high computational load of the IP-CV algorithm, measurements could be available at a lower rate with respect to the navigation filter run frequency. For this reason, the filter must implement an appropriate technique to merge delayed and infrequent measurements. The complete formalisation of the method and its application to the space RDV navigation problem are discussed in [43], but we provide here a brief description. The translational motion is modelled according to the well known Clohessy-Wiltshire-Hill's (CWH) equations [44], a system of linear differential equations that describes the relative motion of the chaser with respect to the target, expressed in the target Local Orbital Frame [LOF, with x axis along the radial Earth-target, z axis along target orbit angular momentum, and y axis completing the right-handed trihedron]. The rotational dynamics is modelled according to the non-linear prediction model:

$$\begin{cases} \dot{q}_{i-tg} &= \frac{1}{2} q_{i-tg} \otimes \begin{bmatrix} 0 \\ \omega_{i-tg}^{tg} \end{bmatrix} \\ \dot{\omega}_{i-tg}^{tg} &= -I_{tg}^{-1} \left(\omega_{i-tg}^{tg} \times I_{tg}\, \omega_{i-tg}^{tg} \right) \end{cases} \tag{11}$$

where q_{i-tg} is the attitude quaternion from inertial (i.e., i) to target RF, ω_{i-tg}^{tg} is target rotation rate with respect to the inertial RF expressed in target RF and I_{tg} is the inertia matrix of the target at its COM.
In order to integrate delayed measurements, Larsen's method is implemented [45]. This method relies on the computation, throughout the delay period, of a correction term to add to the filter estimate when the delayed measurement becomes available. It requires only two matrix multiplications at each time step, as well as the storage of the predicted state and error covariance matrix relative to the time step in which the measurement was acquired (i.e., the camera acquisition time in the case considered for this study). Moreover, Larsen's method allows to merge multi-rate measurements, enabling the use of multiple sensors and tracking algorithms. Larsen's method is sub-optimal for non-linear systems and in the presence of multi-rate measurements, but has a very low computational load compared to optimal methods (e.g., Filter Recalculation method, [46], [47]). The analysis in [43] has proven that, for the RDV problem, Larsen's method is the best trade-off between optimality and computational load, with a performance comparable to the one of the optimal method.

5. INTEGRATED SOLUTION

During RDV operations, the knowledge of the full 6-DOF pose is required only during the final approach, starting from a relative distance around $50 - 40$ m. Above this distance, the navigation is in a *range only* mode, where only the target relative position is estimated. The detection algorithm described in Sec.3 allows, with Eqs.(6) and (7), measuring the relative position even at those distances where the attitude is not correctly retrieved (e.g., between 100 m and 40 m using the camera considered in this study). The measured position is then filtered in the translational dynamic filter modelled according to the CWH equations. When the estimated distance has reached the value selected to begin the 6-DOF pose estimation, the pose acquisition mode is activated. During this phase the chaser is on a hold point and acquires the 6-DOF pose of the target. Once that the pose is initialized, the 6-DOF tracking can start. We detail in the following paragraphs each one of these navigation modes.

6-DOF pose acquisition

The detection method described in Sec.3 with Zernike invariants up to the 9^{th} order is employed in the pose acquisition function. For the particular target geometry chosen for this study, due to the impossibility to distinguish between two symmetric attitudes (see Sec.1), the database is build using only nonnegative values of φ. A total of $N_w = 5000$ samples is used. The presence of small details on the main body of the target breaks this symmetry and is observable for values of $|\varphi| \in [80,\ 100]$ deg. Thus, the database is extended adding 500 random samples satisfying the conditions $\varphi \in [-100,\ -80]$ deg. Using the camera considered for this study, the method allows reliable pose acquisition from a distance of 40 m using the ZM-based detection algorithm developed in Sec.3. Using ZM up to the 9^{th} order, at a distance of 40 m, the method allows the detection of the observed pose with an accuracy (i.e., percentage of detection having an angular error lower than 20 deg) of 95% and a mean error of 2 deg. The method described in Sec.3 needs to be associated to some post-processing function, in order to have a reliable initialization of the pose and to robustly discard the 5% of the outlying measurements. For outlying measurement (or outlier) we mean a measurement which is not attributable to any of the two symmetric solutions branches. The following logic as been defined.

To initialize the pose, three conditions have to be satisfied. Initially, we search for a time-window of a given length (l_{tw}) along which the difference between each consecutive attitude measurements is below a given threshold δ_{t1}. The fulfillment of such a condition ensures, in the observed time-window, that the measurements are "continuous" and therefore compatible with the rigid body dynamics, and that no outliers are present (the probability that in the considered time-window there is a series of outlying measurements close one to the other exists but remains low). At the current time step $t = i$, the new acquired measurement $[\varphi_i,\ \vartheta_i,\ \psi_i]$ is compared with the measurement $[\varphi_{i-1},\ \vartheta_{i-1},\ \psi_{i-1}]$ acquired at the previous instant $t = i - 1$. Two index indicating the angular distance between the two measurements are computed, one taking into account only the roll and pitch angles ($\delta_{\varphi\vartheta_i}$), and the second taking into account the full attitudes ($\delta_{\varphi\vartheta\psi_i}$).

$$\begin{cases} \delta q &= \mathrm{q}([\varphi_i,\ \vartheta_i,\ 0]) \otimes \mathrm{q}^*([\varphi_{i-1},\ \vartheta_{i-1},\ 0]) \\ \delta_{\varphi\vartheta_i} &= 2\left|\mathrm{atan}\left(\sqrt{\delta q_1^2 + \delta q_2^2 + \delta q_3^2}/\delta q_0\right)\right| \end{cases} \tag{12}$$

$$\begin{cases} \delta q &= \mathrm{q}([\varphi_i,\ \vartheta_i,\ \psi_i]) \otimes \mathrm{q}^*([\varphi_{i-1},\ \vartheta_{i-1},\ \psi_{i-1}]) \\ \delta_{\varphi\vartheta\psi_i} &= 2\left|\mathrm{atan}\left(\sqrt{\delta q_1^2 + \delta q_2^2 + \delta q_3^2}/\delta q_0\right)\right| \end{cases} \tag{13}$$

These indices are saved, together with the current measurement, on a shifting memory which collects the last l_{tw} values of indices and measurements (from $t = i + 1 - l_{tw}$ to $t = i$). It should be reminded that the roll and pitch angles are computed at the same time thought a nearest neighbor search, while the yaw angle is computed in a second step.

First, the condition on the continuity of the roll and pitch angles has to be satisfied. If all the stored values of $\delta_{\varphi\vartheta_t}$, $t \in [i + 1 - l_{tw}, i]$ are less than δ_{t_1}, the first condition $cond_1$ is satisfied. A reasonable value of l_{tw} is 10, while for δ_{t1} a value of 20 deg can be used.

Then, the continuity of the yaw angle is checked. The second condition $cond_2$ is automatically satisfied if all the stored values of $\delta_{\varphi\vartheta\psi_t}, t \in [i+1-l_{tw}, i]$ are less than δ_{t_1}. However, as outlined in Sec. 3, there is an indetermination in the estimation of the yaw angle for those couples of $(\varphi,\ \vartheta)$ that correspond to a projected silhouette having a 2-fold rotation symmetry [e.g., $(\varphi = 0 \vee \varphi = 180) \wedge \vartheta = 0$ deg]. For these attitudes, Eq.(4), which allows discriminating between ψ and $\psi + \pi$, is indeterminate due to the fact that the third order geometric moments of these silhouettes are null. This can lead to the presence of values of $\delta_{\varphi\vartheta\psi_t}$ close to 180 deg. Thus, an error in ψ close to 180 deg can indeed be corrected. A threshold δ_{t_2} is defined in order to set the acceptable difference between two consecutive yaw angles (a value of 20 deg can be used). From $t = i + 1 - l_{tw}$ to $t = i - 1$, the yaw discontinuity δ_{ψ_t} =min($|\psi_{t+1} - \psi_t|, 2\pi - |\psi_{t+1} - \psi_t|$) is computed. If at least one value of δ_{ψ_t} is in the interval $[\delta t_2, \pi - \delta t_2]$, the time window is discarded since the discontinuity in ψ cannot be attributed to the rotational symmetry of the silhouette. On the other side, if a value of δ_{ψ_t} is in the interval $[\pi - \delta t_2, \pi]$, a counter c_ψ (initialized at 0) is incremented by one and the value of the yaw angle ψ_{t+1} is shifted of π. When the algorithm has reached $t = i - 1$, c_ψ will indicate the number of shifts that have been performed along the considered time-window.

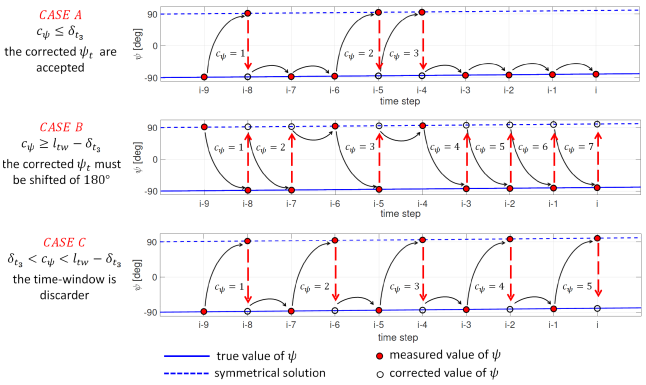


Figure 4. Correction algorithm detecting discontinuity of π (180 deg) in the yaw measurements, with $l_{tw} = 10$.

The value of c_ψ can go from 0 (no corrections have been performed) to $l_{tw} - 1$ (all the value of ψ_t except for $t = i+1-l_{tw}$ have been corrected). A value of $c_{\psi_{max}}$ must be defined to set the maximal number of accepted corrections over a time-window. In this study, a value of $c_{\psi_{max}} = 30\% l_{tw}$ has been choosen. If $c_\psi \le c_{\psi_{max}}$ (Fig. 4, Case A), the corrections of the ψ_t are accepted, and the values of $\delta_{\varphi\vartheta\psi_t}$, $t \in [i + 2 - l_{tw}, i]$ are recomputed in order to check if they verify $cond_2$ (i.e., $\delta_{\varphi\vartheta\psi_t} < \delta_{t_1}$). If $c_\psi \ge l_{tw} - c_{\psi_{max}}$ (Fig.

4, Case B), then all the corrected values of ψ must be shifted of π, and then checked for the verification of *cond*$_2$. Finally, if $c_{\psi_{max}} < c_{\psi} < l_{tw} - c_{\psi_{max}}$(Fig. 4, Case C), there are not enough values of ψ on the same solution branch, and therefore it is impossible to solve the ambiguity caused by the $2 - FRS$, and the time-window is discarded.
If both *cond*$_1$ and *cond*$_2$ are verified, then the current (and eventually corrected) measurement $q_{cam-tg_i} = \mathrm{q}([\varphi_i,\ \vartheta_i,\ \psi_i])$, together with the relative position t^{cam}_{cam-tg} computed according to Eq.(7), is used as prior to refine the measured pose using the contour-based tracking algorithm described in Sec.4, obtaining $\tilde{q}_{cam-tg_i}$, $\tilde{t}^{cam}_{cam-tg}$. Then, the reprojection error of the refined pose is computed according to (10), and compared with a given threshold δ_{t_3}. The value of δ_{t_3} can be chosen equal to the value of r_{max} at that range. If the reprojection error is less than δ_{t_3}, the third condition *cond*$_3$ is verified and the pose $\tilde{q}_{cam-tg_i}$, $\tilde{t}^{cam}_{cam-tg}$ is accepted as initial pose. The time instant i is then referred to as t_{acq_f} and the time instant $i + 1 - l_{tw}$ is referred to as t_{acq_i}.

Symmetry detection— If *cond*$_3$ is verified, the pose $\tilde{q}_{cam-tg_i}$, $\tilde{t}^{cam}_{cam-tg}$ is considered a valid measurement to start the tracking. However, the ambiguity on the attitude $\tilde{q}_{cam-tq_i}$ due to the target symmetry remains. As anticipated, this ambiguity can be solved only if $|\varphi| \in [80, 100]$ deg. If the measured roll angle is in this range during the pose acquisition time-window, when *cond*$_2$ is verified, two refined poses and residuals are computed using the tracking algorithm of Sec.4. The first is computed using the relative attitude $q_{cam-tg_i} = \mathrm{q}[\varphi_i,\ \vartheta_i,\ \psi_i]$, and the second using the symmetric relative attitude $q^{sym}_{cam-tg_i} = \mathrm{q}([-\varphi_i,\ -\vartheta_i,\ \psi_i - \pi])$. The pose providing the lower residual is selected (if the difference between the two residual is above a certain threshold) and the residual is compared to δ_{t_3} to check for *cond*$_3$. If $q^{sym}_{cam-tg_i}$ results being the pose with the lower residual, all the collected measurements $[\varphi_t,\ \vartheta_t,\ \psi_t]$, $t \in [i + 1 - l_{tw}, i]$ are "switched" to the symmetrical solution branch (i.e., $[\varphi_t,\ \vartheta_t,\ \psi_t] \rightarrow [-\varphi_t,\ -\vartheta_t,\ \psi_t - \pi]$). The symmetry is then considered to be initialized.
The tracking is initialized even if during the acquisition time-window the roll angle does not enter in the interval allowing the discrimination of the symmetric attitudes, Then, as soon as the estimated roll angle enters the range $|\varphi| \in [80,\ 100]$ deg, the procedure described above is repeated applying the criterion based on the residual previously described. If, during the tracking, the estimation must be switched to the symmetric solution branch, also the dynamic filter state must be corrected, according to the following:

$$\begin{array}{ll} q_{i-tg} &= [q_0,\ q_1,\ q_2,\ q_3]^T \\ q_{i-tg_{sym}} &= [-q_3,\ q_2, -q_1,\ q_0]^T \end{array} \qquad \begin{array}{ll} \omega^{tg}_{i-tg} &= [\omega_x,\ \omega_y,\ \omega_z]^T \\ \omega^{tg}_{i-tg_{sym}} &= [-\omega_x, -\omega_y,\ \omega_z]^T \end{array} \tag{14}$$

Eq.(14) can be obtained with simple kinematic relations knowing that the true target reference frame and the symmetric target reference frame are separated by a rotation of 180 deg around the axis z^{tg} (i.e., $x^{tg} = -x^{tg_{sym}}$ and $y^{tg} = -y^{tg_{sym}}$). Using the Jacobian J of the linear transformation $[q_{i-tg}, \omega^{tg}_{i-tg}] \rightarrow [q_{i-tg_{sym}}, \omega^{tg}_{i-tg_{sym}}]$ also the state covariance matrix P must be corrected (i.e., $P_{sym} = JPJ^T$).

Filter initialization—When the initial attitude $q_{cam-tg_{t_{acq}}}$ is acquired, the rotational dynamic filter has to be initialized. There is no need to do this for the translational dynamics since the translational KF as already been activated during the Range-only mode. All the measurement acquired (and eventually corrected) during the pose acquisition time-window are taken into account: the measurements are propagated inside a kinematic rotational filter from $t = t_{acq_i}$ to $t = t_{acq_f}$. As described in Sec.4, the filter state is composed by the target absolute attitude quaternion q_{i-tg} and by its rotation rate with respect to the inertial frame expressed in body axis ω^{tg}_{i-tg}. In a rotational kinematic filter, the prediction model associated to the quaternion is equal to the one in (11), while the derivative of the angular velocity is set to zero (i.e. $\dot{\omega}^{tg}_{i-tg} = 0$). The filter is initialized with zero rotation rate and state error covariance matrix equal to the identity matrix of size 7. Successively the kinematic filter goes from $t = t_{acq_i} + 1$ to $t = t_{acq_f}$. This allows to have a robust initialization of the target rotational state at the current instant t_{acq_f}, and allows the state covariance matrix P to start its convergence. The obtained values of the state covariance matrix P and the state x will be used as initialization for the dynamic rotational KF described in Sec.4.

6-DOF pose tracking

The 6-DOF pose tracking relies on two different and independent algorithms: a *detector* and a *tracker*. The detector relies on measurements computed using the method developed in Sec.3, and exploits the rotational kinematic KF to reject outliers and discriminate between multiple solutions caused by the target symmetries. The measurements computed by the detector are completely independent from the ones computed by recursive tracker. The role of the detector is to provide a corrected measurement in case of tracker divergences. On the other side, the recursive tracker is based on the algorithm developed in Sec.4 and its measurements are integrated in the dynamic filter. Only the state estimated by the coupling of the recursive tracker with the dynamic filter will be considered by the Guidance and the Control functions.

Detector— The logic of the detector is very simple. At the current time step t, the method described in Sec.3 is used to obtain a measurement of the relative camera-target attitude $q^{meas}_{cam-tg} = \mathrm{q}([\varphi,\ \vartheta,\ \psi])$. From the Euler angles, the symmetrical measurement $q^{sym}_{cam-tg} = \mathrm{q}([-\varphi,\ -\vartheta,\ \psi - \pi])$ is built. The goal then is to determine which is the more likely solution according to the target state at the previous instants. Thus, a kinematic filter is associated to the method. From the state estimated by the kinematic KF at the instant $t - 1$, the predicted state at the instant t is computed, and, using the current camera absolute attitude quaternion q_{i-cam}, the predicted q_{cam-tg} and the associated Euler angles $eu_{cam-tg} = [\varphi^{pred},\ \vartheta^{pred},\ \psi^{pred}]$ are obtained. The predicted quaternion serves as *anchor point* to select the nearest measurement between the actual output of the detection algorithm, and the symmetric one. In order to do this, only the roll and pitch angles of the predicted and the measured quaternion are considered (i.e., using the metrics $\delta_{\varphi\vartheta}$ of eq. (12), where the quaternion at the instant $i - 1$ is substituted by the predicted quaternion). The ambiguity on the two symmetric solution is solved by choosing the measurement having the lower $\delta_{\varphi\vartheta}$. The measurement is then accepted only if the full metrics $\delta_{\varphi\vartheta\psi}$ (i.e., from Eq.(13), where the quaternion at the instant $i - 1$ is substituted by the predicted quaternion) is below a threshold which can be set to be equal to δ_{t_1} (e.g., 20 deg). Before checking for this condition, the algorithm searches for eventual correctable discontinuities in the yaw angle, similarly to what was done in the pose acquisition function. The angular distance $\delta_{\psi} = \min(|\psi - \psi^{pred}|, 2\pi - |\psi - \psi^{pred}|)$ is computed. If δ_{ψ} is in the interval $[\pi - \delta t_2, \pi]$, then ψ must

be switched of π and the value $\delta_{\varphi\vartheta\psi_{meas}}$ must be updated. At the end of these two step, if the new $\delta_{\varphi\vartheta\psi}$ is below δ_{t_1}, the measurement can be used in the update step of the kinematic filter. Otherwise, the measurement is considered to be an outlier and it is discarded. In such a case, the KF estimated state will be equal to the KF predicted state. A counter c_{kine} is increased by one any time a measurement is discarded, and it is reset to zero any time a measurement is accepted and used in the KF update step. The counter indicates the number of time step during which the state has been propagated without any measurement update. If the counter exceed a certain threshold $c_{kine_{max}}$ (with $c_{kine_{max}} \sim 20$ step), the state is no more reliable and the detector must be reinitialized though the procedure discussed in the previous *6-DOF pose acquisition* paragraph. In such a case, a flag indicating the availability of the detector measurements must be set to *false* as long as a new pose is acquired. It should be noted that this reinitialization will not affect the recursive tracker and the dynamic filter estimated state. Only in case of its own divergence the tracker will "consult" the detector.

Tracker—The tracker is based on the contour-based tracking algorithm discussed in Sec.4. While the IP-CV "core" of the tracker (masking algorithm, matching, cost function minimization) has not been modified for the implementation of the full solution, its interface with the dynamic filter has been changed in order to benefit from the addition of the detection algorithm.
As anticipated, the measurement computed by the frame-by-frame tracker is associated to a residual r (i.e., Eq.(10)). The value of r is used to dynamically change the measurement noise covariance matrix R. If the residual exceeds the threshold r_{max}, the measurement computed by the recursive tracker is discarded. In such a case, the recursive tracker "asks" the detector for its measured pose (downstream of the kinematic KF). Three scenarios are possible:

- The validity flag of the detector measurement is *true*: the estimated relative pose sent by the detector is used as prior to project the 3D model and re-run the tracking algorithm. If the residual associated to this refined measurement is below r_{max}, the measurement is accepted and used in the dynamic KF to perform the state update.
- The validity flag of the detector measurement is *true*, but the residual obtained refining the detector measurement is above the threshold r_{max}. The measurement is discarded, and the state in the dynamic Kalman filter is propagated in "open-loop" (only prediction step).
- The validity flag of the detector estimate is *false*. This is the worst case scenario, since the detector estimate is not available to correct the divergence in the recursive tracker. In such a scenario, the state of the dynamic filter is propagated without the update step as long as the pose is re-acquired.

If the dynamic filter does not receive any update for a given number of time steps (i.e., the number of open-loop cycles of the KF exceeds the threshold $pr\ max$ in Fig.5), it is the filter that ask for a full re-initialization of the pose. If the pose is not reacquired after a given number of time-steps (i.e., $it\ max$ in Fig.5), an error is sent at higher level. If the tracking is lost, the chaser should stop its approach motion to begin a new acquisition phase or even perform a collision avoidance maneuver if necessary. The schema in Fig.5 summarizes the logic of the 6-DOF pose estimation function.

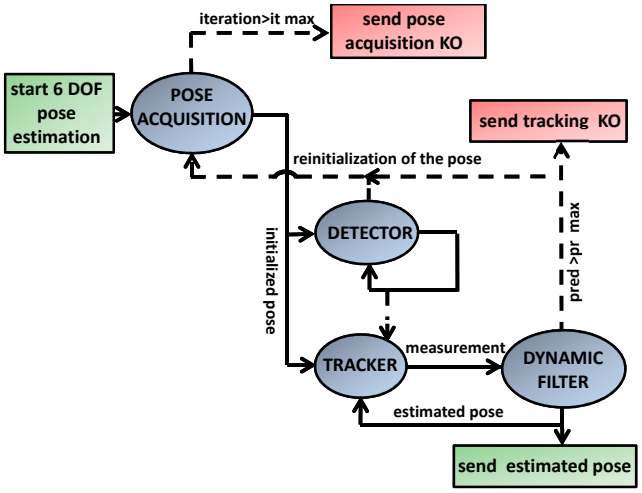


Figure 5. Architecture of the 6-DOF pose estimation function

6. APPLICATION TO AN OPERATIONAL SCENARIO

The navigation architecture has been tested on a forced motion scenario from a relative distance of 100 m to 5 m. In the first 2 minutes of the simulation, the chaser approaches the target along the y axis of the LOF at a relative speed of 0.5 m/s. During this phase, only range measurements are provided using the algorithm discussed in Sec.3. When the chaser reaches a distance of 40 m from the target, it stops the motion and starts the 6-DOF pose acquisition phase, according to the method discussed in Sec.5. After 2 minutes, the chaser start another forced motion towards the target at a relative speed of 0.2 m/s. The approach velocities have been chosen in order to cover all the phases in a relatively short simulation, but remains high compared to the approach rates usually used during close proximity operations. However, this allows to test the tracking algorithm in presence of high displacement of the target from one image frame to the following one. The target rotational rate is initialized at 1.5 deg/s around each body axis. The camera acquisition frequency is 1 Hz, while the navigation filter frequency is 10 Hz. In order to run the detection algorithm, three different databases have been built. The first one is built at a distance of $d_{train} = 40$ m, and it is used in the range-only phase, in the pose acquisition phase and during the tracking, down to a relative distance of 30 m. The second database is built at $d_{train} = 20$ m and it is used for a relative distance in the interval $[30, 15]$ m. For the distances below 15 m a database built at $d_{train} = 10$ m is used. All the databases have the size of 5500×56 (i.e., $N_w = 5500$ samples, $N_f = 51$ Zernike rotation invariants up to the 9^{th} order, and 4 more global features being $\psi_{0_{train}}$, $m_{00_{train}}$, φ_{train}, and ϑ_{train}). As the distance decreases, the interval of distances in which a database built at a given distance d_{train} can provide reliable detections become smaller. This is due to the fact that the weak perspective approximation is not applicable for shorter ranges. For a 30 deg FOV camera and the considered target geometry, the detection algorithm does not provide reliable measurements below ~ 8 m.

Discussion of the results

The simulation is run with three different series of images:

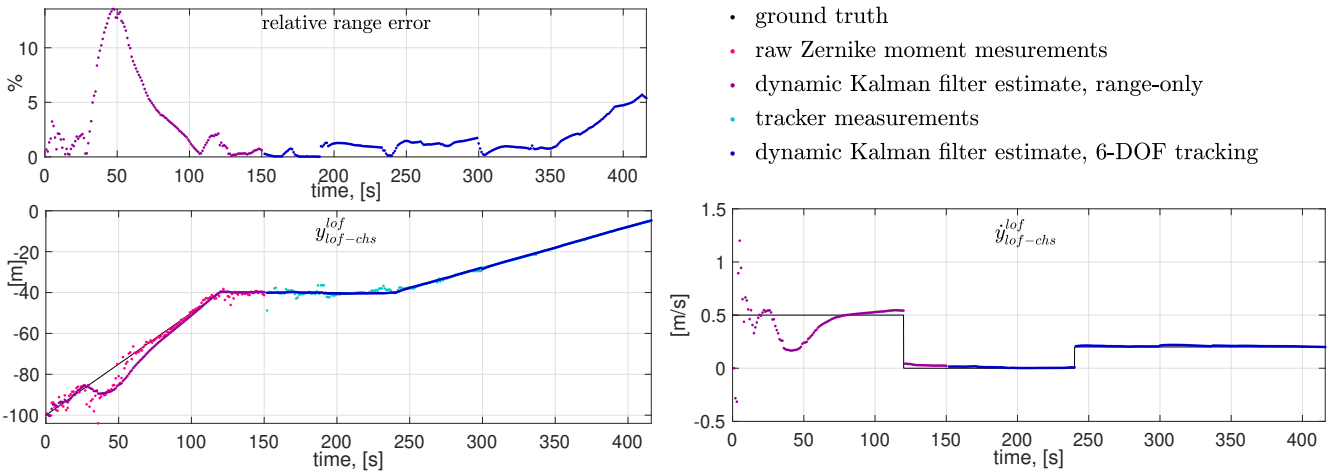


Figure 6. Translational dynamics estimation

- CASE A: The first set of images are perfect binary silhouettes of the target satellite. The simulation run with this set is the baseline simulation which shows the intrinsic performance of the pose estimation algorithm.
- CASE B: The second set of images is composed by grey-scale images obtained illuminating the target satellite with a spot-light placed on the chaser. This could be the case of a rendezvous performed approaching the target from the -R-bar side, or a rendezvous on a geo-stationary orbit performed along the V-bar. The images are affected by shadows, reflections and blur, so that the silhouette retrieved is affected by multiple sources of noise. The silhouettes are retrieved using a simple thresholding technique.
- CASE C: The third set of images is composed by grey-scale images obtained illuminating the target satellite with a spot-light placed on the chaser, and Earth in the background. This is the case that is more likely to happen during rendezvous on a Low Earth orbit. The method discussed in Sec.2 is used to extract the silhouette from the background.

Figures 6 and 7 show the results of the estimation for CASE C, which is the most challenging scenario. In Fig.6 the y component of relative translational state in LOF reference frame is shown, with both the estimated position and the estimated velocity. The pose is acquired at $t_{acq_f} = 155$ seconds. Before t_{acq_f}, the navigation function is in range-only mode and the dynamic translational filter (i.e., violet dots) uses the raw measurements coming from the Zernike moment based algorithm (i.e., magenta dots). After t_{acq_f}, the 6-DOF pose tracking starts and the dynamic translational filter (i.e., blue dots) uses the measurements coming from the tracker (i.e., cyan dots). The relative range error (in percentage) is also shown. Fig.7 shows the angular error of the estimated attitude. The plot shows the angular error from $t = t_{acq_f}$ (i.e., magenta dots), obtained using the dynamic filter and the tracker measurements. When the dynamic filter uses the refined detector measurements to compute the estimated, the error is plotted in violet. When no measurements are available and the filter evolves in open-loop, the angular error is plotted in cyan. The angular errors obtained during the time acquisition window (i.e., $t_{acq_i} \leq t \leq t_{acq_f}$) are displayed in blue. The green dots indicate the angular error obtained shifting the estimate of π, before the ambiguity on the symmetry is solved at $t = 176$ seconds. Note that the angular error is not always decreasing as the relative distance decreases (e.g., there is a peak around $t = 260$ s). This is due to the fact that the estimation performance does not depend only on the relative distance, but also on the current attitude of the target. Table 1 shows the performance in the estimation

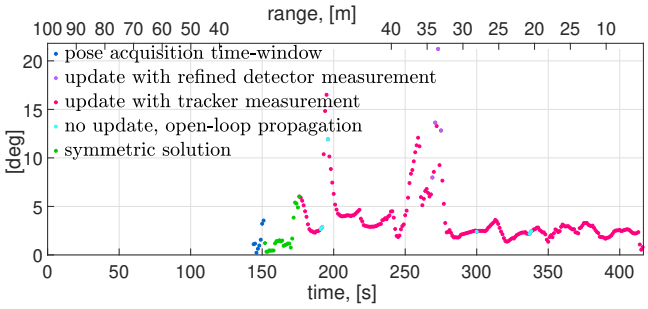


Figure 7. $\delta_{\varphi\vartheta\psi_{est}}$, angular error of the estimated attitude

Table 1. Comparison of the performance of the 6-DOF pose estimation algorithm as a function of the range

Rotational dynamics, estimated $\delta_{\varphi\vartheta\psi}$ [deg]

range [m]:		$= 40$	$\in]40, 30]$	$\in]30, 20]$	$\in]20, 10]$	< 10	*overall*
CASE A	μ	0.98	1.11	0.39	0.43	1.09	0.80
	rms	1.72	1.40	0.45	0.57	1.17	1.28
CASE B	μ	1.30	4.10	1.27	1.02	1.19	1.75
	rms	1.44	5.07	1.47	1.23	1.38	2.52
CASE C	μ	3.76	5.92	2.36	2.52	2.16	3.51
	rms	4.68	7.22	2.42	2.57	2.23	4.48

Translational dynamics: estimated range error [m]

range [m]:		$= 40$	$\in]40, 30]$	$\in]30, 20]$	$\in]20, 10]$	< 10	*overall*
CASE A	μ	0.195	0.267	0.107	0.043	0.045	0.150
	rms	0.206	0.279	0.115	0.047	0.046	0.180
CASE B	μ	0.591	1.845	0.507	0.192	0.024	0.679
	rms	0.649	1.903	0.564	0.224	0.029	0.940
CASE C	μ	0.240	0.422	0.229	0.275	0.366	0.291
	rms	0.307	0.430	0.261	0.288	0.369	0.329

of the target attitude and the relative position of all cases, as a function of the range. The metric used for the rotational dynamics performance is the angular error $\delta_{\varphi\vartheta\psi}$. The metric used for the translational dynamic is the range error. Both the mean knowledge error μ (i.e., the MKE, which is the

mean of the Absolute Knowledge Error AKE according to [48]) and the root mean square of the AKE are listed. It is evident how the performance is degraded in cases B and C. However at close range (i.e., below 20 m), the performance is satisfying in all the scenarios. For what concerns the translation estimation performance, it is interesting to notice how the overall performance of CASE B (i.e., no Earth in background, simple thresholding) is worse than the overall performance of CASE C (i.e., Earth in background, background subtraction and silhouette post-processing). This suggests that the post-processing technique proposed in Sec.2 should be used also in case of deep-space background to enhance the silhouette extraction. For what concerns the performance of CASE C, significant improvement could be obtained by training the background subtractions neural network with a bigger training dataset and increasing the number of layers in the network, but this could lead to an augmentation of the computational load of the algorithm. The performance of the estimation algorithm on the test CASE C shows that the navigation solution is quite robust with respect to the presence of noise and errors on the extracted silhouette. This means that, in the presence of deformation of the target satellite due to the flexibility of the appendages or to damaged parts, the algorithm ensures a certain level of robustness. However, in the presence of big non-modeled differences, the 3D a priori model upon which the estimation algorithms rely could be updated from ground after an inspection phase. In fact, the segmentation algorithm is the only one which requires a relatively long training phase on-ground, but it could be made model-invariant by increasing the training dataset size with images of different spacecrafts and eventually increasing the number of the layers. For what concerns the detection algorithm, the new feature database can be built in less than a hour once that the new model has been estimated, and then updated from ground.

Latency of the algorithms

The segmentation stage has been tested on a 2.20 GHz Intel Core i7 development laptop equipped with a Nvidia GTX 1070 GPU. While real-time performances can be achieved (e.g., averagely 36 millisecond for each image on the development laptop), it is not representative of a real world scenario were the processing chain must run on a space embedded target. The process of porting the neural network on a space compatible FPGA, with the objective of achieving at least a working frequency of 1 Hz and ideally 10 Hz is ongoing. There are also possibilities to decrease, if needed, the neural network resource consumption by alleviating it (e.g., reducing the number of layers and/or the number of kernel in the layers) at the cost of its precision, but this could be compensated by efficient post processing able to correct this. Moreover, reducing the image size could largely speed up the processing with acceptable precision trade-off.
The pose estimation algorithms were tested on a 2.70 GHz Intel Core i7 processor. All the algorithms are autocoded in C++ from MATLAB code. The average time needed to compute ZM invariants up to the 9^{th} order is equal to 208 millisecond, 387 milliseconds and 1.16 seconds respectively at 40 m, 20 m, and 10 m. The latency of the algorithm at 10 m as well as it computational load (i.e., due to the number of silhouette points to process in order to compute the rotation invariants) is not compatible with real-time implementation. However, this computation time can be drastically reduced by subsampling the image before computing the moment invariants. The results shown in [31] prove that the method is robust to changes in the resolution of the images. Note however that this absolute computation time is mostly indicative, since the computation of moment invariants could be optimized [49], [50], [51] and could benefit from FPGAs implementation, which can lead to significant improvements. The time needed to compute the moment invariants is the dominant part of the detection algorithm: with a database of the size considered in this study, the matching requires less than 0.1% of the time needed to compute the 9^{th} order ZM invariants at 20 m.
On the same processor, the tracking algorithm (from the Canny edge extraction to the cost function minimization) , has an average latency of 60.1 millisecond at 40 m (i.e., 16.64 images/s), $62,5$ millisecond at 20 m (i.e., 16 images/s), and 85.8 millisecond par image (i.e., 11.66 images/s) at 10 m.
In the pose acquisition phase, at each time step, the latency will be the summation of the time needed for the segmentation and the time needed for the computation of the ZM invariants. The latency needed to compute the angular distances between two consecutive measurements and eventually correct the ψ angle is almost negligible. At the time instant where $cond_1$ and $cond_2$ are satisfied, the latency of the pose acquisition function must take into account also the execution time of the contour-based tracking algorithm. If the symmetry must be detected, the latency of the tracker must be counted twice. The filter initialization function has a latency that can be neglected with respect to the pose acquisition function. During the 6DOF tracking, if the detector and the tracker are executed in parallel (which is possible since they are independent one from the other), the latency is equal to the time needed by the segmentation plus the highest latency among the tracker and the detector. However, if the tracker measurement isn't accepted, the detector measurement must be corrected using the tracker and the latency of the tracker is therefore doubled.
The performance shown in the previous paragraph has been obtained in presence of a very high relative dynamics (i.e., very high approach velocity of the chaser and the high rotation rate of the target), which is very unlikely to happen in a real-world scenario. This allows us to state that, if - after implementation of the algorithm on a space-qualified processor- the working frequency of 1 Hz is not reached, the acquisition rate could be slightly decreased without any major degradation of the estimation performance.

7. CONCLUSION

The current paper has addressed the problem of monocular model-based pose estimation for close-proximity operations during space rendezvous using a visible monocular camera. A complete navigation solution, covering all the navigation phases from the 6 degrees-of-freedom pose acquisition to the robust full pose tracking, has been presented and tested in different scenarios. The tracking relies on a contour-based recursive tracker which provides very precise measurements and a global-feature based detection algorithm which allows pose initialization and measurement correction in case of divergence of the tracker. A background subtraction algorithm based on convolutional neural network has been integrated to the navigation algorithms. The method enables the robust extraction of the target silhouette from the acquired image, allowing to reach satisfying estimation performance even in presence of complex background such as the Earth. The segmentation algorithm would certainly benefit from the use of multi-spectral images, especially in the thermal infrared spectrum, and from the increase of the train database size. With the implementation of this improvements, the estimation performance in presence of complex background is likely to get closer to the baseline performance with perfect silhouettes.

The proposed estimation chain provides a cost effective solution with a relatively low computational load. The latency of the navigation functions on standard PC allows to assume that, after the implementation of the algorithms on a dedicated hardware such as a FPGA, the navigation chain will satisfy the requirement of 1 Hz acquisition rate. Future works will focus on the porting of the navigation algorithms on a space-qualified hardware.

REFERENCES

[1] D. J. Kessler, N. L. Johnson, J. Liou, and M. Matney, "The kessler syndrome: implications to future space operations," *Advances in the Astronautical Sciences*, vol. 137, no. 8, p. 2010, 2010.

[2] P. Jasiobedzki, M. Abraham, P. Newhook, and J. Talbot, "Model based pose estimation for autonomous operations in space," in *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*. IEEE, 1999, pp. 211–215, https://doi.org/10.1109/ICIIS.1999.810263.

[3] M. E. Polites, *An assessment of the technology of automated rendezvous and capture in space*. NASA, 1998.

[4] CONFERS, "Satellite servicing safety framework technical and operational guidance document," https://www.satelliteconfers.org/publications/, 2018, online; [retrieved 06 December 2019].

[5] J. M. Kelsey, J. Byrne, M. Cosgrove, S. Seereeram, and R. K. Mehra, "Vision-based relative pose estimation for autonomous rendezvous and docking," in *Aerospace Conference, 2006 IEEE*. IEEE, 2006, pp. 20–pp, https://doi.org/10.1109/AERO.2006.1655916.

[6] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "A model-based 3d template matching technique for pose acquisition of an uncooperative space object," *Sensors*, vol. 15, no. 3, pp. 6360–6382, 2015.

[7] ——, "Uncooperative pose estimation with a lidar-based system," *Acta Astronautica*, vol. 110, pp. 287–297, 2015.

[8] V. Lepetit, P. Fua *et al.*, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[9] M. D. Lichter and S. Dubowsky, "State, shape, and parameter estimation of space objects from range images," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 2974–2979, https://doi.org/10.1109/ROBOT.2004.1307513.

[10] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[11] S. Sharma, J. Ventura, and S. D'Amico, "Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous," *Journal of Spacecraft and Rockets*, vol. 55, no. 6, pp. 1414–1429, 2018.

[12] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, and M. Grassi, "Autonomous relative navigation around uncooperative spacecraft based on a single camera," *Aerospace Science and Technology*, vol. 84, pp. 1070–1080, 2019.

[13] D. Rondao and N. Aouf, "Multi-view monocular pose estimation for spacecraft relative navigation," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 2100.

[14] A. Petit, "Robust visual detection and tracking of complex objects: applications to space autonomous rendezvous and proximity operations," Ph.D. dissertation, Université Rennes 1, 2013.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[16] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2257–2264.

[17] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *Proceedings: Image Understanding Workshop*. Science Applications, Inc Arlington, VA, 1977, pp. 21–27.

[18] S. Sharma and S. D'Amico, "Neural network-based pose estimation for noncooperative spacecraft rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2020.

[19] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–12.

[20] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, and S. D'Amico, "Satellite pose estimation challenge: Dataset, competition design and results," *IEEE Transactions on Aerospace and Electronic Systems*, 2020.

[21] J. Sola, "Quaternion kinematics for the error-state kf," Laboratoire d'Analyse et d'Architecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep, Tech. Rep., 2015.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," arXiv:1703.06870v3, 2018.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[25] T. P. Wallace and P. A. Wintz, "An efficient three-dimensional aircraft recognition algorithm using normalized Fourier descriptors," *Computer Graphics and Image Processing*, vol. 13, no. 2, pp. 99–126, 1980.

[26] A. P. Reeves, R. J. Prokop, S. E. Andrews, and F. P. Kuhl, "Three-dimensional shape analysis using moments and Fourier descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 937–943, 1988.

[27] Z. Chen and S.-Y. Ho, "Computer vision for robust 3D aircraft recognition with fast library search," *Pattern recognition*, vol. 24, no. 5, pp. 375–390, 1991.

[28] T. Glais and A. Ayoun, "Image-based air target identification," in *Applications of Digital Image Processing XVII*, vol. 2298. International Society for Optics and Photonics, 1994, pp. 540–551.

[29] S. A. Dudani, K. J. Breeding, and R. B. McGhee, "Aircraft identification by moment invariants," *IEEE transactions on computers*, vol. 100, no. 1, pp. 39–46, 1977.

[30] M. G. Breuers, "Image-based aircraft pose estimation using moment invariants," in *Automatic Target Recognition IX*, vol. 3718. International Society for Optics and Photonics, 1999, pp. 294–304.

[31] A. Comellini, J. LE Ny, E. Zenou, V. Dubanchet, and C. Espinosa, "Global descriptors for visual pose estimation of a non cooperative target in space rendezvous,**Under Review**," *Transaction on Aerospace and Electronic Systems*, pp. –, -.

[32] J. Flusser, B. Zitova, and T. Suk, *Moments and moment invariants in pattern recognition*. John Wiley & Sons, 2009, pp 7,13-30,186-197.

[33] O. Tahri, "Application des moments a l'asservissement visuel et au calcul de pose," *These de doctorat, Université de Rennes*, vol. 1, 2004.

[34] C. Harris and C. Stennett, "Rapid-a video rate object tracker." in *BMVC*, 1990, pp. 1–6, https://doi.org/10.5244/C.4.15.

[35] A. Petit, E. Marchand, and K. Kanani, "Tracking complex targets for space rendezvous and debris removal applications," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4483–4488.

[36] ——, "Vision-based space autonomous rendezvous: A case study," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 619–624.

[37] ——, "Détection et suivi basé modèle pour des applications spatiales," in *Congrès francophone des jeunes chercheurs en vision par ordinateur, ORASIS'13*, 2013, pp. 1–6.

[38] M. Lourakis and X. Zabulis, "Model-based visual tracking of orbiting satellites using edges," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3791–3796.

[39] G. Lentaris, I. Stratakos, I. Stamoulias, D. Soudris, M. Lourakis, and X. Zabulis, "High-performance vision-based navigation on soc fpga for spacecraft proximity operations," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[40] F. L. Markley, "Attitude estimation or quaternion estimation?" *Journal of Astronautical Sciences*, vol. 52, no. 1, pp. 221–238, 2004.

[41] A. Comellini, E. Zenou, V. Dubanchet, and C. Espinosa, "Vision-based navigation for autonomous space rendezvous with non-cooperative targets," in *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 2020, pp. 1–4.

[42] H. Gavin, "The levenberg-marquardt method for nonlinear least squares curve-fitting problems," *Department of Civil and Environmental Engineering, Duke University*, pp. 1–15, 2011.

[43] A. Comellini, D. Casu, E. Zenou, V. Dubanchet, and C. Espinosa, "Incorporating delayed and multirate measurements in navigation filter for autonomous space rendezvous," *Journal of Guidance, Control, and Dynamics*, pp. 1–9, 2020.

[44] W. Fehse, *Automated rendezvous and docking of spacecraft*. Cambridge university press, 2003, vol. 16, pp. 127-133, 185, 289, 424-438.

[45] T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Incorporation of time delayed measurements in a discrete-time kalman filter," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, vol. 4. IEEE, 1998, pp. 3972–3977, doi: https://doi.org/10.1109/cdc.1998.761918.

[46] V. Prasad, M. Schley, L. P. Russo, and B. W. Bequette, "Product property and production rate control of styrene polymerization," *Journal of Process Control*, vol. 12, no. 3, pp. 353–372, 2002, doi: https://doi.org/10.1016/s0959-1524(01)00044-0.

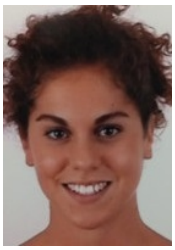
[47] R. Li, A. B. Corripio, M. A. Henson, and M. J. Kurtz, "On-line state and parameter estimation of epdm polymerization reactors using a hierarchical extended kalman filter," *Journal of process control*, vol. 14, no. 8, pp. 837–852, 2004, doi: https://doi.org/10.1016/j.jprocont.2004.03.002.

[48] T. Ott, A. Benoit, P. Van den Braembussche, and W. Fichter, "Esa pointing error engineering handbook," in *8th International ESA Conference on Guidance, Navigation & Control Systems*, 2011, p. 17.

[49] A. Prata and W. Rusch, "Algorithm for computation of Zernike polynomials expansion coefficients," *Applied Optics*, vol. 28, no. 4, pp. 749–754, 1989.

[50] S.-K. Hwang and W.-Y. Kim, "A novel approach to the fast computation of Zernike moments," *Pattern Recognition*, vol. 39, no. 11, pp. 2065–2076, 2006.

[51] C.-Y. Wee, R. Paramesran, and F. Takeda, "New computational methods for full and subset Zernike moments," *Information Sciences*, vol. 159, no. 3-4, pp. 203–220, 2004.



Anthea Comellini *received the M.Sc double-degree in space engineering from Politecnico di Milano, Italy, and ISAE-SUPAERO, France, in 2017, and the M.Sc. in automation and signal processing from Paris-Saclay University, France, in 2018. She is currently pursuing the Ph.D. degree in space engineering with ISAE-SUPAERO and Thales Alenia Space, France. Her research interests include vision-based navigation and GNC for space rendezvous.*

Florent Mayé *received the M.Sc in Intelligent Systems engineering from Ecole des Mines de Saint-Etienne, France and has also studied artificial intelligence and robotics at National Taiwan University. He is currently working as software R&D engineer at Thales Alenia Space since 2016, specialized in artificial intelligence, computer vision and robotics.*

Vincent Dubanchet *received the M.Sc in aerospace engineering from ISAE-SUPAERO, France, and the M.Sc in electric and electronic engineering from Polytechnique Montréal, Canada, in 2012. In 2016 he obtained the Ph.D. degree from ISAE-SUPAERO and Polytechnique Montréal, for the modeling and control of a flexible space robot to capture tumbling debris. He is currently working as a R&D engineer in Guidance, Navigation & Control and robotics in the R&D department in Thales Alenia Space Cannes. His research interests include in-orbit servicing and autonomous rendezvous.*

Davide Casu *received the M.Sc in aerospace engineering from Politecnico di Milano, Italy, in 2019. He is currently working as a R&D engineer in Guidance, Navigation & Control in the R&D department in Thales Alenia Space Cannes, and he is involved in multiple R&D studies and Advanced Projects related to space Rendezvous & On-Orbit Servicing. His research interests include On-Board GNC & Autonomy.*

Emmanuel Zenou *is graduated from "Ecole Normale Supérieure de Paris-Saclay" in 2001 and received a PhD degree from "Ecole Nationale Supérieure de l'Aéronautique et de l'Espace" (SUPAERO) in 2004 in the field of Robotics, Computer Vision and Computer Science. He has a full-time position at ISAE-SUPAERO in the field of Computer Vision and Data Science, in the department of Complex System Engineering (DISC). Since 2017, he is Deputy Director for International Affairs at ISAE-SUPAERO and Deputy Chairman of the Pegasus Network.*

Christine Espinosa *received the Master in Applied Mathematics in 1982, and the PhD in Theoretical and Computational Mechanics from the University of BORDEAUX I (France, 1991) for the development of a non-linear damage anisotropic behaviour for aeronautical composite materials subjected to low velocity impacts. She worked 9 years in DYNALIS Cie as rapid nonlinear simulations engineer and contributed developing FSI solutions in LS-DYNA®. Full Professor of ISAE-SUPAERO since 2018 in Computational Mechanics for strength analysis of structures under severe multiphysic loads, she received the 'Palmes Académiques' in 2019.*