Comellini, Anthea and Zenou, Emmanuel and Espinosa, Christine and Dubanchet, Vincent Vision-based navigation for autonomous space rendezvous with non-cooperative targets. (2020) In: The 11th International Conference on Information, Intelligence, Systems and Applications (IISA2020), 15 July 2020 - 17 July 2020 (Piraeus, Greece).

Vision-based navigation for autonomous space rendezvous with non-cooperative targets

Anthea Comellini, Emmanuel Zenou, Christine Espinosa, and Vincent Dubanchet

***Abstract*—This study addresses the issue of vision-based navigation for space rendezvous with non-cooperative targets. After a brief description of the scenario and its peculiarities, the theory underlying monocular edges-based tracking for pose estimation is recalled and an innovative tracking algorithm is formally developed and implemented. This algorithm is coupled with a dynamic Kalman Filter propagating the dynamics which underlies a space rendezvous. The navigation filter increases the robustness of target position and attitude estimation, and allows the estimation of target translational velocity and rotation rate using only pose measurements. Moreover, the filter implements a computationally efficient delay management technique that allows merging the delayed and infrequent measurements typical of vision-based navigation. The performance of the algorithm is tested in different scenarios with high fidelity synthetic images.**

***Index Terms*—Autonomous space rendezvous, Monocular tracking, Autonomous navigation, Non-cooperative target**

I. Introduction

Autonomous rendezvous (RDV) and docking or capture are key capabilities 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], operations that are rarely performed and up till now only by astronauts with a minimal support of robotic system (e.g. ISS, Hubble Space Telescope and MIR Station repairs) and have a very high cost. Nowadays the failure of a single system of a S/C can irrevocably compromise its whole mission. A change in the design of satellites, leading to a modular structure, could allow in-orbit-repair by a servicer S/C. Moreover, OOS can have a positive impact on space environment and help reducing the proliferation of space debris, providing life extension services to already orbiting S/C and carrying out end-of-life disposal. The two actors in a rendezvous are referred to as the *chaser* and the *target*. According to Ref. [3], "the chaser vehicle is a spacecraft which has both attitude and translational control capability. It actively navigates to the target vehicle in the rendezvous process". Rendezvous scenarios are many and varied, and depend mainly on the nature of the target. The definition and classification of targets in literature has changed through the years as OOS and ADR became a real prospect for space operations. 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 such as visual markers, navigational aids, docking fixture 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 yet to be 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, Control & Navigation) 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. For this reason, the coupling of camera sensors with image processing (IP) and computer vision (CV) algorithms can provide a cost effective solution.

II. Monocular model-based tracking

Image-based pose estimation 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

PhD student, ISAE-SUPAERO, 31400 Toulouse, France, and Thales Alenia Space, 06150 Cannes, France; anthea.comellini@isae.fr

Associate professor, ISAE-SUPAERO, 31400 Toulouse, France; emmanuel.zenou@isae.fr

Research professor, Institut Clément Ader ICA, Université de Toulouse, ISAE-SUPAERO, IMT MINES ALBI, UTIII, INSA, CNRS, 31400 Toulouse, France; christine.espinosa@isae.fr

PhD and GNC engineer at Thales Alenia Space, Cannes, France, 06150; vincent.dubanchet@thalesaleniaspace.com

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 [6]. 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 (which could incorporate also textures, reflectance and other visual attributes). Fiducials are features expressly designed in such a way (shape/colour) that are easily detected and identified with an *ad hoc* method [7] (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, the 3D model of a satellite is exploitable even if the S/C was not supposed to participate in RDV operation. 3D model-based tracking algorithms are therefore more general than fiducial points based ones, and this is why they are the best candidates for this study. The model matching process computes a cost function describing how well the 3D model, in a specific pose or configuration, matches the observed data in image [2]. In the case of a monocular sensor (i.e., a single camera), the matching will be performed between the 2D projection of the 3D model and the image. Within the many 3D model-based techniques for monocular camera, 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 [7]. 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 Ref. [8] and it was one of the first monocular 3D tracker to successfully run in real-time due to its low computational complexity. At instant K, the 3D *a priori* model is projected in the image frame using the pose parameters estimated at instant $K-1$. 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 K, 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.

The first use of RAPiD for vision-based autonomous space rendezvous dates from the year 2006 [5]. The study underlines why simple cameras should be preferred with respect to LIDAR and other scanning and ranging sensor which could provide good relative position accuracy at distances up to several kilometres as well as relative orientation cues at shorter distance, but are typically very expensive and power hungry, and often characterized by a small field of view (FOV). The algorithm in Ref. [5] relies on the iterative recursive least squares method (IRLS) to reject outliers proposed in Ref. [9] and includes a Kalman enhancement (as suggested by Ref. [10]) in order to improve the robustness of the estimation. The application of a non-linear version of RAPiD in space rendezvous has been assessed also in Refs. [11]–[13], 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. [14], [15]. All the cited works integrate, to the pose estimation algorithm, a linear Kalman filter (KF) which propagates a simple kinematic model. However, in the case of high rotation rates typical of a tumbling object, a simple kinematic filter doesn't 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.

This paper has two main contributions. First, it proposes a robust tracking algorithm, based on a non-linear version of RAPID method. The method does not need any GPU augmentation and it is compatible with typical space processing capabilities. Secondly, the paper proposes a coupling of the developed IP-CV algorithm with a navigation filter enabling the estimation of full target rotational and translational state exploiting only relative pose measurement. Moreover, 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 time.

The rest of the paper is structured as follows. In Sec.III the implementation of the proposed tracking algorithm is presented: Sec. III-A recalls the theory underlying RAPID algorithm, while Sec.III-B and Sec.III-C describe in details the proposed IP-CV algorithm; in Sec. III-D the coupling with the dynamic filter is discussed. In Sec.IV the performance of the proposed solution is investigated, and in Sec.V the conclusions are drown.

III. Implementation of the methods

In this section the theory underlying RAPiD and our algorithm will be presented. Both the methods rely on the same steps for the edge extraction, the projection of the *a priori* model, and the matching, but differ in the construction of the cost function. The edge extraction on the greyscale image captured at instant K relies on the Canny edge detector, whose output is a binary image which is white in correspondence of edges and black elsewhere. For the projection of the model, a "masking algorithm", which behaves almost like a classic z-buffer algorithm, has been developed. Firstly the *a priori* model is projected, using the estimated pose at the instant $K-1$, in the 3D image space, and then surfaces and edges are meshed. If the projection in the $x-y$ plane of a point i

belonging to an edge is superposed to the projection of any other point j of the meshed surfaces, point i is retained only if it's z-coordinate is lower than the z-coordinate of j. At the end of the iteration, the output of the masking algorithm is a set of 2D visible points (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 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. Of course this can produce a discrete number of false matching, which is one of the main drawbacks of the mono-directional search that underlies the method fastness. Because the image is a discrete domain, all the pixel that are crossed by the normal vector have to be checked. 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. Simulations have shown that it is useless to underestimate the value of d_{toll} since this prevents the algorithm to recover from a small divergence. On the same way, overestimated values of d_{toll} lead to a high number of false matches. Moreover, above a certain value of d_{toll} the hypotheses of small pose variation underlying the matching procedure are no more valid. Once that the set of matched points is computed, the cost function can be built and the new pose is computed.

A. Linear optimisation with RAPiD

The analytical formalisation of RAPiD proposed in Ref. [7] is now briefly recalled. This is necessary to help the reader to better understand the method proposed in Sec.III-B.

Any point $\mathbf{X} = [x, y, z]$ belonging to the target object and expressed in target reference frame (RF) coordinates (${}^{tg}\mathbf{X} = \mathbf{X}$), can be formulated in camera RF at instant $K-1$ according to:

$$\mathbf{M} = \mathbf{t} + \underline{\underline{\mathbf{R}}}\,\mathbf{X} \tag{1}$$

Note that $\mathbf{t}$ is the translation vector that describe the relative position of target RF with respect to camera RF, expressed in camera RF ($\mathbf{t} = {}^{cam}\mathbf{t}_{cam-tg}$). The rotation matrix $\underline{\underline{\mathbf{R}}} = \underline{\underline{\mathbf{R}}}_{cam-tg}$ is the rotation matrix such that ${}^{cam}\mathbf{X} = \underline{\underline{\mathbf{R}}}_{cam-tg}\,{}^{tg}\mathbf{X}$. The same target point $\mathbf{X}$, at instant K, will have a new pose expressed by the parameters $\mathbf{t}'$ and $\underline{\underline{\mathbf{R}}}'$, which can be described as the composition of the pose at instant $K-1$ ($\mathbf{t}$, $\underline{\underline{\mathbf{R}}}$) and the pose increment ($\delta\mathbf{t}$, $\underline{\underline{\mathbf{\Delta R}}}$):

$$\mathbf{M}' \quad = \mathbf{t}' + \underline{\underline{\mathbf{R}}}'\,\mathbf{X} = \mathbf{t} + \delta\mathbf{t} + \underline{\underline{\mathbf{\Delta R}}}\,\underline{\underline{\mathbf{R}}}\,\mathbf{X} \tag{2}$$

RAPiD relies on the hypothesis that the variation of pose within two subsequent frames is small so that $\underline{\underline{\mathbf{R}}}'$ can be linearised and written as:

$$\underline{\underline{\mathbf{R}}}' \quad \approx \left(\underline{\underline{\mathbf{I}}} + [\boldsymbol{\delta\theta}]_{\text{x}}\right)\underline{\underline{\mathbf{R}}}\ , \quad \text{with} \quad [\boldsymbol{\delta\theta}]_{\text{x}} = \begin{bmatrix} 0 & -\delta\theta_z & \delta\theta_y \\ \delta\theta_z & 0 & -\delta\theta_x \\ -\delta\theta_y & \delta\theta_x & 0 \end{bmatrix} \tag{3}$$

Note that $\underline{\underline{\mathbf{I}}}$ is the 3×3 identity matrix and $[\boldsymbol{\delta\theta}]_{\text{x}}$ is the cross-product matrix of the Euler attitude angle variations $\boldsymbol{\delta\theta} = [\delta\theta_x,\ \delta\theta_y,\ \delta\theta_z]$. Eq.(2) then becomes:

$$\mathbf{M}' \quad = \mathbf{t} + \delta\mathbf{t} + \left(\underline{\underline{\mathbf{I}}} + [\boldsymbol{\delta\theta}]_{\text{x}}\right)\ \underline{\underline{\mathbf{R}}}\,\mathbf{X} = \mathbf{M} + \delta\mathbf{t} + [\boldsymbol{\delta\theta}]_{\text{x}}\,\underline{\underline{\mathbf{R}}}\,\mathbf{X} \tag{4}$$

Using the following notation:

$$\begin{gathered} \mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T, \quad \delta\mathbf{t} = \begin{bmatrix} \delta t_x & \delta t_y & \delta t_z \end{bmatrix}^T \\ \underline{\underline{\mathbf{R}}}\mathbf{X} = \mathbf{P} = \begin{bmatrix} P_x & P_y & P_z \end{bmatrix}^T \end{gathered} \tag{5}$$

and, assuming to rely on a pin-hole calibrated camera having projection matrix $\mathbf{K}$ equal to:

$$\mathbf{K} = \begin{bmatrix} \alpha_u & 0 & u_o \\ 0 & \alpha_v & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

the point can be projected in the image RF using the well known formula for the projection of a generic point $\mathbf{M} = [M_x,\ M_y\ , M_z]^T$ expressed in camera RF:

$$\mathbf{m} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha_u \dfrac{M_x}{M_z} + u_0 \\ \alpha_v \dfrac{M_y}{M_z} + v_0 \end{bmatrix} = \begin{bmatrix} \alpha_u \dfrac{t_x + P_x}{t_z + P_z} + u_0 \\ \alpha_v \dfrac{t_y + P_y}{t_z + P_z} + v_0 \end{bmatrix} \tag{7}$$

Since the components of $\delta\mathbf{t}$ and $[\boldsymbol{\delta\theta}]_{\text{x}}$ are assumed to be infinitesimal, u' and v' can be manipulated in order to retain only terms up to first order, such that the projected point $\mathbf{m}'$ can be written as a linear function of the unknown parameter vector $\delta\mathbf{p} = [\delta t_x, \delta t_y, \delta t_z, \delta\theta_x, \delta\theta_y, \delta\theta_z]^T$:

$$\mathbf{m}' = \mathbf{m} + \underline{\underline{\mathbf{C}}}\ \delta\mathbf{p} \tag{8}$$

where $\underline{\underline{\mathbf{C}}}$ is a 2×6 coefficient matrix which is a function of the known pose at instant $K-1$ ($\mathbf{t}$, $\underline{\underline{\mathbf{R}}}$) and of the coordinates of the point $\mathbf{X}$ in target RF. At this point, it is possible to write the cost function that has to be minimised with respect to the parameter vector $\delta\mathbf{p}$. The function represents the square distance, projected on the normal vector, between a projected point $\mathbf{m}_i'$ and the corresponding measured point $\mathbf{m}_i^{meas}$. Note that $\mathbf{m}_i^{meas}$ is the point on the captured image at instant K that has been matched to $\mathbf{m}_i'$ through the mono-directional search along the normal $\mathbf{n}_i$.

$$f_i(\delta\mathbf{p}) \quad = \left[\mathbf{n}_i^T\left(\mathbf{m}_i' - \mathbf{m}_i^{meas}\right)\right]^2 = \left[\mathbf{n}_i^T\left(\mathbf{m}_i - \mathbf{m}_i^{meas}\right) + \mathbf{n}_i^T\underline{\underline{\mathbf{C}}}_i\delta\mathbf{p}\right]^2 \tag{9}$$

By defining the measured perpendicular distance $l_i^{meas} = \mathbf{n}_i^T\left(\mathbf{m}_i - \mathbf{m}_i^{meas}\right)$, the function $f_i(\delta\mathbf{p})$ can be rearranged. Note that l_i^{meas} is the distance between the control point and its matched image point, projected on the normal vector corresponding to the control point. Computing the summation of the cost function f_i of all the control points i ($i \in [1, m]$), the solution of the minimisation will be:

$$\delta\mathbf{p} \quad = \arg\min_{\delta\mathbf{p}} \sum_i \left(\mathbf{n}_i^T \underline{\underline{\mathbf{C}}}_i \delta\mathbf{p} - l_i^{meas}\right)^2 = \arg\min_{\delta\mathbf{p}} \sum_i \left(\mathbf{A}_i \delta\mathbf{p} - l_i^{meas}\right)^2 \tag{10}$$

Eq.(10) correspond to the linear least-square problem $\underline{\underline{\mathbf{A}}}\delta\mathbf{p} = \mathbf{l}$, where $\underline{\underline{\mathbf{A}}}$ is the $m \times 6$ matrix whose i line is the product of the transposed normal vector $\mathbf{n}_i$ with the coefficient matrix $\underline{\underline{\mathbf{C}}}_i$, and $\mathbf{l}$ is the constant term vector whose i element is equal to l_i.

B. Non-linear optimization: construction of the cost function

In rendezvous with a non-cooperative S/C, the target variation of pose from one frame to the other may be too great to enable linearisation, and the tracking could be rapidly lost. A space debris may be in tumbling, with a total rotation rate that can exceed $4\ deg/s$. In this section we provide an analytical formulation of our estimation method based on the optimisation of a non-linear cost function. Given a fitting non linear function $\hat{y}(\mathbf{p})$ of a vector of n parameters $\mathbf{p}$, 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 $\hat{y}(\mathbf{p})$. The resulting scalar cost function is:

$$\chi^2(\mathbf{p}) \quad = \sum_i^m (y_i - \hat{y}_i(\mathbf{p}))^2 = (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{p}))^T (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{p})) \tag{11}$$

The set of measurement data $\mathbf{y}$ 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 $\mathbf{X}_i$, the 3D point in target RF) along the projected normal $\mathbf{n}_i = [n_{u_i}, n_{v_i}]^T$.

$$y_i = n_{u_i} u_i + n_{v_i} v_i, \quad \mathbf{y} = [y_1, y_2, ..., y_i, ..., y_m]^T \tag{12}$$

The rotation will be described using quaternions, as quaternions have the lowest dimensionality possible for a globally non-singular representation of the rotational group $SO(3)$ [16]. The navigation filter described in III-D also relies on quaternion for the attitude estimation problem.

Let's remind the definition of the vector $\mathbf{t} = {}^{cam}\mathbf{t}_{cam-tg} = [t_x, t_y, t_z]^T$, which is the translation vector from camera RF to target RF, and introduce the quaternion q corresponding to the matrix $\underline{\underline{\mathbf{R}}}$ used in section III-A ($q = q_{cam-tg}$: ${}^{cam}\mathbf{X} = q_{cam-tg} \otimes {}^{tg}\mathbf{X} \otimes q^*_{cam-tg}$). The set of parameters to be estimated will be $\mathbf{p} = [t_x, t_y, t_z, q_0, q_1, q_2, q_3]$, which has size $n = 7$, since the representation of rotation through quaternions employs 4 variables on which the constraint of unit norm must be added. For each matched control point $\mathbf{X}_i$ in target RF, the corresponding point in camera RF will be:

$$\mathbf{M}_i \quad = \mathbf{t} + q \otimes \mathbf{X}_i \otimes q^* \tag{13}$$

Introducing the notation $Q_j(\mathbf{X}_i, q)$ (with $j = 1:3$) in order to refer to the first, the second and the third component of the vector resulting from the rotation described by the function $Q(\mathbf{X}_i, q) = q \otimes \mathbf{X}_i \otimes q^*$, the projection in the image frame of point $\mathbf{M}_i$ can be written, analogously to Eq.(7), as:

$$\begin{aligned} \hat{y}_i \quad &= \mathbf{n}_i^T \begin{bmatrix} \hat{u}_i(\mathbf{X}_i, \mathbf{p}) \\ \hat{v}_i(\mathbf{X}_i, \mathbf{p}) \end{bmatrix} \\ &= n_{u_i}\left(\alpha_u \frac{t_x + Q_1(\mathbf{X}_i, q)}{t_z + Q_3(\mathbf{X}_i, q)} + u_0\right) + n_{v_i}\left(\alpha_v \frac{t_x + Q_2(\mathbf{X}_i, q)}{t_z + Q_3(\mathbf{X}_i, q)} + v_0\right) \end{aligned} \tag{14}$$

and the non linear function will be $\hat{\mathbf{y}}(\mathbf{p}) = [\hat{y}_1(\mathbf{p}), \hat{y}_2(\mathbf{p}), ..., \hat{y}_i(\mathbf{p}), ..., \hat{y}_m(\mathbf{p})]^T$. The non-linear optimisation is solved using the Levenberg-Marquardt (LM) algorithm. The Jacobian of the nonlinear functions $\hat{\mathbf{y}}(\mathbf{p})$ has to be derived in order to compute at each iteration the update of parameter $\mathbf{p}$. For each matched model point $\mathbf{X}_i$, corresponding to the measured point y_i and the fitting function $\hat{y}_i(\mathbf{X}_i, \mathbf{p})$, the Jacobian $\mathbb{J}_i$ will be the 1×7 vector:

$$\mathbb{J}_i = \frac{\partial \hat{y}_i}{\partial \mathbf{p}} = \begin{bmatrix} n_{u_i} \frac{\partial \hat{u}_i}{\partial t_x} + n_{v_i} \frac{\partial \hat{v}_i}{\partial t_x}, & \dots, & n_{u_i} \frac{\partial \hat{u}_i}{\partial q_3} + n_{v_i} \frac{\partial \hat{v}_i}{\partial q_3} \end{bmatrix} \tag{15}$$

The computation of the elements of the Jacobian with respect to the translational terms is straightforward, while it is more complex for the elements derived with respect to the quaternion. The analytical Jacobian of a rotation of a vector $\mathbf{X}_i = [x_i, y_i, z_i]$ with respect to a quaternion q is:

$$\frac{\partial(q \otimes \mathbf{X}_i \otimes q^*)}{\partial q} = \begin{bmatrix} \frac{\partial Q_1(\mathbf{X}_i, q)}{\partial q} \\ \frac{\partial Q_2(\mathbf{X}_i, q)}{\partial q} \\ \frac{\partial Q_3(\mathbf{X}_i, q)}{\partial q} \end{bmatrix} = \begin{bmatrix} A & D & C & -B \\ B & -C & D & A \\ C & B & -A & D \end{bmatrix}, \quad \text{with} \begin{cases} A = & 2(q_0 x_i - q_3 y_i + q_2 z_i) \\ B = & 2(q_3 x_i + q_0 y_i - q_1 z_i) \\ C = & 2(-q_2 x_i + q_1 y_i + q_0 z_i) \\ D = & 2(q_1 x_i + q_2 y_i + q_3 z_i) \end{cases} \tag{16}$$

The terms of the Jacobian matrix in Eq.(15) for the $\hat{u}_i$ and $\hat{v}_i$ components are therefore:

$$\begin{aligned} &\tfrac{\partial \hat{u}_i(\mathbf{X}_i,\mathbf{p})}{\partial t_x} = \alpha_u \tfrac{1}{t_z + Q_3(\mathbf{X}_i, q)} && \tfrac{\partial \hat{v}_i(\mathbf{X}_i,\mathbf{p})}{\partial t_x} = 0 \\ &\tfrac{\partial \hat{u}_i(\mathbf{X}_i,\mathbf{p})}{\partial t_y} = 0 && \tfrac{\partial \hat{v}_i(\mathbf{X}_i,\mathbf{p})}{\partial t_y} = \alpha_v \tfrac{1}{t_z + Q_3(\mathbf{X}_i, q)} \\ &\tfrac{\partial \hat{u}_i(\mathbf{X}_i,\mathbf{p})}{\partial t_z} = -\alpha_u \tfrac{t_x + Q_1(\mathbf{X}_i, q)}{t_z + Q_3((\mathbf{X}_i, q))^2} && \tfrac{\partial \hat{v}_i(\mathbf{X}_i,\mathbf{p})}{\partial t_z} = -\alpha_v \tfrac{t_y + Q_2(\mathbf{X}_i, q)}{t_z + Q_3((\mathbf{X}_i, q))^2} \\ &\tfrac{\partial \hat{u}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_0} = \alpha_u \tfrac{A(t_z + Q_3(\mathbf{X}_i, q)) - C(t_x + Q_1(\mathbf{X}_i, q))}{(t_z + Q_3(\mathbf{X}_i, q))^2} && \tfrac{\partial \hat{v}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_0} = \alpha_v \tfrac{B(t_z + Q_3(\mathbf{X}_i, q)) - C(t_y + Q_2(\mathbf{X}_i, q))}{(t_z + Q_3(\mathbf{X}_i, q))^2} \\ &\tfrac{\partial \hat{u}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_1} = \alpha_u \tfrac{D(t_z + Q_3(\mathbf{X}_i, q)) - B(t_x + Q_1(\mathbf{X}_i, q))}{(t_z + Q_3(\mathbf{X}_i, q))^2} && \tfrac{\partial \hat{v}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_1} = \alpha_v \tfrac{-C(t_z + Q_3(\mathbf{X}_i, q)) - B(t_y + Q_2(\mathbf{X}_i, q))}{(t_z + Q_3(\mathbf{X}_i, q))^2} \\ &\tfrac{\partial \hat{u}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_2} = \alpha_u \tfrac{C(t_z + Q_3(\mathbf{X}_i, q)) + A(t_x + Q_1(\mathbf{X}_i, q))}{(t_z + Q_3(\mathbf{X}_i, q))^2} && \tfrac{\partial \hat{v}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_2} = \alpha_v \tfrac{D(t_z + Q_3(\mathbf{X}_i, q)) + A(t_y + Q_2(\mathbf{X}_i, q))}{(t_z + Q_3(\mathbf{X}_i, q))^2} \\ &\tfrac{\partial \hat{u}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_3} = \alpha_u \tfrac{-B(t_z + Q_3(\mathbf{X}_i, q)) - D(t_x + Q_1(\mathbf{X}_i,))}{(t_z + Q_3(\mathbf{X}_i, q))^2} && \tfrac{\partial \hat{v}_i(\mathbf{X}_i,\mathbf{p})}{\partial q_3} = \alpha_v \tfrac{A(t_z + Q_3(\mathbf{X}_i, q)) - D(t_y + Q_2(\mathbf{X}_i, q))}{(t_z + Q_3(\mathbf{X}_i, q))^2} \end{aligned} \tag{17}$$

The obtained Jacobian is used at each iteration to analytically compute, according to the version of LM algorithm proposed in Ref. [17], the direction of the parameter increment, starting from the initial guess $\mathbf{p}_0$ which is taken to be equal to the estimated pose at instant $K-1$. In this optimisation process, some approximations are introduced: at each update of the pose 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 algorithm shows very good performance even if in presence of this approximation, as it will be illustrated in Sec. IV.

C. Reducing the computational load

One of the major drawbacks of applying the Canny edge extraction on the greyscale images is that the algorithm detects

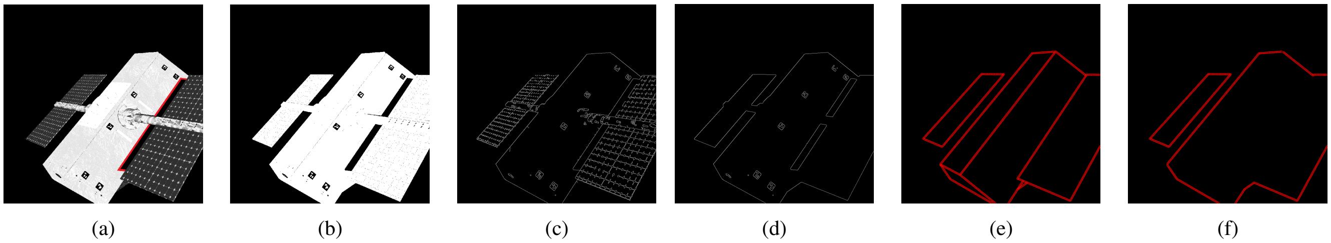


Fig. 1. 1(a) Greyscale image. 1(b) Thresholded image. 1(c) Canny edge extraction on greyscale image. 1(d) Canny edge extraction on thresholded image. 1(e) Projection of all visible geometrical edges. 1(f) Projection of the external perimeter edges.

not only geometrical edges, but also texture discontinuity, which are not managed in the masking algorithm. Moreover, the direction of the light could make invisible some geometrical edge due to the particular reflective texture of the MLI (Multi Layer Insulation), which usually covers S/Cs. Finally, shadows contours are identified by the Canny edge extractor as edges (e.g., as it is possible to see for the shadow contour highlighted in red in Fig.1(a), which appears in the output image of the Canny edge detector in Fig.1(c)). 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 (i.e., as done in Ref. [11]) or shadows, the most robust solution to reduce the number of false matching is to rely only on the external contour of the target (i.e., the silhouette). If the target is artificially illuminated by the chaser, or if there are proper sun illumination conditions (see [18]), the silhouette contour of the target can be retrieved applying threshold and morphological operators on the greyscale image before the Canny edge extraction. This has been done for the image shown in Fig.1(b), whose Canny edge extraction output is shown in Fig.1(d). Although the study presented in this paper relies only in the use of a visible (VIS) camera, the coupling with an infra-red (IR) thermal camera would provide benefits to the silhouette extraction, increasing the robustness towards illumination conditions, shadows and reflective materials. Moreover, the use of multi-spectral imagery could help the segmentation process in presence of the Earth in the background. As only the silhouette contours are now extracted, a new masking algorithm capable of detecting only the external perimeter of the target has been implemented. As for the former algorithm, the *a priori* model is projected in the image frame in the estimated pose of instant K. Then each surface is meshed monodimensionally along its perimeter obtaining a set of possible control points. All the points whose projection in the plane $x-y$ is inside the projected perimeter of any other surface are discarded, since they certainly do not belong to the external projected perimeter of the satellite. This allows to avoid the two dimensional mesh of the surfaces that was done by the former method, resulting in a faster algorithm. For the pose corresponding to Fig.1(a), the computational time of the new algorithm (Fig.1(f)) is decreased by a factor of 5 with respect to the latency time of former masking algorithm (Fig.1(e)). This big difference in the computational time allows the user to employ a more complex 3D model of the target, which models more details thus limiting the sources of false matching and outliers. The new masking algorithm applied to the thresholded image, together with a proper choice of d_{toll}, has proven to generate fewer mismatches than the former technique, thus allowing not to consider outliers rejection algorithms such as RANSAC proposed by Ref. [19] or the IRLS technique proposed by Ref. [5], [9], and therefore providing a fast estimation.

D. Integration of the measurements in the navigation filter

The tracking algorithm discussed in the previous sections provide a measurement of the relative pose target-chaser, 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 high computational load associated to operation such as image segmentation, the 3D model projection, and the non-linear optimisation. There will be therefore a delay between the time of acquisition (i.e., image capture 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 appropriate technique to merge delayed and infrequent measurements. In this section, the dynamic model implemented in the navigation filter, as well as the applied delay management technique, are briefly described. The complete formalisation of the method and its application to the space RDV navigation problem are discussed in Ref. [20].

The translational motion is modelled according to the well known Clohessy-Wiltshire-Hill's equations [18], a system of linear differential equations that describes the relative motion of the chaser with respect to the target. The motion is 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):

$$\begin{cases} \ddot{x} - 3\omega^2 x - 2\omega\dot{y} & = \gamma_x \\ \ddot{y} + 2\omega\dot{x} & = \gamma_y \\ \ddot{z} + \omega^2 z & = \gamma_z \end{cases} \qquad (18)$$

where ω is target's orbit angular rate and $\gamma_{x,y,z}$ are the control accelerations acting on chaser centre of mass (COM). The state transition matrix associated to the system in Eq.18 can be analytically computed, thus the relative translational dynamics can be written in the form of a linear time-discrete system. The rotational dynamics is modelled according to the non-linear prediction model:

$$\begin{cases} \dot{q}_{i-tg} & = \frac{1}{2} q_{i-tg} \otimes \begin{bmatrix} 0 \\ {}^{tg}\omega_{i-tg} \end{bmatrix} \\ {}^{tg}\dot{\omega}_{i-tg} & = -I_{tg}^{-1} \left({}^{tg}\omega_{i-tg} \times I_{tg} \, {}^{tg}\omega_{i-tg} \right) \end{cases} \qquad (19)$$

where q_{i-tg} is the attitude quaternion from inertial to target RF, ${}^{tg}\omega_{i-tg}$ 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 [21]. 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. 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, [22], [23]). The analysis in Ref. [20] 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.

IV. Performance analysis

The proposed non-linear tracking algorithm is tested in order to characterise the performances at different distances in two different conditions. In a first set of simulation, the estimation chain is tested using perfect silhouette images. This excludes illumination conditions, shadows and reflective textures from the possible sources of false matching. Nevertheless, mismatching is still possible. This could be the case when the S/C, rotating, passes from a configuration (instant $K-1$) in which a geometrical edge is visible, to a configuration (instant K) in which that edge becomes invisible (or vice versa). This phenomenon is known as self-occlusion. In such a case, the edge will be present in the *a priori* projected model, but will be invisible in the Canny image (or vice versa), increasing the probability to match the control points with the wrong edge in the image. This make the optimization algorithm converge to a local minimum which does not represent the real pose of the target. Usually, model-based recursive algorithms cannot recover autonomously from these local minima and the tracking diverges. Even if our algorithm has shown good convergence property with respect to RAPiD, there are some configurations in which the tracking is lost (e.g., when the faces of the solar arrays become visible or invisible). For this reason, the coupling with the KF becomes crucial: the dynamic filter propagates a physical model which predicts the motion of the target and makes edges appear and disappear according to the prediction model. In all the simulated scenarios, the images are acquired at a rate of 1 Hz. A latency time of 1 second for the IP-CV algorithm is assumed. Therefore, the measurements become available by the filter after a delay of 1 second, which corresponds to a delay of 10 time step, assuming a filter run frequency of 10 Hz.

In the first set of simulation, target and chaser are at a fixed distance and do not translate with respect to the inertial frame. Chaser's attitude is fixed with the camera pointing to the target, while the target is rotating according to the angular momentum equation under the effect of its initial conditions. The algorithm has been tested at a distance of 30 m, 20 m and 10 m, with a camera having a 30 deg FOV and a size of 1024×1024 pixels. The target has a size of $4 \times 3 \times 1$ meters. These dimensions, translated in pixel, correspond, in the image frame, to an object size of $255 \times 191 \times 64$ pixels at 30 m, $382 \times 287 \times 96$ pixels at 20 m, and $764 \times 573 \times 191$ pixels at 10 m. The 3D model projected by the masking algorithm is simplified and has fewer details than the CAD model used to generate the images, which on the other hand accurately reproduces the geometry of the S/C. For each distance, 100 simulations of 200 seconds have been run, varying on the initial conditions of target attitude and rotation rate. The initial relative attitude quaternions have been uniformly sampled using the algorithm presented in Ref. [24], while target rotation rate components are random variables uniformly distributed in the interval $[-1deg/s, +1deg/s]$. Table I shows the performance averaged on the 100 simulations. For the attitude estimation, the error is given in the axis-angle representation, which provides a positive scalar representation of the error. The mean and the root-mean-square (RMS) are shown both for the error of the measured relative quaternion ($\delta\theta_{meas}$) and for the error of estimated relative quaternion ($\delta\theta_{est}$). Note that the mean of the error is lower for the measured quaternion, while the root-mean-square error is lower for the estimated quaternion. This is due to the fact that the KF is calibrated in order to be robust towards divergent measurements, therefore giving more confidence to the prediction model than to the measurements: when the measurements are good, the estimation is less precise than the measurement itself, but when measurement start diverging, the estimation brings it back to the real pose. The performance of the quaternion estimation can be increased by using a time-varying calibration of the KF that computes in real-time the value of the measurement noise covariance R. Nevertheless, it may be particularly difficult to have a good characterisation of R because the measurement noise depends on many factors, such as the intrinsic noise

TABLE I
PERFORMANCE OF THE NON-LINEAR TRACKING ALGORITHM AT DIFFERENT DISTANCES

	First set of simulation (perfect silhouette)						Second set of simulation (high fidelity RDV simulator)					
	$30\ m$		$20\ m$		$10\ m$		$30\ m$		$20\ m$		$10\ m$	
	mean	RMS	mean	RMS	mean	RMS	mean	RMS	mean	RMS	mean	RMS
$\delta\theta_{meas}$ $[deg]$	1,91	3,34	1,23	2,22	0,80	1,47	15,78	20,11	3,53	4,84	1,73	2,51
$\delta\theta_{est}$ $[deg]$	2,13	3,20	1,34	2,01	0,83	1,25	16,01	20,42	3,81	5,02	1,79	2,42
	σ		σ		σ		σ		σ		σ	
$\delta\omega_{x\,est}$ $[deg/s]$	0,065		0,047		0,023		0,152		0,073		0,046	
$\delta\omega_{y\,est}$ $[deg/s]$	0,169		0,115		0,057		0,274		0,118		0,067	
$\delta\omega_{z\,est}$ $[deg/s]$	0,141		0,087		0,044		0,250		0,107		0,063	

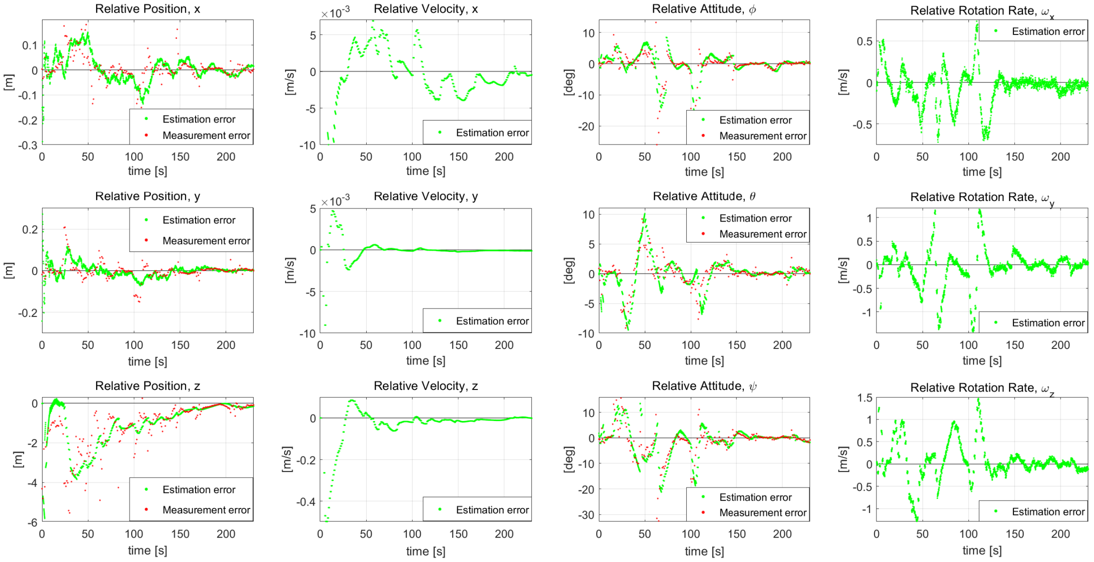
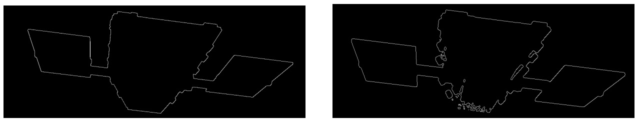


Fig. 2. Estimation and measurements error for an *R-bar* approach from 45 m to 5 m

of the sensor, the relative distance camera-target, the relative rotation rate and velocity, the camera capture rate, the illumination conditions and even the target relative pose itself. For the relative target/chaser rotation rate estimation error ($[\delta\omega_{x\,est}, \delta\omega_{y\,est}, \delta\omega_{z\,est}]$, expressed in chaser RF), the standard deviation σ is shown. The performance of the method in this first set of simulations is the intrinsic performance of the algorithm, as no external sources of error are present. Thus, it must be considered as a benchmark to compare the results of the second set of simulations.



(a) First set of simulation (b) Second set of simulation

Fig. 3. Comparison between perfect (3(a)) and noisy (3(b)) silhouettes

The second set of simulation is run on a full GNC rendezvous simulator modelling the LEO environment, with orbital perturbations affecting both chaser's and target's dynamics. The simulations are performed in closed-loop, meaning that the estimated pose is used to control camera pointing and chaser position, thus adding a source of error. The chaser is equipped with a spot light illuminating the target. The images are simulated using Thales Alenia Space high fidelity image simulator SpiCam. The images are affected by shadows, reflections and blur, so that the silhouette retrieved by the Canny edge extraction is highly affected by noise. Fig. 3 shows, for the same relative pose, the difference between the perfect silhouette used in the first set of simulation and the noisy one used in the second set of simulation. 100 simulation of 200 seconds are run, at a distance of 30 m, 20 m, and 10 m, reproducing the same profile of relative attitudes and rotation rates of the first set of simulations. The performance is shown in Table I and can be directly compared to the benchmark performance of the simulations with the perfect silhouette. A degradation of the performance occurs: at 20 m the method has diverged in 5 runs, and at 30 m in 20 runs. The value shown in Table I are averaged on the simulations in which the tracking was ensured. The high degradation of the performance at a distance of 30 m is due to the reduced size of the S/C in the image frame. As the distance increases, the number of control points used to compute the cost function decreases: this increases the relative weight of a mismatching with the noisy silhouette. To overcome this degradation of the performance, different solution can be envisaged: first, a dynamic calibration of the KF, giving less confidence to the pose measurements when they become less reliable; secondly, the coupling of the VIS camera with an IR camera to enhance the silhouette extraction; lastly, the chaser could be equipped with more cameras having different FOV in order to rely, during each RDV phases, on an optimised sensor.

The estimation chain has been tested in a final approach scenario, with the chaser approaching the target along the radial Earth-target (i.e. R-bar [18]). This allows to characterise the performance of the tracking algorithm also with respect to the relative translational dynamics estimation. Note that the model propagated in the translational KF is linearized with respect to the true orbital dynamics. In a simulation time of 230 seconds the chaser goes from a relative distance of 40 m to 5 m, moving at a relative speed of 0.15 m/s. The target rotation rate is 3 deg/s around each axis. Fig. 2 shows the errors in the position, velocity, attitude (in Euler angles [ϕ, θ, ψ]), and rotation rate estimation. The errors are expressed in camera RF, and this explains why the translational error along z axis, which is camera optical axis, is an order of magnitude greater than the errors along x and y axis. Predictably, the performance increases as the distance decreases. This is compatible with the requirement for RDV operations, since an increasing precision is required as the chaser gets closer to the target.

V. Conclusions

The current paper has addressed the problem of monocular model-based pose estimation for space rendezvous, proposing an innovative method relying on silhouette-edge extraction and tracking. The method has been formally developed and implemented. The coupling with a dynamic filter enables the robust estimation of the relative pose, as well as the estimation of the translational velocity and rotation rate, of the target. The navigation filter integrates a delay management technique allowing robust tracking even in presence of delayed and infrequent measurements. The method has shown high performance and robustness when using perfect silhouette images, and suffered a loss of robustness at high distances when tested in closed-loop on realistic images affected by shadows, blur and reflections. Different solutions to strengthen the estimation have been proposed, such as the coupling of the visible camera with a thermal infra-red camera in order to improve the silhouette extraction. The proposed estimation chain provides a cost effective solution with a relatively low computational load, thus compatible with typical space processing capabilities.

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] 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.

[7] V. Lepetit and P. Fua, "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, https://doi.org/10.1561/0600000001.

[8] 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.

[9] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 932–946, 2002, https://doi.org/10.1109/TPAMI.2002.1017620.

[10] R. Evans, "Kalman filtering of pose estimates in applications of the rapid video rate tracker." in *BMVC*, 1990, pp. 1–6.

[11] 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, https://doi.org/ 10.1109/IROS.2012.6386083.

[12] ——, "Vision-based space autonomous rendezvous: A case study," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 619–624, https://doi.org/10.1109/IROS.2011.6094568.

[13] ——, "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.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] M. A. A. Zisserman, "Robust object tracking," in *Proceedings of the Asian Conference on Computer Vision, Singapore*, 1995, pp. 5–8.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] J. J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 3993–3998, doi: https://doi.org/10.1109/ROBOT.2004.1308895.