ORB-SLAM Applied to Spacecraft Non-Cooperative Rendezvous

Mehregan Dor* and Panagiotis Tsiotras†
Georgia Institute of Technology, Atlanta, GA, 30313, USA

The success of SLAM-based algorithms developed in the ground robotics community motivates their study and application in the domain of autonomous space robotics, especially for relative navigation problems with respect to a non-cooperative space object. In this paper, the application of ORB-SLAM to the non-cooperative rendezvous problem is studied, by establishing the essential definitions, by detailing the algorithm's operation and by identifying the required modifications to reliably implement SLAM solutions to space applications. The traditional ORB-SLAM algorithm demonstrates high robustness and low computation time, by exploiting relatively inexpensive ORB features in a sliding window approach. The algorithm is tested on a sequence of images taken during the rendezvous phase of the Hubble Space Telescope Servicing Mission, as well as using realistic experimental data produced in the ASTROS laboratory environment at Georgia Tech. The results establish the need for further development and specialization, but also showing great potential for use in future space robotics applications.

I. Introduction and Related Work

We consider the problem of relative navigation between two satellites in different orbits around a central celestial body, such as the Earth. Let the first of these satellites be called the chaser, capable of maneuvering about the second satellite, called the target. In the non-cooperative rendezvous scenario, it is assumed that no prior information about the states of the target nor of the relative states is known, and that there is a total lack of direct exchange of information between the target and the chaser satellites during the rendezvous, be it regarding the absolute states of the target, the absolute states of the chaser or the relative states between the two. Furthermore, we make no assumption regarding the motion of the target. In particular, the target satellite may not be passive, but its rotational motion can be actively controlled.

As it circumnavigates the target, the chaser satellite gathers measurements using on-board sensors and gradually builds an estimate of the target's states, provided the latter are observable. Typical means of relative pose measurements include laser, flash LIDAR or radar range-finders and cameras in various wavelength bands, appropriate for proximity operations. Similarly, global information may be provided (when available) by global positioning systems, star trackers and inertial measurement units.

In the context of spacecraft proximity operations in past and current space missions, the severe lack of computational resources has led some authors[1] to consider filtering as the only adapted space-bound navigation solution, generally exploiting range-based measurements[2] producing a point cloud. However, given the ever growing on-board computer resources available in space-bound missions and the possible emergence of space-grade GPU's, the vision-based bundle-adjustment approach can soon enough be implemented as a viable solution for autonomous relative navigation in unmanned space missions, such as in-orbit satellite servicing or in-orbit large structure assembly. Eventually, such a system would enable closed-loop control of the chaser's motion around the target, intended for specific maneuvers, such as grappling and docking.

Augenstein[3] proposed the use of monocular SLAM for the purpose of estimating the target pose by exploiting a Gaussian driven process to model rotation estimation within a Rao-Blackwellized particle filter. Later on, Tweddle[4] established that Simultaneous Localization and Mapping (SLAM) can be used to estimate the

*PhD Student, Department of Aerospace Engineering, Atlanta, USA.
†Dean's Professor, Department of Aerospace Engineering, Atlanta, USA, AIAA Fellow

linear and angular velocities, as well as the position of the center of mass and the diagonal inertia matrix (up to a scale) of a non-cooperative spinning target satellite using stereoscopy, under a torque-free motion assumption. In Tweddle's solution, an approach employing smoothing (also known as bundle adjustment[5]) is favored over one that uses filtering. Tweddle argues that a filtering scheme can converge to a local minimum. In contrast, in a smoothing scheme, initially estimating and converging to the positions, orientations and velocities (linear and angular) of the target will allow for a subsequent estimation of dynamical properties which is richer in structure and information, thus avoiding convergence to local minima. The added certainty regarding convergence comes at an additional computational cost, since the smoothing problem is much larger in scale. Tweddle employs iSAM,[6] which is based on fast incremental factorizations of the naturally sparse smoothing problem information matrix.

Multiple open-source SLAM algorithms are now readily available in the community, as demonstrated by the OpenSLAM online compendium.[7] As a recent, efficient and robust open-source rendition of a bundle adjustment-based localization and mapping algorithm, ORB-SLAM[8] is promising in its applicability to the vision-based rendezvous problem. The algorithm exploits ORB features,[9] which are rotation-invariant image features based on BRIEF binary descriptors. Indeed, ORB features can be rapidly extracted, beating the calculation time of other popular image features, such as SIFT, by several orders of magnitude. This allows for a large number of features to be extracted. By implementing a generous initialization of image keypoints, paired with a harsh culling policy of weaker image keypoints, ORB-SLAM demonstrates higher robustness. In addition, ORB-SLAM implements a bag-of-words method for place recognition, which automates the loop-closure step. Indeed, when a recognizable scene is viewed for a second time after many image frames have passed, the algorithm carries-out a full bundle adjustment. This process greatly reduces the error in the estimation by eliminating, through inference, drifts that may have accumulated between successive keyframes.

In this paper, the applicability of ORB-SLAM to the problem of non-cooperative rendezvous is further explored and tested on the relative pose estimation problem of the Hubble Space Telescope (HST) during a close-up maneuver, as well as on experimental data produced at the 5-DOF experimental platform for Autonomous Spacecraft Testing for Robotics Operations in Space (ASTROS) of the Dyanmics and Control Systems Laboratory (DCSL) at the School of Aerospace Engineering of the Georgia Institute of Technology.

II. Non-Cooperative Rendezvous Pose Estimation

A. Relative Navigation in Orbit

For notation purposes, let $(R_{\mathcal{B}/\mathcal{A}}, t^{\mathcal{B}}_{\mathcal{B}/\mathcal{A}}) \in \mathrm{SE}(3)$ be the rotation matrix and translation vector pair transforming a homogeneous vector $\underline{x}^{\mathcal{A}} \in \mathbb{P}^3$ (here $\mathbb{P}^n = \mathbb{R}_{\neq 0} \times \mathbb{R}^n$ is the $(n+1)$ dimensional projective space corresponding to the n-dimensional space $\mathbb{R}^n$) whose coordinates are expressed in the $\mathcal{A}$ frame to a homogeneous vector $\underline{x}^{\mathcal{B}} \in \mathbb{P}^3$ whose coordinates are expressed in the $\mathcal{B}$ frame, such that

$$\underline{x}^{\mathcal{B}} = \begin{bmatrix} R_{\mathcal{A}/\mathcal{B}} & t^{\mathcal{B}}_{\mathcal{A}/\mathcal{B}} \\ 0 & 1 \end{bmatrix} \underline{x}^{\mathcal{A}}. \tag{1}$$

Equivalently, we denote $T^{\mathcal{B}}_{\mathcal{A}/\mathcal{B}} \in \mathrm{SE}(3)$ to be the pose of frame $\mathcal{A}$ with regards to frame $\mathcal{B}$, as expressed in frame $\mathcal{B}$ coordinates, given by

$$T^{\mathcal{B}}_{\mathcal{A}/\mathcal{B}} \triangleq \begin{bmatrix} R_{\mathcal{A}/\mathcal{B}} & t^{\mathcal{B}}_{\mathcal{A}/\mathcal{B}} \\ 0 & 1 \end{bmatrix}. \tag{2}$$

Let $\mathcal{E} = \{\mathrm{E}; \hat{e}_1, \hat{e}_2, \hat{e}_3\}$ be an Earth Centered Inertial (ECI) frame, where $\hat{e}_1, \hat{e}_2, \hat{e}_3 \in \mathbb{S}^2$ are three mutually orthogonal unit vectors defining the space $\mathbb{R}^3$. Next, let S, T be the centers of mass of the chaser and of the target satellites, respectively, and let $r^{\mathcal{E}}_{\mathrm{S}}, r^{\mathcal{E}}_{\mathrm{T}} \in \mathbb{R}^3$ be the position vectors of S and T expressed in $\mathcal{E}$ frame coordinates, respectively. The relative vector $r^{\mathcal{E}}$ between points S and T as expressed in the $\mathcal{E}$ frame coordinates is then given by $r^{\mathcal{E}} = r^{\mathcal{E}}_{\mathrm{T}} - r^{\mathcal{E}}_{\mathrm{S}}$.

We define the non-inertial local vertical local horizontal (LVLH) reference frame[10,11] $\mathcal{L} = \{\mathrm{S}; \hat{\ell}_1, \hat{\ell}_2, \hat{\ell}_3\}$ as

follows. Let the origin at S, having the three unit vectors $\hat{\ell}_1, \hat{\ell}_2, \hat{\ell}_3 \in \mathbb{S}^2$ oriented such that

$$\hat{\ell}_1 = \frac{\vec{r}_{\mathrm{S}}}{\|\vec{r}_{\mathrm{S}}\|}, \qquad \hat{\ell}_3 = \frac{\vec{h}_{\mathrm{S}}}{\|\vec{h}_{\mathrm{S}}\|}, \qquad \hat{\ell}_2 = \hat{\ell}_3 \times \hat{\ell}_1, \tag{3}$$

where $\vec{v}_{\mathrm{S}}$ is the orbital velocity vector of the chaser and $\vec{h}_{\mathrm{S}} = \vec{r}_{\mathrm{S}} \times \vec{v}_{\mathrm{S}}$ is the angular momentum vector of the chaser.

Let $\mathcal{S} = \{\mathrm{S}; \hat{s}_1, \hat{s}_2, \hat{s}_3\}$ be the chaser satellite fixed-body frame with origin at S and arbitrary body-fixed orientation determined by the mutually orthogonal unit vectors $\hat{s}_1, \hat{s}_2, \hat{s}_3 \in \mathbb{S}^2$. It follows that a rigid-body transformation $T^{\mathcal{L}}_{\mathcal{S}/\mathcal{L}} \in \mathrm{SE}(3)$, which is dependent on the dynamics of the chaser (and thus evolves with time), encodes the pose of frame $\mathcal{S}$ with regards to frame $\mathcal{L}$ expressed in frame $\mathcal{L}$ coordinates.

It is usual[4] to consider a frame $\mathcal{T} = \{\mathrm{T}; \hat{t}_1, \hat{t}_2, \hat{t}_3\}$ such that the unit vectors $\hat{t}_1, \hat{t}_2, \hat{t}_3 \in \mathbb{S}^2$ are aligned with the principal axes of inertia of the target. In the non-cooperative estimation problem, $T^{\mathcal{E}}_{\mathcal{T}/\mathcal{E}} \in \mathrm{SE}(3)$ is unknown, and can be recovered by feeding measurements of pose to an estimation procedure which is programmed with the dynamical model of the target.[2] Since the issue of the relative dynamics between the two satellites is not treated in this paper, the frame $\mathcal{T}$ is not the subject of any study herein. Nevertheless, it is crucial to distinguish this frame from the various geometric frames relevant to the vision-based relative navigation problem. Let $\mathcal{C} = \{\mathrm{C}; \hat{c}_1, \hat{c}_2, \hat{c}_3\}$, with $\hat{c}_1, \hat{c}_2, \hat{c}_3 \in \mathbb{S}^2$, be the sensing camera body-fixed frame centered at

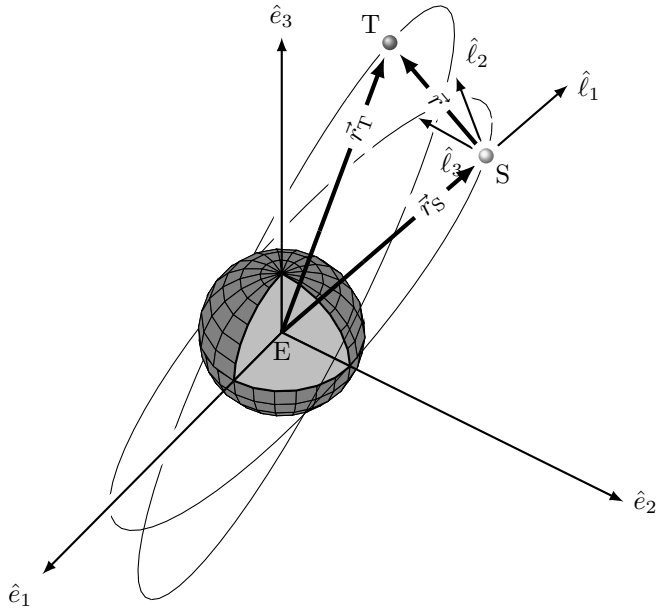

Figure 1. The Orbital Relative Navigation Problem Frame Definitions

point C, the optical center of the sensing camera, and oriented such that $\hat{c}_3$ is aligned with the camera's viewing direction, $\hat{c}_2$ points in the "downwards" direction in the image, and $\hat{c}_1 = \hat{c}_2 \times \hat{c}_3$. The pose $T^{\mathcal{S}}_{\mathcal{C}/\mathcal{S}}$ of the sensing camera with respect to the chaser fixed-body frame $\mathcal{S}$, as expressed in the $\mathcal{S}$ frame coordinates is assumed to be known. If the camera is installed on gimbals, then appropriate encoder information will provide this pose; otherwise, this transformation is constant.

In target pose estimation, the frame of interest $\mathcal{G} = \{\mathrm{G}; \hat{g}_1, \hat{g}_2, \hat{g}_3\}$ has its origin at some arbitrarily chosen point G, and is oriented by the unit vectors $\hat{g}_1, \hat{g}_2, \hat{g}_3 \in \mathbb{S}^2$,[4] and constitutes a geometric reference and coordinate system for expressing the position vectors of the components of the target satellite. Point G's location is known by the designers of the target satellite.

The ORB-SLAM algorithm will generate a map of points, whose position vectors are expressed in an arbitrarily chosen reference frame, that we shall call the geometric feature frame $\mathcal{N} = \{\mathrm{N}, \hat{n}_1, \hat{n}_2, \hat{n}_3\}$ coordinates, with $\hat{n}_1, \hat{n}_2, \hat{n}_3 \in \mathbb{S}^2$. Point N is chosen at the origin of the map, determined during automatic initialization

phase of the algorithm (see Appendix A). At initialization, it is assumed that frame $\mathcal{N}$ coincides exactly with the corresponding frame $\mathcal{C}$.

The goal of a vision-based relative navigation algorithm is then to estimate the rigid-body transformation $T^{\mathcal{S}}_{\mathcal{G}/\mathcal{S}}$ between the target-fixed geometric frame $\mathcal{G}$ and the chaser-fixed geometric frame $\mathcal{S}$, as expressed in $\mathcal{S}$ frame coordinates.

It is important to realize that ORB-SLAM in monocular mode produces only a scale ambiguous version of the transformation $T^{\mathcal{N}}_{\mathcal{C}/\mathcal{N}}$. Indeed, let C$'$ be the point representing the camera in the ORB-SLAM algorithm framework and construct frame $\mathcal{C}' = \{\mathrm{C}'; \hat{c}_1, \hat{c}_2, \hat{c}_3\}$, centered at this point, with $\hat{c}_1, \hat{c}_2, \hat{c}_3$ as previously defined. Then, $t^{\mathcal{C}'}_{\mathcal{N}/\mathcal{C}'}$ is the scale ambiguous position vector of the frame $\mathcal{N}$ with regards to the camera frame $\mathcal{C}'$ as expressed in frame $\mathcal{C}'$ coordinates, as estimated by the algorithm. Assume now that $R_{\mathcal{N}/\mathcal{C}} = R_{\mathcal{N}/\mathcal{C}'}$ and let the unknown scale $\lambda \in \mathbb{R}_{>0}$ be such that $t^{\mathcal{C}}_{\mathcal{N}/\mathcal{C}} = \lambda t^{\mathcal{C}'}_{\mathcal{N}/\mathcal{C}'}$. Then, it follows that

$$T^{\mathcal{C}}_{\mathcal{N}/\mathcal{C}} = \begin{bmatrix} R_{\mathcal{N}/\mathcal{C}'} & \lambda t^{\mathcal{C}'}_{\mathcal{N}/\mathcal{C}'} \\ 0 & 1 \end{bmatrix}. \tag{4}$$

To obtain our desired transformation, we assume that all of the points in the reconstructed map are part of the target body (in other words that they are fixed with regards to the target geometric frame $\mathcal{G}$). It follows that the pose of frame $\mathcal{G}$ with regards to frame $\mathcal{N}$ is encoded in an unknown, though constant, rigid-body transformation $T^{\mathcal{N}}_{\mathcal{G}/\mathcal{N}}$. By cascading transformations, it follows that

$$T^{\mathcal{S}}_{\mathcal{G}/\mathcal{S}} = T^{\mathcal{S}}_{\mathcal{C}/\mathcal{S}} T^{\mathcal{C}}_{\mathcal{N}/\mathcal{C}} T^{\mathcal{N}}_{\mathcal{G}/\mathcal{N}}. \tag{5}$$

In our approach, an estimate of $T^{\mathcal{C}'}_{\mathcal{N}/\mathcal{C}'}$, denoted $\tilde{T}^{\mathcal{C}'}_{\mathcal{N}/\mathcal{C}'}$, is directly obtained from ORB-SLAM. If the information about the geometric model of the target satellite is available, we use the Coherent Point Drift[12] algorithm to obtain an estimate of scale λ and of the transformation $T^{\mathcal{N}}_{\mathcal{G}/\mathcal{N}}$, by comparing the point cloud of the reconstructed map to that of the target satellite 3D model.

B. ORB-SLAM Local Bundle Adjustment and Automatic Map Initialization

It is typical to consider SLAM as a likelihood maximization problem (see Appendix A) where the unknown model parameters are the camera frame poses and the reconstructed map point coordinates, the measurement vectors are the image coordinates of feature points detected in each image frame and the measurement function is a composition of a rigid-body frame transformation of map points followed by their projection into the camera image plane. It is known that the problem reduces to a least-squares minimization problem, under an assumption of jointly Gaussian distribution of the likelihood.

For initialization of the map, two-view geometry[13] is exploited to triangulate a set of points and to calculate a relative pose between the two camera frames in which the map points are detected.[8] The algorithm successively chooses video frames until it can successfully initialize the map. Once achieved, the first frame $\mathcal{N} = \{\mathrm{N}; \hat{n}_1, \hat{n}_2, \hat{n}_3\}$ then becomes the global reference frame and the coordinate system for the remainder of the SLAM process. The second frame $\mathcal{C} = \{\mathrm{C}; \hat{c}_1, \hat{c}_2, \hat{c}_3\}$ corresponds to the pose of the camera for the subsequent video frames.

One of two models is used to evaluate the two-view geometry. The first one assumes that the scene is planar and the second one assumes that the scene is non-planar.[8] Since neither of these models can be preferred for a-priori initialization, a competition based on a fitness score is carried-out between the two possible models, to avoid a corrupted initial map.

III. Experimental Results

In this section, details and results pertaining to the validation of the proposed pose estimation approach are laid out. Firstly, the method was tested using a monocular video footage captured during proximity operations of the NASA STS-125 Servicing Mission 4 (SM4) to the Hubble Space Telescope (HST)[14] in May 2009. Secondly, to further validate the applicability of the proposed approach for non-cooperative navigation, several realistic tests were performed using the ASTROS platform and qualitative comparisons were made.

A. Hubble Space Telescope Servicing Mission Sequence

The Relative Navigation Sensor (RNS) was used in an on-orbit demonstration, storing imagery from the RNS cameras during the Rendezvous Proximity Operations and Docking (RPOD) phases of the mission.

In the footage, a relative maneuver of -90° motion about the HST +V2 axis can be seen, with the +V3 axis pointing away from the camera. The on-board algorithm GNFIR,[14] which minimizes the least square errors between detected edges and a stick model of HST, which contains arbitrarily chosen edges of the 3D HST model, confirms that the relative range varies from 97 m to 45 m during the 20 minutes and 27 seconds[14] of tracking, with a peak pose quality of 99.2%. Hence, true scale relative pose can be recovered with prior information about the geometry of the target. Initially, in our approach, to test a completely non-cooperative navigation in the current study, no model was used.

The sequence of 4,187 RGB images of the sequence, each of size $1,000 \times 1,000$ pixels, were fed to the ORB-SLAM algorithm, which successfully produced a map of points pertaining to the surface of HST, a set of keyframes $\mathcal{K}$ that form the graph of the bundle adjustment optimization problem and, for each video frame $i = 1, \ldots, 4187$, a scale ambiguous transformation (or relative pose) estimate of the geometric feature frame $\mathcal{N}$ with respect to the camera frame $\mathcal{C}'$ expressed in frame $\mathcal{C}'$ coordinates, herein designated as

$$\tilde{T}^{\mathcal{C}'}_{\mathcal{N}/\mathcal{C}'} = (\tilde{R}_{\mathcal{N}/\mathcal{C}'}, \tilde{t}^{\mathcal{C}'}_{\mathcal{N}/\mathcal{C}'}) \in \mathrm{SE}(3) \tag{6}$$

The harsh lighting conditions experienced in the visible spectrum in space can be challenging for a successful detection phase of a visual feature based method like ORB-SLAM. However, camera gain control and appropriate pre-processing operations, akin to contrast-limited adaptive histogram equalization (CLAHE), can mitigate these effects, as shown in Figure 2. The intrinsic camera parameters that were used for this

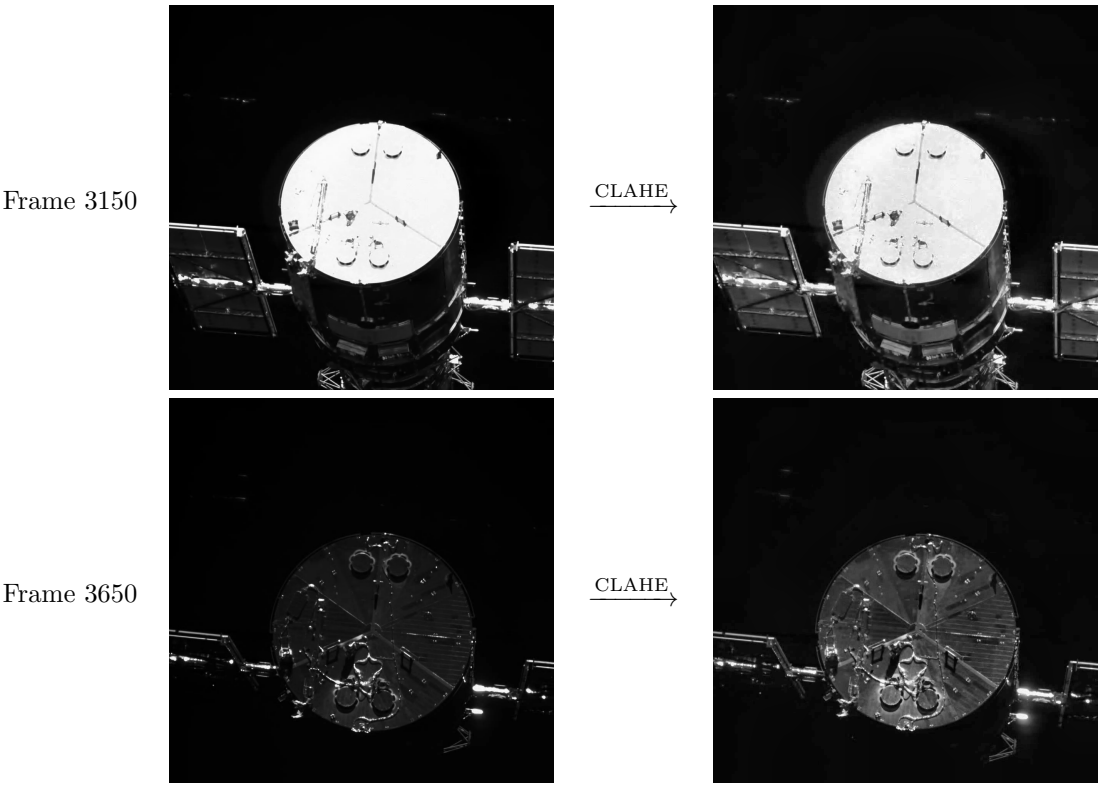

Figure 2. Mitigation of Harsh Lighting Conditions Through CLAHE

experiment are given in Table 1.

In the first phase of the algorithm, a map initialization is attempted. In this step, as illustrated in Fig. 3, detected features are tracked over several frames until sufficient parallax is achieved, so as to establish a valid guess of the transformation between the latest frame and the anchor frame. Noticeably, this step is difficult

Table 1. RNS1 Camera Parameters

Parameter	Value
Detector array size	1000×1000
Pixel size (μm)	6.7
Focal length (mm)	35
Focal length (pixels)	$5,223.5$
f-stop	2.8
Optical center pixel	$(500, 500)$

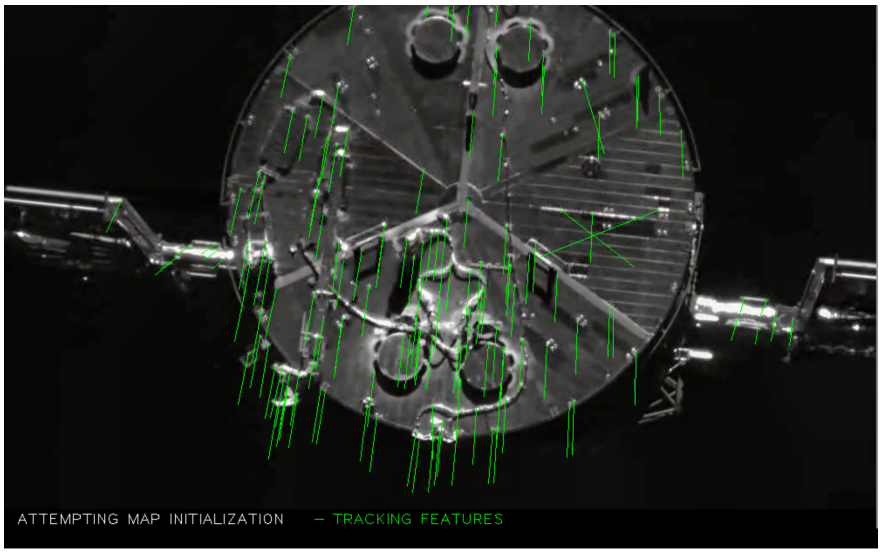

Figure 3. Initialization step of ORB-SLAM on HST sequence.

to accomplish given the narrow field of view of the RNS camera and the long distance to the target. Since the translation of the camera frame induces a parallax in the image plane, for the initialization to succeed, sufficient parallax needs to be achieved. However, given the long distance between the RNS camera and HST, a large translation[13] between frames is thus required for a successful initialization of the map and of the initial camera transformation $T^{\mathcal{N}}_{\mathcal{C}/\mathcal{N}}$. Furthermore, in some initialization trials, cases where a corrupted map was initialized due to a two-fold ambiguity were observed. This is symptomatic of the long distance to the target, meaning that the resulting view geometry is closer to orthographic projection than to perspective projection. In such cases a set of points that respected the view geometry of both frames, but violated the 3D shape of HST, where inserted in the map, leading to false estimation of relative pose subsequently. To resolve this issue, appropriate values of the parameters of ORB-SLAM were chosen. Also note that pure rotation scenarios, where there is no relative translation between the chaser and the target are not conducive to initialization success.

In a second step, as the camera travels around HST, the algorithm accumulates map points by triangulating detected ORB features over multiple views using the local bundle adjustment scheme. This process is illustrated in Fig. 4. The output results of the algorithm are illustrated in Figs. 5 and 7. It is observed that, in general, the SLAM keyframes and the trajectories get closer to each other at the end of the trajectory path. The camera trajectory is the sequential stacking of the result of a motion-only bundle adjustment, which is evaluated at each new image frame. However, due to the constant correction of keyframes in the bundle adjustment thread, which are kept in memory during all the estimation process, a deviation between the past keyframe positions and the trajectory may occur. Note that in a post processing step, evaluation of the unknown scale s is carried-out, by comparing the reconstructed map to a known model of HST. For this purpose, Gaussian mixture models of both point clouds are produced and then compared, via a coherent point drift point set registration.[12]

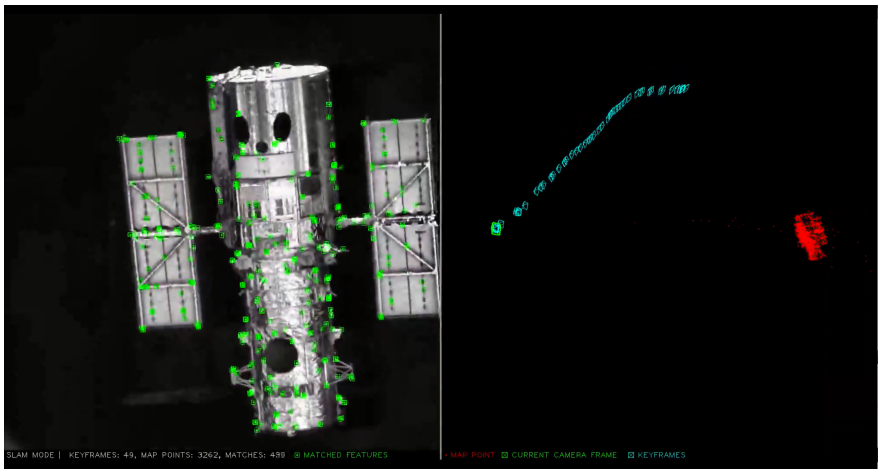

Figure 4. Tracking and Mapping Step of ORB-SLAM on HST sequence.

Since no ground truth data from the considered RNS flight sequence is available, the absolute accuracy of the estimation process could not be evaluated. Instead, the resulting poses were scaled and compared in position and orientation with those resulting from the GNFIR algorithm, originally used in the HST servicing Mission 4, see Figures 5 and 6.

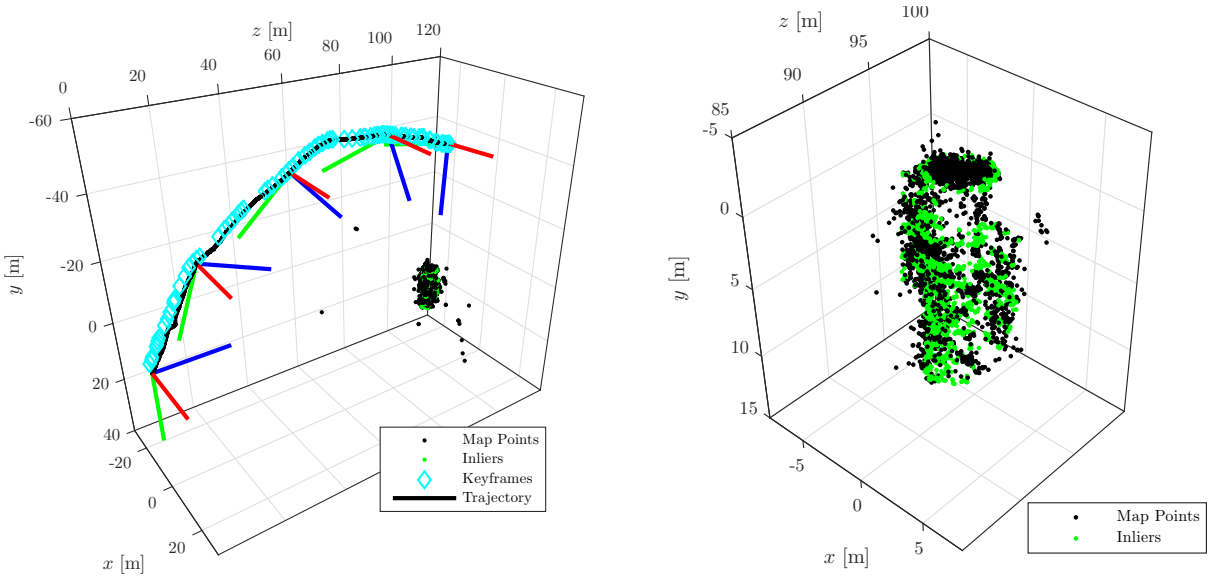

Figure 5. Resulting Pose Estimates and Reconstructed Map

In addition, using a high fidelity 3D model of HST and the estimated scale as mentioned above, the pixel error between the edge map of the camera image and the edge map of the zBuffer of a simulated render, using OpenGL, was calculated. The edge pixels were matched together using an Iterative Closest Point (ICP) algorithm. Errors greater than 5 pixels were rejected, while keeping more than half of the initial edge pixel population. This is due to variability of the edge detection method used (thresholding on a Laplacian of Gaussian). The remaining pixel errors were accumulated, and the normalized root mean square error (NRMSE) metric measuring the normalized average distance between the reprojection of the 3D model using the estimated poses and the detected image was calculated (see Fig. 8). One standard deviation of the distribution of the detected image edge pixels was used as normalization factor, thus mimicking the apparent size of the satellite in the image.

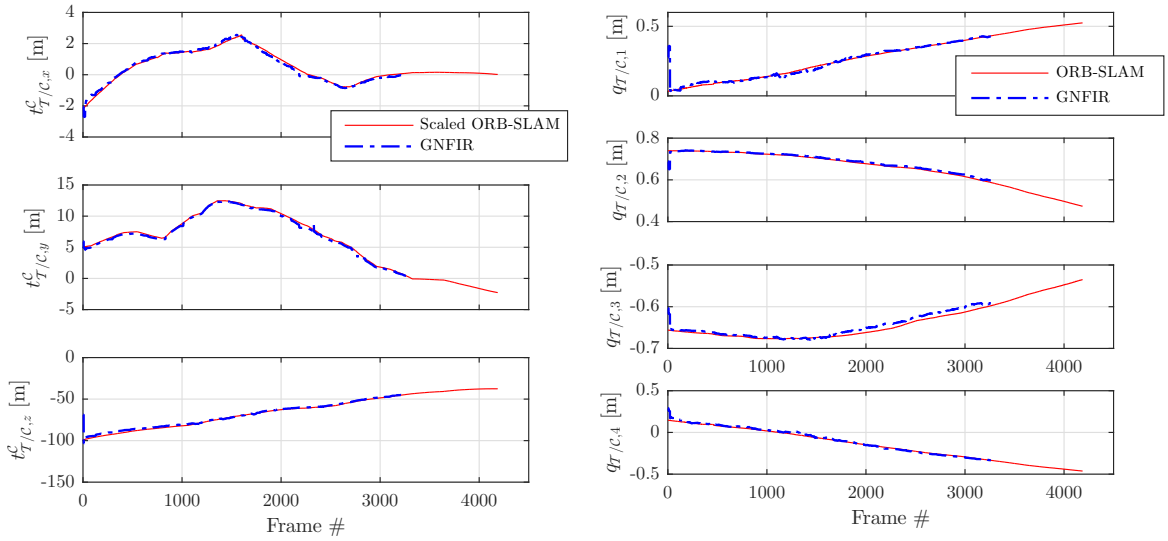

Figure 6. Comparison in Scaled Position and Orientation between GNFIR and ORB-SLAM

B. ASTROS Experimental Facility

The ASTROS platform supports experiments in the area of vision-based autonomous rendezvous and docking, specifically directed towards on-orbit servicing of spacecraft. The system consists of a raised hemi-spherical air bearing which provides 2 DOF of motion, as well as three other linear air bearings situated at the base of the structure, which in turn floats on a near-perfectly flat epoxy floor, providing a further 2 DOF in translation motion and 1 DOF in rotation. The platform is fitted with 4 Variable Speed Control Moment Gyros (VSCMG's) as well as 12 high-pressure thrusters, allowing for numerous experimental scenarios to be tested with a high level of realism. Possessing a real-time SpeedGoat[15] computer on the platform allows for demonstration of on-board capabilities. The system capabilities include the ability to run sophisticated planning and control algorithms, exploiting the actuators, and sampling sensors at high frequencies, in order to execute maneuvers within the testing arena. Additionally, a dedicated NVidia Jetson TX1 Module permits highly parallelizable vision-based and deep learning algorithms to be run simultaneously, with reasonably low power consumption. Furthermore, ASTROS has a suite of sensors including a rate gyrometer, an Inertial Measurement Unit (IMU), magnetometer and Sun sensor. Finally, the platform is fitted with a monocular PointGrey Flea3 camera, which has been put to use in this experiment to produce video footage, with various array sizes.

Table 2. PointGrey FL3-U3-20E4C Camera Parameters

Parameter	Value
Detector array size	600×800
Pixel size (μm)	4.5
Focal length (mm)	8
Focal length (pixels)	1,777
f-stop	3.2
Optical center pixel	(385,296)

The ASTROS facility is also fitted with eight VICON interior global positioning cameras, producing reliable ground truth position and orientation measurements for comparison purposes. Appropriate lighting and dark environment surroundings allow for realistic footages to be shot during tests, see Figure 9. The harsh contrasts inherent to imaging highly reflective surfaces against a dark background can be reliably reproduced in the ASTROS facility, as shown in Figure 10.

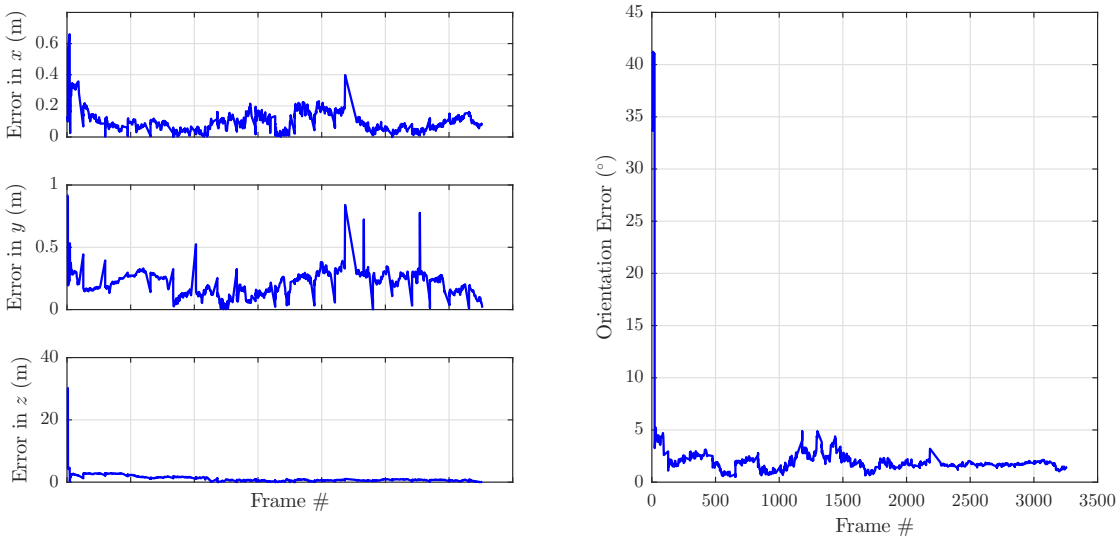

Figure 7. Errors in Scaled Position and Orientation, as Compared to GNFIR

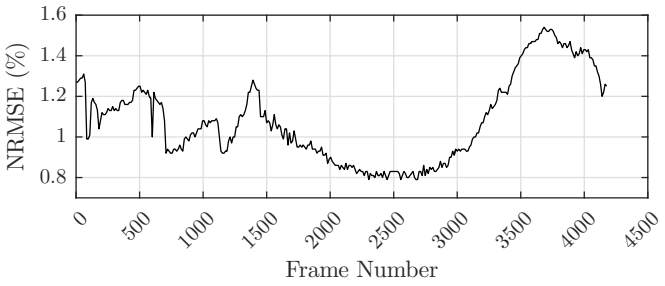

Figure 8. Normalized Root Mean Square Error of Edge Pixels

In this experiment, the floating platform was manually moved around the ASTROS arena, while keeping the camera generally pointed at a dummy 1U CubeSat, which hangs from the ceiling in the center of the arena, having some freedom of motion itself. Several sequences of images were captured using the on-board camera. The ORB-SLAM algorithm was then run with these footages, and the resulting pose data was compared to those output from the VICON system. To mitigate the fact that no appropriate comparison data was available for the validation of the results using the HST sequence, VICON output data was captured so as to track the ground truth position of the ASTROS floating platform for comparison purposes. For each video frame $i = 1 \dots N_{\text{frames}}$ of the camera footage, proper timestamping and communication delay compensation were exploited so to associate a valid data point from the VICON system.

To this end, assume that the VICON system global reference frame is frame $\mathcal{E}$. In practice, tracking the platform amounts to tracking the $\mathcal{S}$ frame at each video frame i over the duration of the test, encoded in transformation $T^{\mathcal{E},i}_{\mathcal{S}/\mathcal{E}}$. Furthermore, the target dummy satellite is also tracked, associated with the $\mathcal{T}$ frame, whose pose is encoded in the transformation $T^{\mathcal{E},i}_{\mathcal{T}/\mathcal{E}}$. An ideal navigation algorithm should directly estimate the pose of the $\mathcal{T}$ frame with respect to the $\mathcal{S}$ frame. Yet, as explained in Subsection A of Section II, ORB-SLAM only outputs a scale ambiguous transformation $T^{\mathcal{N}}_{\mathcal{C}'/\mathcal{N}}$. However, we know that at the initialization frame i_{init} of the algorithm, frame $\mathcal{N}$ coincidences with frame $\mathcal{C}$, and thus we can assume that

$$T^{\mathcal{N},i_{\text{init}}}_{\mathcal{T}/\mathcal{N}} = T^{\mathcal{C},i_{\text{init}}}_{\mathcal{T}/\mathcal{C}}. \tag{7}$$

In turn, we know that

$$T^{\mathcal{C},i_{\text{init}}}_{\mathcal{T}/\mathcal{C}} = T^{\mathcal{T},i_{\text{init}}}_{\mathcal{E}/\mathcal{T}} T^{\mathcal{E},i_{\text{init}}}_{\mathcal{C}/\mathcal{E}} = \left[T^{\mathcal{E},i_{\text{init}}}_{\mathcal{T}/\mathcal{E}}\right]^{-1} T^{\mathcal{E},i_{\text{init}}}_{\mathcal{S}/\mathcal{E}} T^{\mathcal{S}}_{\mathcal{C}/\mathcal{S}}. \tag{8}$$

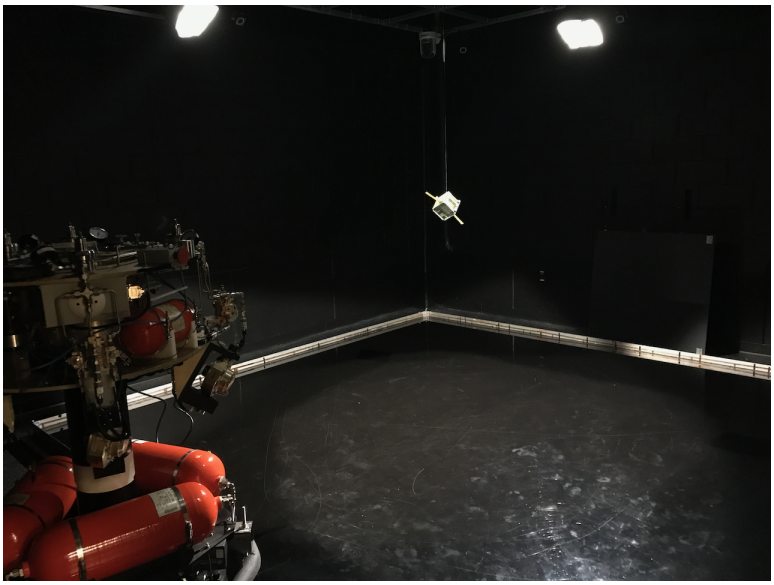

Figure 9. The ASTROS 5-DOF Floating Platform and Arena

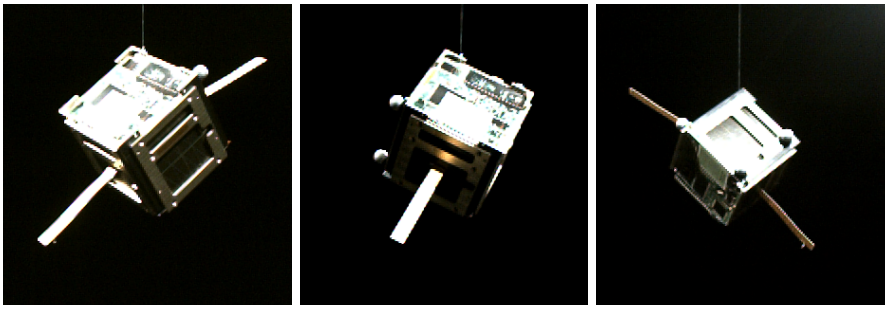

Figure 10. Examples of Realistic Imagery of a Target Cubesat in ASTROS Experimental Facility

It is crucial to note that frames $\mathcal{N}$ and $\mathcal{C}$ are guaranteed to coincide only at the initialization time, since the target is free to translate and rotate. This means that target body-fixed frames $\mathcal{N}$ and $\mathcal{T}$ may be moving with regards to the $\mathcal{E}$ frame during the test. By the rigid body assumption, it follows that

$$T_{\mathcal{N}/\mathcal{T}}^{\mathcal{T},i} = T_{\mathcal{N}/\mathcal{T}}^{\mathcal{T},i_{\text{init}}}, \quad \text{for all } i \geq i_{\text{init}}. \tag{9}$$

Finally, we obtain the resulting λ-scaled ORB-SLAM estimated camera pose with regards to frame $\mathcal{E}$ at the i-th video frame, given by

$$\tilde{T}_{\mathcal{C}/\mathcal{E}}^{\mathcal{E},i} = T_{\mathcal{T}/\mathcal{E}}^{\mathcal{E},i} T_{\mathcal{N}/\mathcal{T}}^{\mathcal{T},i_{\text{init}}} T_{\mathcal{C}/\mathcal{N}}^{\mathcal{N},i} = T_{\mathcal{T}/\mathcal{E}}^{\mathcal{E},i} \left[T_{\mathcal{T}/\mathcal{N}}^{\mathcal{N},i_{\text{init}}} \right]^{-1} \begin{bmatrix} R_{\mathcal{N}/\mathcal{C}'}^{i} & \lambda t_{\mathcal{N}/\mathcal{C}'}^{\mathcal{C}',i} \\ 0 & 1 \end{bmatrix}^{-1}. \tag{10}$$

We then compare the latter transformation to that which is informed by the VICON system, namely

$$T_{\mathcal{C}/\mathcal{E}}^{\mathcal{E},i} = T_{\mathcal{S}/\mathcal{E}}^{\mathcal{E},i} T_{\mathcal{C}/\mathcal{S}}^{\mathcal{S}}. \tag{11}$$

Qualitative results are shown in Figures 12, 14 and 16, which correspond to three test cases executed in the ASTROS facility. As can be seen in these figures, the target satellite shape is approximately captured by the obtained map point cloud. Furthermore, comparison to the ground truth shows good accuracy, up to the predetermined scaling value, as illustrated in Figures 13, 15, 17. It is however noticeable that ORB-SLAM seems to be producing output pose information for only some portions of the its track in the arena, as the

platform circumnavigates the target. It is observable, in the three test cases, that the successful segments of tracking are always on the same side. In fact, by looking at the relative orientation quaternion $q_{\mathcal{T}/\mathcal{C}}$ in Figure 11, we can see that over three revolutions around the target, the segments for which tracking is conserved are centered around a specific reoccurring orientation. Considering the way that the ORB-SLAM algorithm implements tracking, a set of features stored in previously seen keyframes are used for matching, using its Bag of Words (DBoW) method.[8]

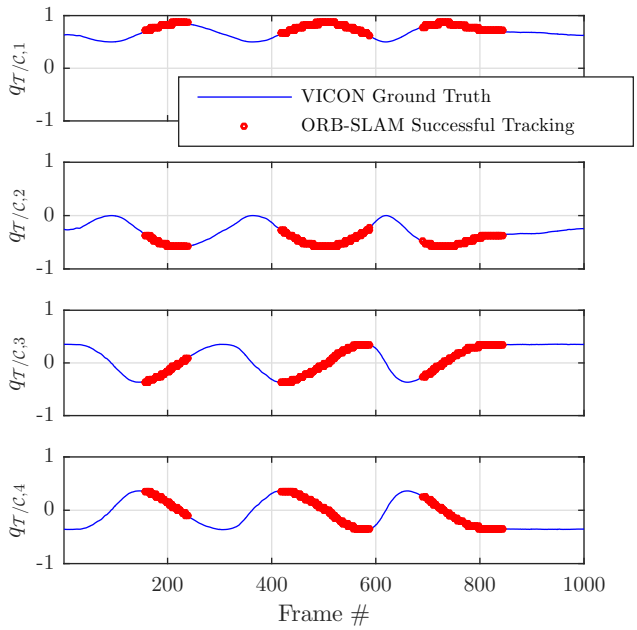

Figure 11. Orientations with Successful Tracking

The loss of tracking can be explained by the fact that since the target is a convex prism, at some orientations, visibility of almost half of the tracked features lying on the same plane (i.e. one of the sides of the CubeSat) is lost at the same time. Even though ORB-SLAM constantly initializes new features and map points, this does not compensate for the gross loss of features. Noticeably, however, the length of the tracked segment seems to grow at each successive revolution, which is the result of the growing log of keyframes that are stored for place recognition.

Indeed, further study has to be dedicated to the specialization of the ORB-SLAM algorithm so to improve the performance of tracking, since the our scenario involving the tracking of features on a convex object departs from the typical ground robotics scenario, where features are tracked on surfaces that usually form a concave object as viewed from the camera, such as the inner walls of a room about which a quadrocoptor maneuvers. It is important to note that even in the classical application, a harsh maneuver causing a loss of visibility of a majority of tracked features between two successive frames will cause the algorithm to lose tracking. Hence, keeping a good portion of tracked features of the previous camera frame is crucial to ensure conservation of tracking. The loss of tracking over a revolution means that the loop-closure capabilities of ORB-SLAM can not be properly tested.

IV. Conclusion

SLAM is usually implemented in a static scene on-board environment-aware robots. By tracking landmark points in this environment and by exploiting odometry measurements, the robot gradually builds a true-scale estimate of its own position within the scene, and also builds a map of the scene at the same time. In contrast, in a spacecraft navigation scenario, little to no dynamics-based measurements, such as accelerations, are available if no maneuvers are executed, yet the spacecraft has relative motion. Moreover, in the relative navigation scheme, very little maneuvering is required to maintain the relative dynamics between the chaser

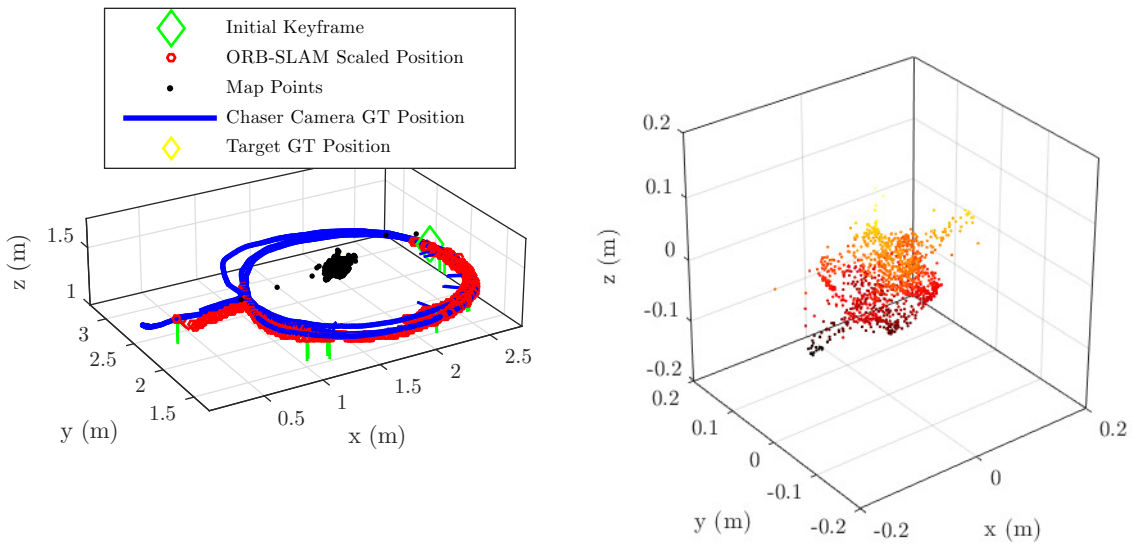

Figure 12. Test 1 Resulting Pose Estimates and Reconstructed Map

and the target. Hence, a robust vision-based algorithm like ORB-SLAM, which autonomously provides relative pose (up to a scale) between the chaser and the target by simply exploiting camera images is a well-adapted part of the navigation solution intended for non-cooperative rendezvous. ORB-SLAM does so without requiring any prior information on the target or any ego-motion information. In this paper, the application of the ORB-SLAM algorithm to real data from experiments demonstrated that the SLAM algorithm, in its bundle adjustment variety, is well adapted for a long sequence of input images. It has to be noted that the HST sequence used in this study did not allow for testing the loop-closure capability of the ORB-SLAM algorithm. Furthermore, in the ASTROS experiment test cases, loop-closure could not be achieved repeatedly, even while a real-world rendezvous scenario was simulated, during which the chaser and target complete several revolutions around each other. ORB-SLAM has demonstrated in ground robotics literature that it can eliminate drift that may have accumulated due to the scale ambiguity, by executing a full bundle adjustment after automatic place recognition. Hence, specializing the algorithm to the specific challenges that arise in spacecraft rendezvous will be the subject of further study.

V. Appendix

A. Brief Formulation of the SLAM Problem and ORB-SLAM Algorithm

The SLAM problem can be viewed as a general maximum likelihood estimation problem. To this end, consider a set of N measurement vectors $u_1, u_2, \dots u_N$, a set of unknown parameters Θ and a measurement function $h(\Theta)$. Estimating Θ amounts to finding the most probable (and hence optimal) parameter set Θ^* that maximizes the likelihood (or, equivalently, minimizes the negative log-likelihood) of the function $p(u_1, \dots, u_N|\Theta)$. Assuming independence between the measurements, i.e., that $p(u_1, u_2, \dots, u_N) = p(u_1)p(u_2)\dots p(u_N)$, it follows that

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, p(u_1, \dots, u_N|\Theta) = \underset{\Theta}{\operatorname{argmax}} \prod_{i=1}^{N} p(u_i|\Theta) \tag{12}$$

$$= \underset{\Theta}{\operatorname{argmin}} - \sum_{i=1}^{N} \log\left(p\left(u_i|\Theta\right)\right). \tag{13}$$

Furthermore, assuming that the likelihood function $p(u_i|\Theta)$ follows a zero-meane Gaussian distribution, that is,

$$p(u_i|\Theta) \propto \exp\left((u_i - h_i(\Theta))^\top \Sigma^{-1} (u_i - h_i(\Theta))\right), \tag{14}$$

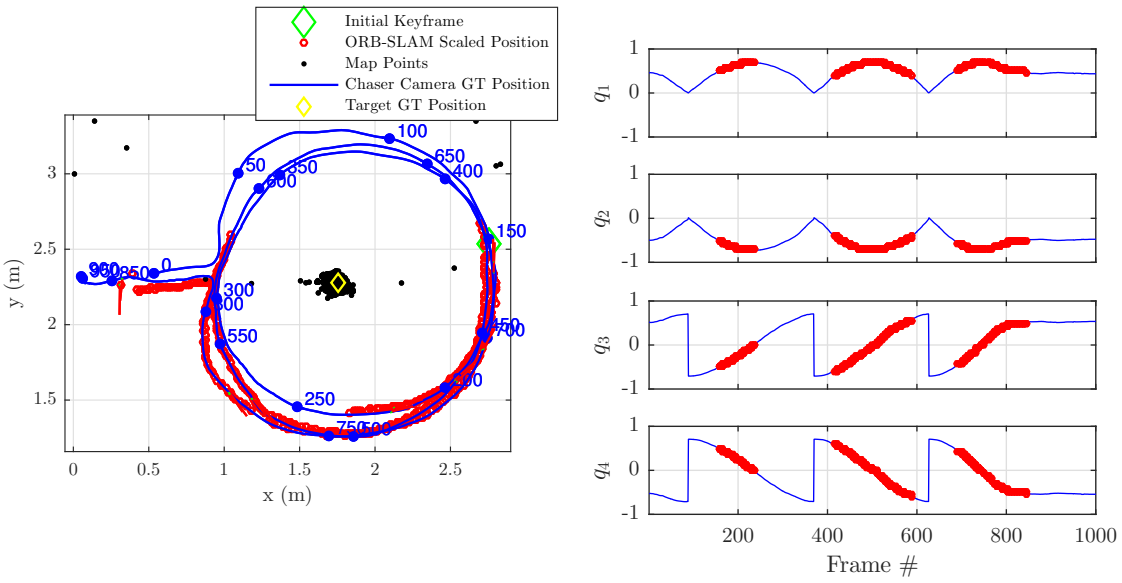

Figure 13. Test 1 Scaled Position and Orientation Comparison with VICON Ground Truth

with the measurement covariance matrix Σ, then the problem is reduced to a least-squares optimization problem given by

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{N} \|u_i - h_i(\Theta)\|_\Sigma^2, \tag{15}$$

where $\|\Delta\|_\Sigma^2 = \Delta^\top \Sigma^{-1} \Delta$.

The monocular pinhole camera projection function $\pi : \mathbb{R}^2 \times \mathbb{R}_{\neq 0} \to \mathbb{R}^2$ is defined as

$$\pi\left([x \quad y \quad z]^\top\right) = \begin{bmatrix} f_x(x/z) + c_x, & f_y(y/z) + c_y \end{bmatrix}^\top \tag{16}$$

where $f_x, f_y \in \mathbb{R}$ are the camera focal lengths, in number of pixels, and $(c_x, c_y) \in \mathbb{R}^2$ are the coordinates of the optical center of the camera, in pixels, determined during an a-priori camera calibration step.

Consider a finite number of camera frames $\mathcal{I}$, corresponding to images taken in a chronological sequence, and note that to each such image frame $i \in \mathcal{I}$ is associated a camera pose given by

$$T_{\mathcal{C}/\mathcal{N}}^{\mathcal{N},i} = \begin{bmatrix} R_{\mathcal{C}/\mathcal{N}}^{i} & t_{\mathcal{C}/\mathcal{N}}^{\mathcal{N},i} \\ 0 & 1 \end{bmatrix},$$

and a set of detected image keypoints $\mathcal{D}_i$, with each keypoint $j \in \mathcal{D}_i$ possessing image coordinates $u_{ji} \in \mathbb{R}^2$.

Let $\mathcal{G} = (\mathcal{K}, E)$ denote a graph, consisting of a finite set of vertices $\mathcal{K} \subseteq \mathcal{I}$ herein known as keyframes, (this is a select group of camera frames over which the SLAM optimization by bundle adjustment is carried-out) and a finite set of edges $E \subseteq \mathcal{K} \times \mathcal{K}$ that connect keyframes, thus encoding the structure of the multiple view geometry.

Let $\mathcal{P}_k, \mathcal{P}_\ell \subseteq \mathcal{M}$ be the set of map points which are visible in keyframes $k, \ell \in \mathcal{K}, k \neq \ell$, respectively. Then keyframes k and ℓ are said to be co-visible if an edge $e \in E$ exists between the two in $\mathcal{G}$. An edge e is added if there are at least $N_{\text{covisible}}$ common visible map points (co-visible), that is,

$$e = \begin{cases} (k, \ell) & \text{if} \quad |\mathcal{P}_k \cap \mathcal{P}_\ell| \geq N_{\text{covisible}}, \\ \emptyset & \text{otherwise}, \end{cases} \tag{17}$$

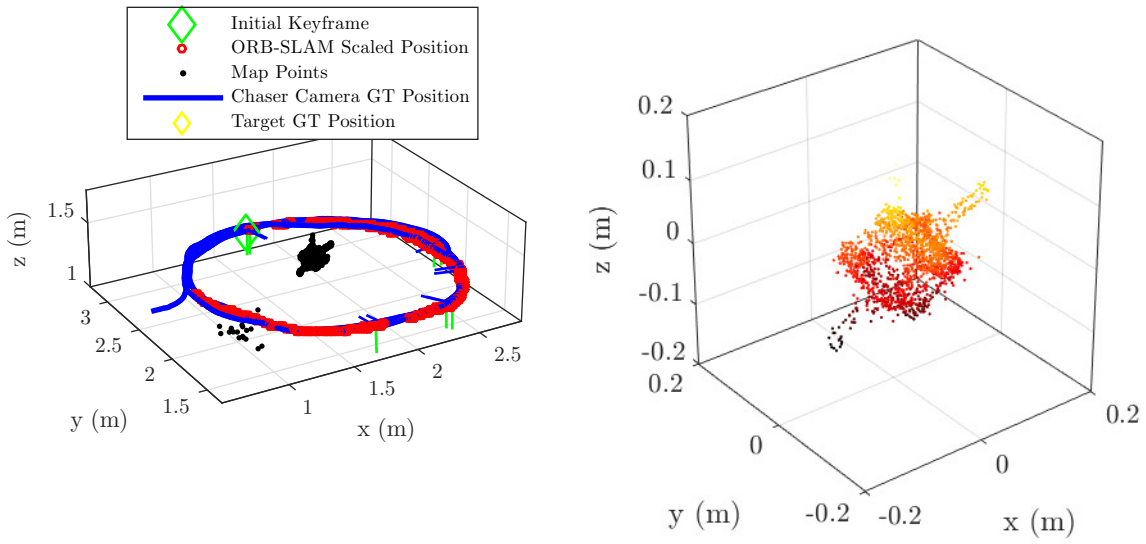

Figure 14. Test 2 Resulting Pose Estimates and Reconstructed Map

where $|\cdot| : \mathcal{S} \to \mathbb{N}$ denotes the cardinality of a set in $\mathcal{S}$.

Let $\mathcal{K}_L \subseteq \mathcal{K}$ be the set of all co-visible keyframes. It follows that the set of all co-visible points $\mathcal{P}_L$ is given by

$$\mathcal{P}_L = \bigcup_{k \in K_L} \mathcal{P}_k. \tag{18}$$

Let $\mathcal{K}_F$ be the set of keyframes which have common visible points with all keyframes $\ell \in K_L$, excluding keyframes within the set of possibly connected keyframes $\mathcal{K}_L$, that is,

$$\mathcal{K}_F = \{k \in \mathcal{K} : |P_k \cap P_L| > 0 \quad \text{and} \quad k \notin \mathcal{K}_L\}. \tag{19}$$

In ORB-SLAM, the least-squares estimation problem is repeated for all $\ell \in \mathcal{K}_L$, the set of co-visible frames, and $m \in \mathcal{P}_L$, the set of points viewed in those co-visible frames, by applying a computationally efficient sliding window scheme. Specifically, consider the pair $(j, r) \in \mathcal{J}_k \subseteq \mathcal{D}_k \times P_k$ matching keypoint $j \in \mathcal{D}_k$ to map point $r \in \mathcal{P}_k$, the image coordinates $u_{jk} \in \mathbb{R}^2$ of keypoint j detected in keyframe k, the projected image coordinates $u_{rk} \in \mathbb{R}^2$ of the of map point r, obtained by applying the camera projection mapping $\pi : \mathbb{R}^2 \times \mathbb{R}_{\neq 0} \to \mathbb{R}^2$ to the position vector $X_r^{\mathcal{C}} \in \mathbb{R}^3$ of the of map point r, expressed in camera frame $\mathcal{C}$ coordinates, which in turn relates to the rigid-body transformation given by the rotation $R_{\ell,\mathcal{C}/\mathcal{N}} \in SO(3)$ and translation $t_{\ell,\mathcal{C}/\mathcal{N}}^{\mathcal{N}} \in \mathbb{R}^3$. Then the sum of the squared errors between u_{jk} and u_{rk} is minimized by varying the coordinates of the co-visible map points $\mathcal{P}_L$ and the pose of the co-visible keyframes $\mathcal{K}_L$, while keeping fixed the pose of keyframes $\mathcal{K}_F$, the set of all keyframes not in $\mathcal{K}_L$ that share a number of common keypoints under a certain threshold. The optimization problem is also programmed with the relevant $SE(3)$-related constraints.

The whole problem can be written as a nonlinear program, for all $m \in \mathcal{P}_L$ and $\ell \in \mathcal{K}_L$

$$\begin{aligned} \underset{X_m,\, R_{\ell,\mathcal{C}/\mathcal{N}},\, t_{\ell,\mathcal{C}/\mathcal{N}}^{\mathcal{N}}}{\text{minimize}} \quad & \sum_{k \in \mathcal{K}_L \cup \mathcal{K}_F} \sum_{(j,r) \in \mathcal{J}_k} \rho\left(\|u_{jk} - u_{rk}\|_{\Sigma_{jk}}^2\right), \\ \text{subject to} \quad & X_m \in \mathbb{R}^3, \\ & R_{\mathcal{C}/\mathcal{N}}^{\ell} \in \mathrm{SO}(3), \\ & t_{\mathcal{C}/\mathcal{N}}^{\mathcal{N},\ell} \in \mathbb{R}^3 \end{aligned} \tag{20}$$

where $u_{rk} = \pi(R_{\mathcal{C}/\mathcal{N}}^{k} X_r + t_{\mathcal{C}/\mathcal{N}}^{\mathcal{N},k})$, X_m is the position vector of point m expressed in the global frame $\mathcal{N}$ coordinates, $R_{\mathcal{C}/\mathcal{N}}^{\ell} \in \mathrm{SO}(3)$ is the camera frame orientation at keyframe ℓ with regards to frame $\mathcal{N}$,

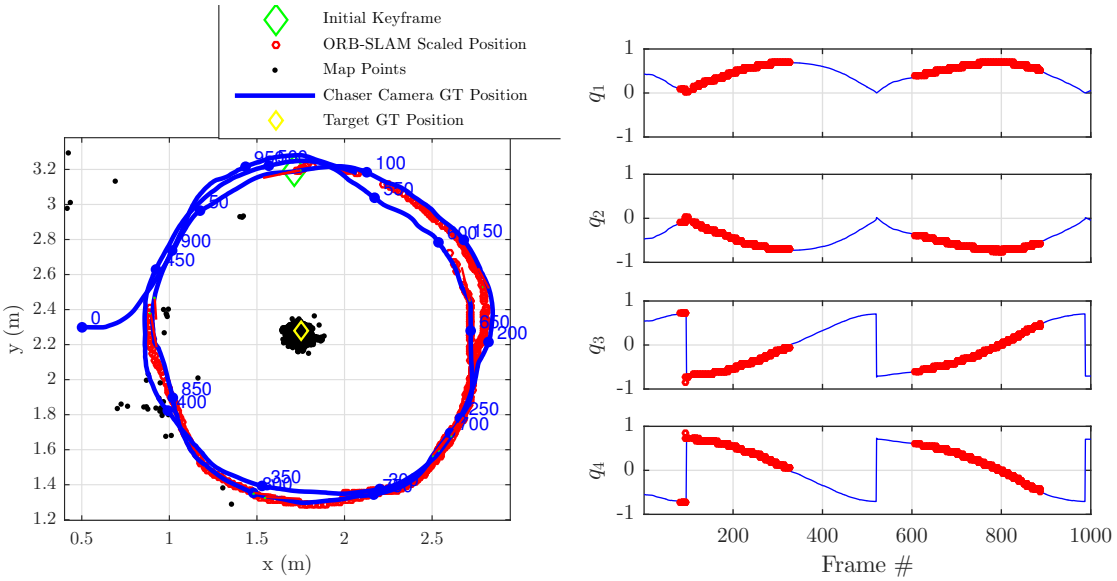

Figure 15. Test 2 Scaled Position and Orientation Comparison with VICON Ground Truth

$t^{\mathcal{N},\ell}_{\mathcal{C}/\mathcal{N}} \in \mathbb{R}^3$ is camera frame position vector at keyframe ℓ with regards to frame $\mathcal{N}$ expressed in global coordinates, $\rho : \mathbb{R} \to \mathbb{R}$ is a robust Huber cost function,[16] $\Sigma_{jk} = \sigma^2_{jk} I_2$ is the covariance matrix related to the scale of the keypoint j in keyframe k, and $\|\Delta\|^2_\Sigma = \Delta^\top \Sigma^{-1} \Delta$.

Let X^*_m for $m \in \mathcal{P}_L$, and $R^{*,\ell}_{\mathcal{C}/\mathcal{N}}$, $t^{*,\mathcal{N},\ell}_{\mathcal{C}/\mathcal{N}}$ for $\ell \in \mathcal{K}_L$, denote the optimal solution to problem (20). Then, when tracking is enabled, X^*_m provides the coordinates of a fixed map of scene points, which will subsequently be used to extract the optimal pose of the camera frame at every image frame that does not correspond to any keyframe $k \in \mathcal{K}$ of the graph $\mathcal{G} = (\mathcal{K}, E)$. This process is called motion-only bundle adjustment.[8]

The ORB-SLAM Local BA requires an initialized map, which is automatically computed at the start of the sequence using two image frames with sufficient parallax. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be the set of detected feature points in the image frames when the camera is at frames $\mathcal{N}$ and $\mathcal{C}$, respectively. Let $\mathcal{J}_{12}$ be the set of pairs of indices corresponding points $\mathcal{D}_1$ to points in $\mathcal{D}_2$, with matching based on a heuristic involving the ORB feature descriptor.[9]

Then, for $(j,k) \in \mathcal{J}_{12}$, $u_j \in \mathbb{R}^2$ is the vector of image coordinates of point $j \in \mathcal{D}_1$ and $u_k \in \mathbb{R}^2$ is the vector of the image coordinates of point $k \in \mathcal{D}_2$, to which are associated the vectors of homogeneous coordinates $\underline{x}_j, \underline{x}_k \in \mathbb{P}^2$, respectively.

If the scene is non-planar, then the transformation between $\underline{x}_j$ and $\underline{x}_k$ is explained by a fundamental matrix $F \in \mathcal{F} = \left\{([t]_\times R) : R \in \mathrm{SO}(3),\ t \in \mathbb{R}^3\right\} \subset \mathbb{R}^{3\times 3}$, where $[\,\cdot\,]_\times$ denotes the skew-symmetric matrix emulating the left cross product. The transformation must satisfy the epipolar constraint for a non-planar scene

$$0 = \underline{x}_k^\top F \underline{x}_j. \tag{21}$$

F can be recovered from (21) by applying the 8-point algorithm.[13]

If the scene is planar in nature, the solution for the transformation between $\underline{x}_j$ and $\underline{x}_k$ obtained through the 8-point method recovering a fundamental matrix will be degenerate. However, the scene might be explained by a homography matrix $H \in \mathbb{R}^{3\times 3}$, satisfying the epipolar constraint for a planar scene

$$0 = [\underline{x}_k]_\times H \underline{x}_j. \tag{22}$$

H can be recovered from (22) using the Direct Linear Transformation (DLT) algorithm (4-point algorithm).[13]

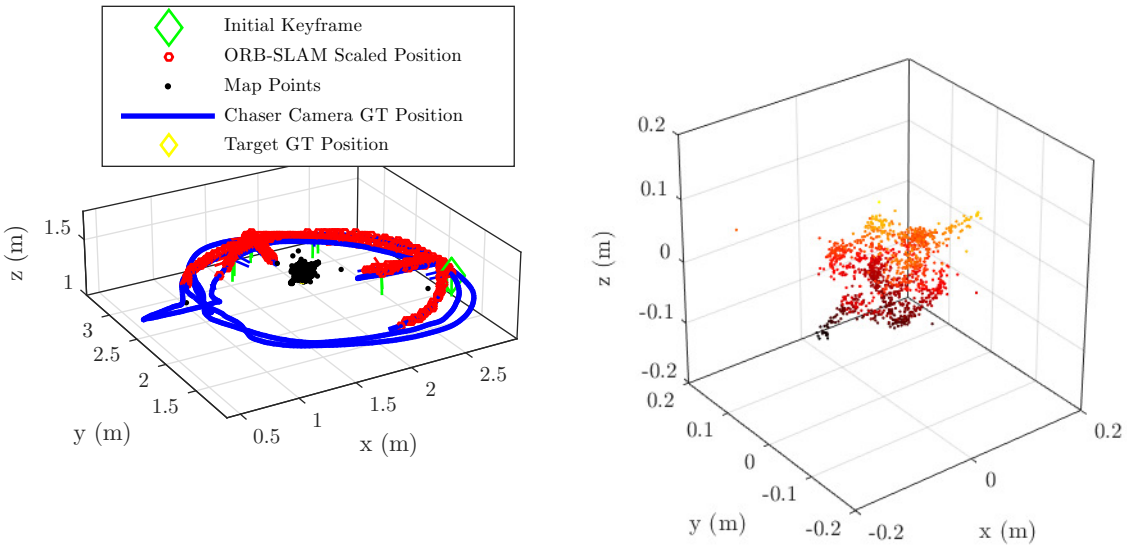

Figure 16. Test 3 Resulting Pose Estimates and Reconstructed Map

References

[1]Aghili, F., Kuryllo, M., Okouneva, G., and McTavish, D., "Robust Pose Estimation of Moving Objects Using Laser Camera Data for Autonomous Rendezvous and Docking," *Proceedings of the International Society of Photogrammetry and Remote Sensing Archives*, Vol. 38, 2009, pp. 3.

[2]Lichter, M. D. and Dubowsky, S., "State, Shape, and Parameter Estimation of Space Objects from Range Images," *IEEE International Conference on Robotics and Automation.*, Vol. 3, 2004, pp. 2974–2979 Vol.3.

[3]Augenstein, S., Rock, S. M., Enge, P., and Tomalin, C. J., *Monocular Pose and Shape Estimation of Moving Targets, for Autonomous Rendezvous and Docking*, Ph.D. thesis, Stanford University, 2011.

[4]Tweddle, B. E., *Computer Vision Based Navigation for Spacecraft Proximity Operations*, Ph.D. thesis, Massachusetts Institute of Technology, 2013.

[5]Strasdat, H., Montiel, J., and Davison, A. J., "Visual SLAM: Why filter?" *Image and Vision Computing*, Vol. 30, No. 2, 2012, pp. 65 – 77.

[6]Kaess, M., Ranganathan, A., and Dellaert, F., "iSAM: Incremental Smoothing and Mapping," *IEEE Transactions on Robotics*, Vol. 24, No. 6, 2008, pp. 1365–1378.

[7]Stachniss, C., Frese, U., and Grisetti, G., "OpenSLAM," `https://openslam.org/`, 2007.

[8]Mur-Artal, R., Montiel, J. M. M., and Tards, J. D., "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, Vol. 31, No. 5, Oct 2015, pp. 1147–1163.

[9]Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: an Efficient Alternative to SIFT and SURF," *IEEE International Conference on Computer Vision*, IEEE, 2011, pp. 2564–2571.

[10]Prussing, J. E. and Conway, B. A., *Orbital Mechanics*, Oxford University Press, 2012.

[11]Curtis, H. D., *Orbital Mechanics for Engineering Students, Second Edition*, Butterworth-Heinemann, 2009.

[12]Myronenko, A. and Song, X., "Point Set Registration: Coherent Point Drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 12, 2010, pp. 2262–2275.

[13]Ma, Y., Soatto, S., Koseck, J., and Sastry, S. S., *An Invitation to 3-D Vision*, Springer, 2010.

[14]Naasz, B., Eepoel, J. V., Queen, S., Southward, C. M., and Hannah, J., "Flight Results from the HST SM4 Relative Navigation Sensor System," *Advances in Astronautical Sciences Guidance and Control Conference*, 2010.

[15]SpeedGoat GmbH, "Real-Time Target Machines," `https://www.speedgoat.com/products-services/real-time-target-machines`, 2017, [Online; Viewed December 5th 2017].

[16]Zhang, Z., "Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting," *Image and Vision Computing*, Vol. 15, No. 1, 1997, pp. 59–76.

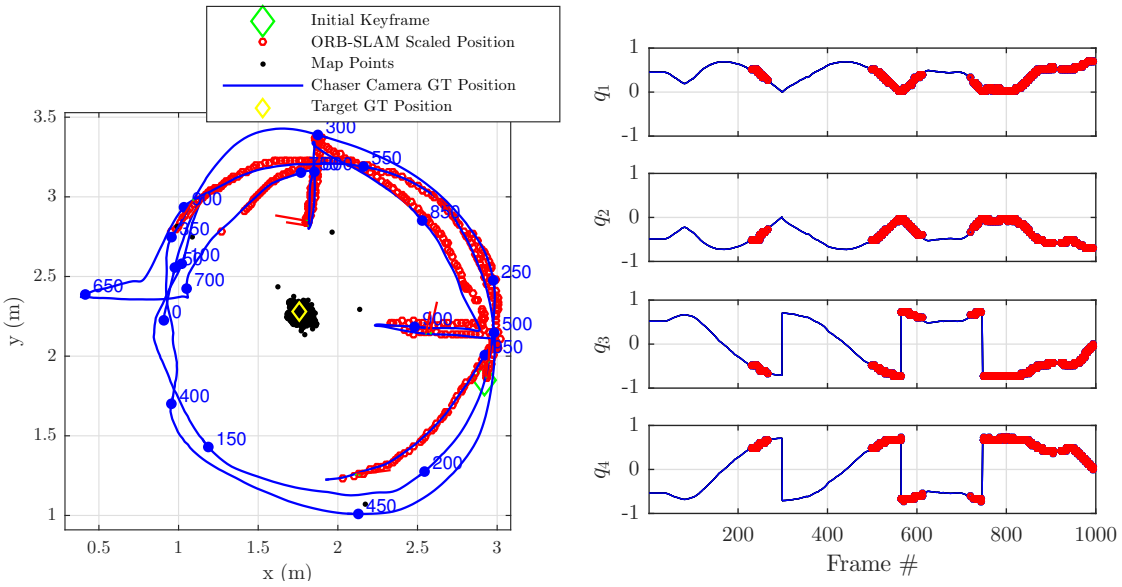

Figure 17. Test 3 Scaled Position and Orientation Comparison with VICON Ground Truth

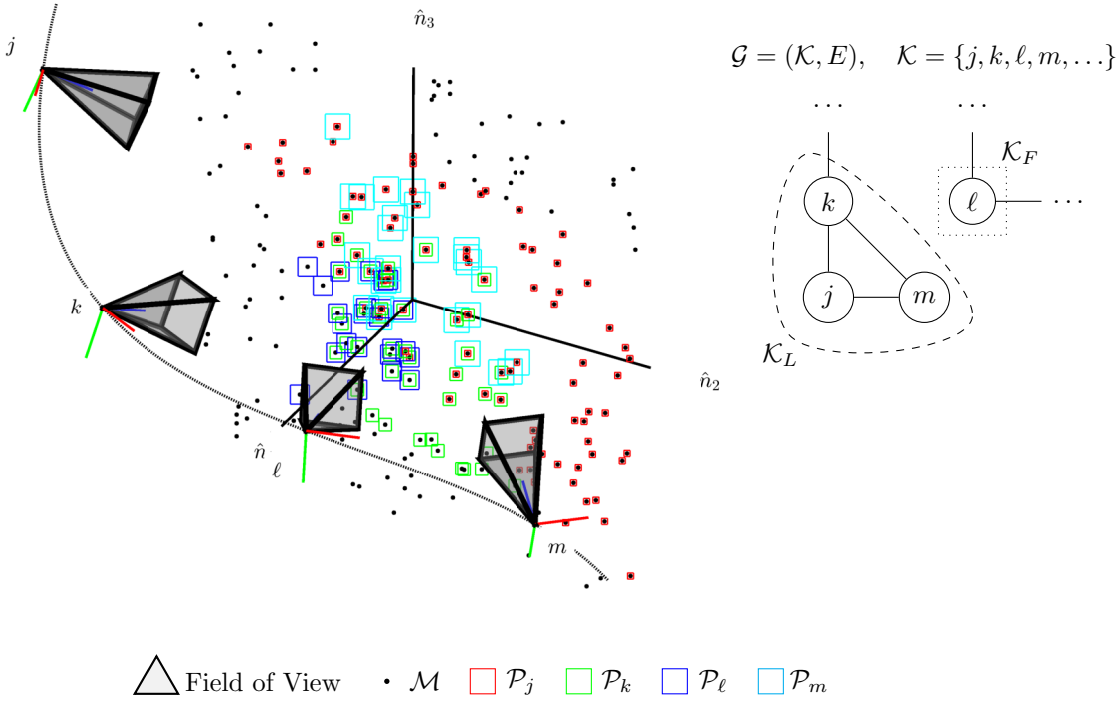

Figure 18. Schematic View of Covisible Keyframes Concept