MIT Open Access Articles

*Mapping and determining the center of mass
of a rotating object using a moving observer*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

Citation: Setterfield, Timothy P et al. "Mapping and Determining the Center of Mass of a Rotating Object Using a Moving Observer." The International Journal of Robotics Research 37, 1 (January 2018): 83–103 © 2018 SAGE Publications

As Published: <https://doi.org/10.1177/0278364917749024>

Publisher: SAGE Publications

Persistent URL: <http://hdl.handle.net/1721.1/113711>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike

Mapping and Determining the Center of Mass of a Rotating Object Using a Moving Observer

Timothy P. Setterfield¹, David W. Miller¹, John J. Leonard², Alvar Saenz-Otero¹

Abstract

For certain applications, such as on-orbit inspection of orbital debris, defunct satellites, and natural objects, it is necessary to obtain a map of a rotating object from a moving observer, as well estimate the object's center of mass. This paper addresses these tasks using an observer that measures its orientation, angular rate, acceleration, and is equipped with a dense 3D visual sensor such as stereo cameras or LiDAR. The observer's trajectory is estimated independently of the target object's rotational motion. Pose graph mapping is performed using visual odometry to estimate the observer's trajectory in an arbitrary target-fixed frame. In addition to applying pose constraint factors between successive frames, loop closure is performed between temporally non-adjacent frames. A kinematic constraint on the target-fixed frame, resulting from the rigidity of the target object, is exploited to create a novel rotation kinematic factor. This factor connects a trajectory estimation factor graph with the mapping pose graph, and facilitates the estimation of the target's center of mass. Map creation is performed by transforming detected feature points into the target-fixed frame, centered at the estimated center of mass. Analysis of the algorithm's computational performance reveals that its computational cost is negligible compared with that of the requisite image processing.

Keywords

SLAM, dynamic, rotation kinematic factor

1 Introduction

Many objects in space such as orbital debris, defunct satellites, and natural objects, are uncooperative and unknown, meaning that they are not equipped with working sensors or actuators, and may have unknown visual appearance and inertial properties. There are several reasons to want to visit these uncooperative space objects with an observer satellite (a.k.a. inspector): to deorbit or deflect dangerous debris; to inspect and/or repair defunct satellites; or to observe, sample, or extract resources from natural objects such as asteroids and comets. After performing orbital rendezvous with the object (a.k.a. target), a mission designed to accomplish one of the aforementioned objectives will typically involve observing the target object from a safe distance, acquiring information and planning the subsequent phases of the mission.

In order to approach and dock to the target, subsequent phases of the mission will require a 3D map for relative pose determination. The target's center of mass will also be required to predict, in conjunction with a motion model, the location of a docking or grasping location on the target in the future. Existing algorithms for mapping an unknown target

have assumed that either the target or the inspector is stationary (Lichter (2005); Augenstein (2011); Tweddle (2013)). However, in realistic cases, both the inspector and the target are moving with similar time constants, and maneuvering of the inspector to obtain new vantage points is both expected and desired. Therefore, existing techniques require fundamental extension.

Lichter (2005), obtains 3D point clouds from a team of static inspectors surrounding an unknown tumbling object. This data allows his algorithm to quickly build an occupancy-grid based map and calculate the object's geometric center. The stationary sensor array subsequently provides measurements of both the orientation and translation of the geometric frame. An unscented Kalman filter (UKF) is used to recursively

¹Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, USA

²Department of Mechanical and Ocean Engineering, Massachusetts Institute of Technology, USA

Corresponding author:

Timothy P. Setterfield, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, USA

Email: tsetterf@alum.mit.edu

estimate the target’s geometric frame orientation, inertial angular velocity, a quaternion parameterization of inertia ratios, and rotation from the true principal axes to the principal geometric axes. A linear Kalman filter is used in parallel for estimation of the translation variables.

Hillenbrand and Lampariello (2005) use simulated range data from a stationary inspector to perform open-loop visual odometry with respect to a passive target which is both rotating and translating. An estimate of target’s center of mass is obtained in a least-squares manner using the visual odometry motion estimates. Although a map is not explicitly presented, the authors suggest that the results could be used to make a geometric model of the target.

Augenstein (2011), estimates the orientation, angular velocity, translation to, and map of a tumbling target with a monocular camera using a hybrid Bayesian estimation and particle filter approach. Although allowing for a moving target, he assumes that the position of the inspector is known at the time of acquisition of all images. For rotational motion, he uses simplified versions of Euler’s equations to predict feature positions resulting from pure rotation for several hypothetical states (particles in a particle filter).

Aghili et al. (2011); Aghili and Su (2016) use a LiDAR sensor to obtain a point cloud from a known but tumbling target. ICP is used to match the point cloud to a 3D CAD model of the target to determine the relative pose from camera to target. In full, the EKF estimates the relative position, velocity, and orientation of the target relative to the inspector, and the body angular velocities, ratios of inertia, principal axes of inertia, and position of the center of mass of the target. Aghili et al. incorporate relative motion of the inspector with respect to the target, driven by orbital mechanics and thrusting; however, their framework has not been applied to targets with unknown 3D structures.

Tweddle (2013) uses a stereo vision system to obtain sparse 3D landmarks of a tumbling target satellite. The inspector satellite is considered inertially static, and a factor graph formulation is used to solve a SLAM problem with inverted dynamics (i.e., static camera and moving scene). Variable nodes in the factor graph are created for the state of the target at each image acquisition, containing the inertial position, velocity, orientation, and angular velocity. Speeded Up Robust Features (SURF) are used as landmarks and are compared globally between all acquired images; landmarks matched in more than one frame are added to the factor graph. Additional variable nodes including the inertia ratios, the orientation of the

principal axes, and the center of mass of the target are added into the factor graph to aid in prediction of the target’s motion into the future. A modified version of iSAM (Kaess et al. (2007)) including factors to incorporate the dynamic target is used to obtain the maximum *a posteriori* estimate for all variables. Tweddle’s algorithm relies on natural motion of the target object and is not applicable when external torques are applied. The algorithm was validated on the ISS using the SPHERES-VERTIGO test platform, yielding good results when the inspector was forced – through control inputs and external sensing – to remain stationary (Tweddle et al. (2014)). Online operation of the algorithm was not possible, with 35 minutes of iSAM batch mode computation required for a 115 image sequence recorded at 2 Hz (Tweddle (2013)).

An outline of the approach taken in this paper is shown in Figure 1. Image data from a 3D visual sensor is processed to determine the range and bearing to the target’s centroid and perform visual odometry with respect to the target; the resulting measurements are passed to trajectory estimation and mapping respectively. Measurements of the inspector’s orientation, angular rate, and relative acceleration are passed to trajectory estimation. Trajectory estimation and mapping are both performed using a smoothing-based estimation approach. The two estimation problems are connected using a novel rotation kinematic factor which enables estimation of the target’s center of mass. The joint estimation of the inspector’s trajectory, the target’s map, and the target’s center of mass, as facilitated by the rotation kinematic factor, is the primary contribution of this paper.

2 Background

2.1 Rotation

Rotation matrices are defined by how they change the frame in which a vector is expressed (i.e., ${}^B\mathbf{t} = {}^B\mathbf{R} {}^A\mathbf{t}$) and are composed as follows.

$${}^C\mathbf{R} = {}^C\mathbf{R} {}^B\mathbf{R} {}^A\mathbf{R} \quad (1)$$

Angle-axis vectors $\boldsymbol{\theta} = \theta \mathbf{a}$ represent both the magnitude and direction of the rotation, where θ is the angle of rotation, and $\mathbf{a}$ is the unit vector along the axis of rotation. The $\wedge$ operator creates a skew-symmetric matrix from an angle-axis vector, whereas

Figure 1. An overview of the approach taken in this paper.

the $^\vee$ operator performs the reverse operation.

$$\theta^\wedge = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}^\vee = \theta \quad (2)$$

The $^\wedge$ operator has the following useful properties.

$$\theta^\wedge \mathbf{t} = \theta \times \mathbf{t} = -\mathbf{t} \times \theta = -\mathbf{t}^\wedge \theta, (\theta^\wedge)^\mathbf{T} = -\theta^\wedge \quad (3)$$

Rotation in three dimensions exists on a manifold called the rotation group $\text{SO}(3)$ and has three degrees of freedom. Although three parameter representations always contain a singularity, they are commonly used for representing uncertainty and performing optimization near the identity rotation. For these applications, the exponential and logarithmic maps can be used to map from angle-axis vectors θ to rotation matrices $\mathbf{R}$ ($\text{so}(3) \Rightarrow \text{SO}(3)$) or rotation matrices $\mathbf{R}$ to angle-axis vectors θ ($\text{SO}(3) \Rightarrow \text{so}(3)$) respectively, where $\text{so}(3)$ is a subspace of $\mathbb{R}^3$ inside the ball of radius π (i.e., $\theta \in \mathbb{R}^3$, $\|\theta\| < \pi$). The exponential map is given

as follows:

$$\begin{aligned} \mathbf{R} &= \text{Exp}(\theta) = \exp(\theta^\wedge) \\ &= \mathbf{I}_3 + \frac{\sin(\theta)}{\theta} \theta^\wedge + \frac{1 - \cos(\theta)}{\theta^2} (\theta^\wedge)^2 \\ &\approx \mathbf{I}_3 + \theta^\wedge \end{aligned} \quad (4)$$

where $\text{Exp}(\theta)$ is used herein as a short form notation for the matrix exponential of a skew symmetric matrix $\exp(\theta^\wedge)$, and the linearization is given by $\mathbf{I}_3 + \theta^\wedge$. Useful properties for algebraic manipulation of the exponential map are given below (Forster et al. (2015)).

$$\mathbf{R} \text{Exp}(\theta) = \text{Exp}(\mathbf{R} \theta) \mathbf{R} \quad (5)$$

$$\text{Exp}(\theta) \mathbf{R} = \mathbf{R} \text{Exp}(\mathbf{R}^\mathbf{T} \theta) \quad (6)$$

The logarithmic map is given as follows:

$$\theta = \text{Log}(\mathbf{R}) = \log(\mathbf{R})^\vee = \frac{\theta}{2 \sin(\theta)} (\mathbf{R} - \mathbf{R}^\mathbf{T})^\vee \quad (7)$$

$$\theta = \arccos\left(\frac{\text{tr}(\mathbf{R}) - 1}{2}\right) \quad (8)$$

where $\text{Log}(\mathbf{R})$ is used herein as a short form notation for the angle-axis vector from matrix logarithm $\log(\mathbf{R})^\vee$.

Small angle-axis vectors $\delta\theta \in \text{so}(3)$ can be used to perturb the rotation. These vectors $\delta\theta$ can be sampled from a Gaussian distribution to represent uncertainty in rotation with respect to the mean value $\bar{\mathbf{R}}$.

$$\mathbf{R} = \bar{\mathbf{R}} \text{Exp}(\delta\theta), \delta\theta = \text{Log}(\bar{\mathbf{R}}^\mathbf{T} \mathbf{R}) \quad (9)$$

$$\delta\theta \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \Sigma_R) \quad (10)$$

2.2 Pose

The pose of coordinate frame B refers to the combination of its rotation and translation with respect to another coordinate frame A . It can be compactly represented by a 4×4 pose matrix ${}^B_A\mathbf{P}$.

$${}^B_A\mathbf{P} = \begin{bmatrix} {}^A_B\mathbf{R} & {}^A\mathbf{t}_{AtoB} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (11)$$

Poses can be inverted as follows.

$${}^A_B\mathbf{P} = {}^B_A\mathbf{P}^{-1} = \begin{bmatrix} {}^A_B\mathbf{R}^\mathbf{T} & -{}^A_B\mathbf{R}^\mathbf{T} {}^A\mathbf{t}_{AtoB} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (12)$$

Poses are composed in the opposite order to rotation matrices.

$${}^C_A\mathbf{P} = {}^B_A\mathbf{P} {}^C_B\mathbf{P} \quad (13)$$

Small vectors $\delta\mathbf{p} = [\delta\theta^\mathbf{T} \delta\mathbf{t}^\mathbf{T}]^\mathbf{T}$ can be used to perturb the pose. Here, the first three terms of the vector $\delta\mathbf{p}$ are

the angle-axis perturbation $\delta\theta$ as in Equation 9, and the final three terms of the vector are the translation perturbation δt .

$$\begin{aligned} {}^B_A\mathbf{P} &= {}^B_A\overline{\mathbf{P}} \delta\mathbf{P} = \begin{bmatrix} {}^A_B\overline{\mathbf{R}} & {}^A\overline{\mathbf{t}}_{AtoB} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \text{Exp}(\delta\theta) & \delta t \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^A_B\overline{\mathbf{R}} \text{Exp}(\delta\theta) & {}^A\overline{\mathbf{t}}_{AtoB} + {}^A_B\overline{\mathbf{R}} \delta t \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \end{aligned} \quad (14)$$

$$\delta\mathbf{p} = [\delta\theta^T \ \delta t^T]^T \sim \mathcal{N}(\mathbf{0}_{6 \times 1}, \Sigma_P) \quad (15)$$

2.3 Smoothing-Based Estimation

Smoothing-based estimation is a batch method which attempts to find the *maximum a posteriori* estimate of a set of random variables Θ . This is accomplished by finding optimal values Θ^* that maximize the joint probability of Θ and noisy measurements $\mathbf{Z}$. In some instances, a set of commanded control inputs $\mathbf{U}$ is also included to parameterize the problem; herein these control inputs are treated as random variables with low covariance. Thus the smoothing-based estimation problem can be defined as finding the optimal vector Θ^* as follows.

$$\begin{aligned} \Theta^* &= \underset{\Theta}{\operatorname{argmax}} P(\Theta, \mathbf{Z}, \mathbf{U}) \\ &= \underset{\Theta}{\operatorname{argmax}} P(\mathbf{Z}, \mathbf{U} | \Theta) P(\Theta) \end{aligned} \quad (16)$$

Assuming that each measurement is Gaussian, and using the monotonic natural logarithm function, this problem can be reduced to the problem of minimizing the sum of several probabilistic factors (Kaess et al. (2012)):

$$\begin{aligned} \Theta^* &= \underset{\Theta}{\operatorname{argmin}} -\log(P(\mathbf{Z}, \mathbf{U} | \Theta) P(\Theta)) \\ &= \underset{\Theta}{\operatorname{argmin}} \frac{1}{2} \sum_k \|\mathbf{z}_k - h(\theta_k)\|_{\Sigma_k}^2 \end{aligned} \quad (17)$$

where $\mathbf{z}_k$ is the measurement value, $h(\theta_k)$ denotes the measurement model as a function of a subset of Θ , and Σ_k is the measurement's covariance.

Using this framework, several Gaussian measurements can be incorporated so long as there exists a measurement model h in terms of the variables Θ and an estimate of the measurement covariance Σ_k . Gaussian prior belief can also be incorporated using the simple measurement model $h(\theta_k) = \theta_k$. For real-world problems, the resultant optimization problem is typically a large, nonlinear, non-convex problem. The state-of-the-art incremental smoothing and mapping technique (iSAM2) exploits the structure and sparsity of the problem to obtain efficient solutions which can commonly be computed in real time (Kaess et al.

(2012)). The open source GTSAM software (Dellaert and Beall (2017)), which implements iSAM2, is used for smoothing-based estimation in this paper.

3 Review of Trajectory Estimation

This section reviews the trajectory estimation technique presented in Setterfield et al. (2017) which is used herein to determine the trajectory of the inspector satellite in a manner that is independent of the target's rotational motion.

An overview of the requisite coordinate frames is shown in Figure 2. The inspector's body frame B is located at its center of mass, and aligned with its principal axes of inertia; calibrated gyroscopes measure the angular velocity in this frame. The pose of B is estimated with respect to the world navigation frame W ; the frame W is a target-centered inertial coordinate frame, meaning that its origin is attached to the center of mass of the target, but it is rotationally inertial (i.e., fixed with respect to the stars). The camera frame C is located at a known position and orientation on the inspector. The target body frame $\underline{B}$ is attached to its center of mass and aligned with its principal axes.

Figure 2. Measurements and coordinate frames required for trajectory estimation and mapping.

Using the technique from Setterfield et al. (2017), the *maximum a posteriori* estimate for the full 6-DOF trajectory of the inspector satellite is estimated. Specifically, this involves solving for the most probable navigation state vector $\mathbf{x}$ over a sequence of time steps $\mathcal{I} = \{0, 1, \dots, h, \dots, i, j, \dots, N\}$:

$$\begin{aligned} \mathbf{x}_{\mathcal{I}} &= \{{}^W_B\mathbf{R}, {}^W\mathbf{t}_{WtoB}, {}^W\mathbf{v}_B, \mathbf{b}^g, \mathbf{b}^a\}_{\mathcal{I}} \\ \mathbf{x} &\in \{\text{SO}(3), \mathbb{R}^3, \mathbb{R}^3, \mathbb{R}^3, \mathbb{R}^3\} \end{aligned} \quad (18)$$

where ${}^W_B\mathbf{R}$ is the orientation of the inspector body frame B in the world frame W , ${}^W\mathbf{t}_{WtoB}$ is the position

of the inspector center of mass in the world frame, ${}^W\mathbf{v}_B$ is the velocity of the inspector center of mass in the world frame, $\mathbf{b}^g$ is the gyroscope bias, and $\mathbf{b}^a$ is the acceleration bias.

Measurements of range $\mathbf{Z}^r = \|\mathbf{t}_{CtoW}\|_{\mathcal{I}}$ and bearing $\mathbf{Z}^b = \{{}^C\mathbf{t}_{CtoW} / \|\mathbf{t}_{CtoW}\|_{\mathcal{I}}\}_{\mathcal{I}}$ from the inspector to the target's approximate centroid are obtained using an averaged depth map from radar, LiDAR, or stereo cameras. Measurements of orientation $\mathbf{Z}^s = {}^W_B\mathbf{R}_{\mathcal{I}}$ of the inspector in the inertial frame are provided by a star tracker. High frequency measurements of angular rate $\mathbf{Z}^g = \{\mathbf{z}_{0:1}^g, \dots, \mathbf{z}_{i:j}^g, \dots, \mathbf{z}_{N-1:N}^g\}$ (where $\mathbf{z}_{i:j}^g = \{{}^B\boldsymbol{\omega}_{B_{i,1}}, {}^B\boldsymbol{\omega}_{B_{i,2}}, \dots, {}^B\boldsymbol{\omega}_{B_{j,0}}\}$) are collected between time steps in $\mathcal{I}$ using a three-axis gyroscope. Similarly, high frequency acceleration "measurements" $\mathbf{Z}^a = \{\mathbf{z}_{0:1}^a, \dots, \mathbf{z}_{i:j}^a, \dots, \mathbf{z}_{N-1:N}^a\}$ (where $\mathbf{z}_{i:j}^a = \{\mathbf{u}_{i,1}^f, \mathbf{u}_{i,2}^f, \dots, \mathbf{u}_{j,0}^f\}/m$) are obtained by dividing the commanded forces $\mathbf{u}^f$ by the spacecraft mass m . Although not performed herein, accelerations from relative orbital dynamics from the Clohessy-Wiltshire equations can be incorporated into these acceleration "measurements".

An overview of the requisite measurements is shown in Figure 2. To determine the full 6-DOF trajectory of the inspector satellite, a smoothing-based estimation approach of the type in Equation 17 is adopted, as represented by the factor graph in Figure 3a. A prior factor ϕ_{t_W} with a very low covariance is placed on the position of the origin of W to enforce its location at the origin, or elsewhere if convenient. A prior belief on initial state is obtained from the long-range estimation scheme and applied as a prior factor ϕ_{x_0} . Range and bearing measurements of the target's geometric centroid from the inspector are considered approximate range and bearing measurements to the origin of B , and input into the problem as a range and bearing factor ρ . Star tracker measurements are applied as partial unary factors σ to the orientation component ${}^W_B\mathbf{R}$ of the inspector's state. Measurements of angular rate and acceleration are applied using preintegrated inertial odometry factor ψ described in Forster et al. (2015). It is assumed that orientation and range and bearing measurements arrive at a lower frequency than angular rate and acceleration measurements.

Observability of the state can be seen intuitively given the measurements. The range measurement from inspector to the target restricts the location of the inspector to a sphere around the target. The star tracker measurement of orientation then restricts the three rotational degrees of freedom of the inspector. The remaining two degrees of freedom are restricted by the bearing measurement to the target, which places the inspector satellite at a specific point on the sphere.

The observability of velocity follows from the time rate of change of position, and the observability of gyroscope and acceleration biases follows from the accurate knowledge of orientation and the observability of velocity respectively.

4 Mapping

The objective of this section is to create a 3D map of the target comprised of visual feature points capable of being redetected in subsequent phases of the mission. To accomplish this, visual odometry is first performed to determine the motion of the inspector satellite relative to the target object. The *maximum a posteriori* pose graph for relative motion is then developed using smoothing-based estimation techniques. This allows the map of the target object to be created by reprojecting visual measurements into a target-fixed coordinate frame.

4.1 Required Measurements and Coordinate Frames

To perform pose graph based mapping of the target object, a set of camera measurements $\mathbf{Z}^c$ over a sequence of time steps $\mathcal{I}$ is required. Herein, camera measurements $\mathbf{Z}^c$ are 3D locations of visual feature points of interest with an attached descriptor; the descriptor associated with each point allows the same feature point to be redetected in other images.

The coordinate frames used in this section are identical to those introduced in Section 3, with the addition of the geometric frame $\underline{G}$. This frame, shown in Figure 2, is fixed to the target object. By convention, its orientation is aligned with the inspector's body frame B at the instant when the first image is taken (i.e., $\frac{G_0}{B_0}\mathbf{R} = \mathbf{I}_3$), and its origin is at the center of all inlying visual features from the first image. Inlying features are those which both match features in the second image and contribute to the visual odometry solution (see Section 4.2).

4.2 Visual Odometry

To form a pose graph, it is desired that the pose change of the camera frame C between two time steps i and $j = i + 1$ be estimated with reference to the target-fixed geometric frame $\underline{G}_i \equiv \underline{G}_j \forall i, j \in \mathcal{I}$. This is equivalent to treating the target object (and consequently the geometric frame $\underline{G}$) as "static" for this section, so that conventional visual odometry can be performed. The 3D image data acquired at time steps i and j must contain overlapping parts of the scene to facilitate motion estimation.

Figure 3. (Bottom (a)) The trajectory factor graph. (Top (b)) The mapping pose graph. (Middle) The rotational kinematic factors connecting the trajectory factor graph with the mapping pose graph. Circular nodes, representing random variables to be solved for are connected via edges to square nodes, representing probabilistic factors.

When performing visual odometry, the choice of algorithm depends largely on the sensor being used. Herein, the discussion is limited to 3D sensors such as stereo cameras, LiDAR, time of flight cameras, etc., although visual odometry has also been demonstrated using monocular cameras (Davison et al. (2007); Engel et al. (2014)). When LiDAR or time of flight cameras are used, a dense 3D point cloud of the scene is obtained. In the case where this 3D scan is not accompanied by an image, an iterative closest point (ICP) type method can be used to solve for the relative motion between frames (Besl and McKay (1992)). The work herein uses images from a stereo camera, so a feature-based visual odometry technique will be discussed in detail.

When two stereo images of the same scene are taken, points of interest can be extracted and matched between all four individual images: that is left and right images in frame C_i and left and right images in frame C_j . Left-to-right feature correspondences will have image coordinates (u_L, v_L) and (u_R, v_R) in the left and right images respectively. Correspondences that have similar vertical coordinates (i.e., satisfy the epipolar constraint $v_L \approx v_R$) are triangulated to get three-dimensional points. The coordinates on the image plane can be summarized by stereo coordinates s ; from this, the 3D location of the point in the left

camera's frame ${}^{C_i}\mathbf{t}_P$ is given as follows (see Figure 4a):

$$\mathbf{s} = \begin{bmatrix} u_L \\ u_R \\ v_L \end{bmatrix}, \quad {}^{C_i}\mathbf{t}_P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} t_x \frac{(u_L - c_{u_L})}{u_L - u_R} \\ t_x \frac{(v_L - c_{v_L})}{u_L - u_R} \\ \frac{f t_x}{u_L - u_R} \end{bmatrix} \quad (19)$$

where (x, y, z) are the coordinates of the point with respect to the left camera's focal point, the z -axis is the optical axis, t_x is the stereo camera baseline, f is the camera's focal length, (c_{u_L}, c_{v_L}) are the image coordinates of the left image's optical center, and the image has been rectified so that the left and right images have the same focus and center (i.e., $f_R = f_L = f$, $c_{u_R} = c_{u_L}$, and $c_{v_R} = c_{v_L}$).

These successfully triangulated points are then matched from frame-to-frame to facilitate motion estimation. Herein, motion estimation is derived as if the scene remains static, and the stereo camera frame moves from frame C_i to frame C_j . The rotation ${}^{C_j}\mathbf{R}_{C_i}$ and translation ${}^{C_j}\mathbf{t}_{C_i \text{ to } C_j}$ of the stereo camera that best represent the camera motion in three dimensions is sought. A measured point in frame C_i , ${}^{C_i}\mathbf{t}_P$, can be transformed into frame C_j , creating a prediction ${}^{C_j}\mathbf{t}_{P_p}$ of the position of the corresponding measured point in frame C_j , ${}^{C_j}\mathbf{t}_P$.

$$\begin{aligned} {}^{C_j}\mathbf{t}_{P_p} &= {}^{C_j}\mathbf{R}_{C_i} ({}^{C_i}\mathbf{t}_P - {}^{C_i}\mathbf{t}_{C_i \text{ to } C_j}) \\ &= {}^{C_j}\mathbf{R}_{C_i} {}^{C_i}\mathbf{t}_P - {}^{C_j}\mathbf{t}_{C_i \text{ to } C_j} \end{aligned} \quad (20)$$

Figure 4. Triangulation with a stereo camera and its associated uncertainty. (a) Definition of the variables required in triangulation. (b) Uncertainty in the least-squares visual odometry formulation. (c) Uncertainty in the maximum-likelihood visual odometry formulation.

The error in prediction of this point ϵ can be found by comparing the predicted point position ${}^{C_j}\mathbf{t}_{P_p}$ to the corresponding measured point position ${}^{C_j}\mathbf{t}_P$.

$$\begin{aligned}\epsilon &= {}^{C_j}\mathbf{t}_P - {}^{C_j}\mathbf{t}_{P_p} \\ &= {}^{C_j}\mathbf{t}_P - \left({}^{C_j}\mathbf{R} {}^{C_i}\mathbf{t}_P - {}^{C_j}\mathbf{t}_{C_i \text{ to } C_j} \right)\end{aligned}\quad (21)$$

In this paper, the following problem is solved twice, in a similar manner to as on the Mars Exploration Rovers (Maimone et al. (2007)).

$$\left\{ {}^{C_j}\mathbf{R}^*, {}^{C_j}\mathbf{t}_{C_i \text{ to } C_j}^* \right\} = \underset{{}^{C_j}\mathbf{R}, {}^{C_j}\mathbf{t}_{C_i \text{ to } C_j}}{\text{argmin}} \sum_{l=1}^M \|\epsilon_l\|_{\Sigma_{\epsilon_l}}^2 \quad (22)$$

$$\epsilon_l \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \Sigma_{\epsilon_l}) \quad (23)$$

Firstly, the problem is solved with $\Sigma_{\epsilon_l} = \mathbf{I}_3$ using the closed form solution of absolute orientation from Horn (1987) wrapped with Random Sampling and Consensus (RANSAC) (Fischler and Bolles (1981)). this step determines the largest subset of inlying features that passes a “rigidity check” and provides a least-squares estimate of $\left\{ {}^{C_j}\mathbf{R}, {}^{C_j}\mathbf{t}_{C_i \text{ to } C_j} \right\}$ that is not weighted by feature position uncertainty (see Figure 4b).

Secondly, a maximum-likelihood estimate is obtained using the method from Matthies (1989) where Σ_{ϵ_l} is the covariance of each feature’s error

ϵ . In stereo vision, points close to the camera tend to have less error than those that are far away; this formulation properly accounts for their relative uncertainty (see Figure 4c). The maximum-likelihood estimator is initialized with the rotation, translation, and the $M = M_{in}$ inlying points found in the least-squares step. The final result of the visual odometry algorithm is the relative camera pose $\left\{ {}^{C_j}\mathbf{P}^* \right\}_{\underline{G}}$ from frame C_i to frame C_j together with an associated uncertainty Σ_{P_C} (Setterfield (2017a)).

$$\left\{ {}^{C_j}\mathbf{P}^* \right\}_{\underline{G}} = \begin{bmatrix} \left\{ {}^{C_j}\mathbf{R}^* \right\}_{\underline{G}} & \left\{ {}^{C_i}\mathbf{t}_{C_i \text{ to } C_j}^* \right\}_{\underline{G}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (24)$$

$$P\left(\left\{ {}^{C_j}\mathbf{P}^* \right\}_{\underline{G}} \mid \mathbf{z}_{C_i}^c, \mathbf{z}_{C_j}^c\right) \sim \mathcal{N}\left(\left\{ {}^{C_j}\mathbf{P}^* \right\}_{\underline{G}}, \Sigma_{P_C}\right) \quad (25)$$

$$\Sigma_{P_C} = \mathbf{E}\left[\begin{bmatrix} \delta\theta_C^T & \delta\mathbf{t}_C^T \end{bmatrix}^T \begin{bmatrix} \delta\theta_C^T & \delta\mathbf{t}_C^T \end{bmatrix}\right] \quad (26)$$

where $\delta\theta_C$ is the $\text{so}(3)$ rotation uncertainty, and $\delta\mathbf{t}_C$ is the $\mathbb{R}^3$ translation uncertainty. The $\underline{G}$ subscript on pose change $\left\{ {}^{C_j}\mathbf{P}^* \right\}_{\underline{G}}$ emphasizes the fact that frame $\underline{G}$ is considered the static reference for the relative camera poses.

Visual odometry is illustrated in Figure 5. With $\underline{G}$ established as a static reference, a conventional visual pose graph approach is pursued. Frame-to-frame feature descriptor matching is attempted between each new image and the previous image as well as each new image with a randomly selected previous image (loop closure). Loop closures improve the quality of the map by preventing drift caused by compounding errors in visual odometry.

4.3 Smoothing-Based Mapping

In this section, the time history of inspector body poses in the geometric frame (i.e., ${}^{B_T}\mathbf{P}$ for a series of time steps $\mathcal{I}$) is estimated.

4.3.1 Pose Graph Initialization Based on the definition of coordinate frame $\underline{G}$ in Section 4.1, the initial pose of the inspector in the target geometric frame $\underline{G}$ is determined as follows:

$${}^{B_0}\mathbf{P} = \begin{bmatrix} \mathbf{I}_3 & {}^{G_0}\mathbf{t}_{G_0 \text{ to } B_0} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (27)$$

$${}^{G_0}\mathbf{t}_{G_0 \text{ to } B_0} = -{}^B\mathbf{R} \left(\frac{1}{M_{in}} \sum_{l=1}^{M_{in}} {}^{C_0}\mathbf{t}_{P_l} \right) - {}^B\mathbf{t}_{B \text{ to } C} \quad (28)$$

where ${}^B\mathbf{R}$ is the known rotation from camera frame C to inspector body frame B .

4.3.2 Pose Changes in the Inspector Body Frame Herein, the pose of the inspector body frame B (as opposed to the camera frame C) is tracked. All

Figure 5. Measurements used and pose constraints obtained using visual odometry. Detected visual feature points on the target are filled with the color that matches the color of the camera frame C in which they were observed. Feature points detected from more than one vantage point are outlined in the color of the second camera frame observation and are used in visual odometry to obtain the pose change between temporally adjacent camera frames (e.g., $\{C_1^c \mathbf{P}\}_G$). Matches between temporally non-adjacent camera frames (e.g., $\{C_N^c \mathbf{P}\}_G$) are used for loop closure.

camera pose changes $\{C_j^c \mathbf{P}\}_G$ from visual odometry are transformed into measurements of inspector body pose changes $\{B_j^b \mathbf{P}\}_G$ using the known transformation $C_B^c \mathbf{P}$.

$$\{B_j^b \mathbf{P}\}_G = C_B^c \mathbf{P} \{C_j^c \mathbf{P}\}_G C_B^c \mathbf{P}^{-1} \quad (29)$$

The covariance Σ_{P_C} must also be transformed into covariance in $\{B_j^b \mathbf{P}\}_G$, denoted Σ_{P_B} , as outlined in Setterfield (2017a).

4.3.3 Pose Prior Factor A prior Gaussian belief $\mathbf{z}^p = B_0^b \mathbf{P}$ with the value given in Equation 27 is applied directly to the initial pose. A low covariance Σ_p is used to enforce the initialization convention. The prior belief $\mathbf{z}^p$ has a rotation component $\mathbf{z}^{pr} = \mathbf{I}_3$ and a translation component $\mathbf{z}^{pt} = G_0 \mathbf{t}_{G_0 \text{ to } B_0}$, and is applied using the prior factor ϕ_{P_G} .

$$P(\mathbf{z}^p | \theta_k = \{B_0^b \mathbf{P}\}) \sim \mathcal{N}(h(\theta_k), \Sigma_p) \\ \mathbf{z}^p \in \{\text{SO}(3), \mathbb{R}^3\} \quad (30) \\ \phi_{P_G} = \left\| \begin{bmatrix} \text{Log} \left(\frac{G_0}{B_0} \mathbf{R}^T \mathbf{z}^{pr} \right) \\ \frac{G_0}{B_0} \mathbf{R}^T [\mathbf{z}^{pt} - G_0 \mathbf{t}_{G_0 \text{ to } B_0}] \end{bmatrix} \right\|_{\Sigma_p}^2$$

4.3.4 Pose Constraint Factor The motion measurement $\mathbf{z}^m = \{B_j^b \mathbf{P}\}_G$ has a rotation component

$\mathbf{z}^{mr} = \{B_j^b \mathbf{R}\}_G$ and a translation component $\mathbf{z}^{mt} = \{B_j^b \mathbf{t}_{B_i \text{ to } B_j}\}_G$ and is applied as a vision factor ν :

$$P(\mathbf{z}^m | \theta_k = \{B_j^b \mathbf{P}, \frac{B_j}{G_j} \mathbf{P}\}) \sim \mathcal{N}(h(\theta_k), \Sigma_{P_B}) \\ \mathbf{z}^m \in \{\text{SO}(3), \mathbb{R}^3\} \quad (31)$$

$$\nu = \left\| \begin{bmatrix} \text{Log} \left(\left(\frac{B_j}{G_j} \mathbf{R} \frac{G_i}{B_i} \mathbf{R} \right)^T \mathbf{z}^{mr} \right) \\ \left(\frac{B_j}{G_j} \mathbf{R} \frac{G_i}{B_i} \mathbf{R} \right)^T [\mathbf{z}^{mt} - \frac{B_i}{G_i} \mathbf{R} (\Delta \mathbf{t}_{G \text{ to } B})] \end{bmatrix} \right\|_{\Sigma_{P_B}}^2$$

where $\Delta \mathbf{t}_{G \text{ to } B} = \frac{G_j}{G_i} \mathbf{t}_{G_j \text{ to } B_j} - \frac{G_i}{G_i} \mathbf{t}_{G_i \text{ to } B_i}$.

4.3.5 Mapping Pose Graph The problem of estimating the maximum *a posteriori* time series of inspector poses in the target object's geometric frame $\frac{B_T}{G_T} \mathbf{P}$ is formulated as a batch optimization problem of the type described in Equation 17. The joint probability distribution of all poses $\frac{B_T}{G_T} \mathbf{P}$ and measurements $\mathbf{Z}^c$ is represented by the factor graph depicted in Figure 3b; note that without the central factors this graph is completely separate from the trajectory factor graph.

5 Estimation of the Target's Center of Mass

In this section, pose estimates of the inspector body frame B with reference to the world frame W from Section 3 are considered together with pose estimates of the inspector body frame B with reference to the target-fixed frame G from Section 4 in order to determine the center of mass of the target object. Firstly, pose estimates for the target object's body-fixed geometric frame G are calculated with reference to the world frame W at each time step $i \in \mathcal{I}$.

$$\frac{G_i}{W} \mathbf{P} = \frac{B_i}{W} \mathbf{P} \frac{G_i}{B_i} \mathbf{P} = \frac{B_i}{W} \mathbf{P} \frac{B_i}{G_i} \mathbf{P}^{-1} \quad (32)$$

Since the target object is assumed to be a rigid body that undergoes only rotational motion, the motion of the target-fixed geometric frame G is constrained by its kinematics. Recalling that the target's body frame B is situated at its center of mass, the translation from the geometric frame to body frame $\frac{G}{B} \mathbf{t}_{G \text{ to } B}$ is constant. Using the summation of vectors illustrated in Figure 6, a rotational kinematic constraint $\epsilon_{\kappa_i} \approx \mathbf{0}_{3 \times 1}$ is constructed between temporally adjacent time steps i and $j = i + 1$.

$$\mathbf{0}_{3 \times 1} \approx \epsilon_{\kappa_i} = -\mathbf{t}_{G_i \text{ to } B_i} + \mathbf{t}_{G_i \text{ to } B_i} + \mathbf{t}_{B_i \text{ to } B_j} \cdots \\ - \mathbf{t}_{G_j \text{ to } B_j} + \mathbf{t}_{G_j \text{ to } B_j} \quad (33)$$

Rotating all translations into the world frame W , using $\mathbf{t}_{B_i \text{ to } B_j} = \mathbf{t}_{W \text{ to } B_j} - \mathbf{t}_{W \text{ to } B_i}$, and recognizing that for

Figure 6. The vector summation that creates the rotational kinematic constraint used in determining the target's center of mass.

the rigid target object $\underline{G}_i \mathbf{t}_{\underline{G}_i \text{to} \underline{B}_i} \equiv \underline{G}_j \mathbf{t}_{\underline{G}_j \text{to} \underline{B}_j} \equiv \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}}$, the constraint ϵ_{κ} can be rewritten as follows.

$$\begin{aligned} \epsilon_{\kappa_i} = & \frac{W}{B_i} \mathbf{R}_{\underline{G}_i}^{B_i} \mathbf{R}_{\underline{G}_i} (\underline{G}_i \mathbf{t}_{\underline{G}_i \text{to} \underline{B}_i} - \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}}) \cdots \\ & + \frac{W}{B_j} \mathbf{R}_{\underline{G}_j}^{B_j} \mathbf{R}_{\underline{G}_j} (\underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}} - \underline{G}_j \mathbf{t}_{\underline{G}_j \text{to} \underline{B}_j}) \cdots \\ & + ({}^W \mathbf{t}_{W \text{to} \underline{B}_j} - {}^W \mathbf{t}_{W \text{to} \underline{B}_i}) \end{aligned} \quad (34)$$

The rotational kinematic constraint created above can be used to solve for the target's center of mass in a linear, least-squares manner (Section 5.1), or in a weighted least-squares manner as part of a factor graph (Section 5.2).

5.1 Linear Solution

A linear least-squares solution for the center of mass can be obtained by rearranging Equation 34 into the form $\mathbf{A}_i \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}} = \mathbf{b}_i$.

$$\begin{aligned} \left(\frac{W}{B_i} \mathbf{R}_{\underline{G}_i}^{B_i} \mathbf{R}_{\underline{G}_i} - \frac{W}{B_j} \mathbf{R}_{\underline{G}_j}^{B_j} \mathbf{R}_{\underline{G}_j} \right) \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}} = & \cdots \\ & ({}^W \mathbf{t}_{W \text{to} \underline{B}_j} - {}^W \mathbf{t}_{W \text{to} \underline{B}_i}) + \frac{W}{B_i} \mathbf{R}_{\underline{G}_i}^{B_i} \mathbf{R}_{\underline{G}_i} \underline{G}_i \mathbf{t}_{\underline{G}_i \text{to} \underline{B}_i} \cdots \\ & - \frac{W}{B_j} \mathbf{R}_{\underline{G}_j}^{B_j} \mathbf{R}_{\underline{G}_j} \underline{G}_j \mathbf{t}_{\underline{G}_j \text{to} \underline{B}_j} \end{aligned} \quad (35)$$

$$\mathbf{A}_i \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}} = \mathbf{b}_i \quad (36)$$

These equations can be stacked and the optimal translation $\underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}}^*$ can be solved for in a linear least-squares sense.

$$\begin{aligned} \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}}^* = & \underset{\underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{A}_0 \\ \vdots \\ \mathbf{A}_{N-1} \end{bmatrix} \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}} - \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_{N-1} \end{bmatrix} \right\|^2 \\ = & \underset{\underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}}}{\operatorname{argmin}} \left\| \mathbf{A} \underline{G} \mathbf{t}_{\underline{G} \text{to} \underline{B}} - \mathbf{b} \right\|^2 \\ = & (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \end{aligned} \quad (37)$$

Note that this problem is poorly conditioned (i.e., high condition number) along the target's axis of rotation when the axis of rotation remains constant or nearly constant over the time steps $\mathcal{I}$. This occurs for datasets over short time intervals, or for scenarios in which the target object is undergoing single-axis rotation. When the condition number is high, the rotation axis can be identified, but the specific point on the rotation axis that coincides with the center of mass is unobservable.

5.2 Rotation Kinematic Factor

The rotational kinematic constraint defined in Equation 34 can be used to create a probabilistic rotation kinematic factor. Deviations in the value of ϵ_{κ_i} from $\mathbf{0}_{3 \times 1}$ can come from two sources. The first is actual translation of the center of mass $\underline{B}$ of the target object between time steps i and $j = i + 1$. Since the world frame W is attached to the target's center of mass $\underline{B}$, and the relative orbital dynamics between the inspector and the target are assumed to be well modeled, this translation could only come from external forces such as solar pressure, air drag, micrometeoroid impact, or unbalanced application of thruster forces. The second potential source of deviation is variation in the measured range and bearing to the target's centroid. In Equation 34, the term $({}^W \mathbf{t}_{W \text{to} \underline{B}_j} - {}^W \mathbf{t}_{W \text{to} \underline{B}_i})$ relies on the estimates of inspector position in the world frame W , which rely on range and bearing measurements to the target's visual centroid (see Section 3). As the vantage point of the camera changes, it is also possible that the target's perceived visual centroid moves in inertial space. This motion may or may not reflect actual relative motion, but will cause errors in ϵ_{κ_i} . Since time steps i and j are temporally adjacent this "vantage point" error is small in magnitude. Inspector navigation states $\mathbf{x}$ are also attached by an inertial odometry factor (described in Section 3) which provides an independent measure of the inspector's motion; this factor will also minimize the effect of "vantage point" error.

The rotational kinematic constraint $\mathbf{z}^\kappa = \boldsymbol{\epsilon}_{\kappa_i} = \mathbf{0}_{3 \times 1}$ is applied as a Gaussian belief with covariance $\boldsymbol{\Sigma}_\kappa$. The value of $\boldsymbol{\Sigma}_\kappa$ can be set and estimated *a priori* based on simulations of the expected shape of the target object and properties of the visual sensor. Alternatively, the least-squares solution from Section 5.1 can be obtained, and the variance of the residuals of $\boldsymbol{\epsilon}_{\kappa_i}$ can be used to determine the order of magnitude of $\boldsymbol{\Sigma}_\kappa$. Note that since errors $\boldsymbol{\epsilon}_{\kappa_i}$ are equally likely in any direction and are uncorrelated between directions, $\boldsymbol{\Sigma}_\kappa$ is diagonal. The novel rotation kinematic factor κ can then be defined as follows.

$$\begin{aligned} P(\mathbf{z}^\kappa \mid \boldsymbol{\theta}_k) &\sim \mathcal{N}(h(\boldsymbol{\theta}_k), \boldsymbol{\Sigma}_\kappa) \\ \mathbf{z}^\kappa &\in \mathbb{R}^3, \quad \kappa = \|\boldsymbol{\epsilon}_\kappa\|_{\boldsymbol{\Sigma}_\kappa}^2 \\ \boldsymbol{\theta}_k &= \left\{ {}^{B_i}\mathbf{P}, {}^{B_i}\mathbf{P}, {}^{B_j}\mathbf{P}, {}^{B_j}\mathbf{P}, {}^G\mathbf{t}_{G\text{to}B} \right\} \end{aligned} \quad (38)$$

To perform efficient nonlinear estimation using the GTSAM 4.0 open source smoothing and mapping software library used in this paper, the Jacobians of the error term $\boldsymbol{\epsilon}_\kappa$ with respect to each of the five variables in $\boldsymbol{\theta}_k$ are required. Note that Jacobians of pose variables consist of side-by-side Jacobians with respect to rotation and translation (i.e., $\mathbf{H}_P = [\mathbf{H}_R \ \mathbf{H}_t]$). For these derivations, it is helpful to define the terms χ_i and χ_j .

$$\chi_i = {}^W\mathbf{R}_{B_i} {}^{B_i}\mathbf{R}_{G_i} ({}^{G_i}\mathbf{t}_{G_i\text{to}B_i} - {}^G\mathbf{t}_{G\text{to}B}) \quad (39)$$

$$\chi_j = {}^{W_j}\mathbf{R}_{B_j} {}^{B_j}\mathbf{R}_{G_j} ({}^{G_j}\mathbf{t}_{G_j\text{to}B_j} - {}^G\mathbf{t}_{G\text{to}B}) \quad (40)$$

First, consider the Jacobian of $\boldsymbol{\epsilon}_\kappa$ with respect to the rotational component of ${}^{B_i}\mathbf{P}$. Jacobians with respect to rotation in SO(3) are taken by perturbing the rotation by a small angle-axis rotation vector $\boldsymbol{\varphi}$ as in Equation 15, and then taking the first derivative with respect to the perturbation. The properties of the $\wedge$ operator from Equation 3 and the adjoint property of rotation from Equation 5 are used below.

$$\begin{aligned} \mathbf{H}_{1_R} &= \frac{\partial \boldsymbol{\epsilon}_\kappa}{\partial {}^W\mathbf{R}_{B_i}} = \frac{\partial \boldsymbol{\epsilon}_\kappa}{\partial \boldsymbol{\varphi}} \\ &= \frac{\partial}{\partial \boldsymbol{\varphi}} \left[{}^W\mathbf{R}_{B_i} \text{Exp}(\boldsymbol{\varphi}) {}^{B_i}\mathbf{R}_{G_i} ({}^{G_i}\mathbf{t}_{G_i\text{to}B_i} - {}^G\mathbf{t}_{G\text{to}B}) \right] \\ &= \frac{\partial}{\partial \boldsymbol{\varphi}} \left[\text{Exp}({}^W\mathbf{R}_{B_i} \boldsymbol{\varphi}) \chi_i \right] \\ &\approx \frac{\partial}{\partial \boldsymbol{\varphi}} \left[\left(\mathbf{I}_3 + ({}^W\mathbf{R}_{B_i} \boldsymbol{\varphi})^\wedge \right) \chi_i \right] \\ &= \frac{\partial}{\partial \boldsymbol{\varphi}} \left[\chi_i - \chi_i^\wedge {}^W\mathbf{R}_{B_i} \boldsymbol{\varphi} \right] \\ &\approx -\chi_i^\wedge {}^W\mathbf{R}_{B_i} \end{aligned} \quad (41)$$

Then, take the Jacobian of $\boldsymbol{\epsilon}_\kappa$ with respect to the translational component of ${}^{B_i}\mathbf{P}$. Jacobians with

respect to the translation portion of a pose are taken by perturbing the translation by $\mathbf{R} \boldsymbol{\delta t}$ as in Equation 15, and then taking the derivative with respect to the perturbation.

$$\begin{aligned} \mathbf{H}_{1_t} &= \frac{\partial \boldsymbol{\epsilon}_\kappa}{\partial \boldsymbol{\delta t}} = \frac{\partial}{\partial \boldsymbol{\delta t}} \left({}^W\mathbf{t}_{W\text{to}B_j} - ({}^W\mathbf{t}_{W\text{to}B_i} + {}^W\mathbf{R}_{B_i} \boldsymbol{\delta t}) \right) \\ &= -{}^W\mathbf{R}_{B_i} \end{aligned} \quad (42)$$

The Jacobian of $\boldsymbol{\epsilon}_\kappa$ with respect to the rotational component of ${}^{B_j}\mathbf{P}$ again relies on the angle-axis perturbation method used in Equation 41.

$$\begin{aligned} \mathbf{H}_{2_R} &= \frac{\partial \boldsymbol{\epsilon}_\kappa}{\partial {}^{G_j}\mathbf{R}_{B_j}} = \frac{\partial \boldsymbol{\epsilon}_\kappa}{\partial \boldsymbol{\varphi}} \\ &= \frac{\partial}{\partial \boldsymbol{\varphi}} \left[{}^W\mathbf{R}_{B_i} \mathbf{R}_{B_i} \left({}^{G_i}\mathbf{R}_{B_i} \text{Exp}(\boldsymbol{\varphi}) \right)^T ({}^{G_i}\mathbf{t}_{G_i\text{to}B_i} - {}^G\mathbf{t}_{G\text{to}B}) \right] \\ &= \frac{\partial}{\partial \boldsymbol{\varphi}} \left[{}^W\mathbf{R}_{B_i} \text{Exp}(-\boldsymbol{\varphi}) {}^{B_i}\mathbf{R}_{G_i} ({}^{G_i}\mathbf{t}_{G_i\text{to}B_i} - {}^G\mathbf{t}_{G\text{to}B}) \right] \\ &= \frac{\partial}{\partial \boldsymbol{\varphi}} \left[\text{Exp}(-{}^W\mathbf{R}_{B_i} \boldsymbol{\varphi}) \chi_i \right] \\ &\approx \frac{\partial}{\partial \boldsymbol{\varphi}} \left[\left(\mathbf{I}_3 - ({}^W\mathbf{R}_{B_i} \boldsymbol{\varphi})^\wedge \right) \chi_i \right] \\ &\approx \frac{\partial}{\partial \boldsymbol{\varphi}} \left[\chi_i + \chi_i^\wedge {}^W\mathbf{R}_{B_i} \boldsymbol{\varphi} \right] \\ &\approx \chi_i^\wedge {}^W\mathbf{R}_{B_i} \end{aligned} \quad (43)$$

The Jacobian of $\boldsymbol{\epsilon}_\kappa$ with respect to the translational component of ${}^{B_j}\mathbf{P}$ relies on the same method as Equation 42.

$$\begin{aligned} \mathbf{H}_{2_t} &= \frac{\partial}{\partial \boldsymbol{\delta t}} \left({}^W\mathbf{R}_{B_i} {}^{B_i}\mathbf{R}_{G_i} ({}^{G_i}\mathbf{t}_{G_i\text{to}B_i} + {}^{G_i}\mathbf{R}_{B_i} \boldsymbol{\delta t} - {}^G\mathbf{t}_{G\text{to}B}) \right) \\ &= {}^W\mathbf{R}_{B_i} \end{aligned} \quad (44)$$

By symmetry with Equations 41–44, and accounting for sign differences when necessary, the Jacobians with respect to the rotational and translational components of ${}^{B_j}\mathbf{P}$ and ${}^{G_j}\mathbf{P}$ can be found.

$$\mathbf{H}_{3_R} = -\chi_j^\wedge {}^W\mathbf{R}_{B_j}, \quad \mathbf{H}_{3_t} = {}^W\mathbf{R}_{B_j} \quad (45)$$

$$\mathbf{H}_{4_R} = \chi_j^\wedge {}^W\mathbf{R}_{B_j}, \quad \mathbf{H}_{4_t} = -{}^W\mathbf{R}_{B_j} \quad (46)$$

Finally, the Jacobian with respect to the translation to the target's center of mass ${}^G\mathbf{t}_{G\text{to}B}$ can be found.

$$\begin{aligned} \mathbf{H}_5 &= \frac{\partial}{\partial {}^G\mathbf{t}_{G\text{to}B}} (\chi_i + \chi_j) \\ &= {}^W\mathbf{R}_{B_j} {}^{B_j}\mathbf{R}_{G_j} - {}^W\mathbf{R}_{B_i} {}^{B_i}\mathbf{R}_{G_i} \end{aligned} \quad (47)$$

5.3 Center of Mass Prior Factor

A prior Gaussian belief $\mathbf{z}^t = {}^G\mathbf{t}_{G\text{to}B}$ is calculated based on information available at the first time step, recalling

Figure 7. The inspection scenario used to study the applicability of the center of mass estimation algorithm.

that $\underline{G}_0 \mathbf{R} = \mathbf{I}_3$.

$$\underline{G} \mathbf{t}_{\underline{G} \text{ to } B} = \underline{G}_0 \mathbf{t}_{\underline{G}_0 \text{ to } B_0} - {}^W_{B_0} \mathbf{R}^T {}^W \mathbf{t}_{W \text{ to } B_0} \quad (48)$$

Here, ${}^W_{B_0} \mathbf{R}$ and ${}^W \mathbf{t}_{W \text{ to } B_0}$ are obtained from the initial state prior belief $\mathbf{z}^{x_0}$, and $\underline{G}_0 \mathbf{t}_{\underline{G}_0 \text{ to } B_0}$ is obtained from Equation 27. A prior factor $\phi_{t_{\underline{G}}}$ with high covariance $\Sigma_{t_{\underline{G}}}$ is then placed on the position of the center of mass $\underline{G} \mathbf{t}_{\underline{G} \text{ to } B}$.

$$P(\mathbf{z}^t \mid \theta_k = \{\underline{G} \mathbf{t}_{\underline{G} \text{ to } B}\}) \sim \mathcal{N}(h(\theta_k), \Sigma_{t_{\underline{G}}})$$

$$\mathbf{z}^t \in \mathbb{R}^3, \quad \phi_{t_{\underline{G}}} = \|\mathbf{z}^t - \underline{G} \mathbf{t}_{\underline{G} \text{ to } B}\|_{\Sigma_{t_{\underline{G}}}}^2 \quad (49)$$

5.4 Combined Factor Graph

When the rotation kinematic factor from Section 5.2 is attached to all five relevant variables, it connects the trajectory factor graph from Figure 3a with the mapping pose graph from Figure 3b. Solving for the entire factor graph shown in Figure 3 is the preferred method for jointly estimating the trajectory, mapping the target, and determining the target center of mass.

5.5 Study of Applicability

This section studies the applicability of the center of mass estimation technique considered herein for

Figure 8. The sensitivity of center of mass estimation accuracy to position of the inspector.

different inspector positions, nutation angles, and sampling frequencies. In this study, a spherical target with radius r_t and normalized inertia ratios $\mathbf{J} = \text{diag}([1.2, 1, 1])$ nutates about its major axis x_B for 5 nutation periods at a nutation angle of β (see Figure 7). An inspector is placed in the x_W - y_W plane at a latitude λ and distance r_i ; the x_W axis is aligned with the target's angular momentum vector $\mathbf{h}_B$. Under this rotational motion, the angular velocity ω_B is constant; the frame rate of the inspector's camera is set to take an image every $t = \omega / \Delta\theta$ seconds. Pose noise with covariance $\Sigma_P = \text{diag}([\sigma_R^2, \sigma_R^2, \sigma_R^2, \sigma_t^2, \sigma_t^2, \sigma_t^2])$ where $\sigma_R = 0.5^\circ$ and $\sigma_t = 0.03 r_t$ is applied to each inspector pose in the geometric frame ${}^B_{\underline{G}} \mathbf{P}$; no noise is added inspector poses in the world frame ${}^B_W \mathbf{P}$. Solutions for the center of mass are obtained using the linear method presented in Section 5.1. For each data point, 10 trials are performed; error bars are used to represent the standard deviation of the results.

5.5.1 Inspector Position The sensitivity of center of mass estimation accuracy to the position of the inspector is shown in Figure 8. For this experiment $\beta = 25^\circ$, and $\Delta\theta = 20^\circ$. The polar orbit experiments correspond to single 360° circumnavigations of the target in the x_W - y_W plane. All other inspections are stationary at the specified latitude and distance. Inspections performed at close proximity yield more accurate results than distant inspections. For stationary inspections, lower latitudes have less error. However, a full polar orbit inspection yields results as accurate as those for a stationary equatorial vantage

Figure 9. The sensitivity of center of mass estimation accuracy to nutation angle.

Figure 10. The sensitivity of center of mass estimation accuracy to sampling frequency.

point. In general, it is recommended that inspections be performed: at the closest safe distance at which the target remains entirely in the field of view; at a variety of vantage points; and in a symmetric manner (i.e., analyze data from a full relative orbit rather than a fraction thereof).

5.5.2 Nutation Angle The sensitivity of center of mass estimation to the nutation angle β is shown in Figure 9. For this experiment $r_i = 4r_t$, $\Delta\theta = 20^\circ$, and a full polar orbit is performed. For small β , the rotation is nearly entirely about a single-axis, making center of mass determination inaccurate. For larger values of β , there is sufficient wobble in the motion of the target for the center of mass to be accurately determined.

5.5.3 Sampling Frequency The sensitivity of center of mass estimation accuracy to the sampling frequency is shown in Figure 10. For this experiment $r_i = 4r_t$, $\beta = 25^\circ$, and a full polar orbit is performed. Error is reduced by choosing a sampling frequency for which the target rotates significantly between image acquisitions. However, the choice of sampling frequency will also require practical considerations. To facilitate frame-to-frame feature matching, images must contain overlapping sections of the scene; additionally, there are limitations to the acceptable off-optical-axis scene rotation that can be accommodated by feature matching algorithms. Real-time processing considerations may limit the minimum $\Delta\theta$ for very quickly rotating target objects. A $\Delta\theta$ between 5 - 30° has been tested herein with favorable results.

6 Implementation

This section outlines the acquisition of real and simulated test data, determination of ground truth, and implementation of the algorithms described in Sections 3–5.

6.1 Data Acquisition

6.1.1 International Space Station Data from the International Space Station (ISS) was obtained using the Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) laboratory for distributed satellite systems, which consists of a set of hardware and software tools developed for the maturation of metrology, control, and autonomy algorithms (Hilstad et al. (2010); Saenz-Otero (2005)).

SPHERES are micro-satellites which operate in a flat floor environment at MIT, on parabolic flights in temporary microgravity, and aboard the ISS in long duration microgravity. They are able to maneuver in six degrees of freedom (6-DOF) in microgravity (3-DOF during ground operation), to communicate with each other and with a laptop control station, and to identify their position with respect to each other and to the experimental reference frame. State estimation on the SPHERES satellites is performed using an extended Kalman filter (EKF) with inputs from gyroscopes, external ultrasonic beacons, and commanded forces and torques. The position and orientation accuracy of this system, which is called “global metrology” is nominally 1 cm and 3° respectively, but can degrade near the edges of the test volume (the space encompassed by the ultrasonic

Figure 11. (Left) The major components of the SPHERES-VERTIGO test platform Tweddle (2013). (Right) SPHERES-VERTIGO in operation aboard the ISS.

beacons). The SPHERES satellites run C code using an onboard digital signal processor; there are currently three SPHERES satellites on the ISS.

Stereoscopic image data was obtained using the Visual Estimation for Relative Tracking and Inspection of Generic Objects (VERTIGO) Goggles, a hardware addition to the SPHERES satellites that enables vision-based navigation research in the 6-DOF, microgravity environment on the ISS. Attaching to the SPHERES expansion port, the Goggles include stereo cameras, an embedded x86 Linux computer, and a high-speed wireless communications system (Tweddle (2013); Tweddle et al. (2015)). The VERTIGO Goggles computer runs C++ code including OpenCV and communicates with the SPHERES satellites via a serial port. Two VERTIGO Goggles units are currently on the ISS. The major components of the SPHERES-VERTIGO test platform are shown in Figure 11.

The experimental data presented herein was collected during SPHERES Test Session #53 Test #7, Run #3 on January 24th, 2014. One SPHERES satellite with attached VERTIGO Goggles was given the role of inspector, and another was given the role of target. The target satellite was equipped with stickers (see Figure 11), adding texture to the satellite to aid in natural feature detection (Tweddle (2013)).

Control of the inspector's orientation is performed by commanding the inspector to point toward the target's visual centroid. To determine the visual centroid, a depth map is found using the Library for Efficient Large-Scale Stereo Matching (libelas) (Geiger et al. (2010)), and restricted to exclude the immediate foreground and the ISS background (a range of depths from 0.1–0.8 m was included). This thresholded depth

map is then averaged to obtain a measurement of range and bearing to the target's visual centroid. The command to point toward the target's centroid does not constrain the inspector's orientation about its camera's optical axis, so the angular rate in this direction, as measured by the onboard gyroscopes, is commanded to zero. Datasets were logged during all tests with the measurement frequencies shown in Table 1. Figure 12 shows an example of the stereo images collected on the ISS.

Table 1. Frequencies of logged data during ISS and simulated experiments.

	Test Session #53	Simulation
Stereo images	2 Hz	5 Hz
IMU data	20 Hz	50 Hz
Thruster firing times	1000 Hz	1000 Hz
Global metrology	5 Hz	5 Hz

6.1.2 Simulation Dynamics simulation is performed using the SPHERES simulation, which uses a combination of MATLAB, Simulink, and C code compiled as binary MATLAB executable (MEX) files (SSL (2016)). Visual simulation is performed using the Blender 3D modeling and animation software (Blender (2017)). Inspired by a scene from the film *Interstellar*, in which the crew must dock to a spinning spacecraft, the visual model chosen for the target in this simulation is the *Endurance* spacecraft from the film. A visual model of *Endurance* was obtained from a Kerbal Space Program modification (Jewer (2015)). The true states resulting from the SPHERES simulation are scaled by 100× to generate the inspector's camera positions in the Blender model. Blender's Python interface is used

Figure 12. An example stereo image of the target SPHERES satellite on the ISS.

Figure 13. An example stereo rendering of the *Endurance* spacecraft using Blender.

to command the poses of the target and the inspector's left and right cameras; a full grayscale 3D rendering with a space background is performed at each time step to create a sequence of stereo images similar to that obtained using SPHERES-VERTIGO (see Figure 13). Despite the scaling to generate camera vantage points in Blender, VERTIGO's camera calibration parameters are used in data analysis, so all results herein are at the scale of the SPHERES test platform.

6.2 Experimental Design

Two separate tests are performed herein to validate the trajectory estimation and mapping, and center of mass determination algorithm on ISS and simulated data.

6.2.1 ISS Test A target SPHERES satellite is commanded to hold position at the origin of the SPHERES test volume; its orientation is made to emulate a passive intermediate axis spin with small off-axis components, using the CAD-determined SPHERES inertia ratios. Once the thrusters properly

establish the spin, orientation control is removed from the target satellite and it is allowed to rotate passively. The inspector SPHERES satellite is commanded to stay at a fixed distance from the target's visual centroid, allowing the inspector to move freely in a sphere surrounding the target. During this test, disturbance forces inside the ISS, likely resulting from airflow, led to an inspector trajectory which circumnavigated the target in a nearly perfect circle.

6.2.2 Simulated Test A (simulated) target SPHERES satellite is commanded to hold position and orientation near the center of the SPHERES test volume. The (simulated) inspector SPHERES satellite is made to circumnavigate the target satellite in a free-orbit ellipse. The position of the inspector's center of mass is commanded to follow a simulated 1.1×0.55 m, 5 minute free orbit ellipse (scaled down from 110×50 m and 91.4 minute to fit within the dimensions and time constraints typical of SPHERES tests). The orientation of the target is commanded to emulate a

low energy, torque-free, major axis spin with small off-axis components, using the inertia ratios of a short ring with a large radius.

6.3 Ground Truth Determination

6.3.1 International Space Station Ground truth orientation estimates are obtained by solving a separate visual-inertial odometry factor graph (see Figure 14) using features in the static ISS background (1–5 m from the camera). The same visual odometry procedure described in Section 4.2 is used to provide pose constraints between two subsequent inspector states in a visual-inertial odometry coordinate frame V . The visual odometry estimates are then aligned with the global metrology estimates by finding the optimal rotation ${}^W_V\mathbf{R}^*$. Ground truth inspector pose is thus comprised of a visual-inertial orientation estimate and a global metrology translation estimate. Ground truth for target pose is provided exclusively by global metrology since the target does not have attached stereo cameras.

Ground truth for the location of the target’s center of mass ${}^G\mathbf{t}_{GtoB}$ is not determined since its calculation requires the use of global metrology estimates whose errors are expected to be on the same order, or larger, than errors in center of mass determination. Comparison of the generated 3D map with the SPHERES CAD model is used in lieu of comparison with a ground truth value.

6.3.2 Simulation Ground truth output from the dynamics simulation is recorded at 5 Hz. The ground truth states of both the inspector and the target are thus available for every time step.

Ground truth for the translation to the center of mass ${}^G\mathbf{t}_{GtoB}$, is calculated by averaging the position of the M_{in} inliers in the first frame and using the ground truth states to determine its exact location, as follows.

$$\begin{aligned} {}^G\mathbf{t}_{GtoB} &= {}^G\mathbf{t}_{GtoB_0} + {}^G\mathbf{t}_{B_0toW} + {}^G\mathbf{t}_{WtoB_0} \\ &= -\frac{1}{M_{in}} \sum_{l=1}^{M_{in}} {}^{B_0}\mathbf{t}_{B_0toP_l} \cdots \\ &\quad + {}^{B_0}_W\mathbf{R} (-{}^W\mathbf{t}_{WtoB_0} + {}^W\mathbf{t}_{WtoB_0}) \end{aligned} \quad (50)$$

Here the position of an inlier in the first body frame B_0 is given by ${}^{B_0}\mathbf{t}_{B_0toP_l}$.

6.4 Implementation

The trajectory estimation and mapping algorithms described in Sections 3–5 are implemented in C++ using the publicly available GTSAM 4.0 library (Dellaert and Beall (2017)). Each variable and factor

Table 2. GTSAM classes used in implementation of trajectory estimation and mapping.

Variables	
Name	GTSAM Class(es)
Inspector state in W , $\mathbf{x}$	Pose3, Vector3, imuBias::ConstantBias
Inspector pose in G , ${}^B_G\mathbf{P}$	Pose3
Target CG in W , ${}^W\mathbf{t}_{WtoB}$	Point3
Target CG in G , ${}^G\mathbf{t}_{GtoB}$	Point3
Factors	
State prior, ϕ_{x_0}	PriorFactor<Pose3>, PriorFactor<Vector3>, PriorFactor<imuBias::ConstantBias>
Pose prior, ϕ_{P_G}	PriorFactor<Pose3>
Tgt. centroid prior, ϕ_{t_G} , ϕ_{t_W}	PriorFactor<Point3>
Range and bearing factor, ρ	BearingRangeFactorWithTransform<Pose3,Point3>
Star tracker factor, σ	PoseRotationPrior<Pose3>
Inertial odometry factor, ψ	CombinedImuFactor
Vision factor, ν	BetweenFactor<Pose3>
Rotation kinematic factor, κ	RotationKinematicFactor

from the combined factor graph presented in Section 5.4 is built up using GTSAM classes as listed in Table 2. The built-in **BearingRangeFactor** is modified to accept a transform between the body and the sensor pose, resulting in the upgraded factor **BearingRangeFactorWithTransform**. The inertial odometry factor is implemented using the **CombinedImuFactor**, with gravity set to zero. The **RotationKinematicFactor** is novel and created from scratch (see Setterfield (2017b) for source code).

The prior factor ϕ_{x_0} is initialized using the global metrology estimate closest to the test start time. For ISS data, star tracker measurements are obtained from the ground truth orientation estimates calculated in Section 6.3.1. For simulated data, star tracker measurements are obtained by adding zero-mean Gaussian noise $\Sigma_s = \text{diag}([\sigma_s^2 \ \sigma_s^2 \ \sigma_s^2])$, with $\sigma_s = 200$ arcseconds to the simulation ground truth orientation in the world frame.

The principal image processing steps as performed on one image are shown in Figure 15. Incoming images from the ISS dataset (but not the simulated dataset) are equalized to increase their contrast and the quantity of detected features. Range and bearing measurements are obtained by averaging a thresholded depth map produced by libelas, as described in Section 6.1.1. The bias in range measurements caused by seeing only the front side of the target is corrected for as in (Setterfield 2017a, p142).

The thresholded depth map is then used to create a bounding box around the object to restrict the

Figure 14. The visual-inertial odometry factor graph used to determine ground truth for ISS datasets.

Figure 15. An example of image processing on the SPHERES-Vertigo platform. (Left) 28 frame-to-frame feature matches with the previous image are shown in green; previous frame and current frame features are shown as small and large green circles respectively, and are connected by a green line. 50 frame-to-frame loop closure correspondences with state 58 are depicted in the same manner in yellow. Unmatched features are shown in grey. (Right) A thresholded depth map is shown in grey, with the centroid represented by a green dot, whose position and velocity in the camera frame is shown in green text at the top. The blue bounding box is used to restrict the feature search space.

search space for features (see the blue rectangle in Figure 15). Four octaves of non-upright SURF features with a Hessian threshold of 400 are detected using OpenCV 2.4.6.1 (OpenCV (2013)). 3D stereo frames are formed by triangulating all of the detected features that satisfy the epipolar constraints discussed in Section 4.2.

Frame-to-frame matching and visual odometry between each current stereo frame and the previous stereo frame are attempted at each iteration using custom C++ implementations of absolute orientation and maximum likelihood estimation (Setterfield (2017a)). If a valid motion estimate is not obtained between the previous frame and current frame, the algorithm continues to seek a match between the previous frame and subsequent frames until one is found (this is referred to as reacquisition herein). A motion estimate is considered valid if: $M_{in} \geq 6$ inliers are found (or $M_{in} \geq 10$ for reacquisition), and the estimated motion is under 45° and 0.5 m in rotation and translation respectively. Nodes $\mathbf{x}_j$ and $\mathbf{P}_{\mathcal{G}_j}^{B_j}$ of the factor graph presented in Section 5.4 are added only

when a valid frame-to-frame motion estimate has been obtained.

Loop closure, as discussed in Section 4.2, is attempted between the current time step and between one to three previous time steps h . At time step j , the stereo frames from time steps 0 through $j - N_{lc}$ ($N_{lc} = 5$) are divided into three equal sets. Firstly, a time step h_1 from the oldest set is selected at random, and loop closure is attempted; if unsuccessful, loop closure is attempted with a randomly selected time step h_2 from the second set; if this is again unsuccessful, loop closure is attempted with a randomly selected time step h_3 from the newest set. This methodology encourages loop closures with early poses. Loop closures are considered valid if all the conditions for a valid reacquisition are met, and the current estimated inspector pose in the geometric frame $\mathbf{P}_{\mathcal{G}_j}^{B_j}$ is within 20° of the pose at the time step of loop closure $\mathbf{P}_{\mathcal{G}_h}^{B_h}$. This rotation proximity requirement for loop closure is used to avoid erroneous loop closures.

These nodes and all connected factors are added to an ISAM2 factor graph optimizer (Kaess et al. (2012)). Estimation is performed incrementally, meaning that an estimate of all variables is available at every time step. When not otherwise stated, the results presented herein are the iSAM2 results from the final time step.

7 Results

This section outlines the results of algorithm validation, and the computational performance and convergence of the algorithms. Herein, scalar errors in estimated position $\mathbf{t}$ and orientation $\mathbf{R}$ with respect to ground truth values $\bar{\mathbf{t}}$ and $\bar{\mathbf{R}}$ are given as follows:

$$\delta t = \|\mathbf{t} - \bar{\mathbf{t}}\| \quad (51)$$

$$\delta \theta = [\delta \theta_1 \ \delta \theta_2 \ \delta \theta_3]^T = \text{Log}(\bar{\mathbf{R}}^T \mathbf{R}), \ \delta \theta = \|\delta \theta\| \quad (52)$$

In the presented results, three-dimensional frames are depicted by triads, for which red, green, and blue lines represent the x , y , and z axes respectively.

7.1 ISS Test

Data obtained from the test described in Section 6.2.1 is used to validate trajectory estimation and mapping experimentally. A summary of the estimated pose of the inspector is shown at the left and top right ($\times 2$) of Figure 16. Errors in trajectory estimation are shown in the top ($\times 2$) plots of Figure 17. This estimation accuracy is acceptable for disambiguating the motion on the inspector from that of the target so that mapping and center of mass determination can be performed.

The rotation $\frac{G}{B}\mathbf{R}$ from the target's body frame to its geometric frame is estimated as described in Setterfield (2017a). The true and estimated orientation of the target's body frame are shown at the bottom right of Figure 16. Error in the estimated orientation are shown in the bottom plot of Figure 17, with an average value of 15.0° . Three major factors could contribute substantially to this discrepancy. Firstly, the ground truth frame in which the inspector's trajectory is estimated differs by 5.4° on average from the global metrology frame. This means that measurements are being compared from two slightly different coordinate frames. Secondly, the target is spinning at approximately $60^\circ/\text{s}$, and global metrology is measured at 5 Hz, or every 0.2 s. Herein, the visual estimates are compared with the temporally closest global metrology estimates, which could have occurred up to 0.1 s earlier or later. This results in a global metrology measurement that may differ by up to 6° in orientation from its true value at the time it was

visually observed. Finally, it is possible that there is a small error in estimation of $\frac{G}{B}\mathbf{R}$ in Setterfield (2017a).

Figure 17. Estimation errors in the ISS test.

A feature map of the SPHERES satellite obtained during this test is shown in Figure 18a. As expected, the triad for the geometric frame $\underline{G}$ lies at the center of the purple first inlier points. Inlying SURF features are depicted by colors pertaining to the order in which they are detected; from oldest to newest, they are colored blue, green, orange, and yellow. Outlying SURF features are depicted as small, grey points. The 3D map of the target SPHERES satellite is compared to a 3D CAD model in Figure 18b. To perform this comparison, the feature map and 3D model are rotated and translated into the estimated and true body frames, respectively. The 3D feature map closely resembles the CAD model; this is especially true for the inliers, shown in green. The closeness of fit between the point cloud and the 3D map demonstrates that the center of mass has been successfully estimated. The superior quality of the inlying visual features suggests that a feature map for use in subsequent relative pose estimation should include only inlying points. Each of the 6684 inlying features have an associated feature descriptor, providing ample features for use in relative pose estimation.

Figure 16. Results from the ISS test. (Left) The pose history of the inspector. The green dot represents the start of the trajectory, the triads represent the B frame, the blue and orange dots represent the end of the global metrology and estimated trajectory respectively, and the red dot represents the center of mass of the target. (Top Right $\times 2$) The pose history of the inspector $\{^W t_{WtoB}, {}^B_W \mathbf{q}\}$. (Bottom) The target orientation ${}^B_W \mathbf{q}$.

Figure 18. (a) Estimated map and body axes of the target SPHERES satellite. (b) Comparison of feature map with a 3D CAD model of the SPHERES satellite.

7.2 Simulated Test

Data obtained from the test described in Section 6.2.2 is used to validate trajectory estimation and mapping in simulation. A summary of the estimated pose of the inspector is shown at the left and top right ($\times 2$) of Figure 19. Errors in trajectory estimation are shown in the top ($\times 2$) plots of Figure 20.

The rotation ${}^G_B \mathbf{R}$ from the target's body frame to its geometric frame is estimated as described in Setterfield (2017a). The true and estimated orientation of the target's body frame are shown at the bottom right of Figure 19. Error in the estimated orientation are shown in the bottom plot of Figure 20, with an average error of 1.4° . Error in the final estimate of the target's center of mass is shown in Table 3.

Figure 19. Results from the simulated test. (Left) The pose history of the inspector. The green dot represents the start of the trajectory, the triads represent the B frame, the blue and orange dots represent the end of the global metrology and estimated trajectory respectively, and the red dot represents the center of mass of the target. (Top Right $\times 2$) The pose history of the inspector $\{\mathbf{}^W \mathbf{t}_{W \rightarrow B}, \mathbf{}^B_W \mathbf{q}\}$. (Bottom) The target orientation $\mathbf{}^B_W \mathbf{q}$.

Table 3. Errors in the final estimates of center of mass in the simulated test.

Direction in $\underline{G}$	True $\underline{G} \mathbf{t}_{G \rightarrow B}$	Estimated $\underline{G} \mathbf{t}_{G \rightarrow B}$	Error Mag.
x	0.0565 m	0.0581 m	0.0016 m
y	0.0078 m	0.0049 m	0.0029 m
z	-0.0213 m	-0.0256 m	0.0043 m
Total			0.0054 m

A map of the *Endurance* spacecraft is shown in Figure 21a. The geometric frame $\underline{G}$ is shown together with the estimated translation to the center of mass of the spacecraft, which is located in the middle of the central module. The feature map is compared with the original Blender model of the spacecraft in Figure 21b. Most inliers, shown in green, surround the surface of the spacecraft, as desired. Some features resulting from faulty correspondences of stars passed both the depth thresholding and the visual odometry inlier checks and do not lie on the surface of the model; this is especially true near the center of mass, where minimal feature motion is present. These features are unlikely to affect future relative pose estimates, since the correspondence of these separate stars is a result of a specific vantage point, and if seen again, they will likely be outnumbered by inlying features.

7.3 Computational Performance and Convergence

Trajectory estimation and mapping using GTSAM is solved on a Linux virtual machine running Ubuntu Linux 10.04. The virtual machine is hosted on an Intel Core i7-3630QM 2.4 GHz laptop and allocated 2 cores and 8GB of RAM.

Retired geostationary satellites have been found to spin at rates between $0.06^\circ/\text{s}$ – $2.6^\circ/\text{s}$ Binz et al. (2014), indicating rotational speeds that are between 9.8 – $1000\times$ slower than those considered herein ($60^\circ/\text{s}$ for the ISS test and $25.5^\circ/\text{s}$ for the simulated test). In the simulated test, 91.4 minute free-orbit ellipse in LEO is compressed by a factor of $18.3\times$ into a 5 minute time period. Therefore, it is reasonable to view the tests conducted in this paper as an analogue for LEO proximity operations sped up by a factor of 18.3 and scaled down in physical dimensions by a factor of 100. This analogy is used when assessing the real-time applicability of the algorithms herein.

7.3.1 ISS Test The total computation times of the algorithms in the ISS test are shown in Table 4. When scaled temporally by $18.3\times$ as discussed above, the 153 s ISS test inspecting a target spinning at approximately $60^\circ/\text{s}$ becomes a 46.7 minute test of a target spinning at $3.3^\circ/\text{s}$ and the 22.9 minutes of computation can then be performed in real time within the inspection period (if the onboard computer

Figure 21. Estimated map and body axes of the *Endurance* spacecraft.

Figure 20. Estimation errors in the simulated test.

has similar computational capabilities as that used herein). The iSAM2 algorithm developed in this paper accounts for just 0.12% of the total computation time.

Therefore, any inspector satellite capable of performing the requisite image processing for inspection of an on-orbit object in real-time will also be able to run the algorithms presented in this paper in real-time at minimal extra computational cost.

Table 4. Total algorithm computation times in the ISS test.

Number of images	310	
Number of nodes	287	
Number of loop closures	10	
Equalize	0.460 s	0.034%
Range and bearing measurement	169.9 s	12.4%
Stereo frame acquisition	625.4 s	45.6%
Visual odometry	143.8 s	10.5%
Loop closure	431.5 s	31.4%
iSAM2	1.66 s	0.12%
Total	1372.8 s	100%

The computation times for each image are shown in Figure 22. Equalization is computationally inexpensive, taking approximately 1.5 ms per image. A visual lock on the target is lost from images 12-22 and 222-232. During this time, no nodes were created and loop closure was not attempted. The bottom plot in Figure 22 shows the iSAM2 computation times in more detail. Spikes in computation time correspond to successful loop closures. Disregarding loop closures, iSAM2 computation time grows linearly at approximately 15.0 μ s per image over the duration of the test. An increased computational load associated with loop closure and a complexity between $\mathcal{O}(1)$ and $\mathcal{O}(N^3)$ are predicted for iSAM2 by Kaess et al. Kaess et al. (2012). The complexity of iSAM2 optimization

Figure 22. Algorithm computation times per image in the ISS test.

for the combined factor graph in this paper appears to be approximately $\mathcal{O}(N)$ for problems of the size tested.

7.3.2 Simulated Test The total computation times of the algorithms in the simulated test are shown in Table 5. When scaled temporally by $18.3\times$ as discussed above, the 19.1 minutes of computation time can be performed nearly in real time within the 18.7 minute inspection period. The iSAM2 estimation algorithm developed in this paper accounts for just 0.29% of the total computation time.

Table 5. Total algorithm computation times in the simulated test.

Number of images	311	
Number of nodes	310	
Number of loop closures	10	
Equalize	0 s	0%
Range and bearing measurement	226.1 s	19.7%
Stereo frame acquisition	400.8 s	34.9%
Visual odometry	131.1 s	11.4%
Loop closure	386.1 s	33.6%
iSAM2	3.30 s	0.29%
Total	1147.5 s	100%

The computation times for each image are shown in Figure 23. Stereo frame acquisition is in general much faster than in the ISS case. This is because extended SURF-128 feature descriptors are used in the ISS case, whereas the standard SURF feature descriptors are used in the simulated case; while providing improved feature matching, the extended descriptor is more computationally expensive to extract and match (Bay et al. (2006)). The iSAM2 computation times in the

Figure 23. Algorithm computation times per image in the simulated test.

Figure 24. Convergence of the center of mass over the duration of the simulated test.

bottom plot of Figure 23 exhibits the same behavior as in the ISS test; during this test the rate of iSAM2 computation time growth is approximately $20.2 \mu\text{s}$ per image.

Convergence of the translation to the center of mass is shown in Figure 24 together with its $1-\sigma$ uncertainty bounds. The center of mass estimate quickly converges to an accuracy of better than 2 cm, with a final error of 5.4 mm; the diameter of the scaled down *Endurance* spacecraft is approximately 30 cm.

8 Conclusions and Future Work

This paper addresses on-orbit inspection of a rotating object using a moving observer, seeking a sufficiently detailed and accurate estimate of the target's appearance and center of mass to aid in the planning and execution of subsequent proximity operations.

To disambiguate the motion of the inspector from that of the target, a novel trajectory estimation algorithm was used (Setterfield et al. (2017)). Efficient

mapping algorithms were created, together with a novel “rotation kinematic factor”. This factor links the dynamics of the inspector with that of the target and is used to determine the target’s center of mass.

These algorithms were validated experimentally on the ISS as well as in simulation. The computational cost of the developed algorithms was shown to be minimal compared to that of image processing required for on-orbit inspection.

Lighting conditions and specularity of the target will make application of the algorithms in this paper difficult when using a passive visual sensor such as stereo cameras (as opposed to an active visual sensor such as LiDAR). It is possible to investigate strategies for coping with these issues using 3D rendering software. In this software, surface properties of the satellite can be set as well as sun direction and intensity; these parameters could be adjusted to emulate on-orbit lighting conditions with high fidelity.

As shown in Section 4, it is always possible to create a feature map of the target, regardless of whether the target is subject to external torques. This is a result of the fact that mapping is done in a target-fixed frame. The self-contained pose graph shown in Figure 3b highlights the separability of mapping the target from inspector dynamics. Solution of this visual odometry pose graph in isolation would facilitate mapping for arbitrary target motion, even if the target is acted upon by unknown external forces.

When an unknown target object does not undergo rotation with respect to the inertial frame, the best estimate of its center of mass is obtained by assuming it has uniform density, and averaging the volume contained by the 3D map of the object. When the target undergoes single-axis rotation, its axis of rotation is observable, but the exact location of the center of mass is not; the best estimate of center of mass is then the projection of the 3D map’s volumetric center onto this axis. When the target undergoes multi-axis rotation, its center of mass is observable. Center of mass determination is applicable for passive and active targets, provided that they are not acted upon by unknown external forces.

This paper sought to advance upon previous techniques for on-orbit inspection of uncooperative objects. Algorithms for trajectory estimation, mapping, and center of mass estimation have been developed. Their validation, both experimentally and in simulation, has demonstrated that a substantial portion of the information required for proximity operations can be accurately and efficiently observed during a preliminary inspection phase. The tools presented herein are

a stepping-stone for future work that may lead to missions involving autonomous satellite repair, the capture and removal of space debris, and the exploration of near-Earth asteroids.

Acknowledgements

The first author would like to acknowledge colleagues Max Yates and Danilo Roascio at the MIT SSL, colleagues Aric Katterhagen and Simeon Kanis at NASA AMES, NASA astronauts Koichi Wakata and Richard Mastracchio, and Professor Emilio Frazzoli. This work was performed with financial support from MIT, and DARPA InSPIRE & InSPIRE II contracts #NNH11CC25C and #NNH13CJ23C.

References

- Aghili F, Kuryllo M, Okouneva G and English C (2011) Fault-Tolerant Position/Attitude Estimation of Free-Floating Space Objects Using a Laser Range Sensor. *IEEE Sensors Journal* 11(1): 176–185.
- Aghili F and Su CY (2016) Robust Relative Navigation by Integration of ICP and Adaptive Kalman Filter Using Laser Scanner and IMU. *IEEE/ASME Transactions on Mechatronics* 21(4): 2015–2026.
- Augenstein S (2011) *Monocular Pose and Shape Estimation of Moving Targets, for Autonomous Rendezvous and Docking*. PhD thesis, Stanford University.
- Bay H, Tuytelaars T and Gool LV (2006) Surf: Speeded up robust features. In: *Computer Vision-ECCV*.
- Besl P and McKay N (1992) A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2): 239–256.
- Binz CR, Davis MA, Kelm BE and Moore CI (2014) Optical Survey of the Tumble Rates of Retired GEO Satellites. In: *Advanced Maui Optical and Space Surveillance Technologies Conference*.
- Blender (2017) Home of the Blender project - Free and Open 3D Creation Software. URL <https://www.blender.org/>.
- Davison AJ, Reid ID, Molton ND and Stasse O (2007) MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(6): 1052–1067.
- Dellaert F and Beall C (2017) GTSAM 4.0. URL <https://bitbucket.org/gtborg/gtsam/>.
- Engel J, Schöps T and Cremers D (2014) LSD-SLAM: Large-scale direct monocular SLAM. In: *Computer Vision-ECCV*. pp. 834–849.
- Fischler MA and Bolles RC (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated

- Cartography. *Communications of the ACM* 24: 381–395.
- Forster C, Carlone L, Dellaert F and Scaramuzza D (2015) IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In: *Robotics: Science and Systems XI*.
- Geiger A, Roser M and Urtasun R (2010) Efficient large-scale stereo matching. In: *Asian Conference on Computer Vision (ACCV)*.
- Hillenbrand U and Lampariello R (2005) Motion and parameter estimation of a free-floating space object from range data for motion prediction. *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*.
- Hilstad MO, Enright JP, Richards AG and Mohan S (2010) The SPHERES Guest Scientist Program. URL <http://ssl.mit.edu/spheres/resources/gsp/spheres-gsp.pdf>.
- Horn BKP (1987) Closed-form solution of absolute orientation using unit quaternions. *Journal of Optical Society of America* 4(4): 629–642.
- Jewer B (2015) Kerbal Space Program Endurance Mod. URL <https://drive.google.com/drive/u/0/folders/0B1DhFQzCfzY4V2RrWDd1ZVNpTjQ>.
- Kaess M, Johannsson H, Roberts R, Ila V, Leonard JJ and Dellaert F (2012) iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. *The International Journal of Robotics Research* 31(2): 216–235.
- Kaess M, Ranganathan A and Dellaert F (2007) iSAM: Fast Incremental Smoothing and Mapping with Efficient Data Association. In: *International Conference on Robotics and Automation*. pp. 1670–1677.
- Lichter MD (2005) *Shape, Motion, and Inertial Parameter Estimation of Space Objects using Teams of Cooperative Vision Sensors*. PhD thesis, Massachusetts Institute of Technology.
- Maimone M, Cheng Y and Matthies LH (2007) Two Years of Visual Odometry on the Mars Exploration Rovers. *Journal of Field Robotics* 24(3): 169–186.
- Matthies LH (1989) *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University.
- OpenCV (2013) OpenCV. URL <http://opencv.org/>.
- Saenz-Otero A (2005) *Design Principles for the Development of Space Technology Maturation Laboratories Aboard the International Space Station*. PhD thesis, Massachusetts Institute of Technology.
- Setterfield TP (2017a) *On-Orbit Inspection of a Rotating Object Using a Moving Observer*. PhD thesis, Massachusetts Institute of Technology.
- Setterfield TP (2017b) Space Inspection GitHub Repository. URL <https://github.com/tsetterf/space-inspection>.
- Setterfield TP, Miller D, Leonard JJ and Saenz-Otero A (2017) Smoothing-Based Estimation of an Inspector Satellite Trajectory Relative to a Passive Object. In: *IEEE Aerospace Conference*. Big Sky, Montana.
- SSL M (2016) MIT SPHERES Simulation. URL <https://sourceforge.net/projects/mit-spheres-simulation/>.
- Tweddle BE (2013) *Computer Vision-Based Localization and Mapping of an Unknown, Uncooperative and Spinning Target for Spacecraft Proximity Operations*. PhD thesis, Massachusetts Institute of Technology.
- Tweddle BE, Setterfield TP, Saenz-Otero A and Miller DW (2015) An Open Research Facility for Vision-Based Navigation Onboard the International Space Station. *Journal of Field Robotics* 33(2): 157–186.
- Tweddle BE, Setterfield TP, Saenz-Otero A, Miller DW and Leonard JJ (2014) Experimental Evaluation of On-board, Visual Mapping of an Object Spinning in Micro-Gravity aboard the International Space Station. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 2333–2340.