(Preprint) AAS 25-294

ENHANCED FULL VISUAL SLAM FOR CLOSE PROXIMITY EXPLORATION OF ASTEROIDS COMBINING INCREMENTAL SMOOTHING WITH FILTERING TECHNIQUES

Pietro Califano[*], Felice Piccolo[†], Paolo Panicucci[‡], Francesco Topputo[§]

This study proposes a visual SLAM algorithm for autonomous asteroid exploration, leveraging state-of-the-art estimation techniques from Robotics. Two variants of a factor graph-based SLAM system are developed: *ISAM2dynamics*, which directly incorporates orbital dynamics constraint into the SLAM estimation process, and *ISAM2concurrent*, featuring a dynamic filter tightly-coupled with the smoother. Both methods provide accurate sparse metric maps of the target while estimating the entire history of attitude and orbital states. A keyframe insertion logic is also proposed to insert graph states in an adaptive manner and only as required by the algorithm. A medium fidelity image processing emulator is presented to avoid the burden of rendering images while retaining representativeness of the measurements' errors. This enabled a comparative Monte Carlo analysis to highlight advantages and drawbacks of the approaches. Results show that incorporating the dynamics directly into the factor graph allows a more accurate and consistent estimation with respect to *ISAM2concurrent* approach. Moreover, it enables the smoother to correct estimation of landmarks that are no longer visible, thus maintaining the entire map consistent throughout the navigation window. However, this comes at the cost of a computational cost of the smoothing that increases over time.

INTRODUCTION

The last two decades have been characterized by an increasing interest of the space community in small-bodies exploration missions. Near-Earth asteroids in particular are the most prominent targets, owing to their proximity to Earth. JAXA's spacecrafts Hayabusa 1 and 2 visited Itokawa and Ryugu, respectively.[1,2] Similarly, NASA achieved great success with the OSIRIS-REx mission studying asteroid Bennu and proving a fully autonomous GNC system during TAG operations.[3,4] Despite OSIRIS-REx's autonomous capabilities, the ground segment was still largely involved in the generation of shape models and landmarks for navigation, and in the testing of GNC and of the Flight Dynamics' trajectory planning. Significantly, all these activities needed to be completed in a time-critical manner.[5] More recently, NASA's DART probe successfully hit the secondary body in the Didymos binary system, exploiting an autonomous GNC system to target the asteroid Dimorphos.[6–8] As follow-up, the ESA Hera mission, launched in October 2024, will reach the same asteroid system to analyze it more closely and to unveil the effects the DART impact caused.[9] In addition, Hera will perform a relative navigation experiment during its Experimental Phase to demonstrate autonomous feature-based navigation and trajectory correction in close proximity.[10] Similarly, the Hera's cubesat Milani will perform close proximity operations and demonstrate autonomous navigation techniques using centroiding algorithms.[11] Yet all of them are still rooted in the ground-based navigation paradigm, that relies on operations being planned on ground and uploaded to the spacecraft.

Despite recent advancements, there is still significant potential to reduce operational burden, improve navigation accuracy through online processing, and enable autonomous operations near small bodies. Algorithms

[*]PhD Student, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan
[†]PhD Student, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan
[‡]Assistant Professor, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milano
[§]Full Professor, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan

solving the Simultaneous Localization and Mapping (SLAM) problem address these issues by jointly estimating the spacecraft state and environment, reducing reliance on prior characterization, which has proven challenging in past missions.[12] In this context, it has been shown that filtering is inherently limited by computational complexity, estimation inconsistency, and inability to utilize loop closures effectively.[13,14] All these limitations are due to the recursive marginalization operation performed at every estimation step of a filter. This induces fill-in of the off-diagonal terms of the covariance matrix, which inevitably causes the computational cost to grow as the cube of the state vector size.[15,16] Moreover, the accumulation of errors due to linearization occurs faster over time with respect to smoothers, leading to inconsistency of the estimator. In contrast, algorithms solving the "Full" SLAM problem, namely solving for the *entire* history of navigation states and map states, overcome these limitations. Relevantly, in the robotics field, frameworks such as factor-graphs-based GTSAM and g2o are capable of guaranteeing real-time estimation by exploiting the inherently sparse structure of the problem. This is made possible by leveraging well-known Sparse Linear Algebra and Incremental optimization.[17–20] At the same time, they could be implemented to exploit parallel processes, indicating that near-future on-board computers for spacecrafts could be able to support such algorithms.[21] In the following Section "Related works and contribution", previous works closely related to this study are reviewed in detail, with a focus on highlighting the differences and unique contributions of the present work. Section "Methodology" provides the basics of the SLAM estimation framework, details the models employed for the analysis and the workflow of the estimation algorithms. Section "Results" characterizes the scenario and the image processing model, and comments the results in-depth. Finally, Section "Conclusions" summarizes the work and outlines the future development steps.

Related works and contribution

Building on the state of the art of the robotics literature, Dor et al. initially proposed to use the ISAM2 algorithm of the GTSAM library for monocular SLAM in small-body navigation, relying on a Perspective-N-Point (PnP) algorithm for smoother initialization without a dynamical model.[22] An image processing based on feature detection and matching from image pairs was used to provide measurements to the SLAM algorithm. The exploitation of prior knowledge stemming from a dynamical model was later addressed in an early pre-print and the successive extended journal version by the same authors.[23,24] In the former, a relative dynamics model between the spacecraft and the asteroid was incorporated as constraint in the factor graph in the form of the *Relative Dynamics* factor, comprising the gravitational force of the asteroid, the 3rd body perturbation of the Sun and the Solar Radiation Pressure. This largely aided estimation accuracy thanks to the odometry-like constraint between time instants and enabled the estimation of the full orbital state. The rotational state of the asteroid was also included, improving upon the work by Setterfield et al.[25] The more recent extension of this work included the estimation of the asteroid gravitational parameters and centre of mass, through a new type of factor dubbed *Relative Kinematics* factor. Essentially, the estimation is performed relative to an arbitrary geometrical reference frame, with a 6D pose offset with respect to the Principal reference frame, in turn part of the estimated quantities. Therefore, this factor binds the relative pose of the spacecraft with respect to the asteroid with its attitude and centre of mass position, exploiting the strong inertial constraint coming from Star tracker measurements.

An alternative approach to include a motion model prior was shown by Nakath et al. for a laser scan matching system, where g2o batch optimization solver was used in combination with an error-State EKF in the covariance form.[26] The latter enabled the update of the velocity states by fusing the dynamical model prior knowledge with the measurements coming from the LIDAR scan matching algorithm. In this scheme, the filter is reset at each graph estimator state, which thus defines its new reference. Additionally, $SE(3)$ pose edges are derived from either the filter prediction or the measurement itself. These act as an odometry-like measurement, therefore constraining the spacecraft orbital states in time. Star tracker attitude measurements were also assumed to be available at a much higher frequency and retrieved as needed by the estimators. Finally, it is relevant to note that this implementation performed pose graph optimization instead of jointly estimating spacecraft and map states. In other words, measurements are incorporated in a loosely-coupled way, an approach that typically attains a lower estimation accuracy. Moreover, the mapping task is not performed by the estimator such that correlations between navigation and map states (i.e. landmarks) cannot be retrieved and exploited in other tasks (e.g., data association).

Inspired by the previously mentioned studies, this work presents a comparison between two variants of a SLAM algorithm. The first, named *ISAM2dynamics*, features our version of the GTSAM library modified to include dynamics information. This is achieved through a factor embedding the Inertial dynamical model expressed in the Quasi-Inertial frame $\mathcal{N}$ with origin in the centre of mass (CoM) of the target, differently from the work by Dor et al.[23] The factor propagates the initial estimated state over the time interval between the two linked orbital states. It also calculates the process noise covariance, which is used to appropriately weight the residuals. The second variant, named *ISAM2concurrent*, consists in a tightly-coupled integration of ISAM2 with a filter. A kinematic constraint for the position in the form of a *Relative Position* factor is derived from the filter prediction. The latter is meant to guide the solver to satisfy the dynamics without modeling it directly in the graph, reducing implementation-related complexity and computational burden. Specifically, while *ISAM2dynamics* strictly requires the addition of a graph state whenever an updated navigation solution is required or a measurement must be processed, *ISAM2concurrent* can exploit the filter to achieve higher estimation frequency without causing an excessive increase of the graph size. Moreover, it does not require any modification of the GTSAM library. A keyframe-based optimization approach inspired by ORB-SLAM is introduced, allowing the incremental smoothing algorithm to be run only at a limited number of keyframes. The insertion of these is adaptively triggered by the image processing or by the filter based on several conditions. Finally, a simple yet effective method is designed to emulate the performance of a feature tracking system based on optical flow. This allows simulating feature tracks that closely reproduce the average behaviour of the algorithm, avoiding both the need to run it during simulations and to generate synthetic images in closed-loop. Moreover, in contrast to the actual algorithm, the low number of tuning parameters enable a better understanding of how the estimation algorithm responds to different levels of noise.

Notation

The notation and conventions used throughout this work are here shown. Firstly, the expressions "front-end" and "back-end" will be used to indicate the image processing and the estimation algorithms, respectively. Moreover, the word "frame" is used as equivalent to "image". Therefore, it should not be confused with "reference frame", where vectors' components are expressed. These are defined in terms of an origin point and three unit vectors composing a right-handed orthogonal basis. Table 1 reports the reference frames definitions and symbols used in the work.

Table 1: Reference frames definitions.

Reference frame name	Symbol	Definition
Camera	$\mathcal{C}$	+Z Nadir pointing, +X aligned with velocity vector, origin in spacecraft CoM
Spacecraft body	$\mathcal{B}$	Coincident with $\mathcal{C}$
Target Fixed	$\mathcal{TF}$	Defined from Hayabusa SPICE* kernels, origin in asteroid CoM
Quasi Inertial	$\mathcal{N}$	Aligned with ECLIPJ2000, origin in asteroid CoM

Rotations are stated in terms of rotation matrices $R_{\mathcal{A},\mathcal{B}}$ rotating a vector from reference frame $\mathcal{B}$ to reference frame $\mathcal{A}$. Vectors are written in bold as $\mathbf{v}^{\mathcal{N}}$ where the superscript indicates the reference frame where its components are expressed. Subscripts can refer to a discrete timestamp or indicate entities the vector refers to. For instance, $\mathbf{r}_{Cam}^{\mathcal{TF}}$ is the position of the camera origin in the target fixed frame.

METHODOLOGY

This section first introduces the key concepts of factor graphs underpinning the GTSAM estimation framework. Then, the assumptions of the present work are critically described. The navigation architecture is

described next, comprising the surrogate model of the image processing to generate measurements and the estimation algorithms.

Probabilistic inference over Factor graphs

A factor graph is a bipartite graph composed of two types of nodes, factors f and variables θ with edges ϵ between them.[27] Mathematically, it is a general graphical model representing the factorization of a generic function $f(\boldsymbol{\Theta})$ as in Equation 1:

$$f(\boldsymbol{\Theta}) = \prod_{i=1}^{M} f_i(\boldsymbol{\theta}_i) \tag{1}$$

defined on the domain of variables $\boldsymbol{\Theta}$, in terms of M functions $f_i(\boldsymbol{\theta}_i)$ (the factors). An edge ϵ_{ij} can only exist between a factor node and a variable node. Moreover, each factor only depends on a subset of the domain variables $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}$. Therefore, a factor graph naturally encodes independence relations between variables, which makes it particularly suitable to represent estimation problems. Specifically, each factor can express a probabilistic relation between measurements $\mathbf{z}_i$ and their predictions through a model $\mathbf{h}_i(\boldsymbol{\theta}_i)$, in the form of a Gaussian probability density function. Hence, a Gaussian factor may be written as Equation 2:

$$f(\boldsymbol{\theta}_i) \propto \exp\left(\frac{1}{2}\|\mathbf{h}_i(\boldsymbol{\theta}_i) - \mathbf{z}_i\|^2_{\Sigma_i}\right). \tag{2}$$

It is worth noting that the cost function expressed in Equation 2 is the Mahalanobis distance with covariance Σ_i. Therefore, the error evaluated in each factor corresponds to norm of a vector of whitened residuals. In this way, the variables elimination algorithm over the factor graph, a method from graphical model inference, becomes equivalent to solving a non linear least squares problem (Equation 3) for the maximum a posterior (MAP) estimate of the variables $\boldsymbol{\Theta}_{\text{MAP}}$. In practice, the problem may be solved by means of any nonlinear optimization method (e.g., Gauss-Newton algorithm), while the factor graph provides insights into the optimal ordering of the variables in the state vector.[28]

$$\boldsymbol{\Theta}_{\text{MAP}} = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^{M} \|\mathbf{h}_i(\boldsymbol{\theta}_i) - \mathbf{z}_i\|^2_{\Sigma_i} \tag{3}$$

Assumptions

The assumptions of this work are listed and briefly analyzed in the following, to provide the right context for the interpretation of the results.

1. The camera frame $\mathcal{C}$ is considered aligned with the $\mathcal{B}$ frame, assuming that their relative pose is perfectly known. In practical terms, this corresponds to assuming an accurate prior extrinsic calibration, which is usually available for space systems. The SLAM solution thus provides the pose of the camera with respect to a navigation frame. The latter corresponds to the target fixed frame $\mathcal{TF}$, to which the pose and map states of the graph estimator are referenced.

2. The (Quasi) inertial frame $\mathcal{N}$ is defined as centred in the centre of mass (CoM) of the asteroid with axes aligned with the ECLIPJ2000. The accelerations acting on the target body are neglected, hence the name "Quasi Inertial", dropped in the following for brevity. The rotational state of the asteroid is assumed known with respect to the $\mathcal{N}$, such that the rotation matrix $R_{N,TF}$ that transforms a vector between the two reference frames is available at any time.

3. Similarly, Sun ephemerides and dynamical parameters of the system used in the algorithms are assumed as perfectly known (i.e., equal to the ground-truth values).

Ground-truth models

The ground-truth poses were generated by integrating an initial orbital state condition. This was chosen such that the resulting ballistic trajectory allowed mapping of the entire body surface in a time horizon of about 1 day (90000 seconds). The ground truth acceleration model comprises spherical gravitational attraction, non-spherical gravity (spherical harmonics) terms up to the 4^{th} degree of the gravitational potential U and the solar radiation pressure (SRP) modeled as the cannonball model.[29,30] In the following equations, the time dependence is dropped for the sake of clarity. The acceleration model is expressed as Equation 4:

$$\mathbf{a}^{\mathcal{N}} = -\mu_{\text{Main}} \frac{\mathbf{r}^{\mathcal{N}}}{r^3} + P_{\text{SRP}} \frac{\rho_{\text{SRP}} A_{\text{SRP}}}{m_{\text{SC}}} \frac{\mathbf{r}^{\mathcal{N}} - \mathbf{r}^{\mathcal{N}}_{Sun}}{||\mathbf{r}^{\mathcal{N}} - \mathbf{r}^{\mathcal{N}}_{Sun}||} + \mathbf{a}^{\mathcal{N}}_{\text{NSG}}. \tag{4}$$

The acceleration due to non-spherical gravity can be written as in Equation 5, noting that the vector must be also rotated to the $\mathcal{N}$ reference frame once computed:

$$\mathbf{a}^{\mathcal{TF}}_{\text{NSG}} = \frac{\partial U_{\text{NSG}}}{\partial \mathbf{r}^{\mathcal{TF}}}; \quad U_{\text{NSG}} = \frac{\mu_{\text{Main}}}{r} \sum_{n=1}^{4} \sum_{m=1}^{4} \left(\frac{R_{\text{Main}}}{r} \right)^n \bar{P}_{nm}(\sin\phi) \left(\bar{C}_{nm} \cos(m\lambda) + \bar{S}_{nm} \sin(m\lambda) \right) \tag{5}$$

where $\mathbf{r}$ is the shorthand for the spacecraft position, P_{nm} are the Legendre polynomial terms, C_{nm} and S_{nm} are the coefficients of the expansion and (ϕ, λ) are the longitude and latitude of the spacecraft in the $\mathcal{TF}$ frame. The parameters ρ_{SRP}, A_{SRP} and m_{SC} are the coefficient of reflection, the average cross-section alarea and the mass of the spacecraft, while P_{SRP} is the solar pressure per unit area. The reference poses were completed by defining the attitude $R_{\mathcal{C},\mathcal{TF}}$ of the camera with respect to the target fixed frame assuming a nadir pointing configuration. The $+Z$ (boresight) axis of the camera frame is therefore opposite to the position vector of the spacecraft. Moreover, its $+X$ axis is fixed so that it aligns with the direction of the velocity vector (i.e. local horizontal). The attitude of the target with respect to the Inertial reference frame was instead retrieved from SPICE kernels. Therefore, the attitude of the camera with respect to the Inertial reference frame was obtained by combining the two rotations above.

Front-end measurements emulation

Visual measurements from images In the visual SLAM problem, 2D image features (keypoints) $\mathbf{u}_i = [u, v]^T_i$, $i = 1, \ldots, N_{kps}$ are generally exploited as measurements to estimate the camera motion relative to a navigation reference frame. These correspond to the projection on the image plane of 3D points (landmarks), which are treated either as static entities (no motion) or as objects with known deterministic motion. Most SLAM algorithms are based on the former assumption, which greatly simplifies the optimization problem. Therefore, the target fixed frame $\mathcal{TF}$ is the natural choice of reference frame, in which landmarks are encoded as constant 3D position vectors. Given a calibrated camera with intrinsic parameters expressed by the camera calibration matrix K_{Cam} and the pose of the camera, the pinhole projection model (Equation 6) relates a landmark $\mathbf{l} = [x, y, z]^T$ to its projection $\mathbf{u}$ in image coordinates. $\mathbf{l}_{proj} = [x, y, z]^T_{proj}$ is the keypoint (i.e., the projected landmark) in homogeneous coordinates:

$$\mathbf{u} = \begin{bmatrix} x_{proj} \\ y_{proj} \end{bmatrix} / z_{proj} \; ; \qquad \mathbf{l}_{proj} = \begin{bmatrix} K_{Cam} \end{bmatrix} \begin{bmatrix} R_{\mathcal{C},\mathcal{TF}} & \mathbf{r}^{\mathcal{TF}}_{Cam} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}^{\mathcal{TF}} . \tag{6}$$

A surrogate model for the front-end was implemented to mimic the behaviour of an optical-flow based tracking algorithm, specifically a Kanade-Lucas-Tomasi tracker (KLT).[31] In previous works, the same image processing algorithm was also tested in a similar navigation scenario, presenting an implementation tailored for resource-constrained systems.[32] This choice was made to isolate the estimation algorithm performance and to avoid the need to render images in a closed-loop simulation. Nevertheless, the emulator was designed to reproduce the performance of the tracker in a stochastic sense, starting from the working principle of the actual algorithm, which is briefly explained in the following. A fixed number of keypoints N_{kps} is extracted in the first reference image using a feature extractor. Each feature is therefore associated to a track,

incrementally built-up by tracking the same keypoint across multiple subsequent frames. This is achieved by iteratively computing its displacement between each pair of frames, based on pixel intensity values. The same principle is applied in the emulator, in which displacements are computed geometrically from the ground-truth keypoints while the stochastic error is modelled as a random process (Equation 7). $\Delta\mathbf{u}_k$ is the true displacement in the image plane computed using the ground truth projections for the frames of the pair, as $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$, i.e. from the true keypoints at the current and previous frames. $\delta\mathbf{u}_k$ is the displacement error assumed to be distributed as Gaussian white noise with a variance $\sigma^2_{\delta u}$.

$$\Delta\tilde{\mathbf{u}}_k = \Delta\mathbf{u}_k + \delta\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\delta u}) \tag{7}$$

$$\tilde{\mathbf{u}}_k = \tilde{\mathbf{u}}_{k-1} + \Delta\tilde{\mathbf{u}}_k \tag{8}$$

The measured keypoint at the current frame is computed using Equations 8, where $\tilde{\mathbf{u}}_{k-1}$ is the measured keypoint the previous frame. In fact, the noisy displacement is added to the previously tracked keypoint, to introduce correlation and drift across frames. The tuning of the model parameter was done by matching the variance of the feature tracking errors produced by the emulator and the ones of a pyramidal KLT.[33] Finally, track losses are simulated using a Poisson distribution to determine how many tracks are marked as terminated at each tracking frame. The rate η_{loss} is the tuning parameter of the model. In addition, all tracks of features that are no longer visible or that are displaced outside the field of view are automatically removed. An example of the results obtained by this model is shown in Section Results.

The vertices of the asteroid shape model were chosen as the ground-truth landmarks to simplify the assessment of their estimation errors later in the analysis. In fact, it is only necessary to keep track of the matching of the ID of each landmark in the estimator with the unique true one. At each image, the simulation determines which vertices are visible, accounting for illumination conditions and occlusions using a ray tracing algorithm.[34] The tracking simulation is then performed to determine which tracks are still active and the associated measured keypoints in the current frame. Feature extraction is performed only at keyframes and simply consists in a random sampling of $N_{ext} = N_{max,kps} - N_{kps}$ features from the subset of visible ground-truth ones. The management of all the tracks is performed by dedicated data structures. Finally, it is mentioned that loop closure constraints were not included in the analysis but will be part of future developments.

Attitude measurements A classical approach in spacecraft navigation is to rely on a separate Attitude Determination and Control system (ADCS) running at much higher frequency with respect to the orbital state estimator. This in turn translates to the availability of attitude knowledge at any time stamp as required by the navigation system, modeled as Equation 9:

$$\tilde{R}_{\mathcal{C},\mathcal{N}} = R_{\mathcal{C},\mathcal{N}} \left[\boldsymbol{\theta}_{\mathcal{N},\tilde{\mathcal{N}}}\times\right] \tag{9}$$

accounting for the constraint of attitude parametrization, where $\boldsymbol{\theta}_{\mathcal{N},\tilde{\mathcal{N}}} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{st} I_3)$, and $R_{\mathcal{C},\mathcal{N}}$ is the true rotation matrix from the inertial reference frame to the camera reference frame. This assumes the error to be sufficiently small for the linearization to hold, which is reasonable for the typical precision of star trackers. Moreover, it is assumed that the measurements on all axes are uncorrelated.

Back-end architecture

Incremental SLAM using ISAM2 The estimation architecture builds upon GTSAM library and its MATLAB wrapper*, in particular employing the Incremental Smoothing And Mapping (ISAM2) solver.[20] This algorithm exploits the conditionally sparse structure of the problem to efficiently update the posterior distribution of all the estimated states. The Bayer tree is the underlying probabilistic representation of the linearized estimation problem, encoding the conditional relations among states.[35] ISAM2 exploits the latter to determine which states are updated each time a new factor is emplaced into the general non-linear factor graph. In fact, its efficiency is made possible by the selective updates and re-linearization, which in practical terms implies iteratively solving for a subset of the large non-linear least squares problem.[28] All the untouched (i.e., not affected by new measurements) branches of the Bayer tree rmains unmodified ant are simply re-attached to the overall Bayes Tree after re-elimination of the updated portion. Finally, recall that an initial

*`https://github.com/borglab/gtsam`. Last visited: 12/15/2024

guess is required for all the states due to the nonlinearity of the problem. The initialization step was found of particular relevance to achieve a fast and stable convergence, whereas poor initialization can easily hinder the robustness of the algorithm and possibly lead to a non-converged state. Since the GTSAM library is tailored for robotics, several modifications were required to use it for spacecraft navigation problems, mainly consisting in implementing a propagation scheme for the orbital state dynamics for the dynamics factor in *ISAM2dynamics*. Moreover, as pointed out in Dor et al., the noise models GTSAM employs to characterize the uncertainty of each factor cannot be modified while the algorithm is running due to limitations of the original implementation.[23] The ability of modifying the noise model is mostly required to adapt its covariance, whose propagation changes due to the State Transition Matrix (STM) being dependent upon the linearization point.

In this research's implementation, the ISAM2 update occurs at frames where state variables are added to the graph and subsequently smoothed over time. These are the frames indicated as "keyframe". This step incorporates all the new measurements, namely: keypoints from feature tracking, attitude from the ADCS, and a relative position constraint from either the *ISAM2concurrent*'s filter or the *ISAM2dynamics*' dynamics factor. Specifically, the filter runs in all frames that are not marked as keyframes, where no feature extraction occurs. The initial guess for the trajectory states of the smoother is always obtained from propagation and from attitude measurements. The keyframes insertion does not occur at a fixed time interval. Rather, it is adaptively triggered based on several conditions.

Keyframes insertion logic Keyframe insertion can be triggered by both the front-end and the back-end according to the following conditions, which constitute additional tuning parameters of the algorithm.

- A threshold for the minimum number of active tracks $N_{min,kps}$;
- A threshold for the minimum number $N_{min,match}$ of landmarks being matched in the filter observation update step;
- Maximum elapsed time interval/number of frames from the last insertion $N_{\Delta kf_{max}}$.

The conditions are verified when the quantity of interest exceeds its threshold, either decreasing if the latter is a lower bound, or increase if it is an upper bound. The tuning of these settings is one of the key elements determining the performance of the algorithm. The front-end is most often the driving factor, due to the requirement of maintaining a sufficient amount of active tracks at each frame. As a consequence, the insertion frequency can vary widely depending on the scenario and the illumination conditions. In general, since the computational cost of the graph estimator grows with the number of states and density of the factors, the insertion logic should seek the proper trade-off between accuracy and number of keyframes. At the same time, note that having keyframes sufficiently displaced in terms of parallax is beneficial for landmarks' distance estimation. This is especially relevant for the landmarks initialization step, when only two observations are available.

Landmarks initialization Similarly to pose and velocity states, 3D landmarks positions require a method to compute their initial guess for the nonlinear estimator. In this work's implementation, this is managed by exploiting the MATLAB triangulation functionality based on two-view geometry. The frames for triangulation always correspond to two adjacent keyframes kf_{k-1} and kf_k, namely the previously inserted keyframe and the one being inserted at the current timestamp. The camera matrices $P_{\mathcal{C},\mathcal{TF}} = \begin{bmatrix} K_{Cam} \end{bmatrix} \begin{bmatrix} R_{\mathcal{C},\mathcal{TF}} & \mathbf{r}_{Cam}^{\mathcal{TF}} \end{bmatrix}$ associated to the keyframes are computed from the best estimate of the pose at $k-1$ retrieved from the smoother, and the current pose at k using the motion model prediction for the position and the attitude measurement. It is worth noting that an accurate initialization of the landmarks in the graph is fundamental both for the accuracy of the estimation and for the robustness of the system due to the sensitivity of the SLAM algorithm to outliers. Specifically, adding largely erroneous points can easily cause the map estimate to be biased in a non recoverable way. For this reason, a simple but effective rejection scheme was designed. Firstly, all landmarks not passing the cheirality check (i.e., ambiguity exists in determining whether the points are in front of or behind the camera) are automatically discarded. Secondly, the empirical statistics of the reprojection error is used to determine a rejection threshold. The 75° quantile was chosen as a good compromise

between the number of accepted points and the improvement of the robustness of the initialization. Furthermore, since this approach always rejects a certain number of points regardless of how small their reprojection error is, an acceptance threshold was also added. This overrides the rejection scheme in case of particularly small errors to let more landmarks to be added to the map.

Navigation filter design As anticipated, the graph-based smoother only estimates states and landmarks at keyframes which may be separated by a large number of frames. A filter was thus implemented to obtain a higher frequency of estimation without increasing the number of graph states. This also figures as a seamless way to provide initial values for the non-linear optimization, if needed. At the same time, it enables the fusion of the keypoints at intermediate images that are anyway required by the tracker and would be wasted otherwise. Moreover, in the ISAM2concurrent variant, the filter dynamics is also exploited to derive a kinematic odometry-like constraint for the positions at adjacent keyframes, adaptively emplaced in case the number of active tracks decreases below an user-specified counter. This effectively aids the graph estimator to avoid possibly underdetermined states (i.e., optimization failure) while it provides an initial pose guess satifying the dynamics. Moreover, it also indirectly forces the optimizer to yield position estimates whose displacements are more compatible with the motion model.

A simple filter design was chosen for this first prototype, which only includes the orbital state in Cartesian coordinates $\left[\mathbf{r}^{\mathcal{N}}, \mathbf{v}^{\mathcal{N}}\right]^T$, estimated with respect to the $\mathcal{N}$ frame. Attitude states were not estimated for a two-fold reason. First, little margin exists to improve upon standard ADCS performance, especially under the statistical assumptions of the design. Second, the additional complexity required to handle attitude states was deemed unnecessary from a space system perspective. A square root Unscented Consider Kalman filter inspired by Geeralt et al. was tested.[36] This choice was made in view of future extensions, which will include a more non-linear dynamics, consider parameters and biases estimation. In this regard the prediction step benefits from the increased covariance realism and robustness that the Unscented transforms enables, especially considering the highly non-linear dynamics of small bodies' environment in close proximity trajectories. Moreover, it enabled quick prototyping as the UKF does not require the computation of Jacobians. The filter uses a simplified dynamical model with respect to the reference acceleration model, written as Equation 10:

$$\begin{cases} \dot{\mathbf{r}}^{\mathcal{N}}(t) = \mathbf{v}^{\mathcal{N}}(t) \\ \dot{\mathbf{v}}^{\mathcal{N}}(t) = \mathbf{a}^{\mathcal{N}}_{\text{Main}}(t) + \mathbf{a}^{\mathcal{N}}_{\text{SRP}}(t) + \boldsymbol{\nu}(t) \end{cases} \tag{10}$$

specifically removing the spherical harmonic expansion terms and adding an input noise $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \sigma_\nu^2 I_3)$. In fact, the state noise compensation approach was used to inject process noise into the dynamics through the velocity states, mapped onto position states under the assumption of piece-wise linear input noise, as common practice in filtering applications.[37,38] The observation update step relies on the same projection model shown in Equation 6, with measurement errors modeled as white Gaussian noise. It must be noted that this requires some prior knowledge of the 3D landmarks positions, which is assumed to come from the SLAM estimator. Specifically, at each keyframe, the filter receives a submap of the estimated landmarks positions from ISAM2 together with the associated unique identifiers of the corresponding tracks. At every new frame, a quick ID matching procedure is used to match the new keypoints tracked by the frontend with the available submap. There are several statistical assumption to be noted in this approach.

1. The landmarks 3D positions retrieved from the graph estimator are assumed to be uncorrelated and all affected by zero mean Gaussian noise error;
2. The correlation between the map states and the state estimates are neglected, to avoid retrieving the covariance of the former from ISAM2;
3. Position states of the filter are reset from ISAM2 pose estimate, but velocity ones are not. The correlation terms of the covariance cannot be retained and are set to zero.

Indeed, these are not formally valid and were found to have an effect on the accuracy, discussed more in depth in Section Results. Possible solutions exist, such as consider states, but their addition has not yet been explored in the current design.[39]

ISAM2 variants and factors formulation The key difference between *ISAM2dynamics* and *ISAM2concurrent* lies in the inertial dynamics constraint. This enables the estimation of velocities directly into the graph estimator and the need of one single pose prior to fix the metric scale thanks to the dynamics factor. In fact, the *ISAM2dynamics* does not rely on the filter neither for initialization nor for velocity estimation. On the contrary, *ISAM2concurrent* requires both a second pose prior and a propagation model for position initialization. From an optimization perspective, this is required to constraintù the metric scale of the problem due to the scale ambiguity inherent in monocular systems. It is worth noting that the accuracy of the reconstruction is underpinned by this prior knowledge.[22,40] Instead, in case of a motion model exists, the accuracy is related to its predictive accuracy, which in this case is mostly determined by the gravitational parameter knowledge. Finally, the dynamics version does not need to handle exchange of variables between smoother and filter resulting in a simpler graph construction and optimization. Specifically, keyframes are inserted based on the same conditions used by *ISAM2concurrent*, with the exception of the filter-related ones.

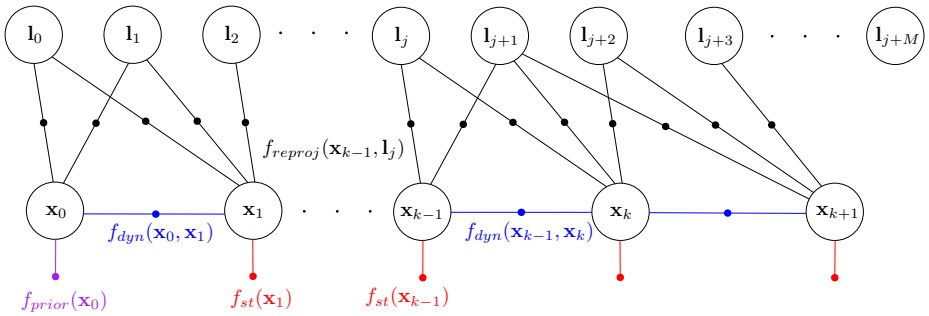

Figure 1: Slice of the factor graph representing the SLAM estimation problem for *ISAM2dynamics*.

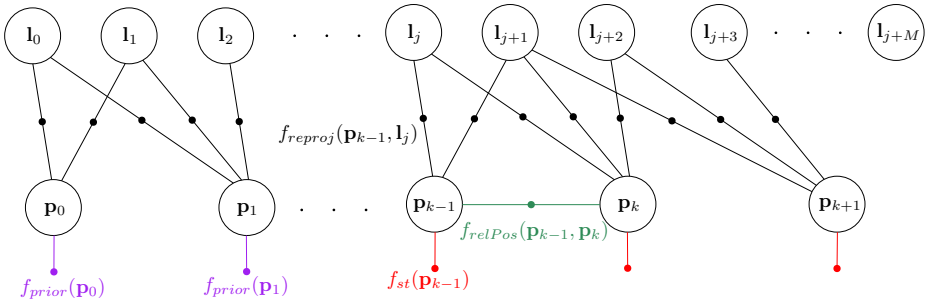

Figure 2: Slice of the factor graph representing the SLAM estimation problem for ISAM2concurrent.

Figures 1 and 2 depict slices of the estimation problem as a factor graph respectively for *ISAM2dynamics* and *ISAM2concurrent*. Colours are used to distinguish the different factors: red indicates the star tracker measurement; black the landmark projections; blue the dynamics factor for *ISAM2dynamics*; green the relative position constraint for *ISAM2concurrent*; finally purple corresponds to a pose prior factor. The left side indicates how the initialization of the algorithm occurs. In particular, the absence of the prior pose on the second state in *ISAM2dynamics* is highlighted, since metric scale is fixed by the dynamics. Finally, in Figure 2, $\mathbf{p}$ is used to stress that ISAM2 only estimates pose states, without velocity. In contrast, $\mathbf{x}$ includes velocity as well.

With the exception of the inertial dynamics factor, all others factors only required minor modifications to the ones included in the original GTSAM. These mainly consisted in modifying the *PriorFactor* and *BetweenFactor* GTSAM classes* to constrain only a subset of the pose type.

$$\boldsymbol{\epsilon}^{\mathcal{N}}_{dyn} = \begin{bmatrix} R_k \mathbf{r}^{\mathcal{TF}}_k \\ R_k \mathbf{v}^{\mathcal{TF}}_k \end{bmatrix} - \boldsymbol{\phi}\left(R_{k-1}\mathbf{r}^{\mathcal{TF}}_{k-1}, R_{k-1}\mathbf{v}^{\mathcal{TF}}_{k-1}\right) \tag{11}$$

*https://gtsam.org/doxygen/4.0.0/a03947.html. Last visited: 12/18/2024

Equation 11 reports the error function definition implemented into the inertial dynamics factor. R is the shorthand for the attitude matrix rotating from $\mathcal{TF}$ to $\mathcal{N}$ frames, while k and $k-1$ indicate the subsequent time instants linked by the factor. ϕ indicates the dynamics flow, computed by numerically integrating the dynamical equations (as in Equation 10) with a fixed-step RK4 scheme. Note that the residuals are computed from the position and velocity in the inertial reference frame $\mathcal{N}$, which require the assumed asteroid attitude knowledge. The optimization process also requires the Jacobians of the error function, which involves the state transition matrix Φ of the problem. This can be easily obtained by integrating the variational equations. A formulation using continuous time integration of the covariance P was adopted to construct the noise model.[41] All the equations to be integrated are grouped as Equation 12, in which the time dependence and reference frames are dropped for the sake of clarity. A is the Jacobian of the dynamics $A = A_{\text{Main}} + A_{\text{SRP}}$.

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{a}_{\text{Main}} + \mathbf{a}_{\text{SRP}} \\ \dot{\Phi} = A\Phi \\ \dot{P} = AP^T + PA^T + BQB^T \end{cases} \tag{12}$$

The continuous time process noise covariance Q is thus mapped onto the states through the input matrix B. The former was assumed as a 3x3 diagonal matrix expressing the variance of the Gaussian white noise affecting the velocity states as shown in Equation 13.

$$Q = \begin{bmatrix} q_{vx} & 0 & 0 \\ 0 & q_{vy} & 0 \\ 0 & 0 & q_{vz} \end{bmatrix} \qquad B = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{1}_3 \end{bmatrix} \tag{13}$$

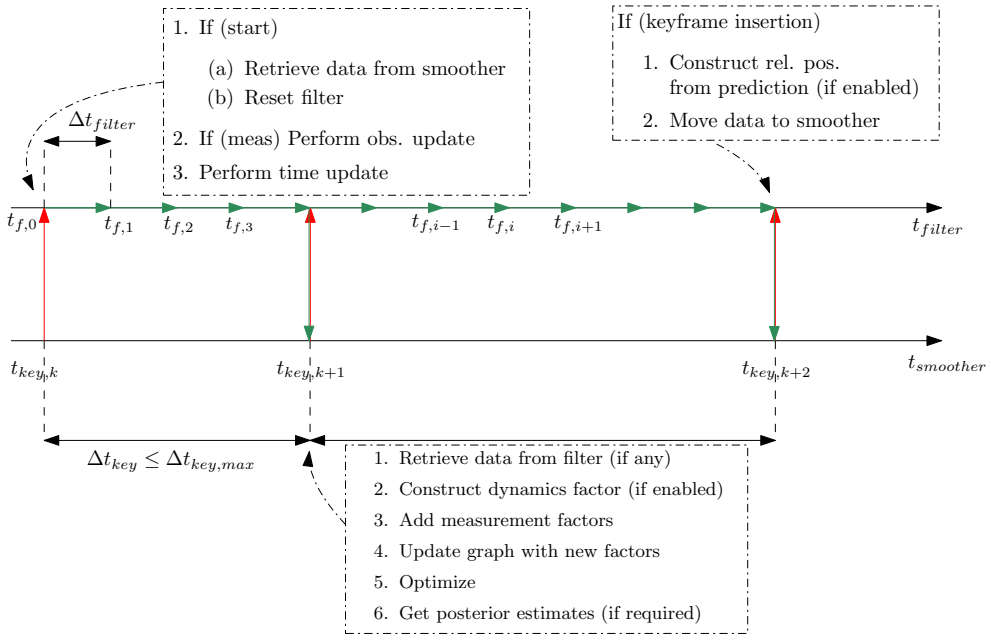

Figure 3: Scheme illustrating the smoother-filter integrated workflow.

Smoother and filter integration approach Figure 3 depicts the workflow of the estimation module. Note that the graph only comprises the discrete set of keyframe nodes $\mathbf{x}_k$ at timestamp $t_{key,k}$. These are the discrete-time poses (i.e., $SE(3)$ mathematical entities) estimated by ISAM2, comprising the attitude and the position of $\mathcal{C}$ reference frame with respect to $\mathcal{TF}$ reference frame. Velocity states are included as well in the $\mathcal{TF}$ reference frame for *ISAM2dynamics*. The filter runs start at the latest keyframe and execute until a new insertion is triggered.

At each new keyframe, the smoother retrieves all the measurements to be added to the graph, together with the initial guess from the ADCS and the filter. An incremental optimization step is then executed and the filter reset takes place, using the state marginal covariance retrieved from the smoother. The latter is also inflated by a coefficient of 1.5 to ensure the filter starts with a conservative uncertainty rather than being inconsistent. Finally, the expected value of a subset of the currently active landmarks, with a maximum of $N_{max,map}$ landmarks, is extracted from the smoother map and made available to the filter for the measurement updates.

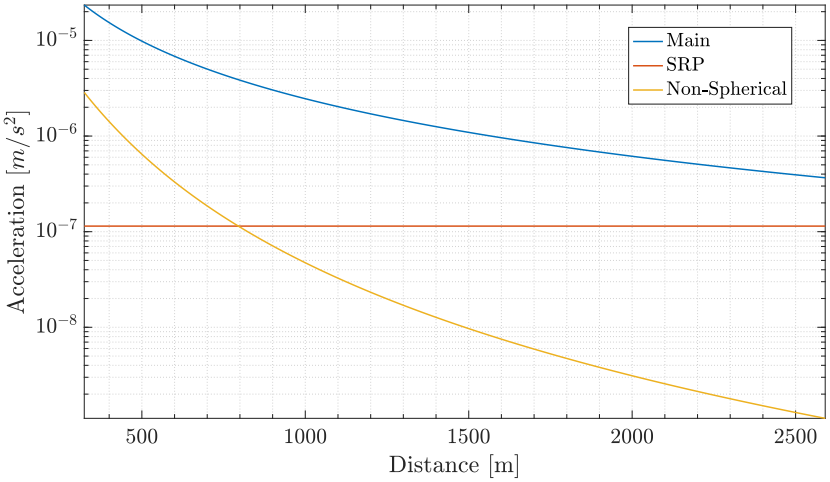

Figure 4: Acceleration norms in m/s^2 for the interval of ranges of the trajectories.

RESULTS

The simulation setup used for the analysis is now described, specifying all the parameters of the ground-truth generation process and the tuning parameters of the algorithms. Finally, the results are critically assessed.

Simulation setup

Scenario and ground-truth The investigation considered the asteroid Itokawa as test case scenario, whose 3D shape model was retrieved from the NASA Planetary Data System (PDS) small bodies node*. A set of 100 trials were simulated to perform a Monte Carlo (MC) analysis and evaluate the algorithms with diverse ballistic trajectories, initial attitude of the small body, and initial illumination conditions. The dynamical parameters of the ground-truth dynamics used for simulation are reported in Table 2. The spherical harmonics coefficients were taken from Scheeres et al.[29] The parameters of the spacecraft mimic a small 6U cubesat.

Table 2: Dynamical and physical properties of the Itokawa asteroid and spacecraft used for simulations.

Parameters	Description	Value
μ_{main} [m^3/s^2]	Asteroid Grav. parameter	2.36
R_{main} [m]	Asteroid reference radius	161.91
m_{SC} [kg]	Spacecraft mass	12.0
A_{SC} [m^2]	Spacecraft area for SRP	0.51
ρ_{SC} [-]	Coeff. of reflection for SRP	1.25

*PDS4 Bundle DOI: 10.26033/3b2j-yy57, retrieved from `https://arcnav.psi.edu/urn:nasa:pds:gaskell.ast-itokawa.shape-model`. Last visited: 01/02/2025.

Table 3: Initial conditions in Inertial frame and the 1σ uncertainty used for the Monte Carlo simulation.

Parameter	**Nominal value** [m, m/s, date]	**Std. deviation** 1σ [m, m/s, hours]
$\mathbf{r}^{\mathcal{N}}$	$\begin{bmatrix}1200.0 & -600.0 & -600.0\end{bmatrix}^T$	45
$\mathbf{v}^{\mathcal{N}}$	$\begin{bmatrix}0.0 & 0.0491 & 0.0\end{bmatrix}^T$	0.005
ET0	Apr 25 2006 06:00:00	4

Figure 4 reports the acceleration norms for the distances of interest for this work. This provides insights to better appreciate the discrepancy between reference dynamical models and the models employed for the estimation. Indeed, it is relevant to note that the spherical harmonic term has a non-negligible effect at these distances, almost comparable to the SRP, indicating that neglecting it could have an effect on the attainable estimation accuracy.

Figure 5 shows the ground-truth trajectory samples obtained by propagating the sampled initial conditions reported in Table 3. The 1σ values used for the sampling in the MC simulations are also shown in the last column. The initial ephemeris time of the simulation was varied as well in an interval equal to one period of rotation of Itokawa, to investigate the effect of the initial orientation on the estimation accuracy. Indeed, this dispersion influences only the attitude of the target body, thus the spherical harmonics accelerations, without considerably changing the position of the Sun.

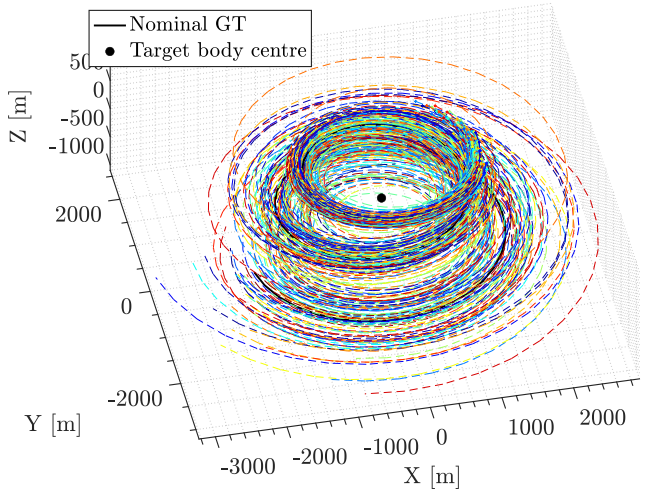

Figure 5: Ground-truth trajectories in $\mathcal{TF}$. Each colour uniquely identifies a single trial.

Table 4: Summary of the parameters selected for measurements simulation.

Measurements parameters	**Description**	**Value**
$\sigma_{\delta u}$ [px]	Pixel displacement std. dev.	0.2
η_{loss} [-]	Mean rate of features loss	15
σ_{st} [arcsec]	Star tracker measurement std. dev	45.0
K_{Cam}	Camera calibration matrix	$\begin{bmatrix}7286.14 & 0 & 1024\\ 0 & 7286.14 & 1024\\ 0 & 0 & 1\end{bmatrix}$

Tuning parameters The tuning parameters of the front-end emulation and back-end algorithms were set according to preliminary single-run experiments and kept fixed for all trials. Table 4 groups all relevant parameters for the frontend. Similarly, Table 5 summarizes all the tuning parameters of the keyframe insertion

Table 5: Tuning parameters selected for the back-end module.

Tuning parameter	Description	Value
$N_{max,kps}$	Max n° of active keypoints	300
$N_{max,map}$	Max n° of landmarks in submaps	300
$N_{min,match}$	Min n° of matches in filter update	1
$N_{\Delta kf_{max}}$	Max n° of frames between keyframes	20
q_{vi} [m^2/s^2]	Input process noise variance	10^{-8}
σ_{st} [arcsec]	Star tracker factor std. dev.	80
σ_{proj} [px]	Projection factor std. dev.	2.2
$\sigma_{proj,filter}$ [px]	Projection meas. std. dev. (filter)	8
σ_{prior} [arcsec, m]	Pose prior factor std. dev.	$[80, 5]$
Δt_{frames} [s]	Time interval between frames	45
$\sigma_{rel.pos}$ [m]	Rel. pos. factor std. dev.	50

logic, the initialization module, and the estimation algorithms (both filter and smoother). The ISAM2 parameters of the GTSAM library are omitted. However, it is mentioned that the relinearization is allowed at each time step since a large time interval is available between keyframes.

Frontend emulator model

The errors in the landmarks' projections as resulting from the emulation model were first assessed, to perform its tuning and to verify the correct emulation of the behaviour of the algorithm. Figure 6 shows the

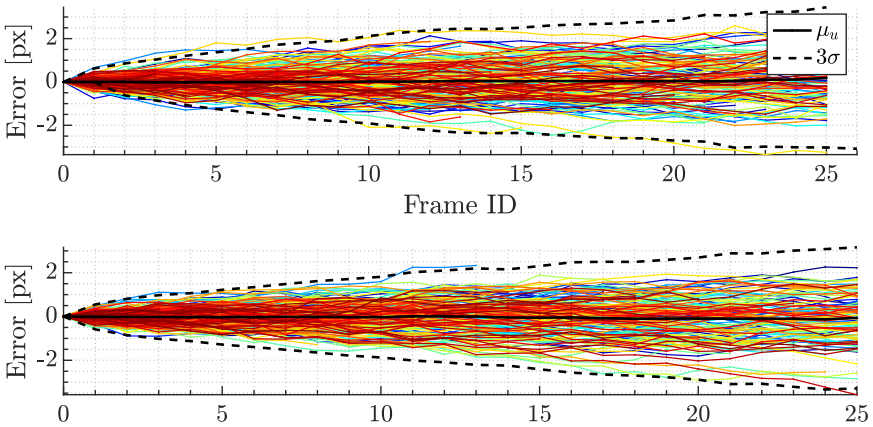

Figure 6: Frontend emulator tracking errors in pixel coordinates across 25 frames.

tracking errors of simulated keypoints in a number of frames representative of the average track length (i.e., tenth of frames). It is pointed out that the track length is largely scenario- and tuning-dependent. Therefore, the emulator should be tuned accordingly. Despite the simplicity of the emulation model, the typical drift of the tracking error is effectively reproduced, as can be seen from the similarity of the error evolution with respect to Figure 5 in Morrell et al.[42] The similarity of the error norm in Figure 7 over tracking time is also noticeable. The extremely outlying tracks are not reproduced because the outlier rejection of the image processing pipeline, which would likely filter them out, was not included in this behavioural model.

Monte Carlo analysis results

The two variants of the algorithm, *ISAM2concurrent* and *ISAM2dynamics*, were firstly compared in terms of estimation accuracy of the position. This was analyzed in the camera frame as estimated by the smoother at

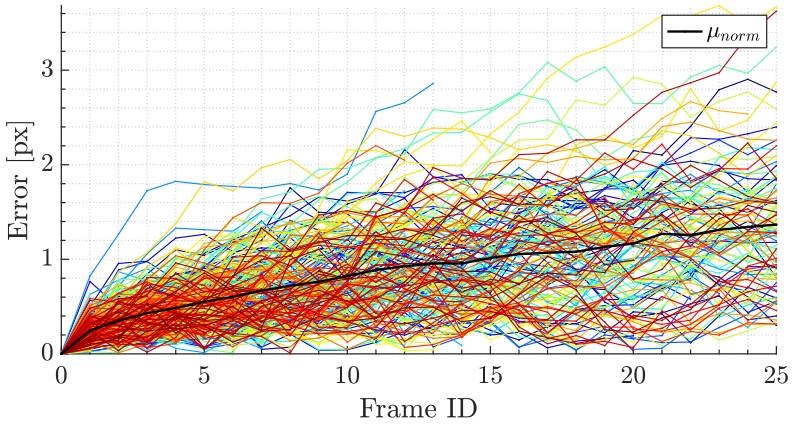

Figure 7: Frontend emulator tracking error norm across 25 frames.

every keyframe. Similarly, the accuracy of the velocity estimation was evaluated by comparing the filter estimates for *ISAM2concurrent* and those of the graph estimator for *ISAM2dynamics*. The Mahalanobis distance was used to assess estimation consistency. Asteroid surface reconstruction errors were finally evaluated.

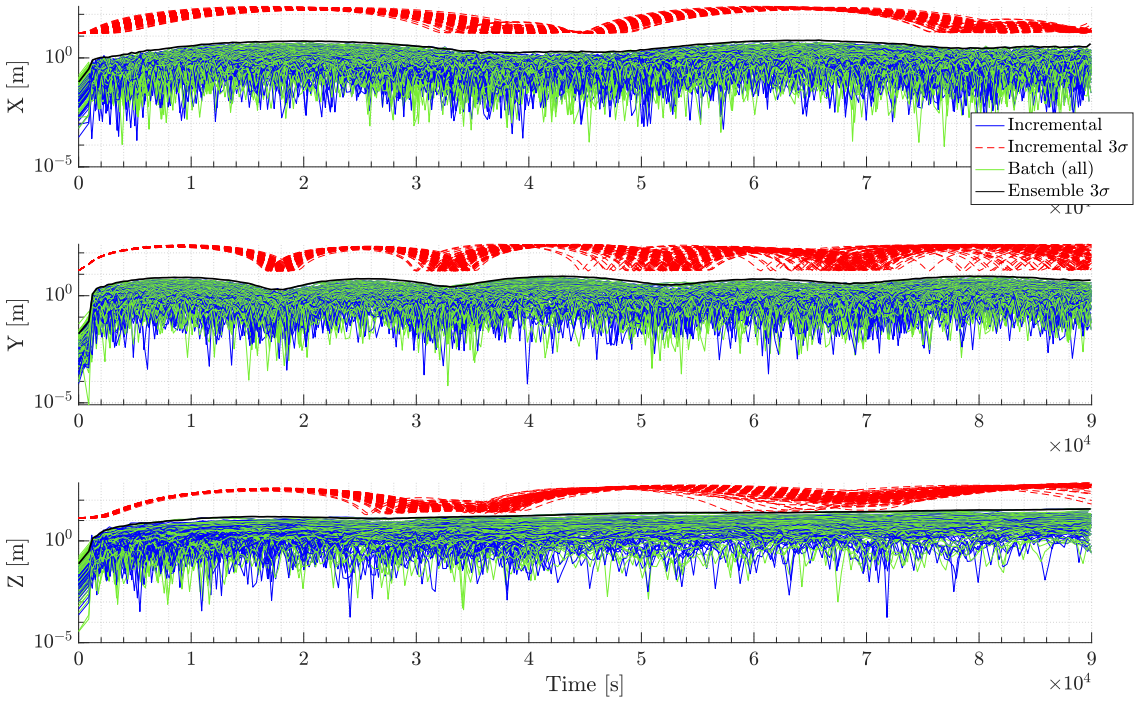

Figure 8: Position estimation errors and covariance in $\mathcal{C}$ reference frame of *ISAM2concurrent*. Note that the y-axis is in logarithmic scale.

Position and velocity estimation Figures 8 and 9 show the estimation error of the position at the keyframes in the camera reference frame for the concurrent and the dynamics variants, respectively. The y-axis is shown in logarithmic scale. The 3σ covariance bounds (red lines) are computed from the diagonal entries of the marginal covariance matrix (i.e., accounting for the contribution of all other states due to correlation). It is highlighted that the incremental solution is here equivalent to the solution the algorithm returns for the spacecraft state at a given keyframe when processing the measurement for the first time. In contrast, the batch solution (green solid lines) is recomputed using the same factor graph and the ISAM2 best solution as

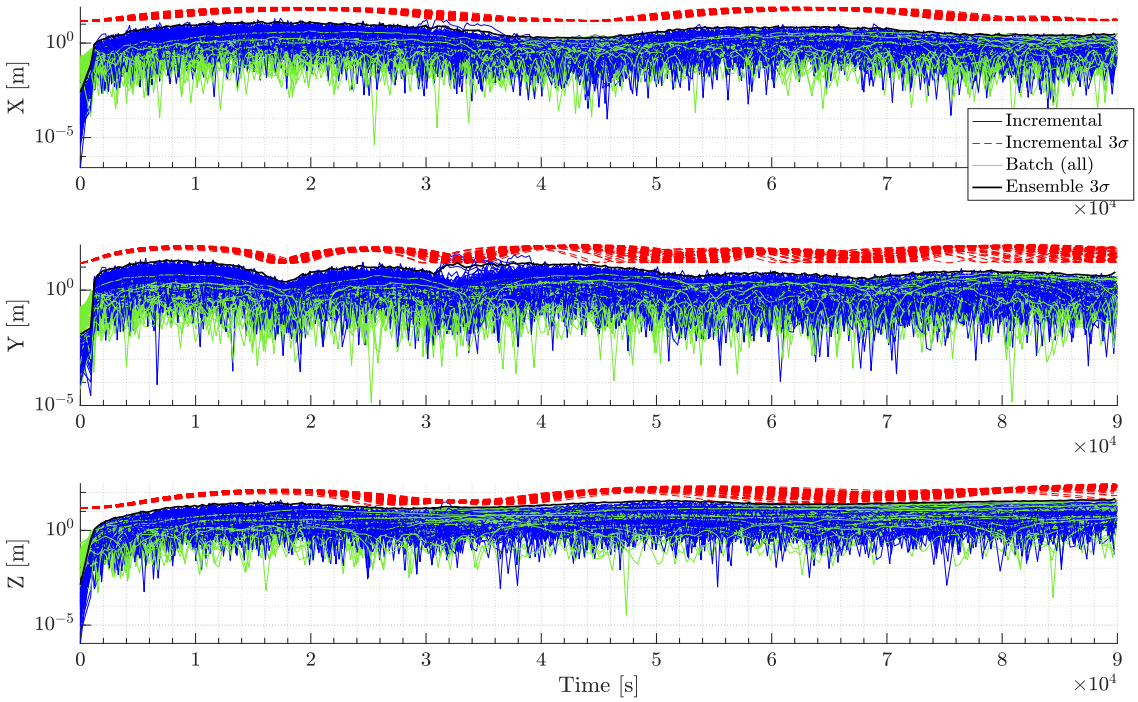

Figure 9: Position estimation errors and covariance in $\mathcal{C}$ reference frame of *ISAM2dynamics*. Note that the y-axis is in logarithmic scale.

initial guess once all keyframe are accumulated. First, it is significant that the batch solution only marginally improves upon the online incremental solution while requiring a computational time of 1 or 2 orders of magnitude with respect to the ISAM2 best estimate. Therefore, there exists margin to gain computational speed at the expense of the accuracy in an actual implementation. In line with previous results, the time evolution of the position error history reveals that the estimates drift along the camera boresight (+Z axis), while both X and Y errors remain bounded. This reflects the fact that the range to the target is only observable through correlations determined by the landmarks' sightings and the dynamics. Significantly, it is was found that *ISAM2concurrent* achieves better accuracy in the position estimation, likely due to the dynamical model mismatch of *ISAM2dynamics*. This happens despite the fact that the filter shares the same dynamical model of *ISAM2dynamics* because *ISAM2concurrent* optimization is only loosely constrained by the dynamics. Additionally, the presence of the 2nd prior factor provides a stronger optimization constraint to fix the metric scale with respect to the dynamical model. Second, in all cases and for both variants, the covariance representation always remains conservative with respect to the estimation error. However, both estimators show underconfidence in their estimation as can be seen qualitatively by comparing the estimated covariance bounds with the true knowledge bounds computed from the Monte Carlo trials. It is noted that these bounds are only an approximation derived by registering the estimation errors evaluated at every keyframe on a common average time-grid. This was required due to the fact that each trial run had different timestamps associated to the keyframes by virtue of the adaptive insertion logic. On a more quantitative basis, Figures 10 and 11 show the computation of the Mahalanobis distance as $D = \sqrt{\delta\mathbf{x}^T P_{\mathbf{x}}^{-1} \delta\mathbf{x}}$ where $\delta\mathbf{x}$ is the estimation error and $P_{\mathbf{x}}$ the estimator covariance. This metric proves the overconfidence mentioned before. It is recalled that, by definition, the estimator is consistent if the Normalized Estimation Error Squared (NEES) approaches the dimensionality d of the estimation error (i.e., 3 in this case).[43] A value of NEES lower than d indicates that the uncertainty of the estimates is over-conservative.

Two time-dependent variations of the covariance bounds are noticeable. The first trend consists in an increase of the covariance volume over time, which is explained by the fact that all the measurements enabling the estimation of the position comes from either the dynamics prior knowledge or the visual observations (or

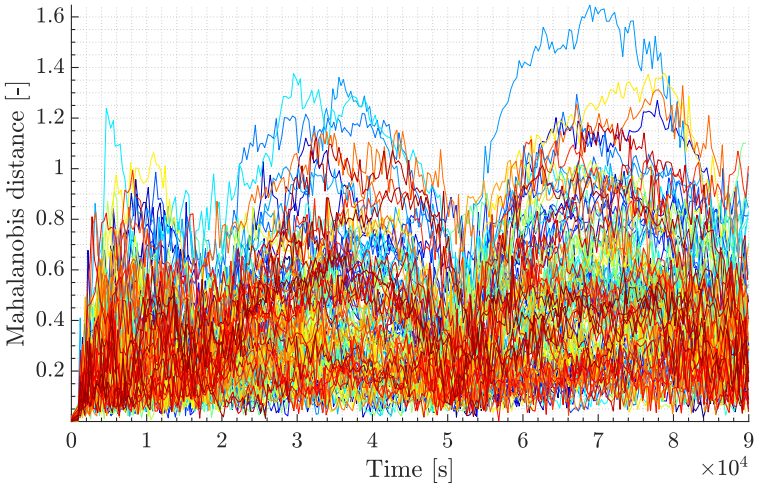

Figure 10: Mahalanobis distance of position estimates of *ISAM2concurrent* in $\mathcal{C}$ reference frame (all trials).

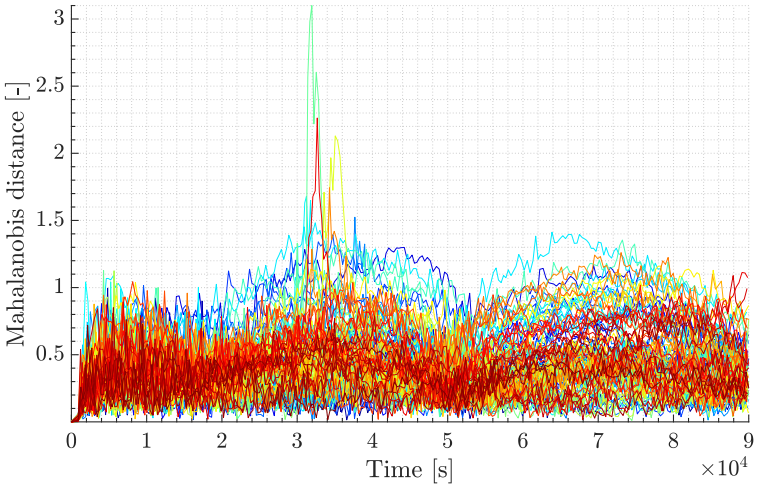

Figure 11: Mahalanobis distance of position estimates of *ISAM2dynamics* in $\mathcal{C}$ reference frame (all trials).

both). Being both "odometry-like", there is no absolute information on the position at all. Furthermore, the error in the feature tracking measurement is accumulated at each frame, which contributes to this issue. In this regard, loop closure constraints would be required to keep this uncertainty bounded over long time horizons.

On the opposite side, the second trend shows a periodic pattern, whose periodicity is similar to the asteroid rotation period of 12 hours. Two contributions were identified in this regard, the first being related to the dynamics, the second with the asteroid shape and the average length of the feature tracks.

1. The covariance associated to the dynamics factor is dynamics-dependent through the propagation of the uncertainty, which may increase either faster or slower depending on the linearization point and the propagation time.

2. Nevertheless, since *ISAM2concurrent* shows similar periodicity even without the dynamics factor, it is conjectured that the asteroid shape matters more. Moreover, this behaviour was found to be present in all trials with practically the same period, but shifted in time. Although it was not possible to single out an exact cause, the main speculation is that the evolution of the rotation influences the length of the

feature tracks, how many features are visible, and - consequently - the number of inserted keyframes as well. Finally, the fact that this pattern is present in the X-Y components expressed in both target body and inertial frames, while absent in the Z component almost aligned with the asteroid rotation axis, strengthens this speculation.

As a final note, it is highlighted that the accuracy of *ISAM2concurrent* is entirely predicated on the second prior pose so that the dynamical model mismatch has virtually no effect on the graph-based estimator, contrarily to *ISAM2dynamics*. The velocity estimation errors and the corresponding uncertainty bounds expressed

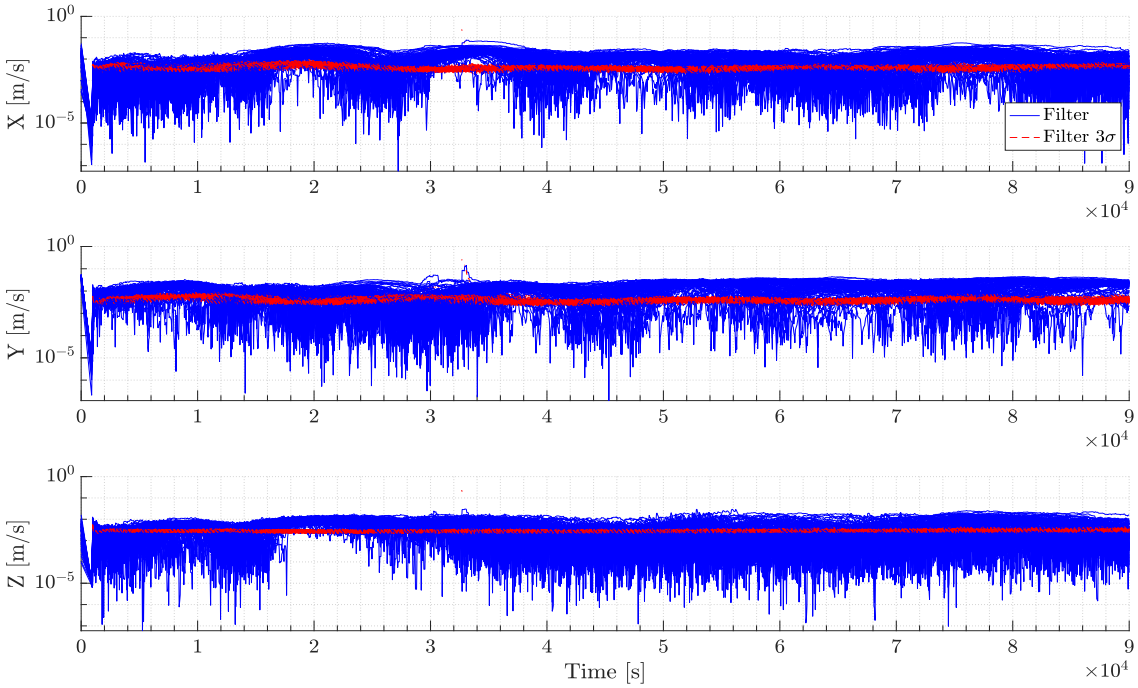

Figure 12: Velocity estimation errors in $\mathcal{TF}$ reference frame of *ISAM2concurrent* (filter). Note that the y-axis is in logarithmic scale.

in the target fixed frame are reported in Figures 12 and 13 for *ISAM2concurrent* and *ISAM2dynamics*, respectively. As a reference, consider that the order of magnitude of the velocity is in the order of 10^{-2} [m/s]. The filter shows an inconsistent and over-optimistic behaviour in most runs, despite the reset of the position provided by the smoother. Also note that the magnitude of the errors is significant with respect to the absolute values of the orbital velocities for this scenario. Nevertheless, the propagation still remains sufficiently accurate to provide a position guess for the smoother. On the other side, for *ISAM2dynamics*, it is first pointed out that the covariance of the velocity shows no evident systematic increase over time. A periodic trend of the uncertainty similar to that of the position is evident in this case as well. More than position estimation, the velocity errors unveil the key difference between *ISAM2concurrent* and *ISAM2dynamics*. Specifically, they prove that the statistical assumptions made in the presented filter design are not sufficiently sound to achieve good accuracy. In fact, this happens despite the fact that spacecraft and landmarks positions are estimated with comparable accuracy by both variants, with the filter position estimates being reset to accurate values by the smoother. On the contrary, the dynamics factor proved to enable velocity estimation by ISAM2 despite the significant mismatch in the dynamics model due to the absence of spherical harmonics accelerations. More in depth, the root cause of the failure of the filter in estimating velocity was mainly found in the landmarks map being used for the measurements updates. In fact, due to the latter being the online estimate of ISAM2, its error is consistent throughout all the landmarks in between keyframes, while this cannot be accounted for by the filter. Consequently, a large time-correlated bias appears in the filter velocity estimates. The issue was also singled out by evaluating the same filter but assuming white noise error on the landmarks available to

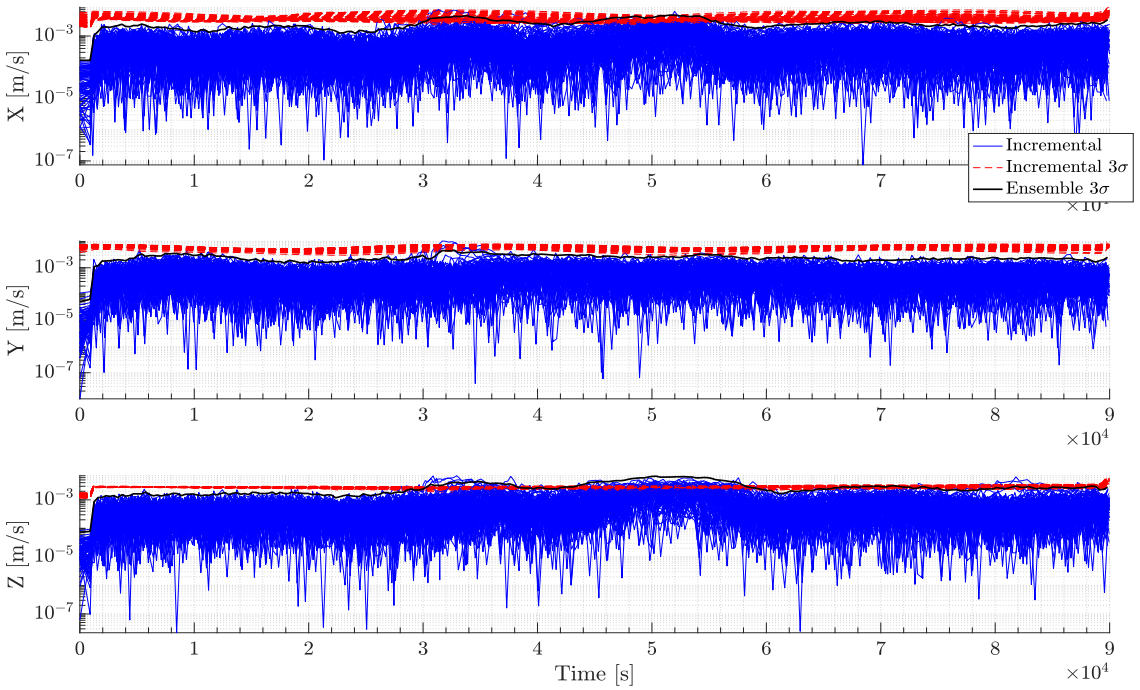

Figure 13: Velocity estimation errors and formal covariance in $\mathcal{TF}$ reference frame of *ISAM2dynamics*. Note that the y-axis is in logarithmic scale.

the filter in place of the ISAM2 estimation error. Notably, the time variant error disappeared in the latter experiment confirming the correctness of the hypothesis. The effect of this bias may be mitigated by estimating it in the filter state vector as a stochastic process (e.g. First Order Gauss-Markov) or by accounting for the landmarks' covariance while processing the measurements.[38] In fact, the poor performance of the filter can be mostly attributed to the simplified design rather than to the concept of *ISAM2concurrent* itself.

Landmarks estimation and reconstruction error Since one of the tasks of SLAM algorithms is the estimation of the environment states, the reconstruction error of the asteroid surface provides useful insights into its performance. Therefore, the Hausdorff distance of the estimated landmarks positions with respect to the dense ground-truth mesh was evaluated.[44] The results at the end of a single trial of *ISAM2dynamics* are reported in Figure 14, showing the interpolated values of the Hausdorff distance superimposed on the ground-truth mesh. Both the variants managed to accurately reconstruct almost the entire asteroid, even though the covariance of these estimates (not reported here for the sake of brevity) remained high, reflecting the covariance of the position estimates. Overall, as expected in SLAM, spacecraft and landmarks estimation errors are similar in magnitude and evolution due to the inherent correlation among them. The peaks of the reconstruction error in Figure 14 are caused by poorly visible regions rather than inaccurate estimates. In fact, these were present in most trials and consistently in the runs of both variants, while being distributed in different locations across the trials. Finally, *ISAM2concurrent* performed slightly worse but still comparably to *ISAM2dynamics*. The main difference between these two was found in the smoothing capability given by the dynamics factor, thanks to the Markov chain it builds between time instants. Specifically, since the front-end tracker does not have the capability of matching previously observed features, *ISAM2concurrent* could not improve landmarks once their tracks become inactive. On the contrary, *ISAM2dynamics* managed to do so, while also enabling the map to remain self-consistent throughout the entire mapping process. However, this capability comes to the cost of largely increased computational time caused by the presence of a flow of information potentially to all past states through the dynamics link. Consequently, the solution is no longer computed in constant time, as more landmarks are added in the graph. In contrast, *ISAM2concurrent* could grant real-time estimation while achieving nearly the same accuracy in both position and map states owing to

the ISAM2 characteristics.

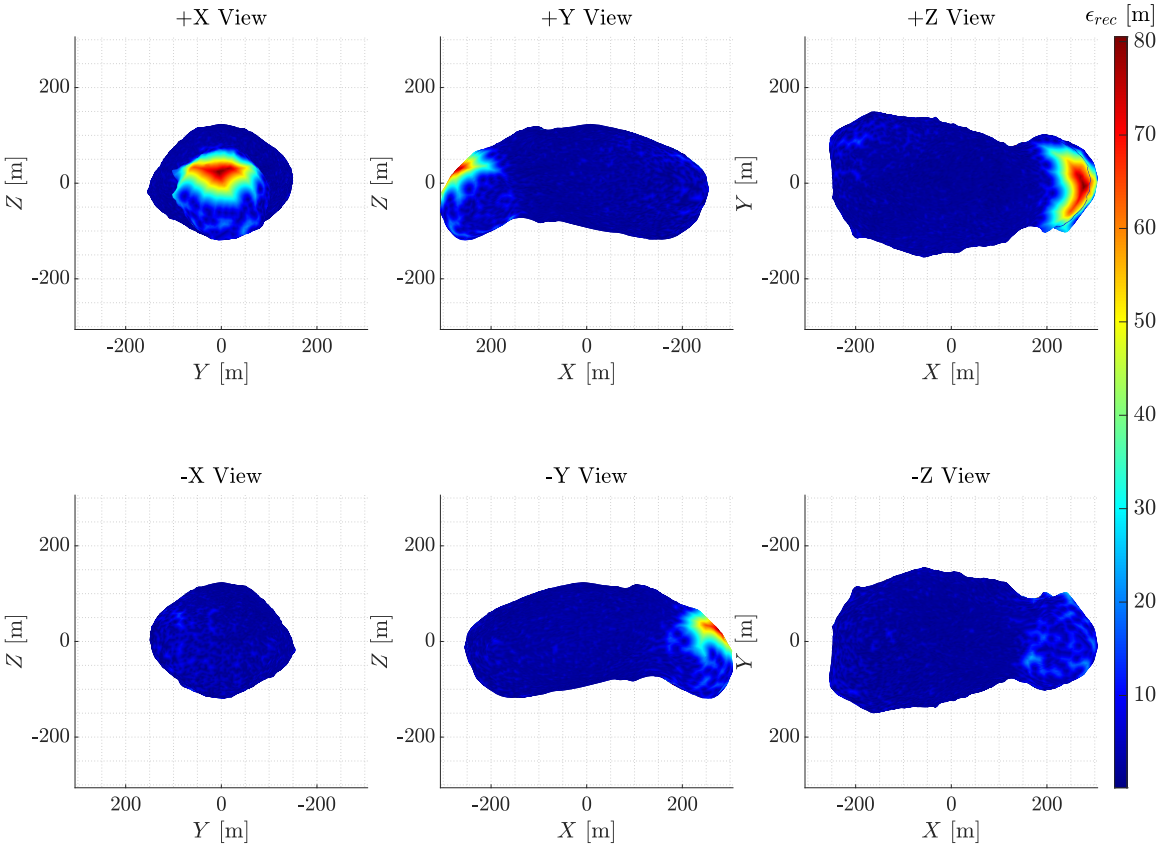

Figure 14: Asteroid reconstruction error for a single-run of *ISAM2dynamics*. Colourmap represents absolute value of Hausdorff distance in meters.

CONCLUSIONS

This study presented two variants of a visual graph-based SLAM algorithm for asteroid exploration, where the dynamics information is embedded either as a kinematic constraint acting the position or directly as a dynamics factor. The *ISAM2concurrent* features a tightly coupled approach to integrate the current practice in on-board spacecraft navigation, namely navigation filters, with advanced smoothing techniques from the robotics field. The limitations of the proposed approach were evaluated by comparing it with *ISAM2dynamics*, where the dynamics is introduced directly into the factor graph. Possible development paths to address the described issues include use of consider parameters, estimation of unmodelled effects in both filter and smoother, adaptive noise parameters, and different measurement models, such as the direction of motion, not relying on the map from the SLAM algorithm.[32,45] The present analysis relied on assumptions of perfect prior knowledge and known asteroid rotation. Future work will focus on relaxing these assumptions by integrating joint asteroid parameter estimation into the SLAM process. Moreover, efficient and reliable ways of implementing loop closures tailored for small bodies exploration are needed and will be investigated. An adaptive keyframe insertion logic was also presented, which poses the basis for a future ORB-SLAM-like culling policy, deemed necessary to achieve long-term computationally sustainable autonomous navigation. Finally, a simple model was proposed to emulate the performance of an optical tracking system, enabling Monte Carlo analysis to be conducted in a fraction of the time required for simulations with rendering of images in the loop while retaining representative errors.

ACKNOWLEDGMENT

This work has been conducted under the COSMICA project funded by the Italian Ministry of University and Research (MUR) through Decree No. R.0000398.24-10-2022.h.10:43.

REFERENCES

[1] M. Yoshikawa, J. Kawaguchi, A. Fujiwara, and A. Tsuchiyama, "The hayabusa mission," *Sample return missions*, pp. 123–146, Elsevier, 2021.

[2] Y. Tsuda, M. Yoshikawa, M. Abe, H. Minamino, and S. Nakazawa, "System design of the Hayabusa 2—Asteroid sample return mission to 1999 JU3," *Acta Astronautica*, Vol. 91, 2013, pp. 356–362.

[3] K. Berry, K. M. Getzandanner, M. C. Moreau, S. M. Rieger, P. G. Antreasian, C. D. Adam, D. Wibben, J. M. Leonard, A. H. Levine, J. Geeraert, *et al.*, "Contact with Bennu! Flight performance versus prediction of OSIRIS-REx "TAG" sample collection," *AIAA Scitech 2022 Forum*, 2022, p. 2521.

[4] D. S. Lauretta, S. S. Balram-Knutson, E. Beshore, W. V. Boynton, C. Drouet d'Aubigny, D. N. DellaGiustina, *et al.*, "OSIRIS-REx: Sample Return from Asteroid (101955) Bennu," *Space Science Reviews*, Vol. 212, Oct. 2017, pp. 925–984, 10.1007/s11214-017-0405-1.

[5] B. Williams, P. Antreasian, E. Carranza, C. Jackman, J. Leonard, D. Nelson, B. Page, D. Stanbridge, D. Wibben, K. Williams, M. Moreau, K. Berry, K. Getzandanner, A. Liounis, A. Mashiku, D. Highsmith, B. Sutter, and D. S. Lauretta, "OSIRIS-REx Flight Dynamics and Navigation Design," *Space Science Reviews*, Vol. 214, June 2018, p. 69, 10.1007/s11214-018-0501-x.

[6] B. T. Tropf, M. Haque, N. Behrooz, and C. Krupiarz, "The DART autonomy system," *2023 IEEE 9th International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, IEEE, 2023, pp. 104–113.

[7] A. F. Cheng, H. F. Agrusa, B. W. Barbee, A. J. Meyer, T. L. Farnham, S. D. Raducan, D. C. Richardson, E. Dotto, A. Zinzi, V. Della Corte, *et al.*, "Momentum transfer from the DART mission kinetic impact on asteroid Dimorphos," *Nature*, Vol. 616, No. 7957, 2023, pp. 457–460.

[8] N. L. Chabot, A. S. Rivkin, A. F. Cheng, O. S. Barnouin, E. G. Fahnestock, D. C. Richardson, A. M. Stickle, C. A. Thomas, C. M. Ernst, R. T. Daly, *et al.*, "Achievement of the planetary defense investigations of the Double Asteroid Redirection Test (DART) mission," *The Planetary Science Journal*, Vol. 5, No. 2, 2024, p. 49.

[9] P. Michel, M. Küppers, A. C. Bagatin, B. Carry, S. Charnoz, J. De Leon, A. Fitzsimmons, P. Gordo, S. F. Green, A. Hérique, *et al.*, "The ESA Hera mission: detailed characterization of the DART impact outcome and of the binary asteroid (65803) Didymos," *The planetary science journal*, Vol. 3, No. 7, 2022, p. 160.

[10] A. PELLACANI, M. GRAZIANO, M. FITTOCK, J. GIL, and I. CARNELLI, "HERA vision based GNC and autonomy," 2019, p. 14 pages. Artwork Size: 14 pages Medium: PDF Publisher: Proceedings of the 8th European Conference for Aeronautics and Space Sciences. Madrid, Spain, 1-4 july 2019, 10.13009/EUCASS2019-39.

[11] M. Pugliatti, A. Rizza, F. Piccolo, V. Franzese, C. Bottiglieri, C. Giordano, F. Ferrari, and F. Topputo, "The Milani mission: overview and architecture of the optical-based GNC system," *AIAA Scitech 2022 Forum*, 2022, p. 2381.

[12] P. Panicucci, *Autonomous vision-based navigation and shape reconstruction of an unknown asteroid during approach phase*. PhD thesis, Toulouse, ISAE, 2021.

[13] S. Julier and J. Uhlmann, "A counter example to the theory of simultaneous localization and map building," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, Vol. 4, Seoul, South Korea, IEEE, 2001, pp. 4238–4243, 10.1109/ROBOT.2001.933280.

[14] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, IEEE, May 2010, pp. 2657–2664, 10.1109/ROBOT.2010.5509636.

[15] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *The International Journal of Robotics Research*, Vol. 23, Aug. 2004, pp. 693–716. Publisher: SAGE Publications Ltd STM, 10.1177/0278364904045479.

[16] M. W. Givens and J. W. McMahon, "Square-Root Extended Information Filter for Visual-Inertial Odometry for Planetary Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 46, Feb. 2023, pp. 231–245, 10.2514/1.G006849.

[17] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," *Foundations and Trends in Robotics*, Vol. 6, No. 1-2, 2017, pp. 1–139, 10.1561/2300000043.

[18] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," *The International Journal of Robotics Research*, Vol. 25, Dec. 2006, pp. 1181–1203, 10.1177/0278364906072768.

[19] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, IEEE, May 2011, pp. 3607–3613, 10.1109/ICRA.2011.5979949.

[20] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, Vol. 31, Feb. 2012, pp. 216–235, 10.1177/0278364911430419.

[21] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, and G. Furano, "High-Performance Embedded Computing in Space: Evaluation of Platforms for Vision-Based Navigation," *Journal of Aerospace Information Systems*, Vol. 15, Apr. 2018, pp. 178–192, 10.2514/1.I010555.

[22] M. Dor, K. A. Skinner, P. Tsiotras, and T. Driver, "Visual SLAM for Asteroid Relative Navigation," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Nashville, TN, USA, IEEE, June 2021, pp. 2066–2075, 10.1109/CVPRW53098.2021.00235.

[23] M. Dor, T. Driver, K. Getzandanner, and P. Tsiotras, "AstroSLAM: Autonomous Monocular Navigation in the Vicinity of a Celestial Small Body – Theory and Experiments," Dec. 2022. arXiv:2212.00350 [cs, eess], 10.48550/arXiv.2212.00350.

[24] M. Dor, T. Driver, K. Getzandanner, and P. Tsiotras, "AstroSLAM: Autonomous monocular navigation in the vicinity of a celestial small body—Theory and experiments," *The International Journal of Robotics Research*, June 2024, p. 02783649241234367, 10.1177/02783649241234367.

[25] T. P. Setterfield, D. W. Miller, J. J. Leonard, and A. Saenz-Otero, "Mapping and determining the center of mass of a rotating object using a moving observer," *The International Journal of Robotics Research*, Vol. 37, Jan. 2018, pp. 83–103, 10.1177/0278364917749024.

[26] D. Nakath, J. Clemens, and C. Rachuy, "Active Asteroid-SLAM: Active Graph SLAM with Landing Site Discovery in a Deep Space Proximity Operations Scenario," *Journal of Intelligent & Robotic Systems*, Vol. 99, No. 2, 2019, pp. 303–333, 10.1007/s10846-019-01103-0.

[27] H. Loeliger, "An Introduction to factor graphs," *IEEE Signal Processing Magazine*, Vol. 21, Jan. 2004, pp. 28–41, 10.1109/MSP.2004.1267047.

[28] C. Taylor and J. Gross, "Factor Graphs for Navigation Applications: A Tutorial," *NAVIGATION: Journal of the Institute of Navigation*, Vol. 71, No. 3, 2024, p. navi.653, 10.33012/navi.653.

[29] D. Scheeres, R. Gaskell, S. Abe, O. Barnouin-Jha, T. Hashimoto, J. Kawaguchi, T. Kubota, J. Saito, M. Yoshikawa, N. Hirata, T. Mukai, M. Ishiguro, T. Kominato, K. Shirakawa, and M. Uo, "The Actual Dynamical Environment About Itokawa," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keystone, Colorado, American Institute of Aeronautics and Astronautics, Aug. 2006, 10.2514/6.2006-6661.

[30] D. A. Vallado, *Fundamentals of astrodynamics and applications*, Vol. 12. Springer Science & Business Media, 2001.

[31] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," *International Journal of Computer Vision*, Vol. 56, Feb. 2004, pp. 221–255, 10.1023/B:VISI.0000011205.11775.fd.

[32] F. Piccolo, C. Balossi, P. Panicucci, M. Pugliatti, F. Topputo, F. Capolupo, *et al.*, "Resource-Constrained Vision-Based Relative Navigation About Small Bodies," *46th AAS Guidance, Navigation and Control Conference*, 2024, pp. 1–18.

[33] J.-Y. Bouguet *et al.*, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel corporation*, Vol. 5, No. 1-10, 2001, p. 4.

[34] T. Moller and B. Trumbore, "Fast Minimum Storage Ray Triangle Intersection," 2005.

[35] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, "The Bayes tree: An algorithmic foundation for probabilistic robot mapping," *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*, Springer, 2011, pp. 157–173.

[36] J. L. Geeraert and J. W. McMahon, "Square-Root Unscented Schmidt–Kalman Filter," *Journal of Guidance, Control, and Dynamics*, Vol. 41, Jan. 2018, pp. 280–287, 10.2514/1.G002921.

[37] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical orbit determination*. Amsterdam ; Boston: Elsevier Academic Press, 2004.

[38] J. R. Carpenter and C. N. D'Souza, "Navigation Filter Best Practices," 2018.

[39] R. Zanetti and R. H. Bishop, "Kalman filters with uncompensated biases," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 327–335.

[40] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, Second Edition*. Cambridge University Press, 2004.

[41] B. E. Tweddle, A. Saenz-Otero, J. J. Leonard, and D. W. Miller, "Factor Graph Modeling of Rigid-body Dynamics for Localization, Mapping, and Parameter Estimation of a Spinning Object in Space," *Journal of Field Robotics*, Vol. 32, No. 6, 2013, pp. 897–933. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21548, 10.1002/rob.21548.

[42] B. Morrell, J. Villa, and A. Harvard, "Automatic Feature Tracking on Small Bodies for Autonomous Approach," 2017.

[43] D. Woodbury and J. Junkins, "On the Consider Kalman Filter," *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, Canada, American Institute of Aeronautics and Astronautics, Aug. 2010, 10.2514/6.2010-7752.

[44] P. Panicucci, R. Brochard, J. Lebreton, R. Lefez, E. Zenou, and M. Delpech, "Localization and Mapping Merging Silhouettes Information and Feature Tracking for Small Body Applications," *Proceedings of the 9th International ESA conference on Guidance, Navigation and Control Systems*, 2021.

[45] J. A. Christian, L. Hong, P. McKee, R. Christensen, and T. P. Crain, "Image-Based Lunar Terrain Relative Navigation Without a Map: Measurements," *Journal of Spacecraft and Rockets*, Vol. 58, Jan. 2021, pp. 164–181, 10.2514/1.A34875.