3D reconstruction of an asteroid shape using visual SLAM for autonomous navigation

Arunkumar Rathinam*, Andrew G. Dempster*

* *Australian Center for Space Engineering Research (ACSER), University of New South Wales, Australia, 2052*

Summary: Exploration missions to small celestial bodies are well known for their challenges in navigation during proximity operations. To ensure mission's success, a robust GNC system is needed on-board the spacecraft. The shift in focus towards developing autonomous on-board navigation techniques is to achieve better accuracy especially during rendezous, touch-down and landing operations in future missions. In the past, optical navigation was carried out using the reference craters and shape model, and this helped the spacecraft to be semi-autonomous over a certain mission period. To be fully autonomous, the spacecraft must be able to perceive its environment and estimate its relative motion with respect to the asteroid. This work presents the graph-SLAM approach to reconstruct the shape of the rotating asteroid using the distinct landmarks features seen on the asteroid surface. Probabilistic models of spacecraft's motion, asteroid's rotation and landmark measurements are used to construct motion and measurement factors and are assembled in a graph during every key frame. Matrix factorization and iterative state error minimization enable the approach to solve for the maximum-a-posteriori estimate. Simulated experiments are discussed and the results suggest that the proposed approach can accurately reconstruct the shape of the asteroid and estimate the unknown parameters.

Keywords: visual navigation; SLAM; asteroids; state estimation; factor graph; shape reconstruction

Introduction

Scientific exploration and resource utilization are two major ideas behind the growing interests towards missions to small celestial bodies (SCB). Asteroids have a complex dynamic environment around them, accompanied by their varying shape and size, makes the spacecraft navigation around them extremely difficult. Each mission needs careful planning and detailed simulations before performing a critical task. Missions in the past used ground based observations to estimate the spacecraft position and orientation as well as the asteroid's pose and this resulted in large errors in state estimates and increased uncertainty. Recent missions have the optical navigation module as a standard subsystem to operate either in tandem with the ground based measurements or stand alone (during later stages of mission) to autonomously perform limited task. To achieve full autonomy, the spacecraft must have a precise estimate of its state, perceive its environment and robustly predict future states. Reconstruction of the asteroid geometry is an essential component for autonomous navigation in future missions. Knowing the geometry helps the spacecraft to constantly adjust its model relatively to achieve the best estimates. Navigating and mapping an unknown environment is an interdependent problem and can be solved using a typical Simultaneous Localization and Mapping (SLAM) framework.

This paper presents a SLAM approach to reconstruct the shape of asteroids that are rotating on the principal axis. A sparse information matrix is constructed based on a factor graph with the motion factors represented by the dynamic motion models of the spacecraft and asteroid, and the measurement factors through the measurements of the distinct landmarks that are visible on the asteroid's surface. The approach uses manifold optimization to represent the error in tangent space which is directly included via a composition operator and thus keeping the state in original manifold space. This work uses sparse Cholesky factorization of the Hessian matrix to compute the solution update. The final section presents a detailed description of the simulations and the results achieved using the approach on the asteroid Itokawa point cloud dataset.

SLAM formulation

Localization and map building are fundamental problems in the field of mobile robotics; in a typical SLAM framework, a robot uses its sensors to observe the environment in order to build a map of the environment while it simultaneously localizes itself in the map. Modern SLAM approaches are grouped into two categories: *filtering* and *smoothing*. The *filtering* approach tracks the most recent pose and landmark state in the environment, while all other poses are marginalized out after every frame (except for the landmarks that may be measured in the future). Filtering-based SLAM approaches are very popular and the extended Kalman Filter (EKF SLAM) is widely used. It calculates a Gaussian posterior over the robot's pose as well as the detected landmarks [1]. Other popular filtering approaches include particle filters [2] and information filters [3].The *smoothing* approach considers optimizing over the entire history of poses and measurements (it only discards measurements that don't contribute to the state estimates). Compared to filtering, the smoothing approach produces a graph that has more elements and these are sparsely inter-connected. Though smoothing approaches require more computation effort, recent advancements on graph-based SLAM [4] exploit the sparse nature of the nodes during optimization and efficiently solve for the state estimates over the entire trajectory [5].

Related Work

SLAM-based state estimation was investigated for a few space mission scenarios, especially for non-cooperative targets. It is grouped into two main categories, *Non-cooperative known target* (includes non cooperating satellites in orbit servicing scenarios) and *Non-cooperative unknown target* (includes asteroids, comets). Augenstein ([6], [7]) studied estimation of the pose of a tumbling target using a hybrid approach based on particle filters. Cocaud ([8], [9]) studied the spacecraft's pose estimation near small celestial bodies for near real-time navigation. Cocaud's approach using a particle filter-based monocular visual SLAM has the advantage of efficient integration of the nonlinear models, but the results suggest that the robustness of the approach depends on the careful selection of parameters such as the number of particles, observation errors, etc. Tweddle [10] studied the factor graph modeling of rigid body dynamics to estimate the state of a non cooperative object spinning about an arbitrary axis. Olson [11] studied EKF-SLAM to estimate the spin state of the small bodies along with its mass, moment of inertia and the estimate of spacecraft's pose and landmark locations. Takeishi [12] studied the dynamic visual SLAM for asteroid navigation using a graph-based

approach, but used the dynamic model with the other sensors combination including attitude and inertial sensors.

Preliminaries

Dynamics of Spacecraft

The pose of the spacecraft at time instance *i* is denoted by $\mathbf{x}_i^s$, where $\{i = 1, ..., N\}$ and includes position **r** = $\begin{bmatrix} x & y & z \end{bmatrix}'$, orientation **q** represented as quaternion $\begin{bmatrix} q_0 & q_x & q_y & q_z \end{bmatrix}'$, linear velocity $\mathbf{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}'$ and angular velocity $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}'$.

$$\mathbf{x}_i^s = \begin{bmatrix} r & q & v & \omega \end{bmatrix}' \tag{1}$$

The spacecraft is assumed to be orbiting around the asteroid and the motion of the satellite is given by the dynamic equations below. $\dot{r}$ and $\dot{v}$ represent the derivatives of position and velocity, while $\dot{q}$ and $\dot{\omega}$ the derivatives of quaternion and angular velocity repectively.

$$\dot{r} = v_s \tag{2}$$

$$\dot{q} = \frac{1}{2} W(q_s)^\top \omega_s \tag{3}$$

where quaternion rates matrix $W : \mathbb{H} \to \mathbb{R}^{3\times4}$, $\quad W(q) = \begin{bmatrix} -q_x & q_0 & -q_z & q_y \\ -q_y & q_z & q_0 & -q_x \\ -q_z & -q_y & q_x & q_0 \end{bmatrix}$

$$\dot{v} = \frac{-\mu}{\|r^3\|}\mathbf{r} + Q_{vs} \tag{4}$$

$$\dot{\omega} = -J^{-1}[\omega\times]J\omega_s + Q_{\omega s} \tag{5}$$

where, J is inertia matrix $\quad [\omega\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$

Q_v, Q_ω process noise model including disturbance forces, torques applied to the spacecraft and μ standard gravitational parameter.

Dynamics of the asteroid

The asteroid is assumed to be rotating on the principal axis and follows rigid body dynamics. The asteroid's state is represented similarly to the spacecraft's state with a 13 state vector $\mathbf{x}_a^i$. The derivative of the state is presented below,

$$\dot{r} = v_a \tag{6}$$

where $v_a \sim 0$

$$\dot{q} = \frac{1}{2} W(q_a)^\top \omega_a \tag{7}$$

$$\dot{v} = Q_{va} \tag{8}$$

where Q_{va} is assumed to be very small disturbance force

$$\dot{\omega} = -J^{-1}\left[\omega\times\right] J\omega_a + Q_{\omega a} \tag{9}$$

Methodology

The overall approach combines discrete-time state-space representations of the spacecraft, asteroid and landmarks into graph representations and uses matrix factorization to solve for error minimization and in turn estimate maximum-a-posteriori. A factor graph is a bipartite graph and has two kinds of nodes, the factor nodes and the variable nodes. An edge always joins a variable node to a factor node and thus the graph is bipartite. Factor nodes represent the probability distribution between the two nodes and often it represents the error between the two variables. Variable nodes represent the state that we want to estimate (including the spacecraft's pose, the asteroid's pose and landmark positions). $f(a, b)$ denotes the factor between the random variables a and b.

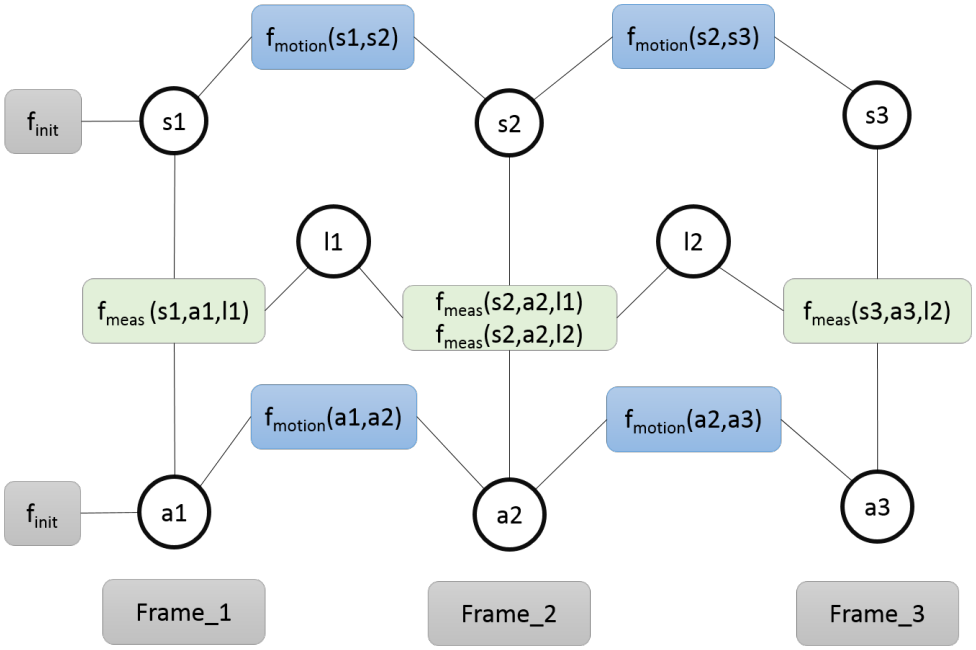

Fig. 1: Factor graph representation of the framework with variables (spacecraft state, landmark position, asteroid state) and factors (measurement and motion)

The conditional probabilities that constitute the motion factors are extracted from the motion models described in the earlier section. Whereas for the measurement factors, a generalised pinhole camera model with depth measurement was used. The spacecraft's state and asteroid's state are denoted by $\mathbf{x}_i^s, \mathbf{x}_i^a$ with Gaussian noise variables $\mathbf{w}_i^s, \mathbf{w}_i^a$ with zero mean and covariances $\mathbf{\Omega}_{si}^{-1}, \mathbf{\Omega}_{ai}^{-1}$ respectively. $\mathbf{v}_k$ is the measurement noise with zero mean and covariance $\mathbf{\Omega}_k^{-1}$.

$$\mathbf{x}_i^s = f_i(\mathbf{x}_{i-1}^s, \mathbf{u}_i^s) + \mathbf{w}_i^s, \quad \mathbf{w}_i^s \sim \mathcal{N}\{0, \mathbf{\Omega}_{si}^{-1}\} \tag{10}$$

$$\mathbf{x}_i^a = g_i(\mathbf{x}_{i-1}^a, \mathbf{u}_i^a) + \mathbf{w}_i^a, \quad \mathbf{w}_i^a \sim \mathcal{N}\{0, \mathbf{\Omega}_{ai}^{-1}\} \tag{11}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_{ik}^s, \mathbf{x}_{ik}^a, \mathbf{l}_{j_k}) + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}\{0, \mathbf{\Omega}_k^{-1}\} \tag{12}$$

The motion and measurement factors are represented as below.

$$\phi_i^s = P(\mathbf{x}_i^s \mid \mathbf{x}_{i-1}^s, \mathbf{u}_i^s) \propto exp\Big(-\frac{1}{2}(\mathbf{x}_i^s - f_i(\mathbf{x}_{i-1}^s, \mathbf{u}_i^s))^T \mathbf{\Omega}_{si} (\mathbf{x}_i^s - f_i(\mathbf{x}_{i-1}^s, \mathbf{u}_i^s))\Big) \tag{13}$$

$$\phi_i^a = P(\mathbf{x}_i^a \mid \mathbf{x}_{i-1}^a, \mathbf{u}_i^a) \propto exp\Big(-\frac{1}{2}(\mathbf{x}_i^a - g_i(\mathbf{x}_{i-1}^a, \mathbf{u}_i^a))^T \mathbf{\Omega}_{ai} (\mathbf{x}_i^a - g_i(\mathbf{x}_{i-1}^a, \mathbf{u}_i^a))\Big) \tag{14}$$

$$\phi_k = P(\mathbf{z}_k \mid \mathbf{x}_{ik}^s, \mathbf{x}_{ik}^a, \mathbf{l}_{j_k}) \propto exp\Big(-\frac{1}{2}(\mathbf{z}_k - h_k(\mathbf{x}_{ik}^s, \mathbf{x}_{ik}^a, \mathbf{l}_{j_k}))^T \mathbf{\Omega}_k (\mathbf{z}_k - h_k(\mathbf{x}_{ik}^s, \mathbf{x}_{ik}^a, \mathbf{l}_{j_k}))\Big) \tag{15}$$

Considering the general form, the eqn.13, 14 and 15 simplify as below.

$$\phi_k = exp(-\frac{1}{2} e_k^T \mathbf{\Omega}_k e_k) \tag{16}$$

It is evident from eqn.16 that the difference between the motion and measurement factors is not important as long as we calculate the error from the k^{th} measurement. The joint probability for the motion and the measurement can be written as the product of all factors[13].

$$P(\mathbf{x}, \mathbf{z}) \propto \prod_{k=1}^{K} \phi_k \propto \prod_{k=1}^{K} exp(-\frac{1}{2} e_k^T \mathbf{\Omega}_k e_k) \tag{17}$$

Maximizing this probability density function (PDF) is equivalent to minimizing its negative log-likelihood

$$F(x) \triangleq -\log P(\mathbf{x}, \mathbf{z}) \tag{18}$$

and by solving the above cost function eqn.we geometry

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{k=1}^{K} e_k(\mathbf{x}_i, \mathbf{x}_j)^T \mathbf{\Omega}_k e_k(\mathbf{x}_i, \mathbf{x}_j) \tag{19}$$

Optimization on the Manifold

A manifold ($\breve{\cdot}$) is a topological space that locally resembles Euclidean space near each point. To optimize the pose estimate we consider the local tangent space where the error can be represented as minimal dimensions. In tangent space, $\Delta\phi$ represents the correction needed for the position estimate, Δq local error quaternion, a composition operator $\oplus$ projects the tangent space on to the manifold. Whereas for the position estimate

$$x_p = \breve{x}_p \oplus \Delta x_p \triangleq \breve{x}_p + \Delta x_p \tag{20}$$

the local quaternion error is,

$$\Delta q = \begin{bmatrix} \sqrt{1 - \|\Delta\phi\|^2} \\ \Delta\phi \end{bmatrix} \tag{21}$$

projection on the manifold is done through the composition operator. ($\otimes$: quaternion product)

$$\breve{q} \oplus \Delta\phi \triangleq \breve{q} \otimes \begin{bmatrix} \sqrt{1 - \|\Delta\phi\|^2} \\ \Delta\phi \end{bmatrix} \tag{22}$$

Algorithm 1 SLAM Framework

Initialization

$\{\mathbf{r}_c\} \leftarrow$ *estimateCoR* $\left\{\mathbf{l}_k^i\left(u,v,z\right)\right\}$

where u,v are pixel coordinates, z depth

$\left\{\mathbf{x}_a\left(\mathbf{r}_c\right),\mathbf{l}_k^i,\mathbf{x}_s\right\} \leftarrow$ *invTransform* $\left\{\mathbf{r}_c,\mathbf{l}_k^i,\mathbf{x}_s\right\}$

$\left\{\mathbf{x}_a^1,\mathbf{l}_k^1,\mathbf{x}_s^1\right\} \leftarrow$ *initialize* $\left\{\mathbf{x}_a\left(\mathbf{r}_c\right),\mathbf{l}_k^i,\mathbf{x}_s\right\}$

`Factor_graph` $\leftarrow$ *addFactors* $\left\{\mathbf{x}_a^1,\mathbf{l}_k^1,\mathbf{x}_s^1\right\}$

Prediction and Estimation

for $i = 2,...,N$ **do**

$\left\{\dot{\mathbf{x}}_a^i,\dot{\mathbf{x}}_s^i\right\} \leftarrow$ *predMotion* $\left\{\mathbf{x}_a^{i-1},\mathbf{x}_s^{i-1},dt\right\}$

`Factor_graph` $\leftarrow$ *addMotionFactor* $\left\{\dot{\mathbf{x}}_a^i,\dot{\mathbf{x}}_s^i\right\}$

if *newLandmarkFeature* **then**

`Factor_graph` $\leftarrow$ *initNewLmk* $\left\{\mathbf{l}_k^i\left(u,v,z\right)\right\}$

`Factor_graph` $\leftarrow$*addMeasFac*$\left\{\mathbf{l}_k^i,\dot{\mathbf{x}}_a^i,\dot{\mathbf{x}}_s^i\right\}$

else

`Factor_graph` $\leftarrow$ *addMeasFac*$\left\{\mathbf{l}_k^i,\dot{\mathbf{x}}_a^i,\dot{\mathbf{x}}_s^i\right\}$

end if

$\left\{\mathbf{x}_a^i,\mathbf{l}_k^i,\mathbf{x}_s^i\right\} \leftarrow$ *solveGraph* {`Factor_graph`}

end for

Fig. 2: Factor graph SLAM framework

Error minimization

The error function is approximated using the first order Taylor series expansion around the current state $\check{\mathbf{x}}$ as mentioned in [4].

$$\begin{aligned} F_k(\check{\mathbf{x}}+\Delta\mathbf{x}) &= e_k(\check{\mathbf{x}}+\Delta\mathbf{x})^T\boldsymbol{\Omega}_k e_k(\check{\mathbf{x}}+\Delta\mathbf{x}) \\ &= (\check{\mathbf{e}}_k+\mathbf{J}_k\Delta\mathbf{x})^T\boldsymbol{\Omega}_k(\check{\mathbf{e}}_k+\mathbf{J}_k\Delta\mathbf{x}) \end{aligned} \tag{23}$$

$$F_k(\check{\mathbf{x}}+\Delta\mathbf{x}) = c + 2\mathbf{b}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x} \tag{24}$$

The matrix $\mathbf{H}$ denotes the sparse information matrix and the quadratic form in 24 helps minimizing the cost function $F_k(\check{\mathbf{x}}+\Delta\mathbf{x})$ by taking the derivative with respect to $\Delta\mathbf{x}$.

$$\mathbf{H}\Delta\mathbf{x}^* = -\mathbf{b} \tag{25}$$

If the Hessian matrix $\mathbf{H}$ is non-singular, a unique solution to above equation is achieved through sparse Cholesky factorization. Once the solution is arrived at, the state vector is updated and the process is iterated until convergence [13].

Experiment

The point cloud of the asteroid Itokawa available here [14] is used for the simulation. The original point cloud data has more than 25000 points, but for our simultion we selected approx. 1200 random points and the simulated model of the asteroid is shown in Fig.3. Experimenting with a known dataset allows for the validation of results against truth data. The spacecraft is assumed to be in circular orbit around the asteroid at approx 600 meters from the center of the asteroid. The simulated spacecraft is equipped with a navigational camera of focal length 125mm and has a sensor array of 1024 x 1024 pixels, with each pixel measuring 8.5

microns to constitute the field of view of 68°. The above specifications are comparable to the navigation camera used in the SCB exploration missions[15]. The rotational period of the asteroid is considered to be around 12hr and the initial angular velocity is assumed to be known approximately through light curve measurements. The graph SLAM toolbox [16] was extended with additional functionality to perform the simulations. The observation error of the navigation camera was assumed to be zero mean and 0.5 pix STD and the altitude sensor depth error STD is 1m. The simulation covers an entire duration of 120 key-frames which is enough to revisit the particular landmark on the surface of the asteroid 2∼3 times.

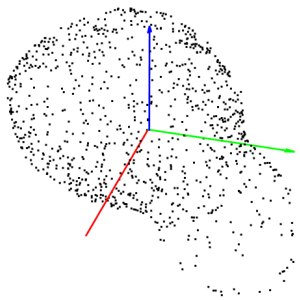

Fig. 3: Pointcloud data of the asteroid Itokawa

Landmark search Optimization

During each keyframe, a new image is acquired by the navigation camera and the distinct landmarks are extracted. To construct the measurement factors during each keyframe, a subset of the landmarks that are both new and known previously are used. In previous SLAM studies, the visible landmarks are either selected randomly or as a whole set is included in the estimation process. This is relatively ineffective because it either select the landmarks with less probability to contribute in the estimation process or increases the computation load with number of landmarks. Propogating all the known landmarks and matching with the visible landmarks is computationally expensive. Since we want to efficiently compute the state, we restricted the number of landmarks included in the estimation process during each keyframe. For e.g. even if 50 landmarks visible, we restricted ourselves to the 30 landmarks with equal distribution of new landmarks and previously known landmarks for particular keyframe. This minimal landmark selection strategy brings another perspective on how to select known landmarks to propogate and match for the current frame. If we select 30 landmarks for each frame after 30 frames we will have atleast 400 500 known distinct landmarks and matching all landmarks to the next frame is unnecessary computation load. We choose to opt distributed landmarks selection with a known landmark available between 4-5 frames rather than tracking a single landmark continously over 10 frames.

In this approach, the model is divided into 12 zones (-5,...,0,...,6) of 30° each and we limit the search for known landmarks through zonal selection. During the first frame, the landmarks are initialised and for each landmark, the zones are recorded based on the azimuth angle of the landmark position in the asteroid coordinate frame. In the subsequent frames, the search for the known landmarks is restricted to three zones with the maximum number of landmarks. With this generated list based on zones, the search for known landmarks is

efficiently carried out in each key frame. Later the new measurement factors are created for both the new and known landmarks, and added to the graph. This method is proven to be effective as it distributes the landmark visibility over 3∼4 keyframes for a large number of landmarks rather than one particular landmark appearing in 8∼10 keyframes. This uniform distribution of landmarks over multiple frames brings efficiency to the estimation process and reduces both state error and landmark position error. This zonal search will work until the estimation algorithm is saturated, i.e. all landmarks visited or all zones of asteroid visited atleast once. Once the model is saturated, we can switch to the search method prioritising the spacecraft's zonal position in each frame rather than the number of landmarks.

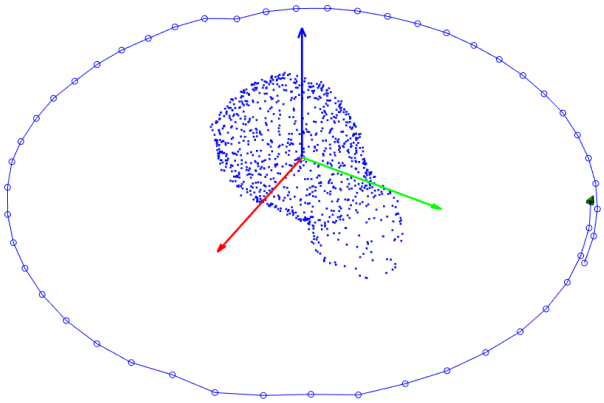

Fig. 4: Reconstructed shape of the landmark and the orbit of the spacecraft

Results

The reconstructed shape of the asteroid with the landmark features is shown in Fig.4. The reconstructed 3D shape of the asteroid includes nearly $\sim$ 98% of the original simulated landmarks, except for the landmarks near both poles. The mean error in landmark positions measured at each simulated key-frame is shown in Fig.5 and the error varies within a $\pm$10m range and similar to expected limits for future missions. The region around 8hr in Fig.5 coincides with the small bump at the spacecraft trajectory (at top of the Fig.4) and this corresponds to the correction of the spacecraft state after the landmarks are revisited for the second time. In the Fig.6, where there is a convergence around $\sim$ 8hr and the error tends to reduce after revisiting the landmarks. During the final stages of the simulation the error in estimated spacecrafts position is within $\pm$5m. This suggests that landmark selection plays a key role and the revisiting the landmarks in multiple frames helps create better estimates and solution convergence.

Conclusion

This paper presented a graph SLAM based approach to reconstruct the 3D shape of an asteroid using the distinct landmarks visible on the surface of the asteroid. The mean error of the reconstructed landmark positions are well within the range of $\pm$5m at the end of the simulation. The spacecraft's state is also estimated over the entire orbit trajectory and

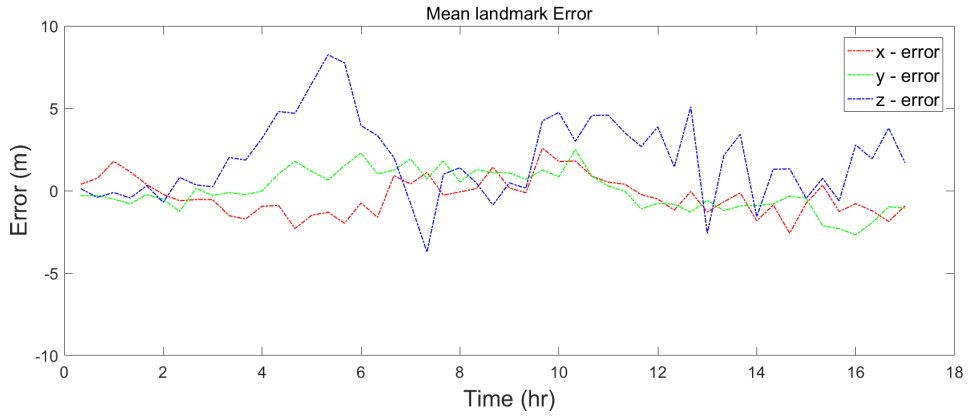

Fig. 5: Mean error in calculated landmark position during each key-frame

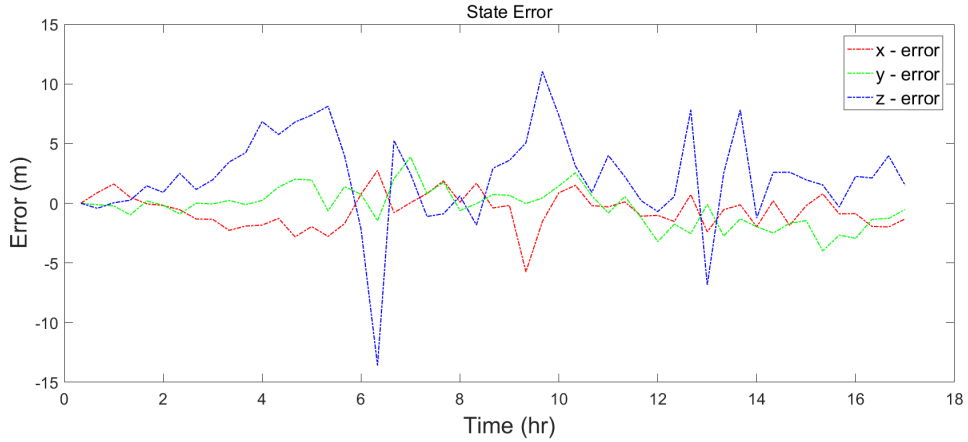

Fig. 6: Mean error in estimated spacecraft position during each key-frame

optimised using the least squares solution. The selection of landmarks and search for the previously visited landmarks are optimised and thus play a vital role in achieving the expected accuracy. Future scope of this work includes developing the framework to accomodate the other unknown parameters and estimate them in the process. Since we used the synthetic landmarks rather than the original images, the data association on the SLAM front end is expected to produce false data association and this might affect the overall performance of the system. Also, another area of possible future work is to examine the importance of landmark weightage based on their distribution in the image and their influence in the state estimate. Possible scope for future work includes improvements in false data associations, landmark feature selection methods, more realistic orbits and intial conditions, improved computational efficiency either to accommodate more landmarks or to implement the framework in realtime on-board the spacecraft.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[2] M. Montemerlo and S. Thrun, "FastSLAM 2.0," in *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 63–90.

[3] 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, no. 7-8, pp. 693–716, 2004.

[4] G Grisetti, R Kummerle, C Stachniss, and W Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

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

[6] S. Augenstein and S. M. Rock, "Simultaneous estimation of target pose and 3-d shape using the fastslam algorithm," in *Proc. AIAA Guidance, Navigation, and Control Conference (GNC)*, 2009.

[7] ——, "Improved frame-to-frame pose tracking during vision-only slam/sfm with a tumbling target," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 3131–3138.

[8] C. Cocaud and T. Kubota, "Probabilistic motion estimation for near real-time navigation and landing on small celestial bodies," *Journal of Aerospace Engineering*, vol. 4, no. 3, p. 2, 2012.

[9] ——, "Autonomous navigation near asteroids based on visual slam," in *Proceedings of the 23rd International Symposium on Space Flight Dynamics, Pasadena, California*, 2012.

[10] 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, pp. 897–933, 2015.

[11] C. Olson, R. P. Russell, and S. Bhaskaran, "Spin state estimation of tumbling small bodies," *The Journal of the Astronautical Sciences*, vol. 63, no. 2, pp. 124–157, 2016.

[12] N. Takeishi and T. Yairi, "Dynamic visual simultaneous localization and mapping for asteroid exploration," in *The International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2016)*, Jun. 2016.

[13] J. Sola, *Course on slam*, Available online: http://www.iri.upc.edu/people/jsola/JoanSola/objectes/toolbox/courseSLAM.pdf, 2017, (accessed on May 1, 2017).

[14] DARTS, *Hayabusa project science data archive:JAXA*, http://darts.isas.jaxa.jp/planet/project/hayabusa/data.html, [Online; accessed 10-Jan-2018], 2008.

[15] B. S. John Ivens, *ORX OCAMS Instrument Kernel*, Available online: https://naif.jpl.nasa.gov/pub/naif/ORX/kernels/ik/orx_ocams_v04.ti, June 30, 2016 (accessed on December 1, 2016).

[16] J. Sola, D. Marquez, J.-M. Codol, and T. Vidal-Calleja, *A graph-slam toolbox for MATLAB*, Available online: https://github.com/joansola/slamtb, 2009 (accessed on December 1, 2016).