SURF-Based SLAM Scheme using Octree Occupancy Grid for Autonomous Landing on Asteroids

C. Cocaud*, T. Kubota**

* Department of Electrical Engineering and Information Systems, The University of Tokyo, Japan
e-mail: cedric.cocaud@gmail.com
** Institute of Space and Astronautical Science (ISAS), JAXA,
3-1-1 Yoshinodai, Sagamihara, Kanagawa 229, Japan

Abstract

As space agencies are currently looking at Near Earth Asteroids as a next step on their exploration roadmap, high precision autonomous landing control schemes using passive sensors (i.e. cameras) will be required for upcoming missions. Attempting to address this need, the Guidance Navigation and Control (GNC) system presented here is an online visual navigation scheme that relies on only one camera and a range sensor to guide and control a spacecraft from its observation point about an asteroid to a designated landing site on its surface. This scheme combines monocular visual odometry with a Rao-Blackwellized Particle Filter-based Simultaneous Localization and Mapping (SLAM) system. The SLAM uses octree occupancy grids to store observed landmarks, mapping the topography of the asteroid while providing inertial data to the spacecraft’s position and attitude controller. The approach, descent and landing scheme used in this work ensures that the spacecraft will complete at least one rotation around the asteroid over the course of the landing phase, guaranteeing loop closure of the SLAM algorithm, justifying the extra computational cost of this high-precision landing scheme.

1 Introduction

As NASA abandons its views on a possible return to the Moon, Near Earth Asteroids (NEA) are being set forth as a potentially first exploration goal before Mars, following on JAXA’s first sample-and-return mission to the asteroids Itokawa. Indeed, NEAs represent a rich and unaltered source of information the past of our Solar system, as well as an abundant and very accessible pool of minerals and rare metals, providing both scientific and economic incentives to explore them [14]. While NASA is still planning its strategy to reach these small celestial bodies, JAXA/ISAS is already preparing for the next Hayabusa 2 mission, changing their aim from the S-type asteroid Itokawa to a potential C-type asteroid.

Developing Guidance Navigation and Control schemes for landing on NEA offers a new range of problems compared to the more widely investigated landing controller for Unmanned Air Vehicle [6]. The absence of atmosphere makes maneuvering around asteroids easier, but their week gravity field yields a very low escape velocity which reduces the speed range at which landing is possible without bouncing off the surface or damaging the lander. Craters and other similar features used for visual navigation [1][17] may be sparse or non-existent considering the topography of the small NEAs Itokawa. Few safe landing areas with good lighting conditions usually exist [16], and communication delays of 15 minutes are to be expected at about 1 AU from Earth, making teleoperation very difficult [17]. Thus, NEA exploration missions clearly require a highly autonomous high-precision landing GNC system.

The GNC system described here aims to address the need for such autonomous pinpoint landing controller. It is based on a monocular visual SLAM (Simultaneous Localization and Mapping) approach, determining the attitude and position of the spacecraft while simultaneously mapping the 3D asteroid topography using occupancy grids.

This paper is divided as follows. Section 2 provides a brief review of related works. Section 3 gives an overview of the GNC system. Section 4 and 5 goes in more depth describing the visual odometry and SLAM components of the system. Section 6 describes the advantages expected from the system’s architecture given the particularities of the application considered.

2 Review of SLAM Techniques

The use of SLAM techniques to alleviate the cumulative error over a mobile robot’s position while relying on inertial measurements has been of particular interest in the past decade. The SLAM problem revolves around two aspects: (i) the correlated uncertainty over the robot and landmarks positions detected using noisy sensors; (ii) the increasing dimensionality of the environment map as new landmarks are detected.

Most approaches proposed until now relied on Kalman Filtering (normal, Extended, or Unscented) to keep track of all detected landmark location and robot pose (i.e. attitude and position), maintaining a correlation
matrix between all the robot state parameters and all landmark positions (e.g. [2][20]).

While the stability and optimization efficiency of the approach was well proven, the computational burden for large number of landmarks limited its use to small contained environment. Moreover, it was shown in [18] that the EKF approach becomes unstable if the uncertainty over the robot controls (i.e. odometric/dead reckoning data) becomes large.

Although the problem of increasingly large feature database was partially solved by using groups of local maps, or by exploiting the sparseness of the correlation matrix when landmark positions are updated (as in [2]), the Kalman Filter still relies on the assumption that the motion and observation error distribution are Gaussian. An alternative approach based on a Monte-Carlo approximation of a Bayesian filters called the Particle Filter was thus proposed to avoid any dependence on a predefined error distribution. The basic principle of this technique is to estimate the posterior (i.e. the joint probability distribution of the robot pose and landmark location) conditioned on the sensors measurements history, the landmark association, and the robot controls which implicitly define its path. The Rao-Blackwell posterior factorization [8][4] simplifies the posterior to a tractable analytic solution. This solution was first proposed in [21] who assumed that landmark positions are independent of each other and solely conditioned on the robot path.

Particle Filters have the advantage of allowing the use of non-linear models for the measurements and the robot motion model. This is of particular interest when dealing with visual-odometry as the main source of landmark observations since feature matching and landmarks triangulation (the two basic step of visual-odometry) are highly non-linear in essence [10].

Different variant of the Rao-Blackwellized Particle Filter (RBPF) were proposed in the literature, the most tractable form relying on occupancy grids (e.g. [15]). Using active sensor – sonars – the real-time feasibility of such RBPF grid-based approach was demonstrated in [9] by mapping the geometry of a flooded naturally formed sinkhole. Real-time performance was also reported for the visual-SLAM approach in [9], although the map generated was used primarily for added precision in pose determination (not for actual environment reconstruction), and consisted in a compressed graph of features rather than a metric map resulting in a sparser description of the environment than in [9].

Most SLAM systems reported in the literature are based on active sensors such as sonar or other range finders. However, certain applications with limited power must rely on passive sensors (such as cameras) to navigate. This approach labeled “visual-SLAM” has been of significant interest recently for autonomous control of Unmanned Air Vehicles [2], and has been investigated for indoor environment mapping in several works (e.g. [22][7]). The work of [7] is of particular interest, the author demonstrating the feasibility of estimating the attitude and position of a hand-held camera in real-time while mapping the 3D world being observed. However, the computer used in this work is much more powerful than available space-proven onboard computer (1 to 2 orders of magnitude). This implies that the real-time feasibility of using SLAM for a spacecraft GNC still remains to be demonstrated.

Current work in Entry Descent and Landing at NASA-JPL [19] concluded that a SLAM-based GNC scheme for landing was unjustified due to its added complexity and the fact that loop closure (i.e. the fact that the robot would revisit previously observed landmarks at a given point in time) could not be achieved during landing on large celestial bodies such as Mars.

Such conclusion no longer holds for asteroids exploration mission based on the Approach Descent and Landing scheme elaborated for JAXA/ISAS’s Hayabusa mission. This scheme requires the spacecraft to stay at an observation position about the asteroid for several weeks, thoroughly mapping its surface using camera and range sensor [16], and then starting the landing sequence. If the SLAM is to active during the observation period, the algorithm will effectively revisit previously observed landmark, effectively closing the loop several time before landing [13].

### 3 GNC System Overview

The goal of the proposed GNC system is to reconstructs the asteroid's 3D shape while continuously providing attitude and position data to the spacecraft. This is to be done in near real-time i.e. in the order of 1 to 0.1 Hz. Although this spacecraft pose estimate may be considered too low, it is still a realistic compromise considering the low approach speed of the spacecraft in microgravity conditions, and the computational limitation of the spacecraft computer. This update frequency is comparable to Hayabusa’s autonomous optical guidance scheme described in [17].

The GNC system described in figure 1 proceeds in two steps. It first observes and triangulates features on the asteroid with the visual odometry (VO) segment, and then samples and assesses the likelihood of a series of possible spacecraft motion through the SLAM segment.

The inputs of the GNC are the frames acquired by the navigation camera at a given time interval, and the range sensor readings. The later is a key component to remove the scale ambiguity inherent to all monocular relative pose estimation algorithms. While the camera frames are vital during each cycle of the controller, the GNC system can remain operational even when range sensor data become sporadically available (bridging the gap by extrapolating the scale of visible landmarks position). The outputs of GNC are the asteroid’s
topographic map, and the spacecraft attitude and position estimates, which are evaluated by selecting the SLAM’s most probable hypothesis on the current spacecraft pose at a given time.

4 The Visual Odometry Segment

4.1 Image sampling

The first step of the algorithm is to acquire one frame from the navigation camera and to assess its information content. Typically, navigation cameras pointing at empty space will result in an image with very low information content. Image rejection or acceptance is determined by calculating the histogram of the acquired frame and calculating its noise, brightness, and corruption level. These three levels are calculated based on the similarity of the image’s histogram with predefined ones representing typical distributions of valid asteroid image, an invalid one where the asteroid is completely shadowed (resulting in an overly dark image), and a typical corrupted image respectively. These three levels are combined through a fuzzy logic module which makes a final pass/no pass decision according to its expert knowledge rule-base.

If the image is rejected, the algorithm stops and gives back control to the main task allocator of the spacecraft’s onboard computer. If the image is accepted, the algorithm continues on by extracting regions of interest for feature extraction in the image frame. In order to minimize false feature detection, large shadow areas and the deep space background are thus removed, considering only what is left as the areas of interest.

The time interval between two frames does not need to be constant. A clock will determine how much time has elapsed between them, such that kinematic parameters such as speed and acceleration can be determined according to the relative pose estimates between two consecutive pair of images.

4.2 Speeded Up Robust Feature Extraction

Once the areas of interest have been identified, Speeded Up Robust Feature (SURF) extraction is carried on the corresponding subsections of the image. This process is rather computationally intensive and requires up to 6 seconds for a 100 features on a 512x512 image on a space-harden computer with a speed in the order of 100 to 200 MHz (see section 6.2). SURF’s are invariant to illumination changes and orientation, while showing good robustness to affine and perspective changes. Based on [3], the repeatability (i.e. percentage of correctly matched features) decreases almost logarithmically with viewpoint angle while remaining unchanged for scale variations of up to 2.25. Following the extraction process, each feature has an associated description vector giving characteristics on the feature’s planar orientation, its size, and its position in image coordinates.

4.3 Feature matching

Once the features between frames at time (t) and time (t-TU) have been extracted - with TU being the time since the last update - the Euclidean distance between
two features’ description vector is calculated. The association between these two features is kept only if pairing these two features with any others results in a lower Euclidean distance, and if the distance calculated for that pair is below a predefined threshold. Features left unpaired after the process are discarded.

Figure 2. SURF feature extraction (computer graphics simulated environment)

4.4 Monte-Carlo estimation of features position and spacecraft motion

Although one of the most powerful techniques to find the features’ 3D position and the camera motion parameters (i.e. translation and rotation) between two frames is the so called Bundle Adjustment algorithm, this approach has the disadvantage of requiring significant computer resources (in terms of time and power), requiring an initial estimate, and not guaranteeing convergence if that initial estimate is too far from the optimal solution [23][12].

Due to the drastic limitations of a spacecraft onboard computer, a novel method is proposed here. This method is an alternative Monte-Carlo approach that can readily generate the input parameters required by the GNC system’s SLAM segment. This algorithm is as follows:

Given: \( N_f \) matched feature pairs (assuming \( N_f \geq 5 \)) corresponding to the measurement points in this algorithm
- \( K \), the camera calibration matrix
- \( x_i \), a normalized measurement point
- \( P = [I \mid 0] \), the camera matrix for the first (reference) frame

Find: sets of possible spacecraft motion between the two frames at time \( (t) \) and \( (t-TU) \) with the corresponding features 3D coordinates.

Algorithm 1: Monte Carlo Structure and Motion Sampling

1. Compute the total number of permutations possible and sample this population to form \( N_p \) sets of \( N_s \) feature pairs.
2. For one set \( S_i \) of the \( N_p \) sets of feature pairs:

2.1 Perform the \( N_f \)-point algorithm (\( N_f \geq 5 \); see the standard 5-points, normalized 8-points, and Least-Square algorithms discussed in [10]) to find the camera matrix \( P = [R \mid t] \) based on the relation on two points correspondence

\[
x_i \times E \times x_j \quad (1)
\]

with \( \{x_i, x_j\} \), the \( i^{th} \) normalized measurement point in frames \( (t) \) and \( (t-TU) \).

\( x_i = [u, v, 1]^T \) in camera coordinates

\( E \), the essential matrix mapping one world point in frame \( (t) \) to the same point observed under a different perspective in frame \( (t-TU) \).

\( R, t \), the rotation matrix and translation vector between the camera orientation and position at time \( (t) \) and \( (t-TU) \).

2.2 For each of the \( N_f \) matched points:

2.2.1 Set the homogeneous equation \( A_i X_i = 0 \), and solve for \( X_i \) using the SVD of \( A \) according to the relation between a 3D point and its image coordinates:

\[
x_i' \times (P' X_i) = 0
\]

\( x_i \times (P X_i) = 0 \)

with \( X_i = [X, Y, Z, 1] \) in homogeneous world coordinates

\( P = [I \mid 0] \) with \( I \), the identity matrix

\( P' = [R \mid t] \)

2.2.2 If the distance of one of the feature point is known (i.e. Range Sensor data are available), compute the scale for this feature point, and propagate it to all other measurement points.

Otherwise, propagate the last scale estimate using the features of frame \( (t-TU) \) that have a match in frame \( (t) \), and had their 3D position computed in a previous step.

2.3 Construct the \( i^{th} \) SLAM’s particle by constructing the vector:

\[
p(i) = [x, y, z, \Theta, \Phi, \phi | \Theta_1, \Theta_2, ..., \Theta_n]
\]

with \( [x, y, z, \Theta, \Phi, \phi] \), the position and attitude of the spacecraft (related to the camera coordinate system by a known rigid transform)

\( [\Theta_1, \Theta_2, ..., \Theta_n] \), the 3D position of each measurement point in the spacecraft coordinates

3. Construct \( N_p \) particles by repeating step (2.) for each of the \( N_s \) sets of feature pairs.

5 The SLAM Segment

5.1 Spacecraft model & robot pose sampling

In order to increase the robustness of spacecraft motion estimation, two processes are used to create the SLAM’s particle. The first and main one was described in section 4.4. The second one is part of the SLAM module and follows the model predictive control.
methodology. A simplified spacecraft dynamic model is used to estimate the spacecraft’s state at time \( t \) based on its previous state at time \( t-Tu \). The spacecraft state corresponds to its position, attitude, bearing, and velocity. Thrusters are accounted for using the impulse and momentum model, under the assumption of constant thrusting force and known or measurable actuation time.

Assuming a Gaussian error distribution over the spacecraft state parameters, a series of motion samples are generated using the mean state values at time \( t-Tu \) minus a random perturbation within the limits of the Gaussian distribution. The spacecraft motion samples are then converted to camera motion [R | t], and the steps (2.2) to (2.3) of the algorithm discussed in section 4.4 are performed for all matched feature points. A predetermined number \( N_p \) of particles are generated in that way, adding up to the \( N_p \) particles created by the visual odometry segment.

5.2 Importance weight calculation

Although importance sampling is an essential process that corrects the particles’ distribution in order to match the true Particle Filter’s posterior distribution at a time \( t \) [18], its main purpose here is assess the probability (or likeliness) of each particles. This is done by adapting the approach of [9] to visual odometry.

Assuming a normal noise model, and given a robot pose \( s \) (the spacecraft’s position and attitude), and a stored map \( \Theta \) of the positions of all previously observed landmarks, the probability that a new observation \( z \) is at the position estimated is:

\[
p(z_i | s, \Theta) = \frac{1}{\sqrt{2\pi\sigma_z^2}} e^{-\frac{(z_i - \hat{z})^2}{2\sigma_z^2}} \quad (3)
\]

with \( \hat{z} \), the distance to a previously observed landmark along the line of sight of the \( i \)th feature (i.e. along a ray from the estimated camera center’s position to the feature point in the camera image plane); \( z \), the distance from the camera center to the observed landmark; \( \sigma_z \), the variance over the landmark 3D position.

Using the factorization discussed in [21], the overall probability \( w \) that the \( M \) new observations of a particle \( j \) were correctly estimated given the previous observations \( \Theta \), is calculated by summing up the log form of equation (3) for all \( M \) new observations:

\[
\log w_j = M \log(2\pi\sigma_z^2) - \frac{1}{2\sigma_z^2} \sum_{i=1}^{M} (\hat{z}_i - z_i)^2 \quad (4)
\]

This probability \( w \) corresponds to the weight assigned to the particle \( j \).

5.3 Importance particle resampling

The total number of particles generated by the process described in section 4.4 and 5.1 is \( N_M = N_n + N_p \). While \( N_p \) depends on the total number of features that were successfully matched, \( N_n \) can be tuned online to ensure that a minimum number of particles are available at each iteration. Increasing the number of particles approximates with greater precision the probabilistic distribution of the spacecraft motion, and therefore increases the controller estimation accuracy. Since the likelihood of new particles can be assessed before having to store the complete map of landmarks associated with each of them, the total number of particles \( M \) generated during one iteration can be much larger than what onboard storage would permit otherwise.

Once the SLAM has estimated the likelihood of each particle, a subpopulation of has to be drawn in order to satisfy onboard storage limitations. Thus, the SLAM proceeds to a resampling phase after probability weights were assigned to each particles. Resampling is done according to a Roulette Wheel selection process. This process randomly picks particles with a probability proportional to their respective weight. In contrast with the implementation described in [6], this Roulette Wheel selection process is done with replacement, implying that the same particle can be picked multiple times. The total number of particles picked and stored between iterations is determined according to the storage cap assigned to the GNC system.

5.4 Octree occupancy grids updates

At the end of the resampling process each particle holds all past observations since the SLAM operation started. These observations are stored using an octree, i.e. a tree data-structure where leaves are cells of various sizes discretizing the 3D world space.

The algorithm implementing the discretization process is as follows [24]: the initial world space is represented as one root cell. If not already subdivided, the root cell is split into 8 cells of equal volume upon insertion of a new landmark. The child cells representing the space where the landmark is located are recursively subdivided into 8 children, until the maximum depth of the octree is reached (i.e. reaching one of the octree leaf cells). This maximum depth is defined based on the volume represented by the root cell, and the minimum cell size required for accurate positioning (usually set equal to the mobile robot’s size modified by some safety factor [6]).

Each cell is assigned an occupancy probability representing the uncertainty that a landmark resides within the space it represents. This probability is initialized to 0.5 to represent the fact that no information exist to prove or disprove the presence of a landmark here.

Building upon the author’s previous work [6][5] and the approach discussed in [9], a cell’s occupancy is stored using the corresponding log-odd value defined in equation (5), and is updated using algorithm detailed below.

\[
LO(\theta) = \log \left( \frac{p(\theta)}{1-p(\theta)} \right) \quad (5)
\]
Given: \( z \), a new measurement i.e. observed landmark position \( \theta \), the cells prior occupancy probability

Find: \( \theta' \) the updated cell prior.

**Algorithm 2: Octree Occupancy Cell Update**

1. Set the current cell equal to the root cell
2. Find recursively the current cell’s child representing the volume enclosing the landmark \( z \) until a leaf cell or a grey cell is found.
3. If the current cell found is a grey cell (i.e. a cell that has no children but is has a depth < max depth), expand it by creating 8 children and initializing their occupancy probability \( \theta \) to 0.5
4. Repeat step 2 and 3 until the current cell is a leaf cell.
5. Update the occupancy probability of the current cell according to:

\[
LO(\theta) = \log \left( \frac{p(\theta|z)}{1-p(\theta|z)} \right) + \log \left( \frac{p(\theta)}{1-p(\theta)} \right)
\]  

6. If \( LO(\theta) \) reaches an upper certainty threshold (indicating that cell is occupied by a landmark with high certainty), the neighboring cells are checked and if all have a log-odd value above that threshold, all cells are merged to a single cell with log-odd value initialized to the threshold value.
7. If \( LO(\theta) \) reaches a lower threshold (indicating that the cell is devoid of landmarks), merge all neighboring cells if they all have a log-odd below the given threshold.
8. Find all cells between the landmark \( z \) cell and the cell corresponding to the spacecraft location and repeat steps 5 to 7 for each of them.

The equation (8) is based on the false but necessary assumption that cells are independent of one another, i.e. assuming otherwise makes the problem intractable.

Since many particles will have entire portions of their octree occupancy grid identical to one another, a unique copy of identical sub-trees is stored in memory. This tree-sharing scheme based upon [9] should significantly reduce memory usage. This is vital considering the scarce resources available on spacecraft onboard computers.

6 Discussion

6.1 Advantages of the proposed system

The algorithm proposed in section 4.4 avoids complex and time consuming non-linear optimization algorithms while readily finding a whole series of probable motions between two frames taken at a known time intervals. Using a Particle Filter instead of a Kalman filter approach enables the GNC to uses the true error distribution of the computer vision’s motion and structure estimation problem (i.e. feature 3D localization and camera pose determination), rather than assuming artificial Gaussian distributions on the problem’s variables.

The main advantage of the proposed algorithm is that it can be performed in minimal constant time, mostly taken by performing at most, two full matrix singular value decompositions (SVD), and one partial SVD per particle, corresponding to an approximate 4515 flops (based on speed calculation from [10]). While it is well known that linear algorithms generally give poorer estimates than equivalent non-linear optimization techniques, the proposed algorithm should prove more robust than conventional linear approaches. The key concept that allows us to rely on this linear algorithm while maintaining high localization accuracy in both features 3D position and spacecraft pose, is the assumption that on a sufficiently large population of feature sets, at least one set will hold only inliers with image coordinates error small enough to be corrected by the Sampson error correction usually integrated with the linear algorithm. Under this assumption, at least one particle will accurately describe the motion of the spacecraft, leading to a much higher probability of being kept and reused at the next SLAM iterations. These highly accurate estimates should eventually dominate the particle population while the SLAM closes the loop over the course of the asteroid mapping process.

6.2 Processing speed considerations

The major difficulties that makes the application of visual SLAM to spacecraft navigation unique when considering past researches on mobile vehicle such as unmanned air vehicles or autonomous underwater vehicles, are the relative poverty of the terrain from which features (and in turn landmarks) have to extracted in order to triangulate the spacecraft position (e.g. when no craters exist), and the extreme computational limitations under which the GNC algorithm has to operate in real-time.

Figure 3 and 4 shows the result of a simulation where the camera attached to the spacecraft is orbiting with respect to a fixed asteroid reference frame. The orbiting speed is double after each asteroid revolution, doubling the angular shift of the camera between two consecutive frames at time \( t = 4140 \) s, \( t = 6288 \) s and \( t = 7350 \) s, effectively. All process speed are scaled to a reference CPU speed of 200 MHz. Image frames generated by the navigation cameras are 512 x 512 pixels.

Figure 4 clearly shows the limitations on the spacecraft speed when using the relative pose estimation algorithms based on feature matching. As the orbiting speed of the camera increases, the number of matched features between two consecutive frames decreases to eventually approach zero features matches (in which case the spacecraft could no longer triangulate its attitude and
position with respect to visible landmarks). While the approach and landing speed during Hayabusa mission was much slower than the one emulated in the simulation results shown here, it still gives an upper limit on the relative speed of the camera with respect to visible features (approximately equal to 7 deg/s here).

Figure 3. Image screening and SURF feature extraction time (512x512 images at 200MHz)

Figure 4. SURF feature extraction and matching vs. camera rotation angle (512x512 images at 200MHz)

The SURF extraction time (figure 3) show that for a GNC cycle time limit of 10 seconds (recalling spacecraft pose update frequency limit discussed in section 3), more than half of the time may be spent only on the extraction of feature points (while the image screening process discussed in section 4.1 takes less than 0.15 s). Given the maximum time left over one GNC cycle, it is reasonable to assume that the only two approaches that remain feasible are: performing a computationally simple algorithm over many different sets of feature points; or performing a complex optimization algorithm over one set or subset of features. Considering that a relatively large number of different feature sets will have to be formed and processed to produce a statistically representative population of camera motions between two frames, the only option remaining is to use the most computationally non-intensive camera motion estimation algorithm over the greatest possible number of feature subsets.

The possible candidates are the 5-points, 7-points or normalized 8-points algorithms [10]. Although the first two of these algorithms generates non-unique solutions, the proposed system’s architecture enables the GNC to treat these algorithms equally without additional post-processing to identify the correct solution from the erroneous ones: Each good or erroneous motion estimate is used to create a new particle. Particles based on erroneous solutions will necessarily have poor agreement with previous observations, yielding to a low importance weight, which will eventually force the SLAM to prune it out of the particle population.

7 Conclusions

The on-line Guidance Navigation and Control system outlined in this paper attempts to address the need for autonomous pinpoint landing controller for the exploration of asteroids or other celestial bodies under micro-gravity conditions. It is based on a monocular visual SLAM approach, determining the attitude and position of the spacecraft while simultaneously mapping the 3D asteroid topography using occupancy grids.

The proposed GNC system uses a Monte-Carlo linear relative pose estimation algorithm as the basis of its visual odometry module. Instead of providing a single optimized solution as common approaches would do, this Monte-Carlo method provides an entire population of possible hypotheses on the spacecraft motion and 3D location of asteroid landmarks. The Rao-Blackwellized Particle Filter-based SLAM maintains this population of hypotheses, generating new hypotheses based on the most probable ones, and pruning out the less likely ones.

The next step will consist in assessing the relative performance of the proposed system with the different linear pose estimation algorithms available. The system performance will then be compare with respect to high-precision non-linear optimization algorithms such as Bundle Adjustment, while considering the enormous speed gain which is paramount if this real-time GNC is to operate on a typical space-harden onboard computer in the order of a few 100’s MHz.
References


