# AUTONOMOUS NAVIGATION NEAR ASTEROID BASED ON VISUAL SLAM

**Cedric Cocaud** [(1)]**, Takashi Kubota** [(2)]

[(1)] *Jet Propulsion Laboratory (JPL-NASA), 4800 Oak Grove Drive, Pasadena, California 91109, USA, cedric.cocaud@gmail.com*

[(2)] *Institute of Space and Astronautical Science (ISAS-JAXA), ISAS campus 3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, Japan, 81-42-759-8311, kubota@isas.jaxa.jp*

***Abstract:*** *The means to detect an ever increasing number of small Near Earth Asteroids and other small celestial bodies (SCB) in the past decades has renewed the interest of a great number of space agencies across the world, their nature having the potential to bring a new understanding on the formation of our Solar System. The constraints associated with small celestial body missions require high precision navigation systems which cannot solely rely upon conventional inertial sensing techniques due to the lack of a gravity field of SCB's. Thus, visual navigation techniques become a valuable alternative despite their heavy computational burden. Combining visual linear pose estimation techniques with inertial motion estimates, this work presents a Rao-Blackwellized Particle Filter Simultaneous Localization and Mapping (RBPF-SLAM) algorithm that aims at estimating the position and attitude - or pose - of a spacecraft maneuvering near a SCB. The RBPF SLAM uses visual signatures to recognize landmarks on the surface and triangulate its position with respect to it. The performance and limitations of this approach are presented through a sensitivity analysis.*

***Keywords:*** *SLAM, visual navigation, pose estimation, small celestial bodies*

## 1. Introduction

Small celestial bodies (SCB) such as asteroids and comets have started to become the center of attention of near-term space exploration programs amongst the space agencies across the world. While JPL-NASA has already completed several missions targeting SCB's with NEAR-Shoemaker, Stardust, or Deep Impact, JAXA's Hayabusa mission was a first in landing on an asteroid, recovering samples, and returning to Earth to deliver them. Since the end of this successful mission to the asteroid Itokawa, ISAS-JAXA has been preparing the follow-up Hayabusa II mission, and is also actively studying new visual navigation schemes that will enhance the autonomy and accuracy for a potential Hayabusa Mark II mission.

The constraints associated with the navigation near SCB is very different than for large planetary bodies. The very rugged surface of SCB's makes safe landing sites sparse and narrow, requiring precision down to 10's of meters. Also the very week gravitational pull imposes strict constraints on the narrow landing speed range required to avoid either damaging the spacecraft or making it bounce back from the surface to outer space [1][2]. Conventional techniques used mainly for landing on large planetary bodies typically results in very large landing error ellipses in the order of 100's meters, making them inadequate for SCB missions [3]. One of the reasons is that the SCB's gravity is too weak to interact with inertial sensors, allowing motion to be detected only with respect to the spacecraft's own reference frame.

Thanks to the increasing computational capabilities of new computers, visual navigation has become a promising alternative, coping with the inertial sensors shortcomings by detecting visual landmarks on the SCB surface and determining the spacecraft's position and orientation according to them, i.e. its *pose*. JPL-NASA has undertaken several researches in the past decade, starting with the DIMES algorithm which combines measurements from a descent camera, a radar altimeter, and an inertial measurement unit (IMU) to measure the spacecraft altitude and attitude [4][5]. This combination of navigation cameras and IMU was then developed further by establishing catalogs of recognizable crater landmarks with 3D coordinates known *a priori* [6], and using them to reconstruct the spacecraft pose [7][8]. Similar schemes using Scale Invariant Feature Transform (SIFT) were also proposed [9], as well as various implementations using Kalman Filtering [10], particle filters, and Unscented Transforms [7].

Although significant work has been done on navigation involving fixed stereo-camera systems, the maximum baseline between two cameras fixed on the spacecraft is smaller than the depth of the features being observed by several order of magnitude during the approach, descent and landing (ADL) phase. Considering the added complexity of a wide baseline stereo-camera system that would requires two spacecrafts or alternatively, a mothership and an orbiter, the most practical approach remains the monocular structure-from-motion techniques [4], [5], [6], [8], [11].

This research builds upon the monocular structure-from-motion techniques by proposing a single-camera-based Simultaneous Localization and Mapping (SLAM) scheme in the context of a Hayabusa-type mission. This paper follows up on the work presented by the authors on using linear visual pose estimation algorithms based on the triangulation of SURF visual descriptors as interest points, for the generation of particles for a Rao-Blackwellized Particle Filter (RBPF) SLAM [12][13].

The SLAM proposed in this work extends the scheme presented previously by introducing a data-fusion method which combines IMU-based particles and vision-based particles, and uses a screening method which discards outlying vision-based particles by verifying that their motion estimate is within the maximum error range of the IMU motion estimate. The proposed SLAM approach assumes no *a priori* knowledge on the SCB topography or the existence of particular surface feature with known geometric properties such as craters.

The proposed SLAM uses the novel visual landmark mapping technique discussed in [14]. It maps landmarks spatially using an octree database designed for very large environments, and sorts landmarks by their visual descriptor -or signature- using a modified binary search tree dynamically linked to their position within the octree. This scheme facilitates and robustify the data association step of the SLAM. It also offers a solution to the inherent problem of false associations of visual landmarks when new landmark observations are matched with previously visited ones solely based on the overlap of their estimated 3D position (see discussion of this problem in [15][16]).

This paper presents an overview of the RBPF-SLAM in section 1; the techniques used to generate motion estimates to propagate particles from one time step to another in section 2; the approach to calculate the particles' likelihood and probability to be sampled for the next iteration

in section 3; the visual landmark database in section 4; and the results and discussion in section 5.

## 1. Overview of the Proposed Monocular RBPF-SLAM algorithm

The algorithm of the Rao-Blackwellized Particle Filter SLAM developed for this work is summarized in Figure 1. It proceeds as follows:

I. *Inputs*:
   1) The IMU motion estimate between the current and previous time steps;
   2) The images of a single navigation camera;
   3) The range sensor reading paired with each camera image, and the known fixed transform between the range sensor frame and the camera image frame.

II. *Relative motion estimates*:
   1) The IMU readily provides the relative motion estimate which describes the spacecraft position and attitude changes with respect to its pose at the previous time step. Given an approximate Gaussian distribution of its mean motion estimate, and a standard deviation modeled according to the upper range of the sensor's rated error, a sample population of motion estimates can be generated within this maximum error range, constituting the first part of the relative motion estimate population.

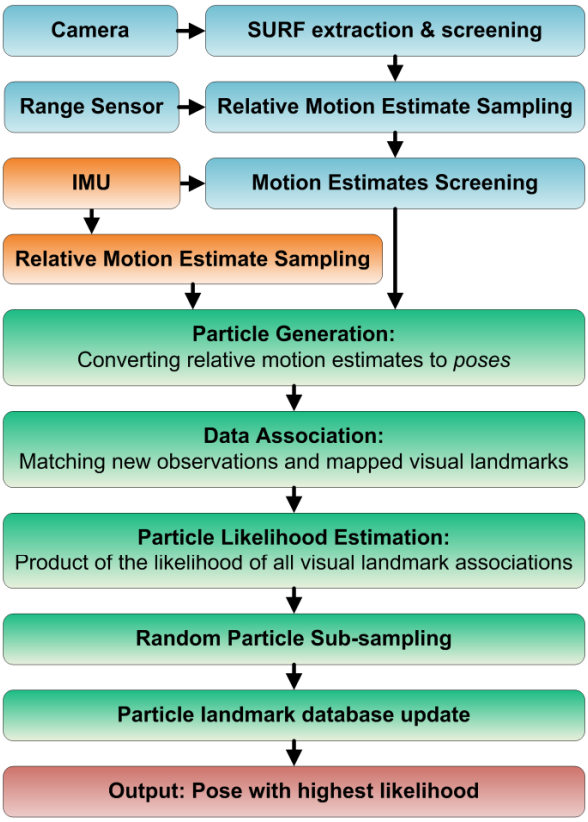


**Figure 1. Monocular RBPF-SLAM**

2) Two subsequent camera images focusing on the same area of the target SCB can be paired and used with linear pose estimation algorithms to provide relative motion estimates after extracting and matching visual features from them. Four of these linear algorithms have been investigated in [11][13], namely the 5-point, 7-point, 8-point, and 8-point RANSAC algorithms. These algorithms give the change in position and attitude of the spacecraft relative to its pose at the previous time step. A sample population of relative motion estimates can be generated by randomly sampling groups of visual features, and using one of these algorithms on each one of these groups. Once scaled using the range sensor information - see section 2.2 for details - the visual-based relative motion estimates provide the same information as the IMU-based motion estimates, and can thus be fused in the same sample population.

III. *Particle Generation*:

1) A sub-sample of the particles of the previous iteration are selected for each relative motion estimate. Each of these particle provides one pose estimation and the associated landmark distribution for the previous time step.
2) The pose of a particle can then be updated by adding to it the relative motion estimate calculated at step II, effectively converting that relative motion estimate into an updated pose expressed with respect to the SLAM's global reference frame.
3) Each particle belonging to the subsample selected for a relative motion estimate lead to the creation of a new particle belonging to the current iteration's population. The particle population of the previous time step is ultimately deleted once the new particle population has been generated and updated.

IV. *Data Association*:

1) Each new particle is assigned all the new observations corresponding to the matched visual features between the current and the previous camera image.
2) A level of similarity between every observations of each particle and all previously mapped landmark is calculated. If this level of similarity is greater than a given threshold, the observation is associated - or matched - to the corresponding visual landmark. The level of similarity between an observation and a visual landmark is calculated based on the Euclidean difference between their Speeded Up Robust Feature descriptor as discussed in [17][18].

V. *Particle Likelihood and Resampling*:

1) Each matched observation of a particle is evaluated according to the likelihood that the statistical distribution of the associated landmark position is equal to the observation's position distribution, i.e. its mean and standard deviation according to a standard Gaussian model.
2) The likelihood of the particle is given as the product of the likelihood of the matched observation and their associated landmark.
3) All newly generated particles are then assigned a probability proportional to their likelihood with respect to the sum of the likelihoods of the rest of the population.
4) The new population is then scaled down to a tractable number by performing a random pick of the particles with no replacement. Each particle has a chance of being selected equal to the probability established in step V.3).

*Visual Landmark Database Update*:

5) The visual landmarks associated with the observations are updated based on the position distribution of the latter.
6) Each new particle inherits the visual landmark database of its predecessor, and thus, each remaining particle from the sub-sampling of step 4) update its own visual landmark database.

VI. *Controller update*: the spacecraft's navigation software updates its latest pose estimate with the pose of the most likely particle for the current time step. A low-pass filter is used to smooth out the changes of the pose estimate over time, providing a more stable output for the spacecraft controller.

## 2. Relative Motion estimates and Particle Generation

### 2.1. IMU Motion estimates

The IMU is used for two purposes: 1) generating relative motion estimates; 2) providing a bracket of realistic motion estimates based on the mean motion reported by the unit and its know maximum error percentage. This bracket is used to screen out all outlying motion estimates obtained from the linear visual pose estimation algorithms.

The model of the IMU is:

$$\mathbf{X}_{(t)} = \mathbf{X}_{(t-1)} + \Delta\mathbf{X}_{true} + \mathbf{\Phi}\,(0, \boldsymbol{\sigma}_x) \quad (1)$$

$$\boldsymbol{\alpha}_{(t)} = \boldsymbol{\alpha}_{(t-1)} + \Delta\boldsymbol{\alpha}_{true} + \mathbf{\Phi}\,(0, \boldsymbol{\sigma}_\alpha) \quad (2)$$

with $\mathbf{X}_{(t)}$, $\boldsymbol{\alpha}_{(t)}$, the 3D position vector and attitude vector at time *t* of the spacecraft in the SLAM's global reference frame

$\Delta\mathbf{X}$, $\Delta\boldsymbol{\alpha}$, the true displacement and angular shift of the spacecraft at time *t* with respect to the spacecraft's reference frame at time (*t*-1)

$\boldsymbol{\sigma}_x$, $\boldsymbol{\sigma}_\alpha$, the white Gaussian noise associated with the IMU measurement

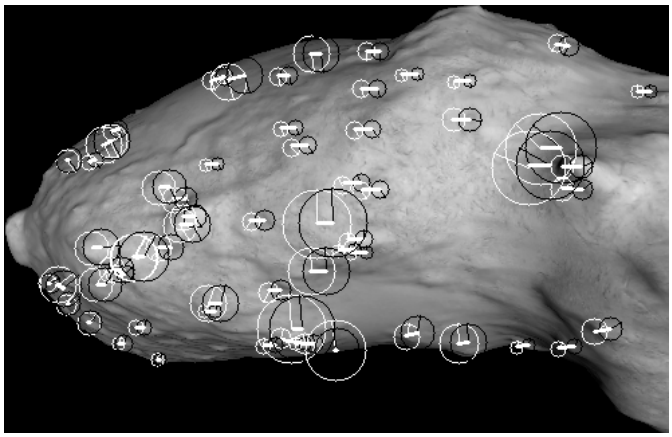

**Figure 2. Matched SURF features in a pair of subsequent images of the asteroid Itokawa**

The bracket outside of which visual pose estimates are screened out is defined as:

$$\mathbf{X}_{(t)} \in \{\mathbf{X}'_{(t-1)} - 3\boldsymbol{\sigma}_x,\ \mathbf{X}'_{(t-1)} - 3\boldsymbol{\sigma}_x\} \quad (3)$$

$$\boldsymbol{\alpha}_{(t)} \in \{\boldsymbol{\alpha}'_{(t-1)} - 3\boldsymbol{\sigma}_\alpha,\ \boldsymbol{\alpha}'_{(t-1)} - 3\boldsymbol{\sigma}_\alpha\} \quad (4)$$

### 2.2. Visual Motion Estimates

Relative visual motion estimates are calculated based on the 5-point algorithm by Nistér [19], the 7-point algorithm by Hartley and the 8-point algorithms by Longuet-Higgins [20]. The approach is to generate in a first step the population of visual features that can be matched between the current image frame and the one generated at the previous time step - see Figure 2. The second step is to randomly sample sub sets of 5, 7 and/or 8 matched visual features. The third step consists of the following algorithm described in [13] which will be briefly summarized here for the sake of completeness:

1. Solve the linear set of equations linking the normalized image points $\boldsymbol{x}$ of the image from the previous time step to the normalized image points $\boldsymbol{x}'$ of the image of the current time step, and decompose the essential matrix E into its translation and rotation components $T$ and $R$ respectively:

$$\boldsymbol{x}' E_{[3x3]} \boldsymbol{x} = 0 \quad (5)$$

$$E_{[3x3]} = T_s^u R \quad (6)$$

   with $T_s^u$, the skew symmetric matrix of the unscaled translation vector $T^u$

2. Triangulate the pairs ($\boldsymbol{x_i}$, $\boldsymbol{x_i}'$) of image points to obtain the depth $x_{i,z}$ using the possible solutions $T^u$, $R$ resulting from the decomposition of the Essential matrix, and keep only the $R$, $T^u$ solution giving a positive depth:

$$x_{i,Z} = \frac{\|x'\|^2 \left(-R\,x \cdot T\right) - \left(-R\,x \cdot x'\right)\left(x' \cdot T\right)}{\|-R\,x\|^2 \|x'\|^2 - \left(-R\,x \cdot x'\right)^2} \quad (7)$$

3. Find the scaling factor $s$ that convert the unscaled translation vector $T^u$ into the scaled vector $T = s\,T^u$, by measuring with the range sensor the distance $D$ of the reference image point $x_{R/F1}$ corresponding to the 3D point $X_{R/F1}$, and assuming that the distance to the reference landmark $\boldsymbol{\theta}_i$ corresponding to the image points ($\boldsymbol{x_i}$, $\boldsymbol{x_i}'$) is the same as $X_{R/F1}$, provided that $\boldsymbol{x_i}$ or $\boldsymbol{x_i}'$ is within the vicinity of $x_{R/F1}$ in the image plane defined by the axes $\boldsymbol{x_{Fj}}$, $\boldsymbol{y_{Fj}}$ - see Figure 3.

$$X_{R/F1} = R_{R1/F1}\left[0, 0, D\right]^T + T_{R1/F1} \quad (8)$$

$$s = X_{R/F1} / x_{i,z} \quad (9)$$

   with $R_{D/F1}$, $T_{D/F1}$, the fixed rotation and translation components of the rigid transform mapping the measured distance $D$ from the range sensor reference frame R1 to the camera reference frame $F1$ with axes $\boldsymbol{x_{C1}}$, $\boldsymbol{y_{C1}}$, $\boldsymbol{z_{C1}}$.

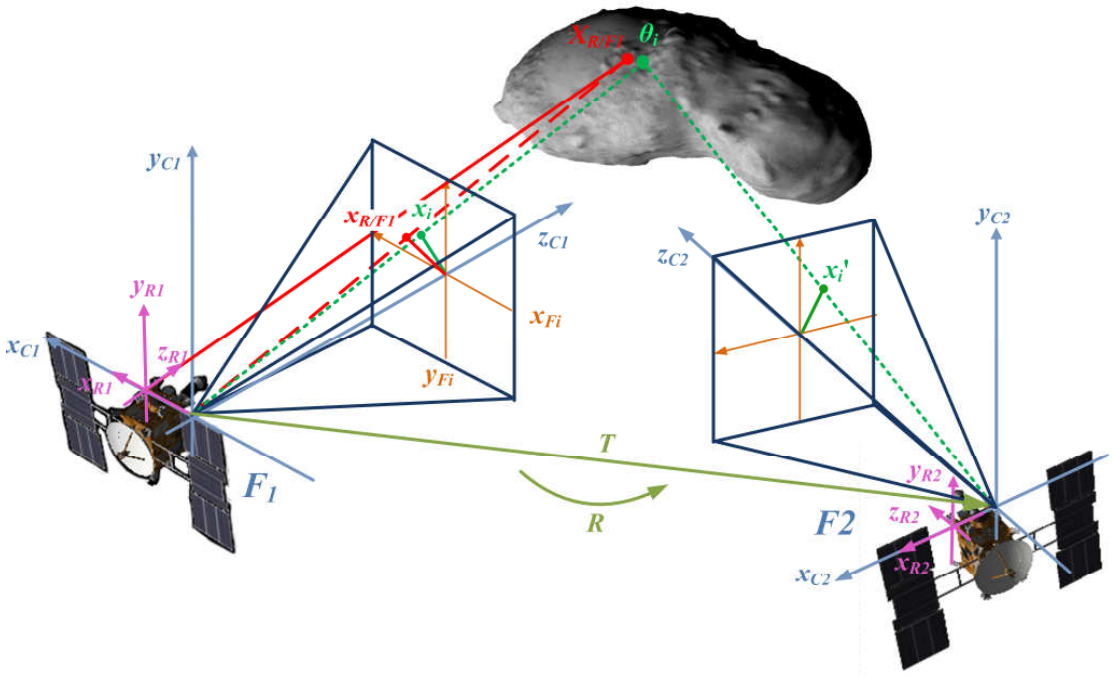


**Figure 3. Monocular motion estimate from visual features between pairs of images**

Relative motion estimates generated from visual pose estimation algorithms are typically un-scaled, implying that the rotation $R$ and the translation direction $T^u$ of the spacecraft are fully determined, but the magnitude of the translation is not. The algorithm described above uses the readings of the range sensor to measure the true distance to the visual landmark detected in the image frame, and thus resolve the scale ambiguity. When no visual features are close to the range sensor's reference points $x_{R/F1}$ in the camera frame *F1* or point $x_{R/F2}$ in *F2*, the algorithm can use the depth of previously triangulated visual landmarks that remains in the field of view of *F1* or *F2*, and reconstruct the scale of the scene using the forward propagation algorithm described in more details in [11].

### 2.3. Particle Generation From a Relative Motion Estimate

For each relative motion estimate for time step $t_k$, a sub-sample of the particles $p_{t_{k-1},G}^{[i]}$ inherited from the previous time step $t_{k-1}$ are selected and copied to constitute part of the new population of particles $p_{t_k,G}^{[i]}$. For each pair of particles $\left\{p_{t_k,G}^{[i]}, p_{t_{k-1},G}^{[i]}\right\}$, the updated pose $\left\{\bar{q}_{t_k,G}^{[i]}, \bar{X}_{t_k,G}^{[i]}\right\}$ of the spacecraft expressed as a quaternion and a 3D position vector with respect to the SLAM's global reference frame $G$ is calculated by incrementing the pose of the particle $p_{t_{k-1},G}^{[i]}$ by the relative motion estimate [R,T] converted to a quaternion-position pair $\left\{\bar{q}_{F2/F1}, T_{F2/F1}\right\}$ defining the orientation and position of reference frame *F2* with respect to *F1*:

$$\bar{q}_{t_k,G} = \bar{q}_{t_{k-1},G} \; \bar{q}_{F2/F1} \tag{10}$$

$$\begin{bmatrix} 0 \\ \bar{X}_{t_k,G} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{X}_{t_k-1,G} \end{bmatrix} + \bar{q}_{t_k-1,G} \begin{bmatrix} 0 \\ T_{F2/F1} \end{bmatrix} \left(\bar{q}_{t_k-1,G}\right)^{-1} \tag{11}$$

Once the relative motion has been propagated, the new particle has the most up-to-date pose estimate conditional to that particle's path as defined by the Rao-Blackwellization approach described in [15]. The final step of the creation of a new particle is to express all new observations $\left\{\Theta^s_{1,t_k} \ldots \Theta^s_{N_m,t_k}\right\}$, i.e. visual landmark detected in *F2* and matched in *F1*, from their local reference frame *F2* to the SLAM's global reference frame *G*:

$$\begin{bmatrix} 0 \\ \Theta^s_{t_k,G} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{X}_{t_k-1,G} \end{bmatrix} + \bar{q}_{t_k-1,G}\ \bar{q}_{F2/F1} \begin{bmatrix} 0 \\ \Theta^s_{i,t_k,F2} \end{bmatrix} \left(\bar{q}_{F2/F1}\right)^{-1} \left(\bar{q}_{t_k-1,G}\right)^{-1} \tag{12}$$

## 3. Particle Likelihood Estimation and Resampling

The likelihood *L* of a particle is calculated as the product of the likelihood of all the observations being matched with a previously mapped landmark, and the likelihood of the unmatched observation considered as new landmarks. The latter is evaluated by locating the triangulated position of the unmatched observation within the SLAM's occupancy map, and to use the probability that an obstacle exists within that area - i.e. the obstacle probability - as the observation's likelihood.

The likelihood of matched observation is calculated using the null hypothesis test that the position distribution of the landmark observed is equal to the position distribution of the landmark as mapped in the SLAM's landmark database. Figure 4 represent the distribution of the observed landmark as $(\mu_1,\sigma_1)$, and the distribution of the previously mapped landmark as $(\mu_2,\sigma_2)$. The *P*-value related to the confidence interval that the two distributions are equal gives a useful approximation of the likelihood that the two are a match.

The overall likelihood for that observation-landmark pair is given by applying the same test and calculating the P-value along the x,y,z axes defining the volumetric uncertainty distribution of the landmark:

$$L_p = P_x\, P_y\, P_z \tag{13}$$

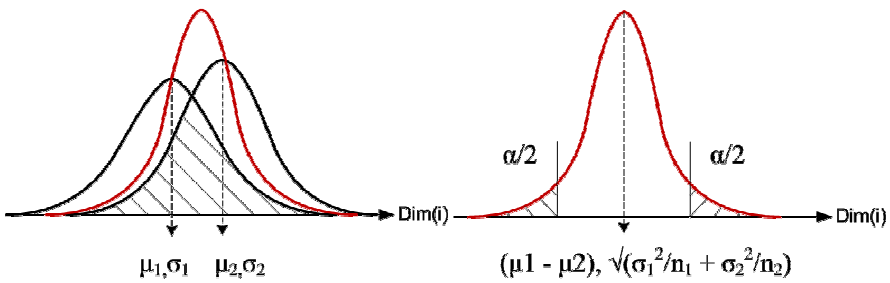


**Figure 4. Likelihood based on the P-value between two landmark position distribution**

$$P = \Phi\left(-\frac{(\mu_1 - \mu_2)}{(\sigma_1^2 + \sigma_2^2/n_2)}\right) + \Phi\left(+\frac{(\mu_1 - \mu_2)}{(\sigma_1^2 + \sigma_2^2/n_2)}\right) \quad (14)$$

with $\Phi$ the cumulative probability distribution function.

## 4. The SLAM's visual landmark database

The proposed landmark database combines an enhanced version of Farfield's memory-shared octree occupancy grid [21] with a Red-Black binary search tree [22] to catalog the visual landmark by their visual signature. The visual signature may be a SIFT [23] or SURF descriptor [17], or any other visual feature descriptor consisting of a consistently ordered array of floating point values describing various visual characteristics. Figure 5 shows a schematic representation of how the SLAM's map information is stored in memory.

The concept of the octree, is to represent a volume of space by a single cell, and to recursively subdivide this cell into 8 sub-cells of equivalent volumes until each cell has reached a homogeneous state or the maximum depth $\boldsymbol{d_M}$, i.e. the smallest cell permissible representing the highest resolution $\mathbf{r_M}$ of the map.

The nodes shown in the structure of Figure 5 are either *grey* node or *leaf* nodes with an obstacle probability varying from 0.0 (white) to 1.0 (dark red). An obstacle probability is assigned to each leaf node as a measure of the likeliness that a landmark is present within the volume represented by them.

The likelihood of the landmark defined by the spatially distributed Gaussian model described in section 3 is the basic input giving the obstacle probability for each cell in the octree. The state of an octree cell can be one of the three as described graphically in Figure 6**Figure 7**: a free cell (no landmarks); a mixed cell (low probability of having a landmark); an obstacle cell (landmark present with a high certainty).

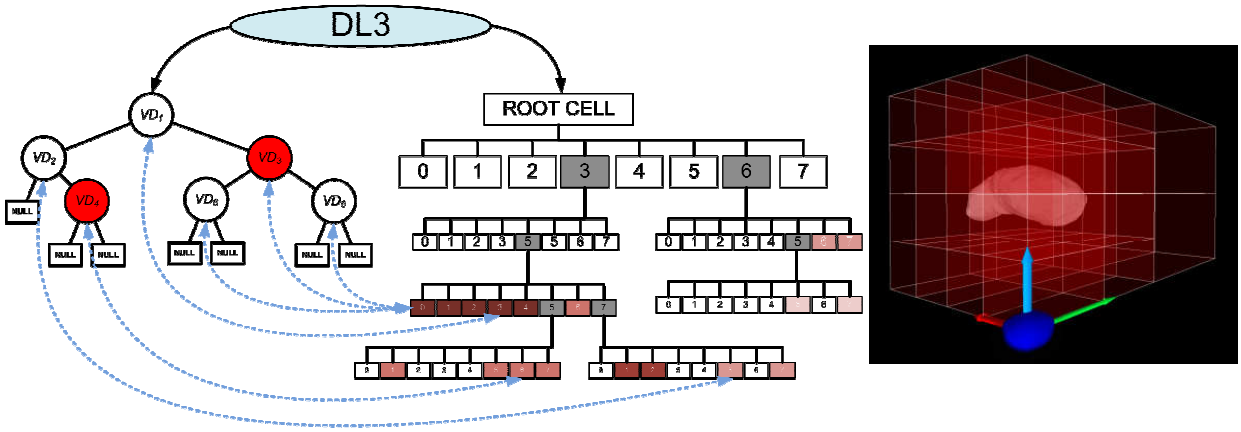


**Figure 5. The SLAM's map comprising a Red Black tree sorting visual landmark signatures and an octree occupancy tree (left) representation in memory (right) volumetric representation of the octree encapsulating Itokawa**

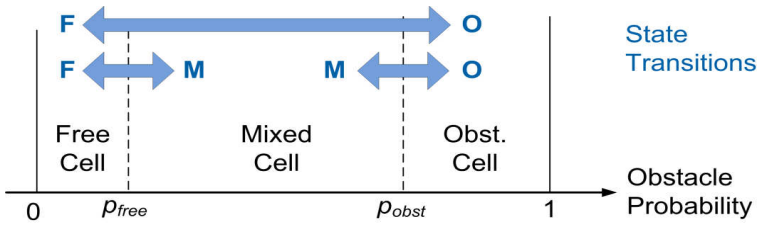


**Figure 6. Octree cell states and state transitions as a function of the obstacle probability**

Because octrees use a lot of memory, particularly if one is attached to each particle of a population of hundreds, memory optimization is mandatory. In this work, leaf cells are merge into their single parent cell when their state have all made them free or obstacle cells. Redundant information is thus deleted by having 1 cell containing the information of 8. If a leaf cell becomes mixed but has not reached the maximum octree depth, it is subdivided into 8 children in an attempt to have them all reaching a higher degree of certainty by becoming either free or obstacle cells. The threshold $\mathbf{p_{free}}$ and $\boldsymbol{p_{obst}}$ defining the state of a cell has a direct impact on the frequency at which cells are merged or subdivide based on new observations updating the cell's obstacle probability.

While octree cells contain the obstacle probability that a landmark is present nearby, the nodes of the Red Black tree (RBT) contains the information on the visual landmark's mean position, standard deviation, as well as the landmark's visual signature, i.e. its SURF or SIFT descriptor. In order to make the system more stable, the updated mechanisms for the obstacle probability of the octree cells uses a low pass filter with a time constant $\boldsymbol{\alpha_O}$. Low-pass filters with another constant $\boldsymbol{\alpha_V}$ implement the same mechanism for updating the statistical information of the RBT nodes with the information provided by new observations:

$$p_{update} = p_{k-1} + \alpha_O \left( p_k - p_{k-1} \right) \qquad \text{with } \alpha_O \in [0, 1] \tag{15}$$

## 5. Results and discussion

The overall performance of the SLAM in estimating the 6 DOF pose of the spacecraft during the approach descent and landing (ADL) phase of a SCB mission where JAXA's Hayabusa spacecraft performs close proximity operations near the Itokawa asteroid have been presented in [14]. For the sake of completeness, the result of a simulation run is presented in Figure 7.

The list of the main parameters affecting the performance of the SLAM using the proposed algorithms are listed in Table 2. The *module* column refers to the navigation sub-system responsible for each parameter, and the *effect* column gives the logical description of how the parameter affects the performance of the SLAM. Due to the complex interactions between those parameters, the search space mapping the performance of the SLAM as a function of each one of them is highly non-linear, and the problem of globally optimizing the whole set of parameters quickly become intractable. For that reason, the sensitivity study that was conducted to optimize the performance of the SLAM was restricted to a subset of parameters, and for a limited range for each one of them. The range of values tested is specified in the description column of the table.

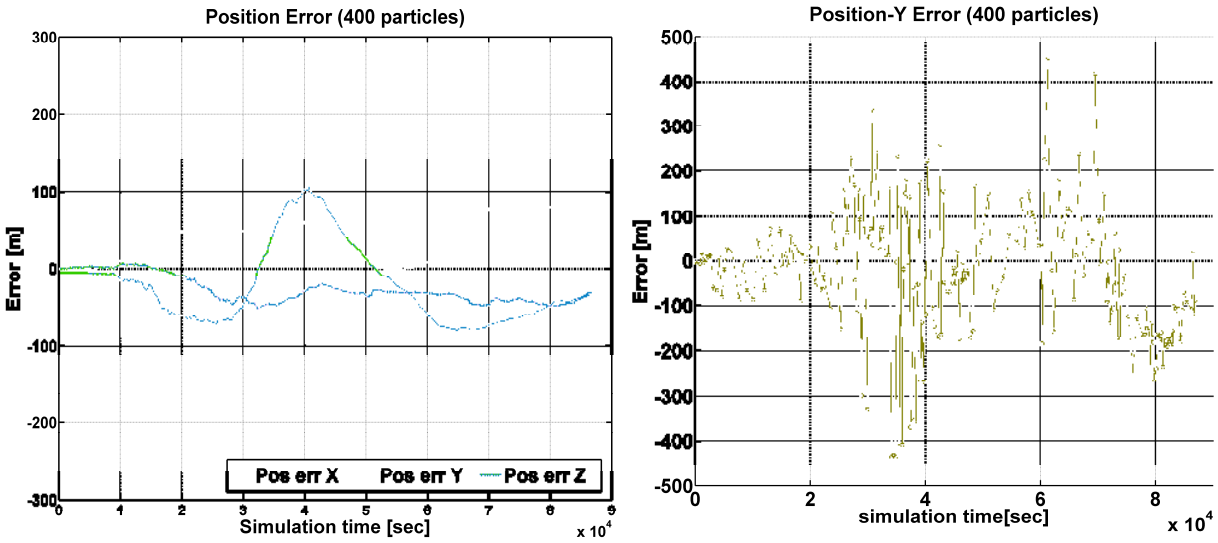


**Figure 7. Error over pose of the spacecraft during the ADL phase for 400 resampled particles: (a) Filtered x,y,z, position, (b) unfiltered y-position**

**Table 1: Simulated scenario parameters and noise sources**

| ***Parameters*** | ***Description*** |
|---|---|
| Target Body: Asteroid Itokawa | Size: 556 x 303 x 243 m<br>Angular speed: 0.000144 rad/s<br>Gravity Const: 2.51 x $10^{-9}$ $km^3/s^2$ |
| Spacecraft conditions (with respect to the asteroid reference frame) | Initial altitude: 700 m<br>Initial relative speed: 0.1 m/s<br>Final relative speed: 0.014 m/s<br>Trajectory within the equatorial plane of the asteroid |
| Navigation Camera: | Image size: 512 x 512<br>Noise: White noise of ± 10 grey intensity levels for every pixel (i.e. ± 4%) |
| Range sensor | Noise: ±10% of distance measurement |
| SLAM localization system | Simulated sampling time: 259.72 sec, i.e. (4.33 min) |
| Lighting conditions: | Reflectance Model: Lambert model<br>Configuration: Sun, asteroid, and spacecraft are aligned (same as for the Hayabusa mission) |

These results are based on simulation runs where the Hayabusa spacecraft is first facing the flank of Itokawa asteroid, performing 2 full observation orbits. The sensors noise and the other parameters of the simulation are shown in Table 1, while the effects of each parameters on the system is described qualitatively in Table 2, and quantitatively in Table 3.

**Table 2: Main parameters affecting the performance of the SLAM**

| ***Parameters*** | ***Module*** | ***Description*** | ***Effect*** |
|---|---|---|---|
| Number of resampled particles $\boldsymbol{N_R}$ | SLAM importance resampling | Response tested for 50 ~ 400 resampled particles | High number increases population diversity and convergence rate toward the true spacecraft pose, but also increases computational time and memory usage |
| Number of motion estimates $\boldsymbol{N_M}$ | SLAM motion model | Response tested for 500 ~ 2000 resampled particles | High number increases chances of having accurate estimates, but also increases computational time and decreases probability of screening out erroneous particles |
| $\boldsymbol{P_{free}}$ and $\boldsymbol{P_{obstacle}}$ | Landmark database | See description of these thresholds in section 4<br>Response tested for $\boldsymbol{P}_{free}$ = 0.05 ~ 0.2, and $\boldsymbol{P}_{obstacle}$ = 0.70 ~ 0.95 | Changes visual landmarks deletion rate from the database<br>Changes the merge and expansion rate of octree cells in the database, affecting memory usage |
| Low-pass filters constant { $\boldsymbol{\alpha_V}$ , $\boldsymbol{\alpha_O}$ } | Landmark database | Response tested for $\alpha_V$ = 0.004 ~ 0.03, and $\alpha_O$ = 0.004 ~ 0.016 | Changes rate of variability of occupancy grid, impacting likelihood estimation for new landmarks and particle resampling<br>Changes the landmarks deletion rate from the database, limiting the number of reobservable landmarks<br>Changes merge and expansion rate of octree cells, affecting memory usage |
| Maximum octree depth $\boldsymbol{d_M}$ and map resolution $\mathbf{r_M}$ | Landmark database | Response tested for a maximum octree depth $d_M$ = 4 ~ 6<br>And $r_M$ = 150 ~ 800 meters | Changes merge and expansion rate of octree cells, as well as the map size, directly affecting memory usage and computation time |

The upper half of Table 3 provides the set of values tested, and the lower half provides the mean error {e (x), e (y), e (z), e (θ)} and the standard deviation {σ (x), σ (y), σ (z), σ (θ)} over the position and the orientation of the spacecraft after the loop-closure of the SLAM. In cases where ranges are given for parameters rather than single values, the results shown are the best one obtained for that range. The intent is to prove that entire ranges of parameters are actually sub-optimal considering that the best case obtained with them gives poorer results than other sets of parameters.

As it can be seen in Table 3, the landmark database parameters are critical to reach convergence. While the octree depth and the cell size with respect to the visual landmark uncertainty were major contributors to the performance of the SLAM, these parameters proved less sensitive than the time constants $\boldsymbol{\alpha_O}$ and $\boldsymbol{\alpha_V}$ of low-pass filters of the octree cell's obstacle probability and the landmark position respectively. The study indicated that only values of less than 10% for $p_{free}$ and above 90% for $p_{obstacle}$ give acceptable results in terms of computational speed and convergence.

**Table 3: Result of sensitivity analysis**

| ***Set#:*** | **1** | **3** | **2** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|
| $\alpha_V^{-1}$ | 125 | 125 | 125 | 125 | 125 | 125 | 30 ~ 100 | 150 ~ 250 | 125 |
| $\alpha_O^{-1}$ | 200 | 200 | 200 | 200 | 225 ~ 300 | 60 ~ 150 | 200 | 200 | 200 |
| $P_{free}$ | 0.10 | 0.10 | 0.10 | 0.20 ~ 0.30 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| $P_{obstacle}$ | 0.90 | 0.90 | 0.90 | 0.70 ~ 0.95 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| $N_R$ | 400 | 400 | 50 ~ 200 | 400 | 50 ~ 400 | 50 ~ 400 | 50 ~ 400 | 50 ~ 400 | 50 ~ 400 |
| $N_M$ | 1000 | 1500 | 1e3 ~ 2e3 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| $d_M$ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4; 5 |
| $r_M$ | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 400 ~ 800; 150 ~ 200 |
| *Response** | S | S | C | C | C | C | C | U | U |
| *e(x)* [m]<br>*σ(x)* [m] | -49.1<br>62.1 | 143.1<br>77.7 | 75.0<br>126.3 | -15.8<br>286.0 | 11.5<br>250.9 | 94.5<br>264.2 | 50.4<br>137.8 | 20.2<br>305.5 | 136.6<br>552.8 |
| *e(y)* [m]<br>*σ(y)* [m] | 78.6<br>90.8 | 10.4<br>80.5 | 22.3<br>115.3 | 296.6<br>142.8 | 68.2<br>132.3 | 22.4<br>165.8 | -2.0<br>80.3 | 164.5<br>188.9 | 144.3<br>255.6 |
| *e(z)* [m]<br>*σ(z)* [m] | -41.3<br>57.02 | -144.0<br>85.3 | -112.5<br>124.5 | -269.7<br>153.6 | -227.0<br>195.0 | -4.6<br>242.7 | -100.8<br>113.4 | -281.2<br>223.7 | -117.2<br>379.0 |
| *e (θ)* [deg]<br>*σ (θ)* [deg] | 18.8<br>7.2 | 17.7<br>8.2 | 22.3<br>8.9 | 39.1<br>8.6 | 33.9<br>17.1 | 33.8<br>17.4 | 21.5<br>8.8 | 44.5<br>6.2 | 78.8<br>11.8 |

**(*)** *Response : (S) Stable convergence, (C) Critically stable (i.e. with oscillations), (U) Unstable (i.e. divergence)*

A broader range of value for $\alpha_O$ and $\alpha_V$ were found to give good results within the range 0.0033 to 0.0167, corresponding to the inverse time constants $\alpha_O^{-1}$ and $\alpha_V^{-1}$ ranging from 60 to 300 respectively. The optimal set of parameters is shown in the column for set #1.

The SLAM is particularly sensitive to the number of resampled particles, convergence occuring only in cases where that number is 400. Cases with higher number of resampled particles could not be performed due to the memory limitations imposed by the operating system under which the system was developed and tested. However, the trend outlined above clearly shows that higher number of resampled particles leads to convergence and to smaller errors over the spacecraft pose.

## 6. Conclusion

Combining visual linear pose estimation techniques with inertial motion estimates, this work presents a Rao-Blackwellized Particle Filter Simultaneous Localization and Mapping (RBPF-SLAM) algorithm that aims at estimating the position and attitude - or pose - of a spacecraft maneuvering near a small celestial body. The RBPF SLAM uses visual signatures to recognize landmarks on the surface and triangulate its position with respect to it. Although it is well know

that RBPF SLAM are very sensitive to the number of particles generated and resampled at each iteration, this research has found that the stability and precision of the SLAM was also highly dependent on the operational parameters of the map which directly influences: i) the level of discretization of the space where visual landmarks are mapped; ii) how long a visual landmark is kept in memory; iii) how often its visual signature and position distribution is updated; iv) at what rate the landmark information changes as a function of new observations errors. While the proposed scheme is computationally intensive, it has the merit of offering a new approach to visual navigation, and to identify some key parameters of the SLAM that have a direct impact on its performance.

## 7. References


[1] Maruya M. Ohyama H., Uo, M., Muranaka N., Morita H., Kubota T., Hashimotoa T., Saito J., Kawaguchi J. "Navigation Shape and Surface Topography Model of Itokawa", AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, Colorado, 2006

[2] Scheeres D., Gaskell R., Abe S., Barnouin-Jha O., Hashimoto T., Kawaguchi J., Kubota T., Saito J., Yoshikawa M., Hirata N., et al. "The actual dynamical environment about itokawa", Proceedings of the Astrodynamics Specialist Meeting and Exhibit, Keystone, Colorado, 2006

[3] Mourikis A. I., Trawny N., Roumeliotis S.I., Johnson A.E., Ansar A., Matthies L. "Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing", IEEE Transactions on Robotics, Vol 25, No 2, pp 264-280, 2009

[4] Cheng Y., Johnson A., Matthies L., "MER-DIMES : a planetary landing application of computer vision", IEEE Comp. Soc. Int. Conf. on Computer Vision and Pattern Recognition, San Diego, USA, 2005

[5] Johnson A., Willson R., Cheng Y., Coguen J., Leger C., Sanmartin M., Matthies L., "Design Through Operation of an Image-Based Velocity Estimation System for Mars Landing", International Journal of Computer Vision, Vol 74, No 3, pp 319–341, 2007

[6] Ansar A., Cheng Y., “An Analysis of Spacecraft Localization from Descent Image Data for Pinpoint Landing on Mars and Other Cratered Bodies”, Photogrammetric Engineering & Remote Sensing, Vol 71, No 10, pp 1197-1204, 2005

[7] Trawny N., Mourikis A.I., Roumeliotis S.I., Johnson A.E., Montgomery J.F., “Vision-Aided Inertial Navigation for Pin-Point Landing using Observations of Mapped Landmarks”, J. of Field Robotics Vol 24, No 5, pp 357–378, 2007

[8] Johnson A., Ansar A., Matthies L., Trawny N., Mourikis A.I., Roumeliotis S. I., "A General Approach to Terrain Relative Navigation for Planetary Landing," AIAA Infotech at Aerospace Conference, Rohnert Park, USA, 2007

[9] Ansar A., Cheng Y., “Vision Technologies for Small Body Proximity Operations”, i-SAIRAS Conference Proceedings, Sapporo, Japan, pp 402-409, 2010

[10] Flandin G., Perrimon N., Polle B., Vidal P., Philippe C., Drai R., "Vision Based Navigation for Space Exploration", ESA Workshop on GNC for Small Body Missions, ESTEC, 2009

[11] Cocaud C., Kubota T., "SLAM-Based Navigation Scheme for Pinpoint Landing on Small Celestial Body", Advanced Robotics, Vol 26, No 15, pp 1747-1770, 2012

[12] Cocaud, C. , Kubota, T. "SURF-Based SLAM Scheme using Octree Occupancy Grid for Autonomous Landing on Asteroids", Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Sapporo, Japan, pp 275-282, 2010

[13] Cocaud C., Kubota T., "Probabilistic motion estimation for near real-time navigation and landing on small celestial bodies", International Symposium on Space Flight Dynamics (22nd ISSFD), Sao Jose Dos Campos, Brazil, pp 1-13, 2011

[14] Cocaud C., Kubota T., "SLAM visual landmark 3D mapping system for autonomous navigation and landing on small celestial bodies", 35th Annual American Astronomical Society (AAS) - Guidance and Control Conference, Breckenridge, USA, 2012

[15] Montemerlo M., "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association", Ph.D. thesis, Carnegie Mellon University, USA, 2003.

[16] Thrun S., Burgard W., Fox D. "Probabilistic Robotics", MIT Press, 2005

[17] Bay H., Ess A., Tuytelaars T., Van Gool L., "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol 110, No 3, pp. 346-359, 2008

[18] Evans C., "Notes on the OpenSURF Library", online report, Retrieved from: http://www.chrisevansdev.com/computer-vision-opensurf.html, last consulted Nov 26, 2011

[19] Nistér, D., "An Efficient Solution to the Five-Point Relative Pose Problem", IEEE Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Vol 2, pp. 195-202, 2003

[20] Hartley R., Zisserman A., "Multiple View Geometry in Computer Vision Second Edition", Cambridge University Press, 2nd Edition, 2003

[21] Fairfield N., Kantor G.A., and Wettergreen D., "Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels", Journal of Field Robotics, Vol 24, No 1/2, pp 3-21, 2007

[22] Dale N., Joyce D.T., Weems C., "Object-Oriented Data Structures using Java", Jones and Bartlett Publishers, 1st Edition, 2002

[23] Lowe D.G., "Distinctive image features from scale-invariant Keypoints," International Journal of Computer Vision, Vol 60, No 2, pp. 91-110, 2004