CAMERA-AIDED INERTIAL NAVIGATION FOR PINPOINT PLANETARY LANDING ON RUGGED TERRAINS

Jeff Delaune[1], Guy Le Besnerais[2], Martial Sanfourche[2], Jean-Loup Farges[1], Clément Bourdarias[3], and Thomas Voirin[4]

[1]*ONERA-DCSD, BP74025 - 2 avenue Édouard Belin - 31055 Toulouse Cedex 4 (France),*
Email: jeff.delaune@onera.fr, jean-loup.farges@onera.fr
[2]*ONERA-DTIM, Chemin de la Hunière - 91761 Palaiseau Cedex (France),*
Email: guy.le_besnerais@onera.fr, martial.sanfourche@onera.fr
[3]*Astrium ST, 66 route de Verneuil - 78133 Les Mureaux Cedex (France), Email: clement.bourdarias@astrium.eads.net*
[4]*ESA-ESTEC, Postbus 299 - 2200 AG Noordwijk (The Netherlands), Email: thomas.voirin@esa.int*

ABSTRACT

Pinpoint landing is a needed capability in order to increase the number of possible target sites for future planetary robotic landers and maximize their mission return. This article proposes a navigation system fusing image-to-map landmark matches and inertial data through an extended Kalman filter to land close to a point selected on a planetary surface map. No assumption is made about terrain topography so that it is compatible with rugged landing sites which can appear at very low altitudes or on bumpy bodies. Two methods are proposed to identify online image feature points on the on-board map using their geometric context. They are memory-efficient and robust to illumination changes. Monte Carlo simulations results are presented using a lunar landing dynamic simulator coupled with a virtual image generator.

1. INTRODUCTION

Exploration of the solar system has relied a lot on robotic landing vehicles to reach the surface of planetary bodies and enhance our knowledge on them. Some missions target regions of scientific interest spotted from orbital data. Some others aim at reaching previously-landed assets, like astronauts operating on the surface or a rover. In addition to these mission objectives, technical constraints must also be met by the landing sites. These can concern the presence of surface hazards to avoid or specific illumination patterns for solar power generation. Areas meeting all these requirements are usually of limited extent. Thus, the more accurate the landing system, the more candidate sites become available to maximize mission return. Landing accuracy is defined as the actual touchdown distance to the surface point chosen from map data before launch. A 100-meter precision is usually agreed as the definition for pinpoint landing.

Achieving such an accuracy is a challenge for the navigation, guidance and control systems. Navigation shall estimate the state of the lander in terms of attitude, velocity and position of frame $\{b\}$ attached to the vehicle, with respect to frame $\{g\}$ tied to the planet and in which maps are referenced. These outputs are passed forward to the guidance system, which updates the target trajectory accordingly, and to the control system, which tries to achieve it. In space exploration, communication delays associated to long distance from Earth force these capabilities to be embedded.

Without external radio or satellite help, the navigation system can only rely on terrain sensors to attain the expected precision. An optical camera sensor is chosen here as it is lightweight, cheap, space-qualified, and also because only such passive devices can operate at any altitude from orbit to touchdown. It does require sunlight to provide images but this constraint fits most current mission scenarios, which target illuminated areas. New high-resolution planetary maps provided by recent orbiters have surface resolutions of a few meters only [3]. Such details allow landmarks to be identified on the maps even at very low altitudes to make the navigation system more accurate. However when close to the ground, 3D aspects of most terrains are enhanced by perspective distortions and cannot be neglected any more. This is also true for the whole descent towards more bumpy bodies, like asteroids for instance. Handling the 3D geometry of the terrain is usually challenging for cameras because an image measurement is fundamentally 2D. Active LiDAR sensors would be more adapted to 3D terrains by nature, but they cannot operate at high altitude and are harder to accommodate on a spacecraft. This work is thus focused on developing a camera-based pinpoint landing navigation system compatible with these low-altitude flight phases and rugged terrains in general, while fulfilling real-time mission hardware constraints.

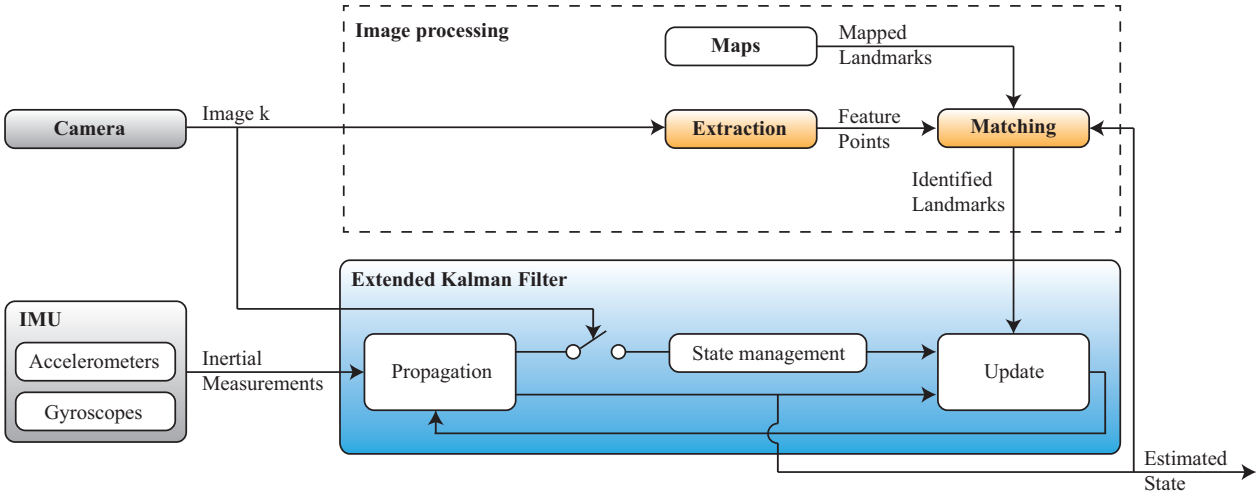


Figure 1: Visual and inertial data integration structure.

2. SYSTEM OVERVIEW

The navigation system proposed in Fig. 1 fuses inertial data from an Inertial Measurement Unit (IMU) with image data from the camera. The IMU measures non-gravitational acceleration and angular rates in an inertial frame. Time integration of these measurements allow for a high-frequency estimation compatible with control loop requirements. It also ensures that a continuous navigation solution is outputted, even if the image processing block fails to produce measurements. Descent images are processed on-board to extract and identify online features points on cartographic data and correct the inertially-propagated state estimates through a filtering architecture.

Section 3 is devoted to the process of finding reliable matches between online image features and mapped orbital landmarks. Reference [11] draws a brief state of the start of methods used for this purpose. We chose to work on methods identifying landmarks thanks geometrical description only. Unlike the use of radiometric signatures, they are more efficient in terms of on-board memory cost and robustness to illumination changes between the map and the online images. Such a method should be able to deal with 3D landmark distributions and also be robust to non-detections, i.e. a landmark is not present in the online image feature set, and to a large number of false alarms, i.e. online image features which do not correspond to any landmark. We propose a two-step process. The first step consists in searching for putative matches. It is based upon the similarity between the geometric distributions of 2D features sets. These sets are the online features set and the set of landmarks projected on the image plane using the current estimation of the camera pose. The second step is a RANSAC-based robust 3D-2D pose computation [7]. Note that we are not interested in the pose itself but in the inlier set of matches provided by RANSAC.

Data fusion is implemented through an Extended Kalman Filter (EKF), detailed in Section 4, to handle the non-linearities in the visual measurement model. The architecture is based on that proposed by [17]. Inertial navigation equations allow to propagate the attitude, velocity and position of the spacecraft by successive integrations in the predictive part of the filter. These quantities are estimated in frame $\{g\}$ using initial conditions provided by the previous flight phase. Image measurements of absolute features matched with the map are processed in the update part of the filter. Their known 3D coordinates in $\{g\}$ allow to predict their image coordinates from the a priori state estimation done in the filter and thus work out the measurement innovation for each of them. Computer processing delays associated to image processing are taken into account by adding the pose [1] of the camera frame at the time last image was taken in the state vector of the filter. Unlike loose data fusion architectures, where the vision subsystem estimates the state independently of inertial navigation, tight fusion structures like that proposed here allow to update the state vector in the filter even with very few image point measurement. They are thus then less sensitive to degradation of visual observation conditions. Estimation is also more accurate thanks to modeling the system better into the filter.

3. MAPPED LANDMARK MATCHING

In computer vision, there is usually a first step of image processing aimed at extracting features from the images, so as to reduce the number of input parameters. Here, we employ corner point features extracted by the Harris and Stephen's detector [9] both on the online and orbital images. These features capture intensity extrema computed from spatial derivatives. For planetary images, we assume a significant amount of the corners are due to terrain topography and thus can be reliably detected both in

[1] Attitude and position.

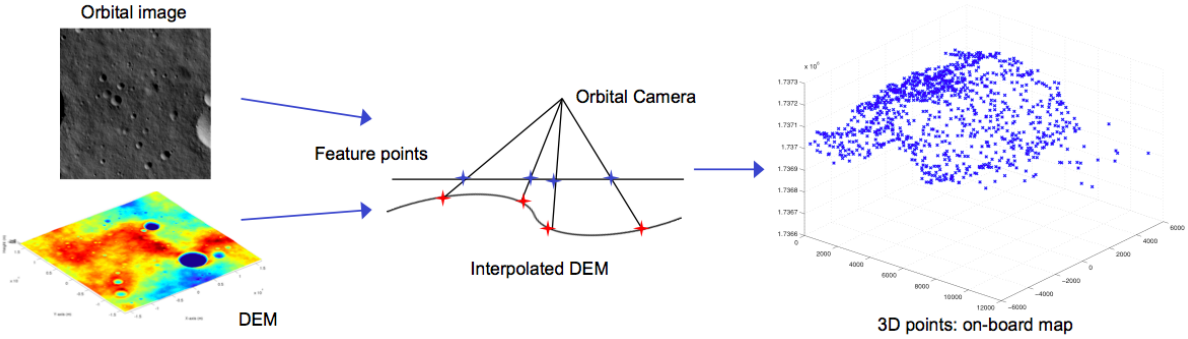


Figure 2: Orbital map generation process

a descent and orbital image of the same area. The generic nature of these image features means that they can be used on any type of planetary terrain, unlike craters which cannot be found everywhere [15].

Absolute matching consists in matching on-board landmarks extracted from orbital data acquired about the selected landing area with features extracted from online descent images. Let us emphasize that in the chosen tight integration scheme, this goal of image processing is not to estimate through camera pose computation but to provide a set of reliable matched features to be used in the measurement equations of the EKF. The absolute matching goes in four steps:

1. Online extraction of image feature points
2. Projection of the 3D model onto the expected focal plane,
3. Putative matching process,
4. RANSAC-based robust matching.

3.1. On-board and online data

The map stored on the on-board memory consists in the 3D coordinates of landmarks extracted from an orbital image. They are recovered from a Digital Elevation Model (DEM) of the area collected prior to the landing in a process illustrated on Fig. 2. Current image and laser ranging technology provide DEMs and orbital images of planetary surfaces with resolutions close to a meter [3]. Orbital landmarks are obtained by feature extraction on the orbital image of the landing zone. Knowing the position and attitude in planetary frame $\{g\}$ of the satellite which took the images, along with the calibration matrix of its camera, one can determine the 3D lines going from the optical center of the camera to each terrain point [10]. The intersections of these lines with the planetary surface rendered from the DEM points through 2D interpolation constitute the map. This 3D model aims at corresponding to potential online descent features and is the only previous knowledge of the terrain required by the proposed navigation filter.

3.2. 3D model projection

From the current camera pose estimate from the filter, and from the known camera calibration model, the embedded 3D model of the map can be projected onto the expected focal plane of the image to get a prediction of the online image feature point distribution. This eventually leads to two sets of image features: one extracted from the online image, and one predicted from the map and state estimates. An example is shown in Fig. 3.

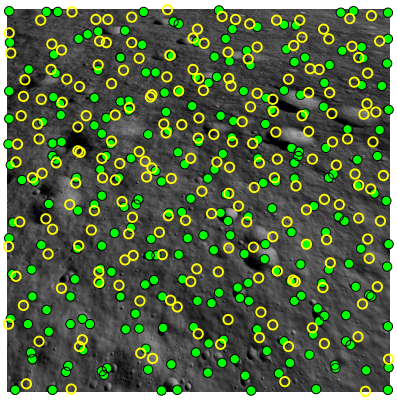


Figure 3: Extracted online image features (green dots) and projected map landmarks (yellow circles)

3.3. Putative matches

This step aims at selecting a fraction of all the possible matches between an image feature and the map in which the correct matches will be sought for. Even though several outliers couples are selected, this pre-matching step allows to select the most promising match candidates used as input in the following robust matching step. Two alternatives were tested: Shape Context description and General Hough Transform.

3.3.1. Shape Context description

Following the Shape Context description for object recognition introduced by [1] and applied to planetary descent navigation by [14], we characterize a feature by the geometric distribution of its neighbors in the image, in terms of distance and polar angle, as shown in Fig. 4. The number of neighbors in each quadrant is counted within a region comprised between a minimum radius br and maximum radius pr, and stored into an histogram.

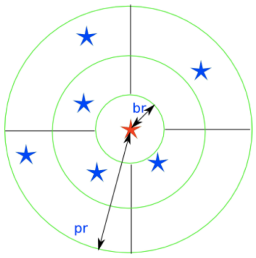


Figure 4: Geometric signature employed [14]

Signatures in the online image are compared one to one with that of the projected 3D landmarks within an uncertainty area derived from the filter states covariance. The comparing criterion is the Chi-square distance. All pairs of points between the online and prediction image for which the Chi-square is below a certain threshold are considered as potential matches.

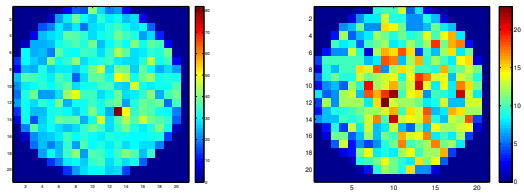
3.3.2. General Hough Transform

Here, it is assumed that the image error between the landmark image point and their estimate can be approximated by a translation in the image plane. This is not rigorously true because of perspective distortion effects in the image but should the translation be searched within a set of quantized vectors with quantization step large enough, a voting accumulation effect will happen for the correct ones. This approach is known as Generalized Hough Transform in the object recognition field [8] and makes the basis of operational real-time infrared missile guidance systems [6]

The matching process is the following:

1. Each possible match between an online feature and a projected orbital match defines a possible 2D translation.
2. Accumulate all translations:
   - whose norms are below a threshold related to the estimated camera pose covariance,
   - after quantization with a step related to the expected perspective distortion effects.
3. Select the peak of the accumulator: it yields the estimated discrete translation.
4. Shift the projected landmarks according to the estimated translation and match them with the closest descent point.
5. Accept the match if the closest point is not too far.

Fig. 5 shows the quantized grid in which translation votes are accumulated in two cases: correct accumulation on Fig. 5a and ambiguous vote on Fig. 5b. The ambiguous case happens when the camera is so close to the ground that not enough orbital landmarks appear in the field of view and thus the accumulation effect is not sufficient.



(a) Correct (b) Ambiguous

Figure 5: Correct and ambiguous results of the voting process. The ambiguous case occurs when very close to the ground. Not enough landmarks appear in the field of view and thus the accumulation effect is not sufficient.

3.4. RANSAC-based robust matching

Fischler and Bolles [7] introduced the RANdom SAmple Consensus (RANSAC) algorithm which aims at fitting a model to experimental data and provide the associated set of inliers. In our case, the model is the camera projection matrix[2], and the experimental data taken as inputs are the potential $(2D, 3D)$ potential matching pairs. The calibrated camera projection matrix can be computed in a closed form from a minimal set of 3 $(2D, 3D)$ point correspondences as shown in Fischler and Bolles [7] too. Three correspondences actually bring multiple solutions, but we select that for which camera position is the closest to the filter position estimation. The aim of this final step is to get the set of matches which correspond to camera projection matrix with the highest number of inliers. Pose computation in the RANSAC is used as a transformation model to find the inliers but the computed pose is only a byproduct which is not used in the filter.

The RANSAC-based final algorithm operates as follows from a 5xN potential match matrix made from a set S of N $(2D, 3D)$ couples:

1. A set s of 3 potential matches is randomly selected from S;

[2]Since the camera is calibrated, the knowledge of the projection matrix is equivalent to that of the attitude and position of the camera.

2. If there is any degenerated configuration[3] in the 2D points of s, then do step 1 again – this may happen because of the multiple potential matches for a same descent image;

3. Compute the possible camera projection matrices associated to s,

4. Select the matrix for which the camera position is closest to the filter estimate,

5. Determine the inliers for the projection matrix selected in step 4. Namely, inliers are potential matches for which the 2D projection of the 3D terrain point, according to the selected projection matrix, is within a certain image distance threshold of the 2D image point. The image threshold is selected as $3\sigma_{im}$, where σ_{im} is the standard deviation of the noise associated to image feature extraction.

6. If the number of inliers is greater than the previous one, then store the inlier vector.

7. Back to step 1 until the maximum set of iteration is reached.

The maximum number of iterations in step 7 is determined adaptively to ensure with a probability $p = 0.99$ that at least one of the set s is free from outliers [10]. Once the maximum number of iterations is reached, the stored inlier vector is outputted as final matches to the navigation filter. It corresponds to the camera projection matrix having the largest number of inliers. Note that through this camera matrix, part of the state is already computed in this matching step outside of the filter. Although it is not done here, this redundancy can be used for system cross-checks with the estimated state output from the filter.

3.5. Discussion

The two algorithms which have been presented in this section aim at robustly matching mapped landmarks with online images. Unlike many other landmark matching methods in the literature, no flat world assumption was made in the process. This allows such methods to operate over highly-3D terrains, as was shown by the preliminary geometrical simulation in [5]. We use DEM-based 3D landmark position as in [15], but our use of geometric signatures and RANSAC-based robust matching allows to prevent false matches more efficiently. Note that RANSAC is a proven technique in real-time visual odometry for terrestrial robotics.

[3] For instance, if the three points are aligned or two of them are identical.

4. ESTIMATION

We now go further into the EKF estimation architecture used in this approach. The reader can refer to [17] to learn more about the derivation of the filter equations.

4.1. Model

The system considered can be put into equations as:

$$\begin{aligned} \dot{\boldsymbol{q}}_g^b = \tfrac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{gb}^b)\boldsymbol{q}_g^b,\ \dot{\boldsymbol{b}}_{gyr} = \boldsymbol{n}_{bgyr}, \\ \dot{\boldsymbol{v}}_{gb}^g = \boldsymbol{a}_{gb}^g,\ \dot{\boldsymbol{b}}_{acc} = \boldsymbol{n}_{bacc},\ \dot{\boldsymbol{p}}_{gb}^g = \boldsymbol{v}_{gb}^g \end{aligned} \tag{1}$$

The vehicle state is noted $\boldsymbol{x}_V = \begin{bmatrix} \boldsymbol{q}_g^{bT} & \boldsymbol{b}_{gyr}^T & \boldsymbol{v}_{gb}^{gT} & \boldsymbol{b}_{acc}^T & \boldsymbol{p}_{gb}^{gT} \end{bmatrix}^T$. $\boldsymbol{q}_g^b$ is the quaternion describing the rotation from the world frame $\{g\}$ to the vehicle frame $\{b\}$. $\boldsymbol{v}_{gb}^g$ and $\boldsymbol{p}_{gb}^g$ are respectively the velocity and positions of $\{b\}$ with respect to $\{g\}$ projected in the axes of $\{g\}$. Lastly, $\boldsymbol{b}_{gyr}$ and $\boldsymbol{b}_{acc}$ are the gyrometer and accelerometer biases projected on the vehicle frame $\{b\}$ which is assumed to be the same as the IMU frame.

The other parameters in the dynamic model of the Eqs. 1 are: gaussian centered white noises $\boldsymbol{n}_{bgyr}$ and $\boldsymbol{n}_{bacc}$ introduced only to allow for the estimation of gyrometer and accelerometer biases, the vehicle acceleration with respect to the world frame $\boldsymbol{a}_{gb}^g$, the angular velocity vector $\boldsymbol{\omega}_{gb}^b$ projected on the vehicle frame, and the operator $\boldsymbol{\Omega}$ defined by:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}\wedge] \end{bmatrix},$$
$$[\boldsymbol{\omega}\wedge] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_y & 0 \end{bmatrix}.$$

We shall note the classic convention used here for the quaternions with the first component representing the scalar part, and the three others for the imaginary parts. The angular velocity and acceleration data are inferred from the IMU measurements $\boldsymbol{\omega}_{IMU}$ and $\boldsymbol{a}_{IMU}$ using Eqs. 2 and 3:

$$\boldsymbol{\omega}_{IMU} = \boldsymbol{\omega}_{gb}^b + \boldsymbol{C}(\boldsymbol{q}_g^b)\boldsymbol{\omega}_{ig}^g + \boldsymbol{b}_{gyr} + \boldsymbol{n}_{gyr}, \tag{2}$$

$$\begin{aligned} \boldsymbol{a}_{IMU} = \boldsymbol{C}(\boldsymbol{q}_g^b)(\boldsymbol{a}_{gb}^g - \boldsymbol{g}_b^g + 2\left[\boldsymbol{\omega}_{ig}^g\wedge\right]\boldsymbol{v}_{gb}^g \\ + \left[\boldsymbol{\omega}_{ig}^g\wedge\right]^2\boldsymbol{p}_{gb}^g) + \boldsymbol{b}_{acc} + \boldsymbol{n}_{acc}. \end{aligned} \tag{3}$$

$\boldsymbol{C}(.)$ is the coordinate change matrix associate to a quaternion, $\boldsymbol{\omega}_{ig}^g$ is the angular velocity vector of the planet frame with respect to the inertial frame, $\boldsymbol{g}_b^g$ is the local gravity vector, $\boldsymbol{n}_{gyr}$ and $\boldsymbol{n}_{acc}$ are gaussian white noises centered on IMU measurements.

4.2. State propagation

We just saw how the system can be modeled using a function f such that $\dot{\boldsymbol{x}}_V = f(\boldsymbol{x}_V, \boldsymbol{n}_{IMU})$, where $\boldsymbol{n}_{IMU} = \begin{bmatrix} \boldsymbol{n}_{gyr}^T & \boldsymbol{n}_{bgyr}^T & \boldsymbol{n}_{acc}^T & \boldsymbol{n}_{bacc}^T \end{bmatrix}^T$ is the IMU measurement noise, of covariance matrix $\boldsymbol{Q}_{IMU}$. Function f is depending in inertial measurements and thus evolves through time.

By applying the expectation for each term in the Equation system 1, one obtains the Eqs. 4 to propagate forward in time the state of the vehicle $\hat{\boldsymbol{x}}_V$:

$$\begin{gathered} \dot{\hat{\boldsymbol{q}}}_g^b = \tfrac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}})\hat{\boldsymbol{q}}_g^b,\ \dot{\hat{\boldsymbol{b}}}_{gyr} = \boldsymbol{0}_{3\times1}, \\ \dot{\hat{\boldsymbol{v}}}_{gb}^g = \boldsymbol{C}(\boldsymbol{q}_g^b)^T\hat{\boldsymbol{a}} - 2\left[\boldsymbol{\omega}_{ig}^g\wedge\right]\hat{\boldsymbol{v}}_{gb}^g - \left[\boldsymbol{\omega}_{ig}^g\wedge\right]^2\hat{\boldsymbol{p}}_{gb}^g + \boldsymbol{g}_b^g, \\ \dot{\hat{\boldsymbol{b}}}_{acc} = \boldsymbol{0}_{3\times1},\ \dot{\hat{\boldsymbol{p}}}_{gb}^g = \hat{\boldsymbol{v}}_{gb}^g \end{gathered} \tag{4}$$

where $\hat{\boldsymbol{a}} = \boldsymbol{a}_{IMU} - \hat{\boldsymbol{b}}_{acc}$ and $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_{IMU} - \hat{\boldsymbol{b}}_{gyr} - \boldsymbol{C}(\hat{\boldsymbol{q}}_g^b)\boldsymbol{\omega}_{ig}^g$.

These equations are simply integrated using first-order trapezoidal rule given the high frequency $f_{IMU} = 100$ Hz of the IMU.

4.3. Covariance propagation

To estimate the system state in an EKF structure, the state estimation error $\delta\boldsymbol{x}_V = \boldsymbol{x}_V - \hat{\boldsymbol{x}}_V$ is used to provide a linearized model with respect to the estimated state vector given in Eq. 5

$$\delta\dot{\boldsymbol{x}}_V = \boldsymbol{F}_V\delta\boldsymbol{x}_V + \boldsymbol{G}_V\boldsymbol{n}_{IMU} \tag{5}$$

where one can refer to the journal article [17] for the details of these matrices. The error state vector is parametrized with 15 components $\delta\boldsymbol{x}_V = \begin{bmatrix} \delta\boldsymbol{\theta}_g^{bT} & \delta\boldsymbol{b}_{gyr}^T & \delta\boldsymbol{v}_{gb}^{gT} & \delta\boldsymbol{b}_{acc}^T & \delta\boldsymbol{p}_{gb}^{gT} \end{bmatrix}$. The error quaternion $\delta\boldsymbol{q}$ is defined by $\boldsymbol{q} = \hat{\boldsymbol{q}} \otimes \delta\boldsymbol{q}$, with $\otimes$ the quaternion product. Using the small angle approximation, one can write $\delta\boldsymbol{q} \simeq \begin{bmatrix} 1 & \frac{1}{2}\delta\boldsymbol{\theta}^T \end{bmatrix}^T$ and thus $\delta\boldsymbol{\theta}$ is a correct minimal representation of the error quaternion.

In practice, as soon as the second camera image is available, the error state vector of the filter employed for this study contains 21 components such that $\delta\boldsymbol{x} = \begin{bmatrix} \delta\boldsymbol{x}_V^T & \delta\boldsymbol{\theta}_g^{c_{k-1}T} & \delta\boldsymbol{p}_{gc_{k-1}}^{gT} \end{bmatrix}$. Frame $\{c_{k-1}\}$ is that attached to the camera at previous time instant $k-1$. This structure of the error state vector allows to deal with delays associated with image processing time. It keeps in memory the attitude and position of the camera at the last time instant, assumed separated from the current one by the image processing computation time.

Error covariance matrix at time $k-1$ is divided in blocks as shown in Eq. 6:

$$\boldsymbol{P}_{k-1/k-1} = \begin{bmatrix} \boldsymbol{P}_{VV_{k-1/k-1}} & \boldsymbol{P}_{EC_{k-1/k-1}} \\ \boldsymbol{P}_{EC_{k-1/k-1}}^T & \boldsymbol{P}_{CC_{k-1/k-1}} \end{bmatrix} \tag{6}$$

where $\boldsymbol{P}_{VV_{k-1/k-1}}$ is the 15×15 matrix representing the covariance of the vehicle error state vector, $\boldsymbol{P}_{CC_{k-1/k-1}}$ is the 6×6 covariance matrix for the camera error states, and $\boldsymbol{P}_{EC_{k-1/k-1}}$ is the 15×6 correlation matrix for vehicle and camera error states.

This covariance is propagated to time instant k with the help of IMU measurements delivered at frequency f_{IMU} :

$$\boldsymbol{P}_{k/k-1} = \begin{bmatrix} \boldsymbol{P}_{VV_{k/k-1}} & \boldsymbol{P}_{EC_{k/k-1}} \\ \boldsymbol{P}_{EC_{k/k-1}}^T & \boldsymbol{P}_{CC_{k-1/k-1}} \end{bmatrix} \tag{7}$$

with $\boldsymbol{P}_{EC_{k/k-1}} = \boldsymbol{\Phi}(t_{k-1} + 1/f_{IMU}, t_{k-1})\boldsymbol{P}_{EC_{k-1/k-1}}$. Matrix $\boldsymbol{P}_{VV}$ is propagated through numerical integration of Eq. 8 [2]:

$$\dot{\boldsymbol{P}}_{VV} = \boldsymbol{F}_V\boldsymbol{P}_{VV} + \boldsymbol{P}_{VV}\boldsymbol{F}_V^T + \boldsymbol{G}_V\boldsymbol{Q}_{IMU}\boldsymbol{G}_V^T. \tag{8}$$

The transition matrix $\boldsymbol{\Phi}(t_{k-1} + 1/f_{IMU}, t_k)$ between time instants $k-1$ and k of system 5 formulated in dicret time is integrated by definition for $\tau \in \begin{bmatrix} 0, & 1/f_{IMU} \end{bmatrix}$ from Eq. 9:

$$\dot{\boldsymbol{\Phi}}(t_{k-1} + \tau, t_{k-1}) = \boldsymbol{F}_V\boldsymbol{\Phi}(t_{k-1} + \tau, t_{k-1}) \tag{9}$$

with initial condition $\boldsymbol{\Phi}(t_{k-1}, t_{k-1}) = \boldsymbol{I}_{\mathbf{15}}$.

4.4. State management

At each new image capture, the current camera pose estimate $(\hat{\boldsymbol{q}}_g^c, \hat{\boldsymbol{p}}_{gc}^g)$ is appended to the state vector, which then virtually extends to 27 components. The Eqs. 10 and 11 show the computation of this camera pose from the estimated vehicle pose and from the known pose of the camera frame $\{c\}$ with respect to $\{b\}$:

$$\hat{\boldsymbol{q}}_g^c = \hat{\boldsymbol{q}}_g^b \otimes \boldsymbol{q}_b^c \tag{10}$$

$$\hat{\boldsymbol{p}}_{gc}^g = \hat{\boldsymbol{p}}_{gb}^g + \boldsymbol{C}(\boldsymbol{q}_g^b)^T\boldsymbol{p}_{bc}^b \tag{11}$$

The state error covariance matrix is augmented too, with the Jacobian matrix J calculated from Eqs. 10 and 11:

$$\boldsymbol{P}_{k/k} \leftarrow \begin{bmatrix} \boldsymbol{I_{21}} \\ \boldsymbol{J} \end{bmatrix} \boldsymbol{P}_{k/k} \begin{bmatrix} \boldsymbol{I_{21}} \\ \boldsymbol{J} \end{bmatrix}^T \tag{12}$$

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{C}(\boldsymbol{q}_b^c) & \boldsymbol{0}_{3\times 9} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 6} \\ [\boldsymbol{C}(\boldsymbol{q}_g^b)^T \boldsymbol{p}_{bc}^b \wedge] & \boldsymbol{0}_{3\times 9} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 6} \end{bmatrix} \tag{13}$$

One must note that matrix $\boldsymbol{P}_{k/k}$ should be reduced after the following update step of the Kalman filter by removing the covariances corresponding to the previous camera pose.

4.5. Measurement model

Once the states and the covariances of their errors are propagated, it becomes possible to build an update in the EKF from image measurements. The absolute matching algorithm detailed in Section 3 provides the 3D coordinates $\boldsymbol{p}_{gl_j}^g$ in world frame of point j identified at normalized image coordinates $\boldsymbol{z}_j$ at previous time[4] $k-1$. The measurement model is given in Eq. 14:

$$\boldsymbol{z}_j = h_j(\boldsymbol{x}) + \boldsymbol{n_j} = \frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{p}_{cl_j}^c}{[\ 0 \quad 0 \quad 1\ ]\, \boldsymbol{p}_{cl_j}^c} + \boldsymbol{n_j} \tag{14}$$

where $\boldsymbol{p}_{cl_j}^c = \boldsymbol{C}(\boldsymbol{q}_g^c)(\boldsymbol{p}_{gl_j}^g - \boldsymbol{p}_{gc}^g)$ is the 3D position of the point in the camera frame, $\boldsymbol{n_j}$ is the feature extraction noise of covariance matrix $\boldsymbol{R}_j = \sigma_{im}^2 \boldsymbol{I}_2$.

The measurement prediction is computed from this model and from the knowledge of $\boldsymbol{p}_{gl_j}^g$ in Eq. 15:

$$\hat{\boldsymbol{z}}_j = \frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{C}(\hat{\boldsymbol{q}}_g^c)(\boldsymbol{p}_{gl_j}^g - \hat{\boldsymbol{p}}_{gc}^g)}{[\ 0 \quad 0 \quad 1\ ]\, \boldsymbol{C}(\hat{\boldsymbol{q}}_g^c)(\boldsymbol{p}_{gl_j}^g - \hat{\boldsymbol{p}}_{gc}^g)} \tag{15}$$

The innovation $\delta \boldsymbol{z}_j = \boldsymbol{z}_j - \hat{\boldsymbol{z}}_j$ can be expressed linearly in Eq. 16:

$$\delta \boldsymbol{z}_j \simeq \boldsymbol{H}_j \delta \boldsymbol{x} + \boldsymbol{n_j} = \boldsymbol{H}_{j,\delta\boldsymbol{\theta}} \delta \boldsymbol{\theta}_g^{c_{k-1}} + \boldsymbol{H}_{j,\boldsymbol{p}} \delta \boldsymbol{p}_{gc_{k-1}}^g + \boldsymbol{n_j} \tag{16}$$

with

$$\boldsymbol{H}_{j,\delta\boldsymbol{\theta}} = \frac{[\ \boldsymbol{I}_2 \quad -\hat{\boldsymbol{z}}_j\ ] \left[\boldsymbol{C}(\hat{\boldsymbol{q}}_g^c)(\boldsymbol{p}_{gl_j}^g - \hat{\boldsymbol{p}}_{gc}^g)\wedge\right]}{[\ 0 \quad 0 \quad 1\ ]\, \boldsymbol{C}(\hat{\boldsymbol{q}}_g^c)(\boldsymbol{p}_{gl_j}^g - \hat{\boldsymbol{p}}_{gc}^g)}$$

$$\boldsymbol{H}_{j,\boldsymbol{p}} = \frac{[\ \boldsymbol{I}_2 \quad -\hat{\boldsymbol{z}}_j\ ]\, \boldsymbol{C}(\hat{\boldsymbol{q}}_g^c)}{[\ 0 \quad 0 \quad 1\ ]\, \boldsymbol{C}(\hat{\boldsymbol{q}}_g^c)(\boldsymbol{p}_{gl_j}^g - \hat{\boldsymbol{p}}_{gc}^g)}$$

$$\boldsymbol{H}_j = [\ \boldsymbol{0}_{2\times 15} \quad \boldsymbol{H}_{j,\delta\boldsymbol{\theta}} \quad \boldsymbol{H}_{j,\boldsymbol{p}} \quad \boldsymbol{0}_{2\times 6}\ ]$$

By concatenating in columns the innovations $\delta \boldsymbol{z}_j$ and the Jacobian matrices of the measurements $\boldsymbol{H}_j$ for each point j, one can work out the overall innovation $\delta \boldsymbol{z} = \boldsymbol{H}\delta\boldsymbol{x} + \boldsymbol{n}$ which allows for the computation of Kalman gain and subsequently the estimated state and covariance updates.

5. SIMULATION AND RESULTS

In this section, we describe the simulation environment and the preliminary results of the navigation solution proposed.

5.1. Simulator

The results obtained in this section have been generated for descent trajectories to a terrain rendered in the PANGU planetary scene generator [13]. An example of descent image is shown in Fig. 6.

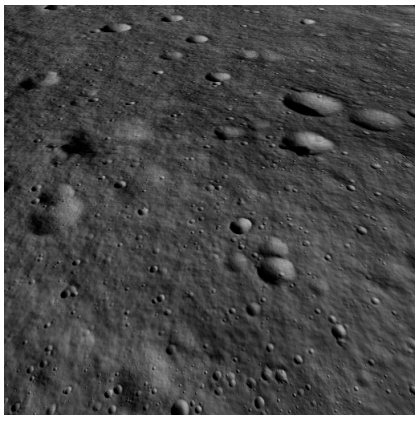


Figure 6: Descent image generated in PANGU.

PANGU-generated descent images cover a field of view of 70 deg with a 512x512 pixels 8-bits sensor. Images are degraded by an additional Gaussian image noise with zero mean and standard deviation equal to one intensity level. The orbital images used for building the on-board are 2048x2048 8-bits images, covering a field of view of 2.5 degrees from a 50-km altitude so as to be similar to NASA's Lunar Reconnaissance Orbiter (LRO) narrow angle camera data [3]. An illumination difference of 20 degrees in azimuth was rendered between the orbital and descent images.

Inertial data are generated through an IMU model calibrated to match performances expected on planetary

[4] As a reminder, delay in measurement processing is due to image processing time.

landing missions of this kind. The trajectory selected corresponds to that of an approach phase for a Moon landing. It lasts 80 seconds, starts at a 2-km altitude to end 10 m above the ground [4]. Guidance is based on that of Apollo Lunar Module [16].

5.2. Matching method comparison

The geometric methods proposed in Subsection 3.3 to generate putative matches were compared to select the most performant one in given simulation conditions. The on-board map was generated from a mosaic of orbital images covering the area seen during the descent and transformed so as to have a resolution equivalent to an image taken at 260 km altitude. This resolution gave the best matching results up to the latest part of the descent for which a standard orbital image with the LRO characteristics described above was used to generate more landmarks at low altitude. 4000 landmarks were extracted from the low-resolution 16x16 km^2 orbital mosaic, and 4000 others from the standard orbital image. 250 features are extracted per online image. The parameters of the Shape Context signature, shown in Fig. 4, are: a minimum radius $br = 10$ pixels, a maximum radius $pr = 100$ pixels, 10 rings and 20 wedges. For the General Hough Transform method (GHT), a circular grid of radius 100 pixels and discretized in squares of 10x10 pixels were used.

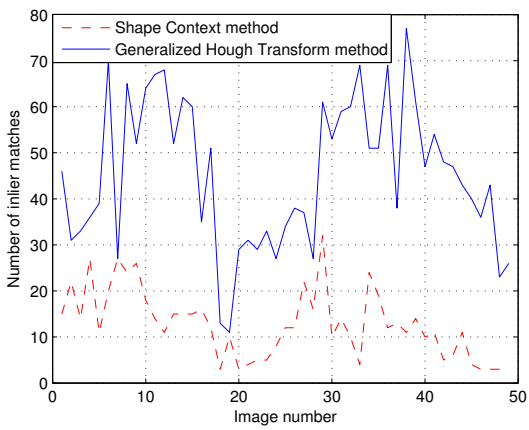


Figure 7: Putative matching method comparison.

The criterion to select between these two methods was the number of landmarks after the RANSAC robust matching step. After RANSAC, inlier matches can be assumed as correct. Fig. 7 shows that the GHT method clearly achieves more matches: 45 against 15 in average. These matches are also more distributed across the image which provides more information to the filter so the GHT was selected as the reference method for putative matching in the following.

5.3. GHT performances

We present the results of a preliminary Monte Carlo performance study of the proposed vision-based navigation filter. 200 Monte Carlo simulations were run using GHT for putative matching in combination with the RANSAC algorithm for robustness and outlier removal. The initial errors were sampled according to independent zero-mean Gaussian distribution on each axis, with 3σ values equal to 1 degree in attitude, 10 m/s in velocity, and 100 m in position. These values are regarded as realistic based on the 3σ uncertainty in attitude and velocity expected during the landing maneuver.

Figs. 8, 9, and 10 show the Monte Carlo results for the estimation error and the 3σ bound[5] from the filter for each axis respectively for attitude, velocity and position of frame $\{b\}$ tied to the vehicle with respect to frame $\{g\}$ tied to the planetary surface. Fig. 11 shows the evolution of matched landmarks with time.

Because of the limited landmarks on the map, less and less points are visible are the lander progresses down its trajectory. After t = 4050 s, the mapped landmark matching block does not see enough landmarks to provide reliable matches. Before t = 4050 s is the so-called visual phase, when image measurement updates are processed by the filter. After t = 4050 s is the inertial phase when IMU measurements only, suffering drift, are propagated. Convergence can be observed on the figures for 93.5 percents of the runs. The overall statistics,including the divergent runs, are given in Table 1.

Table 1: 3σ uncertainties at the end of visual phase (V) and at touchdown (TD)

Variable	$3\sigma_V$	$3\sigma_{TD}$
Attitude / Yaw (deg)	0.3	0.2
Attitude / Pitch (deg)	0.7	0.4
Attitude / Roll (deg)	0.7	0.3
Velocity / X-axis (m/s)	1.7	1.7
Velocity / Y-axis (m/s)	0.7	0.9
Velocity / Z-axis (m/s)	1.3	1.1
Position / X-axis (m)	66.3	120.5
Position / Y-axis (m)	26.8	50.4
Position / Z-axis (m)	52.0	85.2

Position error at the end of the visual phase falls below 67 meters on each axis. The position error increases then until touchdown due to inertial drift. These results include the runs which diverge, namely whose final estimation error is above the 3σ bound estimated by the filter. They led to position errors of up to 400 m in one case, which alter

[5]This 3σ envelope is that of the run which converged but had the largest final standard deviation, it is displayed for illustration purposes but has no other meaning.

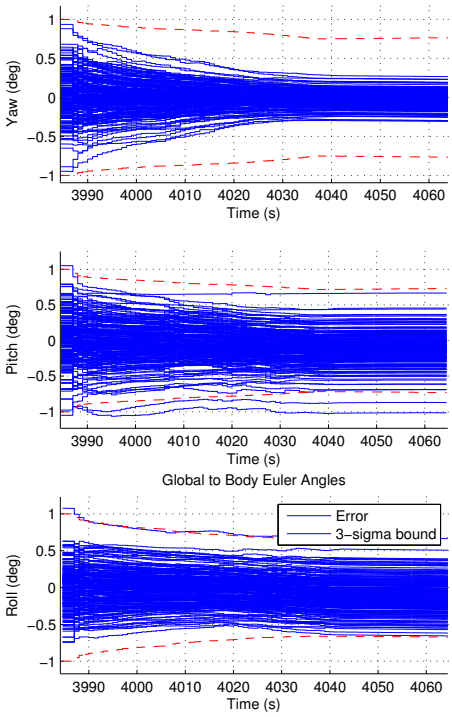


Figure 8: Attitude errors Monte Carlo results.

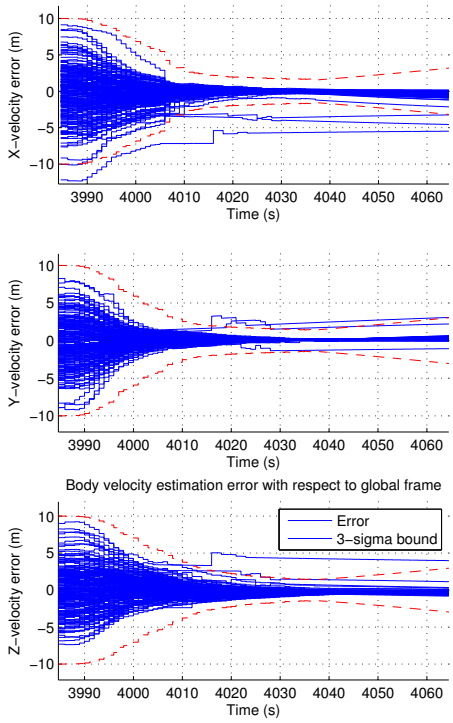


Figure 9: Velocity errors Monte Carlo results.

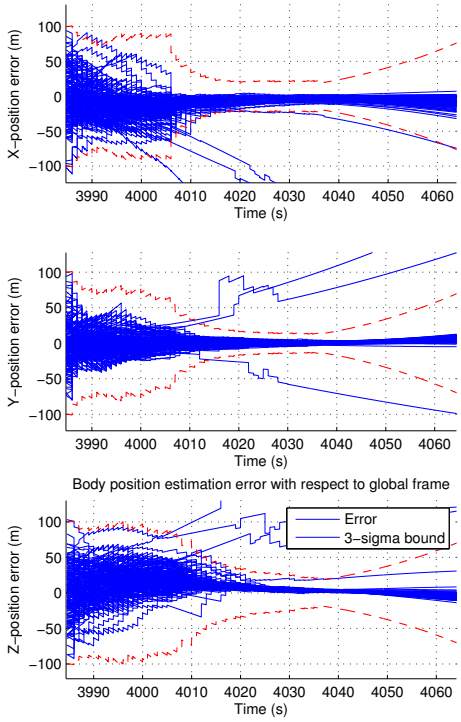


Figure 10: Position errors Monte Carlo results.

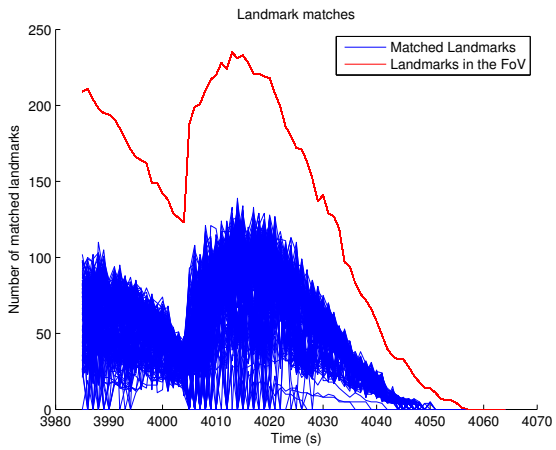


Figure 11: Numbers of landmarks matched with map through time using GHT+RANSAC. Red line above show the actual number of landmarks in the field of view.

the performance statistics strongly. We wanted to have a look at the results for the 93.5 percents of runs which did converge. These are shown in Table 2.

Table 2: 3σ uncertainties at the end of visual phase (V) and at touchdown (TD) for the 93.5 percents of runs which converged

Variable	$3\sigma_V$	$3\sigma_{TD}$
Attitude / Yaw (deg)	0.3	0.2
Attitude / Pitch (deg)	0.6	0.3
Attitude / Roll (deg)	0.6	0.3
Velocity / X-axis (m/s)	0.4	0.7
Velocity / Y-axis (m/s)	0.2	0.3
Velocity / Z-axis (m/s)	0.7	0.4
Position / X-axis (m)	7.4	21.8
Position / Y-axis (m)	4.5	7.0
Position / Z-axis (m)	4.6	10.8

The statistics have tremendously improved using the converging runs only, with position errors falling below 8 meters on each axis at the end of the visual phase, and below 22 meters at touchdown. Reducing the numbers of failures causing divergent runs though improvement of the matching strategy is the subject of on-going work.

5.4. Discussion on EKF linearization

The extended Kalman filter used here employs linearization of the system and measurement model to be able to use regular Kalman filtering architecture. We discuss in this subsection the consequences of this linearization from a filtering point of view.

The true physical sensor noise of associated to image point measurements σ_{im} is of the order of one pixel in 512x512 pixels images. Nevertheless, due to the highly-nonlinear camera projection model, the filter measurement noise can have to be set up to tens of pixels in order to avoid divergence. A value of 40 pixels was used in this article. This value is reached after tuning the filter: the process noise and covariance initialization are tuned to match the system performance (here the inertial navigation performance), and the measurement noise is lowered as much as possible within the stability limit of the filter. Such high values of the measurement noise degrades the efficiency of one measurement to update the state estimate. Improvements could lower the convergence time and the final error.

We evaluated the linearization error $\epsilon_{lin,j} = \|\delta \boldsymbol{z}_j - H_j \delta \boldsymbol{x}\|$ in pixels for each point j in images generated for the trajectory studied in this article. Fig. 12 shows the evolution of the mean of linearization error in the image with respect to altitude. Note that this plot was generated for 1024x1024 pixels images so values should be divided by two to match data studied in the rest of this article for 512x512 images. It can be seen the non-linearities up to 10 pixels, which is largely above the true sensor noise. It is to handle these errors that the measurement noise was made artificially higher (40 pixels) for the EKF to converge, but this decreased its performance. Other architectures such Unscented Kalman Filtering (UKF) or Particle Filtering (PF) are more suited to non-linear systems and constitute a path for future work.

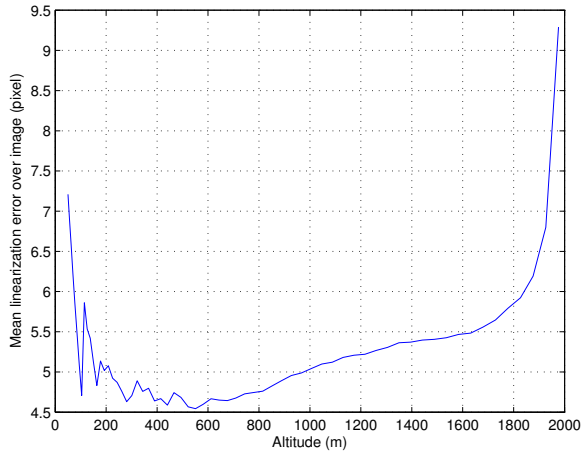


Figure 12: Mean of the linearization error in pixels for all image point measurements in 1024x1024 pixels image.

6. CONCLUSION AND FUTURE WORK

New high-resolution planetary maps generated by recent orbital probes are enabling mapped landmark recognition at very low altitude to increase navigation accuracy. However, the 3D aspects of the terrain must be taken into account when this close to the ground. Handling 3D effects is also important for higher altitude phases of the descent to non-flat bodies, like asteroids. The proposed approach is a tight vision-aided navigation filter fusing data from an IMU with absolute image measurements of features matched with an on-board map through an Extended Kalman filter. The matching step is based on the geometrical distribution of landmarks, is memory-efficient and works under illumination changes. The main feature of our filter is to conduct image processing without planar approximation. This is in contrast with most studies in the field [12, 14], which use planar homography warping in the matching step. Because it is compatible with rugged terrains, our approach allows to increase the duration of the vision-aided navigation phase towards challenging landing sites, hence reducing the final navigation error and enhancing mission return. Preliminary Monte Carlo validation show promising results on a lunar approach phase scenario. Low-altitude global navigation could also be enabled using our approach for planetary aerial mobility vehicles, like airships.

The next step is to work further the understanding the events provoking divergence in 6.5 percents of the runs so as to decrease this figure and make the over all performance better. Implementing a method including filter state uncertainty into the voting process for putative matching to decrease the rate of outliers is to be investigated. Testing should be proceeded to on a complete orbit-to-touchdown lunar landing simulator. This should enhance scale problems in the extraction of image features between the online and orbital images which were noticed already on the low-altitude trajectory tested. If confirmed, other corner extractors taking scale information into account such as the Harris-Laplace features should be tested and their repeatability rates compared to more specific features like craters for instance. The robustness to illumination changes should be also be investigated for more various conditions. It is finally desired to compare the performance of the EKF with that of the UKF and PF to handle non-linearities.

REFERENCES

[1] Belongie, S. & Malik, J. 2000, in IEEE Workshop on Contentbased Access of Image and Video Libraries

[2] Candy, J. V. 2009, Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods (John Wiley & Sons, Inc.)

[3] Chin, G., Brylow, S., Foote, M., et al. 2007, Space Science Reviews, 129, 391

[4] Delaune, J., De Rosa, D., & Hobbs, S. 2010, in AIAA Guidance, Navigation, and Control Conference

[5] Delaune, J., Le Besnerais, G., Farges, J.-L., et al. 2011, in 11th Symposium on Advanced Space Technologies in Robotics and Automation

[6] Duclos, D., Quinquis, N., Broda, G., et al. 2009, in Proceedings of SPIE, the International Society for Optical Engineering

[7] Fischler, M. A. & Bolles, R. C. 1981, Communications of the ACM, 24

[8] Grimson, W. E. L. 1991, Object Recognition by Computer: The Role of Geometric Constraints (MIT Press)

[9] Harris, C. & Stephens, M. 1988, in Proceedings of the 4th Alvey Vision Conference

[10] Hartley, R. & Zisserman, A. 2004, Multiple View Geometry in computer vision, 2nd edn. (Cambridge University Press)

[11] Johnson, A. E. & Montgomery, J. F. 2008, in IEEE Aerospace Conference, Big Sky, MT, USA

[12] Mourikis, A. I., Trawny, N., Roumeliotis, S. I., et al. 2009, IEEE Transactions on Robotics, 25, 264

[13] Parkes, S. M., Martin, I., Dunstan, M., & Matthews, D. 2004, in Eighth International Conference on Space Operations

[14] Pham, B. V., Lacroix, S., Devy, M., Drieux, M., & Philippe, C. 2009, in AIAA Guidance, Navigation and Control, Chicago, IL, USA

[15] Singh, L. & Lim, S. 2008, in AIAA Guidance, Navigation and Control Conference and Exhibit

[16] Sostaric, R. R. & Rea, J. R. 2005, in AIAA Guidance, Navigation, and Control Conference and Exhibit

[17] Trawny, N., Mourikis, A. I., Roumeliotis, S. I., Johnson, A. E., & Montgomery, J. 2007, Journal of Field Robotics