

Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing

Anastasios I. Mourikis, *Member, IEEE*, Nikolas Trawny, *Student Member, IEEE*,
Sergios I. Roumeliotis, *Member, IEEE*, Andrew E. Johnson, Adnan Ansar, and Larry Matthies, *Senior Member, IEEE*

Abstract—In this paper, we present the vision-aided inertial navigation (VISINAV) algorithm that enables precision planetary landing. The vision front-end of the VISINAV system extracts 2-D-to-3-D correspondences between descent images and a surface map (mapped landmarks), as well as 2-D-to-2-D feature tracks through a sequence of descent images (opportunistic features). An extended Kalman filter (EKF) tightly integrates both types of visual feature observations with measurements from an inertial measurement unit. The filter computes accurate estimates of the lander's terrain-relative position, attitude, and velocity, in a resource-adaptive and hence real-time capable fashion. In addition to the technical analysis of the algorithm, the paper presents validation results from a sounding-rocket test flight, showing estimation errors of only 0.16 m/s for velocity and 6.4 m for position at touchdown. These results vastly improve current state of the art for terminal descent navigation without visual updates, and meet the requirements of future planetary exploration missions.

Index Terms—Descent and landing (EDL), entry, localization, sensor fusion, space robotics, vision-aided inertial navigation.

I. INTRODUCTION

FUTURE solar system exploration will include landing and sample-return missions to moons, planets, asteroids, and comets [1]. Autonomous precision landing capabilities (i.e., reducing the landing ellipse to subkilometer accuracy) would provide safe and affordable access to landing sites that promise the highest science return and pose minimal risk to the spacecraft. For some missions, such as sampling potential eruptions of subsurface liquids on Europa, pinpoint landing technology is a critical prerequisite. Precision landing, in turn, requires

Manuscript received January 2, 2008; revised June 23, 2008. Current version published April 3, 2009. This paper was recommended for publication by Associate Editor K. Iagnemma and Editor W. J.-P. Laumond upon evaluation of the reviewers' comments. This work was supported in part by the NASA Mars Technology Program under Grant MTP-1263201, by the NASA ST9 New Millennium Program-System Flight Validation Projects, and by the National Science Foundation under Grant IIS-0643680. Part of the research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This paper was presented in part at the Proceedings of Robotics: Science and Systems, June 26–30 2007, Atlanta, GA, in part at the AIAA Infotech@Aerospace Conference, May 7–10 2007, and in part at the Proceedings of the NASA Science Technology Conference (NSTC'07), College Park, MD, June 2007.

A. I. Mourikis is with the Department of Electrical Engineering, University of California, Riverside, CA 92521 USA (e-mail: mourikis@ee.ucr.edu).

N. Trawny and S. I. Roumeliotis are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: trawny@cs.umn.edu; stergios@cs.umn.edu).

A. E. Johnson, A. Ansar, and L. Matthies are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: aej@jpl.nasa.gov; ansar@jpl.nasa.gov; lhm@jpl.nasa.gov).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2009.2012342



Fig. 1. Sounding Rocket Experiment 41.068.

high-accuracy velocity, position, and attitude (pose) estimation. Due to the lack of infrastructure such as global positioning system (GPS), past robotic lander missions have determined their pose through integration of acceleration and rotational velocity measurements from an inertial measurement unit (IMU), augmented by velocity and altitude information from Doppler radar. These methods suffer from relatively large errors in the touchdown position estimate (e.g., for the Mars Science Laboratory in the order of kilometers), caused by integration of noise and biases, as well as errors in initialization.

An alternative option for increasing the navigation accuracy during entry, descent, and landing (EDL) is to use camera measurements. Cameras operate in almost any EDL scenario, from orbit to touchdown, are small, lightweight, and consume little power, and are therefore excellent sensors for EDL applications. Most importantly, the rich information content of images, if properly used, facilitates precise motion estimation in real time. In particular, two types of visual input can be utilized for spacecraft EDL: First, absolute pose information is obtained by observing features on a planet's surface, whose 3-D positions are known from satellite images and digital elevation maps (DEMs). Second, information about the rotational and translational velocity of the camera can be inferred by tracking the displacement of features in the images.

In this paper, we present a vision-aided inertial navigation (VISINAV) system for planetary landing applications that utilizes both types of information outlined earlier. The visual measurements are combined in an optimal fashion with measurements from an IMU, to produce estimates of the spacecraft's position, attitude, and velocity during EDL. Certain aspects of the VISINAV algorithm have been described in previous publications; a high-level system overview is presented in [2] and [3], and details of the estimator in [4]. The aim of this paper is to provide a detailed, comprehensive description of both the computer vision and estimation algorithms and their integration, as

well as performance evaluation through experiments and sensitivity studies. The main contributions of this paper are the following:

- 1) Image processing algorithms that are specifically adapted to EDL applications. These algorithms process the camera images to extract two types of features: 1) *mapped landmarks* (MLs), i.e., features whose global coordinates can be inferred from a map (a coregistered pair of satellite image and DEM) of the landing area and 2) *opportunistic features* (OFs), i.e., features that can be reliably detected in image sequences, but not in an *a priori* map of the planet's surface.
- 2) An extended Kalman filter (EKF)-based algorithm that estimates the lander's position, velocity, and attitude using visual and inertial measurements. The presented pose-estimation algorithm relies on tight integration of the various sources of information (MLs, OFs, IMU measurements), thus leading to excellent estimation accuracy and robustness in the presence of modeling uncertainties, outliers, and nonlinearities.
- 3) Experimental validation of the VISINAV system during a sounding rocket atmospheric reentry test (cf. Fig. 1). In this experiment, estimation errors of magnitude 0.16 m/s in velocity and 6.4 m in position at touchdown were attained. This accuracy is several orders of magnitude better than the state of the art in EDL, and satisfies the requirements of future planetary exploration missions.

The proposed approach is capable of meeting the hard real-time constraints that arise during spacecraft landing. Specifically, the image-processing component of the system is well-suited for implementation on field programmable gate array (FPGA) hardware, while the estimation algorithm has only linear computational complexity in the number of features. Consequently, efficient real-time operation on spacecraft computers can be ensured.

In the next section, related work on feature extraction and vision-based navigation is discussed. Section III provides an overview of the VISINAV system for EDL. Section IV presents the image processing component of the algorithm, while Section V describes in detail the pose-estimation algorithm. Experimental results from a sounding rocket test, with a dynamic profile similar to planetary landing scenarios, are presented in Section VI. Finally, Section VII provides a sensitivity study of the algorithm, with particular emphasis on the vision front-end. Together, these results demonstrate the VISINAV system's robustness and its superior accuracy compared to current state-of-the-art EDL navigation approaches.

II. RELATED WORK

A. Feature Extraction

Numerous computer vision techniques can be applied to generate image measurements for vision-aided navigation. In the risk-averse world of space science missions, the goal is to develop an algorithm with the following properties:

- 1) Robustness: The algorithm should be robust to matching outliers.

- 2) Computational efficiency: Hard real-time constraints exist during EDL.
- 3) General applicability: Planetary terrains vary widely in appearance, from feature-rich cratered terrain to smooth featureless plains.
- 4) High accuracy: Certain missions are only possible if meter-level landing accuracy is available.

In light of these requirements, we hereafter evaluate several existing approaches. Cheng *et al.* [5], [6] propose using craters as landmarks for navigation. Craters are abundant on most bodies of interest in our solar system, and this makes them a useful feature. Moreover, their detection can be carried out efficiently, and under varying image scale, viewpoint, and illumination conditions. However, there exist sites (e.g., the polar regions of Mars, Europa, comets) where craters are not present, and therefore, more general feature types are required.

One image-feature detection algorithm that was considered during our system's design is the scale-invariant feature transform (SIFT) [7]. SIFT keypoints can be reliably matched between images under large scale and in-plane orientation changes. However, the scale and rotation invariance of the SIFT keys, which increases the processing requirements, is not necessary in the EDL scenario considered here. In particular, initial altitude and attitude are provided by altimeters and star-trackers, respectively, and throughout the descent these quantities are estimated very accurately by the EKF. Thus, at least four DOFs are known fairly accurately, and we can employ this knowledge to increase the computational efficiency of image processing (cf. Section IV-A.2).

In our paper, we have instead chosen to base our algorithm design on Harris corners [8] and normalized correlation [9]. Corner features can be extracted very efficiently, and they typically correspond to textured areas in images, suitable for correlation-based tracking. Additionally, image correlation has already been tested and validated in actual EDL applications. Specifically, correlation was the basis of the descent image motion estimation subsystem (DIMES) of the Mars Exploration Rover (MER) Mission [10]. The successful operation of correlation-based feature matching during the MER spacecraft landing demonstrated its reliability and appropriateness for planetary EDL. Furthermore, many components of the proposed image processing algorithms (i.e., homography transform estimation, homography image warping, Harris interest operator, and spatial correlation) are already contained in the flight-validated software used in MER-DIMES.

Finally, we point out that image correlation does not rely on a specific geometric model of landmarks to enable detection, as is the case for craters; hence it can be applied to any planetary terrain of sufficient texture. Due to its maturity, efficiency, and general applicability, we have chosen 2-D image correlation as the basis for landmark detection and tracking.

B. State Estimation

When the projections of at least three points with known global coordinates (MLs) are detected in an image, an estimate

of the camera pose can be computed.¹ Several algorithms, both closed-form and iterative, have been proposed for this task (e.g., [6], [11]–[13] and references therein). However, in the context of EDL, such approaches are not sufficient because 1) they require that at least 3 MLs are visible at all times and 2) they can only provide pose estimates at the time instants when images are recorded. The latter point is very important, since controlled spacecraft landing requires very low-latency, high-bandwidth pose and velocity estimates. Therefore, some form of filtering must be employed for pose estimation during EDL.

Only a few recursive estimation approaches that utilize measurements of *a priori* known features have been proposed in the literature. In [14], a statistical (zero acceleration) model is employed for propagating the pose estimate between ML observations. However, the use of a statistical model (rather than inertial measurements), limits the applicability of such an approach to maneuvers with slow dynamics that occur, for example, during spacecraft rendezvous and docking. In [15] and [16], inertial measurements are fused with observations of artificial rectangular targets, and with heading measurements from a magnetometer. In their work, the authors use measurements both of the coordinates of a target's projection, as well as of the area of this projection. Area measurements, though, may be difficult or unreliable when dealing with real planetary imagery, where visual features are less structured. In [17], inertial measurements are fused with bearing measurements to MLs, but the spacecraft's attitude is assumed to be perfectly known, which is not a valid assumption for EDL. Finally, [18] and [19] present EKF-based algorithms for fusing IMU measurements and observations of previously mapped features, while estimating the entire spacecraft pose and IMU biases. In addition to processing ML observations, in this paper we also utilize measurements to features that can be tracked in the images, but whose position in the global frame is not known in advance (OFs).

The standard method for treating OFs is to include their positions in the EKF state vector, and to estimate them along with the vehicle's trajectory. This is the well-known simultaneous localization and mapping (SLAM) formulation. SLAM with visual measurements and inertial sensors has recently attracted significant interest [20]–[22]. However, the need to maintain the landmark estimates in SLAM results in increased computational complexity (quadratic in the number of features for EKF-SLAM). Moreover, the main benefit of performing SLAM is the ability to achieve "loop closing" when revisiting an area. Due to the nature of EDL trajectories, loop closing is not an important consideration, and thus the quadratic computational complexity of SLAM does not appear to be justified in the context of EDL. In contrast, our algorithm attains complexity only linear in the number of OFs.

In our work, OF measurements are employed for imposing constraints between multiple camera poses [4], [23]. This can be viewed as a generalization of approaches such as [17], [24], and [25], where only the latest two images are utilized for this

purpose. Pairwise relative-pose constraints have also been employed for pose estimation in approaches that maintain a history of multiple camera poses (e.g., [26], [27], and references therein). Contrary to these, the proposed algorithm does not use the measurements for deriving pairwise relative-pose estimates. This reduces the computational burden, avoids possible correlations between the displacement measurements [28], and is more robust to nonlinearities. Our approach is similar in spirit to the variable state dimension filter (VSDF) [29], where a sliding window of poses is also maintained. However, the VSDF is tailored for cases where no motion model is available (this is not the case in EDL, where IMU measurements can be integrated in a kinematic model). When a motion model is employed, the computational complexity of the VSDF is at least quadratic in the number of features [30].

C. Vision-Based Approaches for Planetary Landing

The DIMES system was used on the 2003 MER mission [10] to estimate the spacecraft's ground-relative horizontal velocity. This estimate controlled retrofiring rockets to limit the horizontal touchdown velocity to within landing gear specifications. In particular, three images were taken at approximately 2000 m, 1700 m, and 1400 m above ground. Feature templates were warped and scaled based on attitude and altitude estimates from the gyroscopes and altimeter, so that 2-D correlation could be employed for feature tracking. A total of four feature tracks (two between first and second image, and two between second and third) were used to compute a velocity estimate. Note that, contrary to the algorithm proposed in this paper, DIMES did not support tight integration of visual and inertial measurements. The IMU-based state estimate was not fused with the velocity measurements obtained from the feature tracks, but only served as a sanity check to validate the image-based results.

In [31], a vision-aided inertial navigation system is described, which relies on IMU measurements and observations of OFs. This system, currently under development by the European Space Agency (ESA), is based on an EKF estimator in which the positions of the features are included in the state vector for performing SLAM. A key difference compared to our study is that no MLs are used, and therefore, the global lander position cannot be accurately estimated. Instead, a safe landing site is selected based on the descent images, and navigation is performed relative to this site. Unfortunately, [31] only contains a description of the overall system architecture, but no experimental validation. Finally, a vision-based EDL navigation system based on optic flow computation is proposed in [32]. This system extracts velocity information directly from the optic flow, as well as absolute pose information from matching a DEM of the landing site with a 3-D surface map computed from the optic flow. Contrary to the VISINAV system described in this paper, however, no tight integration of the visual and inertial measurements occurs.

III. VISINAV SYSTEM OVERVIEW

The VISINAV system developed in our work employs an EKF for fusing the inertial measurements of acceleration and

¹Although, in general, observations of four known points are required for uniquely determining the pose of a camera, only three are necessary when a prior estimate is available.

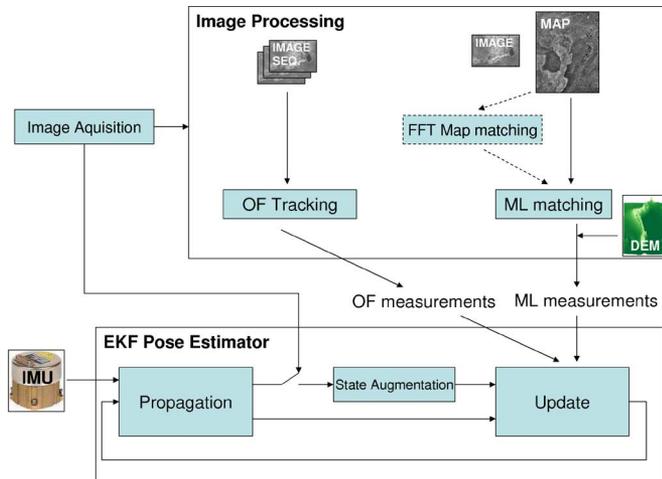


Fig. 2. VISINAV block diagram.

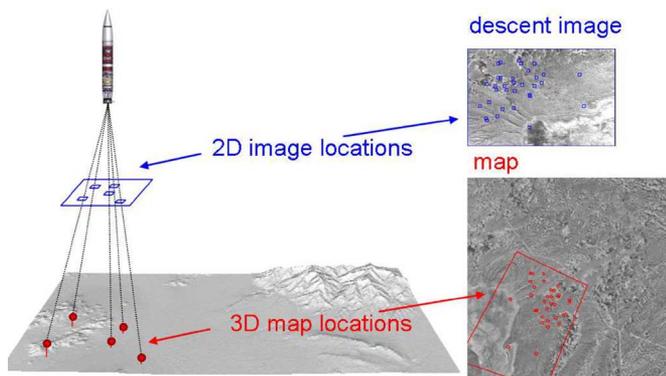


Fig. 3. ML algorithm concept: by matching templates between descent images and a map of the landing site, the algorithm produces measurements of the image projections of features with known 3-D coordinates (i.e., MLs).

rotational velocity with visual feature observations (cf. Fig. 2). The goal of the system is to estimate the 3-D position, orientation, and velocity of the *body frame* $\{B\}$, which is affixed to the spacecraft's IMU, with respect to a *global frame* of reference $\{G\}$. In this work, $\{G\}$ is selected as a planet-centered, planet-fixed frame of reference, which rotates with the planet. As briefly mentioned in the introduction, the system uses two types of visual measurements: MLs and OFs. We now justify this approach, based on the specific characteristics of EDL that affect motion estimation.

During EDL, at the point where imaging becomes possible, the position errors are typically large: for instance, in the order of 100 m for small bodies [33], and of 10 km for the Mars Science Laboratory (MSL) at the time of parachute deployment [34]. Clearly, if meter-level position accuracy is required at touchdown, these position errors are unacceptably large. In order to reduce their magnitude, absolute pose information must be utilized. In our study, this is provided by the ML measurements. The key idea of the ML algorithm is illustrated in Fig. 3. The algorithm produces measurements of the image projections of features with known 3-D coordinates, by matching templates between descent images (acquired during EDL) and a map of

the surface (known from satellite images). These are the ML measurements, which are processed in the EKF for state updates (cf. Section V-D). Since each of the MLs has known coordinates on the planet surface (absolute coordinates), observing such features provides information about the absolute pose of the camera.

ML matching, however, becomes increasingly difficult as the lander approaches the ground. The larger the resolution difference between the descent image and the surface map, the less reliable the matching process becomes. The cutoff point is mission-dependent; for the case of Mars descent, this occurs during the powered descent phase, nominally between 1000 m and 500 m above ground level. At this point, to reduce the rate of position error growth, our approach relies on measurements of OFs. Specifically, we obtain constraints between all the poses from which each OF was seen, by tracking its motion in consecutive images. This information is optimally combined (up to linearization errors) with IMU measurements in the EKF, as described in Section V-E.

In our system, IMU measurements are processed continuously for propagating the pose estimates. Every time a new image is recorded, a copy of the current camera pose is appended to the state vector (state augmentation), and the image processing module begins processing the latest image. Once the ML and/or OF measurements of this image become available, an EKF update takes place. In particular, the ML measurements of the latest image are processed immediately as they become available, while OF updates occur whenever an OF, which has been tracked in a number of images, is lost by the tracker. At that time, all the measurements of this feature are used for an EKF update, as detailed in Section V-E. The following sections present the various components of the VISINAV system in detail.

IV. IMAGE PROCESSING

We first describe the image-processing front-end of the VISINAV system, which is responsible for providing the ML and OF measurements to the estimator.

A. ML Algorithm

Depending on the uncertainty of the camera pose estimate, the ML algorithm consists either of a single step, *ML matching*, or of two consecutive ones, *fast Fourier transform (FFT) map matching* followed by *ML matching* (cf. Fig. 2). The reasons for selecting this two-tier scheme are described hereafter.

During nominal operation, the camera pose is very accurately known from the EKF, and ML matching (Section IV-A.2) can be directly applied to locate image templates in the map. When the first image is recorded, however, only the camera altitude and the attitude² are available, while its horizontal position is very uncertain. In this case, robustness considerations require

²The altitude is known from radiometric tracking, optical approach imagery of the whole planet, and/or altimetry. After star-tracker initialization, the attitude is computed by integrating the rotational velocity measurements provided by the IMU.

Algorithm 1 FFT Map Matching

Inputs: map image, descent image, lander attitude estimate and altitude measurement

Output: single 2D match position in the map image

Algorithm:

1. Select a feature in descent image (Harris operator)
2. Warp image around selected feature to create large template using a homography transform constructed from the camera geometric model and the attitude and altitude measurements
3. Normalize template and map intensities by subtracting mean and dividing by variance in 5×5 pixel windows
4. Correlate template with map using FFT
5. Mark match as valid if correlation peak passes checks on height, width, and ratio to second-highest peak
6. If match is valid, pass it to ML matching algorithm

searching very large areas of the map for matching each descent image template. Since this will induce unacceptably high computational cost if carried out by direct spatial correlation, in our algorithm ML matching is preceded by FFT-based map matching (Section IV-A.1). This additional step provides a rough horizontal position estimate, which is subsequently used to reduce the size of the search areas for ML matching. The details of both FFT matching and ML matching are described in what follows.

1) *FFT Map Matching:* The goal of the FFT map-matching step is to obtain a coarse estimate of the camera pose relative to the surface map. This process can be implemented very efficiently by utilizing additional sensor information. In particular, in operational missions, attitude is normally known with error smaller than 1° , from star tracker initialization and IMU integration. Moreover, when the terrain is within range of radar altimetry (e.g., 2.4 km altitude for the MER landers), the lander altitude is known to within 1%. Therefore, when images become available, only two DOFs, corresponding to horizontal position relative to the surface, are highly uncertain. This allows us to reduce map matching to a 2-D search, which is significantly faster than a search over all the 6 DOFs of the camera pose.

The map-matching algorithm (cf. Algorithm 1) starts by employing the standard Harris corner detector [8] to select a feature in the descent image. Using the lander altitude and attitude estimates, and assuming that the planet surface is locally approximately flat, the homography \mathbf{H}_w that describes the transformation between the template in the image, τ , and the map is computed by a standard texture mapping technique [35], [36]. Specifically, we consider a coordinate frame on the ground, having its x and y axes parallel to the map, its z -axis pointing down, and its origin directly beneath the lander. The position of the camera in this frame is given by $\mathbf{t} = [0 \ 0 \ -a]^T$, where a is the lander altitude estimate.

The first step for estimating \mathbf{H}_w is to project the four corner points of the template, $\mathbf{u}_{\tau_i} = [u_{\tau_i} \ v_{\tau_i} \ 1]^T$, $i = 1 \dots 4$ to points on the ground. Specifically, the optical rays passing through these points are described by the equations $\mathbf{t} + \lambda \mathbf{u}_{g_i}$, where $\lambda \in \mathbb{R}_+$, and \mathbf{u}_{g_i} , $i = 1 \dots 4$ are the unit vectors defined by the camera optical center and the template corners. These

unit vectors are expressed in the ground frame, using the known internal camera parameters and the camera attitude estimate. The intersections of these rays with the ground plane (i.e., with the plane $z = 0$), are given by:

$$\mathbf{p}_{g_i} = \frac{a}{\mathbf{e}_3^T \mathbf{u}_{g_i}} \mathbf{u}_{g_i} + \mathbf{t}, \quad i = 1 \dots 4 \quad (1)$$

where $\mathbf{e}_3 = [0 \ 0 \ 1]^T$. The corresponding pixel coordinates in the map are then computed from the orthographic map projection:

$$\mathbf{u}_{m_i} = \text{diag}(s_x, s_y, 1) \mathbf{p}_{g_i}, \quad i = 1 \dots 4 \quad (2)$$

where s_x and s_y are scalar constants depending on the map scale. Given the four point correspondences between points in the image template, \mathbf{u}_{τ_i} , and points in the map, $\mathbf{u}_{m_i} = [u_{m_i} \ v_{m_i} \ 1]^T$, the homography \mathbf{H}_w is computed by solving the linear system [36]

$$\mathbf{M}\mathbf{h} = \mathbf{b} \quad (3)$$

where \mathbf{h} is an 8×1 vector parameterizing the homography

$$\mathbf{H}_w = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \quad (4)$$

while \mathbf{M} is a matrix with block rows \mathbf{M}_i , $i = 1 \dots 4$, given by

$$\mathbf{M}_i = \begin{bmatrix} u_{\tau_i} & v_{\tau_i} & 1 & 0 & 0 & 0 & -u_{\tau_i} u_{m_i} & -u_{m_i} v_{\tau_i} \\ 0 & 0 & 0 & u_{m_i} & v_{m_i} & 1 & -u_{\tau_i} v_{m_i} & -v_{\tau_i} v_{m_i} \end{bmatrix} \quad (5)$$

and \mathbf{b} is a block vector with elements $\mathbf{b}_i = [u_{m_i} \ v_{m_i}]^T$, $i = 1 \dots 4$. \mathbf{H}_w transforms template pixels to map pixels, so it is used to warp the descent image around the selected feature to create a template that has the same scale and orientation as the map. An efficient implementation of this warp is described in [11].

Before convolving with the map, the intensity $i(u, v)$ of the warped template is normalized for mean $m_{i(u,v)}$ and standard deviation $\sigma_{i(u,v)}$ over local 5×5 pixel windows, i.e., the intensity values of the normalized template are given by

$$i_n(u, v) = \frac{i(u, v) - m_{i(u,v)}}{\sigma_{i(u,v)}}. \quad (6)$$

The normalization reduces the sensitivity of correlation to illumination differences (e.g., the opposition effect [34] and shadows from large features) between the map and the image.

We note that, to avoid false matches, a relatively large template (larger than 32×32 pixels) is used in the FFT map-matching algorithm.³ Moreover, the map image is typically very large (e.g., 2048×2048 pixels). Thus, the step of convolving the template with the map can be very computationally demanding, unless convolution is carried out in the frequency domain [37]. For efficiency, the FFT of the normalized map is calculated off-line, and stored in memory. The steps required on-line for

³This increases robustness, but will also result in slightly worse template localization, if the scene is not flat. However, this is acceptable, since we only seek to obtain a coarse pose estimate in the FFT map-matching step.

Algorithm 2 ML Matching

Inputs: map, descent image, coarse lander position and attitude relative to map

Output: Multiple ML matches between image locations (u, v) and map locations (x, y, z)

Algorithm:

1. Select multiple templates in descent image (Harris operator)
2. Compute homography that rectifies descent image to map using the lander position and attitude estimate
3. Warp each template using homography
4. Spatially correlate each warped template with map
5. Mark match as valid if correlation peak passes checks on height, width and ratio to second-highest peak
6. If match is valid, generate ML

computing the correlation image are: 1) computation of the FFT of the descent image template; 2) multiplication of the FFTs of the map and the template to obtain the FFT of the correlation image; and 3) use of the inverse FFT to obtain the correlation image.

Once the correlation image is computed, the pixel with the peak correlation is found, and a biquadratic surface

$$au^2 + bv^2 + cuv + du + ev + f \quad (7)$$

is fit to the 3×3 pixel neighborhood around it. The biquadratic is used to solve for the subpixel image match location

$$u_p = (-2bd + ce)/(4ab - c^2) \quad (8)$$

$$v_p = (-2ae + cd)/(4ab - c^2). \quad (9)$$

The second-highest peak outside the neighborhood of the biquadratic fit is also found. To eliminate incorrect matches, thresholds on the minimum height and maximum width of the highest correlation peak, as well as on the ratio of the highest to the second-highest peak are imposed. If the highest peak passes these tests, a match is declared, and the algorithm terminates. The horizontal lander position computed from map matching, along with the altimeter measurement of height and the IMU-based attitude estimate, form the desired estimate of the camera pose.

2) *ML Matching:* For the ML matching algorithm (cf. Algorithm 2), 50–100 small templates (e.g., 15×15 pixels) are selected in the descent image using the Harris corner operator. The camera pose estimate and its uncertainty, which are available either from the EKF or from the FFT map-matching process, are used to define 1) a homography \mathbf{H}_w that rectifies the image to the map (computed as described earlier) and 2) search windows in the map, where template matches will be sought. Each of the selected descent image templates is warped to the map frame using \mathbf{H}_w , and matched in the spatial domain by finding the peak of the pseudonormalized cross correlation of the template τ with the corresponding map search

window⁴ w :

$$\begin{aligned} \gamma(x_o, y_o) &= \frac{2 \sum_{x,y} ((w(x,y) - m_w)(\tau(x - x_o, y - y_o) - m_\tau))}{\sum_{x,y} (w(x,y) - m_w)^2 + \sum_{x,y} (\tau(x - x_o, y - y_o) - m_\tau)^2} \end{aligned}$$

where m_τ is the mean of the template intensity and m_w is the mean of $w(x, y)$ in the region under the template.

After correlation, subpixel matching and validity checks are applied for each template, in the same way as in the map-matching algorithm. If a successful match is declared, the feature's latitude and longitude from the map are combined with elevation information from a coregistered DEM to obtain the global 3-D coordinates of the ML. The resulting pairings between 2-D image coordinates and 3-D feature positions are passed to the EKF for processing. An example of ML matching applied on data collected during a rocket flight experiment (cf. Section VI) is shown in the multimedia material accompanying the paper.

B. OF Tracking

In some cases, feature matching between image and map is unreliable, e.g., due to large resolution differences at low altitudes. In these situations, however, it is still possible to track features between images. Several feature tracking techniques have been proposed for tracking templates when the change in the appearance of a scene between images is small. The most widely used technique is the Kanade–Lucas–Tomasi (KLT) tracker [38], and several variations of it [39], [40]. Planetary descent imposes additional problem-specific challenges; in particular, significant changes in image scale and orientation over short time periods due to the dynamic flight profile. Attitude variations can be large due to parachute oscillations and maneuvers during powered terminal descent. In addition, as the vehicle approaches the planet surface, the image scale continuously increases. If these effects are not accounted for, any feature tracker based on 2-D correlation will quickly lose track of features.

In our study, we compensate for changes in scale and orientation by adding a homography image-warping stage before 2-D correlation. In particular, assuming that the scene is relatively flat (a reasonable assumption for planetary landing sites), a homography can be applied to warp the initial image to successive images. Given a set of features in one image and a set of matching features in a subsequent image, a homography can be computed that aligns the images (and thus the feature templates).⁵ Similar to the case of map matching, this homography is computed by solving a linear system [cf. (3)–(5)]. Since, in this case, multiple (i.e., more than four) matched templates are

⁴We point out that each template is correlated independently, to allow for differences in elevation and camera radial distortions across the descent image not captured in the homography transform.

⁵Alternatively, the homography could be computed using the IMU estimate of rotation, and the altimeter measurement of height above ground. However, for relatively small interframe rotations (less than 12° in our experiments), the vision-only approach works well, so we have chosen not to rely on external information.

Algorithm 3 OF Tracking**Input:** sequence of descent images**Outputs:** multiple correspondences between image pixels in initial and successive images**Algorithm:**

1. Select multiple features in initial image (Harris operator)
2. current features = features selected in first image
3. current image = first image, next image = second image
4. Repeat until number of tracks is small:
 - a. 2D correlate current features between current and next image to get current-to-next tracks
 - b. Concatenate initial-to-current tracks with current-to-next tracks to get initial-to-next tracks
 - c. Compute homography from initial to next tracks
 - d. Warp initial features using homography
 - e. 2D correlate warped initial features w/ next image
 - f. Discard tracks with poor correlation peak
 - g. Compute homography based on remaining tracks
 - h. Discard tracks with residual with respect to homography larger than one pixel, and recompute homography
 - i. Set current feature templates to projection of initial features templates based on final homography
 - j. Increment current image
 - k. Increment next image

typically available, this system is overdetermined, and thus a least-squares solution is obtained [36].

The OF tracking algorithm comprises interleaved steps of 1) homography estimation, based on the current set of feature matches; 2) template warping given this homography; and 3) 2-D spatial correlation-based matching of the warped templates. This tracking process is very stable, because the only free parameter during correlation is the template displacement (instead of displacement and the parameters of the template warp) [39]. Additionally, since multiple features are used to estimate the common homography, this estimate is very accurate, which, in turn, increases the localization accuracy of the template matches. Matches with poor correlation peaks or excessive residuals with respect to the homography are discarded. No RANSAC was employed at this stage, since the number of outliers is very small, and simply discarding features with large residuals works very well. Once too many feature tracks have been lost, new features are again initialized using the Harris corner detector [8]. The detailed steps of the entire OF tracking algorithm are shown in Algorithm 3, while an example of its performance is shown in the multimedia material accompanying the paper.

V. ESTIMATOR DESCRIPTION

An overview of the EKF pose estimation algorithm is given in Algorithm 4. The IMU measurements are processed immediately as they become available for propagating the state and covariance estimates, as shown in Section V-B. On the other hand, each time an image is recorded, the current camera pose estimate is appended to the state vector, and the covariance matrix is appropriately augmented. State augmentation is necessary for two reasons: First, due to the time delays introduced

Algorithm 4 EKF Pose Estimation Algorithm**Propagation:** For each IMU measurement received, propagate the state and covariance estimates (cf. Section V-B).**Image registration:** Every time a new image is recorded,

- augment the state and covariance matrix with a copy of the current camera pose estimate (cf. Section V-C).
- image processing module begins processing the new image (cf. Section IV).

Update: When the ML and OF measurements of a given image become available, perform an EKF update using

- all ML measurements of this image (cf. Section V-D)
- the OFs that are no longer detected in the image sequence (cf. Section V-E).

by the image processing module, the camera measurements are not available immediately.⁶ Second, maintaining a window of camera poses enables the processing of OF measurements (cf. Section V-E). Therefore, at any time instant, the EKF state vector comprises 1) the *evolving* state, \mathbf{x}_E , which describes the current state of the spacecraft and 2) copies of N past poses of the camera. The maximum length of the camera pose history, N , is selected by preflight testing, and is chosen to be equal to the maximum number of images through which an OF can be tracked. In the following sections, we present the various steps of the algorithm in detail. Note that any additional exteroceptive sensor measurements (e.g., altimeter, Doppler radar) can be naturally incorporated and processed in the EKF. However, in the following sections, we focus on the processing of the visual measurements, which is the main contribution of this work.

A. Structure of the EKF State Vector

The evolving state of the EKF is described by the vector

$$\mathbf{x}_E = [{}^B_G\bar{\mathbf{q}}^T \quad \mathbf{b}_g^T \quad G\mathbf{v}_B^T \quad \mathbf{b}_a^T \quad G\mathbf{p}_B^T]^T \quad (10)$$

where ${}^B_G\bar{\mathbf{q}}$ is the unit quaternion [41] describing the rotation from the global frame to the body frame, $G\mathbf{p}_B$ and $G\mathbf{v}_B$ are the position and velocity of the body expressed with respect to the global frame, and finally \mathbf{b}_g and \mathbf{b}_a are 3×1 vectors that describe the biases affecting the gyroscope and accelerometer measurements, respectively. The IMU biases are modeled as random walk processes, driven by the white Gaussian noise vectors \mathbf{n}_{wg} and \mathbf{n}_{wa} , respectively.

Given the definition of the evolving state in (10), the error-state vector for \mathbf{x}_E is defined accordingly, as

$$\tilde{\mathbf{x}}_E = [\delta\boldsymbol{\theta}_B^T \quad \tilde{\mathbf{b}}_g^T \quad G\tilde{\mathbf{v}}_B^T \quad \tilde{\mathbf{b}}_a^T \quad G\tilde{\mathbf{p}}_B^T]^T \quad (11)$$

For the position, velocity, and biases, the standard additive error definition is used (i.e., the error in the estimate \hat{x} of a quantity x is defined as $\tilde{x} = x - \hat{x}$). However, for the quaternion, a different error definition is employed. In particular, if $\hat{\mathbf{q}}$ is the estimated

⁶Consider that at time-step k , an image is recorded, and that the image measurements become available at time-step $k + d$. During the time interval $[k, k + d]$, IMU measurements are continuously processed for state propagation. When, at time-step $k + d$, the feature measurements that occurred at time-step k become available, applying an EKF update is possible, because the camera pose at time-step k is included in the state vector.

value of the quaternion $\bar{\mathbf{q}}$, then the orientation error is described by the *error quaternion* $\delta\bar{\mathbf{q}}$, which is defined by the relation $\bar{\mathbf{q}} = \delta\bar{\mathbf{q}} \otimes \hat{\mathbf{q}}$. In this expression, the symbol \otimes denotes quaternion multiplication. The error quaternion is

$$\delta\bar{\mathbf{q}} \simeq \left[\frac{1}{2} \delta\boldsymbol{\theta}^T \quad 1 \right]^T. \quad (12)$$

Since attitude corresponds to 3 DOFs, using $\delta\boldsymbol{\theta}$ to describe the attitude errors results in a minimal representation.

Assuming that N camera poses are included in the EKF state vector at time-step k , this vector has the following form:

$$\hat{\mathbf{x}}_k = [\hat{\mathbf{x}}_{E_k}^T \quad C_1 \hat{\mathbf{q}}^T \quad G \hat{\mathbf{p}}_{C_1}^T \quad \dots \quad C_N \hat{\mathbf{q}}^T \quad G \hat{\mathbf{p}}_{C_N}^T]^T \quad (13)$$

where $C_i \hat{\mathbf{q}}$ and $G \hat{\mathbf{p}}_{C_i}$, $i = 1 \dots N$ are the estimates of the camera attitude and position, respectively. The EKF error-state vector is defined accordingly

$$\tilde{\mathbf{x}}_k = [\tilde{\mathbf{x}}_{E_k}^T \quad \delta\boldsymbol{\theta}_{C_1}^T \quad G \tilde{\mathbf{p}}_{C_1}^T \quad \dots \quad \delta\boldsymbol{\theta}_{C_N}^T \quad G \tilde{\mathbf{p}}_{C_N}^T]^T. \quad (14)$$

B. Propagation

Every time a new IMU measurement is received, it is used for propagating the EKF state and covariance estimates. To derive the filter propagation equations, we employ discretization of the continuous-time IMU system model, as outlined in the following:

1) *Continuous-Time System Modeling*: The system model describing the time evolution of the evolving state is [42]

$$\begin{aligned} \dot{\mathbf{q}}^B &= \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}(t)) \mathbf{q}^B, \quad \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t) \\ \dot{\mathbf{v}}_B &= G \mathbf{a}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad \dot{\mathbf{p}}_B = G \mathbf{v}_B(t). \end{aligned} \quad (15)$$

In these expressions, $G \mathbf{a}$ is the body acceleration in the global frame, $\boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]^T$ is the body rotational velocity expressed in the body frame, and $\boldsymbol{\Omega}(\boldsymbol{\omega})$ is defined as

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

The gyroscope and accelerometer measurements, $\boldsymbol{\omega}_m$ and \mathbf{a}_m , respectively, are given by

$$\begin{aligned} \boldsymbol{\omega}_m &= \boldsymbol{\omega} + C(\hat{\mathbf{q}}^B) \boldsymbol{\omega}_G + \mathbf{b}_g + \mathbf{n}_g \\ \mathbf{a}_m &= C(\hat{\mathbf{q}}^B) (G \mathbf{a} - G \mathbf{g} + 2[\boldsymbol{\omega}_G \times] G \mathbf{v}_B + [\boldsymbol{\omega}_G \times]^2 G \mathbf{p}_B) \\ &\quad + \mathbf{b}_a + \mathbf{n}_a \end{aligned}$$

where $C(\cdot)$ denotes the rotational matrix corresponding to the quaternion argument, and \mathbf{n}_g and \mathbf{n}_a are zero-mean, white Gaussian noise processes. It is important to note that, since the frame $\{G\}$ is not inertial, but rather planet-fixed, the IMU measurements incorporate the effects of the planet's rotation, $\boldsymbol{\omega}_G$. Moreover, the accelerometer measurements include the gravitational acceleration $G \mathbf{g}$, expressed in the local frame.

Applying the expectation operator to the state propagation equations [cf. (15)], we obtain the equations for propagating the

estimates of the evolving state

$$\begin{aligned} \dot{\mathbf{q}}^B &= \frac{1}{2} \boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}) \hat{\mathbf{q}}^B, \quad \dot{\mathbf{b}}_g = \mathbf{0}_{3 \times 1} \\ \dot{\mathbf{v}}_B &= C_{\hat{\mathbf{q}}}^T \hat{\mathbf{a}} - 2[\boldsymbol{\omega}_G \times] G \hat{\mathbf{v}}_B - [\boldsymbol{\omega}_G \times]^2 G \hat{\mathbf{p}}_B + G \mathbf{g} \\ \dot{\mathbf{b}}_a &= \mathbf{0}_{3 \times 1}, \quad \dot{\mathbf{p}}_B = G \hat{\mathbf{v}}_B \end{aligned} \quad (16)$$

where, for brevity, we have denoted $C_{\hat{\mathbf{q}}} = C(\hat{\mathbf{q}}^B)$, $\hat{\mathbf{a}} = \mathbf{a}_m - \hat{\mathbf{b}}_a$, and $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g - C_{\hat{\mathbf{q}}} \boldsymbol{\omega}_G$. The linearized continuous-time model for the evolving error state is given by

$$\dot{\tilde{\mathbf{x}}}_E = \mathbf{F}_E \tilde{\mathbf{x}}_E + \mathbf{G}_E \mathbf{n}_{\text{IMU}} \quad (17)$$

where $\mathbf{n}_{\text{IMU}} = [\mathbf{n}_g^T \quad \mathbf{n}_{wg}^T \quad \mathbf{n}_a^T \quad \mathbf{n}_{wa}^T]^T$ is the system noise. The covariance matrix of \mathbf{n}_{IMU} , \mathbf{Q}_{IMU} , depends on the IMU noise characteristics and is computed off-line during sensor calibration. Finally, the matrices \mathbf{F}_E and \mathbf{G}_E , that appear in (17) are

$$\mathbf{F}_E = \begin{bmatrix} -[\hat{\boldsymbol{\omega}} \times] & -\mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ -C_{\hat{\mathbf{q}}}^T [\hat{\mathbf{a}} \times] & \mathbf{0}_{3 \times 3} & -2[\boldsymbol{\omega}_G \times] & -C_{\hat{\mathbf{q}}}^T & -[\boldsymbol{\omega}_G \times]^2 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}$$

where \mathbf{I}_3 is the 3×3 identity matrix, and

$$\mathbf{G}_E = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -C_{\hat{\mathbf{q}}}^T & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}.$$

2) *Discrete-time Implementation*: The IMU samples the signals $\boldsymbol{\omega}_m$ and \mathbf{a}_m with a period T , and these measurements are used for state propagation in the EKF. Every time a new IMU measurement is received, the IMU state estimate is propagated using 4th-order Runge–Kutta numerical integration of (16). Moreover, the covariance matrix of the EKF state has to be propagated. For this purpose, we introduce the following partitioning for the covariance matrix:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{EE_{k|k}} & \mathbf{P}_{EC_{k|k}} \\ \mathbf{P}_{EC_{k|k}}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix} \quad (18)$$

where $\mathbf{P}_{EE_{k|k}}$ is the 15×15 covariance matrix of the evolving state, $\mathbf{P}_{CC_{k|k}}$ is the $6N \times 6N$ covariance matrix of the camera pose estimates, and $\mathbf{P}_{EC_{k|k}}$ is the $15 \times 6N$ correlation matrix between the errors in the evolving state and the camera pose estimates. With this notation, the covariance matrix of the propagated state is given by

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{EE_{k+1|k}} & \boldsymbol{\Phi}(t_k + T, t_k) \mathbf{P}_{EC_{k|k}} \\ \mathbf{P}_{EC_{k|k}}^T \boldsymbol{\Phi}(t_k + T, t_k)^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix}$$

where $\mathbf{P}_{EE_{k+1|k}}$ is the covariance of the evolving state at time-step $k+1$, given IMU measurements up to time-step $k+1$, and

exteroceptive measurements up to time-step k . This is computed by numerical integration of the Lyapunov equation

$$\dot{\mathbf{P}}_{EE} = \mathbf{F}_E \mathbf{P}_{EE} + \mathbf{P}_{EE} \mathbf{F}_E^T + \mathbf{G}_E \mathbf{Q}_{IMU} \mathbf{G}_E^T. \quad (19)$$

Numerical integration is carried out for the time interval $(t_k, t_k + T)$, with initial condition $\mathbf{P}_{EE|k}$. The state transition matrix $\Phi(t_k + T, t_k)$ is similarly computed by numerical integration of the differential equation

$$\dot{\Phi}(t_k + \tau, t_k) = \mathbf{F}_E \Phi(t_k + \tau, t_k), \quad \tau \in [0, T] \quad (20)$$

with initial condition $\Phi(t_k, t_k) = \mathbf{I}_{15}$.

C. State Augmentation

When a new image is recorded, the camera pose estimate is computed from the body pose estimate as follows:

$$\mathbf{C}_G^C \hat{\mathbf{q}} = \mathbf{C}_B^B \bar{\mathbf{q}} \otimes \mathbf{C}_G^B \hat{\mathbf{q}} \quad (21)$$

$${}^G \hat{\mathbf{p}}_C = {}^G \hat{\mathbf{p}}_B + \mathbf{C}_{\hat{\mathbf{q}}}^T \mathbf{p}_C \quad (22)$$

where $\mathbf{C}_B^B \bar{\mathbf{q}}$ is the quaternion expressing the rotation between the body and camera frames, and ${}^B \mathbf{p}_C$ is the position of the origin of the camera frame with respect to $\{B\}$, both of which are known. This camera pose estimate is appended to the state vector, and the covariance matrix of the EKF is augmented accordingly

$$\mathbf{P}_{k|k} \leftarrow \begin{bmatrix} \mathbf{I}_{6N+15} \\ \mathbf{J} \end{bmatrix} \mathbf{P}_{k|k} \begin{bmatrix} \mathbf{I}_{6N+15} \\ \mathbf{J} \end{bmatrix}^T \quad (23)$$

where the Jacobian \mathbf{J} is derived from (21)–(22) as

$$\mathbf{J} = \begin{bmatrix} \mathbf{C}(\mathbf{C}_B^B \bar{\mathbf{q}}) & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 3} \\ [\mathbf{C}_{\hat{\mathbf{q}}}^T \mathbf{p}_C \times] & \mathbf{0}_{3 \times 9} & \mathbf{I}_3 \end{bmatrix}. \quad (24)$$

D. Measurement Model for ML Observations

We now describe the EKF measurement model for treating visual observations of MLs. Consider that feature j , whose global coordinates are known *a priori*, is observed from the i th camera pose included in the EKF state vector. In normalized image coordinates, this observation is described by the equation

$$\mathbf{z}_i^{(j)} = \frac{1}{c_i z_j} \begin{bmatrix} c_i x_j \\ c_i y_j \end{bmatrix} + \mathbf{n}_i^{(j)} \quad (25)$$

where $\mathbf{n}_i^{(j)}$ is the 2×1 image noise vector, with covariance matrix $\mathbf{R}_i^{(j)} = \sigma_{\text{im}}^2 \mathbf{I}_2$. The feature position expressed in the camera frame ${}^{C_i} \mathbf{p}_{\ell_j}$ is given by

$${}^{C_i} \mathbf{p}_{\ell_j} = \begin{bmatrix} c_i x_j \\ c_i y_j \\ c_i z_j \end{bmatrix} = \mathbf{C}(\mathbf{C}_G^C \hat{\mathbf{q}}) ({}^G \mathbf{p}_{\ell_j} - {}^G \mathbf{p}_{C_i}) \quad (26)$$

where ${}^G \mathbf{p}_{\ell_j}$ is the *known* position of the landmark in the global frame. The expected value of the measurement $\mathbf{z}_i^{(j)}$ is computed

using the state estimates:

$$\hat{\mathbf{z}}_i^{(j)} = \frac{1}{c_i \hat{z}_j} \begin{bmatrix} c_i \hat{x}_j \\ c_i \hat{y}_j \end{bmatrix} \text{ with } \begin{bmatrix} c_i \hat{x}_j \\ c_i \hat{y}_j \\ c_i \hat{z}_j \end{bmatrix} = \mathbf{C}(\mathbf{C}_G^C \hat{\mathbf{q}}) ({}^G \mathbf{p}_{\ell_j} - {}^G \hat{\mathbf{p}}_{C_i}). \quad (27)$$

From (25) and (27), we can compute the *residual* of this ML measurement, $\mathbf{r}_{\text{ML}_i}^{(j)} = \mathbf{z}_i^{(j)} - \hat{\mathbf{z}}_i^{(j)}$. By linearization of (25), $\mathbf{r}_{\text{ML}_i}^{(j)}$ is written as

$$\mathbf{r}_{\text{ML}_i}^{(j)} \simeq \mathbf{H}_{\delta\theta}^{(j)} \delta\theta_{C_i} + \mathbf{H}_{\mathbf{p}_i}^{(j)G} \tilde{\mathbf{p}}_{C_i} + \mathbf{n}_i^{(j)} = \mathbf{H}_{\text{ML}_i}^{(j)} \tilde{\mathbf{x}} + \mathbf{n}_i^{(j)} \quad (28)$$

where

$$\mathbf{H}_{\delta\theta}^{(j)} = \frac{1}{c_i \hat{z}_j} [\mathbf{I}_2 \quad -\hat{\mathbf{z}}_i^{(j)}] [\mathbf{C}(\mathbf{C}_G^C \hat{\mathbf{q}}) ({}^G \mathbf{p}_{\ell_j} - {}^G \hat{\mathbf{p}}_{C_i}) \times]$$

$$\mathbf{H}_{\mathbf{p}_i}^{(j)} = -\frac{1}{c_i \hat{z}_j} [\mathbf{I}_2 \quad -\hat{\mathbf{z}}_i^{(j)}] \mathbf{C}(\mathbf{C}_G^C \hat{\mathbf{q}})$$

$$\mathbf{H}_{\text{ML}_i}^{(j)} = \begin{bmatrix} \mathbf{0}_{3 \times 15} & \mathbf{0}_{3 \times 6} & \dots & \underbrace{[\mathbf{H}_{\delta\theta}^{(j)} \quad \mathbf{H}_{\mathbf{p}_i}^{(j)}]}_{i\text{-th camera block}} & \dots & \mathbf{0}_{3 \times 6} \end{bmatrix}. \quad (29)$$

The residual defined in (28) is employed for performing EKF updates, as described in Section V-F.

E. Measurement Model for OF Observations

We present the OF measurement model for the case of a *single* feature f_j that is observed from a set of M_j poses, \mathcal{S}_j . Each observation of this feature is described by the measurement model of (25). Since the global coordinates of f_j are *not* known in advance, in order to compute the expected value of the measurements, we obtain an estimate of the position of the observed feature, ${}^G \hat{\mathbf{p}}_{\ell_j}$, by employing a least-squares minimization algorithm (cf. Appendix). Once this estimate has been computed, the expected value of each of the feature measurements can be evaluated, similarly to (27), with the sole difference that the *estimate* of the landmark position is used, instead of an *a priori* known value.

Linearization yields the following expression for the residual, $\mathbf{r}_i^{(j)} = \mathbf{z}_i^{(j)} - \hat{\mathbf{z}}_i^{(j)}$, of the i th measurement

$$\begin{aligned} \mathbf{r}_i^{(j)} &\simeq \mathbf{H}_{\delta\theta}^{(j)} \delta\theta_{C_i} + \mathbf{H}_{\mathbf{p}_i}^{(j)G} \tilde{\mathbf{p}}_{C_i} + \mathbf{H}_{f_i}^{(j)G} \tilde{\mathbf{p}}_{\ell_j} + \mathbf{n}_i^{(j)} \\ &= \mathbf{H}_{\mathbf{x}_i}^{(j)} \tilde{\mathbf{x}} + \mathbf{H}_{f_i}^{(j)G} \tilde{\mathbf{p}}_{\ell_j} + \mathbf{n}_i^{(j)}. \end{aligned}$$

Note that in contrast to the case of ML observations, the measurement residual in this case is also affected by the error in the estimate of the landmark position ${}^G \tilde{\mathbf{p}}_{\ell_j}$. In the last expression,

$\mathbf{H}_{f_i}^{(j)} = -\mathbf{H}_{\mathbf{p}_i}^{(j)}$ is the Jacobian of the residual with respect to ${}^G \tilde{\mathbf{p}}_{\ell_j}$ [cf. (29)], and

$$\mathbf{H}_{\mathbf{x}_i}^{(j)} = \begin{bmatrix} \mathbf{0}_{3 \times 15} & \mathbf{0}_{3 \times 6} & \dots & \underbrace{[\mathbf{H}_{\delta\theta}^{(j)} \quad \mathbf{H}_{\mathbf{p}_i}^{(j)}]}_{i\text{-th camera block}} & \dots & \mathbf{0}_{3 \times 6} \end{bmatrix}.$$

By stacking the residuals corresponding to all the observations of this feature over a certain time interval, we obtain

$$\mathbf{r}^{(j)} \simeq \mathbf{H}_{\mathbf{x}}^{(j)} \tilde{\mathbf{x}} + \mathbf{H}_f^{(j)} G \tilde{\mathbf{p}}_{\ell_j} + \mathbf{n}^{(j)} \quad (30)$$

where $\mathbf{r}^{(j)}$, $\mathbf{H}_{\mathbf{x}}^{(j)}$, $\mathbf{H}_f^{(j)}$, and $\mathbf{n}^{(j)}$ are block vectors or matrices with elements $\mathbf{r}_i^{(j)}$, $\mathbf{H}_{\mathbf{x}_i}^{(j)}$, $\mathbf{H}_{f_i}^{(j)}$, and $\mathbf{n}_i^{(j)}$, for $i \in \mathcal{S}_j$. Assuming that the feature observation noise in different images is independent, the covariance matrix of the noise vector $\mathbf{n}^{(j)}$ is $\mathbf{R}^{(j)} = \sigma_{\text{im}}^2 \mathbf{I}_{2M_j}$.

It should be clear that the residual derived in (30) *cannot* be directly used for performing EKF updates, since the landmark position error $G \tilde{\mathbf{p}}_{\ell_j}$ is correlated with the state errors (recall that $G \tilde{\mathbf{p}}_{\ell_j}$ is computed using the state estimates and the measurements $\mathbf{z}_i^{(j)}$ in a least-squares minimization process). To overcome this problem, we define a residual $\mathbf{r}_{\text{OF}}^{(j)}$, by projecting $\mathbf{r}^{(j)}$ on the left nullspace of the matrix $\mathbf{H}_f^{(j)}$. Specifically, if we let \mathbf{U} denote the unitary matrix whose columns form the basis of the left nullspace of $\mathbf{H}_f^{(j)}$, we obtain

$$\begin{aligned} \mathbf{r}_{\text{OF}}^{(j)} &= \mathbf{U}^T (\mathbf{z}^{(j)} - \hat{\mathbf{z}}^{(j)}) \simeq \mathbf{U}^T \mathbf{H}_{\mathbf{x}}^{(j)} \tilde{\mathbf{x}} + \mathbf{U}^T \mathbf{n}^{(j)} \\ &= \mathbf{H}_{\text{OF}}^{(j)} \tilde{\mathbf{x}} + \mathbf{n}_{\text{OF}}^{(j)}. \end{aligned} \quad (31)$$

It is worth noting that $\mathbf{r}_{\text{OF}}^{(j)}$ and $\mathbf{H}_{\text{OF}}^{(j)}$ can be computed without explicitly evaluating \mathbf{U} . Instead, these projections of \mathbf{r} and $\mathbf{H}_{\mathbf{x}}^{(j)}$ on the nullspace of $\mathbf{H}_f^{(j)}$ can be computed very efficiently using Givens rotations [43]. The covariance matrix of the noise vector $\mathbf{n}_{\text{OF}}^{(j)}$ can be shown to be equal to $\sigma_{\text{im}}^2 \mathbf{I}_{2M_j-3}$.

The residual $\mathbf{r}_{\text{OF}}^{(j)}$ is independent of the errors in the feature coordinates, and thus EKF updates can be performed based on it. Equation (31) defines a linearized constraint between all the M_j camera poses from which the feature f_j was observed. This residual expresses all the available information that the measurements $\mathbf{z}_i^{(j)}$ provide for the M_j camera states, and thus the resulting EKF update is optimal, except for the inaccuracies caused by linearization.

F. EKF Updates

In the preceding sections, we presented the measurement models that we employ for treating ML and OF observations. Once all the ML and OF measurements that must be processed at a given time step are determined (as described in Algorithm 4), the corresponding residual vectors and measurement Jacobians [cf. (28) and (31)] are created. Stacking all these together yields the following residual vector:

$$\mathbf{r} = \mathbf{H} \tilde{\mathbf{x}} + \mathbf{n} \quad (32)$$

where \mathbf{r} is a block vector with elements $\mathbf{r}_{\text{ML}_i}^{(j)}$ and $\mathbf{r}_{\text{OF}}^{(j)}$, \mathbf{H} is a block matrix with elements $\mathbf{H}_{\text{ML}_i}^{(j)}$ and $\mathbf{H}_{\text{OF}}^{(j)}$, and \mathbf{n} is a noise vector of dimension L (equal to the length of \mathbf{r}), with covariance matrix $\mathbf{R} = \sigma_{\text{im}}^2 \mathbf{I}_L$. One important issue is that typically, the dimension of \mathbf{r} can be large, even larger than the dimension of the filter state vector. In that case, to reduce the computational complexity of the EKF update, we employ the QR decomposition of

the matrix \mathbf{H} [17]. Specifically, we denote this decomposition as

$$\mathbf{H} = [\mathbf{V}_1 \quad \mathbf{V}_2] \begin{bmatrix} \mathbf{T}_H \\ \mathbf{0} \end{bmatrix}$$

where \mathbf{V}_1 and \mathbf{V}_2 are unitary matrices whose columns form bases for the range and nullspace of \mathbf{H} , respectively, and \mathbf{T}_H is an upper triangular matrix. With this definition, (32) yields

$$\mathbf{r} = [\mathbf{V}_1 \quad \mathbf{V}_2] \begin{bmatrix} \mathbf{T}_H \\ \mathbf{0} \end{bmatrix} \tilde{\mathbf{x}} + \mathbf{n} \Rightarrow \quad (33)$$

$$\begin{bmatrix} \mathbf{V}_1^T \mathbf{r} \\ \mathbf{V}_2^T \mathbf{r} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_H \\ \mathbf{0} \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} \mathbf{V}_1^T \mathbf{n} \\ \mathbf{V}_2^T \mathbf{n} \end{bmatrix}. \quad (34)$$

From the last equation, it becomes clear that by projecting the residual \mathbf{r} on the basis vectors of the range of \mathbf{H} , we retain all the useful information in the measurements. The residual $\mathbf{V}_2^T \mathbf{r}$ is only noise and can be completely discarded. For this reason, instead of the residual shown in (32), we employ the following residual for the EKF update:

$$\mathbf{r}_n = \mathbf{V}_1^T \mathbf{r} = \mathbf{T}_H \tilde{\mathbf{x}} + \mathbf{n}_n. \quad (35)$$

In this expression, $\mathbf{n}_n = \mathbf{V}_1^T \mathbf{n}$ is a noise vector whose covariance matrix is equal to $\mathbf{R}_n = \mathbf{V}_1^T \mathbf{R} \mathbf{V}_1 = \sigma_{\text{im}}^2 \mathbf{I}_r$, with $r < 6N$ being the number of columns in \mathbf{V}_1 . Using the previously defined residual vector, the EKF update proceeds according to the standard equations [44]. At the end of the update step, the oldest camera pose is marginalized out of the EKF state, to allow for the inclusion of the next one.

It is interesting to examine the computational complexity of the operations needed during the EKF update. In particular, when the number of measurements is large, i.e., $L \gg N$, evaluating \mathbf{r}_n and \mathbf{T}_H are the most computationally demanding operations, which determine the algorithm's complexity. Clearly, constructing the residual \mathbf{r} and the matrix \mathbf{H} has computational complexity linear in the number of measurements. Moreover, the residual \mathbf{r}_n , as well as the matrix \mathbf{T}_H , defined earlier, can be computed using Givens rotations in $O(r^2 L)$ operations, without the need to explicitly form \mathbf{V}_1 [43]. Thus, the computational complexity of the EKF updates is linear in the number of feature measurements, which is important for attaining real-time estimator performance.

VI. EXPERIMENTS

In order to validate the algorithm's performance in conditions as close to actual planetary landing as possible, a sounding rocket experiment was conducted in April 2006 at the White Sands Missile Range (WSMR), New Mexico.

A. Hardware Description

A commercially available analog camera (Pulnix TM-9701) was added to an existing mission payload consisting of a GLN-MAC IMU and a Thales G12 GPS, onboard a Terrier Orion Sounding Rocket (cf. Fig. 1 for the experimental setup). The nadir-pointing camera provided descent imagery from parachute deployment to landing at 30 frames/s with a resolution of

TABLE I
COMPARISON BETWEEN SOUNDING ROCKET AND MARS EDL

Parameter	Sounding Rocket	Mars Landing
Parachute Deploy Alt.	4200 m above ground	2000 m above ground
Vertical Velocity	10 m/s at touchdown	1 m/s at touchdown
Horizontal Velocity	3 m/s at touchdown	< 0.5 m/s at touchdown
Off-nadir angle	$\leq 12^\circ$	$< 20^\circ$
Off-nadir angular rate	$\leq 19^\circ/\text{s}$	$< 60^\circ/\text{s}$
Roll Rate	$\leq 360^\circ/\text{s}$	$< 60^\circ/\text{s}$

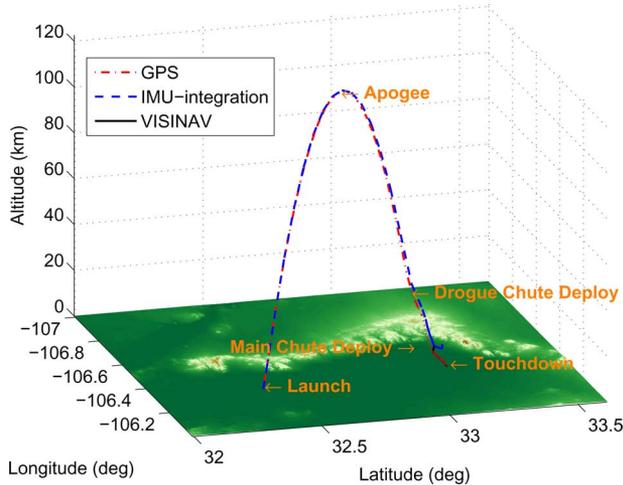


Fig. 4. 3-D trajectory of sounding rocket superimposed on terrain map.

768 × 484 pixels, 8 bits/pixel, and a field of view (FOV) of 38° × 24°. A common GPS time tag produced using a commercial timecode generator was used to synchronize 50 Hz IMU, 10 Hz GPS, and 30 Hz image data. Images, IMU data, and GPS measurements were downlinked in real time during flight over an S-band telemetry channel and recorded on the ground.

The data collected during this experiment were processed off-line. We should note, however, that our algorithm is capable of real-time operation. The FFT correlation-based feature matching is predicted to run at 5–20 Hz in an FPGA-based implementation currently under development at JPL,⁷ and the current C++ implementation of the pose estimator runs at more than 30 Hz on a 2 GHz CPU, with the number of stored poses set to $N = 20$.

B. Experiment Profile

The rocket reached an apogee altitude of 123 km, followed by drogue and main parachute opening at 28 km and 4.2 km altitude, respectively. After a total flight time of 805 s, 376 s of which on the parachute, the vehicle landed 78 km downrange from the launch pad. The dynamics encountered during the parachuted phase of the sounding rocket flight are compared to those during an actual Mars landing in Table I.

Fig. 4 shows the rocket's trajectory superimposed on a 3-D map of the area. A zoomed-in view of the flight path after the

⁷Note that although images were available at 30 Hz, in this experiment images were only processed at most at 3 Hz.

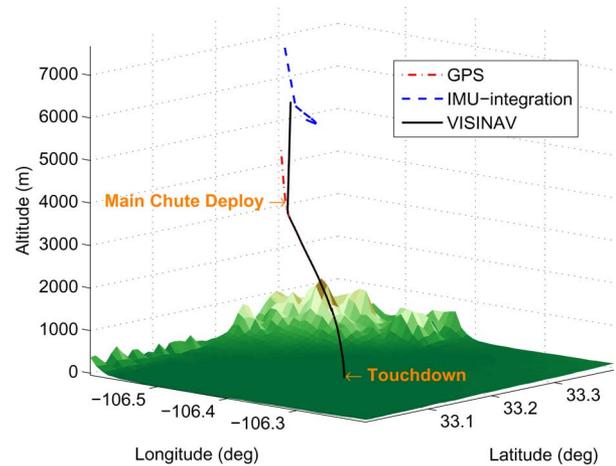


Fig. 5. Zoomed-in view of trajectory after main parachute deployment.

TABLE II
CONDITIONS FOR THE DIFFERENT EKF UPDATE PHASES

	Altitude (m)	Time (s)	Position	Velocity
			Error (m)	Error (m/s)
Beg. 1st ML set	3800	454	2722.3	10.45
End 1st ML set	3100	514	16.9	0.18
Beg. 2nd ML set	1600	647	78.2	1.38
End 2nd ML set	230	781	5.1	0.23
Beg. OFs	330	771	3.7	0.15
Touchdown	0	805	6.4	0.16
Touchdown (IMU-only)	0	805	9169.5	32.70

main parachute's deployment is depicted in Fig. 5. Pure integration of the IMU measurements (blue dashed line) yielded fairly accurate results until right after the deployment of the main parachute, but then quickly diverged. The reason is that the opening of the parachute caused the rocket's motion to be extremely jerky for several seconds. Integrating the large acceleration measurements recorded in this period, using the attitude estimates that had error accumulated over the preceding 431 s of flight, resulted in large position errors. Note that up to this point no images were available. Once the first few images are processed, the VISINAV algorithm corrects the accumulated error, and the estimated trajectory becomes virtually indistinguishable from that measured by the GPS.

As shown in Table II, ML measurements were processed during two separate phases of flight, one between 3800 m and 3100 m, and the other between 1600 m and 230 m above ground. This intermediate period of pure open-loop IMU integration without camera updates was intentionally introduced to test the image processing algorithm's ability to redetect MLs. In an actual EDL scenario, such a profile could arise when correcting a large position deviation during powered descent. In particular, if a significant offset between desired and actual position is determined during the first phase using MLs, the ensuing correction maneuver might cause large off-nadir viewing angles

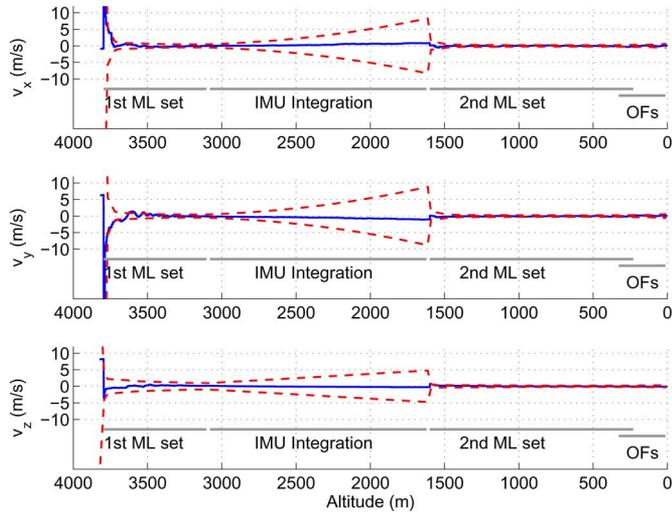


Fig. 6. Velocity error expressed in NED frame (solid blue lines) and corresponding 3σ bounds (dashed red lines). Note that x - y - z in the plots corresponds to N-E-D.

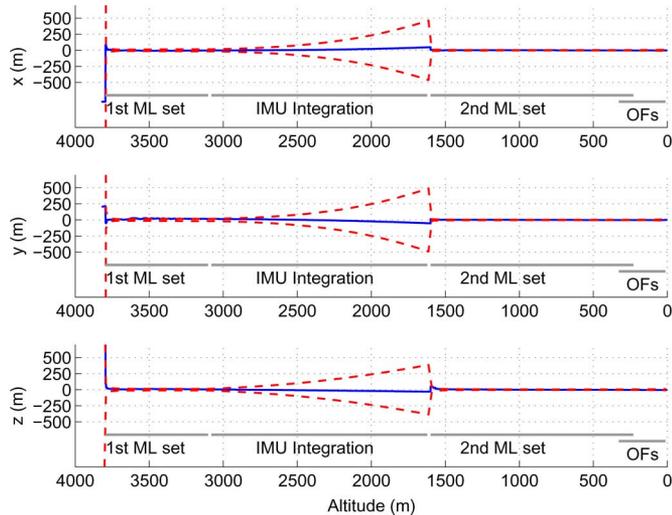


Fig. 7. Position error expressed in NED frame (solid blue lines) and corresponding 3σ bounds (dashed red lines).

that could preclude ML detection. After the maneuver, MLs can again be detected and used for pose estimation until shortly before touchdown. In the sounding rocket experiment, the image processing algorithm was able to successfully reacquire MLs after the open-loop phase, and the filter recovered an accurate pose estimate within a few seconds after resuming ML updates (cf. Figs. 6 and 7 at 1600 m altitude).

In the first phase of ML updates, approximately 40 MLs per image were identified after processing one out of every ten image frames (3 Hz). In the second phase, only one frame per second was processed, and approximately 80 features/image were detected, as is shown in Fig. 8. We note that the template size used for FFT map matching was 64×64 pixels, while for ML matching we have employed 15×15 pixel templates.

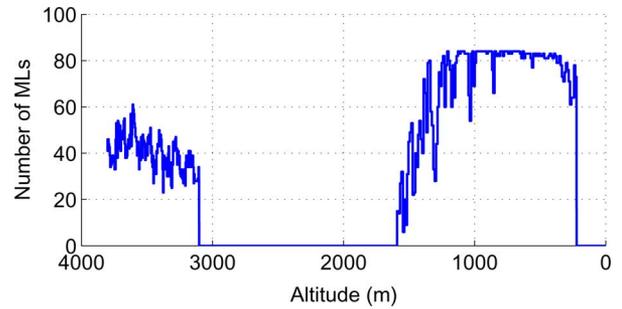


Fig. 8. Number of detected MLs vs. altitude.

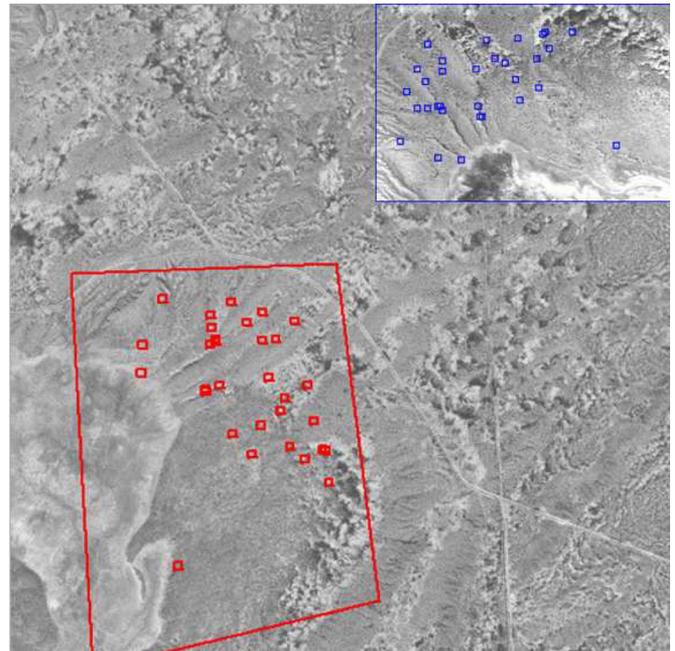


Fig. 9. ML matching between map (large) and descent camera image (upper right).

The 3-D ground coordinates for the MLs were obtained from USGS 1-m digital orthoimagery quarter quadrangles taken in 2001, combined with 1 arcsecond finished elevation data from the Shuttle Radar Topography Mission [45]. For feature matching in the first set, the entire 7×8 km map image was resampled to approximately 7 m/pixel, while for the second set the map was cropped to 2×2 km and used at its base resolution of 1 m/pixel. These choices resulted in a map of dimensions 952×822 pixels for the first set, and 2048×2048 pixels for the second. Fig. 9 shows an example of matched MLs between the map and a camera image obtained at an altitude of approximately 1600 m.

At some point during descent, the number of features within the FOV becomes too small, and the difference in resolution between camera and map too significant to allow successful ML correspondences to be established. In this experiment, we emulated this behavior by stopping ML updates at 230 m altitude. To compensate, starting at 330 m, the filter began to perform OF updates at a frequency of 3 Hz. The size of the image templates

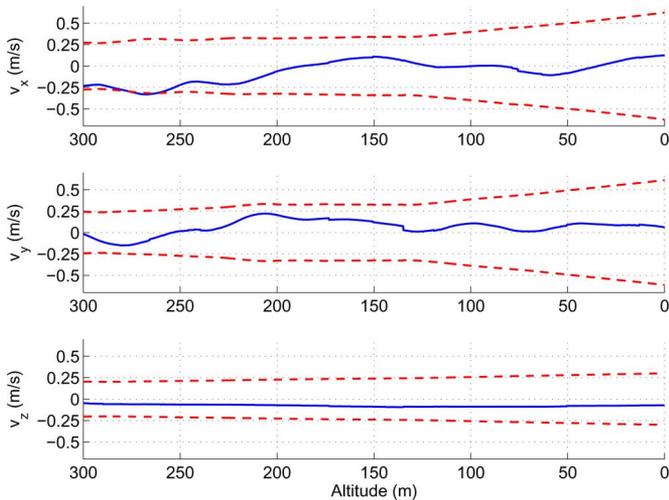


Fig. 10. Velocity error in NED frame (zoomed-in view of Fig. 6 before touchdown).

used for OF tracking was 11×11 pixels, sampled at the original image resolution. Even though OFs do not make the camera pose observable (as opposed to images containing at least three MLs), they allow for precise estimation of linear and rotational velocity, resulting in very small error growth during the final 230 m of descent.

C. Algorithm Performance

Ground truth for position and velocity was obtained from GPS measurements. Figs. 6 and 7 show the resulting errors and the corresponding 3σ bounds for velocity and position in the local north-east-down (NED) frame. Table II gives the error norms for the position and velocity estimates at the beginning and end of the different update phases. We note that no ground truth for orientation was available for this experiment.

1) *First ML Set*: When the first MLs become available, the algorithm converges in about 5 s, from the large error (~ 2700 m) accumulated during IMU integration, to within 18 m of GPS ground truth (cf. Figs. 5 and 7). During the open-loop integration phase between 3100 and 1600 m altitude, the pose uncertainty is again increasing.

2) *Second ML Set and OFs*: The algorithm almost instantaneously recovers accuracy once resuming ML updates at 1600 m. ML and OF updates (the latter starting at 330 m) reduce the position uncertainty bounds to approximately ± 4 m, and the velocity uncertainty to less than ± 0.25 m/s along each axis (3σ). Notice that the actual error at the beginning of OF updates is smaller than at the end of ML processing, 10 s later. This can be attributed to the already decreasing number of detected MLs at this altitude, due to the difference in resolution between the satellite and camera images (cf. Fig. 8). With the discontinuation of ML processing at 230 m altitude, the pose uncertainty continues to increase, although still at a very low rate (cf. the zoomed-in view of the errors for the final 300 m before touchdown in Figs. 10 and 11). As predicted, this is the result

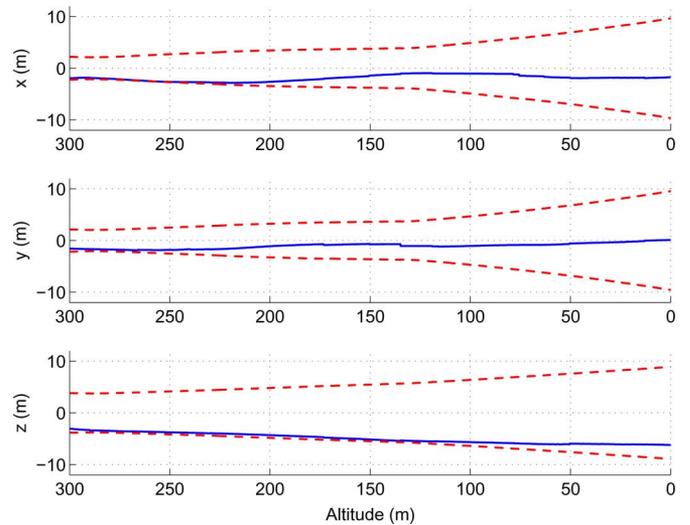


Fig. 11. Position error in NED frame (zoomed-in view of Fig. 7 before touchdown).

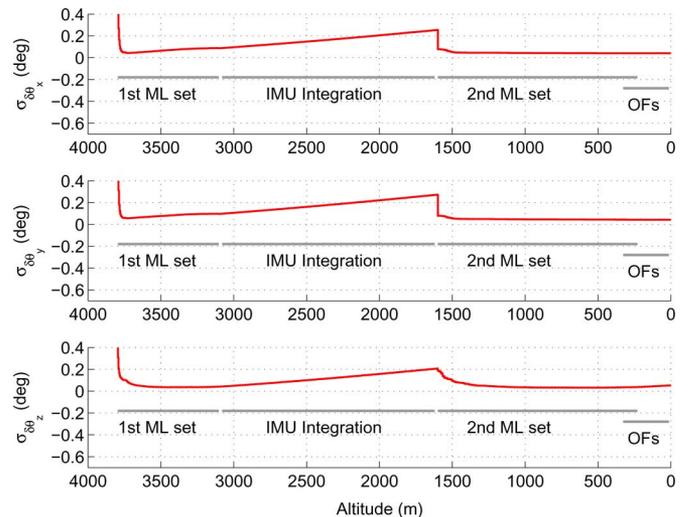


Fig. 12. Attitude uncertainty.

of the system becoming unobservable. Table II shows the final velocity and position errors' magnitude at touchdown, which are approximately 6.4 m in position and 0.16 m/s for velocity.

Similar to position and velocity, the attitude uncertainty bounds were decreased to $\pm 0.15^\circ$ accuracy along each axis (3σ) when processing ML measurements, with a temporary increase to $\pm 0.9^\circ$ during open-loop IMU integration between 3100 m and 1600 m altitude (cf. Fig. 12). Note that due to the lack of ground truth, the actual values of the attitude errors are unknown. However, the figures for the position and velocity errors (cf. Figs. 6, 7, 10, and 11) show that the filter is consistent, which indicates that the attitude estimates are also correct. The filter attitude estimate was further verified through an independent measurement of the final attitude at touchdown using a compass.

TABLE III
PERFORMANCE OF FFT MATCHING

Map size, Res. (pix., m)	Template (pix.)	% valid	RMS Pos. error (m)	Mean time (s)	FPGA time (s)
3000×840, 3.8	112x112	99.2	0.88	7.32	0.17
1500×420, 7.5	56x56	99.2	1.53	1.72	0.04
750×210, 15	28x28	87.2	2.75	0.49	0.01

VII. SENSITIVITY STUDIES

The sounding rocket test, presented in the previous section, was important for testing our algorithms during an EDL experiment using real sensor data. However, for validating the VISINAV system's performance under various operational conditions, thorough testing of all of its components is required. In [19] and [23], extensive results on the performance of the EKF-based estimator are presented. In this section, we outline results regarding the performance of the FFT map-matching module, which is critical for acquiring good initial pose estimates in the VISINAV system. Specifically, the sensitivity of the algorithm to planetary terrain appearance, illumination, terrain changes, and altitude errors is investigated.

A. Planetary Terrain Appearance

A simulation tool for investigating the FFT map-matching algorithm's performance has been created by extending an existing descent image simulator [46]. This simulator uses real Mars orbital imagery, a high-fidelity model of the descent camera's optical and noise characteristics, and descent trajectories from a simulation of EDL dynamics to produce synthetic imagery.

Monte-Carlo trials for matching synthetic descent images to orbital images were performed, to determine primarily the matching precision and the percentage of invalid matches during FFT map matching. For these tests, 850 images from the Meridiani Planum and Gusev crater were used, and the results are reported in Table III. The columns of this table describe 1) the size and resolution of the map; 2) the size of the template used for FFT matching; 3) the percentage of correct matches; 4) the rms map-matching localization error; 5) the mean processing time on a 400 MHz general-purpose processor, and 6) the estimated runtime of the FFT matching algorithm on an FPGA implementation. These results demonstrate precision and reliability rates consistent with the requirements of a precision landing system. Moreover, since the image processing algorithms will ultimately be implemented on an FPGA, it becomes clear that the VISINAV system can meet the runtime constraints of a real EDL mission.

B. Illumination Variation

Illumination insensitivity is achieved by local image normalization, which drives pixel intensities toward zero mean and uniform variance. With this simple and highly efficient scheme,

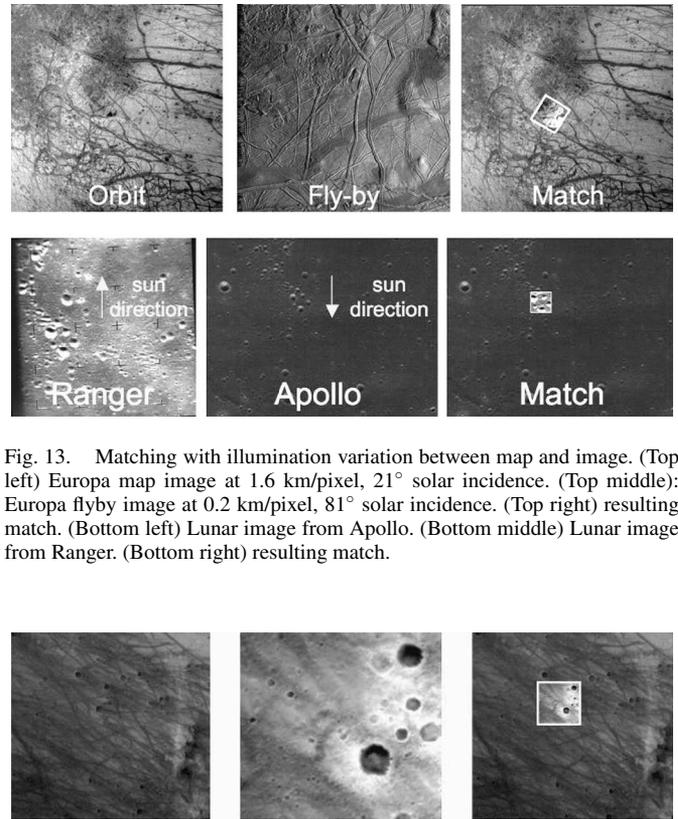


Fig. 13. Matching with illumination variation between map and image. (Top left) Europa map image at 1.6 km/pixel, 21° solar incidence. (Top middle): Europa flyby image at 0.2 km/pixel, 81° solar incidence. (Top right) resulting match. (Bottom left) Lunar image from Apollo. (Bottom middle) Lunar image from Ranger. (Bottom right) resulting match.

Fig. 14. Matching after appearance change. (Left) THEMIS image from Mars odyssey. (Middle) Descent image from MER lander taken more than two years later. (Right) Resulting match.

we achieve excellent results on test cases using real planetary imagery. The top row of Fig. 13 shows two images of Jupiter's moon Europa acquired by the solid-state imager onboard the Galileo spacecraft. The first was taken from a distance during initial approach and the second during a closer flyby. The solar incidence angles in the two cases differ by 60°, resulting in an obvious appearance change. Nevertheless, the map-matching algorithm is able to successfully determine the location of the image patch in the map. The bottom row of Fig. 13 shows two images of the Moon acquired by an Apollo and a Ranger mission, respectively. In the two cases, the sun azimuth angles are approximately 180° apart, and there is also some perspective distortion between the images. Despite the substantial appearance change, the map-matching algorithm is successful. This is due to the large template size, which provides enough support for normalized correlation to compensate for the illumination variations in the two images.

C. Terrain Appearance Variation

In Fig. 14, we show an example that demonstrates the insensitivity of FFT matching to terrain appearance changes. The map image was taken by the visible camera on the thermal emission imaging system (THEMIS) onboard the Mars Odyssey orbiter on Feb. 25, 2002. The descent image was taken on January

4, 2004 by the DIMES camera on the MER-Spirit lander. The streaks in the images are dust devil tracks that changed in the intervening two years. Despite this appearance change, the match is still successful. Finally, we should note that the map image used during the sounding rocket test (cf. Section VI) was taken 5 years before the test. Since the WSMR is an active test range, there were appearance differences between the map and the descent images due to human activity, vegetation growth, and the effects of weather. However, for the hundreds of images processed, this did not have a notable impact on map-matching performance.

D. Altitude Measurement Accuracy

To test sensitivity to errors in template (descent image) rescaling resulting from inaccurate altitude estimates, we compared a typical THEMIS map image at 19 m/pixel resolution with a “descent” image at approximately 3 m/pixel resolution from the Mars orbiter camera (MOC) on the Mars global surveyor spacecraft. The MOC image was rescaled to between 80% and 120% of the correct factor, corrupted with a small amount of Gaussian white noise with standard deviation equal to 0.5% of the mean image brightness, and compared to the map. We performed 50 iterations at each rescaling factor, and the study showed that just over $\pm 5\%$ error in scale is well-tolerated by the map-matching algorithm. With larger errors, match success drops off abruptly. This translates directly to a 5% accuracy requirement in the altitude estimate, which is well within existing mission capabilities.

VIII. CONCLUSION

In this paper, we have presented the analysis and experimental validation of a vision-aided inertial navigation system for planetary landing applications. The VISINAV system is based on tight integration of inertial and visual information, and is capable of providing very accurate estimates of the lander’s terrain-relative position, attitude, and velocity in real time. These estimates can be directly applied for precision guidance and control during landing. The system consists of a visual front-end, which processes the camera images to extract ML and OF measurements, and of an EKF that optimally fuses all available positioning information. Results from a sounding rocket test showed estimation errors of magnitude 0.16 m/s in velocity and 6.4 m in position at touchdown. These results vastly improve current state of the art for non vision-based EDL, and meet the requirements of future planetary exploration missions [1].

Sensitivity studies were presented in this paper, which validated the system’s robustness to changes in surface appearance, as well as its ability to operate with several different types of planetary terrains. In our future research, we intend to further probe the VISINAV system in a series of test flights. The immediate next step is to implement the vision and estimation algorithms on flight-qualifiable CPUs and FPGAs. Moreover, we intend to pursue several interesting research topics that arise in relation to the problem at hand. For example, the selection of an optimal subset of the most informative image features

to track is of great interest. That would allow the system to obtain high-precision pose estimates even under the most stringent real-time constraints, by adaptively managing its computational resources to focus on a few “good” features.

APPENDIX

To compute an estimate of the position of a tracked feature f_j , we employ intersection [47]. To avoid local minima, and for better numerical stability, during this process we use an inverse-depth parametrization of the feature position [48]. In particular, if $\{C_n\}$ is the camera frame in which the feature was observed for the first time, then the feature coordinates with respect to the camera at the i th time instant are

$${}^i C_i \mathbf{p}_{\ell_j} = \mathbf{C}({}_{C_n}^{C_i} \bar{\mathbf{q}}) {}^{C_n} \mathbf{p}_{\ell_j} + {}^{C_i} \mathbf{p}_{C_n}, \quad i \in \mathcal{S}_j. \quad (36)$$

In this expression, $\mathbf{C}({}_{C_n}^{C_i} \bar{\mathbf{q}})$ and ${}^{C_i} \mathbf{p}_{C_n}$ are the rotation and translation between the camera frames at time instants n and i , respectively. Equation 36 can be rewritten as

$${}^i C_i \mathbf{p}_{\ell_j} = {}^{C_n} z_j \left(\mathbf{C}({}_{C_n}^{C_i} \bar{\mathbf{q}}) \begin{bmatrix} \frac{{}^{C_n} x_j}{{}^{C_n} z_j} \\ \frac{{}^{C_n} y_j}{{}^{C_n} z_j} \\ 1 \end{bmatrix} + \frac{1}{{}^{C_n} z_j} {}^{C_i} \mathbf{p}_{C_n} \right) \quad (37)$$

$$= {}^{C_n} z_j \left(\mathbf{C}({}_{C_n}^{C_i} \bar{\mathbf{q}}) \begin{bmatrix} \alpha_j \\ \beta_j \\ 1 \end{bmatrix} + \rho_j {}^{C_i} \mathbf{p}_{C_n} \right) \quad (38)$$

$$= {}^{C_n} z_j \begin{bmatrix} h_{i1}(\alpha_j, \beta_j, \rho_j) \\ h_{i2}(\alpha_j, \beta_j, \rho_j) \\ h_{i3}(\alpha_j, \beta_j, \rho_j) \end{bmatrix}. \quad (39)$$

In the last expression, h_{i1} , h_{i2} , and h_{i3} are scalar functions of the quantities $\alpha_j, \beta_j, \rho_j$, which are defined as

$$\alpha_j = \frac{{}^{C_n} x_j}{{}^{C_n} z_j}, \quad \beta_j = \frac{{}^{C_n} y_j}{{}^{C_n} z_j}, \quad \rho_j = \frac{1}{{}^{C_n} z_j}. \quad (40)$$

Substituting from (39) into (25), we can express the measurement equations as functions of α_j, β_j , and ρ_j only

$$\mathbf{z}_i^{(j)} = \frac{1}{h_{i3}(\alpha_j, \beta_j, \rho_j)} \begin{bmatrix} h_{i1}(\alpha_j, \beta_j, \rho_j) \\ h_{i2}(\alpha_j, \beta_j, \rho_j) \end{bmatrix} + \mathbf{n}_i^{(j)}. \quad (41)$$

Given the measurements $\mathbf{z}_i^{(j)}$, $i \in \mathcal{S}_j$, and the estimates for the camera poses in the state vector, we can obtain estimates for $\hat{\alpha}_j, \hat{\beta}_j$, and $\hat{\rho}_j$, using Gauss–Newton least-squares minimization. Then, the global feature position is computed as

$${}^G \hat{\mathbf{p}}_{\ell_j} = \frac{1}{\hat{\rho}_j} \mathbf{C}^T({}_{C_n}^{C_i} \hat{\mathbf{q}}) \begin{bmatrix} \hat{\alpha}_j \\ \hat{\beta}_j \\ 1 \end{bmatrix} + {}^G \hat{\mathbf{p}}_{C_n}. \quad (42)$$

We note that during the least-squares minimization process, the camera pose estimates are treated as known constants, and their covariance matrix is ignored. As a result, the minimization can be carried out very efficiently, at the expense of the optimality

of the feature position estimates. Recall, however, that up to a first-order approximation, the errors in these estimates do not affect the measurement residual [cf. (31)]. Thus, no significant degradation of performance is inflicted.

REFERENCES

- [1] NASA. (2006, Sep.) "Solar system exploration roadmap," [Online]. Available: http://solarsystem.nasa.gov/multimedia/downloads/SSE_RoadMap_2006_Report_FC-A_med.pdf
- [2] A. E. Johnson, A. Ansar, L. H. Matthies, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "A general approach to terrain relative navigation for planetary landing," presented at the AIAA Aerospace@Infotech Conf., Rohnert Park, CA, May 2007.
- [3] N. Trawny, A. Mourikis, S. Roumeliotis, A. Johnson, J. Montgomery, A. Ansar, and L. Matthies, "Coupled vision and inertial navigation for pin-point landing," presented at the NASA Sci. Technol. Conf. (NSTC 2007), College Park, MD, Jun. 19–21.
- [4] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, and L. Matthies, "Vision-aided inertial navigation for precise planetary landing: Analysis and experiments," in *Proc. Robot. Sci. Syst.*, Atlanta, GA, Jun. 26–30, 2007, pp. 158–165.
- [5] Y. Cheng, A. E. Johnson, C. F. Olson, and L. H. Matthies, "Optical landmark detection for spacecraft navigation," presented at the AAS/AIAA Space Flight Mech. Meeting, Ponce, Puerto Rico, Feb. 2003.
- [6] Y. Cheng and A. Ansar, "Landmark based position estimation for pinpoint landing on Mars," in *Proc. 2005 IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 4470–4475.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 2, no. 60, pp. 91–110, 2004.
- [8] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, 1988, pp. 147–151.
- [9] J. P. Lewis, "Fast template matching," in *Vis. Interface*, Quebec City, CA, May 15–19, 1995, pp. 120–123.
- [10] A. Johnson, R. Willson, Y. Cheng, J. Goguen, C. Leger, M. SanMartin, and L. Matthies, "Design through operation of an image-based velocity estimation system for Mars landing," *Spec. Joint Issue Int. J. Comput. Vis. Int. J. Robot. Res. Vis. Robot.*, vol. 74, no. 3, pp. 319–341, 2007.
- [11] D. Mortari, J. Rojas, and J. Junkins, "Attitude and position estimation from vector observations," presented at the AAS/AIAA Space Flight Mech. Meeting, Maui, HI, Feb. 2004.
- [12] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 578–589, May 2003.
- [13] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Noelle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int. J. Comput. Vis.*, vol. 13, no. 3, pp. 331–356, Dec. 1994.
- [14] J. L. Crassidis, R. Alonso, and J. L. Junkins, "Optimal attitude and position determination from line-of-sight measurements," *J. Astronaut. Sci.*, vol. 48, no. 2–3, pp. 391–408, Apr./Sep. 2000.
- [15] A. Wu, E. Johnson, and A. Proctor, "Vision-aided inertial navigation for flight control," presented at the AIAA Guid., Navigat. Control Conf., San Francisco, CA, Aug. 2005, Paper AIAA 2005-5998.
- [16] G. F. Ivey and E. Johnson, "Investigation of methods for simultaneous localization and mapping using vision sensors," presented at the AIAA Guid. Navigat. Control Conf., Keystone, CO, Aug. 2006, Paper AIAA 2006-6578.
- [17] D. S. Bayard and P. B. Brugarolas, "An estimation algorithm for vision-based exploration of small bodies in space," in *Proc. 2005 Amer. Control Conf.*, Portland, OR, Jun. 2005, vol. 7, pp. 4589–4595.
- [18] S. Li, P. Cui, and H. Cui, "Vision-aided inertial navigation for pinpoint planetary landing," *Aerosp. Sci. Technol.*, vol. 11, no. 6, pp. 499–506, Sep. 2007.
- [19] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, and J. Montgomery, "Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks," *J. Field Robot.*, vol. 24, no. 5, pp. 357–378, May 2007.
- [20] D. Strelow, "Motion estimation from image and inertial measurements," Ph.D. dissertation, Carnegie Mellon Univ., Nov. 2004.
- [21] J. W. Langelaan, "State estimation for autonomous flight in cluttered environments," Ph.D. dissertation, Dept. Aeronaut. Astronaut., Stanford Univ., 2006.
- [22] J. Kim and S. Sukkarieh, "Autonomous airborne navigation in unknown terrain environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 1031–1045, Jul. 2004.
- [23] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Rome, Italy, Apr. 2007, pp. 3565–3572.
- [24] S. Roumeliotis, A. Johnson, and J. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Washington, DC, 2002, pp. 4326–4333.
- [25] D. Diel, P. DeBitetto, and S. Teller, "Epipolar constraints for vision-aided inertial navigation," in *Proc. IEEE Workshop Motion Video Comput.*, Jan. 2005, pp. 221–228.
- [26] R. Garcia, J. Puig, O. Ridao, and X. Cufi, "Augmented state Kalman filtering for AUV navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Washington, DC, May 2002, pp. 4010–4015.
- [27] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1100–1114, Dec. 2006.
- [28] A. I. Mourikis and S. I. Roumeliotis, "On the treatment of relative-pose measurements for mobile robot localization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Orlando, FL, May 15–19, 2006, pp. 2277–2284.
- [29] P. McLauchlan, "The variable state dimension filter," Centre Vis., Speech Signal Process., Univ. Surrey, UK, Tech. Rep., 1999.
- [30] M. C. Deans, "Maximally informative statistics for localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Washington, DC, May 2002, pp. 1824–1829.
- [31] B. Frapard, B. Polle, G. Flandin, P. Bernard, C. Vetel, X. Sembely, and S. Mancuso, "Navigation for planetary approach and landing," in *Proc. 5th Int. ESA Conf. Spacecraft Guid., Navigat. Control Syst.*, Frascati, Italy, Oct. 22–25, 2002, pp. 159–168.
- [32] K. Janschek, V. Tchernykh, and M. Beck, "Performance analysis for visual planetary landing navigation using optical flow and DEM matching," presented at the AIAA Guid. Navigat. Control Conf., Keystone, CO, Aug. 2006, Paper AIAA 2006-6706.
- [33] B. G. Williams, "Technical challenges and results for navigation of NEAR Shoemaker," *Johns Hopkins APL Tech. Dig.*, vol. 23, no. 1, pp. 34–45, Jan./Mar. 2002.
- [34] D. W. Way, R. W. Powell, A. Chen, A. D. Steltzner, A. M. S. Martin, P. D. Burkhart, and G. F. Mendek, "Mars Science Laboratory: Entry, descent, and landing system performance," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, Mar. 3–10, 2007, pp. 1–19.
- [35] P. Heckbert, "Fundamentals of texture mapping and image warping," Master's thesis, Univ. California, Berkeley, Jun. 1989.
- [36] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [37] W. Pratt, *Digital Image Processing*. New York: Wiley, 1978.
- [38] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 1981, pp. 674–679.
- [39] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR 1994)*, pp. 593–600.
- [40] S. Baker and J. Matthews, "Lucas–Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, pp. 221–255, Mar. 2004.
- [41] W. G. Breckenridge, "Quaternions proposed standard conventions," Jet Propulsion Lab., Pasadena, CA, Interoffice Memorandum (IOM) 343-79-1199, 1999.
- [42] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Inst. Aeronautics and Astronautics, 1997.
- [43] G. Golub and C. van Loan, *Matrix Computations*. London, U.K.: The Johns Hopkins Univ. Press, 1996.
- [44] P. S. Maybeck, *Stochastic Models, Estimation and Control*. vol. 1–2, New York: Academic, 1979.
- [45] U.S. Geological Survey. (2006). "Seamless data distribution system," [Online]. Available: <http://seamless.usgs.gov/index.asp>
- [46] R. Willson, A. Johnson, and J. Goguen, "MOC2DIMES: A camera simulator for the Mars exploration rover descent image motion estimation system," in *Proc. 8th Int. Symp. Artif. Intell., Robot. Autom. Space (i-SAIRAS 2005)*, Munich, Germany, Sep., pp. 102–109.
- [47] B. Triggs, P. McLauchlan, R. Hartley, and Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice*, New York: Springer-Verlag, 2000, pp. 298–375.
- [48] J. Montiel, J. Civera, and A. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proc. Robot. Sci. Syst.*, Cambridge, MA, Jun. 2006, pp. 81–88.



Anastasios I. Mourikis received the Diploma in electrical engineering from the University of Patras, Patras, Greece, in 2003, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, in 2008.

He is currently an Assistant Professor in the Department of Electrical Engineering at the University of California, Riverside. His current research interests include autonomous vehicle navigation, multirobot systems, distributed estimation in robot networks, simultaneous localization and mapping, and structure

from motion.

Dr. Mourikis is a member of the IEEE Robotics and Automation Society and the Technical Chamber of Greece. He has served as a Reviewer for a number of technical journals and conferences. While at the University of Minnesota, he was awarded the University's 2007 Doctoral Dissertation Fellowship, as well as the Computer Science Department's 2005 and 2006 Excellence in Research Awards.



Nikolas Trawny received the DEA in control and automation from Institut Supérieur de l'Aéronautique et de l'Espace (formerly SUPAERO) Toulouse, France in 2003, and the Diploma in aerospace engineering from the University of Stuttgart, Stuttgart, Germany, in 2004. He is currently working toward the Ph.D. degree at the Department of Computer Science and Engineering, University of Minnesota, Minneapolis.

His current research interests include the areas of vision-aided inertial navigation and multirobot localization and mapping. He was the recipient of the 2006 Guidant Award from the Department of Computer Science and Engineering, University of Minnesota.



Stergios I. Roumeliotis received the Diploma in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1995, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1999 and 2000, respectively.

From 2000 to 2002, he was a Postdoctoral Fellow at the California Institute of Technology, Pasadena, CA. Between 2002 and 2008, he was an Assistant Professor, and currently he is an Associate Professor in the Department of Computer Science and Engineering

at the University of Minnesota, Minneapolis, MN. His current research interests include inertial navigation of aerial and ground autonomous vehicles, distributed estimation under communication and processing constraints, and active sensing for reconfigurable networks of mobile sensors.

Dr. Roumeliotis was the recipient of the National Science Foundation (NSF) Presidential Early Career Award for Scientists and Engineers (PECASE) in 2008, the NSF CAREER Award in 2006, the McKnight Land-Grant Professorship Award in 2006–2008, and the International Conference on Robotics and Automation (ICRA) Best Reviewer Award in 2006, and he is the co-recipient of the One National Aeronautics and Space Administration (NASA) Peer Award in 2006 and the One NASA Center Best Award in 2006. Papers he has coauthored have received the Robotics Society of Japan Best Journal Paper Award in 2007, the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Best Student Paper Award in 2006, the NASA Tech Briefs Award in 2004, and one of them was the Finalist for the International Conference on Intelligent Robots and Systems (IROS) Best Paper Award in 2006. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS.



Andrew E. Johnson received the B.S. degree (with Highest Distinction) in engineering physics and mathematics from the University of Kansas, Lawrence, in 1991, and the Ph.D. degree from the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, in 1997, where he developed the spin-image surface signature for 3-D object recognition and surface matching.

Currently, he is a Principal Member of the Technical Staff at NASA's Jet Propulsion Laboratory (JPL), Pasadena, CA, where he is engaged in developing image-based techniques for autonomous navigation and mapping during descent to planets, moons, comets, and asteroids. He is the Project Element Manager for the JPL component of the Autonomous Landing and Hazard Avoidance Technology Project that is developing safe and precise landing technologies for the next generation manned lunar lander. At JPL, he has worked on technology development tasks as well as flight projects. For the Mars Exploration Rover Project, he was the Lead Algorithm Developer for the Descent Image Motion Estimation Subsystem (DIMES), the first autonomous machine vision system used during planetary landing. Following the successful development and execution of DIMES, he is now moving back to the development of systems for landing hazard avoidance, pinpoint landing, and rover navigation.

Dr. Johnson was the recipient of the JPL Lew Allen Award in 2003 for Excellence for "his groundbreaking contributions in the area of machine vision algorithms for safe and precise landing."



Adnan Ansar received the B.A. degree in physics, the M.A. degree in mathematics, and the M.S. and Ph.D. degrees in computer science, in 1993, 1993, 1998, and 2001, respectively, all from the University of Pennsylvania, Philadelphia, with the last earned at the General Robotics, Automation, Sensing, and Perception (GRASP) Laboratory.

He has been a member of the Computer Vision Group at NASA's Jet Propulsion Laboratory (JPL), Pasadena, CA, since January 2002. While at JPL, his research has included work on image-based position estimation, camera calibration, stereo vision and multimodal data registration.



Larry Matthies received the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA, in 1989.

He then joined the Jet Propulsion Laboratory, Pasadena, CA, where he supervises the Computer Vision Group. He is an Adjunct Professor of Computer Science at the University of Southern California, Los Angeles. His current research interests include 3-D perception, state estimation, terrain classification, and dynamic scene analysis for autonomous navigation of unmanned vehicles on Earth and in space. His research on real-time stereo vision and accurate visual odometry in the early 1990s led to the incorporation of those capabilities in the twin Mars exploration rovers in 2003. His group also developed vision systems for estimating horizontal velocity of the Mars landers during descent and for detecting clouds and dust devils onboard the rovers. The focus of his National Aeronautics and Space Administration (NASA)-related research is now vision for safe and precise landing in planetary exploration. He has made numerous contributions to perception for unmanned ground vehicles in Defense Advanced Research Projects Agency (DARPA) and Army robotics projects since the early 1990s.

Prof. Matthies is a member of the editorial boards of the *Autonomous Robots Journal* and the *Journal of Field Robotics*. He was a joint winner in 2008 of the IEEE's Robotics and Automation Award for his contributions to robotic space exploration.