

# Vision-Aided Inertial Navigation for Precise Planetary Landing: Analysis and Experiments

Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis  
Dept. of Computer Science & Engineering,  
University of Minnesota, Minneapolis, MN 55455  
Email: {mourikis|trawny|stergios}@cs.umn.edu

Andrew Johnson and Larry Matthies  
Jet Propulsion Laboratory,  
Pasadena, CA 91125.  
Email: {aej|lhm}@jpl.nasa.gov

**Abstract**—In this paper, we present the analysis and experimental validation of a vision-aided inertial navigation algorithm for planetary landing applications. The system employs tight integration of inertial and visual feature measurements to compute accurate estimates of the lander’s terrain-relative position, attitude, and velocity in real time. Two types of features are considered: mapped landmarks, i.e., features whose global 3D positions can be determined from a surface map, and opportunistic features, i.e., features that can be tracked in consecutive images, but whose 3D positions are not known. Both types of features are processed in an extended Kalman filter (EKF) estimator and are optimally fused with measurements from an inertial measurement unit (IMU). Results from a sounding rocket test, covering the dynamic profile of typical planetary landing scenarios, show 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 navigation, and meet the requirements of future planetary exploration missions.

## I. INTRODUCTION

Space missions involving Entry, Descent and Landing (EDL) maneuvers require high-accuracy position and attitude (pose) determination to enable precise trajectory control. On solar system bodies other than Earth, this is challenging due to the absence of artificial navigation aids such as GPS or radio-beacons. To date, robotic lander missions have used integration of acceleration and rotational velocity measurements from an inertial measurement unit (IMU), augmented by velocity and altitude information from Doppler radar. In these cases, integration of noise and biases, as well as errors in initialization, result in relatively large errors in the touchdown position estimate (e.g., for Mars Pathfinder and the Mars Exploration Rovers in the order of 100 km). However, several future planetary exploration missions will require meter-level landing accuracy, to facilitate the study of geological features of scientific interest [1]. In the absence of radiometric navigation beacons, the most attractive option for increasing the navigation accuracy during EDL is to use camera measurements. Cameras operate in almost any environment, are small, lightweight, and consume little power, and are therefore ideal sensors for space applications.

The Extended Kalman Filter (EKF)-based estimation algorithm presented in this paper combines inertial measurements from an IMU with camera observations of two types of features: (i) *Mapped Landmarks* (MLs), i.e., features whose global 3D coordinates can be determined from a map of the

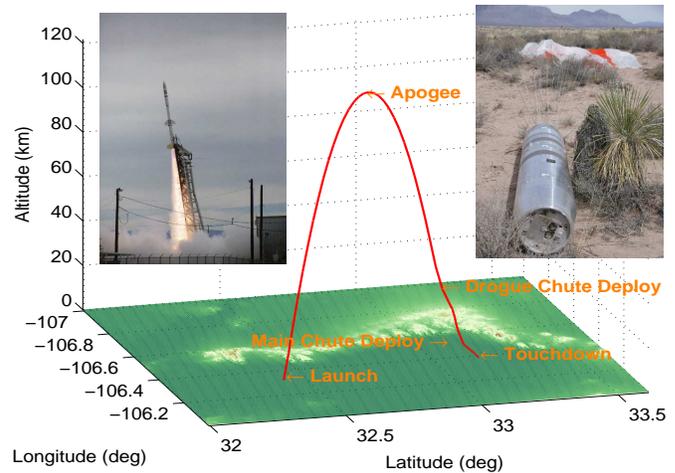


Fig. 1. Sounding Rocket Experiment 41.068

landing site [2], and (ii) *Opportunistic Features* (OFs), i.e., features that can be reliably detected in image sequences, but not in a map of the planet’s surface [3]. We note that the term “map” refers to a co-registered pair of a digital elevation map (DEM) and satellite image, which is available *a priori*.

For precise vision-based EDL, observations of MLs are necessary, since they provide absolute pose information. However, due to the finite resolution of satellite imagery, MLs cannot typically be extracted during the last stages of the descent, when the lander is very close to the ground. To address this problem, in our work we employ OFs for improving the accuracy of pose estimation in these stages. Even though OFs cannot provide absolute pose information, they can be thought of as providing velocity information. Intuitively, viewing a static feature from multiple camera poses provides geometric constraints involving all these poses. The proposed EKF-based estimation algorithm (cf. Section IV-E) utilizes all the constraints from the OF and ML measurements optimally, while maintaining computational complexity only linear in the number of features. These characteristics enable high-accuracy pose estimation in real-time. The results from a sounding-rocket experiment (cf. Fig. 1), which are presented in Section V, demonstrate the filter’s robustness and superior accuracy compared to existing state-of-the-art EDL navigation algorithms, which do not utilize vision information.

## II. RELATED WORK

Only a few recursive estimation approaches that utilize measurements of *a priori* known features have been proposed in the literature. In [4], 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 [5], [6], 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 *and* 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. Finally, in [7] 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.

In addition to processing measurements to MLs, in this work we also utilize measurements to features whose position in the global frame is *not* known in advance (OFs). The standard method of treating such features is to include their positions in the state vector, and to estimate them along with the camera trajectory. This is the well-known Simultaneous Localization and Mapping (SLAM) problem, for which numerous approaches that employ vision and inertial sensing have recently been proposed (e.g., [8] and references therein). 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. In a similar spirit, in [7], [9], a history of only the latest *two* camera poses is considered, and visual measurements are utilized for imposing constraints between them. Moreover, pairwise relative-pose constraints are employed for pose estimation in several approaches that maintain a history of multiple camera poses (e.g., [10] and references therein). Contrary to that, the proposed algorithm does not use the measurements for deriving relative-pose estimates. This reduces the computational burden, avoids possible correlations between the displacement measurements [11], and is more robust to non-linearities. Finally, a sliding window of poses is also maintained in the VSDF algorithm [12]. However, the VSDF is tailored for cases where no motion model is available (in EDL the IMU measurements provide such a model), and its computational complexity is at least quadratic in the number of features.

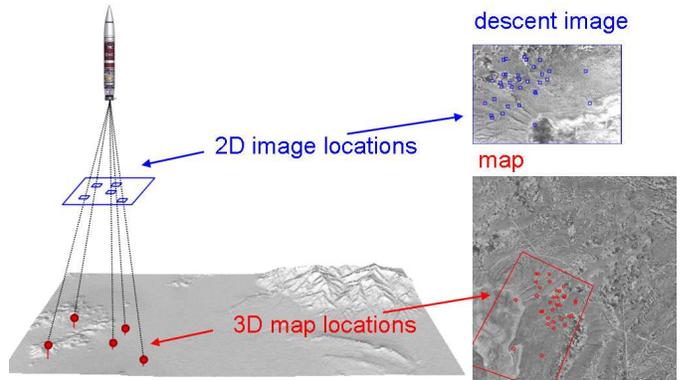


Fig. 2. ML algorithm concept: by matching templates between descent images and a visual map of the landing site, the algorithm produces measurements of the image projections of features with known 3D coordinates (i.e., MLs).

## III. IMAGE PROCESSING

Several techniques can be applied for feature extraction in planetary landing applications. For example, Cheng *et al.* [13] propose using craters as landmarks for navigation. Craters are abundant on most bodies of interest in our solar system, and this makes them a very useful feature. However, there exist sites (e.g., in the polar regions of Mars) where craters are not present. In this case, more general feature types can be used, such as SIFT keypoints [14], and Harris corners [15]. In our work, the image processing module relies on Harris corners and normalized correlation, because (i) corner features can be extracted more efficiently than SIFT keys, since they do not require a search over image scale, (ii) Harris corners have been shown to be more robust to illumination changes than SIFT [16], and (iii) image correlation is a field-tested technology, which has already been employed in EDL applications [17].

The ML algorithm applies normalized correlation to match templates from the descent images to the map (cf. Fig. 2). Each selected template is warped prior to correlation, so as to have the same scale and orientation as the map. This enables us to reduce matching to a 2-D search, which can be carried out very efficiently. We note that the size of the search windows for correlation depends on the accuracy of the camera pose estimate. When the camera pose is very uncertain (e.g., when the first images are recorded), the search windows are very large, and directly applying correlation search for all features is computationally expensive. In that case, an FFT map-matching step is employed prior to ML matching. During this step, a rough alignment of the image with the map is obtained, by frequency-domain correlation of a single large template. Using this initial estimate, the dimensions of the search areas for ML matching are significantly reduced. For OF tracking, we perform pairwise correlation matching between images. To reduce the effects of the changing camera viewpoint, the homography that is induced by the camera motion between consecutive images is accounted for during correlation. For a detailed description of the image-processing algorithms, the reader is referred to [18].

---

**Algorithm 1** Vision-aided Inertial Navigation Algorithm

---

**Propagation:** For each IMU measurement received, propagate the state and covariance estimates (cf. Section IV-B).

**Image acquisition:** Every time a new image is recorded,

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

**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 IV-D)
  - the OFs that are no longer detected in the image sequence (cf. Section IV-E).
- 

#### IV. ESTIMATOR DESCRIPTION

The proposed estimation algorithm (cf. Algorithm 1) employs an EKF to estimate the 3D pose of the *body frame*  $\{B\}$ , which is affixed on 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 that rotates with the planet. The IMU measurements are processed immediately as they become available, for propagating the EKF state and covariance estimates, as discussed in Section IV-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 processing delays introduced by the image processing module, the camera measurements cannot be processed immediately<sup>1</sup>. Second, maintaining a window of camera poses enables the processing of OF measurements (cf. Section IV-E). Therefore, at any time instant the EKF state vector comprises (i) the *evolving* state,  $\mathbf{X}_E$ , which describes the current state of the spacecraft, and (ii) copies of up to  $N$  past poses of the camera. The maximum length of the camera pose history,  $N$ , is selected by pre-flight testing, and is chosen to be equal to the maximum number of images through which an OF can be tracked.

Every time the image processing module produces ML and/or OF measurements, an EKF update takes place. In our implementation, the ML measurements of the latest image are processed immediately as they become available. On the other hand, OF updates occur whenever an OF that has been tracked in a number of images is no longer detected in the latest image. At that time, all the measurements of this feature are processed using the method presented in Section IV-E. In the following sections we present the various steps of the algorithm in detail.

<sup>1</sup>Consider 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 processed normally for state propagation. When, at time step  $k + d$ , the 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.

##### A. Structure of the EKF state vector

The evolving state of the EKF is described by the vector:

$$\mathbf{X}_E = \left[ {}^B_G\bar{q}^T \quad \mathbf{b}_g^T \quad G\mathbf{v}_B^T \quad \mathbf{b}_a^T \quad G\mathbf{p}_B^T \right]^T \quad (1)$$

where  ${}^B_G\bar{q}$  is the unit quaternion [19] 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  $3 \times 1$  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 Eq. (1), the error-state vector for  $\mathbf{X}_E$  is defined accordingly, as:

$$\tilde{\mathbf{X}}_E = \left[ \delta\boldsymbol{\theta}_B^T \quad \tilde{\mathbf{b}}_g^T \quad G\tilde{\mathbf{v}}_B^T \quad \tilde{\mathbf{b}}_a^T \quad G\tilde{\mathbf{p}}_B^T \right]^T \quad (2)$$

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

$$\delta\bar{q} \simeq \left[ \frac{1}{2}\delta\boldsymbol{\theta}^T \quad 1 \right]^T \quad (3)$$

Since attitude corresponds to 3 degrees of freedom, 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:

$$\hat{\mathbf{X}}_k = \left[ \hat{\mathbf{X}}_{E_k}^T \quad C_1 \hat{q}^T \quad G\hat{\mathbf{p}}_{C_1}^T \quad \dots \quad C_N \hat{q}^T \quad G\hat{\mathbf{p}}_{C_N}^T \right]^T \quad (4)$$

where  $C_i \hat{q}$  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:

$$\tilde{\mathbf{X}}_k = \left[ \tilde{\mathbf{X}}_{E_k}^T \quad \delta\boldsymbol{\theta}_{C_1}^T \quad G\tilde{\mathbf{p}}_{C_1}^T \quad \dots \quad \delta\boldsymbol{\theta}_{C_N}^T \quad G\tilde{\mathbf{p}}_{C_N}^T \right]^T \quad (5)$$

##### B. Propagation

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 [20]:

$${}^B_G\dot{\bar{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t)){}^B_G\bar{q}(t), \quad \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t) \quad (6)$$

$$G\dot{\mathbf{v}}_B(t) = G\mathbf{a}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad G\dot{\mathbf{p}}_B(t) = G\mathbf{v}_B(t)$$

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

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

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

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

where  $\mathbf{C}(\cdot)$  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 in the state propagation equations (Eq. (6)) we obtain the equations for propagating the *estimates* of the evolving state:

$$\begin{aligned}\hat{q}_G^B &= \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}){}^B\hat{q}_G^B, \quad \hat{\mathbf{b}}_g = \mathbf{0}_{3 \times 1}, \\ {}^G\hat{\mathbf{v}}_B &= \mathbf{C}_{\hat{q}}^T \hat{\mathbf{a}} - 2[\boldsymbol{\omega}_G \times]^G \hat{\mathbf{v}}_B - [\boldsymbol{\omega}_G \times]^2 {}^G \hat{\mathbf{p}}_B + {}^G \mathbf{g} \quad (7) \\ \hat{\mathbf{b}}_a &= \mathbf{0}_{3 \times 1}, \quad {}^G\hat{\mathbf{p}}_B = {}^G \hat{\mathbf{v}}_B\end{aligned}$$

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

$$\dot{\tilde{\mathbf{X}}}_E = \mathbf{F}_E \tilde{\mathbf{X}}_E + \mathbf{G}_E \mathbf{n}_{\text{IMU}} \quad (8)$$

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  $\mathbf{n}_{\text{IMU}}$ ,  $\mathbf{Q}_{\text{IMU}}$ , depends on the IMU noise characteristics and is computed off-line during sensor calibration. Finally, the values of the jacobians  $\mathbf{F}_E$  and  $\mathbf{G}_E$ , which appear in Eq. (8), are given in [18].

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 5th order Runge-Kutta numerical integration of Eqs. (7). Moreover, the covariance matrix of the EKF state has to be propagated. For this purpose, we introduce the following partitioning for the covariance matrix:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{EE_{k|k}} & \mathbf{P}_{EC_{k|k}} \\ \mathbf{P}_{EC_{k|k}}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix} \quad (9)$$

where  $\mathbf{P}_{EE_{k|k}}$  is the  $15 \times 15$  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 correlation 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:

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

where the covariance of the evolving state at time-step  $k+1$  is computed by numerical integration of the Lyapunov equation:

$$\dot{\mathbf{P}}_{EE} = \mathbf{F}_E \mathbf{P}_{EE} + \mathbf{P}_{EE} \mathbf{F}_E^T + \mathbf{G}_E \mathbf{Q}_{\text{IMU}} \mathbf{G}_E^T \quad (10)$$

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

$$\dot{\boldsymbol{\Phi}}(t_k + \tau, t_k) = \mathbf{F}_E \boldsymbol{\Phi}(t_k + \tau, t_k), \quad \tau \in [0, T] \quad (11)$$

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

### C. State Augmentation

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

$${}^C\hat{q} = {}^C_B\hat{q} \otimes {}^B\hat{q} \quad (12)$$

$${}^G\hat{\mathbf{p}}_C = {}^G\hat{\mathbf{p}}_B + \mathbf{C}_{\hat{q}}^T {}^B\mathbf{p}_C \quad (13)$$

where  ${}^C_B\hat{q}$  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 [3].

### 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:

$$\mathbf{z}_i^{(j)} = \frac{1}{c_i Z_j} \begin{bmatrix} c_i X_j \\ c_i Y_j \end{bmatrix} + \mathbf{n}_i^{(j)} \quad (14)$$

where  $\mathbf{n}_i^{(j)}$  is the  $2 \times 1$  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,  ${}^{C_i}\mathbf{p}_j$ , is given by:

$${}^{C_i}\mathbf{p}_j = \begin{bmatrix} c_i X_j \\ c_i Y_j \\ c_i Z_j \end{bmatrix} = \mathbf{C}({}^{C_i}\hat{q})({}^G\mathbf{p}_{\ell_j} - {}^G\mathbf{p}_{C_i}) \quad (15)$$

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

$$\hat{\mathbf{z}}_i^{(j)} = \frac{1}{c_i \hat{Z}_j} \begin{bmatrix} c_i \hat{X}_j \\ c_i \hat{Y}_j \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} c_i \hat{X}_j \\ c_i \hat{Y}_j \\ c_i \hat{Z}_j \end{bmatrix} = \mathbf{C}({}^{C_i}\hat{q})({}^G\mathbf{p}_{\ell_j} - {}^G\hat{\mathbf{p}}_{C_i}) \quad (16)$$

From Eqs. (14) and (16) 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 Eq. (14),  $\mathbf{r}_{\text{ML}_i}^{(j)}$  is written as:

$$\mathbf{r}_{\text{ML}_i}^{(j)} \simeq \mathbf{H}_{\delta\theta}^{(j)} \delta\boldsymbol{\theta}_{C_i} + \mathbf{H}_{\mathbf{p}_i}^{(j)} {}^G\tilde{\mathbf{p}}_{C_i} + \mathbf{n}_i^{(j)} = \mathbf{H}_{\text{ML}_i}^{(j)} \tilde{\mathbf{X}} + \mathbf{n}_i^{(j)} \quad (17)$$

where

$$\mathbf{H}_{\delta\theta}^{(j)} = \frac{1}{c_i \hat{Z}_j} \begin{bmatrix} \mathbf{I}_2 & -\hat{\mathbf{z}}_i^{(j)} \end{bmatrix} \lfloor \mathbf{C}({}^{C_i}\hat{q})({}^G\mathbf{p}_{\ell_j} - {}^G\hat{\mathbf{p}}_{C_i}) \rfloor$$

$$\mathbf{H}_{\mathbf{p}_i}^{(j)} = -\frac{1}{C_i \hat{z}_j} \begin{bmatrix} \mathbf{I}_2 & -\hat{\mathbf{z}}_i^{(j)} \end{bmatrix} \mathbf{C}(C_i \hat{q})$$

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

The residual defined in Eq. (17) is employed for performing EKF updates, as described in Section IV-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 Eq. (14). 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. Once this estimate has been computed, the expected value of each of the feature measurements can be evaluated, similarly to Eq. (16), 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:

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

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. In the last expression,  $\mathbf{H}_{f_i}^{(j)} = -\mathbf{H}_{\mathbf{p}_i}^{(j)}$  is the Jacobian of the residual with respect to  ${}^G\hat{\mathbf{p}}_{\ell_j}$ , and

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

By stacking the residuals corresponding to all the observations of this feature, we obtain:

$$\mathbf{r}^{(j)} \simeq \mathbf{H}_{\mathbf{X}}^{(j)} \tilde{\mathbf{X}} + \mathbf{H}_f^{(j)G} \tilde{\mathbf{p}}_{\ell_j} + \mathbf{n}^{(j)} \quad (18)$$

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 feature observations in different images are 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 Eq. (18) *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  ${}^G\hat{\mathbf{p}}_{\ell_j}$  is computed using the state estimates and the measurements  $\mathbf{z}_i^{(j)}$  in a least-squares minimization routine). 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:

$$\mathbf{r}_{\text{OF}}^{(j)} = \mathbf{U}^T (\mathbf{z}^{(j)} - \hat{\mathbf{z}}^{(j)})$$

$$\simeq \mathbf{U}^T \mathbf{H}_{\mathbf{X}}^{(j)} \tilde{\mathbf{X}} + \mathbf{U}^T \mathbf{n}^{(j)} = \mathbf{H}_{\text{OF}}^{(j)} \tilde{\mathbf{X}} + \mathbf{n}_o^{(j)} \quad (19)$$

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 [21]. The covariance matrix of the noise vector  $\mathbf{n}_o^{(j)}$  can be easily 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. Eq. (19) defines a *linearized* constraint between all the 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  $M_j$  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 1), the corresponding residual vectors and measurement Jacobians (Eqs. (17) and (19)) are created. Stacking all these together yields the following residual vector:

$$\mathbf{r} = \mathbf{H} \tilde{\mathbf{X}} + \mathbf{n} \quad (20)$$

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  $\mathbf{R} = \sigma_{\text{im}}^2 \mathbf{I}_L$ . Once the residual,  $\mathbf{r}$ , and the measurement Jacobian matrix,  $\mathbf{H}$  of Eq. (20) have been computed, the EKF update proceeds according to the standard equations [22]. In our work, we employ the QR decomposition of the matrix  $\mathbf{H}$  to reduce the computational complexity of EKF updates [7]. 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.

## V. 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 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  $768 \times 484$  pixels, 8 bits/pixel, and a field of view (FOV) of  $38^\circ \times 24^\circ$ . A common GPS time tag from 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

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 td.
Horizontal Velocity	3 m/s at touchdown	< 0.5 m/s at td.
Off nadir angle	$\leq 12^\circ$	< $20^\circ$
Off nadir angular rate	$\leq 19^\circ/\text{s}$	< $60^\circ/\text{s}$
Roll Rate	$\leq 360^\circ/\text{s}$	< $60^\circ/\text{s}$

TABLE I  
DYNAMICS COMPARISON BETWEEN SOUNDING ROCKET AND MARS EDL.

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, and the current C++ implementation of the pose estimator runs at 20 Hz on a 2 GHz CPU, with the number of stored poses set to  $N = 20$ .

### B. Experiment Profile and Relevance

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 similar to those during an actual Mars landing, as shown by the comparison in Table I.

Fig. 1 shows the rocket's trajectory superimposed on a 3D map of the area. A zoomed-in view of the flight path after main parachute deployment is depicted in Fig. 3. 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 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 were processed, the VISINAV algorithm corrected the accumulated error, and the estimated trajectory became 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, using images at 3 Hz, and the second between 1600 m and 230 m above ground, processing only one frame per second (1 Hz), and yielding up to 80 MLs per image (cf. Fig. 4). The artificial gap, during which the filter had to rely on open-loop IMU integration, was introduced to emulate a temporary failure of ML detection. In EDL, this could arise, for example, due to large off-nadir viewing angles

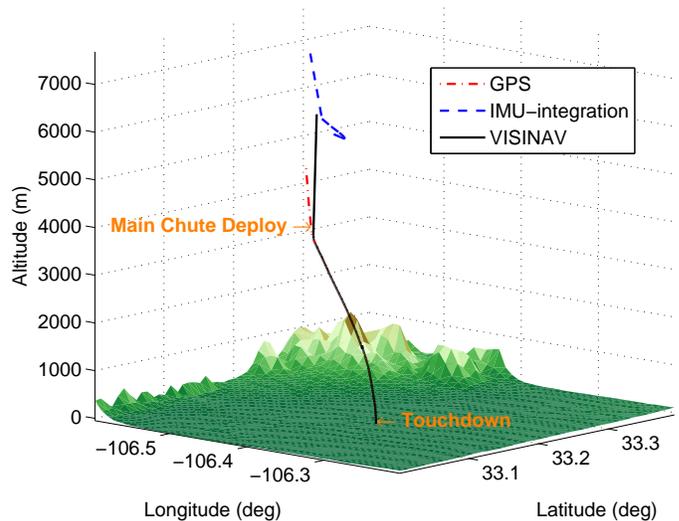


Fig. 3. Zoomed-in view of trajectory after main parachute deployment.

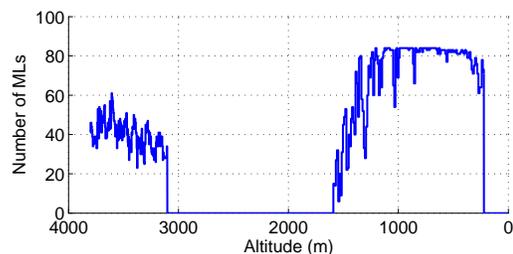


Fig. 4. Number of detected MLs vs. altitude.

caused by a pose correction maneuver during powered descent. Once images become available after this intermediate open-loop phase, a significant change in image scale has occurred. Despite this, the ML algorithm is capable of successfully matching MLs, and the EKF estimates' uncertainty drops sharply within a few seconds after resuming ML updates (cf. Figs. 5, 7 at 1600 m altitude).

The 3D ground coordinates for the MLs were obtained from USGS 1 Meter Digital Orthoimagery Quarter Quadrangles taken in 2001, combined with 1 arc second finished elevation data from the Shuttle Radar Topography Mission [23]. For feature matching in the first set, the entire  $7 \times 8$  km map image was resampled to  $\sim 7$  m/pixel, while for the second set the map was cropped to  $2 \times 2$  km and used at its base resolution of 1 m/pixel.

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. Even though OFs do not make the system 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. 5 and 7 show the resulting errors and the corresponding  $3\sigma$ -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.

a) *First ML set*: When processing the first MLs, the algorithm converges within about 5 seconds from the large error ( $\sim 2700$  m) accumulated during IMU integration to within 18 m of GPS ground-truth (cf. Figs. 3, 7). During the open-loop integration phase between 3100 m to 1600 m altitude, the pose uncertainty is again increasing.

b) *Second ML set and OFs*: The pose estimates almost instantaneously converge close to ground truth once ML updates resume at 1600 m. ML- and OF-updates (the latter starting at 330 m) reduce the position uncertainty bounds to approximately  $\pm 4$  m, and the velocity uncertainty to less than  $\pm 0.25$  m/s along each axis ( $3\sigma$ ). Notice that the actual error at the beginning of OF updates is smaller than at the end of ML processing ten seconds later. This can be attributed to the already decreasing number of detected MLs within the FOV at this altitude, due to the difference in resolution between satellite and camera image (cf. Fig. 4). With the discontinuation of ML processing at 230 m altitude, the pose uncertainty begins to increase even more, although still at a very low rate (cf. the zoomed-in view of the errors for the final 300 m in Figs. 6 and 8). As predicted, this is the result of the system becoming unobservable. Table II shows the final velocity and position error magnitudes at touchdown, which are approximately 6.4 m for 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 ( $3\sigma$ ) during processing of ML updates, with a temporary increase to  $\pm 0.9^\circ$  during open-loop IMU integration between 3100 m and 1600 m altitude. Due to the lack of ground-truth, the attitude uncertainty was determined from the EKF estimates of the covariance matrix. The figures for position and velocity errors (cf. Figs. 5-8) show that the filter estimates for these variables are consistent, indicating that the attitude estimates are also consistent. The filter attitude estimate was further verified through an independent measurement of the final attitude at touchdown using a compass.

## VI. CONCLUSION

In this paper, we have presented the analysis and experimental validation of a vision-aided inertial navigation algorithm for planetary landing applications. Results from a sounding rocket test, covering the dynamic profile of typical planetary EDL scenarios, 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 EDL navigation without vision, and meet the requirements of future planetary exploration missions [1]. The algorithm tightly couples IMU and camera measurements of mapped landmarks (MLs) and opportunistic features (OFs), in a resource-adaptive and hence

	Altitude (m)	Time (s)	Position Error (m)	Velocity 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

TABLE II  
CONDITIONS FOR THE DIFFERENT EKF UPDATE PHASES.

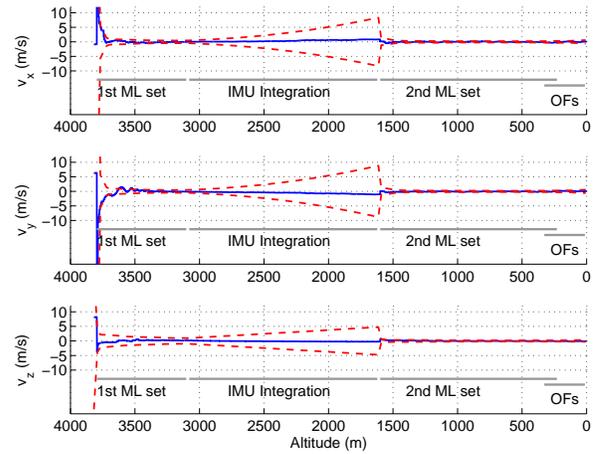


Fig. 5. Velocity error expressed in NED frame (blue solid line) and corresponding  $3\sigma$ -bounds (red dashed line). Note that  $x-y-z$  in the plots corresponds to N-E-D.

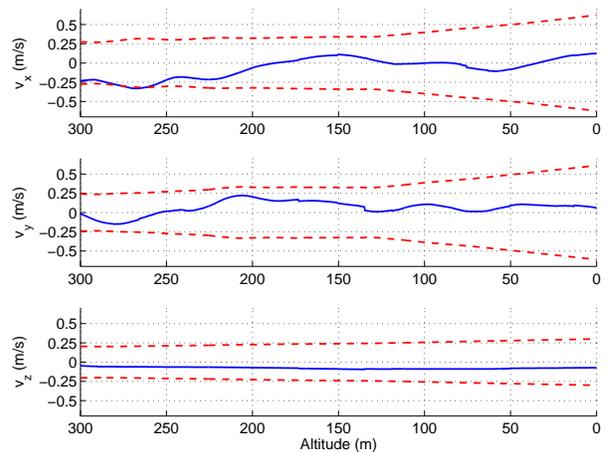


Fig. 6. Velocity error in NED frame (zoomed-in view of Fig. 5 before touchdown).

real-time capable fashion. It is thus able to provide very accurate, high-bandwidth estimates for precision guidance and

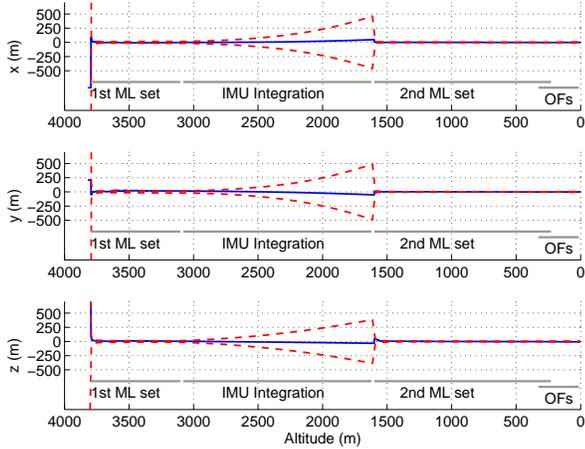


Fig. 7. Position error expressed in NED frame (blue solid line) and corresponding  $3\sigma$ -bounds (red dashed line).

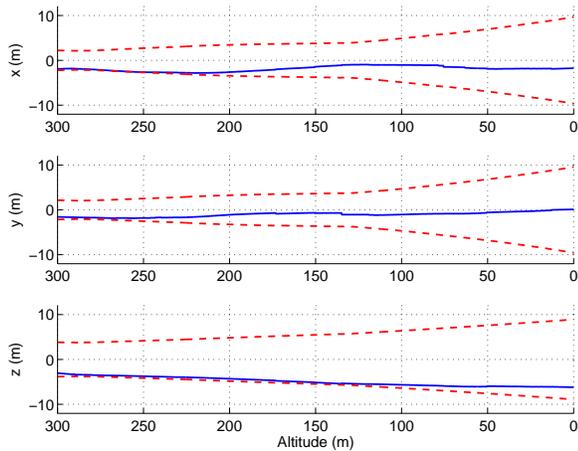


Fig. 8. Position error in NED frame (zoomed-in view of Fig. 7 before touchdown).

control. It should be pointed out that the proposed estimator is modular and easily extendable to incorporate additional sensor information (e.g., doppler radar), as well as different image processing algorithms that provide unit vector measurements to corresponding features. Future work includes optimal selection of a subset of the most informative image features to track. This would allow efficient use of camera information in the presence of limited computational resources.

#### ACKNOWLEDGEMENTS

This work was supported by the University of Minnesota (DTC), the NASA Mars Technology Program (MTP-1263201), and the National Science Foundation (EIA-0324864, IIS-0643680).

#### REFERENCES

[1] NASA, "Solar system exploration roadmap," [http://solarsystem.nasa.gov/multimedia/downloads/SSE\\_RoadMap\\_2006\\_Report\\_FC-A\\_med.pdf](http://solarsystem.nasa.gov/multimedia/downloads/SSE_RoadMap_2006_Report_FC-A_med.pdf), Sep. 2006.

[2] 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," *Journal of Field Robotics*, vol. 24, no. 5, pp. 357–378, May 2007.

[3] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, April 2007, pp. 3565–3572.

[4] J. L. Crassidis, R. Alonso, and J. L. Junkins, "Optimal attitude and position determination from line-of-sight measurements," *The Journal of the Astronautical Sciences*, vol. 48, no. 2–3, pp. 391–408, April–Sept. 2000.

[5] A. Wu, E. Johnson, and A. Proctor, "Vision-aided inertial navigation for flight control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, no. AIAA 2005-5998, San Francisco, CA, Aug. 2005.

[6] G. F. Ivey and E. Johnson, "Investigation of methods for simultaneous localization and mapping using vision sensors," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, no. AIAA 2006-6578, Keystone, CO, Aug. 2006.

[7] D. S. Bayard and P. B. Brugarolas, "An estimation algorithm for vision-based exploration of small bodies in space," in *Proceedings of the 2005 American Control Conference*, vol. 7, Portland, Oregon, Jun. 2005, pp. 4589–4595.

[8] D. Strelow, "Motion estimation from image and inertial measurements," Ph.D. dissertation, Carnegie Mellon University, Nov. 2004.

[9] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, Washington D.C., 2002, pp. 4326–33.

[10] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, Dec. 2006.

[11] A. I. Mourikis and S. I. Roumeliotis, "On the treatment of relative-pose measurements for mobile robot localization," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Orlando, FL, May 15-19 2006, pp. 2277 – 2284.

[12] P. McLauchlan, "The variable state dimension filter," Centre for Vision, Speech and Signal Processing, University of Surrey, UK, Tech. Rep., 1999.

[13] Y. Cheng and A. Ansar, "Landmark based position estimation for pin-point landing on mars," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 4470–4475.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, 2004.

[15] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[16] A. Ansar, "2004 small body GN&C research report: Feature recognition algorithms," in *Small Body Guidance Navigation and Control FY 2004 RTD Annual Report (Internal Document)*. Pasadena, CA: Jet Propulsion Laboratory, 2004, no. D-30282 / D-30714, pp. 151–171.

[17] 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," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 319–341, September 2007.

[18] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *Journal of Guidance, Control, and Dynamics*, 2007, submitted.

[19] W. G. Breckenridge, "Quaternions proposed standard conventions," Jet Propulsion Laboratory, Pasadena, CA, Interoffice Memorandum IOM 343-79-1199, 1999.

[20] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1997.

[21] G. Golub and C. van Loan, *Matrix computations*. The Johns Hopkins University Press, London, 1996.

[22] P. S. Maybeck, *Stochastic Models, Estimation and Control*. New York: Academic Press, 1979, vol. 1+2.

[23] U.S. Geological Survey, "Seamless data distribution system," <http://seamless.usgs.gov/index.asp>, 2006.