
Differential Geometry Final Project

Tim Player

Harvey Mudd College

Claremont, CA 91711

tplayer@hmc.edu

Abstract

An algorithm for recovering the complete pose of a rocket from inertial data and outward-facing cameras was developed. Complete algorithms are presented for recovering the horizon from outward-facing videos, and for fusing the video pitch and yaw information with inertial sensors via an extended Kalman filter operating on orientation deviation states. With the filter data, the rocket achieved an apogee of 2,742 meters and a top speed of 324 meters per second.

1 Motivation

Precisely localizing moving platforms using inertial sensors is a fundamental problem in robotics. These sensors, such as gyroscopes, accelerometers, and magnetometers, allow changes in position and orientation (together, “pose”) to be determined over short time scales in a process called *dead reckoning*. This approach is used, for instance, in aerospace where precise estimates of aircraft pose may lead to safer landings. However, the cheapest sensors provide noisy measurements which cause dead-reckoning to be generally ineffective over long periods of time.

MEMS magnetometers, which are used as digital compasses to provide estimates of orientation with respect to the Earth’s magnetic field, can provide erroneous estimates due to the presence of magnetic material near the sensor, unmodeled geological perturbations in the Earth’s field, miscalibration, and white observation noise. This can lead to uncertainty in orientation measurements on the order of several degrees (1). The error, in turn, causes exponentially-growing errors in position estimation, as we will discuss in Section 2. As a result, absolute measurements of orientation which are independent of magnetometers have direct utility in robotic state estimation.

This project implements direct orientation measurement via computer vision horizon estimation. Using inertial and visual data collected on a supersonic sounding rocket in April 2019, we develop an algorithm for pose estimation which incorporates visual horizon estimation as an independent, redundant measure of orientation to address the deficiencies of magnetometers. As shown in Fig. 1, cameras mounted inside of the rocket face opposite directions so they can see the horizon. As a result, changes in rocket attitude manifest in the camera image as coupled and opposite tilting, raising, or lowering of the horizon in each image. This allows direct estimation of pitch and yaw in a global frame.

In this paper, we first provide an explicit overview of the pose-estimation problem in section 2 and detail some existing algorithms. In section 4, we develop the dynamical model used to represent the physical constraints of the rocket trajectory, including the parametrization of orientation with unit quaternions. Appendix A presents in more depth the available data collected during the April flights and the steps needed to prepare it for use in a sensor fusion algorithm. Section 5 gives an overview of the multiplicative Extended Kalman Filter we use, and results follow.



(a)



(b)

Figure 1: Visual horizon identification scheme during rocket flight. Approximate field of view of opposite-facing cameras is shown in red in sub-figure (a). Camera output is shown in sub-figure (b).

2 Pose Estimation Problem

The problem of *sensor fusion* comprises strategies to intelligently combine observations from the various sensors which indirectly measure different elements of the platform’s pose. Whereas the system’s pose includes

$$\{x\text{-position}, y\text{-position}, z\text{-position}, \text{pitch}, \text{roll}, \text{yaw}\} = \{x, y, z, r, p, y\},$$

direct measurements of these items are not available. The gyroscope, for instance, observes the time-derivatives $\{\dot{r}, \dot{p}, \dot{y}\}$, and the accelerometer measures *specific force*, which provides gravity-biased measurements of $\{\ddot{x}, \ddot{y}, \ddot{z}\}$ in a local frame. As discussed above, magnetometers provide a noisy absolute measurement of the orientation $\{r, p, y\}$.

It is necessary to define several frames of reference. First, the *body* frame **b** is the coordinate frame of the moving rocket. This has the pitch axis x_b , yaw axis y_b , and roll axis z_b shown in Fig. 2.

The navigation frame **n** is the coordinate frame in which we wish to localize the rocket, whose origin is at the launch pad. To this we ascribe the East, North, Up (ENU) triad. Note that both systems described have positive orientation.

To be fully comprehensive, we must also include the *inertial* frame **i** and the *Earth* frame **e**. The accelerometer and rate gyroscope take measurements with respect to the inertial frame rather than the navigation frame or body frame, and thus the effects of Coriolis acceleration caused by the Earth’s rotation and body acceleration caused by the Earth’s orbit may be factored in. However, as the rocket flight is on the order of minutes, the frames do not become significantly different so we make the valid simplifying assumption that the navigation frame **n** is inertial, and we remove **i** and **e** from further discussion.

To achieve a pose in the navigation frame based on measurements taken in the body frame, we must correct for the presence of gravity in the accelerometer measurements prior to integrating. As shown in Fig. 3, the rocket’s orientation is first estimated via direct measurement with the magnetometer and by integrating of rate gyroscope values from the last time step. With this known orientation — a rotation between the **b** and **n** frames — the direction of gravity is known. Then gravity can be subtracted from the accelerometer readings to provide a measurement of true rocket acceleration in the navigation frame rather than specific force. Lastly, this acceleration is integrated twice to arrive at velocity and position.

To estimate pose, our approach to dead reckoning must intelligently utilize all data. Specifically, at each step t along the rocket’s trajectory we seek to quantify a conditional belief distribution regarding

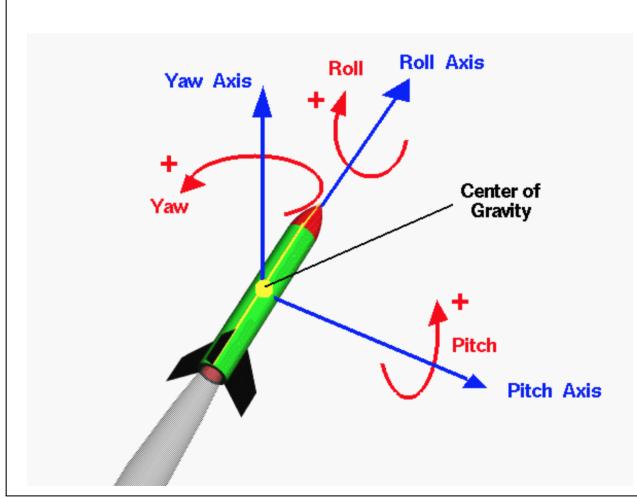


Figure 2: Attitude axis conventions. Camera 1's aperture lies on the pitch axis.

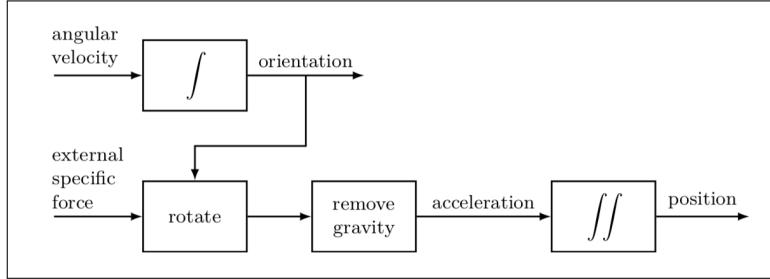


Figure 3: Schematic illustration of dead-reckoning, where the accelerometer measurements (external specific force) and the gyroscope measurements (angular velocity) are integrated to position and orientation. From (?).

the rocket's pose x_t

$$bel(t) = p(x_t | z_{1:t}),$$

where $z_{1:t}$ are all of the measurements available from sensors through times 1 through t . Under the Markov assumption, which is that our description of the rocket's state is complete enough that the knowledge of a prior state specifies belief of future states as well, then Eq. 2 reduces to

$$bel(t) = p(x_t | x_{t-1}, z_t),$$

which is to say that the information from all measurements $z_{1:t-1}$ is included in the state x_{t-1} . This forms the basis of the Bayes Filter in general. We will implement a specific closed-form solution of the Bayes Filter in section 5.

3 Quaternion Orientation Parametrization

To model the orientation of the rocket, several parametrizations are available. The orientation of the rocket is given as a rotation from the navigation frame to the body frame of the rocket. Rotations are succinctly represented in matrix form as the set of all orthogonal linear transformations with unit determinant. That is, rotations form the special orthogonal group $SO(n)$. In three dimensions,

$$SO(3) = \{A \in M_{3 \times 3} \mid A^T A = I, \det A = 1\}.$$

The set of all rotations $SO(3)$ is a group, so it has, for all $A \in SO(3)$, an inverse A^{-1} . Above, you can see that $A^{-1} = A^T$. Additionally, $SO(3)$ is closed under multiplication, the associative property applies, and the identity element $I \in SO(3)$. In fact, $SO(3)$ is a *Lie group*, since it has manifold structure and an exponential map from the corresponding Lie algebra (?).

Although the above matrix representation has an intuitive interpretation as the change-of-basis matrix representing the projection of each body-frame basis vector onto the navigation frame basis triad, it is inconvenient to work directly with matrices due to problems numerical instability and the high-dimensionality intrinsic to matrices. In fact, we choose to represent the orientation of the rocket as a *unit quaternion*, an identical parametrization of $SO(3)$.

A unit quaternion q is a member of the Hamiltonian numbers such that

$$q = (q_0 \ q_1 \ q_2 \ q_3)^\top = \begin{pmatrix} q_0 \\ q_v \end{pmatrix}, \quad q \in \mathbb{R}^4, \quad \|q\| = 1.$$

As Hamiltonian numbers, the three elements of q_v behave similarly to the cross product in \mathbb{R}^3 , such that

$$i^2 = j^2 = k^2 = ijk = -1.$$

Unit quaternions may be thought of as directly representing an rotation vector. That is, any orientation can be specified by indicating a unit axis of rotation $\eta^v = (x, y, z)^\top$ and an angle α to rotate a given basis triad about the axis. In fact,

$$q^{uv}(\eta^v, \alpha) = \begin{pmatrix} \cos \frac{\alpha}{2} \\ \sin \frac{\alpha}{2} \eta^v \end{pmatrix}.$$

A vector $u \in \mathbb{R}^4$ can then be rotated using quaternion multiplication, defined such that

$$x^u = q^{uv} \odot x^v \odot (q^{uv})^C,$$

where $(q^{uv})^C = q^{vu}$ denotes the quaternion complement, defined as

$$q^C = (q_o(-q_v)^\top)^\top.$$

As equivalent representations of the special orthogonal group, we may convert between unit quaternions and rotation matrices via their corresponding exponential maps from rotation vectors. In the remainder of this paper, we will use the mappings

$$q = \exp_q(\eta) = \begin{pmatrix} \cos \|\eta\|_2 \\ \sin \|\eta\|_2 \frac{\eta}{\|\eta\|_2} \end{pmatrix} \quad (1)$$

and

$$R = \exp_R([\eta \times]) = I_3 + \sin(\|\eta\|_2) \left[\frac{\eta}{\|\eta\|_2} \times \right] + (1 - \cos(\|\eta\|_2)) \left[\frac{\eta}{\|\eta\|_2} \times \right]^2 \quad (2)$$

where $[\cdot \times]$ denotes the standard matrix cross product. This will allow us to make the approximations, for small η , that

$$\exp_q(\eta) \approx \begin{pmatrix} 1 \\ \eta \end{pmatrix} \quad (3)$$

$$\exp_R(\eta) \approx I_3 + [\eta \times]. \quad (4)$$

4 Dynamic Model

We borrow the dynamics equations from (1) to relate the position p_t^n , velocity v_t^n , and orientation to the measured acceleration and gyroscope inputs via

$$\begin{pmatrix} p_{t+1}^n \\ v_{t+1}^n \\ q_{t+1}^{nb} \end{pmatrix} = \begin{pmatrix} p_t^n + T v_t^n + \frac{T^2}{2} (R_t^{nb} y_{a,t} + g^n + e_{p,a,t}) \\ v_t^n + T (R_t^{nb} y_{a,t} + g^n + e_{v,a,t}) \\ q_t^{nb} \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - e_{\omega,t}) \right) \end{pmatrix} \quad (5)$$

where

$$\begin{aligned} e_{p,a,t} &\sim \mathcal{N}(0, \Sigma_a), & e_{v,a,t} &\sim \mathcal{N}(0, \Sigma_a), \\ e_{\omega,t} &\sim \mathcal{N}(0, \Sigma_{\omega}). \end{aligned}$$

with $\Sigma_a = \sigma_a^2 I_3$ and $\Sigma_{\omega} = \sigma_{\omega}^2 I_3$. Then, the problem becomes to estimate the state using known time series data $y_{a,t}$ and $y_{\omega,t}$ for each step t .

Additionally, measurements of orientation are available from the magnetometer as

$$y_{m,t} = R_t^{bn} m^n + e_{m,t}. \quad (6)$$

4.1 Measurement Equations for Camera Pitch and Yaw

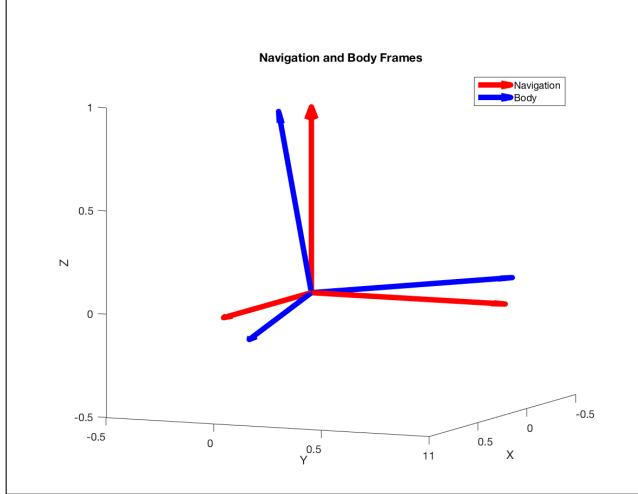


Figure 4: Navigation and body frames representing a rotation matrix R_{nb} .

A thorough treatment of the image formation process involves concepts from projective geometry and optics; such a treatment is not given here. Instead we direct the reader to Appendix A.2 or Szeliski's *Computer Vision: Algorithms and Applications*. For our purposes, we only intend to recover the pitch and yaw of the rocket from the two camera horizons available. This does not require significant projective geometry, though the following intuitive relationship is derived from it.

Figures 1 and 2 are useful in determining this relationship. Since Camera 1 is directed along the pitch axis of the rocket and Camera 2 is directed in the opposite direction, the rotation and height of the horizon in the image directly describe the global pitch and yaw of the rocket respectively.

Furthermore, consider the body-frame axes shown in blue in Figure 4. The rotation matrix

$$R^{nb} = [v_1 \ v_2 \ v_3]$$

is comprised of the three columns representing the local yaw, pitch, and roll axes expressed in the global x , y , and z basis. The rocket's pitch and yaw angles are therefore directly related to the projection of v_1 and v_2 onto the global z axis. In fact, since the vectors of R^{nb} are unit vectors, the pitch and yaw of the rocket are given by

$$p_t = \sin^{-1}[R_t^{nb}]_{3,2}, \quad (7)$$

$$y_t = \sin^{-1}[R_t^{nb}]_{3,1}, \quad (8)$$

where R_t^{nb} is the rocket's rotation matrix at time t .

This relationship is verified experimentally. Figure 5 shows that the angular values calculated from video frames from Camera 1 and 2 closely track the predicted pitch and yaw values *before* using the camera as an input to the sensor fusion scheme.

It is straightforward to determine p_t and y_t after identifying the horizon in the camera images through the technique given in Appendix A.1. Figure 1b shows the view from the two cameras with horizon slopes θ_1 and θ_2 and heights h_1 and h_2 . Then, under a small angle approximation, the measurements of pitch and yaw become

$$y_p = \frac{\theta_2 - \theta_1}{2},$$

$$y_y = (0.0073) \frac{h_2 - h_1}{2},$$

where the constant factor in the computation of y_y relates the sensitivity of image pixels to perturbations about the yaw axis. Together, these measurements provide an independent measurement of rocket attitude relative to the magnetometer alone, which eliminates the degree of freedom originating from inability identify rocket rotation parallel to the magnetic field.

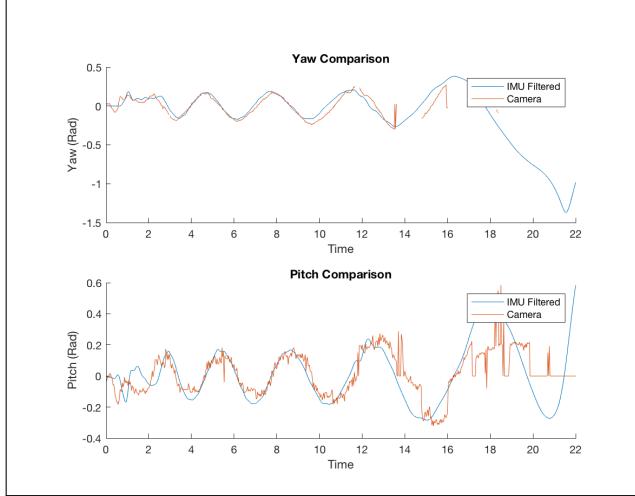


Figure 5: Comparison of yaw and pitch values calculated from images with those calculated from inertial data. The granularity in the camera pitch is an artefact of the sensitivity of the Hough line-detecting transform to irregularities in the horizon.

5 Sensor Fusion Algorithm

We will use the multiplicative Extended Kalman Filter with a quaternion orientation estimation, modified to achieve pose estimation rather than orientation alone. The unmodified algorithm is from (1).

This Extended Kalman Filter operates on *orientation deviations* rather than orientations alone. Rather than keeping track of the four-element orientation quaternion, only the three-element incremental rotation vector is recorded. Due to Euler's theorem, this representation is sufficient to store the entire state history at a given time.

As a consequence of working with orientation *deviations*, the measurement matrix H includes only the derivative of the measurement relationship. This will not be explained in detail in this draft, but its derived in (1).

Additionally, a complete derivation of the Kalman filter algorithm is given in (2), and a fascinating visual derivation is available at this link. However, these results will not be reproduced here.

6 Results

The algorithm successfully predicted the trajectory of the rocket. Figure 6 shows the Kalman-filtered prediction of the ascent trajectory. The orange circles show the GPS locations that were available during the descent, which carried the rocket to the East due to westerly wind. The GPS was unavailable while the rocket was rising because commercial GPS devices shut off if they are going very fast. The fact that the estimated rocket position is very close to the known GPS coordinate suggests the accuracy of this approach.

Figure 7 shows the traces of the local yaw, pitch, and roll axes during the first second of flight. This figure demonstrates the successful operation of the rate gyroscopes as well as the orientation-deviation update algorithm. Shown in black is the vector cross product

$$y\omega \times R^{nb},$$

which projects the angular derivative of the system onto the tangent bundle of $SO(2)$ in order to give a sense of the direction the system is evolving. In this case, the rocket is rotating clockwise around the roll axis.

Figure 8 shows the one-standard-deviation uncertainty ellipses at several points throughout the trajectory. The covariance was modified during the operation of the Kalman filter by the quadratic

Algorithm 1: Extended Kalman filter for pose estimation with orientation deviation state

Input: Inertial data $\{y_{a,t}, y_{\omega,t}\}_{t=1}^N$, magnetometer data $\{y_{m,t}\}_{t=1}^N$ and covariance matrices Σ_ω , Σ_a , and Σ_m .

Output: An estimate of the state $\{x, y, z, v_x, v_y, v_z, \eta_i, \eta_j, \eta_k\}$ and orientation $\tilde{q}_{t|t}^{nb}$ for $t = 1, \dots, N$.

- 1 Compute $\tilde{q}_{1|1}^{nb}$ using the TRIAD method and set the initial state covariance $P_{1|1}$.
 - 2 **for** $t = 2, \dots, N$ **do**

(a) Time Update

$$\hat{x}_{t|t-1} = \begin{pmatrix} p_{t|t-1}^n \\ v_{t|t-1}^n \\ \hat{\eta}_{t|t-1}^n \end{pmatrix} = \begin{pmatrix} p_t^n + T v_t^n + \frac{T^2}{2} (R_t^{nb} y_{a,t} + g^n) \\ v_t^n + T (R_t^{nb} y_{a,t} + g^n) \\ \mathbf{0}_{3 \times 1} \end{pmatrix}, \quad (9)$$

$$\tilde{q}_{t|t-1}^{nb} = \tilde{q}_{t-1|t-1}^{nb} \odot \exp_q \left(\frac{T}{2} y(\omega, t-1) \right), \quad (10)$$

$$P_{t|t-1} = F_t P_{t|t-1} F_t^\top + G_{t-1} Q G_{t-1}^\top \quad (11)$$

with

$$G_{t-1} = \begin{pmatrix} T I_6 & \mathbf{0} \\ \mathbf{0} & \tilde{R}_{t|t-1}^{nb} \end{pmatrix}, \quad Q = \begin{pmatrix} I_6 & \mathbf{0} \\ \mathbf{0} & \Sigma_\omega \end{pmatrix}, \quad F_t = \begin{pmatrix} I_3 & T I_3 & -\frac{T^2}{2} (R_{t-1}^{nb} [y_{a,t} \times]) \\ \mathbf{0} & I_3 & -T (R_{t-1}^{nb} [y_{a,t} \times]) \\ \mathbf{0} & \mathbf{0} & I_3 \end{pmatrix}.$$

(b) Measurement Update

$$\hat{\eta}_t^n = K_t (y_t - \hat{y}_{t|t-1}), \quad (12)$$

$$\tilde{P}_{t|t} = P_{t|t-1} - K_t S_t K_t^\top, \quad (13)$$

with

$$y_t = \begin{pmatrix} y_{m,t} \\ y_{p,t} \\ y_{y,t} \end{pmatrix}, \quad \hat{y}_{t|t-1} = \begin{pmatrix} \tilde{R}_{t|t-1}^{bn} m^n \\ \sin^{-1}([\tilde{R}_{t|t-1}^{nb}]_{3,2}) \\ \sin^{-1}([\tilde{R}_{t|t-1}^{nb}]_{3,1}) \end{pmatrix},$$

$$S_t = H_t P_{t|t-1} H_t^\top + R, \quad K_t = P_{t|t-1} H_t S_t^{-1}$$

$$H_t = \begin{pmatrix} \mathbf{0} & \tilde{R}_{t|t-1}^{bn} [m^v \times] \\ \mathbf{0} & \frac{1}{\sqrt{1 - [\tilde{R}_{t|t-1}^{nb}]_{3,2}}} ([\tilde{R}_{t|t-1}^{nb}]_{2,2}, \tilde{R}_{t|t-1}^{nb} [y_{a,t} \times]_{2,1}, 0) \\ \mathbf{0} & \frac{1}{\sqrt{1 - [\tilde{R}_{t|t-1}^{nb}]_{3,1}}} ([\tilde{R}_{t|t-1}^{nb}]_{2,1}, \tilde{R}_{t|t-1}^{nb} [y_{a,t} \times]_{1,1}, 0) \end{pmatrix} \quad R = \begin{pmatrix} \Sigma_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Sigma_y \end{pmatrix}$$

(c) Relinearize

$$\tilde{q}_{t|t}^{nb} = \exp_q \left(\frac{\hat{\eta}_t^n}{2} \right) \odot \tilde{q}_{t|t-1}^{nb} \quad (14)$$
-

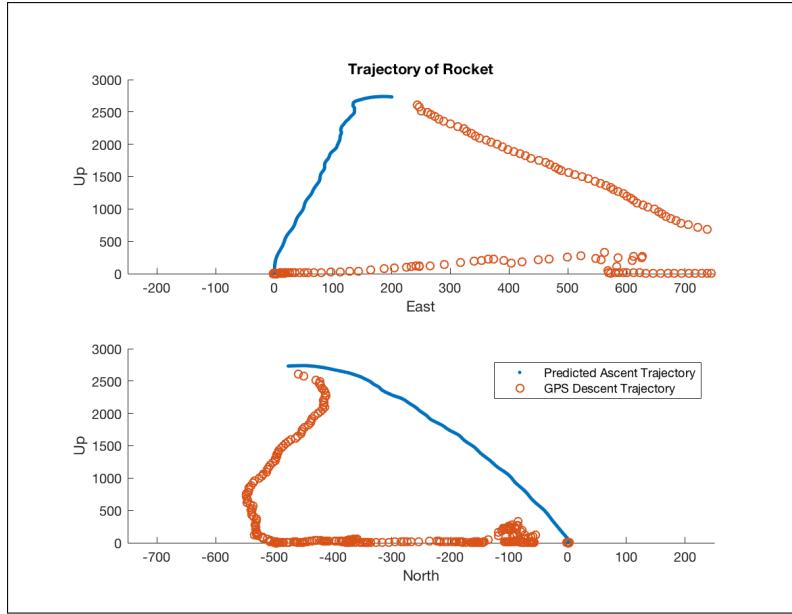


Figure 6: Kalman-filtered trajectory of the rocket.

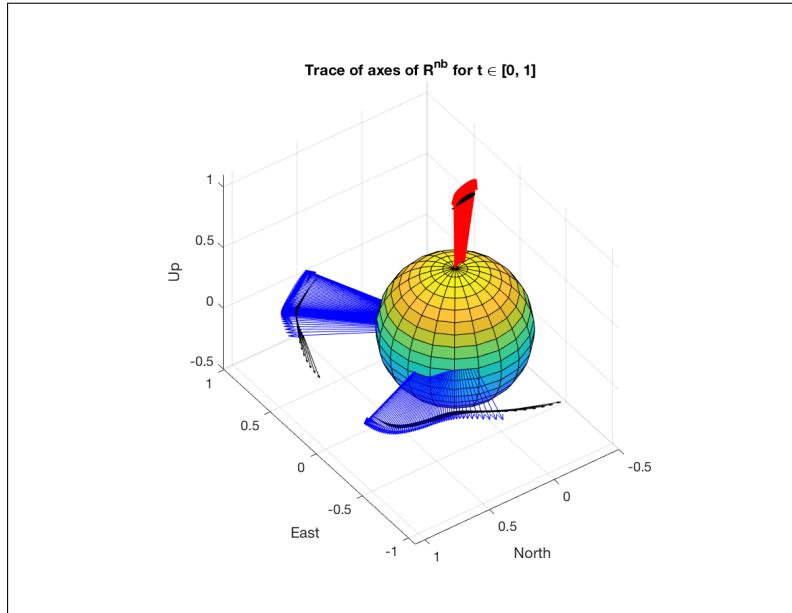


Figure 7: Trace of the three columns of R^{nb} during the first second of flight.

dynamics update and then the Kalman measurement update, as

$$P_{t|t-1} = F_t P_{t|t-1} F_t^\top + G_{t-1} Q G_{t-1}^\top$$

$$\tilde{P}_{t|t} = P_{t|t-1} - K_t S_t K_t^\top.$$

Because the system's position was not directly observable, the dynamics update dominated the behavior of the covariance matrix, resulting in approximately quadratic growth. Additionally, the model was highly sensitive to the initial uncertainty in the position estimate.

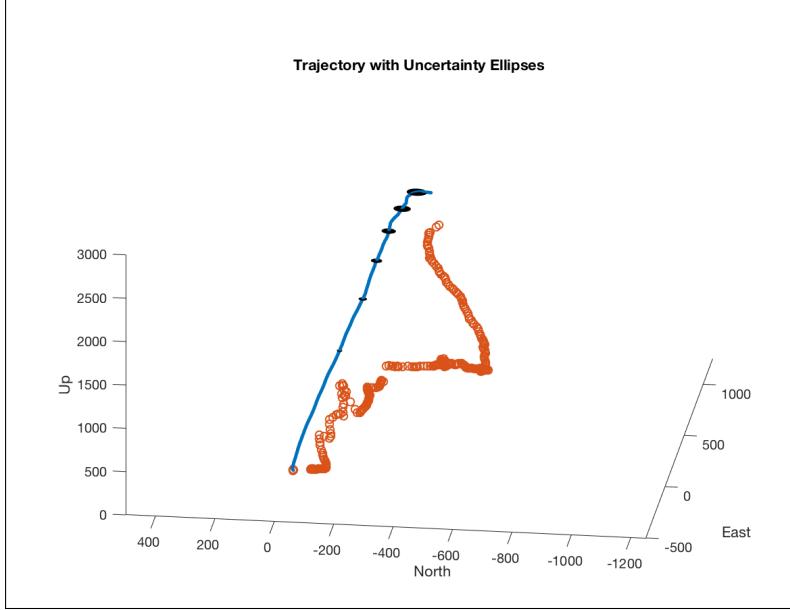


Figure 8: Trajectory including growing ellipses of uncertainty associated with positional covariance.

7 Conclusion

The algorithm shown here achieved a reasonable estimate of the rocket's trajectory, while operating on the $SE(3)$ manifold. Understanding the derivation of these results has relied heavily on concepts from the study of differential geometry.

Future fruitful work would continue draw on concepts from the study of differential geometry. Of particular interest are:

- a lower dimensional parametrization of the system configuration
- an efficient method to integrate arbitrary distributions for measurement and state transition probabilities
- a computationally efficient approach to broader problem of inertial-visual simultaneous localization and mapping

A Data Handling

Our data-processing pipeline must include the following.

1. Find initial GPS location on the launch pad.
2. Convert GPS data (not available during ascent) into meters from origin
3. Apply laboratory bias-calibrations to raw accelerometer, gyroscope, and magnetometer data.
4. Flip magnetometer x, y axes to get right-handed system
5. Scale magnetometer dimensions individually to normalize to $(-1, 1)$.
6. Scale magnetometer readings so that at every step t the L_2 norm is 1.
7. Get initial magnetometer orientation from the pre-launch sample mean. Also get sensor variance.
8. Resample magnetometer using smooth interpolation.
9. Convert gyroscope data to radians/sec.
10. Convert accelerometer data to meters/sec using local gravitational constant.
11. Get acceleration baseline from pre-launch sample mean. Also get sensor variance.
12. Resample accelerometer and gyroscope data.
13. Determine the global pitch and yaw corresponding to each image.
14. Smooth the roll using complementary filter.
15. Determine initial orientation using TRIAD method.

This sequence of pre-processing steps should yield a unified set of relevant time-series data for sensor fusion. However, item 13 deserves more elaboration. We do so below.

A.1 Computer Vision Algorithm for Horizon Estimation

An algorithm to automatically detect the horizon was implemented in OpenCV. The algorithm works by translating the image to an HSV (hue, saturation, and value) colorspace and identifying the sky. The edges of the sky region are then identified, and a straight line is fitted to the horizon. More specifically, the algorithm

1. Makes a mask that includes the sky but not the ground, using an HSV transform and thresholding.
2. Finds the edges of that mask using Canny edge detection.
3. Picks the longest edge. Assumes that contains the horizon.
4. Crops the image to only examine the central rectangle (and not the edges of the circular window).
5. Uses Hough line transform to identify a long straight line.

An example set of computer vision intermediate steps is shown in Fig. 9.

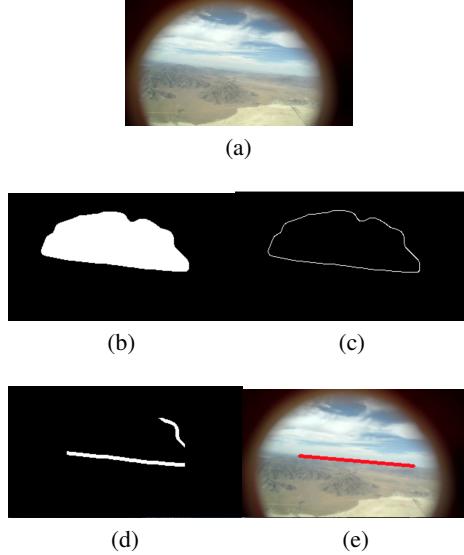


Figure 9: Computer vision algorithm for identifying the horizon in an image.

A.2 A General Approach to Retrieving Pose from Images

The problem of estimating the rocket’s attitude from camera output is governed by the underlying projective geometry of the camera-world system. Consider the elementary graphics equation relating individual points in 3D space to camera pixels,

$$\mathbf{y} = f(\mathbf{x}) \quad (15)$$

$$= H(R|\mathbf{t})\mathbf{x}. \quad (16)$$

Here, $\mathbf{y} = (x_{image}, y_{image}, 1)^T$ gives the position of the image on the camera plane, and is a function of $\mathbf{x} = (x_{object}, y_{object}, z_{object}, 1)^T$, the position of the object. H is the lens distortion matrix relating points on the ideal image plane to the actual pixel output. R is the rotation matrix between the camera orientation and global coordinates, and \mathbf{t} is the translation vector $(x, y, z, 1)^T$ defining the position of the camera.

Eq. 15 , implies a general approach to the problem of visual pose estimation. The camera pose matrix $(R|\mathbf{t})$, which is an element of $M_{4 \times 3}$, may be estimated by the correct identification of five spatially distinct features. This technique is known as “direct linear transformation”. That approach, which is akin to triangulation, is subject to the same sensitivity problem as conventional 2D triangulation: the features must be separated by a large angular distance in order to properly identify the camera pose within reasonable error. The addition of more point pairs increases the certainty of the estimation.

The problem of extracting the pose of a camera from a single view is common in the computer vision and robotics literature, and the general approach described above has been efficiently implemented in the limiting case of structure-from-motion as “visual odometry” and “visual simultaneous localization and mapping”. These approaches are very clever, but we implemented a simpler scheme to recover only pitch and yaw instead of the full pose.

By inspection of Figure 2, it can be seen that the slope and height of the horizon in a camera image represent respectively the pitch and yaw of the rocket. If the rocket pitches into the page, then the horizon will slope clockwise in the starboard camera image, and if the rocket yaws to the left, then the horizon will rise in the page; the opposite is true for the port camera.

We make the simplification that the lens distortion matrix H can be neglected, which is reasonable given that the fisheye lens results in only minimal distortion of lines passing near the center of the image (Fig. 1. Thus, in theory, raw camera output can be converted to rocket pitch and yaw measurements by identifying the slope and height of the horizon in the image, inferring the corresponding camera attitude, and correcting for the relative rotation between the camera and rocket. The specifics of this technique are detailed in Section 4.1

References

- [1] “Using Inertial Sensors for Position and Orientation Estimation - Now Foundations and Trends books.” [Online]. Available: <https://ieeexplore.ieee.org/document/8187588>. [Accessed: 03-Dec-2019].
- [2] R. Grover. Brown and P. Y. C. Hwang, Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, 3rd ed. New York: Wiley, 1997.
- [3] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics. Cambridge, Mass.: MIT, 2005.