Sensor-Based Image Stabilization for High-Altitude Weather Balloons

Corey Hoard

December 12, 2014

Introduction

High altitude weather balloons often employ onboard cameras to track their progress and record environmental information. However, the chaotic movement inherent in balloon-based travel creates shaky, wildly moving footage that is often disorienting and impossible to easily analyze. While various physical methods can be employed to mitigate camera movement, it is impossible to completely eliminate.

I propose that onboard inertial measurement sensors can be placed adjacent to the cameras for the purposes of stabilizing video as a post-process procedure. To this end, I derived an algorithm for compensating camera motion using motion and orientation data taken from sensors on a high-altitude weather balloon.

Theory

The onboard inertial measurement sensors I chose to install provide three metrics: angular velocity, acceleration, and magnetic field strength, each in three dimensions. This data, along with GPS position and movement data, was time-stamped then logged to a memory card. The data was to be extracted after the capsule was recovered, then synchronized with the video data. The algorithm presented in the body of this paper would be applied to the inertial data–measured from a rotating frame of reference–in order to extract a single orientation vector representing the camera's direction vector in a fixed earth frame of reference. The pixel data would be mapped onto the interior of a sphere, transforming each pixel's direction vector to a fixed frame of reference for easier viewing.

Sensor Fusion Algorithm

A sensor fusion algorithm is needed to effectively combine each of the sensors. Each inertial metric provides part of the requisite information, but has its own drawbacks: the gyroscope provides accurate high frequency, short-term data and is less prone to noise, but operates on a relative frame of reference, and suffers from long-term drift inaccuracies; measurements of the earth's magnetic friend provide an absolute frame of reference and low frequency, long-term data, but are prone to distortion and high noise. The accelerometer data was not used as the capsule undergoes too much lateral translation to get an accurate measurement of gravitational acceleration. The sensor fusion algorithm combines the short-term accuracy of the gyroscopic measurements with the stabilizing effects of the magnetometer.

I started with a basic two-dimensional Kalman filter, seen below, and worked on translating it to three dimensions. I rewrote the traditional form using complex numbers, and intended on using a similar equation with quaternions for the three-dimensional version. This proved to be excessively complex, so I instead converted to direction-cosine-matrices to represent orientation.

2D Kalman: $\theta_i - \theta_{i-1} = \alpha(\theta_{accel} - \theta_{i-1}) + (1+\alpha)(\Delta t)(\theta'_{gyro})$
2D Kalman, Complex (Partial): $z_{i+1} = z_i e^{j\omega\Delta t}$
2D Kalman, Matrix (Partial): $\boldsymbol{R}_{i+1} = \boldsymbol{R}_i \boldsymbol{G}(\Delta t)$

Using information from the papers "Nonlinear Complementary Filters on the Special Orthagonal Group" (Mahony, Hamel, Pflimin) and "Direction Cosine Matrix IMU: Theory" (Premerlani, Bizard), I formulated an algorithm for calculating an orientation vector given IMU data:

Store stationary magnetometer data as $\vec{M}_{ref} \in \Re^3$

Initialize rotation matrix $\mathbf{R}_0 = \mathbf{I}^3$, correction factor $\vec{\omega}_{corr} = \vec{0}$

For each loop iteration,

1. Read in gyroscope data as $\vec{\omega} \in \Re^3$, calculate $\partial\vec{\theta} = (\vec{\omega}_{gyro} + \vec{\omega}_{corr})\partial t$

2. Populate rotation approximation matrix $\mathbf{G} = \begin{bmatrix} 1 & -\partial\theta_z & \partial\theta_y \\ \partial\theta_z & 1 & -\partial\theta_x \\ -\partial\theta_y & \partial\theta_x & 1 \end{bmatrix}$

3. Update $\mathbf{R}_{i+1} = \mathbf{R}_i\mathbf{G}$

4. Find orthagonality error $err_{orth} = r_{xx}r_{yx} + r_{xy}r_{yy} + r_{xz}r_{yz}$

5. Calculate rotations needed to orthagonalize $(\vec{X}_{orth}, \vec{Y}_{orth}, \vec{Z}_{orth} = \vec{X}_{orth} \times \vec{Y}_{orth})$

6. Distribute orthagonalization to each axis and renormalize

7. Find computed magnetometer vector $\vec{M}_{comp} = \vec{M}_{ref}\mathbf{R}_i$

8. Compare to measured value, get correction factor $\vec{C} = \vec{M}_{comp} \times \vec{M}_{meas}$

9. Use a PI controller for correction factor: $\vec{\omega}_{Icorr} = \vec{\omega}_{Icorr} + K_i\vec{C}\partial t$; $\vec{\omega}_{corr} = K_p\vec{C} + \vec{\omega}_{Icorr}$

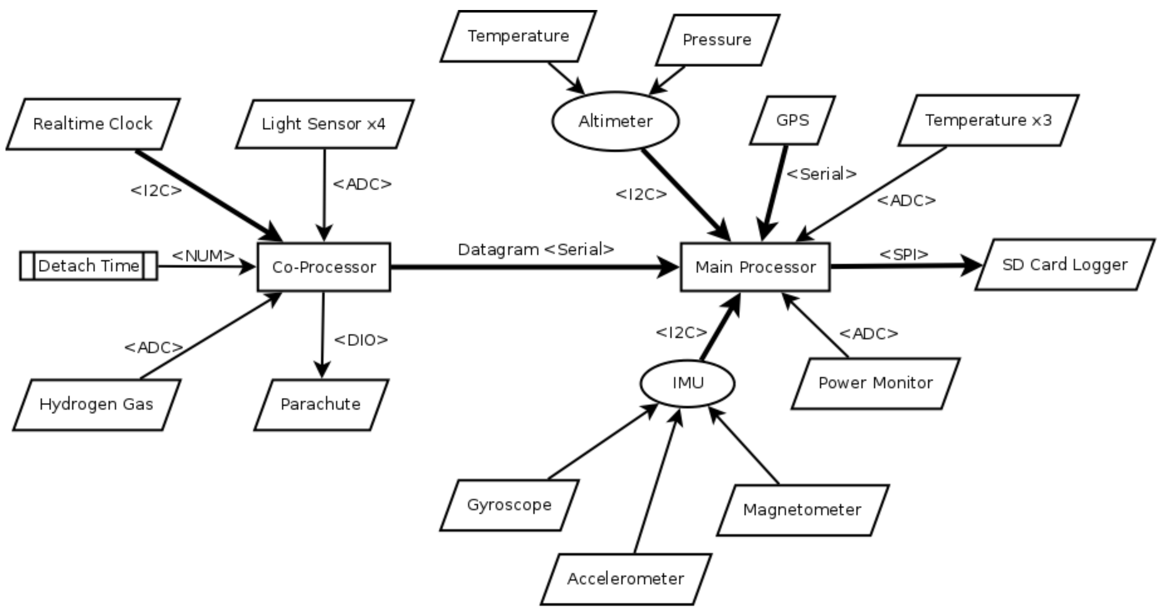
Generate vector for each pixel in video based on body frame of reference

Use $\mathbf{R}(t)$ to transform pixel to earth frame of reference, plot pixels as colored voxels in $\Re^3$

(Note: pixel vectors are calculated using the lens's field of view and the inverse tangent function to convert from normalized linear pixel space to angular space, then transformed using **R**.)

Data Collection Module

I also worked on the data collection module used on the high-altitude balloon. The system used two parallel processors, each collecting data from a set on sensors. These were combined into a single data stream then logged to an SD card. Below is a diagram of the data flow used on the final electronics board. Data from the inertial sensors is logged at 50Hz, while the remaining sensors are recorded at 4Hz to reduce processor load. Video data was recorded from two GoPro cameras at 60Hz.



Conclusion

Unfortunately, the weather balloon that I was using to collect video and motion data landed in the ocean and was unrecoverable. I intend on generating simulated data or capturing sensor feeds from my phone's sensors. I will then implement the equations in MATLAB, calculate orientation vectors, synchronize to video data, then map onto a sphere.