

# BAROMETRIC AND GPS ALTITUDE SENSOR FUSION

Vadim Zaliva and Franz Franchetti

Carnegie Mellon University

**Index Terms**—sensor fusion, GPS, barometric, altitude

## ABSTRACT

The altitude of a moving vehicle as reported by GPS suffers from intermittent errors caused by temporary obstruction of the satellites by buildings, mountains, etc. Additionally, it is affected by systematic errors caused by multipath effects, ionospheric and tropospheric effects, and other hardware design limitations and natural factors.

Atmospheric pressure, measured by a portable barometric sensor, could also be used to determine altitude, is not susceptible to problems caused by obstruction of satellites, and can provide reliable measurements outdoors even in urban and mountainous regions.

In this paper, we propose an algorithm which improves accuracy and provides tighter confidence bounds of altitude measurements from a mobile phone (or any device equipped with GPS and barometric sensors) by means of sensor fusion techniques without the need for calibration.

Our experiments have shown that the proposed algorithm provides more accurate measurements with tighter confidence bounds compared to using either of the two sensors, barometric or GPS, alone.

## 1. INTRODUCTION

Sensor fusion techniques are used to combine information from multiple sources (sensors) with the goal of improving accuracy or reliability over the results from an individual source. A good introduction into the subject of sensor fusion is given in [1], and more details on its mathematical methods can be found in [2].

In this paper, we will consider a realistic scenario where a vehicle (or a pedestrian) is moving outdoors on the earth's surface. The movement trajectory is assumed to be smooth in altitude and not to feature sudden altitude jumps. The vehicle must be equipped with GPS and atmospheric pressure sensors, such as can be found in many modern mobile phones, like *Samsung Galaxy S4* and *Google Nexus 4*.

This material is based on research sponsored by DARPA under agreement number FA8750-12-2-0291. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

We will apply a sensor fusion approach to this particular scenario. Our algorithm will use knowledge of how these two particular sensors behave. Some of the assumptions about altitude sensor behaviour are shown in the following table:

Feature	GPS	Barometric
Affected by obstructions	Yes	No
Always available	No	Yes
Biased	No	Yes
Bias drifts with weather	No	Yes
Typical accuracy	5m[3]	0.17-0.5m[4]
Accuracy reported by sensor	Yes	No

Our approach is based on the premise that barometric altitude measurements provide more accurate information on relative altitude changes compared to GPS. It does have measurement noise, but the magnitude of this noise is typically significantly smaller than that of GPS measurements. However, an accurate estimation of absolute altitude based on measurements of barometric pressure requires calibration. Such calibration is valid for only limited time periods due to natural changes in the atmospheric pressure caused by weather.

However we can use GPS to dynamically “calibrate” barometric measurements to reflect true absolute altitude. Using statistical properties of GPS and barometric measurements as well as some assumptions about the effects of weather, we derive confidence intervals for fused measurements. Then, using these intervals, we define a cost function which is used to optimize the temporal window size used in fusion.

In related work Yao[5] integrated barometric altitude measurements as a virtual satellite into GPS position calculations. Our approach does not replace GPS position calculation and could be used on top of other GPS accuracy improvement models[6], as well as with other GNSS systems like GLONASS. In other related work Gebre-Egziabher[7] presents an empirical barometric altitude confidence bound based on historical meteorological data without attempting to estimate the altitude.

We claim that our algorithm achieves the following:

1. Resulting fused altitude measurements will typically

have tighter tolerance intervals, compared to GPS sensor alone.

2. Resulting fused altitude measurements will typically have more accurate absolute values, compared to barometric sensor alone.
3. The algorithm provides not only fused absolute altitude values, but also statistical confidence intervals.

Additionally benefits of our approach are that the altitude could be estimated accurately even when GPS signal is temporarily lost (dead zones). This is an online algorithm which could be used in real-time and the calibration step is not required.

## 2. THE ALGORITHM

### Sensor Fusion

Let the true (unknown) altitude at time  $t$  be  $a(t)$ . We will use  $g(t)$  to denote GPS altitude measurements at time  $t$ . According to Android API documentation[8] they are normally distributed:  $g \sim \mathcal{N}(a, \sigma_g^2)$ .

Without loss of generality, we will assume GPS measurements are sampled at fixed intervals with the sample number  $i$  being an altitude  $g[i]$  and standard deviation  $\sigma_g[i]$ . The samples are numbered in reverse order with  $i = 1$  being the most recent. The algorithm could be easily adopted for variable rate sampling.

Similarly, we will model the barometric altitude as  $b \sim \mathcal{N}(a + \Delta, \sigma_b^2)$ . Here,  $\Delta$  denotes the unknown bias of barometric altitude measurements. Unlike GPS, the variance  $\sigma_b^2$  is not provided by the sensor and is unknown.

We will start by estimating the parameters of the  $g(t)$  distribution using MLE based on the last  $n$  samples. The *pooled variance* and the *sample mean* (biased) are:

$$s_g^2 = \frac{1}{n} \sum_{i=1}^n \sigma_g^2[i] \quad (1)$$

$$\mu_g = \frac{1}{n} \sum_{i=1}^n g[i] \quad (2)$$

The mean has *standard error of the mean (SEM)*  $\frac{s_g}{\sqrt{n}}$  and is normally distributed around the true altitude:  $\mu_g \sim \mathcal{N}(a, \frac{s_g^2}{n})$

Next, we will estimate the parameters of the  $b(t)$  distribution using MLE based on the last  $m$  samples.

Here,  $m$  and  $n$  denote the number of GPS and barometric samples, respectively, for a given time interval. Generally speaking,  $m \leq n$ , because the barometric measurements are always available, even when GPS measurements are missing.

Sample mean and sample variance (biased) are

$$\begin{aligned} \mu_b &= \frac{1}{n} \sum_{i=1}^n b[i] \\ s_b^2 &= \frac{1}{n} \sum_{i=1}^n (b[i] - \mu_b)^2 \end{aligned} \quad (3)$$

Using SEM, we describe the distribution for  $\mu_b \sim \mathcal{N}(a + \Delta, \frac{s_b^2}{n})$ .

Now, estimate the value of  $\Delta$  as:  $\hat{\Delta} = \mu_b - \mu_g$ . This estimated value is the sum of two normally distributed random variables. Such a sum will be also distributed normally:  $\hat{\Delta} \sim \mathcal{N}(\Delta, \frac{s_b^2}{m} + \frac{s_g^2}{n})$ .

We obtain the corrected altitude for sample  $i$  as:  $\hat{a}[i] = b[i] + \hat{\Delta}$ . This value as the sum of two normally distributed random variables will have distribution:

$$\hat{a}[i] \sim \mathcal{N}(\mu_b - \Delta, s_b^2 + \frac{s_b^2}{m} + \frac{s_g^2}{n}) \quad (4)$$

### Ordinary Least Squares Variance Correction

The sample variance estimation in Equation (3) estimates the spread of  $b$  values relative to the horizontal line  $b = \mu_b$  which corresponds to a vehicle moving on a horizontal plane. However, if the vehicle ascends or descends (e.g. car climbing a hill), this method is not suitable for estimation of Gaussian noise in  $b$ . Intuitively, the correct approach would be to estimate the variance orthogonal to the direction of the movement.

First, we fit a straight line through the  $\{b_1 \dots b_m\}$  using the *ordinary least squares* method. According to the Gauss-Markov theorem, this is the best linear unbiased estimator (BLUE). This would give us  $\alpha$  and  $\beta$  parameters of a linear model  $\hat{b} = \beta t + \alpha$ . Residual errors of such fit are  $\epsilon = b - \hat{b}$ .

Under the *normality assumption*, the first and second moments of the  $\beta$  distribution could be estimated as[9]:

$$\mu_\beta = \beta \quad (5)$$

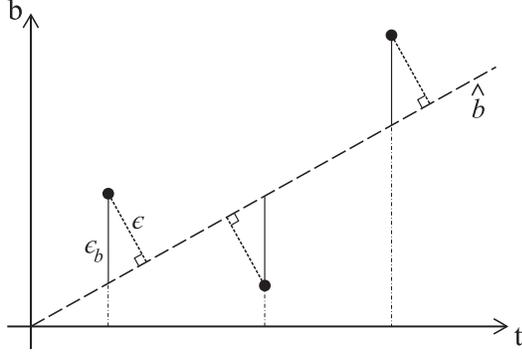
$$s_\beta = \sqrt{\frac{\frac{1}{m-2} \sum_{i=1}^m \epsilon[i]^2}{\sum_{i=1}^m (t[i] - \mu_t)^2}} \quad (6)$$

To determine how  $\epsilon$  errors affect  $b$  measurements, we need to project them vertically to the fitted line as shown in Figure 1:

$$\epsilon_b = \epsilon \sqrt{1 + \beta^2} \quad (7)$$

Now, we will estimate the variance of  $\epsilon_b$ . To do so, let us set  $k(\beta) = \sqrt{1 + \beta^2}$ , so we can write:

$$\epsilon_b = \epsilon k \quad (8)$$



**Fig. 1.** Geometric interpretation of  $\epsilon$  and  $\epsilon_b$ .

Since  $\epsilon$  and  $k$  are independent, the variance of their product is [10]:

$$\begin{aligned} \text{var}(\epsilon_b) &= \text{var}(\epsilon k) \\ &= [E(\epsilon)]^2 \text{var}(k) + [E(k)]^2 \text{var}(\epsilon) + \text{var}(\epsilon) \text{var}(k) \end{aligned} \quad (9)$$

In the equation above,  $E(\epsilon)$  and  $\text{var}(\epsilon)$  could be estimated using sample statistics as:

$$\mu_\epsilon = \frac{1}{m} \sum_{i=1}^m \epsilon[i] = \frac{1}{m} \sum_{i=1}^m (b[i] - \beta t[i] - \alpha) \quad (10)$$

$$\begin{aligned} s_\epsilon &= \sqrt{\frac{1}{m} \sum_{i=1}^m (\epsilon[i] - \mu_\epsilon)^2} \\ &= \sqrt{\frac{1}{m} \sum_{i=1}^m (b[i] - \beta t[i] - \alpha - \mu_\epsilon)^2} \end{aligned} \quad (11)$$

Treating  $k$  as a function of random variable  $\beta$ , it is possible to estimate the first and the second moments,  $E(k)$  and  $\text{var}(k)$  respectively, using first order Taylor's expansions [11]:

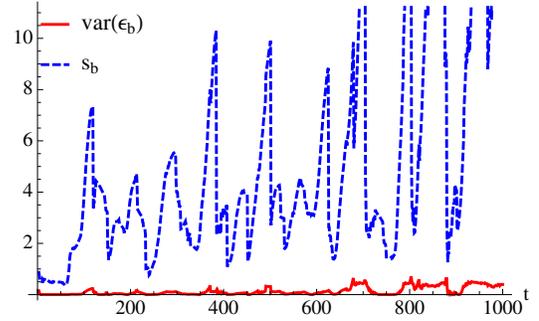
$$E[k(\beta)] \approx k(\mu_\beta) + \frac{k''(\mu_\beta)}{2} \sigma_\beta^2 = \frac{2(\mu_\beta^3 + \mu_\beta) + s_\beta^2}{2(\mu_\beta^2 + 1)^{3/2}} \quad (12)$$

$$\text{var}[k(\beta)] \approx (k'(\mu_\beta))^2 \sigma_\beta^2 = \frac{\mu_\beta^2 s_\beta^2}{\mu_\beta^2 + 1} \quad (13)$$

Substituting (12,13) into (9) and using values obtained by (5,6,10,11), we get the final formula for  $\text{var}(\epsilon_b)$  which we will be using instead of  $s_b$  in (4):

$$\text{var}(\epsilon_b) \approx \frac{\mu_\beta^2 s_\beta^2 \mu_\epsilon^2}{\mu_\beta^2 + 1} + \frac{\mu_\beta^2 s_\beta^2 s_\epsilon}{\mu_\beta^2 + 1} + \frac{s_\epsilon (2(\mu_\beta^3 + \mu_\beta) + s_\beta^2)^2}{4(\mu_\beta^2 + 1)^3} \quad (14)$$

Experimental results of variance correction using the method described above are shown in Figure 2. The dashed blue line shows  $s_b$  (uncorrected), and the red line shows  $\text{var}(\epsilon_b)$  (corrected):



**Fig. 2.** Results of variance correction using ordinary least squares.

### Atmospheric Pressure Drift

So far, we have been assuming that while atmospheric pressure has Gaussian noise, it does not change over time, if the vehicle is stationary. In real life, however, barometric pressure fluctuates with atmospheric conditions. Luckily these fluctuations are rather slow, compared to the pressure changes caused by the vehicle ascending or descending.

We used a well-known empirical formula, used in aerospace industry [12] to convert barometric pressure  $p$  to barometric altitude  $b$ :

$$b(p) = 44330.8 - 4946.54p^{0.1902632} \quad (15)$$

Let us assume that the maximum natural change of atmospheric pressure  $p_h$  is 400 hPa per hour.

The function in Equation (15) is non-linear, but continuous and monotonically decreasing with  $p$ . We estimate the maximum drift of barometric altitude from the initial value  $p_0$  over time  $t$  as:

$$\begin{aligned} \Delta b(p_0, t) &= \max \left( b\left(p_0 - \frac{t \cdot p_h}{3600}\right) - b(p_0), \right. \\ &\quad \left. b(p_0) - b\left(p_0 + \frac{t \cdot p_h}{3600}\right) \right) \end{aligned} \quad (16)$$

### Optimization

One of our goals was to minimize the tolerance interval of the estimated altitude  $\hat{a}[i]$ . For a normally distributed variable, the tolerance interval is usually expressed as the number of standard deviations  $D$ . The probability that a value lies less than  $D$  standard deviations away from the mean is  $\text{erf}\left(\frac{D}{\sqrt{2}}\right)$ .

For a given  $D$ , minimizing the standard deviation from Equation (4):

$$\sigma_{\hat{a}} = \sqrt{s_b^2 + \frac{s_b^2}{m} + \frac{s_g^2}{n}}$$

will shrink the tolerance region.

Before trying to minimize this function, let us consider the physical meanings of  $m$  and  $n$ . These variables will control the number of previous samples we will consider. Intuitively, we want to increase these values to get a smaller variance. However, our algorithm acts as a low-pass filter on  $g$ , and bigger values of  $n$  would cause us to miss subtler changes in the GPS-measured altitude.

As the value of  $m$  increases, the duration of the time interval we are considering also increases. Let us denote it as  $\Delta t = t_1 - t_m$ . Using Equation (16), we estimate the maximum potential natural drift of the barometric altitude over time  $t$ . Thus, the tolerance region of the estimated altitude value needs to be extended by potential barometric altitude drift. For a given  $\hat{a}[i]$ , the tolerance region would extend to:  $\pm \left( D\sigma_{\hat{a}} + \frac{\Delta b(p_i, t_i - t_{i+m})}{2} \right)$ , so the cost function we need to minimize is:

$$J(m, n; i, D) = D \sqrt{s_b^2 + \frac{s_b^2}{m} + \frac{s_g^2}{n}} + \frac{\Delta b(p_i, t_i - t_{i+m})}{2} \quad (17)$$

Since we only have access to sampled values of  $b$  and  $g$ , this cost function for each sample could only be minimized empirically through a simple iterative algorithm:

1. Vary value  $m$  from experimentally chosen  $m_{min}$  and  $m_{max}$  values. Our experiments have shown that the respective values, 10 and 200, are sufficiently big, as the cost maxima typically lies below that.
2. Find the number of GPS measurements  $n$  in the time interval corresponding to each  $m$ .
3. For each  $m$  and corresponding  $n$  estimate error using Equation (17)
4. Choose  $m$  corresponding to the smallest error and estimated altitude using Equation (4)

Typical plot of  $J$  for different values of  $m$  is shown in Figure 3.

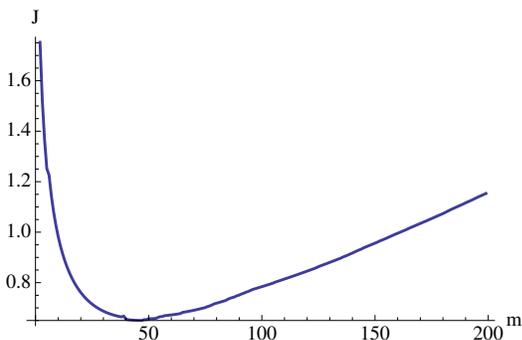


Fig. 3. Dependency of altitude estimation error from  $m$ .

### 3. EXPERIMENTAL RESULTS

To test our algorithm, we recorded multiple datasets using the *Samsung Galaxy S4* and *Google Nexus 4* phones. Datasets were recorded while hiking, bicycling, and riding in a car in both urban and mountainous environments in and around the San Francisco Bay Area. In order to debug the algorithm on flat terrain, additional datasets were recorded on board a boat cruising in San Francisco Bay.

Typical experimental results are shown in Figure 4. GPS altitude measurements are shown in dashed black and barometric in dotted green. A 68% confidence bounds of GPS measurements, as reported by sensor, is shown in yellow. The altitude corrected by our algorithm is shown in blue, along with an estimated 68% confidence bounds, shown in pink.

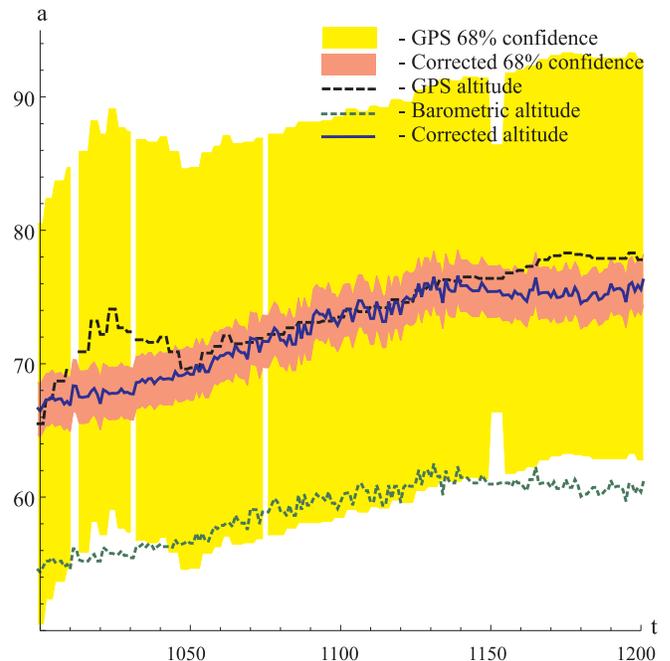


Fig. 4. Confidence bounds of GPS, barometric, and fused altitude measurements.

### Conclusion

Our experiments have shown that by using the proposed algorithms, we were able to obtain corrected altitude measurements which have confidence bounds on average 85% smaller than the confidence bounds of the original GPS measurements.

#### 4. REFERENCES

- [1] H B Mitchell, *Multi-Sensor Data Fusion: An Introduction*, 2007.
- [2] D L Hall and S A H McMullen, *Mathematical Techniques in Multisensor Data Fusion*, vol. 2 of *Artech House microwave library*, Artech House, 2004.
- [3] United States Department of Defense, “Global positioning system standard positioning service performance standard,” 2008.
- [4] BOSCH Sensortec, “BMP180 digital pressure sensor data sheet,” 2013.
- [5] Yi Yao, Zhi gang Huang, and Rui Li, “Data fusion algorithm of baro/gps based on delta-altitude,” in *Information, Computing and Telecommunication, 2009. YC-ICT '09. IEEE Youth Conference on*, 2009, pp. 522–525.
- [6] Xiaoguang Luo, *GPS Stochastic Modelling: Signal Quality Measures and ARMA Processes (Springer Theses)*, Springer, 2013.
- [7] D. Gebre-Egziabher, T. Walter, and P. Enge, “Improving GPS-based landing system performance using an empirical barometric altimeter confidence bound,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 127–146, Jan. 2008.
- [8] Google, “Android 4.3r1 *Location.getAccuracy()* method reference,” 2013.
- [9] F.S. Acton, *Analysis of straight-line data*, Dover books on intermediate and advanced mathematics. Dover Publications, 1959.
- [10] GW Bohrnstedt and AS Goldberger, “On the Exact Covariance of Products of Random Variables,” *Journal of the American Statistical Association*, 1969.
- [11] A.H.S. Ang and W.H. Tang, *Probability Concepts in Engineering: Emphasis on Applications to Civil and Environmental Engineering*, Wiley, 2006.
- [12] Portland State Aerospace Society, “A Quick Derivation relating altitude to air pressure,” Tech. Rep., Portland State Aerospace Society, 2004.