Tracking Drone Orientation with Multiple GPS Receivers

Mahanth Gowda
University of Illinois (UIUC)
Urbana, IL, USA
gowda2@illinois.edu

Justin Manweiler
IBM Research
Yorktown Heights, NY, USA
jmanweiler@us.ibm.com

Ashutosh Dhekne
University of Illinois (UIUC)
Urbana, IL, USA
dhekne2@illinois.edu

Romit Roy Choudhury
University of Illinois (UIUC)
Urbana, IL, USA
croy@illinois.edu

Justin D. Weisz
IBM Research
Yorktown Heights, NY, USA
jweisz@us.ibm.com

ABSTRACT

Inertial sensors continuously track the 3D orientation of a flying drone, serving as the bedrock for maneuvers and stabilization. However, even the best inertial measurement units (IMU) are prone to various types of correlated failures. We consider using multiple GPS receivers on the drone as a failsafe mechanism for IMU failures. The core challenge is in accurately computing the relative locations between each receiver pair, and translating these measurements into the drone's 3D orientation. Achieving IMU-like orientation requires the relative GPS distances to be accurate to a few centimeters – a difficult task given that GPS today is only accurate to around 1-4 meters. Moreover, GPS-based orientation needs to be precise even under sharp drone maneuvers, GPS signal blockage, and sudden bouts of missing data. This paper designs *SafetyNet*, an off-the-shelf GPS-only system that addresses these challenges through a series of techniques, culminating in a novel particle filter framework running over multi-GNSS systems (GPS, GLONASS, and SBAS). Results from 11 sessions of 5-7 minute flights report median orientation accuracies of 2° even under overcast weather conditions. Of course, these improvements arise from an increase in cost due to the multiple GPS receivers, however, when safety is of interest, we believe that tradeoff is worthwhile.

CCS Concepts

•**Computer systems organization** → *Embedded and cyber-physical systems;*

Keywords

Differential GPS; Carrier Phases; UAV; Drones; IMU

Mobicom 2016 October 03-07 2016, New York City, NY, USA

©2016 ACM. ISBN 978-1-4503-4226-1/16/10...$15.00
http://dx.doi.org/10.1145/2973750.2973768

1. INTRODUCTION

Despite excitement, we are not yet ready for a world of drones flying among people. Challenges of liability, regulation, and limited societal acceptance all arise because drones are still too dangerous for responsible use in public places. A capstone obstacle is the untrustworthiness of inertial sensors [1–5].

A drone's inertial sensors, also called the Inertial Measurement Unit (IMU), is akin to a human's inner ear. Without a good awareness of orientation and balance, a drone cannot effectively modulate current to its rotors – even hovering becomes impossible. In the best case, the drone may not take off, or fall vertically from the sky. However, the worse case is when the drone tries to correct for its poor sensory inputs, and accelerates uncontrollably in an unintended direction with even greater force. For a package delivery drone measured in 10s of kilograms, this might be deadly.

A natural response is to install redundant IMUs (i.e., accelerometer, gyroscope, and compass). Given that each sensor is relatively small, light, and inexpensive, this is an easy fix – some commercial drones have already adopted this [6, 7]. Unfortunately, redundancy only addresses unreliable hardware. Correlated noise sources, including motor vibration, electromagnetic interference, and ambient ferromagnetic influences, remain as serious concerns [8–12].

To provide stronger IMU failover guarantees, we consider a GPS based approach *completely orthogonal to the fundamental nature of IMU sensing*. We install 4 GPS receivers at the corners of a single drone – upon IMU failure, we utilize these GPSs to estimate the drone's 3D orientation. Put differently, *we apply GPS to estimate pitch, roll, and yaw, without any inertial or magnetometer assistance*.

The drone's pitch, roll, and yaw can be expressed as a function of the pairwise 3D vectors between the GPS receivers – as the drone flies, these vectors change with respect to the Earth's reference frame. SafetyNet's core task is to precisely estimate these 3D vectors. To achieve IMU-like accuracy ($\approx 1^\circ$), the vectors need to be estimated at centimeter scale precision. This precision needs to be upheld constantly over time, across agile flight patterns, poor satellite SNR, and even short periods of missing data.

Recent research [13, 14] has improved on Differential GPS (DGPS) to achieve $\approx 15cm$ error for relative localization. This is adequate for many on-road vehicular applications, such as lane change detection, peer-to-peer car coordination, etc. However, the accuracy requirement for drone orientation is much higher. Multi-GPS orientation estimation techniques have been proposed for ships, planes and low dynamic drone flights [15–17]. However, extending them to commercial drones of small form factor and rapidly changing orientation is non-trivial and challenging. The demands are far stricter, both in terms of accuracy and time-granularity of tracking. Finally, the estimation techniques must be lightweight to be able to serve as a real-time IMU replacement during flights.

SafetyNet appropriately adopts ideas and techniques from the mature GPS literature, and builds over them to mitigate the challenges. The core additions, although closely interspersed with existing techniques, can be distilled as follows:

(1) Manipulating measurements across pairs of GPS receivers, satellites, and consecutive time points, to ultimately capture the 3D orientation of the drone from 2 different perspectives. Using these 2 perspectives to formulate an estimation problem, amenable to Kalman Filtering (KF). While GPS literature is fraught with KF and measurement manipulations (called "double differentials"), SafetyNet combines the double differentials in a way that is novel to the best of our knowledge.

(2) GPS phase measurements are composed of an unknown portion, called "integer ambiguity". In attempting to resolve this ambiguity, past work have adopted techniques akin to "hard decoding", where the most likely state-estimate is propagated across time. We design the equivalent of "soft decoding", whereby top-K possibilities of the ambiguity are propagated, each associated with an inferred probability. A particle filter is used to execute this idea – the particle filter degenerating back into the Kalman Filter when the ambiguity is resolved confidently.

(3) We break away from the classical particle filter approach and "adjust" the state of the particles based on available measurements. This speeds up convergence of the system, while requiring fewer particles (considerably reducing the computational complexity). As a result, the overall SafetyNet system lends itself to real time operation on today's drone hardware.

(4) Finally, SafetyNet is a complete system borne out of significant engineering effort, including carrier-phase outlier detection, integration of multi-satellite systems for robustness, automatic baseline calibration, etc.

Our evaluation platform is composed of a 3DR octocopter (8 rotor) mounted with 4 off-the-shelf NEO-M8T GPS receivers, all connected to a Raspberry Pi via USB. A GoPro camera is mounted in the underbelly of the drone, facing vertically downward – vision data serves as the ground-truth for both IMU and GPS. We present extensive experimental data from 11 aggressive flight sessions, performed under a wide range of weather conditions. Critically, our results demonstrate comparable accuracy to IMU even under the most aggressive aerial maneuvers within the capability of our drone.

Some Natural Questions

Is power consumption excessive with 4 GPS receivers? While GPS receivers are indeed considered power hungry on smartphones [18, 19], we note that the drone's physical flight requires 1000x more power, requiring a separate LiPO battery [20] weighing 804 grams. Hence the marginal increase in power from GPS is negligible.

Do GPS receivers add to the cost of drones? SafetyNet is mostly needed for bigger drones carrying heavier payloads (costly cameras, delivery packages). The cost of such drones and their payloads easily justify the additional GPS cost. Moreover, commercial drones already provide multiple GPSes for redundancy [21] – we believe the additional GPS cost should not be an issue.

What is the on-board IMU's orientation accuracy? How does it compare to SafetyNet? Results in this paper will demonstrate comparable accuracy distributions between SafetyNet and the on-board IMU, even at the 99^{th} percentile. To be specific, pitch and roll are slightly worse with SafetyNet, but yaw is slightly better. This level of accuracy is adequate as a fallback mechanism – upon IMU failure, the drone could avoid aggressive maneuvers and fly back safely to its base.

What if GPS also fails? Crash reports typically indicate IMU failures [4]. This is because engine vibrations cause drift in IMU sensors, accumulating error over time [8, 11, 12]. Drone electronics might interfere with magnetometers [9, 10, 22]. However, none of these error sources affect GPS. While GPS could still fail due to deliberate jamming or extreme weather conditions, the possibility of both failing is naturally lower.

The rest of the paper expands on these techniques, experimental findings, and contributions. We begin with an abridged technical primer on GPS processing (mostly derived from Differential GPS and related techniques), followed by system design, error mitigation, and an end-to-end system evaluation.

2. GPS FOUNDATIONS

We present GPS foundations from first principles and end with discussions on modern techniques. As a result, this section is long. However, given that this paper builds over core GPS algorithms, the material is necessary. We also believe the material is easy to follow.

2.1 Global Navigation Satellite System (GNSS)

GNSS is the generic name given to satellite systems that provide localization services to receiver's on earth. The Global Positioning System (GPS) [23] is one example of a GNSS, developed by the US Government during 1970-80s. GPS consists of a constellation of 31 satellites orbiting the earth at a height of 20,000 km. The satellites are simultaneously and continuously transmitting unique pseudo-random noise (PRN) sequences using CDMA at 2 different frequencies – 1575.42 (L1) and 1227.60 MHz (L2). They also broadcast ephemeris data using which the (satellite) position and time of transmission can be calculated. A GPS receiver on the ground localizes itself by trilateration, i.e., measuring and combining the time-of-flight (ToF) of PRNs from different satellites. Velocity is computed from the doppler shifts from each of the satellites.

GLObal NAvigation Satellite System (GLONASS) is another GNSS system launched by Russia in the 1980's [24]. Similarly, GALILEO [25] is an European GNSS system currently under development. Many GNSS receivers are capable of decoding signals from multiple GNSS systems, providing increased accuracy. This paper will also use such receivers and exploit the advantages of satellite diversity.

2.2 GPS Localization and Error Sources

A GNSS receiver on the ground can compute its 3D location, time, and velocity. The key idea is to measure various attributes of the arriving signal (e.g., time of flight, phase, etc.), and then apply statistical algorithms to estimate the errors and ambiguities in measurements. We describe below an overview of the techniques that underpin GPS; other GNSS systems rely on similar techniques.

Pseudorange

When a satellite transmits a signal, it includes the starting time of the transmission (obtained from its atomic clock). The ground receiver records the time of reception also using its less accurate local clock. The time-of-flight (ToF) is the difference between these timestamps. When multiplied by the speed of light, the result gives the *rough* distance to the satellite, called *pseudorange*.

$$\text{Pseudorange} = \text{ToF} * (\text{speed of light})$$

Of course, the measured ToF is inaccurate because the clock of a typical GPS receiver is not synchronized to the GPS satellites. The resulting error can be up to 300 km. In addition, when the GPS signal enters the Earth's atmosphere, it can get delayed due to refractions in the Ionosphere and Troposphere. A signal also passes through a multipath channel, adding more errors. Assuming the true range between a satellite s and receiver i is ρ_i^s, the measured Pseudorange P_i^s, inclusive of all error sources, can be modeled as

$$P_i^s = \rho_i^s + ct_i - ct^s + A + M_i + \epsilon_i^s \tag{1}$$

Here, t_i and t^s are receiver and satellite clock biases, respectively, with respect to true time. A represents the range error due to refractions in the atmosphere. M_i^s denotes Multipath, ϵ_i^s is receiver's hardware noise and c is the speed of light. In today's systems, the satellite clock error t^s is small[1], and ct_i proves to be the major source of error. Hence, ct_i is modeled as an unknown, and ρ_i^s is written as a function of the *unknown* 3D receiver location ρ_i and the *known* 3D satellite position ρ^s. This results in a total of 4 unknowns. Once we have a lock with 4 satellites, the time bias and the 3D locations can be jointly estimated resulting in a position fix. The ignored error sources, i.e., t^s, A, M, and ϵ, could contribute to an error of 1-4 meters, depending upon the environmental conditions.

GPS receivers on phones and car dashboards use the above techniques. However, higher accuracy applications such as 3D orientation tracking require better performance. To this end, modern GPS research has leveraged the *phase* of the arriving signals, as detailed next.

Carrier Phase

Once a satellite lock is acquired, the phase of the arriving signal, ϕ_i^s, is constantly tracked by a *phase lock loop* (PLL). The true range between the satellite and the receiver, ρ_i^s, can be expressed as a multiplicative factor of wavelength λ.

$$\rho_i^s = \lambda N_i^s + \lambda \phi_i^s \tag{2}$$

N_i^s is an unknown integer, meaning that the PLL measurement of ϕ_i^s only captures the fractional part of the range. However, due to atmospheric effects, multipath, and clock issues, the above equation can be updated:

$$\lambda \phi_i^s = \rho_i^s + ct_i - ct^s + A + M_i + \epsilon_i - \lambda N_i^s \tag{3}$$

Estimating N_i^s is non-trivial and several algorithms have been proposed [26,27]. We discuss more in Sec. 5.1.

One advantage of carrier phase is that *its changes over time* can be tracked reliably by utilizing the doppler shift in the signal [28]. Hence, $\phi_i^s(t_2)$ takes the same mathematical form as Equation 3. Thus, if the initial value of N_i^s can somehow be estimated, the tracking thereafter can be good. We now explain how today's systems like Differential GPS (DGPS) with this integer ambiguity, N_i^s, and other error sources.

2.3 Computing Differentials

Environmental error sources in Equation 3 are correlated over short time periods and within small geographical areas (200 km). Thus, two GPS measurements across time can be subtracted (or differenced) to eliminate some of these factors. Similarly, simultaneous measurements from multiple GPS receivers can also be differenced. Differential GPS (DGPS) [29] performs such operations on *pseudoranges*, while Real-Time Kinematic (RTK) [30] applies differentials to *carrier phase*. For our purpose of precise orientation tracking, the latter is more relevant. To this end, we outline 4 kinds of carrier phase differentials.

(1) Single Differentials across Receivers (SD_{ij})

Consider the carrier phase equations for two GPS receivers i and j from the same satellite s (let us ignore multipath and noise).

$$\lambda \phi_i^s = \rho_i^s + ct_i - ct^s + A^s - \lambda N_i^s \tag{4}$$

$$\lambda \phi_j^s = \rho_j^s + ct_j - ct^s + A^s - \lambda N_j^s \tag{5}$$

Differencing the above two equations yields the relative position between i and j with fewer error terms. Correlated error sources of atmospheric delays and satellite clock biases disappear.

$$\lambda \Delta \phi_{ij}^s = \Delta \rho_{ij}^s + c\Delta t_{ij} - \lambda \Delta N_{ij}^s \tag{6}$$

Figure 1 illustrates the scenario. Assuming that the satellite is far away, $\Delta \rho_{ij}^s$ can be approximated as $\rho_{ij} Cos\theta$, where ρ_{ij} is the true relative position between i and j (called **baseline** vector). Replacing $\rho_{ij} Cos\theta$ as a vectorial projection of ρ_{ij} on to the line of sight unit-vector $\hat{l_s}$ of satellite s, we have:

$$\lambda \Delta \phi_{ij}^s = \rho_{ij}.\hat{l_s} + c\Delta t_{ij} - \lambda \Delta N_{ij}^s \tag{7}$$

[1] Satellites estimate the errors themselves from mutually exchanged signals as well as from ground sources.

While some errors have disappeared [2], a function of the clock bias errors, $c\Delta t_{ij}$ still remains, motivating the need for double differentials.

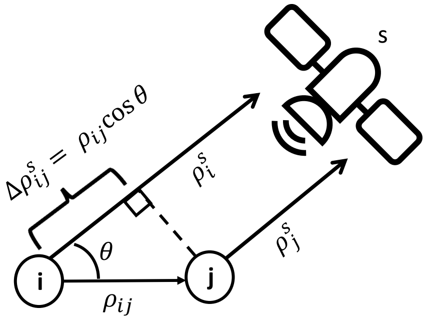

Figure 1: $\rho_i^s - \rho_j^s = \Delta\rho_{ij}^s = \rho_{ij}.\hat{l_s}$

(2) Double Differentials across Receivers and Satellites (DD_{ij}^{sk})

Given multiple satellites in range, the GPS ground receivers i and j can perform the same measurements with satellite k. Equation 7 can then be rewritten as:

$$\lambda\Delta\phi_{ij}^k = \rho_{ij}.\hat{l_k} + c\Delta t_{ij} - \lambda\Delta N_{ij}^k \tag{8}$$

Subtracting the single differential equations 7 and 8, we have a double differential (DD) as follows:

$$\lambda\nabla\Delta\phi_{ij}^{sk} = \rho_{ij}.(\hat{l_s} - \hat{l_k}) - \lambda\nabla\Delta N_{ij}^{sk} \tag{9}$$

The double differential (DD) eliminates the clock biases and the residue is only the integer ambiguity terms. We will discuss the resolution of integer ambiguity in Section 5.1, but assuming that the ambiguities are magically fixed, this provides us a reasonably precise estimate of relative positions (called baselines) between receiver pairs. Ignored factors, including multipath, noise, and antenna phase center errors, add up to a few centimeters of error. Thus, if one of the receiver's absolute position is known in the granularity of millimeters, the absolute position of the other receiver can be estimated precisely as well. Real Time Kinematics (RTK) technology operates exactly as above – *it uses the accurately known location of the reference receiver to calculate the location of the other.* Figure 2 shows the relative distance between two receivers placed roughly 45 cm apart. Differencing techniques convincingly outperform naive subtraction of 3D GPS positions.

(3) Single Differentials across Time ($SD^{t_{12}}$)

Similar to differentials across receivers and satellites, we can also perform differentials across time for the same receiver. This can eliminate the integer ambiguity as follows:

$$\lambda\phi_i^s(t_1) = \rho_i^s(t_1) + ct_i(t_1) - ct^s(t_1) + A^s - \lambda N_i^s(t_1) \tag{10}$$
$$\lambda\phi_i^s(t_2) = \rho_i^s(t_2) + ct_i(t_2) - ct^s(t_2) + A^s - \lambda N_i^s(t_2) \tag{11}$$

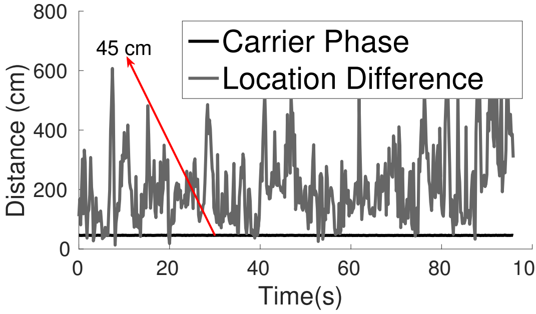

Figure 2: Relative positioning using carrier phase and double differentials is an order of magnitude more accurate than naïve location differencing. Here, a 45 cm baseline is correct within a few cm.

When no cycle slips occur (i.e., $N_i^s(t_1) = N_i^s(t_2)$) and given that the satellite clock bias is well known and hardly changes in small time intervals $[t_1, t_2]$, we have

$$\lambda\Delta\phi_i^s(t_{12}) = \Delta\rho_i^s(t_{12}) + ct_i(t_{12}) \tag{12}$$

Since the distance to the satellite is very large compared to the displacement of the receiver during $[t_1, t_2]$, the satellite location can be assumed fixed for this interval, incurring negligible errors with this approximation. The motion of the receiver is then represented in Figure 3. As described earlier, $\Delta\rho_i^s(t_{12})$ can now be expressed as the projection of the relative motion vector $\rho_i(t_{12})$ on to the line-of-sight unit vector $\hat{l_s}$ of the satellite. Thus, Equation 12 becomes:

$$\lambda\Delta\phi_i^s(t_{12}) = \rho_i(t_{12}).\hat{l_s} + ct_i(t_{12}) \tag{13}$$

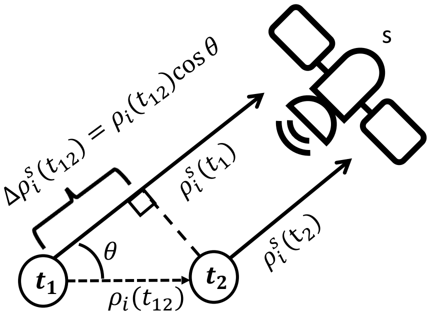

Figure 3: $\rho_i^s(t_1) - \rho_j^s(t_2) = \Delta\rho_{ij}^s(t_{12}) = \rho_{ij}^s(t_{12}).\hat{l_s}$

Observe that $\rho_i^s(t_{12})$ (the relative displacement of the receiver during $[t_1, t_2]$) and the receiver's clock bias $t_i(t_{12})$ add up to 4 unknowns. Measurements across 4 satellites can help jointly estimate the *relative displacement* and clock bias. This estimated relative displacement is quite accurate since the integer ambiguity N_i^s does not pollute it. We verified this with a static receiver placed on the ground; the estimation resulted in about 1 cm/s of motion, far more accurate than velocity estimations from Doppler shifts.

[2] We are aware that differentials can amplify noise and multipath, however, since carrier phase noise is in the granularity of few mm [31], we ignore noise in the rest of the paper. We also assume multipath is not excessive, such as the drone flying low in Manhattan-like areas.

(4) Double Differentials across Receivers and Time ($DD^{t_{12}}_{ij}$)

Our final double differential combines receivers and time. We compute the single differential between receivers ij from Equation 7 but write them for consecutive time points t_1 and t_2.

$$\lambda\Delta\phi^s_{ij}(t_1) = \rho_{ij}(t_1).\hat{l_s} + c\Delta t_{ij}(t_1) + \lambda\Delta N^s_{ij}(t_1) \quad (14)$$

$$\lambda\Delta\phi^s_{ij}(t_2) = \rho_{ij}(t_2).\hat{l_s} + c\Delta t_{ij}(t_2) + \lambda\Delta N^s_{ij}(t_2) \quad (15)$$

Assuming no cycle slips (we will relax this assumption later), $\Delta N^s_{ij}(t_2) = \Delta N^s_{ij}(t_1)$, hence subtracting the above two equations eliminates integer ambiguity:

$$\lambda\nabla\Delta\phi^s_{ij}(t_{12}) = (\rho_{ij}(t_1) - \rho_{ij}(t_2)).\hat{l_s} + c.\Delta t_{ij}(t_{12}) \quad (16)$$

One may physically interpret this equation as the subtraction of two vectors, where the first vector is a drone baseline at time t_1 and the second vector is the same baseline at t_2. In other words, this captures the relative motion of the drone across time.

2.4 The Bigger Picture

We take-away 2 key points from the discussion on the differentials:

- The double differentials across receivers and satellites (DD^{sk}_{ij}) yields the drone's baseline vectors at any given time point (Equation 9). However, this estimate is still polluted by integer ambiguity.
- The double differential across receivers and time ($DD^{t_{12}}_{ij}$) yields the drone's *relative change* in the baseline vectors during flight (Equation 16). Importantly, this relative estimate is free of the integer ambiguities.

Thus, we now have *two* separate estimates of the drone's 3D baseline vectors, each with different error properties. SafetyNet recognizes the opportunity of combining these two noisy estimates to precisely track the drone baselines, ultimately tracking orientation.

3. SYSTEM MODEL

SafetyNet will model orientation tracking as a state estimation problem, where the state is defined as 3D orientation. We formally define "3D orientation" first and then design the model.

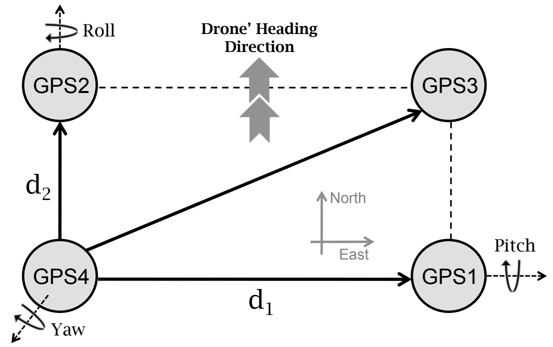

Figure 4: The drone baseline vectors ρ_{41} and ρ_{42} aligned with Earth's reference frame.

Figure 4 pretends 4 GPS receivers have been placed on a drone – their locations denoted as ρ_1, ρ_2, ρ_3 and ρ_4. The baseline vectors joining one of the receivers (say ρ_4) to the others can be defined as $\rho_{ij} = \rho_i - \rho_j$. When the "baselines" are aligned with the North-East reference axes, the baseline matrix B_o can be written as:

$$B_o = [\rho_{41}\ \rho_{42}\ \rho_{43}] \quad (17)$$

Assuming that the magnitude of the baseline vectors are d_1 and d_2, we can expand B_o as:

$$B_o = \begin{bmatrix} d_1 & 0 & d_1 \\ 0 & d_2 & d_2 \\ 0 & 0 & 0 \end{bmatrix} \quad (18)$$

Figure 4 also shows the rotation conventions of the 3 Euler angles – *pitch, roll and yaw*. Applying these rotations on B_o will obviously yield the new baseline matrix, B. For all our results, we will express rotations in terms of the Euler angles (i.e., degrees), which are intuitive to understand. However, for the purpose of mathematical efficiency, we will use *quaternion mathematics*, an alternative representation to Euler angles. Briefly, the baselines at an arbitrary orientation, called quaternion q, can be expressed in terms of the initial orientation q_o (aligned with reference axes), as below:

$$\rho_{ij}(q) = A(q)'\rho_{ij}(q_o) \quad (19)$$

Here $A(q)$ is the *rotation matrix* associated with the orientation quaternion q. Similarly, extending the effect of rotations to the entire baseline matrix, we have:

$$B(q) = A(q)'B_o \quad (20)$$

Effectively, orientation is about estimating the rotation matrix $A(q)$ using projected measurements of $B(q)$ on various satellite directions. Details on conversion between quaternions, Euler angles and rotation matrices can be found in [32]. Regardless of these mathematical conversions, *the core conceptual question still pertains to estimating the matrix B at any given time.*

4. SYSTEM DESIGN: PHASE 1

We adopt a Bayesian filtering approach for tracking drone orientation. Figure 5 shows the model: (1) A state transition function, derived from the incremental changes in orientation ($DD^{t_{12}}_{ij}$), models the next state of the drone. Recall that these changes are affected by the hardware noise and multipath errors. (2) A measurement function, (DD^{sk}_{ij}), reflects the absolute orientation of the drone at any given time. Of course, this measurement is polluted by integer ambiguity. We adopt a Kalman Filter to combine the uncertainties from the transition and measurement functions, and track the most likely state of the system through time. We describe this basic design first. Then we focus on resolving the error sources (such as integer ambiguity, cycle slips, and missing data), and redesign the framework to accommodate these optimizations. Our final design is an "adjusted" particle filter algorithm that tracks drone orientation with consistent accuracy.

4.1 State Transition Model

The relative baseline changes over time are directly obtained

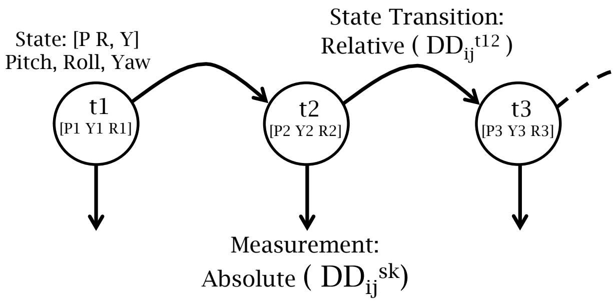

Figure 5: Bayesian filtering approach to tracking orientation state over time.

from Equation 16, copied for convenience:

$$\lambda\nabla\Delta\phi_{ij}^{s}(t_{12}) = (\rho_{ij}(t_1) - \rho_{ij}(t_2)).\hat{l_s} + c.\Delta t_{ij}(t_{12}) \tag{21}$$

Omitting details, we rewrite with quaternions:

$$\lambda\nabla\Delta\phi_{ij}^{s}(t_{12}) = \rho_{ij}(q_o)\lfloor A(q_1).\hat{l_s}{}_{\times}\rfloor\delta\theta + c.\Delta t_{ij}(t_{12}) \tag{22}$$

where, q_1 is the orientation quaternion at time t_1, $\delta\theta$ is the rotation vector [33] associated with quaternion δq. $\lfloor\ _\times\rfloor$ is the vector cross operator [34]. We now solve Equation 22 for various satellites s and GPS receiver-pairs ij using least-squares estimation. The result yields an estimate of the rotation vector $\delta\theta$ (hence δq) between two time points t_1 and t_2. We can thus estimate the new orientation quaternion q_2 as:

$$q_2 = \delta q \otimes q_1 \tag{23}$$

Here, $\otimes$ is the quaternion multiplication operator. When translated back to Euler angles, the result is the relative orientation change – pitch, yaw, and roll – from one state to the next. As mentioned earlier, this estimate is polluted by hardware noise and multipath.

4.2 Absolute Orientation Measurement

Equation 9 showed that double differentials across receivers and satellites (DD_{ij}^{sk}) are estimates of the *absolute* baseline vectors of the drone. While they cancel out clock bias errors, they leave the integer ambiguities as follows:

$$\lambda\nabla\Delta\phi_{ij}^{sk} = \rho_{ij}.(\hat{l_s} - \hat{l_k}) + \lambda\nabla\Delta N_{ij}^{sk} \tag{24}$$

Translating to quaternions again, using very similar conventions as described earlier, we have:

$$\begin{gathered}\lambda\nabla\Delta\phi_{ij}^{sk} - \rho_{ij}(q_o).A(q_n).(\hat{l_s} - \hat{l_k}) = \\ \rho_{i,j}(q_o).\lfloor A(q_n).(\hat{l_s} - \hat{l_k})_{\times}\rfloor\delta\theta + \lambda.\nabla\Delta N_{ij}^{sk}\end{gathered} \tag{25}$$

We develop techniques for resolving integer ambiguities in Section 5.1. For now, let's assume they are resolved – then, we are left with a set of linear equations over different satellite pairs sk and baselines ij. By solving them using standard Least Squares Estimation (LSE), the rotation vector $\delta\theta$ and associated quaternion δq can be obtained. Hence the orientation q is.

$$q = \delta q \otimes q_n \tag{26}$$

Here q_n is an initial orientation estimate for the purposes of linearization of Equation 25 (usually comes from the transition model). Note that this estimate q is absolute in the earth's reference frame, since the satellite locations sk are both known in that reference frame.

4.3 Applying Extended Kalman Filter (EKF)

Kalman filter is an estimation algorithm to systematically combine erroneous observations about a system state. In our case, since the transition function is intrinsically non-linear (because of the conversion from rotation matrices to quaternions), we make linearized approximations using Extended Kalman Filter (EKF) [35]. We briefly sketch the methodology here.

Transition: The state transition function in Equation 22 can be re-written in a generic non linear model as:

$$\hat{q_{k+1}} = f(q_k) + w_k \tag{27}$$

Here, k and $k+1$ are two successive time points, f is of the (non-linear) form:

$$f = \delta q \otimes q_k \tag{28}$$

and w_k is the estimated process noise derived from the GPS signal SNR. δq is the quaternion of $\delta\theta$ in Equation 22. Now, linearizing Equation 27 around the estimated rotation vector associated with quaternion q_k, we have:

$$\delta\theta_{k+1} = F.\delta\theta_k + w_k \tag{29}$$

where F is given by $\frac{\partial f}{\partial\theta}|_{\theta_{\hat{k+1}}}$

Measurement: As with the transition function above, the measurement function can be Equation 25, written in the following form:

$$y - y_o = H.\delta\theta_{k+1} + v_t \tag{30}$$

where k used for both satellite and sample index. v_t is the measurement noise derived from GPS SNR.

$$y = \lambda\Delta\phi_{ij}^{sk} - \lambda.\Delta N_{ij}^{sk} \tag{31}$$

$$y_o = \rho_{ij}(q_o).A(\hat{q_{k+1}}).(\hat{l_s} - \hat{l_k}) \tag{32}$$

$$H = \rho_{ij}(q_o).\lfloor A(\hat{q_{k+1}}).(\hat{l_s} - \hat{l_k})_{\times}\rfloor \tag{33}$$

Equations. 29 and 30 can now be combined using an Extended Kalman Filter for a refined estimate of $\delta\theta_{k+1}$. Its associated quaternion δq_{k+1} can then be used to estimate the most likely orientation quaternion q_{k+1}:

$$q_{k+1} = \delta q_{k+1} \otimes \hat{q_{k+1}} \tag{34}$$

While the noise terms of Transition and Measurement models are not completely independent, we introduce Particle Filters later (Section 5.3) to lift the Gaussian assumption of Kalman Filters.

5. RESOLVING AMBIGUITIES: PHASE 2

The above EKF has been designed under the convenient assumption that integer ambiguity and cycle slips have been resolved. However, this resolution is challenging, as evident from the numerous papers written on this topic [36–39]. Since these error sources seriously affect drone orientation, we comprehensively address them next.

5.1 Resolving Integer Ambiguity

Recall from Equation 2 that the Phase Lock Loop (PLL) only measures the fractional part of the range, leaving an unknown of λN. Once the drone is flying, additional multiples of wavelengths can also accumulate, called *cycle slips*. We first describe ways to mitigate the initial λN and discuss cycle slips thereafter.

Note that we do not need to estimate N since we are not computing the actual range (or location) of the GPS receiver. Since we only care about orientation, we have been operating in the space of differentials. Thus, when we compute $N_{ij}^{s} = N_i^s - N_j^s$, followed by $N_{ij}^{sk} = N_{ij}^{s} - N_{ij}^{k}$, we recognize that N_{ij}^{sk} are also integers. Our goal is to estimate these N_{ij}^{sk} integers, for all combinations of ij and sk.

Our core intuition is to compute a Cos function on the value of $2\pi N_{ij}^{sk}$; since N_{ij}^{sk} should be an integer, $Cos(2\pi N_{ij}^{sk})$ should equal 1. Now, let's say this Cos function is summed up over all the possible values of N_{ij}^{sk}, then we have $\sum_{ij,sk} Cos(2\pi N_{ij}^{sk})$. If there are ψ possible tuples of $< ij, sk >$, then the summation should add up to ψ. Now, this summation can be rewritten as a function of orientation q as follows:

$$cos(2\pi N_{ij}^{sk}) = cos(2\pi \frac{\lambda \phi_{ij}^{sk} - \rho_{ij}(q_o).A(q).(\hat{l_s} - \hat{l_k})}{\lambda}) \qquad (35)$$

Here N_{ij}^{sk} is derived from a combination of Equations 9 and 19. Then,

$$M(q) = \sum_{ij,sk} cos(2\pi N_{ij}^{sk}) \qquad (36)$$

In an ideal case, the correct q should make the RHS of Equation 36 equal to ψ. However, given that phase ϕ_{ij}^{sk} in Equation 35 has errors, for a given q, the corresponding values of N_{ij}^{sk} are not all 1. When these N_{ij}^{sk} are plugged into Equation 36, the value is less than ψ. Hence, we search across all values of q at the granularity of 5 degrees and pick a q_{coarse}, such that

$$M(q_{coarse}) = \max_{q} M(q) \qquad (37)$$

5 degrees is not an arbitrary design choice. Rather, 5 degree is a function of GPS signal wavelength and the diagonal of the drone, together ensuring that the estimated values of N_{ij}^{sk} do not over or undershoot an integer by more than 0.2 with high probability. So long the offset is less than 0.5, the actual N_{ij}^{sk} values can be estimated by rounding off. The equation below shows our final estimation of integer ambiguity from q_{coarse}.

$$\hat{N_{ij}^{sk}} = \lfloor \frac{\lambda \phi_{ij}^{sk} - \rho_{ij}(q_o).A(q_{coarse}).(\hat{l_s} - \hat{l_k})}{\lambda} + 0.5 \rfloor \qquad (38)$$

Our results show that when the drone is static and ready to take off, the estimates of N_{ij}^{sk} are correct (also, the Cos function is close to 1). However, once the drone starts flying, and especially after sharp maneuvers, integer ambiguity again builds up. This is because poor SNR, exacerbated by sudden phase changes in the BPSK GPS signal, causes the PLL to lose lock. The result is both full and half cycle slips[3]. Now, even if q_{coarse} correctly estimates a large fraction of N_{ij}^{sk}, the small erroneous fraction influences subsequent estimation and the orientation diverges over time. Moreover, half cycles also increase the search space by $8x$ since the search is 3-dimensional (yaw, pitch, roll). Thus, coarse grained attitude, q_{coarse}, can no longer be used for consistent tracking.

Fig.6 compares the *cycle slip percentage* for static and flying drones – we use the view from the drone camera as the ground truth [40] (described later in Section 6). Evidently, cycle slips occur much more frequently (almost once every second) when the drone is flying. This means that even our assumption in Equation 12 – that $N_i^s(t_2) = N_i^s(t_1)$ – is not necessarily true, and can drastically pollute estimates. To this end, SafetyNet approximates q_{coarse} for Equation 38 from the estimate of the State Transition Model in Section 4.1. While this is not perfect, the Outlier detection and Particle Filter techniques discussed next will help absorb most errors.

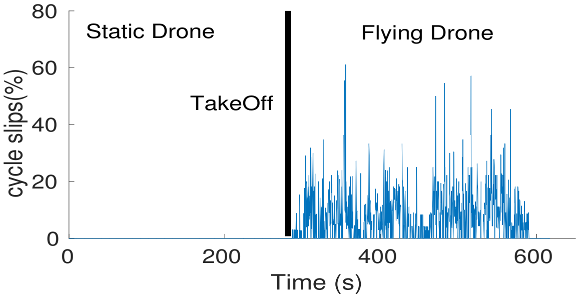

Figure 6: Cycle slips occur more often during flight.

5.2 Cycle Slip Detection

Needless to say, our cycle slip detection scheme has to be quick (to be able to run every few seconds). Towards this goal, we form temporal differential measurements from multiple satellites as shown in Equation 13. After compensating for known satellite biases $t^s(t_{12})$, we are left with four unknowns – 3D change in position $\rho_i(t_{12})$ and clock bias difference $t_i(t_{12})$. Solving these using *Robust Least Squares (RLS)* [41] will typically isolate the outliers, with Error Residuals for the inliers being less than $2cm$. Our transition model (Section 4.1) incorporates corrections from this module. While there is a possibility that RLS may not converge, we use GLONASS satellites to boost the redundancy.

Unlike GPS satellites, GLONASS uses FDMA, i.e., satellites transmit in different frequencies. Since the wavelengths are now λ^s and λ^k from satellites s and k, the integer ambiguity terms do not group themselves into a single integer in the DD_{ij}^{sk} double differentials. Hence, we use GLONASS only for $DD_{ij}^{t_{12}}$, since the signals from the same satellite cancel across time, t_{12}. However, when even GLONASS+GPS fails (perhaps because of too few satellites under poor weather), we resort to Particle Filters (Section 5.3).

5.3 Unified Particle Filter Framework

[3] Half cycle slips occur because some PLLs square the signal to remove BPSK messages; squaring makes the correlator match the original signal twice within a signal period, indicating phase change. However, this squaring effectively halves the wavelength, meaning that N_{ij}^{sk} is now multiples of 0.5.

The net result of all the above techniques can be summarized as follows: *the orientation q, and hence the values of N_{ij}^{sk}, are mostly correct, and the Kalman filter can track it well. However, occasionally q is incorrect, and this error accumulates over time causing complete divergence.* Figure 7 illustrates an example of the drift with Kalman Filters (KF). We bring Particle Filters to better handle these cases since multiple particles (i.e., orientation states) can be propagated through time, while relaxing linearity and Gaussian assumptions required for KF. Thus, SafetyNet adopts a hybrid approach, i.e., the system uses Kalman Filters in general but invokes a particle filter when:

- Orientation estimates have poor confidence based on the summed Cos metric in Equation 36.
- Missing data in one or more of the receivers derails the state transition function.

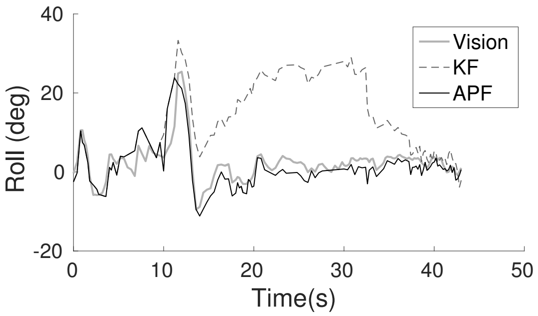

Figure 7: KF based orientation drifts away due to poor SNR or missing data.

Figure 8 summarizes the flow of events. When the confidence on q (based on the Cos metric) is high, SafetyNet continues to run the Kalman filter. When low, we select K values of orientation q – the values whose corresponding Cos metric ranks in the top-K. We initialize K particles at each of these orientation states and update each particle based on the transition and measurements functions. These particles are essentially "trackers" of different integer ambiguities, and we propagate them until the Cos metric for one becomes high. Now, if the particles are too close to each other – within 3° – we merge these particles into one. If a particle exhibits extremely low weight, we also remove that particle. If merging or removal leaves one particle, SafetyNet switches back to Kalman filtering again. Else, the confidence function is computed again. If the confidence function is low, more particles are added and resampled – resampling is performed such that the resulting particles are an equal mix of the (highest weighted) current and new particles. However, if the confidence is high, adding and resampling is not necessary. The process repeats until the system converges to a single particle.

5.4 Adjusted Particle Filter (APF) Design

We propose an improvement to the particle filter design. The opportunity arises from the observation that in the conventional particle filter, a particle's weight is typically proportional to its likelihood function (i.e., proximity to the true state). Thus, when the system resamples, the concentration of particles are proportionally greater at the higher weight particles. Over time, the hope is that one of the particles would converge to the correct state. However, we ask: *why not move the particle to a state that maximizes the likelihood.* In fact, given that the error variances of the transition and measurement functions are known, the best estimate can be computed as a combination of the two. We could move each particle to the "best" state in its neighborhood, and then perform the resampling step.

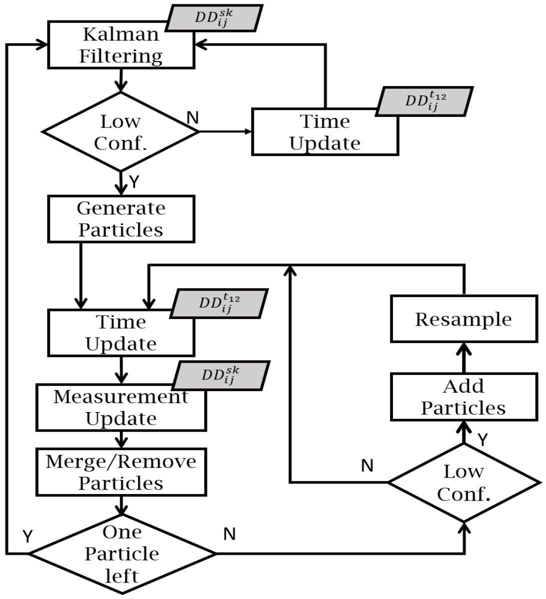

Figure 8: Flow chart of unified particle filter

Figure 9 illustrates the idea for a case of three particles. The middle particle is propagated to time $time_1$ using the transition model $DD_{ij}^{t_{12}}$. Similarly, the measurement function DD_{ij}^{sk} produces an estimate of the new state. The transitioned and measured states along with their error distributions are depicted. The two estimates are combined using a Kalman filter like approach, and the particle "adjusted" to this best state. This particle is then propagated further to time $time_2$, which now has a smaller error variance.

We note that each particle essentially tracks a particular N_{ij}^{sk} integer ambiguity vectors. As a result, every carrier-phase measurement yields distinct orientation q for each of these vectors. The benefits arise because: (1) For the correct integer ambiguity vector, the error properties of the measurement function becomes Gaussian. Hence, the combining process indeed is like a Kalman filter, leading to faster convergence. (2) For incorrect integer ambiguity vectors, the gaussian error properties will not hold. Moreover, the transitioned particle will be adjusted towards a wrong state because of incorrect integer ambiguity resolution. Thus, the net outcome is faster convergence for the correct ambiguities, while disappearing particles for incorrect ambiguities. Figure 7 shows the efficacy of the *Adjusted Particle Filter* (APF) even when KF diverges. The next section presents more detailed results.

6. EVALUATION

Our evaluation will comprehensively compare SafetyNet's ori-

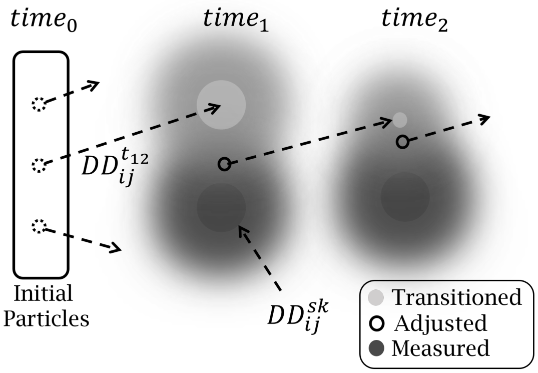

Figure 9: Adjusted Particle Filter (APF) results in faster reliable convergence even with few particles.

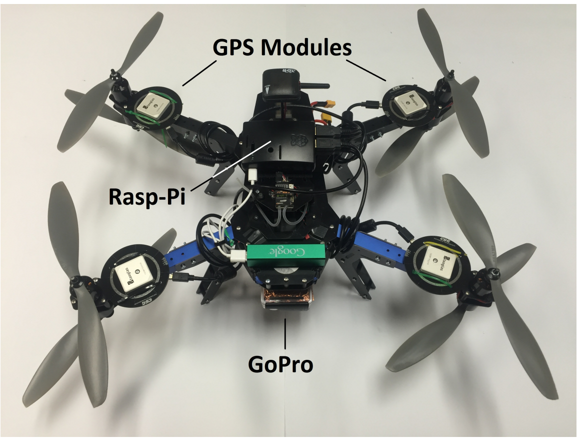

Figure 10: Four off-the-shelf GPS modules, a Raspberry Pi, and a GoPro on an X-8 Octocopter

entation accuracy against two reference points: (1) the real IMU of a professional-grade drone (using algorithms based on [42]) and (2) high-precision estimates of ground truth through computer vision, which is orthogonal to both inertial and GPS-based sensory modalities. All results are from 11 flight sessions of ≈ 6 minutes each, across a diverse set of weather and satellite conditions; we will also report 95^{th} percentile performance to capture robustness. We aim to answer the following critical questions.

- What is SafetyNet's accuracy in estimating a drone's pitch, roll, and yaw during flight? *(Figures 11, 12a)*
- Is SafetyNet's accuracy comparable to IMU, relative to ground truth from computer vision? *(Figure 12(b,c))*
- Is SafetyNet robust to weather impairing satellite visibility? *(Figures 13, 14)*
- Is SafetyNet robust under aggressive maneuvers? *(Figures 11, 15)*
- Is SafetyNet computationally practical for embedded, real-time applications? *(Figure 16)*
- Which modules of SafetyNet provide the largest performance gains? *(Figure 17)*

Our results show that SafetyNet demonstrates consistently dependable performance in each measure, and that all of SafetyNet's modules are critical to the net outcome. We begin by describing the platform and methodology.

6.1 Experimental Platform and Methodology

We conduct all experimentation using a 3DR X-8 octocopter (8-rotor), pictured in Figure 10. "X" implies 4 arms with unequal spacing – yielding greater stability of roll angle versus pitch (recall that pitch is the dominant motion when an airplane takes off or lands, roll is dominant when the plane makes a left/right turn). Two rotors attach to each arm: one above, one below. To the X-8, we mounted 4 u-Blox NEO-M8T multi-GNSS receivers, adjacent to the top motor of each arm. We also mounted a Raspberry Pi 2 near the drone's center of mass – the GPS receivers transfer the data to the Pi via USB at 5 Hz. IMU data at 10 Hz is recovered as binary logs from the X-8's USB interface. [4]

A GoPro camera, affixed to the underbelly of the airframe, points vertically downwards. The GoPro captures video at 30 fps with a fast shutter (to minimize motion blur and rolling shutter effects). We use `ffmpeg` to sample the video into stills at 7.5 fps (every fourth frame). We post-process the images using *Pix4D*: advanced commercial photogrammetry software that uses *structure from motion* (SfM) to perform 3D registration of each frame. From Pix4D's outputs, we can recover a high-precision estimate of drone attitude (accuracy $\approx 0.05^\circ$ [40, 43]). post-process the images using *Pix4D*: advanced commercial photogrammetry software that uses *structure from motion* (SfM) to perform 3D registration of each frame. Note: structure from motion would not be practical as a realtime IMU replacement – each 5-7 minute flight requires several hours of processing on a server of 16 CPU cores, 32 GB RAM, and CUDA on a high-end NVIDIA GRID K2 GPU.

6.2 Data Alignment for Comparison

SafetyNet and IMU data have precise GPS timestamps but the GoPro does not – this makes precise comparison difficult. Therefore, at the beginning/end of each flight, we point the GoPro towards an Android phone displaying the current GPS time. Once the video frame is synchronized with the GPS time, we extrapolate and time stamp every frame – this is possible because we extensively verified that the inter-frame spacing of the GoPro is uniform (33.33 ms at 30 fps). Still, residual error remains from unpredictable OS delay in the Android application displaying the current time. We align the yaw *angle sequence* between SafetyNet and vision. We also perform this synchronization procedure with IMU to eliminate residual misalignment. Finally, we align the GoPro and GPS X-Y axes by finding the appropriate rotation matrix that eliminates offsets in yaw, pitch, and roll. We now present

[4] Experiments were conducted as per FAA regulations. Operator trained by a general aviation pilot, over relatively vacant fields, with flight height not exceeding 45m.

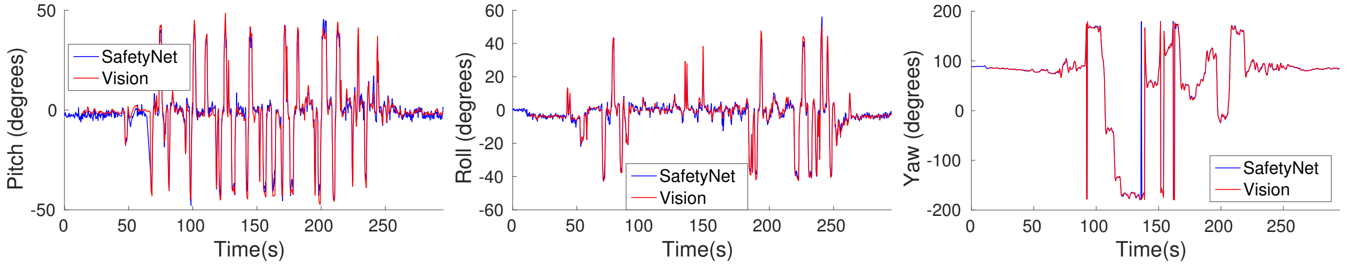

Figure 11: Time trace of SafetyNet orientation compared with Vision (a) Pitch (b) Roll and (c) Yaw. SafetyNet can track highly aggressive maneuvers well

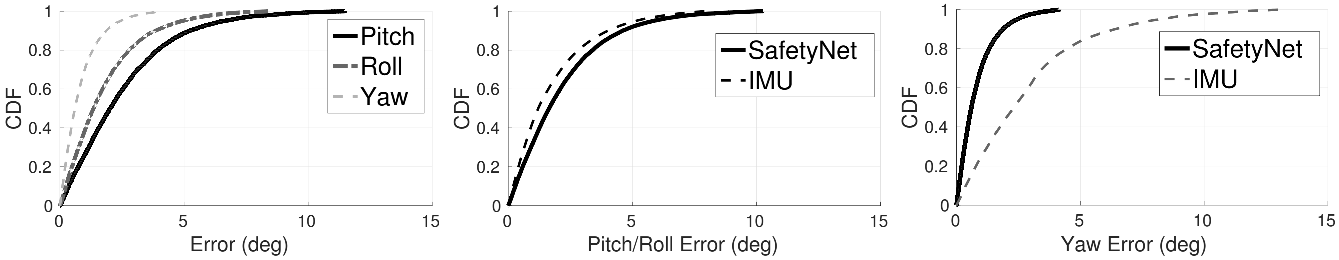

Figure 12: (a) CDF of errors for an hour of aggressive flight data shows the effectiveness of SafetyNet. (b) Comparison of SafetyNet's Pitch/Roll against IMU's. (c) Comparison of Yaw against IMU. SafetyNet outperforms IMU in Yaw by 4x while Pitch/Roll is 30% worse.

results.

6.3 Performance Results

(1) Overall Tracking Accuracy: Figure 11 depicts SafetyNet closely tracking (a) pitch, (b) roll, and (c) yaw against Pix4D/vision. Figure 12(a) aggregates 60 minutes of flying data: median error of 2.07° for pitch, 1.38° for roll, and 0.61° for yaw. Higher accuracy for roll versus pitch is attributable to our X-shaped drone: the x-axis baseline is 46% longer than the y-axis. Accordingly, roll is 49% more accurate. As expected, yaw accuracy is higher than pitch or roll as it is relatively less subject to GPS dilution of precision from satellite geometry.

(2) Comparison with IMU: Figure 12(b and c) compares the accuracy of SafetyNet with IMU, treating Pix4D/vision results as ground truth. SafetyNet is relatively less accurate ($\approx 30\%$ worse) than IMU in pitch and roll. However, SafetyNet comprehensively outperforms IMU in yaw angle accuracy ($\approx 300\%$ more accurate / one-quarter error). Inferior yaw accuracy of IMU, relative to pitch and roll, is expected due to reliance on magnetometer versus the accelerometer gravity vector. Again, SafetyNet's superior yaw performance is expected. We are encouraged that SafetyNet's pitch/roll accuracy is acceptable as a plausible IMU substitute.

(3) Robustness to Weather: Figure 13 plots median accuracy over flights/flying conditions. Error bars denote 5th and 95th percentiles. Flights 7-11 were conducted in fog. Despite fewer detectable satellites, SafetyNet performance remains consistent and robust. To emulate extreme weather scenarios, Figure 14 shows accuracy with varied fractions of satellites artificially removed from processing. The medians and 75th percentiles are mostly robust to satellite skipping. As evident from the 95th percentile line, worst case errors start increasing at 40%. Having fully exploited the u-Blox NEO-M8T multi-GNSS capabilities, SafetyNet maintains high satellite count, increasing reliability.

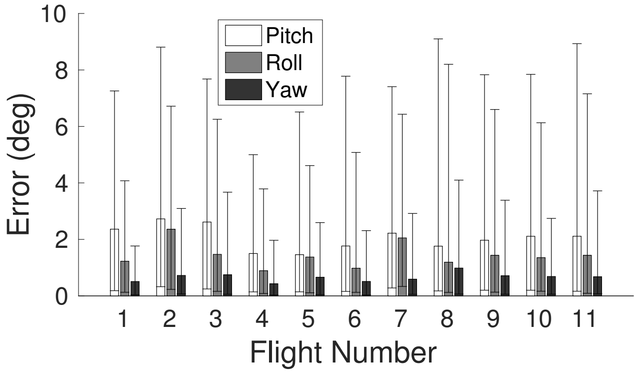

Figure 13: Accuracy in SafetyNet is consistent over different flights under diverse satellite visibility

(4) Robustness to Maneuvers: Figures 11 and 15 show that SafetyNet tracks aggressive maneuvers. Median error along with 5th and 95th percentiles is plotted as a function of the pitch/roll/yaw rates. Tracking of extreme motions (120+° per second) becomes challenging because of satellite blockage, cycle slips, and missing samples. Yet, SafetyNet's Particle Filter Framework is robust enough to even the most challenging motions.

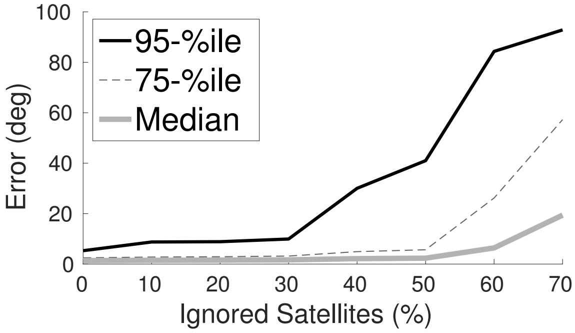

Figure 14: Number of outliers can increase with fewer satellite availability, but SafetyNet maximally exploits multi GNSS satellite availability

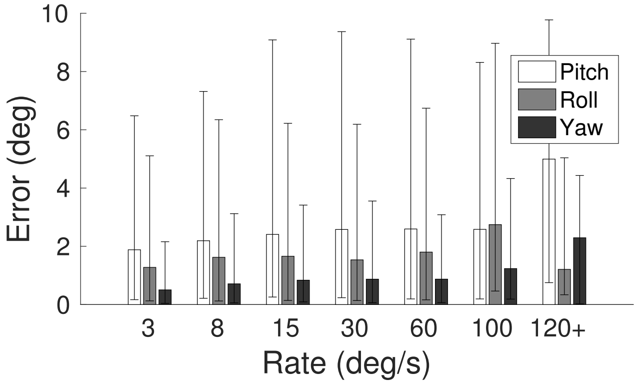

Figure 15: SafetyNet's accuracy degrades gracefully with aggressiveness

(6) Implementability: Figure 16 plots the 95%-ile error in pitch, roll and yaw as a function of number of particles. There are two take-aways: (1) SafetyNet's particle filter dramatically reduces error and (2) only few particles are required – the introduction of only a second particle yields a 300% improvement for pitch/roll. We need not trade away the real time implementability of Kalman filters to gain accuracy. SafetyNet can be implemented in realtime on embedded hardware.

(7) Gain by Module: Figure 17 isolates error reduction from each SafetyNet module: Integer Ambiguity resolution (IA), Outlier elimination (OT), Kalman Filter (KF), Simple Particle Filter (PF), Adjusted Particle Filter (APF) and multi-GNSS (GS). IA suffers from cycle slips that OT can eliminate (7x gain in median and 2.5x at 95th). After OT, some wrong integer ambiguities may remain causing KF drift with slow convergence back to a correct state. APF enables rapid convergence (100% median accuracy gains over both PF and KF/ error reduced by 50%). Integrating GLONASS and SBAS [44] improves median performance and dramatically reduces worst case errors (33% median gain, 6x gain at 95th).

7. RELATED WORK

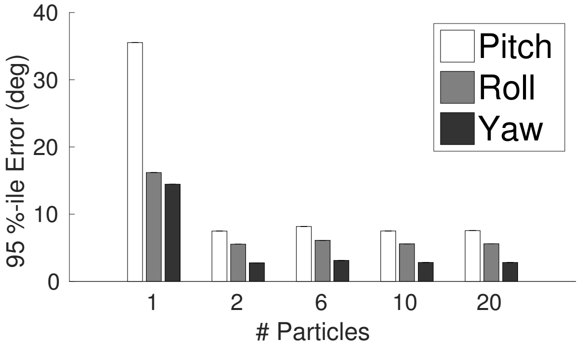

Figure 16: SafetyNet can eliminate outliers with very few particles

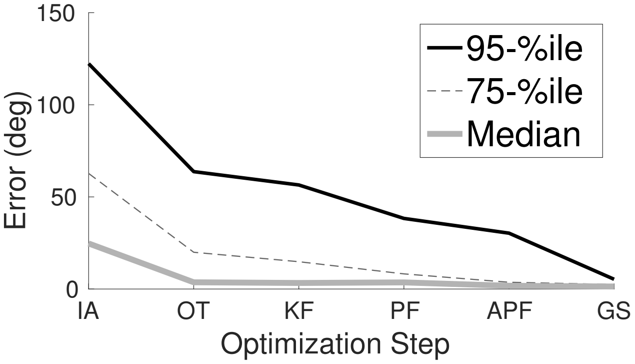

Figure 17: Errors vs Optimization Modules – Integer Ambiguity Resolution (IA), Outlier Elimination (OT), Kalman Filter (KF), Simple Particle Filter (PF), Adjusted Particle Filter (PF) and Multi-GNSS (GS) together contribute to the performance of SafetyNet

Relative GPS: Today's state of the art in GPS relative localization are [13,14], which outperform DGPS [29] and RTK [30]. Among these, *APT* in [14] is closer to SafetyNet and also uses carrier phases combined with double differentials; they achieve sub-meter level accuracy. SafetyNet's differences include: (1) A completely different mathematical model compared to the modeling strategy used for relative localization in *APT*. (2) A notion of "soft decoding" where multiple uncertainties are propagated from the Particle Filter's Cos metric in SafetyNet. SafetyNet also picks new particles from the *angular domain* resulting in dramatic reduction of search space, whereas the search cube based recalibration in *APT* is highly complex. This also facilitates meeting the high accuracy requirement in SafetyNet under a more challenging cycle slip rate. (3) SafetyNet incorporates the inclusion of GLONASS, which is non-trivial due to satellites operating on different frequencies (FDMA). (4) SafetyNet addresses half-cycle slips by resolving integer ambiguities in steps of 0.5 – critical here, given the higher accuracy requirement (1-2 cm) compared to *APT*.

Orientation/Velocity from Multi-GPS: While several works have considered GPS for orientation of spacecraft, full-scale aircraft, or ships [15,16,45], none address the dynamic errors for a small drone. While [17] controls a model helicopter using multi-GPS, the motions are highly constrained (slow changes less than 10°). SafetyNet is unique in addressing dynamic integer ambiguities across an aggressive flight (angular changes up to $150°/s$). Solutions using single receiver, multi-antenna GPS [46] will also benefit from SafetyNet for dynamic error handling. Work in [47] uses a moving antenna for orientation estimates. While interesting, the authors note that the motion has to be slow enough to not break GPS locks, hence it is not possible to adopt such solutions for high dynamic flights. Angular velocity determination using multi-GPS is explored in [48, 49]. However, SafetyNet is unique in integrating angular velocity with double differentials (across time and satellites) into a Particle Filtering framework.

Multi-GNSS Fusion: Several works have attempted to fuse GPS with GLONASS satellite systems for enhanced relative positioning [50–52]. However, each use pseudo-range data only or assume a specialized multi-antenna GPS/GLONASS receiver with a single clock. SafetyNet exploits time-differenced carrier phase measures to leverage GLONASS data in double differentials with off-the-shelf multi-GNSS receivers.

Reliability for IMU: Drift on gyroscopes [53–55] and interferences to magnetic compasses [56–58] is well known in smartphone applications and many of these researchers have proposed application specific inferencing techniques to handle them. While the same error properties can extend to drones as well, the results could be catastrophic and error handling is very critical. Work in [8] proposes inferencing techniques for correcting IMU errors from engine vibrations for drones. Similarly, [9] models magnetic sources of interferences to nullify the effect on IMU. Work in [11] models above similar noisy sources affecting IMU updates into a sophisticated Kalman filter. In contrast to these works SafetyNet offers a completely orthogonal solution relying only on GNSS receivers.

IMU/GPS fusion: Fusion of IMU sensors like accelerometers, gyroscopes, magnetometer and GPS has been extensively used for orientation tracking [59]. Extended Kalman Filters [35, 42, 60, 61] have been used for handling non-linearity. UKF, Sigma Point Filters and Particle filters have also been used for better accuracy with higher complexity [62–65]. While complementary to SafetyNet in reducing orientation uncertainty, these do not address IMU failure.

Orientation from Vision: Pitch/roll angles can be estimated by tracking the horizon [66, 67]. Optical flow [68], stereo vision [69] and feature tracking [70–73] have been proposed as vision-based IMU supplements. Vision techniques are complementary to SafetyNet. Offline, we leverage high-fidelity structure from motion to validate our accuracy.

8. FUTURE WORK

We briefly discuss a few points of future work.

• **Fusion with IMU**: SafetyNet is a completely orthogonal solution, but it is possible to combine GPS with IMU for even better accuracy. We leave this to future work.

• **GPS Sampling Rate**: Sampling rate of our GPS modules was only 5 Hz, but all of our techniques are fundamentally applicable to high rate GPS receivers.

• **Latency**: The processing power on drones continues to evolve – various drones offer diverse computing capabilities. Our future work would need to carefully profile the latency to characterize the kinds of drones that could execute SafetyNet in real-time.

• **Vehicles**: Orientation estimation of self driving cars can also benefit from SafetyNet, adding an extra layer of reliability.

9. CONCLUSION

Grand visions abound for applications of drones. While Amazon, Google, and others have made great strides towards package delivery, etc., drones remain an unacceptable hazard to persons and property. By comprehensively addressing IMU failure through GNSS – a failover completely orthogonal to inertial methods, SafetyNet progresses the state of the art in drone safety.

10. ACKNOWLEDGMENTS

We sincerely thank the reviewers and the anonymous shepherd for their valuable feedback. We are also grateful to NSF (CNS - 1423455) for partially funding the research.

11. REFERENCES

[1] Y. Zhang, A. Chamseddine, C. Rabbath, B. Gordon, C.-Y. Su, S. Rakheja, C. Fulford, J. Apkarian, and P. Gosselin, “Development of advanced fdd and ftc techniques with application to an unmanned quadrotor helicopter testbed,” *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2396–2422, 2013.

[2] “diydrones.” http://diydrones.com/forum/topics/problem-quad-copter-spins-around-itself.

[3] “Google forum.” https://groups.google.com/forum/#!msg/drones-discuss/fnWYM48pGys/kwPI_q_qv0QJ.

[4] “Drone crash survey.” https://docs.google.com/forms/d/1R4MaX8iZWRoyzKJ6rKr0U-p3x-6j0ZVm5a0p83hHty0/viewanalytics.

[5] G. De Pasquale and A. Somà, “Reliability testing procedure for mems imus applied to vibrating environments,” *Sensors*, vol. 10, no. 1, pp. 456–474, 2010.

[6] “3d robotics: Pixhawk.” https://3dr.com/pixhawk-is-ready-for-takeoff/.

[7] “Dji phantom 4.” https://www.dji.com/product/phantom-4/info.

[8] X. Liu and R. Randall, “Blind source separation of internal combustion engine piston slap from other measured vibration signals,” *Mechanical Systems and Signal Processing*, vol. 19, no. 6, pp. 1196–1208, 2005.

[9] M. Wells, *Attenuating magnetic interference in a UAV system*. PhD thesis, Carleton University Ottawa, 2008.

[10] V. Malyavej, P. Torteeka, S. Wongkharn, and T. Wiangtong, “Pose estimation of unmanned ground vehicle based on dead-reckoning/gps sensor fusion by unscented kalman filter,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th*

International Conference on, vol. 1, pp. 395–398, IEEE, 2009.

[11] Y. S. Suh, "Attitude estimation by multiple-mode kalman filters," *IEEE Transactions on industrial electronics*, vol. 53, no. 4, pp. 1386–1389, 2006.

[12] Y.-C. Lai and S.-S. Jan, "Attitude estimation based on fusion of gyroscopes and single antenna gps for small uavs under the influence of vibration," *GPS solutions*, vol. 15, no. 1, pp. 67–77, 2011.

[13] W. Hedgecock *et al.*, "Regtrack: a differential relative gps tracking solution," in *Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services – MOBISYS 2013*, 2013.

[14] W. Hedgecock *et al.*, "Accurate real-time relative localization using single-frequency gps," in *SENSYS 2014*, ACM, 2014.

[15] C. E. Cohen, E. G. Lightsey, B. W. Parkinson, and W. A. Feess, "Space flight tests of attitude determination using gps," *International Journal of Satellite Communications*, 1994.

[16] G. Lachapelle, M. Cannon, G. Lu, and B. Loncarevic, "Shipborne gps attitude determination during mmst-93," *Oceanic Engineering, IEEE Journal of*, 1996.

[17] A. R. Conway, *Autonomous control of an unstable model helicopter using carrier phase GPS only*. PhD thesis, stanford university, 1995.

[18] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, (New York, NY, USA), pp. 85–98, ACM, 2009.

[19] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, (New York, NY, USA), pp. 299–314, ACM, 2010.

[20] "Lipo 4s battery." http://www.droneshop.biz/4s-10000mah-lipo-low-weight-high-efficiency.html.

[21] "Dji matrice 600." http://www.dji.com/product/matrice600.

[22] "Magnetic interference causing crashes." http://diydrones.com/forum/topics/crazy-magnetic-interference.

[23] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[24] R. B. Langley, "Glonass: review and update," *GPS world*, vol. 8, no. 7, pp. 46–51, 1997.

[25] "Galileo." https://en.wikipedia.org/wiki/Galileo_(satellite_navigation).

[26] B. Zhang, P. J. Teunissen, and D. Odijk, "A novel un-differenced ppp-rtk concept," *Journal of Navigation*, vol. 64, no. S1, pp. S180–S191, 2011.

[27] D. Laurichesse, F. Mercier, J.-P. BERTHIAS, P. Broca, and L. Cerri, "Integer ambiguity resolution on undifferenced gps phase measurements and its application to ppp and satellite precise orbit determination," *Navigation*, vol. 56, no. 2, pp. 135–149, 2009.

[28] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[29] B. W. Parkinson and P. K. Enge, "Differential gps," *Global Positioning System: Theory and applications.*, vol. 2, pp. 3–50, 1996.

[30] C. Mekik and M. Arslanoglu, "Investigation on accuracies of real time kinematic gps for gis applications," *Remote Sensing*, vol. 1, no. 1, pp. 22–35, 2009.

[31] G. Blewitt, "Basics of the gps technique: observation equations," *Geodetic applications of GPS*, pp. 10–54, 1997.

[32] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[33] "Rotation vectors." http://farside.ph.utexas.edu/teaching/301/lectures/node100.html.

[34] "Vector cross product." http://soe.rutgers.edu/~meer/GRAD561/ADD/antisymm.pdf.

[35] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982.

[36] C. E. Cohen and B. W. Parkinson, "Integer ambiguity resolution of the gps carrier for spacecraft attitude determination," in *Guidance and Control 1992*, vol. 1, pp. 107–118, 1992.

[37] A. Conway, P. Montgomery, S. Rock, R. Cannon, and B. Parkinson, "A new motion-based algorithm for gps attitude integer resolution," *Navigation*, vol. 43, no. 2, pp. 179–190, 1996.

[38] P. De Jonge and C. Tiberius, "The lambda method for integer ambiguity estimation: implementation aspects," *Publications of the Delft Computing Centre, LGR-Series*, vol. 12, no. 12, pp. 1–47, 1996.

[39] E. Sutton and R. Collins, "Calibration of differential phase map compensation using single axis rotation," in *PROCEEDINGS OF ION GPS*, vol. 11, pp. 1831–1842, INSTITUTE OF NAVIGATION, 1998.

[40] J. Vallet, F. Panissod, C. Strecha, and M. Tracol, "Photogrammetric performance of an ultra light weight swinglet uav," in *UAV-g*, no. EPFL-CONF-169252, 2011.

[41] J. O. Street, R. J. Carroll, and D. Ruppert, "A note on computing robust regression estimates via iteratively reweighted least squares," *The American Statistician*, vol. 42, no. 2, pp. 152–154, 1988.

[42] M. E. Pittelkau, "Rotation vector in attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 6, pp. 855–860, 2003.

[43] S. Harwin and A. Lucieer, "Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (uav) imagery," *Remote Sensing*, vol. 4, no. 6, pp. 1573–1599, 2012.

[44] "Wide area augmentation system." https://en.wikipedia.org/wiki/Wide_Area_Augmentation_System.

[45] Y. Li, K. Zhang, C. Roberts, and M. Murata, "On-the-fly gps-based attitude determination using single-and double-differenced carrier phase measurements," *GPS Solutions*, vol. 8, no. 2, pp. 93–102, 2004.

[46] D. E. Kirkpatrick, "Design of a hardware platform for

gps-based orientation sensing," 2015.

[47] A. G. Evans, "Roll, pitch, and yaw determination using a global positioning system receiver and an antenna periodically moving in a plane," *Marine Geodesy*, vol. 10, no. 1, pp. 43–52, 1986.

[48] M. Ueno, R. Santerre, and A. Kleusberg, "Direct determination of angular velocity using gps," *Journal of Navigation*, vol. 53, no. 02, pp. 371–379, 2000.

[49] P. Montgomery, H. Uematsu, and B. Parkinson, "Analysis of angular velocity determination using gps," in *ION GPS-94*, pp. 697–706, 1994.

[50] J. Wang, M. P. Stewart, and M. Tsakiri, "Modelling glonass measurements for precise positioning," *Survey Review*, vol. 36, no. 280, pp. 110–120, 2001.

[51] S. Han, L. Dai, and C. Rizos, "A new data processing strategy for combined gps/glonass carrier phase-based positioning," in *Proc. ION GPS-99*, pp. 1619–1627, 1999.

[52] T. Tsujii, M. Harigae, T. Inagaki, and T. Kanai, "Flight tests of gps/glonass precise positioning versus dual frequency kgps profile," *Earth, planets and space*, vol. 52, no. 10, pp. 825–829, 2000.

[53] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: indoor pathway mapping made easy," in *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation – NSDI 2013*, pp. 85–98, USENIX Association, 2013.

[54] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking – MOBICOM 2012*, pp. 293–304, ACM, 2012.

[55] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proceedings of the 20th annual international conference on Mobile computing and networking – MOBICOM 2014*, pp. 605–616, ACM, 2014.

[56] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services – MOBISYS 2012*, pp. 197–210, ACM, 2012.

[57] J. G. Manweiler, P. Jain, and R. Roy Choudhury, "Satellites in our pockets: an object positioning system using smartphones," in *Proceedings of the 10th international conference on Mobile systems, applications, and services – MOBISYS 2012*, pp. 211–224, ACM, 2012.

[58] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services – MOBISYS 2014*, MobiSys '14, (New York, NY, USA), pp. 315–328, ACM, 2014.

[59] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[60] D. B. Kingston and R. W. Beard, "Real-time attitude and position estimation for small uavs using low-cost sensors," in *AIAA 3rd unmanned unlimited technical conference, Workshop and exhibit*, pp. 2004–6488, sn, 2004.

[61] W. Y. Liang, W. T. Miao, L. J. Hong, X. C. Lei, and Z. Chen, "Attitude estimation for small helicopter using extended kalman filter," in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, pp. 577–581, IEEE, 2008.

[62] D. Marina, H. Garcia, F. J. Pereda, J. M. Giron-Sierra, and F. Espinosa, "Uav attitude estimation using unscented kalman filter and triad," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 11, pp. 4465–4474, 2012.

[63] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *Journal of guidance, control, and dynamics*, vol. 26, no. 4, pp. 536–542, 2003.

[64] G. G. Rigatos, "Nonlinear kalman filters and particle filters for integrated navigation of unmanned aerial vehicles," *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 978–995, 2012.

[65] R. Van Der Merwe and E. A. Wan, "Sigma-point kalman filters for integrated navigation," in *Proceedings of the 60th Annual Meeting of the Institute of Navigation (ION)*, pp. 641–654, 2004.

[66] S. Thurrowgood, D. Soccol, R. J. Moore, D. Bland, and M. V. Srinivasan, "A vision based system for attitude estimation of uavs," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 5725–5730, IEEE, 2009.

[67] C. Demonceaux, P. Vasseur, and C. Pégard, "Omnidirectional vision on uav for attitude computation," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 2842–2847, IEEE, 2006.

[68] G. L. Barrows, J. S. Chahl, and Y. V. Srinivasan, "Biomimetic visual sensing and flight control," in *Proc. Bristol UAV Conf*, pp. 159–168, 2002.

[69] R. J. Moore, S. Thurrowgood, D. Bland, D. Soccol, and M. V. Srinivasan, "A stereo vision system for uav guidance," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 3386–3391, IEEE, 2009.

[70] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 15–22, IEEE, 2014.

[71] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision–ECCV 2014*, pp. 834–849, Springer, 2014.

[72] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, IEEE, 2007.

[73] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vslam algorithm for robust localization and mapping," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 24–29, IEEE, 2005.