High-Accuracy Differential Tracking of Low-Cost GPS Receivers

Will Hedgecock
will.hedgecock@vanderbilt.edu

Miklos Maroti
miklos.maroti@vanderbilt.edu

Janos Sallai
janos.sallai@vanderbilt.edu

Peter Volgyesi
peter.volgyesi@vanderbilt.edu

Akos Ledeczi
akos.ledeczi@vanderbilt.edu

Institute for Software Integrated Systems, Vanderbilt University
Nashville, TN, USA

ABSTRACT

In many mobile wireless applications such as the automated driving of cars, formation flying of unmanned air vehicles, and source localization or target tracking with wireless sensor networks, it is more important to know the precise relative locations of nodes than their absolute coordinates. GPS, the most ubiquitous localization system available, generally provides only absolute coordinates. Furthermore, low-cost receivers can exhibit tens of meters of error or worse in challenging RF environments. This paper presents an approach that uses GPS to derive relative location information for multiple receivers. Nodes in a network share their raw satellite measurements and use this data to track the relative motions of neighboring nodes as opposed to computing their own absolute coordinates. The system has been implemented using a network of Android phones equipped with a custom Bluetooth headset and integrated GPS chip to provide raw measurement data. Our evaluation shows that centimeter-scale tracking accuracy at an update rate of 1 Hz is possible under various conditions with the presented technique. This is more than an order of magnitude more accurate than simply taking the difference of reported absolute node coordinates or other simplistic approaches due to the presence of uncorrelated measurement errors.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and Embedded Systems

General Terms

Algorithms, Measurement, Performance

Keywords

GPS, localization, differential tracking, relative localization

MobiSys'13 June 25–28, 2013, Taipei, Taiwan

1. INTRODUCTION

Precise node location information is highly important for many mobile systems. For outdoor applications, GPS is quickly becoming the de-facto standard for location determination; however, low-cost receivers, such as those found in smartphones, can exhibit large errors on the order of tens of meters, especially in urban areas or under dense foliage. Higher quality devices provide much better accuracy, but can cost hundreds of dollars, and commercial-grade surveying deployments can reach into the thousands while requiring extensive setup and calibration before they are usable.

While GPS is remarkable in its ability to provide absolute coordinates anywhere on Earth, many applications, including typical wireless sensor network (WSN) deployments, do not need precise *absolute* locations. In fact, most of today's WSN localization methods rely on estimating the range between nodes to derive their locations in a *relative* coordinate system. Furthermore, numerous applications beyond WSNs also need accurate relative locations, such as formation flying of UAVs and other robotic applications, precision agriculture, land surveying, autonomous driving (e.g. platoon formation and collision avoidance), and others. Existing solutions in use today are based on absolute GPS coordinates (e.g. Differential GPS) or additional sensors such as LIDAR or gyroscopes, but these are typically very costly.

The main question this paper seeks to answer is whether it is feasible to build a localization service that provides highly accurate (centimeter-scale) relative positions for a set of mobile nodes relying on pure GPS alone. We introduce **RegTrack** (Relative GPS Tracking), a novel approach for using a network of two or more traditional, low-cost, single-frequency GPS receivers to derive highly accurate tracking vectors between them in a relative coordinate system. One might argue that since most GPS error sources are highly correlated in the same geographic region, simply taking the difference of the reported absolute locations from two GPS devices over time would result in very accurate pairwise location estimates. There are very specific conditions under which this may be true, but due to inherently uncorrelated errors such as minuscule receiver clock biases from GPS time, it is almost never the case. Additionally, substantial errors can accumulate in the presence of multipath due to signal reflections, especially in urban areas or in the likely case that the receivers cannot see the exact same set of satellites. We believe that an increase in GPS accuracy (in the

relative sense) without a corresponding increase in cost will enable the production of novel applications that are either not economical today or are out of reach altogether.

The underlying idea of our approach is simple: let a network of cooperating GPS receivers share their raw satellite measurements with one another to solve for the 3D pairwise relative changes in position between them as opposed to absolute locations. These vector solutions are derived locally on each node with the resulting tracks of all other nodes computed using the local node's location as a reference position. As such, the relative locations of all nodes in a region will be known accurately, regardless of whether the given node is stationary (as in surveying applications) or moving (as in vehicle-based applications). The only caveat is that *the initial relative positions of the sensor nodes must be known* to make the tracking results useful.

While a procedure to automatically determine initial relative positions is a significant missing piece in our goal of a complete localization system, this paper will show that our novel method is already usable for highly accurate tracking in applications such as land surveying or precision agriculture, where the initial node locations are already well-known. It can also be used in its current state for applications involving temporal feature extraction, such as the mapping of certain events (e.g. a lane change or a left turn in a vehicle's track) to a specific time tag, or in applications in which additional sensing modalities may enable periodic re-initialization of any of the participating nodes' relative locations. More importantly, however, we will conclude this paper with an outline on how we plan to definitively solve the location initialization problem without imposing the requirement of the stationary calibration phase present in all existing methods that are able to achieve centimeter-scale localization precision, tracking or otherwise.

An in-depth discussion and evaluation of RegTrack, along with its relevant error sources and corrections, is the subject of the rest of this paper. We show that simple approaches to GPS relative localization do not work well, and our novel tracking approach provides significantly better results than existing methods. In a benign environment with clear views of the sky, our technique showed as much as a 20x improvement over standard GPS positioning algorithms, with relative precision on the scale of centimeters. Additionally, our method produced results ranging from a 3-7x improvement in moderately obstructed and highly dynamic environments, with precision on the order of decimeters, regardless of the baseline length between a pair of receivers which, in our experiments, varied from 0 m all the way up to 3.5 km.

We also show that RegTrack is unlike Differential GPS or Real-Time Kinematic Surveying, both of which require precise knowledge of the absolute locations of one or more of the nodes and can costs thousands of dollars. On the other hand, our baseline accuracy is currently dependent on the initial relative location estimates of the participating nodes, but our methodology requires only instantaneous relative location information for initialization and can be used with low-cost, single-frequency (L1) GPS receivers.

The rest of the paper is organized as follows. We start with a brief overview of GPS and related work. We then continue with an in-depth discussion of the error and observation models necessary to make our algorithm work, as well as the actual methodology used in RegTrack. This is followed by a short description of the hardware platform and software implementation used to carry out this work. We continue with an evaluation of the system and an analysis of the experimental results, and conclude by describing our ideas for the continuation of the work.

2. BACKGROUND AND RELATED WORK

The Global Positioning System (GPS) is a satellite-based navigation system developed by the U.S. Department of Defense in the 1970s for military and civilian applications. Currently, it consists of 31 satellites on six orbital planes about 20,000 km above the Earth's surface. This constellation enables accurate location information to be computed at any place, any time on Earth using trilateration with range measurements between an observer and a few visible satellites.

Range estimation is performed using the times-of-flight of radio signals transmitted by the satellites. Although each satellite is equipped with a highly accurate and synchronized atomic clock, receivers are driven by much less accurate crystal oscillators. This design decision allowed for a truly ubiquitous localization service, but makes it impossible to directly measure the times-of-flight of radio signals traveling at the speed of light with acceptable accuracy. (Note that in GPS research, distance estimates based on an imprecise and arbitrary local clock are called pseudoranges). Instead, the receiver's clock offset from true absolute time becomes another unknown in addition to its 3D position coordinates. Thus, at least four independent measurements (satellites) are needed for a position fix to be computed. For the same reason, a low-cost GPS receiver can provide a highly accurate global time reference (tens of nanoseconds of accuracy) when locked onto the minimum number of satellites [5].

Many high-accuracy GPS algorithms make use of an additional signal observation, called the *carrier phase* or *carrier range* of a satellite signal, to increase positioning accuracy. These values are very precise (centimeter-scale) measurements of the change in range between a satellite and receiver over time, and are sometimes referred to as the *integrated Doppler shift*. Since the initial phase or range at the start of integration is unknown, however, these measurements are ambiguous and must be used in conjunction with other observations, such as the pseudorange, to be of use in localization.

GPS satellites continuously transmit messages using a Code Division Multiple Access (CDMA) spread spectrum technique, which allows them to share the exact same carrier frequency and, more importantly, use a single local oscillator (LO) at the receiver; thus, the phase noise and frequency instability of the LO affect all received signals in the same way. The GPS signal has a rich hierarchical structure with each signal derived from the same atomic clock source. All satellites transmit on at least two carrier frequencies (L1: 1.57542 GHz and L2: 1.2276 GHz) which are modulated by a pseudo-random noise (PRN) sequence. For civilian applications, each satellite uses a unique sequence of 1023 bits (known as the Course Acquisition, or C/A code) transmitted continuously at 1.023 million bits/s, whereas military applications make use of a much longer sequence (the Precision, or P code) at 10.23 million bits/s. Receivers need to know the assigned sequences *a priori* to lock onto the signals and separate them from noise and from each other. At the top of the signal hierarchy, each satellite transmits a low speed navigation message at 50 bits/s that contains its own clock and location (ephemeris) information.

A typical GPS receiver consists of a low-noise *analog frontend* for amplifying, filtering, and down-converting the antenna signal (∼1.5 GHz), an *analog-to-digital converter* (ADC) for digitizing the quadrature IF signal (∼2 MHz), and a *highly parallel digital signal processor* implemented in an ASIC along with a traditional processor core. The key element of this architecture is the custom digital signal processor, operating on tens of independent signal paths (channels) in parallel. Each channel uses a digital correlator (assigned to one of the known satellite codes) to find, track, and correlate a PRN sequence in the received data with the expected PRN signal, shifted and scaled in time. The tracked shift and scale values that give the maximum correlation value represent the signal time-of-flight and Doppler shift (arising from the relative motions of the satellites, with orbital velocities of ∼4000 m/s), respectively.

Each standalone GPS receiver is prone to several sources of measurement error, and understanding the error budget is essential to enhancing accuracy. First, the propagation speed of radio signals in the ionosphere and troposphere is different than in free space; thus, both refract GPS signals differently, potentially causing significant error in the overall result. Second, each GPS satellite broadcasts its orbital position (ephemeris) and clock correction data in the navigation message, as described above. Both pieces of information are based on a predictive model and are, therefore, inaccurate. These errors however, along with the atmospheric effects, are correlated within a bounded geographic region and can be compensated later when more accurate information is available. Other sources of error include multipath and measurement noise in the GPS receiver. It is much more difficult to mitigate these (using more refined antennas or higher precision receivers that increase the cost, size, and power requirements of the unit); fortunately, these effects represent a much smaller percentage of the total error.

Consumer GPS receivers—used mainly for navigational and recreational applications—represent the standalone class and are affected by all error sources described above. The prevalence of these receivers has dramatically driven down cost, size, and power requirements over the past decade. However, for applications where standalone accuracy may be insufficient, the huge price gap to a more precise system can be prohibitive. Our work aims at bridging this gap when only accurate *relative* locations are needed.

High-precision GPS receivers and receiver systems employ a variety of techniques to mitigate these errors. Atmospheric effects can be corrected by using both L1 and L2 carrier frequencies, as refraction is frequency dependent. In fact, as part of the GPS modernization program, there is an additional proposed L5 carrier frequency for civilian receivers. In military-grade GPS, receivers can use the faster P(Y) code for more precise time-of-arrival measurements. Carrier phase measurements and tracking can provide even higher accuracy (comparable to the wavelength of the carrier signal), but generally require extensive post-processing.

Finally, there are several solutions based on differential corrections. While all of these approaches employ more than one receiver, they differ widely in geographic scale, real-time properties, complexity, and their effectiveness in mitigating atmospheric, ephemeris, and satellite clock errors. One end of the spectrum is represented by Satellite-Based Augmentation Systems (SBAS), like the Wide Area Augmentation System (WAAS) used in the U.S., which broadcasts real-time correction information via geostationary satellites based on measurements from several ground reference stations. Most consumer-grade GPS receivers are capable of using this coarse correction information. The other extreme is based on offline post-processing, where highly accurate correction data is calculated at a reference station with a precise clock (such as an atomic clock source) and a well-known location.

Most similar to our work is Differential GPS (DGPS), which also uses multiple receivers to increase GPS accuracy. In DGPS, mobile receivers can calculate their absolute positions with increased accuracy by altering their received satellite measurements according to corrections sent out by one or more static base stations which have been well-calibrated and know their own positions and clock biases to a high degree of accuracy [15]. This method, while providing absolute instead of relative coordinates, requires preliminary setup of expensive, stationary base stations which precludes it from being used in everyday mobile system deployments.

Another class of methods which has been gaining popularity and increasing in accuracy over recent years is called Real Time Kinematic (RTK) satellite navigation [13, 11], which works in a similar fashion to DGPS, except that all coordinates are found using carrier phase measurements with respect to a dedicated sensing station. Again, the sensing station is located at a well-known, pre-defined location and transmits its own position and raw carrier phase measurements to other receivers in the vicinity. As such, participating nodes can calculate their coordinates *relative to a base receiver* with a high degree of accuracy. In RegTrack, there is no dedicated base station or reference node. Every receiver considers itself to be the reference when computing the relative locations of its neighbors. Hence, the architecture is symmetric and there is no single point of failure.

Another drawback of RTK is that the carrier phase measurements it uses are exclusively relative measurements which require knowledge of an unknown (but constant) number of carrier signal wavelengths between a satellite and receiver at an arbitrary point in time to be of any use in absolute positioning; thus, the system must either:

1. Have extremely accurate knowledge of the absolute base station position,
2. Use dual-frequency receivers to minimize single-point errors such as atmospheric delay or measurement noise, or
3. Spend a considerable amount of time solving for the carrier ambiguities in the system - a very active area of research itself [16, 12].

Even after taking these steps into consideration, the system breaks down readily in the face of difficult multipath environments, moderate to severe line of sight obstructions to satellite visibility, satellite losses of lock, or cycle slips, making it most appropriate for use in low-dynamic situations with a clear, broad view of the sky such as nautical navigation or land surveying [9].

There is also ongoing research in enhancing the core methods and algorithms used in standard GPS positioning techniques. Researchers from several Brazilian universities have tried to improve upon the RTK positioning algorithm by using wavelets as the foundation for a mathematical model of the GPS system [4]. Additionally, there has been research

into better methods of DGPS base station design [6], as well as the use of relative positioning in automotive safety applications [2]. The advantage of our research, however, is that it is general and extensible to most any type of mobile sensing network, whereas these papers present very application-specific results which do not provide a means of building upon them or extending them to other areas.

Finally, there are active research efforts in the mobile systems community centered around GPS with the focus of reducing power consumption. Jurdak et al. [8] and Paek et al. [3] explore various duty cycling strategies. Liu et al. [10] go one step further and offload location determination to a remote server, trading off GPS usage for increased communication overhead. These efforts are orthogonal to our approach, as we are focusing on increasing positioning accuracy and disregarding the associated cost in power in this first phase of our project.

3. ERROR MODELING

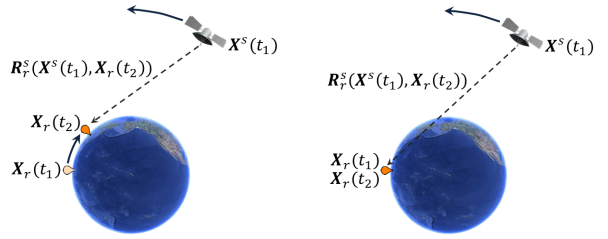
A significant amount of research has been undertaken to characterize the various sources of error that may affect a GPS radio signal as it propagates from satellite to receiver. The generally accepted error sources include both the satellite and receiver's clock synchronization biases from actual GPS system time, delays due to propagation speed retardation as the signal traverses the Earth's atmosphere, multipath interference, antenna phase center offsets, satellite orbital errors, and receiver noise. From these errors, we obtain generally-accepted pseudorange and carrier phase observation models for a single receiver:

$$P_r^s(t) = \rho_r^s(t) + c\tau_r(t) - c\tau^s(t) + d_{iono}^s + d_{tropo}^s + M_{r,\rho}^s(t) + \phi_r^s(t) + E^s + \epsilon_P \quad (1)$$

$$\lambda_{L1}\Phi_r^s(t) = \rho_r^s(t) + c\tau_r(t) - c\tau^s(t) - d_{iono}^s + d_{tropo}^s + B_r^s + M_{r,\Phi}^s(t) + \phi_r^s(t) + E^s + \epsilon_\Phi \quad (2)$$

where P denotes pseudorange, λ_{L1} is the wavelength of the carrier signal in a vacuum ($\sim$19.05 cm), Φ is the carrier phase observation, ρ is the geometric range traveled by the radio signal from receiver to satellite, c is the speed of light, τ is clock bias, d is an atmospheric delay, M denotes multipath, ϕ is any antenna phase center offset, E is the satellite orbital error, B is a constant ambiguity term, and ϵ is receiver noise. Additionally, any subscripts refer to a receiver, superscripts refer to specific satellites, and t values indicate that the corresponding term is dependent on time. These conventions hold throughout the rest of this paper.

In addition to these explicit error sources, there also exists an implicit effect known as the Sagnac effect due to the rotation of the Earth during the time of signal transmission [1]. Once a GPS signal leaves a satellite, it must travel a finite amount of time before it reaches a receiver. Since the Earth is rotating during this time, the propagation time (and hence the raw observations) will either grow or shrink as the receiver rotates away from or toward the incoming signal. If receiver and satellite positions were calculated in a non-rotating, Earth-Centered Inertial (ECI) frame, this would not cause a problem since the receiver position at the time of reception would indicate the correct propagation distance from the satellite at the time of transmission. In GPS, however, satellites transmit their ephemeris data to allow users to calculate their positions in an Earth-Centered,



(a) ECI Frame: Since the coordinate system is fixed in space, the receiver position, $\mathbf{X}_r(t_1)$, at the time of signal transmission, t_1, is not equal to its position, $\mathbf{X}_r(t_2)$, at the time of signal reception.

(b) ECEF Frame: Since the coordinate system rotates with the Earth, the receiver positions at the time of signal transmission and reception appear identical, $\mathbf{X}_r(t_1) = \mathbf{X}_r(t_2)$

Figure 1: **The Sagnac Effect**

Earth-Fixed (ECEF) coordinate frame which rotates along with the Earth. As such, any changes in propagation delay induced by the rotation of the Earth are not taken into account in the calculation of the satellite positions. Figure 1 explains this phenomenon.

Since the ECEF coordinate system rotates along with the Earth, it is clear from Figure 1b that the signal propagation range from satellite to receiver, $\mathbf{R}_r^s = ||\mathbf{X}^s(t_1) - \mathbf{X}_r(t_2)||$, computed under the assumption that the coordinate system has remained stationary between t_1 and t_2 will be incorrect and will not coincide with the observed pseudorange and carrier phase measurements. In order to account for this problem, the user must correct the computed satellite positions for the amount of rotation the receiver experienced during the time of signal transmission. Corrections for this effect are commonly referred to as Earth rotation corrections and a number of algorithms exist for computing them. Left uncorrected, the Sagnac effect can contribute to satellite position errors on the order of 30 m, so this type of effect cannot be ignored.

3.1 Dual-Receiver Error Sources

There exist several subtleties regarding the Sagnac effect that do not seem to appear in literature at this time of writing, stemming from the fact that the Earth continues to rotate in the presence of time delay errors such as atmospheric perturbations or clock biases:

1. Most Earth rotation corrections are calculated using raw pseudorange observations that include two types of errors: signal errors (which directly affect the actual propagation time of the satellite signal) and bias errors (which appear in the observation but are not due to signal delays). Due to the *bias* errors, we cannot know the exact time of reception of the satellite signals, which means we do not know the real propagation time of the signal, and hence cannot accurately estimate how far the Earth has rotated during this time.
2. In the two-receiver case, if measurements are made at two slightly different times, the satellite positions at the time of each signal transmission will have *actually* been different.
3. Also in the two-receiver case, the change in satellite positions during any measurement time discrepancy

will result in an error analogous to the motion of the receivers described earlier in this section; namely, the signal propagation times will have been different, and thus, the Earth will have experienced differing amounts of rotation.

Due to these subtleties, two receivers making observations at apparently the same time (according to their local clocks) may calculate satellite positions that are up to ∼9 m apart from one another. In reality, these two combined error sources typically amount to 1-3 meters of error per satellite.

Ideally, we'd like to take the satellite geometry "snapshots" according to both receivers and extrapolate the measurement data to correspond to what the observations and satellite positions *should* have been if taken exactly at the GPS epoch in the absence of any clock errors. Unfortunately, this requires a slight increase in computational complexity, and extrapolation of measurement data to a specific *reception* time is not quite as simple as it may seem.

4. TECHNICAL APPROACH

Our approach to solving the localization problem posed in this paper involves:

- the addition of a new data conditioning algorithm to remove the two-receiver error sources mentioned in the previous section,
- a new observation model that leverages the time-varying constellation of a set of mutually visible satellites into a single set of equations that we call the *Dual-Epoch, Double-Differenced Model*, and
- the application of a novel tracking algorithm through time to produce accurate 3D pairwise mappings of all the relative node locations in a network of receivers.

4.1 Data Conditioning

As mentioned in the previous section, the optimal way to overcome the error subtleties arising from the two-receiver case is to extrapolate the data from two or more receivers to the exact common point in time when the receivers *should* have made their measurements, namely the GPS epoch. In order to begin the extrapolation procedure, it is necessary to first determine the clock bias of each receiver so that we know how far forward or backward to extrapolate the data. This step can be as simple as solving for the standalone 3D position and bias of each receiver using a simple least-squares optimization routine as described in [5].

Once the receiver clock bias is known, the obvious next step would be to simply rotate each of the receivers' coordinate frames and data sets to coincide with the nominal GPS epoch. This will not result in a satisfactory solution however, especially in the multiple receiver case, because although the receivers themselves can be rotated into a cohesive frame, the satellite positions do not "rotate" with the Earth, but rather follow their own orbits. This means that errors due to the discrepancies between satellite positions at actually different transmit times will still be present. Additionally, the amount of Earth rotation experienced during the measured signal propagation interval cannot be assumed to be identical to the amount of rotation that would have been experienced at the extrapolated time.

In order to correctly extrapolate the data, we need to determine the change in signal propagation time (and by extension, the propagation range, R_r^s) if the measurement had been correctly made at the GPS epoch. Fortunately, the instantaneous change in range can be inferred quite accurately using the Doppler shift observable (necessarily computed by all GPS chips), which directly indicates the line-of-sight *range rate* between a satellite and receiver at any given time:

$$\Delta R_r^s = f_D^s(t) * \lambda_{L1} * t_{bias} \tag{3}$$

where $f_D^s(t)$ is the instantaneous Doppler frequency to satellite s reported by the receiver, λ_{L1} is the vacuum wavelength of the carrier signal, and t_{bias} is the receiver clock bias from GPS system time. Assuming a Doppler shift accuracy of ± 1 Hz, this change in range is guaranteed to be accurate to better than 191 μm.

Since we know both the receiver clock bias and what the measurement observables should have been, we can iteratively calculate the satellite positions, $\mathbf{X}^s$, and Earth rotation corrections to find the updated transmit times of each GPS signal that would have been received at the exact GPS epoch. Essentially, we are using the ΔR_r^s results from Equation 3 to minimize the non-linear function:

$$\epsilon = \Delta R_r^s - (R_r^s(t_t') - R_r^s(t_t)) \tag{4}$$

where t_t is the transmit time of the signal we actually received, t_t' is the updated transmit time corresponding to the signal that would have been received at the exact GPS epoch, $R_r^s(t_t) = ||\mathbf{X}^s(t_t) - \mathbf{X}_r||$ is a constant equal to the satellite-receiver range at the actual time of reception, and $R_r^s(t_t') = ||\mathbf{X}^s(t_t') - \mathbf{X}_r||$ is the calculated range at the exact GPS epoch. Put another way, we are trying to find the corrected transmit time, t_t', which would result in an $R_r^s(t_t')$ that minimizes Equation 4. The following procedure explains how this can be done:

1. Calculate the range at the actual time of reception: $R_r^s(t_t) = ||\mathbf{X}^s(t_t) - \mathbf{X}_r||$,
2. Update the measurement observables by the computed ΔR_r^s value,
3. Correct the signal transmit time according to: $t_t' = t_t - t_{bias}$,
4. Calculate the satellite position at the new transmit time, t_t',
5. Translate the satellite positions into the receiver's ECEF frame using Givens Rotations (see [5]),
6. Calculate the corrected range according to the new satellite positions: $R_r^s(t_t') = ||\mathbf{X}^s(t_t') - \mathbf{X}_r||$,
7. Compute the range error (Equation 4),
8. Iterate through the following until the magnitude of the new range error exceeds that of the previous:
   - Update the possible transmit time, t_t', by either $\pm 1\mu$s,
   - Calculate the new satellite position at this time and translate into the receiver's ECEF frame.
   - Compute the updated range and calculate the corresponding range error.

Luckily, the propagation biases are small enough that only 1-3 iterations of the above algorithm are usually required to find a solution that extrapolates satellite positions to μm accuracy. This can be verified by examining the apparent satellite positions according to two co-located receivers at the same epoch and noting that the positions should now be identical. Table 1 summarizes actual results using this algorithm.

Satellite #	Mean Difference without Extrapolation	Mean Difference with Extrapolation
6	1.7 m	< 1 mm
14	2.0 m	< 1 mm
15	1.4 m	< 1 mm
18	1.4 m	< 1 mm
21	1.9 m	< 1 mm
22	2.1 m	< 1 mm
27	1.8 m	< 1 mm

Table 1: **Average satellite positional differences as seen from three receivers at the same GPS epoch over the course of 30 minutes.**

4.2 Dual-Epoch, Double-Differenced Model

With the independent rotational effects removed and the data extrapolated to the proper GPS epoch, it is now possible to introduce a new set of observation models involving two receivers and a single satellite that contain a minimum amount of residual error while maintaining a stronger geometry than similar triple differencing models. This new set of equations (See Equations 5 and 6) constitutes the Dual-Epoch (dual-receiver, single-satellite), Double-Differenced model.

Unlike the standard double-differenced equations that use data from a single epoch by necessitating the inclusion of a "reference satellite," (see [5]), this model uses two single-differenced observations from consecutive epochs to generate a new observation. The effect of this alteration is twofold:

1. No reference satellite is required; therefore, individual satellites observations are, in fact, independent of one another and do not require additional manipulations in situations when a reference satellite would have changed or disappeared, and
2. Traditional double-differenced carrier range models result in an integer ambiguity that is dependent on the satellite, receiver, and reference satellite; whereas, in the dual-epoch, double-differenced model, this ambiguity term completely cancels out.

The following describes this new model (where $\nabla\Delta$ denotes the double-differencing operation between satellite s and receivers j and k over two epochs):

$$\nabla\Delta P^s_{jk} = \nabla\Delta\rho^s_{jk} + c\nabla\Delta\tau_{jk} + \nabla\Delta M^s_{jk} + \nabla\Delta\phi^s_{jk} + \nabla\Delta\epsilon_P \tag{5}$$

$$\nabla\Delta\lambda\Phi^s_{jk} = \nabla\Delta\rho^s_{jk} + c\nabla\Delta\tau_{jk} + \nabla\Delta M^s_{jk} + \nabla\Delta\phi^s_{jk} + \nabla\Delta\epsilon_\Phi \tag{6}$$

We can see here that the carrier phase model (Equation 6) no longer includes an ambiguity term. If we assume that the multipath, antenna phase center offsets, and receiver noise are negligible, this equation can be re-written:

$$\nabla\Delta\lambda\Phi^s_{jk} = \nabla\Delta\rho^s_{jk} + c\nabla\Delta\tau_{jk} \tag{7}$$

where the result includes only the difference between the change in ranges of two receivers over the course of one epoch and the difference between the two receivers' clock drifts. Equation 7 is unique in that it uses highly accurate carrier phase observations to produce unambiguous estimates of the change in relative ranges between a satellite and two receivers through time without requiring any sort of reference satellite or node.

4.3 Tracking Algorithm

This new observation model can be used directly in the creation of a novel tracking algorithm with the potential for centimeter-scale precision through time. We can note that by subtracting one receiver's carrier phase observation to satellite s from a second receiver's similar observation, the result is the range difference between the two receivers to that satellite, which is equal to the baseline between the two receivers projected onto the line-of-sight unit vector from receiver to satellite. Note that the GPS satellites are such a great distance from the surface of the Earth that the unit direction vectors can be assumed to be identical for both receivers with little effect on the result as long as the two receivers are located in the same geographic region. We perform the same operation at the next time epoch, $t+1$, and then subtract the two results to create the dual-epoch, double-differenced value through time, with model corresponding to Equation 7.

Since $\nabla\Delta\rho^s_{jk}$ represents how much the projected baseline described above has changed over the course of one epoch, the model can be expanded as follows:

$$\nabla\Delta\lambda\Phi^s_{jk}(t) = R^s_k(t) - R^s_j(t) - R^s_k(t-1) + R^s_j(t-1) + c\nabla\Delta\tau_{jk} \tag{8}$$

When the range is written recursively, we see that:

$$\begin{aligned}\nabla\Delta\lambda\Phi^s_{jk}(t) &= (R^s_k(t-1) + \Delta R^s_k) - \\ &\quad (R^s_j(t-1) + \Delta R^s_j) - R^s_k(t-1) + \\ &\quad R^s_j(t-1) + c\nabla\Delta\tau_{jk} \\ &= \Delta R^s_k - \Delta R^s_j + c\nabla\Delta\tau_{jk}\end{aligned} \tag{9}$$

This shows that the dual-epoch, double-differenced result is actually a measure of the change in range between two nodes and a satellite (plus the difference in relative clock drifts). Geometrically, this corresponds to the change in receiver-receiver baseline ($\Delta\mathbf{b}$) projected onto the unit direction vector from receiver to satellite ($\mathbf{a}^s_r$):

$$\nabla\Delta\lambda\Phi^s_{jk}(t) = \Delta\mathbf{b}\cdot\mathbf{a}^s_r + c\nabla\Delta\tau_{jk} \tag{10}$$

It is clear from this result that neither of the initial positions of the receivers must be precisely known in order for the relative motions between them to be accurately tracked. Additionally, it is unnecessary to solve for the motions of *both* of the receivers, since Equation 10 allows for a direct solution of the change in baseline based on our general knowledge of the unit direction vector to satellite s. Thus, the assumption that one node is stationary ($\Delta R^s_j = 0$), even when it is not, causes all of the relative motion to be attributed to the second node ($\Delta R^s_k = \Delta\mathbf{b}\cdot\mathbf{a}^s_r$), which is exactly what is desired, even though it is incorrect in the *absolute* sense.

In order to solve a system of tracking Equations 9 or 10, a simple least-squares optimization routine can be set up to estimate the tracked 3D coordinates through time along with the clock drift difference. As long as consistent satellite locks are maintained with at least four satellites, the relative positions of any node after initialization can be determined via simple dead reckoning (i.e. adding the current tracking result to the last relative position estimate).

We should note here that our implementation does carry out a small amount of *preprocessing* to filter bad data and minimize the effects of multipath and random outliers. To begin, data from satellites appearing lower than 12° on the horizon are disregarded, as signals from such satellites have significantly more atmosphere to traverse than higher-elevated satellites. This has been shown to significantly decrease the accuracy of the signals, not to mention the fact

that lower-angled satellites are more likely to be intermittently blocked by obstructions and influenced by multipath reflections. Next, carrier phase data is tested at each epoch to ensure that no cycle slips (sub-millisecond losses of satellite lock) have occurred, as a slip of only one cycle will introduce ∼19 cm of error into the satellite observation.

Cycle slips are detected by comparison of the received carrier phase observation ($\Phi(t)$) to a prediction of what the observation is expected to be ($\Phi'(t)$), calculated by extrapolating the carrier phase from the previous epoch forward in time using the instantaneous Doppler shift observation ($f_D(t)$), which is immune to such cycle slips. Recall that Doppler shift tells us the frequency offset of a received carrier signal from that of the transmitted signal. The carrier phase observable is obtained by counting the number of cycles that pass in a beat signal formed by mixing the transmitted carrier with a sine wave equal to the frequency of the received signal. Thus, the two observations are linked mathematically, with the Doppler shift giving a direct estimate of the number of additional carrier cycles that should pass over the next second. Thus, the error between the received carrier phase and the predicted carrier phase, where $\Phi'(t) = \Phi(t-1) + f_D(t-1)$, can be used to detect cycle slips. Currently, any error over the threshold of one-half a carrier cycle is deemed to be a cycle slip, and the corresponding measurement is disregarded.

Finally, a rudimentary form of multipath detection is carried out by disregarding satellite observations which maintain a residual error of greater than one-quarter the wavelength of a carrier cycle after the tracking solution has been found. If such an observation exists, the offending satellite is removed and the tracking algorithm is repeated until no further residuals of this magnitude are present.

In addition to allowing for the tracking of relative motions without requiring precise *a priori* knowledge of any node locations, our tracking methodology has the added advantage that neither of the nodes must remain stationary during the tracking procedure due to the fact that:

- Any absolute changes in position are irrelevant since all motion gets attributed to the "relative" receiver, and
- The unit direction vectors are virtually identical for a node that has only moved an order of meters in one epoch.

Note that the only components in tracking Equation 10 are the change-in-baseline term and the double-differenced clock bias (which in this case, equals the difference between the two receivers' clock drifts). We can analyze the error introduced by the assumption that the unit direction vector to a satellite does not change between consecutive epochs numerically for the worst case by noting that a satellite completes a full orbit every 11 hours and 58 minutes. In other words, the largest angle it could trace out in the sky over one second would occur when it passes through the zenith directly overhead. At the standard orbital period, 1 second equates to 0.008357°. This means that the most influence the change in unit direction could have would be $1.0 - (1.0 \times \cos 0.008357) = 10.636 \times 10^{-9}$ times the baseline. This is such a small number that it can be disregarded, but it is very important to note that the error due to this effect increases with increasing baseline lengths. Anything over 1,000 km will start to introduce errors on the centimeter level so our method should only be used for so-called *short baselines* less than 1,000 km.

5. IMPLEMENTATION DETAILS

In order to test the aforementioned algorithms, we implemented a combined version of the error correction procedure and tracking algorithm (called **RegTrack**) as a mobile smartphone app using standard Java. We chose a network of HTC Desire Android smartphones to experimentally evaluate RegTrack. Each phone was paired with exactly one custom Bluetooth-headset [14] that includes a μblox LEA-6T GPS receiver. We chose this particular L1 receiver because it supplies raw measurement data, whereas many low-cost receivers do not. We do not see this as a significant roadblock since the data is necessarily available internally; it is simply a question of providing it directly as an output. The standalone accuracy of the LEA-6T is around 2.5 m with an unobstructed view of the sky.

In our setup, the headset streamed raw GPS data (pseudorange, carrier phase, ephemeris, etc.) over a virtual COM port to the phone, using the UBX protocol of the μblox GPS receiver. The GPS coordinates computed and reported by the GPS receiver were also streamed to the phone to allow for comparison between RegTrack and the built-in algorithm supplied by μblox.

To avoid having a single point of failure, we opted for a distributed, symmetric software architecture where each smartphone shares its raw GPS data with the entire network and runs the localization algorithm independently on the GPS data received from its peers, as well as from the local GPS receiver. We relied on an Android port of the JGroups reliable group communication protocol stack for network-wide broadcast of raw GPS readings [7]. JGroups provides group membership management, guarantees atomic multicast messages (all or none message delivery), maintains packet ordering, and handles retransmissions of lost packets – a variety of features that are required to ensure that all smartphones in our network operate on the same input data set. In our prototype, we configured JGroups to use UDP (IP Multicast) over WiFi as the underlying transport protocol. Essentially, each phone was set up to act as a Bluetooth-to-JGroups bridge that receives raw GPS data via the Bluetooth link and disseminates it to the network, as well as a network receiver, accepting JGroups messages from the network and forwarding them to the solver, which, in turn, computes the relative coordinates of the nodes in the network.

6. EVALUATION

Our evaluation approach compares the accuracy of RegTrack to the ground truth. However, to put the results into context, we also need another methodology to create a reference. We decided to use a simple method we call Absolute-to-Relative (A2R) coordinate determination. In this technique, we simply subtract the absolute positions of one or more receivers from the absolute position of a reference receiver, creating a map of relative locations with respect to the reference receiver's coordinate system. If, by extension, we would like to find the baseline between two mobile nodes that are moving relative to the stationary reference (and to one another) using A2R, we would simply subtract the relative position vectors (from the reference to each of the

Figure 2: **Experiment with three nodes exactly 9 feet apart moving around a running track, relative to an additional stationary node.**

mobile nodes) from one another, which would result in a position vector from one mobile node to the other *as seen from the reference node*.

This method, while simple and easy to implement, is limited by the uncorrelated errors present in the absolute positioning technique used. On the other hand, the μblox chip represents the top-end of several low-cost, L1 receiver lines, so if we are able to demonstrate improvements over it, this shows a useful contribution to localization research.

6.1 Experiment: Running Track

Establishing the ground truth for localization experiments is always difficult, even more so when mobility is involved. The site for our first set of experiments, a running track with precisely marked lanes, helped greatly in this regard; hence, we tried out multiple different scenarios. The first experiment was a stationary setup with four nodes placed on the track at the corners of a rectangle approximately 100x50 m in size. The second experiment utilized a 9-foot long pole. We placed one receiver at each end of the pole and walked around the track multiple times in the middle of lane 4. The pole was kept perpendicular to the direction of movement; hence, the nodes were moving approximately on the border between lanes 2 and 3, and between 5 and 6, respectively. We also placed an additional stationary node on the track. Finally, we constructed an equilateral triangle of three 9-foot long poles and put a node at each corner. This setup was similar to the single pole configuration with an additional node moving along the center of lane 4 ahead of the other two nodes (see Figure 2). Once again, a stationary node was also used.

These experiments are useful because the relative distances of the nodes attached to the poles are precisely known at all times, with only the relative directions changing. As such, they provide us with a way to measure the performance of the technique in the "free mobility" case with extreme precision. The additional static node adds range mobility to the experiment, since both the relative directions **and** distances to the nodes are changing continuously when viewed from this receiver. This provides performance statistics on the technique when used with baselines of varying lengths.

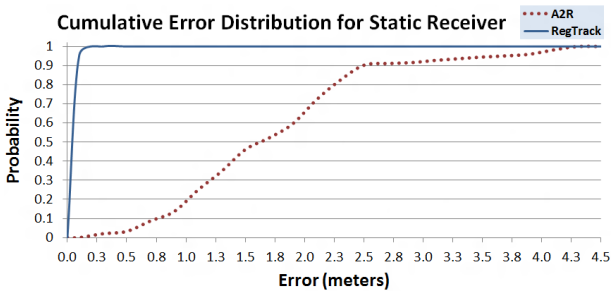


Figure 3: **Cumulative error distributions for a stationary receiver.**

Finally, the stationary node gives us the ability to create absolute ground truth maps for easier visualization, such as using Google Earth as shown in Figure 5.

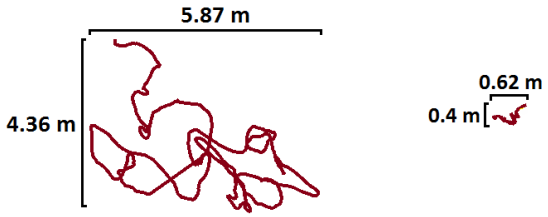
6.1.1 *Stationary Setup*

In this setup, we had four nodes placed at the corners of a rectangle approximately 100x50m in size. The track had trees and a building next to it, so the nodes had a partially obstructed view of the sky. Nevertheless, this can still be considered a benign GPS environment. To test system stability, we let the system run for more than 25 minutes. Note that RegTrack made no assumptions that the nodes were stationary.

To evaluate the accuracy of the system, we set the absolute initial position of each node to be equal to the reported absolute location from the GPS chip (from which the RegTrack algorithm can deduce the initial relative positions of each receiver). This gives a fair comparison since both the A2R and RegTrack algorithms start with identical views of all of the node positions in the network. Each receiver then tracks the motions of the other nodes through time using the pairwise relative tracking vectors between the remote node and itself. The error is determined from the distance between the computed relative location at any point in time and the ground truth. Since the nodes are completely stationary, the ground truth is simply the starting location of each node. In other words, the position of a node in the network relative to any of the other nodes should be identical to its relative position when tracking was initialized for both A2R and RegTrack.

The results shown in Figure 3 indicate the cumulative error distribution for one of the three remote nodes as viewed from an arbitrarily chosen reference receiver. Note that different choices of reference and/or remote node result in almost identical distributions, so only one of the 12 possible options is shown here. In every case, RegTrack produced very accurate results with mean errors ranging from 7.5-10.3 cm (with standard deviations of 4.9-6.8 cm). The errors using A2R were significantly worse, with mean errors ranging from 1.46-1.82 m and standard deviations from 0.7-0.84 m. This is almost a 20-time improvement in average error, with an extremely substantial increase in stability as evinced by the low standard deviations. Figure 4, taken from an actual track experiment, shows what this type of improvement looks like graphically over a 25-minute period. Note that RegTrack resulted in both significantly slower and smaller error accumulations than A2R; however, the errors

in A2R appear to be zero-mean, whereas the RegTrack errors indicate a slow bias over time.



(a) A2R (b) RegTrack

Figure 4: **Static tracks using A2R vs. RegTrack.**

6.1.2 *Fixed 9-foot distance*

The results of the experiment with two mobile nodes on a 9-foot (2.74 m) pole and a single stationary node are probably the easiest to visualize. Figure 5 shows the ground truth of this experiment—the tracks of the two mobile nodes in Google Earth. Figure 6 shows the tracks estimated by A2R *as seen by the stationary node*, under the assumption that its precise absolute location is known. Finally, Figure 7 shows the tracks computed by RegTrack, indicating the relative location vectors between the stationary node and the two mobile nodes through time (and again assuming that the stationary node is perfectly localized and the starting positions are known). In other words, these results indicate the tracks of the two mobile nodes as seen from the reference with relative ranges varying from 0 to ∼125 m. In the latter two figures, the ground truth is shown using a dashed line, and the background image is removed for clarity.

As the figures show, the tracks reported by RegTrack are generally closer to the ground truth. For example, the two nodes are always perpendicular to the track and finish almost exactly where they started. This is also apparent in Figure 8, which shows the estimated distance between the mobile node pair (as seen from the stationary reference) as it moved around the lap. Notice by comparing Figure 6 to Figure 8 that the A2R tracks are not only inconsistent in their ability to stay in the correct lane position, the nodes also frequently move in an orientation that is not perfectly parallel to the track, as indicated by the greater than 9-ft range errors in Figure 8 that correspond to track locations in Figure 6 at times when they visually appear to be *less* than 9-ft, and also in the starting locations of the two nodes in Figure 6 (with a range of close to 5 m as opposed to the expected 2.74 m).

Knowing the ground truth between the two mobile nodes to be a fixed 9 feet, or 2.74 m, the mean error using A2R (as seen from the stationary reference node) was 89 cm with a standard deviation of 68.5 cm. Taking the *difference of the vectors* provided by RegTrack between each mobile node and the reference, the mean error was 16.8 cm with a standard deviation of 8.4 cm.

Note that we can also track the relative distance of the two mobile nodes without an explicit reference node. Each of the mobile nodes can consider itself to be the reference and track the relative location of the other. (This is the default operating mode of RegTrack, and we refer to it as "RegTrack-Direct" in the rest of this paper. The stationary reference node was only necessary to give results over changing baseline ranges and to show the absolute track in

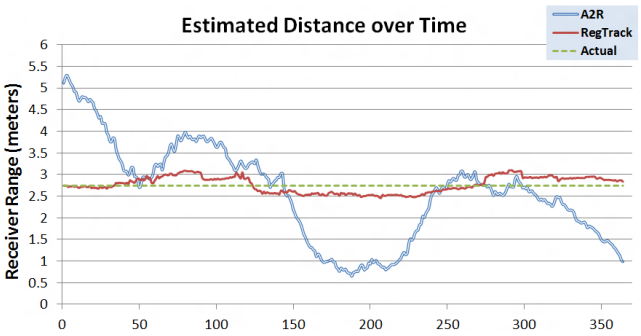


Figure 8: **Estimated distance between the mobile nodes (as seen by the stationary node) as a function of time as they moved around the track.**

Figures 6 and 7.) This method provides a mean error of 14.2 cm according to both receivers, with a standard deviation of 7.8 cm. It is interesting to note that the relative errors are very similar whether we use a reference node up to 125 m away or just the two mobile nodes that are less than 3 m apart. In conclusion, this experiment showed a factor of 6 improvement over the A2R approach, with almost a 10-time improvement in stability as indicated by the standard deviation values.

6.1.3 *9-foot equilateral triangle*

This experiment was very similar to the previous one with the addition of an extra mobile node. Again, our approach is compared to A2R; that is, the value computed by taking the difference of the absolute positions reported by the three μblox receivers involved and the stationary reference. The combined mean error of RegTrack for all three baselines was 23.2 cm, and the standard deviation came to 17.6 cm, which was a 3.5-time improvement over the A2R standard deviation of 63.2 cm. The mean error of A2R was 1.12 m, giving us a 6.6x improvement in overall mean error.

Notice that in this setup, each mobile node sees the other two at a 60-degree angle. Since RegTrack computes location vectors, we can estimate this angle easily. It is also straightforward to do with A2R. Figure 9 shows the distribution of all three angle estimates throughout the lap. These results show a marked improvement over A2R, with a clear spike in the distribution curve around the 60-degree mark. The standard deviation of the RegTrack results was 8°, while the standard deviation of the A2R results was 33°. Note, however, that due to the close proximity of the receivers, a small error in range can result in significant angular errors.

6.2 Experiment: Automobile

Since multiple experiments in a fairly benign environment all produced marked improvements over A2R, we decided to test the limits of our system by running experiments in both a partially-obstructed area and under high-dynamic conditions, namely driving in a car. We should note here that the data for the following experiments was, by necessity, logged and then processed after the experiment instead of in real-time, due to the fact that we were driving in an area far wider than a single wireless router was able to cover. In the future, we may investigate alternate means of data dissemination to

Figure 5: **Ground truth**

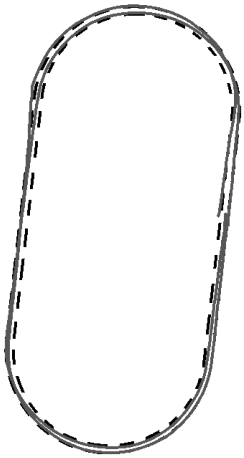


Figure 6: **A2R.**

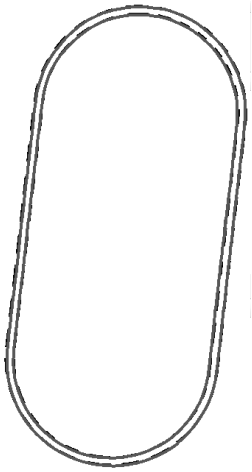


Figure 7: **RegTrack**

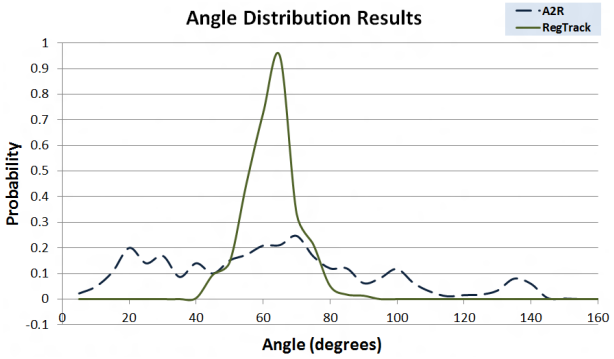


Figure 9: **Angle distributions for one lap.**

overcome this problem, such as ad-hoc phone-to-phone networks or even a cellular data implementation.

In carrying out the following experiments, we noted (as expected) that there were times when one or more of the receivers could not maintain enough satellite locks to continue using RegTrack. Since re-calibration was out of the question in such dynamic experiments, we overcame this problem by simply extrapolating the node tracks based on the most recent velocity estimates of the receiver. So long as the velocity did not change abruptly during an outage, this solution worked sufficiently well for us to be able to evaluate the accuracy of our algorithms in these more difficult environments. Note, however, that this is an extremely naive solution to dealing with temporary losses of satellite locks. In the future, we plan to devise better, more robust ways of dealing with this problem, but for this phase of the project, we simply needed a way to keep the tracks moving to evaluate our technique under these dynamic conditions.

The entirety of this experiment involved placing three GPS nodes on top of a car and driving in a 12.2-km loop, where the course was broken up into two distinct parts. The first part involved driving at relatively low speeds (<50 km/h) between 2-4 story tall buildings and under numerous leafy trees; in other words, a slightly higher dynamic environment than in previous experiments with significantly more opportunities for multipath, cycle slippage, and losses of lock. This was followed by an extremely high-dynamic situation in which we drove on an interstate highway, where the main obstructions were the numerous overpasses that temporarily blocked satellite visibility. Finally, we exited the interstate and closed the loop to return back to our starting point.

In addition to the three mobile nodes on the roof of the car, we also set up a stationary node at the starting position to test the accuracy of our algorithm when a receiver is tracking multiple nodes many kilometers away (a maximum of 3.5 km in this case).

6.2.1 Driving in Obstructed Area

Three nodes were placed on top of a vehicle situated in a wide open parking lot with a clear view of the sky, two in the front approximately 1.04 m apart and a third on the back left side approximately 1.35 m behind the front left node. After initialization, the car was driven slowly through a very narrow alleyway with buildings on either side and significant tree cover. The car then drove through a one-lane, suburban-type road with occasional tree cover, followed by a main road (at approximately 50 km/h) with tree cover. The total length of this portion of the experiment was 1.478 km, which took 5 minutes to complete including a few stops due to traffic. Table 2 summarizes the results as they accumulated over time, where "RegTrack" indicates the pairwise node ranges as computed with reference to the stationary node, and "RegTrack-Direct" indicates the results as computed directly (pairwise) from each of the mobile nodes on the vehicle relative to one another, excluding the stationary node.

Somewhat surprisingly, the overall RegTrack results were only marginally less accurate than those from the running track, even though this experiment included a much more difficult GPS environment. We would have expected multipath and losses of satellite locks to have more of an impact, but it is possible that the close proximity of the receivers to one another meant that any multipath effects were actually quite similar for all receivers, and therefore canceled out.

It is interesting to note that the direct application of RegTrack from each of the roving nodes to one another resulted

in less accurate results than when viewed from the stationary node by about a factor of 3. The most likely reason for this is that the stationary node had a clear view of the sky, meaning it was more likely to have a larger number of visible satellites in common with each of the mobile nodes at any given time than they did with one another. In either case, RegTrack outperformed the A2R method by a factor of 2 for the direct case, and a factor ranging from 4.5-6.5 when using the stationary node as a reference.

6.2.2 *High-Speed Driving*

For this portion of the driving test, we entered an interstate highway with a clear view of the sky, obstructed only by occasional overpasses. It is important to note here that we did not stop and re-initialize the RegTrack algorithm, but rather continued on from the end of the previous portion of the driving experiment. Therefore, any error that had accumulated during that time was also present to begin this test.

The total length of the path traversed during this experiment was 7.027 km, taking 4.75 minutes to complete. The receivers were moving an average speed of 90 km/h during this time. The results for two of the mobile node pairs are summarized in Figure 10:

Clearly, the improvement of RegTrack over A2R in the mean error graph for mobiles nodes #1 and #2 looks significantly better than the results from nodes #2 and #3. We included both sets of results to emphasize an important point, namely to show how crucial it is that the initial relative positions of the nodes be set properly. Nodes #1 and #2 correspond to the front two receivers on the car, which, from looking at both GPS tracks and analyzing the result data, accrued only a modest amount of error over the first part of the driving experiment. Node #3 was the node on the back of the car directly behind node #2, and it was apparent that this node had accumulated more error than the other two in the directional sense (in other words, the reported ranges between the nodes were quite close to the actual ranges, however the direction vectors from node #3 to the other nodes were slightly incorrect). As such, the tracking algorithm could have been working flawlessly and the comparison to ground truth would still show significant error since the initial positions at the beginning of this portion of the experiment had already accumulated error. Again, this just serves to show the importance of correct position initialization in any tracking algorithm.

Since the initial positions of nodes #1 and #2 were shown to be more closely representative of the ground truth, the results in the first graph are more indicative of the performance of the RegTrack algorithm. Once again, RegTrack

Method	Distance Traveled	Mean Error	Standard Deviation
RegTrack	500 m	15.2 cm	10.5 cm
RegTrack-Direct	500 m	50.7 cm	24.0 cm
A2R	500 m	93.6 cm	36.5 cm
RegTrack	1 km	15.0 cm	10.2 cm
RegTrack-Direct	1 km	48.4 cm	24.1 cm
A2R	1 km	97.5 cm	36.5 cm
RegTrack	1.48 km	25.3 cm	24.8 cm
RegTrack-Direct	1.48 km	62.2 cm	36.4 cm
A2R	1.48 km	113.4 cm	55.6 cm

Table 2: **Cumulative accuracy for one of the node pairs while driving through a difficult GPS environment.**

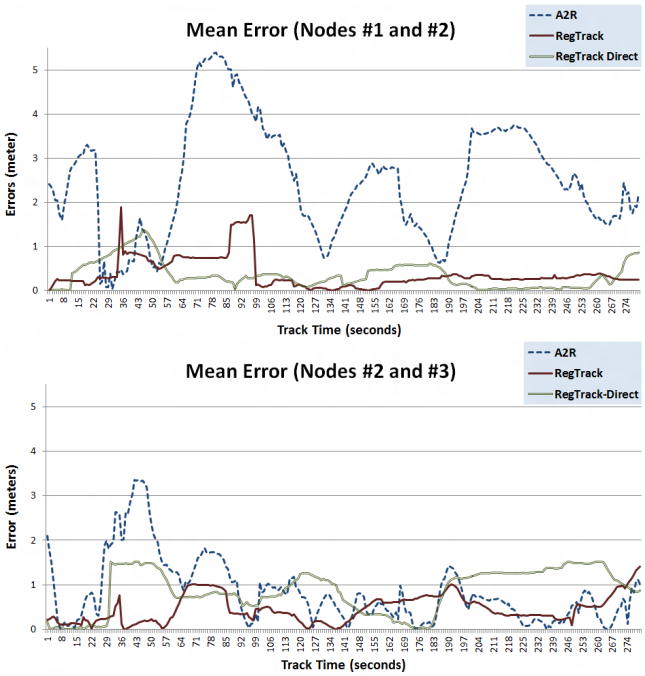


Figure 10: **Mean errors over time for two of the mobile node pairs attached to the roof a car driving along the interstate.**

outperformed A2R by a significant margin, only this time the direct localization approach (i.e. the positions of the mobile nodes as seen from one other, excluding the stationary node) performed slightly better than the stationary node case (as expected). This again furthers our theory that low satellite visibility in the first part of the experiment was to blame for the decrease in accuracy.

For numerical comparison, the mean error of the A2R method was 2.47 m with a standard deviation of 1.29 m, whereas the mean error of the RegTrack method was 37.6 cm with a standard deviation of 34.5 cm, and the mean error of the RegTrack-Direct method was 34.8 cm with a standard deviation of 31 cm. As a frame of reference, this level of accuracy allows for quite obvious feature extraction of important driving events such as lane-changing as shown in Figure 11, approximately 1 km into the experiment. For a video of the entire experiment, go to `http://tinyurl.com/bvelw4c`.

Figure 11: **Track of car carrying out a lane change.**

6.2.3 *Closing the Loop*

At the very end of the interstate experiment, our vehicle passed under a wide overpass while at the same time subtly

changing directions. Also, each of the nodes re-acquired satellite locks at slightly different times. As a result, the simple velocity extrapolation method used throughout the rest of the experiment failed in this case, with the 3 mobile receivers emerging from the overpass with tens of meters of distance between them as seen from the stationary reference node.

One node, however, was able to re-acquire its satellite locks quickly enough to have only accrued a minimal amount of error, enabling us to continue to track its position relative to the stationary node as we closed the loop and returned back to the starting position. This required an additional 3.696 km of driving over 5.2 minutes. At this point, the remaining node appeared 2.3 m away from its starting position. While this number by itself is quite impressive, considering RegTrack relies on dead reckoning and the entire experiment lasted 15 minutes and covered 12.2 km of moderate to difficult GPS terrain, we expect that the result may have even been better had the final overpass not temporarily disabled the RegTrack algorithm.

The final important thing to note about this driving experiment is the level of accuracy that was able to be maintained relative to a stationary reference node, even though the range to that node varied up to 3.5 km away. This shows that our error analyses and the corrections used in RegTrack produce precise enough satellite observations that accuracy is largely independent of the pairwise distances between the mobile nodes being tracked. As a summary, the following table shows the results of all experiments described in this section:

Experiment	Method	Mean Error	Standard Deviation
Track: *Stationary*	RegTrack	9.1 cm	6.1 cm
	A2R	168.4 cm	77.3 cm
Track: *9-ft Pole*	RegTrack	16.8 cm	8.4 cm
	RegTrack-Direct	14.2 cm	7.8 cm
	A2R	89.0 cm	68.5 cm
Track: *Triangle*	RegTrack	23.2 cm	17.6 cm
	A2R	112.0 cm	63.2 cm
Driving: *Obstructed/ Multipath*	RegTrack	25.3 cm	24.8 cm
	RegTrack-Direct	62.6 cm	36.4 cm
	A2R	113.4 cm	55.6 cm
Driving: *High-Speed*	RegTrack	37.6 cm	34.5 cm
	RegTrack-Direct	34.8 cm	31.0 cm
	A2R	247.0 cm	149.0 cm

Table 3: **Summary of experimental results for all nodes in each experiment. Note that in the stationary case, the RegTrack and RegTrack-Direct methods of evaluation are synonymous.**

7. FUTURE WORK

Before we can fully realize our vision of a complete standalone localization system, RegTrack needs an additional component to compute the instantaneous relative positions of any of the nodes participating in the localization process following a temporary loss of all satellite locks. This is also necessary to initialize the tracking at start-up time and because any complete loss of satellite locks resulting in fewer than four visible satellites will necessitate a complete re-initialization of the tracking algorithm. To tackle this problem, we have a very clear idea for the continuation of our work. Aside from the obvious advantage of being able to accurately track a set of receivers through time, the centimeter-scale results obtained from our algorithm are useful because they enable us to mathematically reverse the relative motions of a receiver such that an initial receiver-receiver baseline can be estimated *as if both receivers had remained stationary*. That is, it will be possible to estimate the relative positions of two receivers at any given time based on data from multiple epochs.

To make this clearer, assume that a node exists with a general idea of its (reference) location on Earth and is trying to localize a second roving node. Since only a general idea of the absolute location of a node is required to produce an extremely accurate satellite direction vector, it can be assumed that the reference location is precisely known and does not change over time; likewise, this paper has demonstrated that the relative motions of a roving node can be tracked very accurately, leaving only the coordinates of the initial baseline vector to be determined. Thus, a system of range equations can be written as a function of the initial baseline coordinates with respect to the "stationary" reference at some arbitrary time. Using this model as a basis for our future work, our next research step will be to determine a way to use it to provide centimeter-scale relative positions *without* requiring tracking initialization.

A final natural extension to RegTrack that has the potential to improve the overall results is to perform a network-wide localization step using the the individual location vectors computed by the nodes. In surveying terms, this is referred to as "closing the loop," whereby the addition of any number of relative vectors which form a closed-loop from one receiver back to itself should equal 0. In a network of n nodes, $(n-1)$ such vectors are sufficient to compute the relative locations of any of the nodes. In our system, we will have far more than this required number, resulting in an overdetermined system. Any number of optimization approaches could be used to estimate the node locations while minimizing overall error.

8. CONCLUSION

This paper presented a novel approach to GPS-based differential tracking of mobile nodes. The evaluation of the technique under various conditions clearly showed the feasibility of achieving an order of magnitude improvement in localization accuracy over the traditional absolute positioning algorithms provided by standard GPS while relying on low-cost, single frequency receivers alone.

Unlike most GPS-based navigation solutions, our approach does not snap positions to maps or try to use a dynamic model for the motions of the receivers (other than during satellite losses of lock). Also unlike other high-precision GPS localization techniques used in applications for which our system may have utility in its current tracking-based state (e.g. Differential GPS, Real-Time Kinematic navigation, or any number of post-processing methods used in applications such as land surveying, precision agriculture, temporal feature extraction, or multiple sensing-modality localization apps), our approach does not require a stationary calibration phase, using only instantaneous relative node locations for tracking initialization.

By allowing a network of GPS receivers to share their raw satellite measurements with one another, we were able to achieve centimeter-scale tracking accuracy via:

- Better modeling and correction of the error sources unique to the two-receiver localization case,

- Creation of a new observation model called the *Dual-Epoch, Double-Differenced* model which is able to use independent satellite observations from two receivers through time to create a set of unambiguous, highly-accurate carrier phase equations representing the relative satellite-receiver changes in range, and
- Integration of this new model into a tracking algorithm that produces pairwise 3D location vectors between a local node and any number of remote receivers.

In three different walking experiments, we were able to achieve improvements ranging from 7-20x better than the corresponding "standard" GPS positioning techniques, and sub-meter accuracy was likewise achieved while driving at various speeds, with varying baseline lengths, and under difficult GPS conditions. *The demonstrated high-accuracy of RegTrack is notable because the approach uses a form of dead reckoning in which errors accumulate over time.*

Likewise, our experiments included receiver-receiver baseline lengths ranging from 0 m all the way up to 3.5 km, with little to no impact on the precision of the results. We conclude, therefore, that our method is quite robust to changing baseline lengths (within reason), and the limiting factor may be the various mathematical assumptions mentioned in Section 4.3 regarding use of the Dual-Epoch, Double-Differenced model in tracking. As such, we anticipate similar levels of accuracy to be achievable for baselines up to $\sim$1000 km in length.

Finally, we note that our method has several unique characteristics that make it attractive for a wide array of use cases. First, since our tracking algorithm works on pairwise sets of satellite data, its computational complexity does not grow with the addition of GPS receivers to the network. In fact, with modern processing speeds, the primary bottleneck is the communication bandwidth required to broadcast the raw satellite measurements among a potentially large number of receivers. Since the amount of data transmitted from a single node every second is actually quite low (on the order of 500 bytes/second), there are better ways (than the naive multicast approach used in our experiments) of approaching this problem if and when communication scalability becomes a problem. Secondly, since each and every node considers itself to be the "reference node" in our technique, the system is symmetric and contains no single point of failure. If any node experiences problems, it will simply drop out of localization participation until the problem is resolved.

The approach presented in this paper represents a large first step toward our ultimate goal of a complete standalone relative localization system using only low-cost, commercial GPS receivers. The research in its present state already has significant utility in the realm of precision tracking, which currently has a notable lack of simple, low-cost solutions. While our technique is elegant in its simplicity, we truly believe that it has the ability to open doors for further research and applications that are either difficult, out of reach, or prohibitively expensive given the current state of the art.

9. ACKNOWLEDGMENTS

The research presented in this paper was supported in part by the National Science Foundation under the CNS-NeTS program (Grant #1218710), a Google Research Award, the DARPA Transformative Apps program, Vanderbilt University, and Project #TAMOP-4.2.2.A-11/1/KONV-2012-0073 of the European Union and the European Social Fund. Additionally, we would like to thank Thyagarajan Nandagopal for his helpful suggestions regarding our research. Finally, we are very grateful to our anonymous paper reviewers and shepherd, Marco Gruteser, for their collective input. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

10. REFERENCES

[1] Neil Ashby. The Sagnac effect in the Global Positioning System. *Relativity in Rotating Frames*, 2012.

[2] Chaminda Basnayake, C. Christopher Kellum, James Sinko, and Joseph Strus. GPS-based relative positioning test platform for automotive active safety systems. In *ION GNSS 19th International Technical Meeting of the Satellite Division*, ION GNSS International Technical Meeting, pages 1457–1467, Fort Worth, TX, USA, September 2006.

[3] K. Borre, D.M. Ákos, N. Bertelsen, P. Rinder, and S.H. Jensen. *A Software-Defined GPS and Galileo Receiver*. Springer Verlag, 2007.

[4] Eniuce Menezes de Souza, Joao Francisco Galera Monico, and Aylton Pagamisse. GPS satellites Kinematic Relative Positioning: Analyzing and improving the functional mathematical model using wavelets. *Mathematical Problems in Engineering*, 2009, June 2009.

[5] Christopher Hegarty Elliott D. Kaplan. *Understanding GPS: Principles and Applications*. Artech House, 2.00 edition, 2005.

[6] Jay Farrell and Tony Givargis. Differential GPS reference station algorithm – Design and analysis. *IEEE Transactions on Control Systems Technology*, 8:519–531, May 2000.

[7] Jgroups: A toolkit for reliable multicast communications. `http://www.jgroups.org`.

[8] Raja Jurdak, Peter Corke, Dhinesh Dharman, and Guillaume Salagnac. Adaptive gps duty cycling and radio ranging for energy-efficient localization. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 57–70, New York, NY, USA, 2010. ACM.

[9] Lao-Sheng Lin. Application of GPS RTK and total station system on dynamic monitoring land use. In *Proceedings of the ISPRS Congress Istanbul 2004, Commission VII*, ISPRS Congress, Istanbul, Turkey, July 2004.

[10] J. Liu, B. Priyantha, T. Hart, H.S. Ramos, A.A.F. Loureiro, and Q. Wang. Energy efficient gps sensing with cloud offloading. 2012.

[11] RTKLIB: An open source program package for GNSS positioning. `http://gpspp.sakura.ne.jp/rtklib/rtklib.htm`, 2011.

[12] Eric Sutton and Rockwell Collins. Calibration of differential phase map compensation using single axis rotation. In *Proceedings of the 11th International Technical Meeting of the Satellite Division of the Institute of Navigation*, ION GPS 1998, pages 1831–1840, Nashville, TN, USA, September 2008.

[13] TxDOT survey manual - GPS RTK surveying. Texas Department of Transportation, http://onlinemanuals.txdot.gov/txdotmanuals/ess/gps_rtk_surveying.htm, April 2011.

[14] Péter Völgyesi, Sándor Szilvási, János Sallai, and Ákos Lédeczi. External smart microphone for mobile phones. In *Proceedings of the International Conference on Sensing Technology*, ICST, 2011.

[15] What is differential GPS. Rose India Technologies, http://www.roseindia.net/technology/gps/what-is-Differential-GPS.shtml, February 2008.

[16] Yang Yunchun, Ronald R. Hatch, and R. T. Sharpe. Minimizing the integer ambiguity search space for RTK. *Wuhan University Journal of Natural Sciences*, 8:485–491, 2003.