Energy Efficient GPS Acquisition with Sparse-GPS

Prasant Misra*, Wen Hu†, Yuzhe Jin‡, Jie Liu‡, Amanda Souza de Paula§, Niklas Wirström¶, Thiemo Voigt¶‖
*Robert Bosch Center for Cyber Physical Systems, Indian Institute of Science, Bangalore, India
†CSIRO Computational Informatics, Brisbane, Australia
‡Microsoft Research, Redmond, USA
§University of Sao Paulo, Sao Paulo, Brazil
¶SICS Swedish ICT, Stockholm, Sweden
‖Uppsala Universitet, Uppsala, Sweden
Email: prasant.misra@rbccp.org*, wen.hu@csiro.au†, {yuzjin, jie}@microsoft.com‡,
amanda@lcs.poli.usp.br§, {niwi,thiemo}@sics.se¶

***Abstract*—Following rising demands in positioning with GPS, low-cost receivers are becoming widely available; but their energy demands are still too high. For energy efficient GPS sensing in delay-tolerant applications, the possibility of offloading a few milliseconds of raw signal samples and leveraging the greater processing power of the cloud for obtaining a position fix is being actively investigated. In an attempt to reduce the energy cost of this data offloading operation, we propose Sparse-GPS[1]: a new computing framework for GPS acquisition via sparse approximation. Within the framework, GPS signals can be efficiently compressed by random ensembles. The sparse acquisition information, pertaining to the visible satellites that are embedded within these limited measurements, can subsequently be recovered by our proposed representation dictionary. By extensive empirical evaluations, we demonstrate the acquisition quality and energy gains of Sparse-GPS. We show that it is twice as energy efficient than offloading uncompressed data, and has 5-10 times lower energy costs than standalone GPS; with a median positioning accuracy of 40 m.**

***Index Terms*—GPS, synchronization, location sensing, energy efficiency, sparse approximation, compressed sensing**

I. Introduction

Location is an important service in mobile sensing. The global positioning system (GPS) is the most pervasive technology that provides this fundamental service. The ubiquity of GPS has, henceforth, grown beyond billions of smart phones to embedded devices for enabling many novel outdoor applications across several domains. GPS receivers have, therefore, become more versatile in terms of cost, size and weight; but are *still* demanding in energy usage. It is an artifact of the computationally intensive GPS receiving operation, which accounts for more than 80% of the total energy expenditure of the sensing platform on which it is coupled [1]–[3].

The high energy profile is a combined effect of two primary factors. *First*, the GPS Ephemeris data, which contains the time and satellite trajectory information, are sent by the satellites at a very low data rate of 50 bps. As a result, a standalone GPS receiver needs to be turned on for up to 30 seconds in order to receive a complete data packet from the satellite. *Second*, the task of identifying and tracking the visible satellites, decoding their navigational information and performing the least-square calculation involves a significant amount of processing. It gets more intensive due to the weak GPS signal strengths (about 20 dB below the noise level); and Doppler frequency shifts (≈ 4.2 kHz) caused by the satellite motion (and receiver movement on the ground) [4]. As a consequence, the first factor makes it difficult to duty-cycle the GPS receiver for saving energy; while the second necessity introduces the need for a sophisticated CPU for complicated computations. The existing state-of-the-art methodology of obtaining a GPS position fix is, therefore, *not* suitable for many mobile sensing applications such as livestock [5] and wildlife monitoring [6]; which need non-intrusive position tracking support on resource constrained platforms (such as sensor nodes) for long durations.

Application context and challenges. This paper is motivated by the need to monitor *megabats*: flying foxes, a species that is responsible for the spread of deadly diseases such as Hendra, Ebola, and SARS-like Coronavirus. For this application, the existing platform (packaged as a collar) is a custom designed small, lightweight, battery powered, multi-modal sensing capable wireless sensor node. It is severely *constrained* in terms of available energy resources required for sensing, processing, storage and communication; and can only harvest limited energy. It *must* collect data in a delay-tolerant way; and also be capable of long-term, unsupervised operation during which it may not be in contact with the base station for data offloading. It uses GPS as the *primary* sensor for location tagging; which is the best modality for outdoor localization, but leads to faster energy depletion than non-GPS aided sensing techniques [6].

With respect to delay-tolerant applications [5]–[7], existing solutions provide coarse-grained GPS activity control. They consider GPS as a *black-box* module, and tradeoff the energy expense in deriving the position fix by adaptively using it with other sensors [3]. Ramos et al. [8] and Liu et al. [9], by their recent LEAP and CO-GPS solutions, have shown that *significant* energy savings can be achieved by exploiting the coding nature of the GPS signal, and splitting the post-processing mechanism into local and cloud computation. Such a solution offers a two-fold advantage of: (i) significant duty-cycling of the GPS device, as it only needs to run for a few milliseconds at a time to collect the most crucial information from the satellites; and (ii) avoids the need for a powerful local CPU, by transferring raw data to the cloud and leveraging its greater processing power to calculate the location. While such a data acquisition and processing model is well suited for the target application, the task of offloading data to the base station (or a cloud server) introduces an *additional* cost in energy.

[1]Prasant Misra* was a postdoctoral fellow, and Wen Hu† was a visiting researcher at SICS Swedish ICT in Stockholm during the course of this work.

Motivated by the need to *limit* this expenditure, we propose Sparse-GPS.

Contribution. Sparse-GPS (or, S-GPS) is based on a mechanism to *compress* and transmit the condensed GPS data to the offloading device; wherein the coarse information of the carrier frequency (from Doppler shifts), and time delay of the satellite signals can be efficiently recovered to determine the visible satellites. Cross correlation is the *conventional* method of obtaining these parameters[2]; but, given its *sparse* information content, we make use of the theoretical results in sparse approximation to achieve similar performance. The underlying information theory suggests that a signal can be recovered by ℓ^1-minimization [11], when its representation is sufficiently *sparse* with respect to an over-complete dictionary of base elements. The feasibility of such a mechanism was first demonstrated by Misra et al. [12] for static ranging scenarios with acoustic signals. However, recovering information from compressed GPS (radio) signal is non-trivial due to its *very weak* signal strength, and *two-dimensional* (time delay, frequency shift) search mechanism.

In this paper, we overcome these challenges, and make the following contributions.

• We introduce Sparse-GPS, a *new* computing framework for energy efficient GPS acquisition via sparse approximation. We propose a *new* dictionary that combines the information sparsity along *all* search dimensions, and achieves up to *two* order of magnitude better sparse representation than standard DCT and FFT domains.

• We analyze the dependency of the received signal-to-noise ratio and satellite acquisition count on *data length*. We show that using a data length of 10-20 ms over 2 ms, there is a *high* probability of acquiring 50% additional satellites with both the conventional and S-GPS method.

• We demonstrate the GPS acquisition capability and energy gains by empirical evaluations on real GPS signals. We show that S-GPS is *twice* as energy efficient than offloading uncompressed data, and has 5-10 times *lower* energy cost than a standalone GPS; with a median positioning error of 40 m.

In light of our contributions; we elaborate on the design of S-GPS in Section II, present its evaluation in Section III, survey related work in Section IV, and summarize our work with concluding remarks in Section V.

II. The Design of S-GPS

A. Motivating Application

To ground our discussion, we consider the application context of monitoring flying foxes. Wildlife managers are greatly interested in studying their ecology as flying foxes are the potential carriers of a number of infectious diseases that threaten livestock and humans; and are recognized as agricultural pests that cause damage to fruit crops worth millions of dollars. In this regard, position tracking is an *important* requirement for obtaining their roost camp locations, and also to understand their movement patterns (of which, little is known).

[2]Another approach could be to calculate the cross-correlation result on the receiver itself, and only offload the resultant coefficient. However, on-board processing is still expensive as a large fraction ($\approx 75\%$) of the GPS energy is consumed by this operation [10].

System architecture. The system is composed of *three* units. The *mobile sensing* unit consists of smart collars that are deployed on flying foxes (typically, attached to the neck of the bat by trained, expert handlers). They are responsible for gathering and logging sensed data. The *base station* unit consists of resourceful gateway nodes that, on one hand, facilitate offloading of data from the smart collars over a low-power wireless radio connection; while on the other hand, upload the same to the central servers over a 2G/3G connection. This static infrastructure is placed at known bat roost camps, and can be expanded as newer ones are discovered. The central storage and control *servers* form the final system unit where data is permanently stored, processed, and analyzed.

Platform details. Each smart collar, with a combined weight of less than 30 g, incorporates a system-on-chip (SoC) with a GPS module [13]; inertial, acoustic, air pressure and temperature sensors; flash storage, two solar panels, and a 300 mAh Li-Ion battery. The SoC comprises of a microcontroller core, and an IEEE 802.15.4 complaint radio transceiver.

Operational goals. Flying foxes are known to travel into/across remote, inaccessible areas where cellular coverage may not be available. The mobile sensing platform, therefore, must operate with high delay tolerance in collecting both daytime and nighttime position logs and operate over long periods; until the bats return to the roost camps where the sensed data can be transfered to the base station. Hence, in this application, energy is a *critical* resource and its conservation is *greatly* valued.

Road-map. Of all the sensing units on the platform, GPS is the most energy consuming module. For decreasing the energy cost of obtaining a position fix, we adopt an approach similar to Liu et al. [9] where the raw GPS baseband signals are stored on-board, and the intensive location computation is delayed to the point in time when the data is uploaded to the central servers. However, given the resource constraints, our *key* idea is to store compressed GPS samples that would translate to *lower* offloading cost of energy over the low-power wireless radio compared to its uncompressed case. In the following section, we present a crisp overview of the GPS receiving operation in order to identify *unique* features that can facilitate efficient compression.

B. An Overview of GPS Receiving

The 32 GPS satellites continuously transmit CDMA coded navigational messages at a (low) data rate of 50 bps. This allows them to share the same carrier frequency of 1.575 GHz, and yet encode a different 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) that is transmitted at 1023 kbps. The C/A code, therefore, *repeats every millisecond* that results in 20 recurrences for each data bit sent.

A GPS receiver can calculate (or multilaterate [14]) its position by computing the travel time of the RF signals from each (visible) satellite to itself, and combining it with the respective satellites' trajectories at the time. Considering the high (300 m/μs) propagation speed of RF waveforms, the receiver's estimate of the time delay *must* be precise to the microsecond level. The respective algorithm derives the millisecond (NMS) and sub-millisecond (subMS) part of the

propagation time *differently*; wherein the NMS is decoded from the packet frame (every 6 s), while the subMS is calculated by *correlating* the C/A code in the acquisition phase.

GPS acquisition. It is, generally, the start-up mode in the post-processing chain[3] that aims to determine the set of visible satellites. The presence of any of the 32 satellites can be detected by identifying their unique 1023 bit C/A code; and is done by *cross-correlating* the C/A codes in the received GPS signal, typically shifted and scaled in time, with each known (C/A code) reference copy. Since the C/A codes are *orthogonal* to each other, any visible satellite will record a detectable spike in the correlation result. The motion of the satellites, however, introduces a *Doppler shift* in the received signal that needs to be corrected.

In order to minimize the detection anomaly and successfully decode the data from a given satellite, the acquisition algorithm performs a two dimension search on the received waveform (say, $\mathbf{x} \in \mathbb{R}^n$); wherein, for *each* satellite, the locally generated reference signal copy (say, $\mathbf{p} \in \mathbb{R}^n$) is cross-correlated with $\mathbf{x}$. $\mathbf{p}$ ensembles combinational values from two different sweeps:

- over all possible 1023 code shifts τ
- (minimum of) 41 equally spaced frequency bins ω_d of width 500 Hz within ± 10 kHz of the center frequency ω_c.

The sequence $\mathbf{r} \in \mathbb{R}^{\tau \times \omega_d}$ is the cross-correlation of $\mathbf{p}$ and $\mathbf{x}$, and is defined as:

$$\mathbf{r} = \sum_{n=0}^{n-1} p[n-\tau] e^{-j\omega_d n} \mathbf{x}[n] \tag{1}$$

$\mathbf{p}$ is separately modulated by carriers $e^{-j\omega_d n}$ and shifted in time. The center frequency of the respective bins in ω_d are given as: $[(\omega_c - \Delta\omega), ..., (\omega_c + \Delta\omega)]$ where, $\Delta\omega$ is referred to as the frequency bin width or search step. The maximum likelihood estimate of (τ, ω_d) (say, $\hat{\tau}$ and $\hat{\omega_d}$), which represent the subMS part of the propagation time and Doppler shift, is obtained by maximizing the function:

$$(\hat{\tau}, \hat{\omega_d}) = \arg\max |\mathbf{r}|^2 \tag{2}$$

Road-map. The operation of computing Eq. 2 is *expensive*, and demands high memory and energy resources. Considering the limited energy reserves on our target platform, it is desirable to scale down its complexity by a simpler process while still being capable of precisely estimating $(\hat{\tau}, \hat{\omega_d})$ during acquisition. This, coupled with the application-specific *requirements* of high delay tolerance and data offloading support, raises the scope for a new framework.

Fig. 1 shows the correlation outputs, as given by Eq. 2, for a collected GPS trace. In theory, only *one* dominant peak should be observed at the correct (code phase, frequency bin) combination; whereas, peaks of smaller magnitude may co-exist due to signal and noise interference. Fig. 1 exactly echoes this predication, where the most dominant coefficient (or correlation peak) is the *only* useful information, and is representative of the signal's time delay and the Doppler shift. Therefore, our idea is to exploit the underlying *information sparsity* in the signal model to design a simpler acquisition scheme that supports efficient compression, and later recovery. In the next section, we discuss the theory of sparse approximation that can exploit this sparse feature.

[3]If the existing position lock of satellites are within a second, the receiver can skip the acquisition process and directly start tracking.

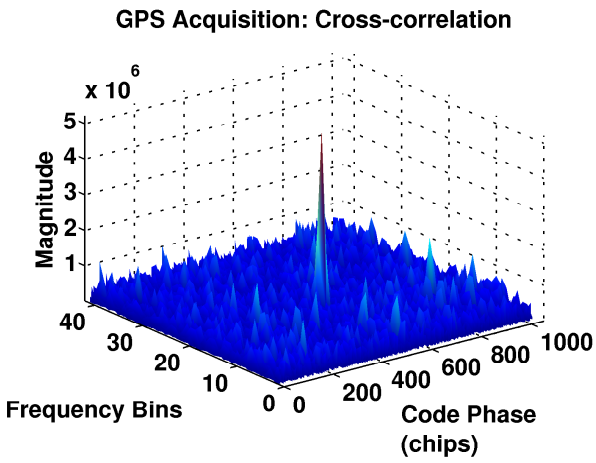


Fig. 1: **GPS acquisition by cross-correlation.** *The information content is sparse as the value of the code phase (i.e., time delay) and frequency bin (i.e., Doppler shift) corresponding to the correlation peak is only useful.*

C. An Overview of Sparse Approximation

Motivating insight. One can accurately and efficiently recover the information of a high dimensional signal (as $\mathbf{x}$) from only a small number of compressed measurements, when the signal-of-interest is sufficiently *sparse* in a certain transform domain [11], [15].

The sparsifying domain, referred to as a *dictionary* $\Psi \in \mathbb{R}^{n \times d}$, is a collection of parametrized waveforms that express $\mathbf{x}$ as a linear combination of a few significant elements. It is represented as:

$$\mathbf{x} = \Psi \mathbf{s} = \sum_{i=1}^{d} s_i \psi_i \tag{3}$$

where $\mathbf{s} \in \mathbb{R}^d$ is a coefficient vector of $\mathbf{x}$ in the Ψ domain, and ψ_i is a column of Ψ. If $\mathbf{s}$ is sparse, then it is possible to recover the position and value of its coefficients by a combinatorial problem; but is intractable. In pursuit of a polynomial time solution, Donoho [11] showed that, for a large system of equations, $\mathbf{s}$ can still be recovered by the following ℓ^1-minimization problem with high probability.

$$(\ell^1): \quad \hat{\mathbf{s}}_1 = \arg\min \|\mathbf{s}\|_1 \ \text{ subject to: } \ \mathbf{x} = \Psi \mathbf{s} \tag{4}$$

It has also been shown that dimensionality *reduction* by *random* linear projections preserves the ℓ^2 distance (i.e., all useful information) in the projection domain [16]. ℓ^1-minimization can still be used to recover the sparse $\mathbf{s}$ from the projected measurements with an overwhelming probability, even though, its dimension is significantly reduced. This operation can be achieved using a random sensing matrix $\Phi \in \mathbb{R}^{m \times n}$ as:

$$\mathbf{y} = \Phi \mathbf{x} = \Phi(\Psi \mathbf{s}) = (\Phi\Psi)\mathbf{s} \tag{5}$$

where $m \ll n$ and $\mathbf{y} \in \mathbb{R}^m$ is the measurement vector. However, for recovery, the columns of $(\Phi\Psi)$ should be as independent as possible so that the information regarding each coefficient of $\mathbf{s}$ is contributed by a different direction; and this is achievable if Φ and Ψ are *more* incoherent. Ensembles of

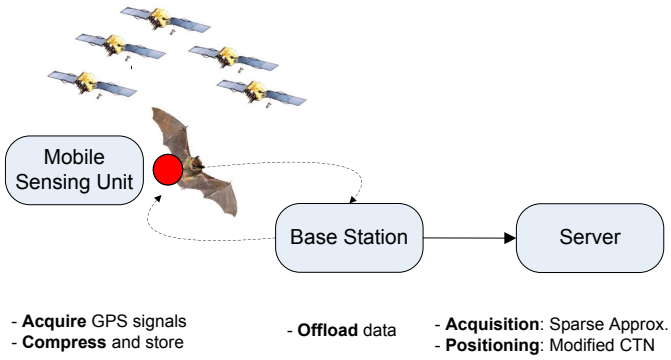


Fig. 2: **Architecture of S-GPS.**

random matrices sampled independently and identically (i.i.d.) from Gaussian and ± 1 Bernoulli distributions are *largely* incoherent with any fixed dictionary Ψ, and therefore, permit computationally tractable recovery of $\mathbf{s}$ [11], [15].

D. Details of S-GPS

The critical aspect for *casting* the GPS acquisition problem into the general framework of sparse approximation is to *design* a sparsifying representation dictionary. In this section, we explain the architecture of S-GPS (Fig. 2) that supports the essential design principles to efficiently compress the GPS signals on the target platform, offload to the base station, and facilitate GPS acquisition on the control server. The back-end operation to convert the acquisition measurements into position coordinates leverages the modified coarse-time navigation (CTN) technique [8], [9].

1) Design of Representation Dictionary:

Design guidelines. The general criteria for designing a reliable representation dictionary Ψ requires it to sufficiently sparsify the signal $\mathbf{x}$. However, the motion of transmitter (satellite) relative to the receiver introduces Doppler shifts that also need to be accounted during the GPS acquisition mechanism. This 2D search over the delay-Doppler binned-space introduces an important design criteria; where, Ψ should be able to preserve the propagation channel profile and frequency shift information while adhering to the basic design guidelines outlined by the underlying theory. We also improvise an additional criteria where Ψ should facilitate a faster recovery mechanism that implicitly derives the code phase and frequency bin results without reconstructing the original signal. Therefore, the design complexity is to identify and construct a befitting representation dictionary that satisfies *all* of the aforesaid requirements.

Design intuition. To this end, we were guided by Eq. (1) where the locally generated reference copy ensembles values from two different sweeps: a frequency sweep over all possible carrier frequencies of IF± 10 kHz in steps of 500 Hz (resulting in 41 frequency bins), and a code phase sweep over all 1023 different code phases. This suggests that the received signal $\mathbf{x}$ could be sparsely represented in the 2D delay-Doppler binned-space if we design a representation dictionary having column element that enumerate the C/A codes of the i^{th} (where $i \in \{1\text{-}32\}$) satellite for each possible (codephase, frequency bin) combination.

Design execution. For realizing this design goal, we adopt a circular matrix design of $\Psi_b \in \mathbb{R}^{n \times 1023}$ for each frequency bin b. The columns of Ψ_b correspond to the 1023 different versions of the C/A code $(c_1, c_2, ..., c_{1023})$ with different code phases multiplied with respective phase points of the center frequency of bin b. Ψ_b is of length of n corresponding to the length of a complete code (i.e., 1 ms sampled at a chosen frequency).

For each satellite i, the coarse acquisition phase provides an estimate of the (code phase) delay τ and the Doppler shift ω_d, which are integer multiples of the chip duration and frequency search step, respectively. One chip in the code is roughly $1\,\mu$s that converts to a delay measurement resolution of about 300 m. Therefore, for obtaining a finer code phase precision, an oversampling factor λ is introduced into the design of Ψ_b such that, for each frequency bin b, $\Psi_b \in \mathbb{R}^{n \times (\lambda \times 1023)}$.

Design types. Depending on the manner of solving for the code phase, frequency bin and satellite, the following *three* solution categories can be identified. We will evaluate the quality of three different dictionary designs in next section.

• *Multi-channel stacked.* The key element in a multi-channel design (Fig. 3(a)) is to provision for independent channels, each dedicated to a satellite, that can be processed in parallel. For each satellite i, the algorithm needs to iterate over b bins, each with its respective Ψ_b. This, in fact, bears resemblance to the native method of GPS acquisition.

• *Multi-channel flattened.* For each satellite i, the dictionary is given by $\Psi_i \in \mathbb{R}^{n \times (b \times \lambda \times 1023)}$ (Fig. 3(b)); thereby replacing b different instances of Ψ_b with a single flattened matrix Ψ_i.

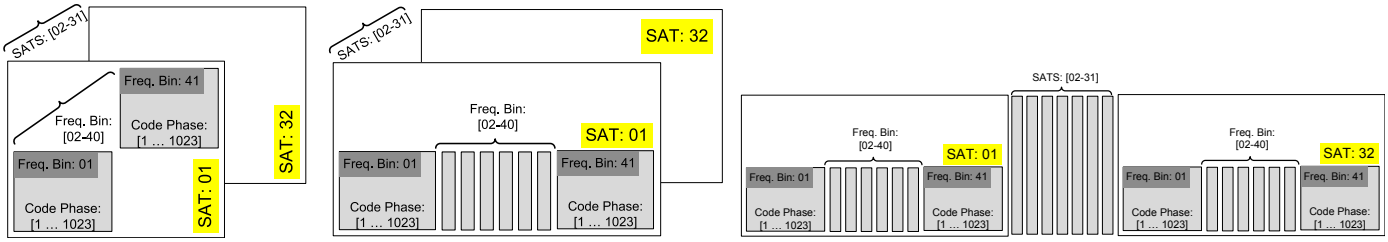
• *Single-channel flattened.* In contrast to the previous two designs, a single-channel design aims to process the information for all satellites at once (Fig. 3(c)). The dictionary is, therefore, given by $\Psi_a \in \mathbb{R}^{n \times (i \times b \times \lambda \times 1023)}$.

2) Compression, Recovery and Acquisition: We adhere to the same notation of $\mathbf{p} \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$ to represent the transmitted and the received GPS signal vectors corresponding to 1 ms, for each satellite i.

Compression. On the target platform, the dimensions of $\mathbf{x}$ are significantly reduced by multiplying it with a random sensing matrix $\Phi \in \mathbb{R}^{m \times n}$ resulting in the measurement vector $\mathbf{y} \in \mathbb{R}^m$ ($m \ll n$) as: $\mathbf{y} = \Phi\mathbf{x}$. m is related to n by the compression factor α given as: $m = \alpha\, n$, where $\alpha \in [0, 1]$. Φ is a binary sensing matrix with its entries i.i.d. sampled from a balanced symmetric Bernoulli distribution of ± 1. A balanced Φ consists of ± 1 at equal probability, where each row contains equal number of 1's and -1's. Therefore, in each row of Φ, the sum of the elements is always zero. A balanced Φ provides a higher probability of detection (at recovery) if the noise in $\mathbf{x}$ is Gaussian [17]. The m samples of $\mathbf{y}$ are transferred to the base station.

Recovery via sparse approximation. The base station uploads the compressed measurements to a service application on the control server. It requires the a-priori knowledge of the seed that generates Φ, and the dictionary Ψ. $\Psi \in \{\{b \times \Psi_b\}, \Psi_i, \Psi_a\}$ provides the representation basis where $\mathbf{x}$ can be sparsely represented by $\mathbf{s}$[4] as: $\mathbf{x} = \Psi\mathbf{s}$. However, due to noise (white Gaussian) $\mathbf{v} \in \mathbb{R}^n$ present in real data, $\mathbf{x}$ may not be exactly

[4]Note: depending on the choice of Ψ, $\mathbf{s}$ provides a different result. For example: for $\Psi \in \Psi_i$, $\mathbf{s}$ specifies the joint (delay, Doppler) result for the respective satellite i; while, for $\Psi \in \Psi_a$, $\mathbf{s}$ provides the same pair of estimates for all 32 satellites.



(a) Multi-channel stacked (b) Multi-channel flattened (c) Single-channel flattened

Fig. 3: **Representation dictionary Ψ designs.**

expressed as a sparse superposition of $\mathbf{s}$; and so, $\mathbf{x} = \Psi\mathbf{s} + \mathbf{v}$, where $\mathbf{v}$ is bounded by $\|\mathbf{v}\|_2 < \epsilon_0$. The sparse coefficient vector $\mathbf{s}$ is recovered by solving the following ℓ^1-minimization problem for a given tolerance ϵ via the second-order cone programming.:

$$(\ell_r^1): \ \hat{\mathbf{s}} = \arg\min \|\mathbf{s}\|_{\ell_1} \text{ s.t: } ||\Phi\Psi\mathbf{s} - \mathbf{y}||_2 \leq \epsilon \qquad (6)$$

(ℓ_r^1) is a stable and reduced version of ℓ^1-minimization. It is known as Lasso in the statistical literature, and regularizes *highly* undetermined linear systems when the desired solution is sparse. Depending on the choice of Ψ, (ℓ_r^1) needs to be solved individually for each satellite, or once for all satellites.

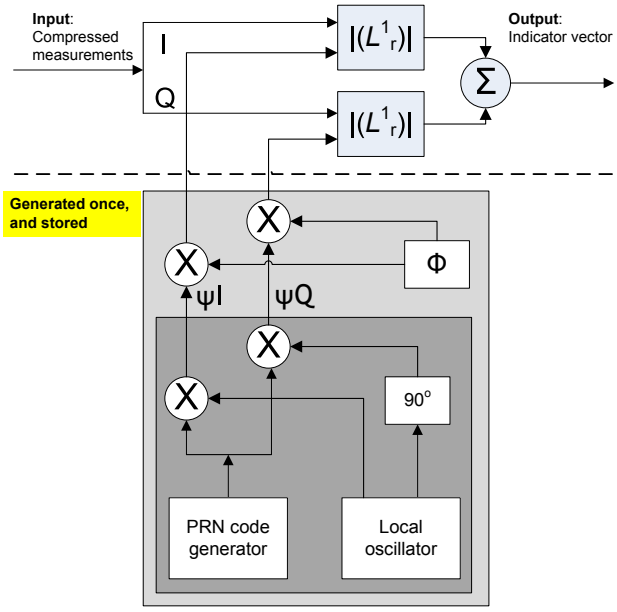


Fig. 4: **Search algorithm of S-GPS.**

Acquisition. As explained earlier, Ψ is obtained by the multiplication of the locally generated code sequences of the respective satellite(s) and the locally generated carrier signal. It is important to *emphasis* that the multiplication by a locally generated carrier signal produces the phase signal I, and multiplication with its 90^o phase-shifted version generates the quadrature signal Q. Therefore, Eq. 6 is solved with ΨI (i.e., in-phase part of Ψ) and ΨQ (i.e., quadrature-phase of Ψ) to obtain the respective solutions $\hat{\mathbf{s}}$I and $\hat{\mathbf{s}}$Q. Finally, the summation of the point-wise absolute values of $\hat{\mathbf{s}}$I and $\hat{\mathbf{s}}$Q provide an estimate of $\hat{\mathbf{s}}_1$ (Fig. 4).

As the C/A code is modulated onto the I part, the correct match should only be located in the I part of the signal. However, as the phase of the acquired signal is unknown, the I part of the signal generated at the satellite may *not* necessarily correspond to its recovered version. It is, therefore, important to search across both the I and Q components of Ψ. The location of the tallest peak in the coefficient vector $\hat{\mathbf{s}}_1$ (also referred to as the *indicator* vector: Fig. 4) provides an estimate of the propagation delay of the signal from the satellite(s) to the receiver, and its respective Doppler shift.

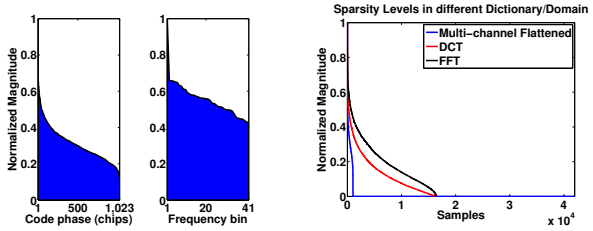
E. Analysis of S-GPS

In this section, we analyze different dictionary designs, and also, the challenges in correctly and reliably identifying the visible satellites. Since our requirement was to modify the manner of GPS acquisition, standard black-box methods that directly output the (code phase, frequency bin) information were not suitable.

Experimental platform. We conduct this study (and further evaluations) using a SiGe GN3S v3 USB RF front-end [18]. It consists of two integrated circuits: *first*, for RF amplification, filtering, down conversion, and baseband sampling; and *second*, for reading the digital samples (obtained from the first stage) and sending them in real-time to the PC through the USB. The sampling frequency, intermediate frequency, and capture data format of the front-end are user configurable. The captured GPS data is then post-processed on the PC using a software-defined implementation of S-GPS.

Data size. As the C/A code repeats every millisecond, 1 ms of data is enough for acquiring satellites. However, as the data packets are modulated by the C/A code at 50 bps, there is a possibility of a bit transition every 20 ms. If this bit transition occurs in a 1 ms signal sample, there is a high possibility of failure in acquiring the corresponding satellite details. Therefore, 2 ms chunk of data is more reliable for satellite acquisition, and is *widely* adopted in practice [4], [8]–[10], [14].

1) Representation Dictionary: The multi-channel stacked matrix design $\Psi \in \{b \times \Psi_b\}$ breaks down the joint (delay, Doppler) estimation problem, for each satellite, into b sub-problems (of smaller dimensions) that need to be solved independently. The b solutions are, subsequently, accumulated to identify the position of the strongest coefficient. Such an



(a) Multi-channel stacked (b) Multi-channel flattened vs. {DCT & FFT}

Fig. 5: **Comparison of sparsity levels.** *The signal-of-interest has two order of magnitude more sparse representation in the multi-channel flattened dictionary than the DCT and FFT domains. Its combined (delay, Doppler) sparsity is also significantly better than their respective individual levels provided by the multi-channel stacked dictionary. The figure has been scaled down by a factor of 16 for clearer depiction.*

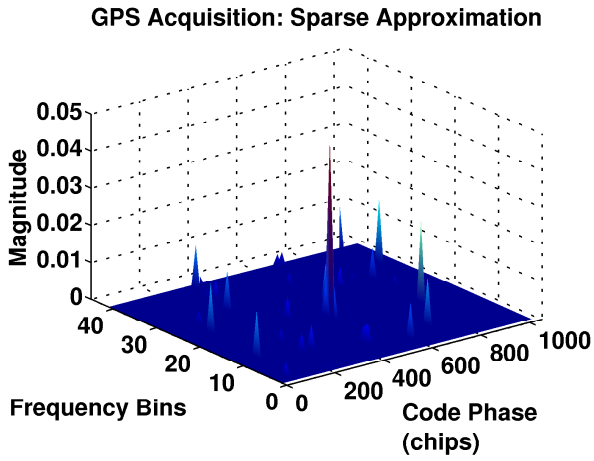


Fig. 6: **GPS acquisition by sparse approximation.** *The quality of acquisition is the same as in Fig. 1 for the same data set.*

approach results in b locally optimal solutions that may not necessarily be the best (global) sparse representation of the (delay, Doppler) combination for the respective satellite. This can be justified by Fig. 5(a) where the b stacked dictionaries are sparse in the representing the code phase delay, but not the frequency bin.

The multi-channel flattened matrix design $\Psi \in \Psi_i$ is a combined representation of the (delay, Doppler) sparsity for each satellite, and facilitates their joint recovery by Eq. 6. The single-channel flattened matrix design $\Psi \in \Psi_a$ enhances the idea of jointly recovering all sparse information. It combines the (delay, Doppler) sparsity of all available GPS satellites in one compact representation that can be solved at once by Eq. 6. Although such as design is valid, our results showed that the ℓ_1-minimizer gets biased towards the satellite with the strongest signal; and hence, under performs in detecting remaining satellites with weaker signal levels. In terms of memory resources, Ψ_a needs 32 times more space than Ψ_i; and can easily scale into ten of gigabytes or terabytes depending on the choice of the oversampling factor λ. In this regard, Ψ_i strikes a good tradeoff between sparsity and space complexity. Therefore, we adopt the *multi-channel flattened* Ψ_i design as the representation dictionary[5].

We evaluate the quality of the new dictionary, inherently based on the foundations of a circular matrix design; and also, compare it against the popular DCT and FFT domains. A circular matrix is a special kind of Toeplitz matrix where each row vector is rotated one element to the right (or left) relative to the preceding row vector. In this regard, the design of Ψ for each Ψ_b bears similarity to the Toeplitz matrix dictionary proposed by Misra et al. [12]. However, their design did not consider the effect of motion and Doppler errors; and hence, sparsely represents $\mathbf{x}$ only in the delay space but not in the Doppler space. Fig. 5 compares the sparsity levels of $\mathbf{x}$ in the different representation dictionaries. The respective coefficients are sorted by their magnitude that decay like the power law; i.e., the d^{th} largest entry of the reordered $\mathbf{s}$ obeys $|s|_{(d)} \leq Const \cdot d^{-r}$ for $r \geq 1$. The fastest decay characteristic is observed in the multi-channel flattened dictionary. It offers the most sparse representation; and hence, will facilitate the most accurate approximation of the sparse solution with the smallest number of measurements. Here, it is important to note that sparsity is a *direct* outcome of the design of the dictionary.

2) Acquisition Challenges and Mitigation:

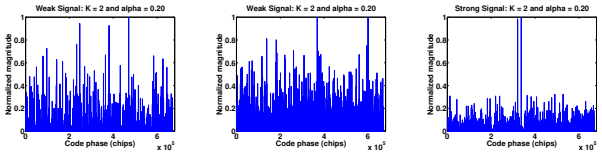
Basic performance test. Using 20 sets of raw GPS data taken from four different locations and time of the day, we analyze the satellite acquisition performance of S-GPS against SoftGNSS [14], a commercially available MATLAB supported software-defined GPS package. The acquired signals[6] were compressed at different configurations of α; ranging between 0.10-0.30 that translate to 90%-70% compression, respectively.

For illustration purpose, the acquisition quality of S-GPS with 70% (i.e., $\alpha = 0.30$) compressed measurement is shown in Fig. 6; with the same data set that was used for obtaining the cross-correlation result shown in Fig. 1. We observe that both the methods yield the same result (code phase = 515 chips and frequency bin index = 19) for the position of the tallest spike. Although measurement noise leads to additional correlation peaks of smaller magnitudes, they are not important parameters for distance estimation.

A satellite was assumed to be acquired if the *metric* P, obtained by dividing the maximum/peak coefficient in $\hat{\mathbf{s}}_1$ by the noise floor, exceeds a certain threshold T. By empirical tests, we found that T greater than 5 was a reliable threshold. On an average, S-GPS was able to correctly view 2-3 satellites in each 2 ms chunk of every data segment, while the same was 2.5 times more for SoftGNSS. The tally of the detected satellites was, therefore, *considerably* lower than the minimum count required for obtaining a GPS position fix.

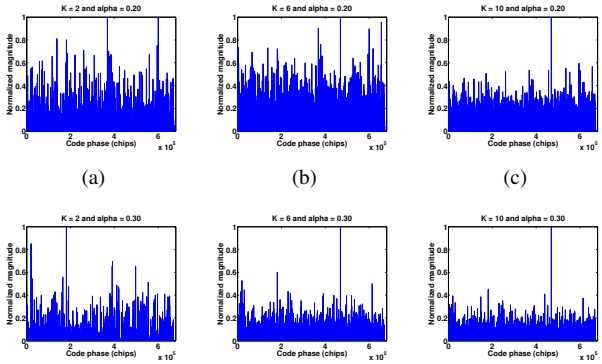
[5]The I and Q components of Ψ_i are generated *once* for all satellites, and stored; rather than repeated re-generations at runtime (Fig. 4).

[6]SiGe GN3S v2 front-end configuration: sampling frequency: 8.1838 MHz, intermediate frequency: 38.4 KHz, data format: 1bit real in short char binary



(a) (b) (c)

Fig. 7: **S-GPS acquisition challenges with 2 ms of data.** (a): *Low peak sharpness (i.e., ratio of the first to the second tallest peak), and high recovery noise* (b): *Incorrect recovery* (c): *Low peak sharpness, but low recovery noise. The correct peak is at chip location* (a)&(b): *4.8e+5* (c): *3.5e+5*. *Note: K = # data chunks of length 1 ms*; α = *compression factor*.



(a) (b) (c)

(d) (e) (f)

Fig. 8: **Improving acquisition reliability.** *Consolidating the coefficient vector points over K operations improves the magnitude of the correct peak coefficient with respect to the noise floor. For all the cases, the correct peak is at chip location: 4.8e+5.*

We investigate the reasons for the unsatisfactory performance of S-GPS. We understand that GPS signal strengths are, generally, very weak; and are in the range of 18-25 dB below the noise level. While the signal is efficiently captured by a limited number of random measurements and is recovered by ℓ^1-min in the best possible manner, the low SNR levels (especially below 20 dB) have a significant impact on the sparse approximation process. The higher measurement noise contributes to either low peak sharpness (Fig. 7(a)) or inaccurate recovery (Fig. 7(b)), which eventually lead to failed or erroneous detection. Even for relatively stronger signals (that are slightly above 20 dB), the metric P may clearly surpass the detection threshold; but still leaves scope for further improvement to precisely recover the correct peak from the vicinity of near similar magnitude peaks (Fig. 7(c)). Although Fig. 7 shows the observation from α=0.20, similar outcomes were also logged for α=0.10 and α=0.30.

Improving acquisition reliability. A simple solution to overcome the satellite detection anomalies is to boost the magnitude of the correct peak coefficient, which is an *approximate* indication of the SNR, with respect to the noise floor. We, therefore, append a SNR improvement stage to the S-GPS framework. We calculate the consolidated point by the accumulation of K coefficient vectors (where each K corresponds to the recovery result of 1 ms of GPS data), and summing their intensities.

$$\mathbf{s}_c = \sum_{k=1}^{K} (\hat{\mathbf{s}}_1)_k \tag{7}$$

Satellite i is said to be detected if the metric P_c, obtained by dividing the maximum coefficient in $\mathbf{s}_c$ by the noise floor, exceeds the threshold T. This operation is performed for all i satellites.

As mentioned before, the navigational data is transmitted at a rate of 50 bps; and therefore, will result in a possible bit transition every 20 ms. This method also provides robustness to the negative effect of transitions in the data bits; as it considers the accumulated and summed contributions over a longer time-window rather than analyzing individual 1 ms data segments.

Fig. 8 demonstrates the benefits of this mechanism. For α = 0.20, K = 2 does not correctly recover the index of the peak coefficient (Fig. 8(a)); but as K is increased to 6 by combining the contribution of six 1 ms chunks, the estimation becomes more accurate (Fig. 8(b)). The problem of the high noise floor, which is still evident at this stage, is significantly reduced as K reaches 10 (Fig. 8(c)). The same observations are also valid for a higher α of 0.30 and K = 2 (Fig. 8(d)). However, estimation reliability is attained at K = 6 (Fig. 8(e)) without any significant improvements with additional contributions (Fig. 8(f)). It is *important* to note that K is also *indicative of the corresponding data length to be processed*. For example, K = 6 is the consolidated result obtained by processing 6 data chunks of 1 ms length; or a combined data length of 6 ms.

III. Evaluation

In this section, we evaluate the quality and limitations of the S-GPS framework.

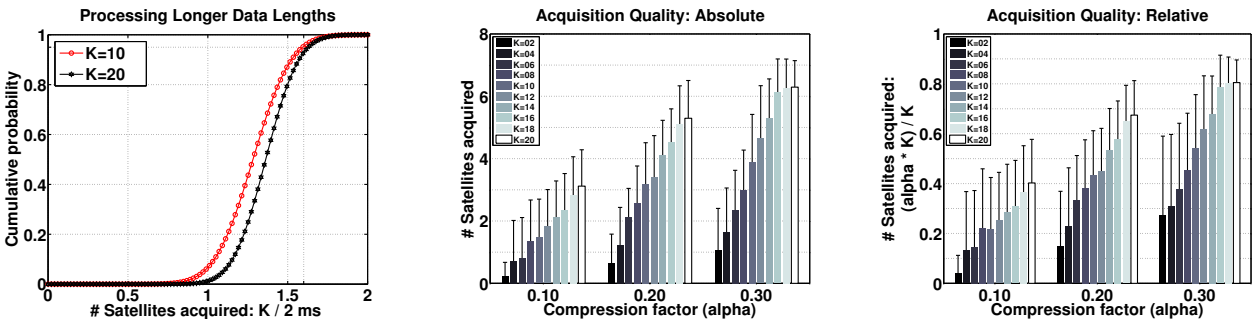
Setup. This study uses 20 sets of raw GPS data, collected using the experimentation platform described in Section II-E. It offered the flexibility of varying both the GPS signal (between 10-60 s) and sample length with configurable parameters. Also, as mentioned before, the ground-truth was obtained with SoftGNSS.

When considering the presented results, it is important to note the categorical difference of S-GPS from the state-of-the-art acquisition method; where our aim is to attain similar performance, but with significantly fewer observations. While such a requirement may not arise in regular applications, it is a necessity in our motivating application and other related ones.

S-GPS aims to achieve the best possible energy efficiency in data offloading and GPS sensing. In this regard, data size is a *key* parameter that determines a good tradeoff between accuracy and energy.

A. Acquisition and Location Performance

Acquisition quality. The purpose of acquisition is to identify all visible satellites (along with their respective code phase delay and frequency shifts); since, acquiring more satellites improves the overall location accuracy. Therefore, as a functional goal, any GPS acquisition algorithm must maximize the probability of successful satellite acquisitions.



(a) (b) (c)

Fig. 9: **Acquisition quality.** (a): **Benefits of processing longer segments.** *There is a high probability of acquiring an additional 1.5× satellites by processing 10-20 ms over 2 ms.* (b)&(c): **Acquisition quality of S-GPS**, *for different data length {10-20} ms, at compression factor of {0.10,0.20,0.30}. Note: K = 20 is the consolidated result obtained by processing 20 data chunks of 1 ms length; and therefore, K is also an indicator of the data length.*

In this regard, an important question is whether processing *longer* data lengths (such as 10/20 ms) promises better acquisition quality than the accepted practice of using 2 ms. The result for this study, using the state-of-the-art cross-correlation method, is shown in Fig. 9(a); and it suggests that there is a 95% probability of detecting 1.5× more satellites by processing longer data segments. We also observe the same results with S-GPS as depicted in Fig. 9(b) and Fig. 9(c).

While such a mechanism may not be a good alternative for on-board acquisitions (as they are more energy consuming due to greater computational demand and processing time), it can be *more* viable if the computation load can be transferred to the control server. Also, in applications such as tracking flying foxes, there may not be any previous knowledge about the visit locations or their approximate perimeters. In such a scenario, it is difficult to predict the acquisition quality. Therefore, considering the limited reserves of energy, it may be a good choice to safeguard against failed acquisitions by collecting longer data chunks.

With this background, we investigate a reliable data size measure that optimizes energy and acquisition quality of S-GPS. Fig. 9(b) & Fig. 9(c) characterizes the tradeoff between compression factor (α) and accumulated contribution from K (1 ms) chunks against the number of satellites acquired. The optimal choice of α and K is important to ascertain the best acquisition performance that can be achieved with the least measurements m (refer to Section II-D2), where a smaller m leads to lower storage and transmission cost.

Fig. 9(b) shows the absolute satellite count with different α and K; and suggests that their lower values result in deteriorated performance. $\alpha = 0.10$ (for any K) is, typically, not useful as it does not meet the minimum tally of (at least 5) satellites for obtaining a position fix with the modified CTN method; while $\alpha = 0.20$ for (K = 12-20) and $\alpha = 0.30$ for (K = 10-20) meet this criteria. However, neither of these parameter choices are able to mirror the ground truth result of 8 satellites. Fig. 9(c) shows the ratio of the number of detected satellites using compressed data of length ($\alpha\times$ K) ms to its respective uncompressed K ms case. A ratio of 1 implies that the performance of the compressed case is similar to its uncompressed counterpart. However, the results suggest that the best outcome is only 0.8; and is obtained at $\alpha = 0.20$ for (K = 18-20) $\alpha = 0.30$ for (K = 12-20). This means that S-GPS detects 1-2 fewer satellite(s) than its ground truth.

The above studies, however, do not make a fair comparison based on the *absolute* data length. For example, data compressed with $\alpha = 0.20$ and K = 20 and $\alpha = 0.30$ and K = 20 have an absolute length of 4 ms and 6 ms, respectively. Therefore, they would not derive the same performance indicators as their uncompressed 20 ms segment. Fig. 10(a) and Fig. 10(b) make the direct comparison, and express it as a ratio of [(0.20× K)/4 ms] and [(0.20× K)/6 ms], respectively. Both the results suggest that there is a 95% probability of detecting 1.2× more satellites by recording 20 ms and compressing it down to 4 ms/6 ms; rather than simply processing their uncompressed version of equivalenty data length. This improvement is a result of information embedding in random ensembles; which preserves the energy of its respective higher dimension (i.e., 20 ms) representation, although the absolute data length is significantly reduced by 70-80%.

Alternately, for acquiring the same number of satellites as their equivalent uncompressed data lengths, there is a 95% probability of success by using as low as K = 12 for $\alpha = 0.20$ and K = 10 for $\alpha = 0.30$. These parameters translate to a 40-50% savings on the required data size for mirroring the acquisition quality as the ground truth. Fig. 10(e) shows the optimal configuration of α and K, as suggested by Fig. 10(a) and Fig. 10(b), for obtaining a similar acquisition quality as their uncompressed (base) cases of 4 ms and 6 ms.

Location accuracy. Fig. 10(c) and Fig. 10(d) shows a box-plot of the overall location accuracy for α = 0.20 and α = 0.30. For the respective cases, the median error (depicted as a blue 'o') is less than 70 m and 40 m. Although the median error is quite consistent across all cases, the interquartile range shows some variations due to the non-uniform tally of acquired satellites in each configuration and data points. Fig. 10(f) groups the observations according to the visible satellites count (that vary from 5-7), and shows the corresponding error in location. As S-GPS also uses the modified CTN technique for positioning, a minimum of 5 satellites are required. The accuracy improves

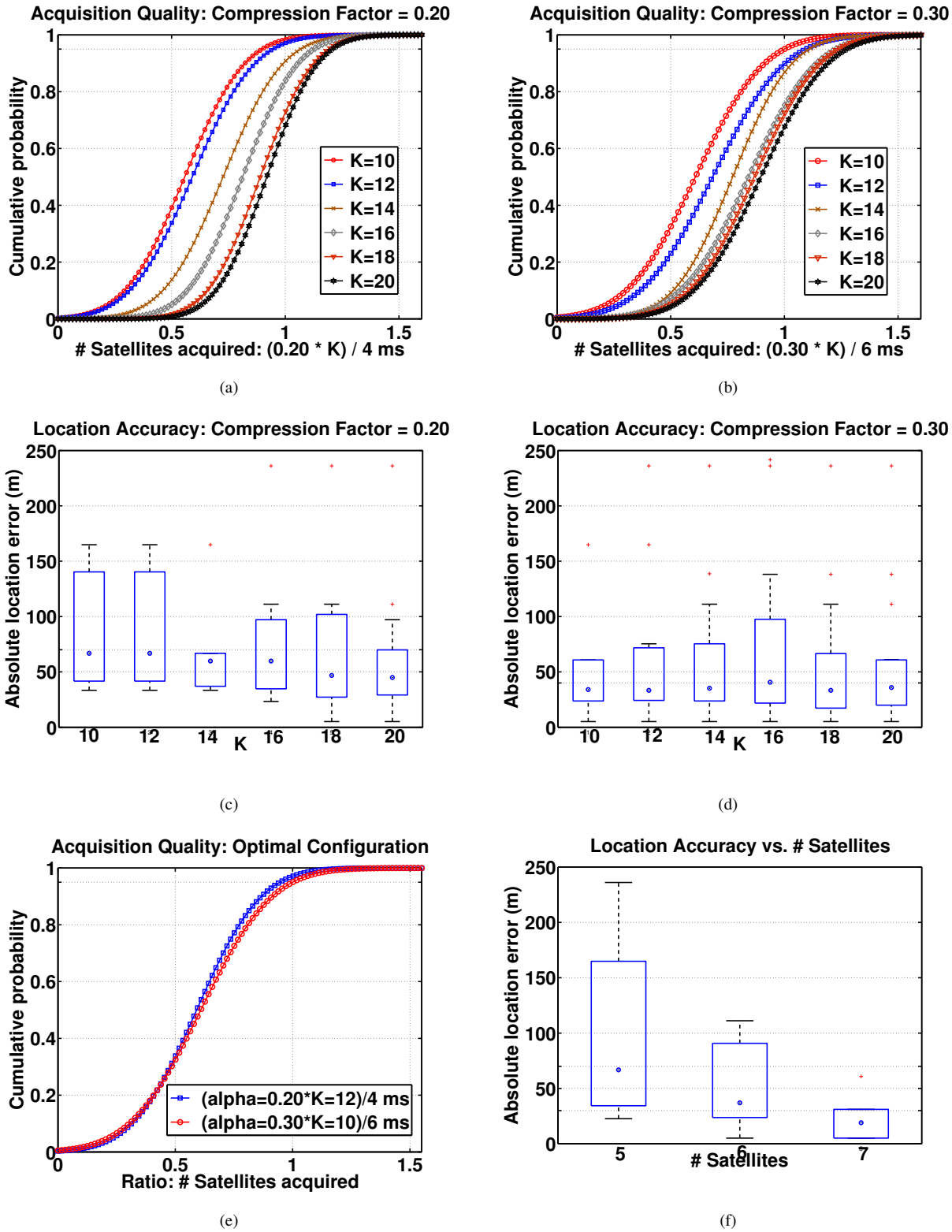


Fig. 10: **Acquisition quality and location accuracy over an absolute data length of** 2-4 **ms &** 3-6 **ms.**

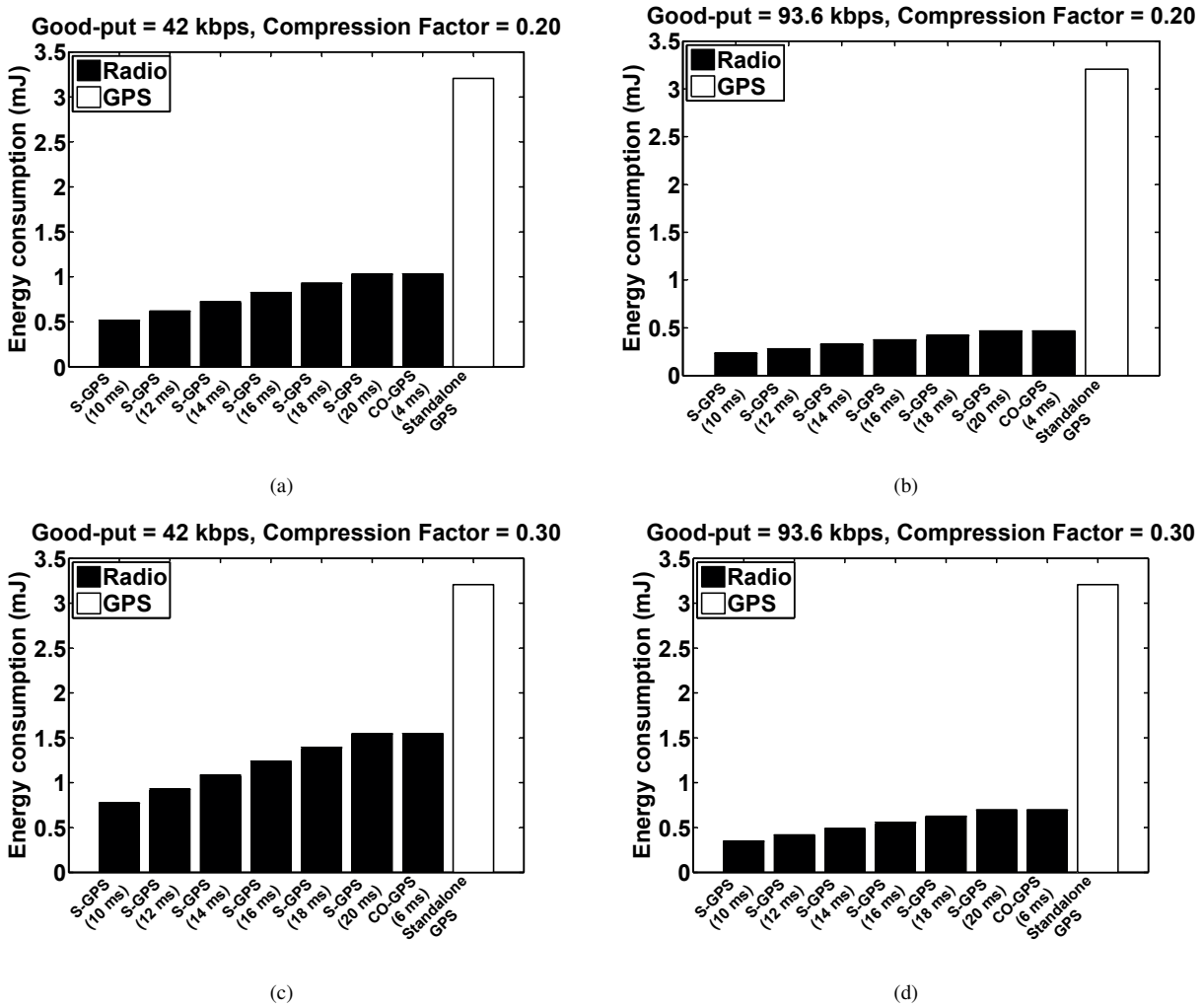


Fig. 11: **Energy consumption.** *S-GPS is twice as energy efficient than CO-GPS, and has 5-10 times lower energy costs than a standalone GPS. S-GPS and CO-GPS consume a few μJ due to their very small sensing cycles; and so, the radio is the major contributor to the energy cost for cloud-offloading solutions.*

from a median error of 70 m to 30 m with the availability of more satellites. The location accuracy of S-GPS is similar to LEAP and CO-GPS ($<$ 35-40 m), and also duly satisfies our application's requirement.

B. Energy consumption

We provide an estimate of the GPS acquisition cost, in terms of energy budget, on the target platform for obtaining a single position fix.

Components evaluated. In our study, we used the approach of capturing digitized GPS samples; followed by their compression by random ensembles with its entries i.i.d. sampled from a balanced symmetric ($\pm$ 1) Bernoulli distribution. However, as part of a efficient GPS receiver system, this *entire* step can be substituted by a compressed sensing analog front-end (referred to as: CS-ADC). A CS-ADC can sample sparse signals at sub-Nyquist rates, and deliver compressed measurements *directly*. Many existing hardware prototypes of a CS-ADC have shown that its power consumption is *less* than a traditional analog-to-digital converter [19], [20] as it does not have to sample at the full (Nyquist) rate. Considering these empirical facts, we only evaluate the energy costs of the *two* most power intensive modules on the platform: (i) u-blox MAX-6 GPS [13] and (ii) IEEE 802.15.4 complaint radio transceiver that, respectively, consume 74 mW and 99 mW in full operation mode. Previous work by Sommer et al. [21] and Afanasyev et al. [22] have reported that the data goodput of IEEE 802.15.4 complaint transceivers are between 42 kbps and 93.6 kbps; while the GPS (coldstart and hotstart) time-to-first-fix is 26 s.

Taking all of the above empirical observations into account, we show the corresponding energy consumption in Fig. 11. S-GPS is able to achieve a competing level of acquisition performance at K = 12 for α = 0.20 and K = 10 for α = 0.30 over its respective uncompressed case of sensing 4 ms and 6 ms of raw data. S-GPS is 1.8 times (at K = 12 for α = 0.20) and 2 times (at K = 10 for α) more energy efficient than CO-GPS, which is based on a simple sense and offload paradigm. In comparison to a standalone GPS module, which performs all operations on the platform itself without optimized sensing and offload cycles; S-GPS saves 5-10 times more energy.

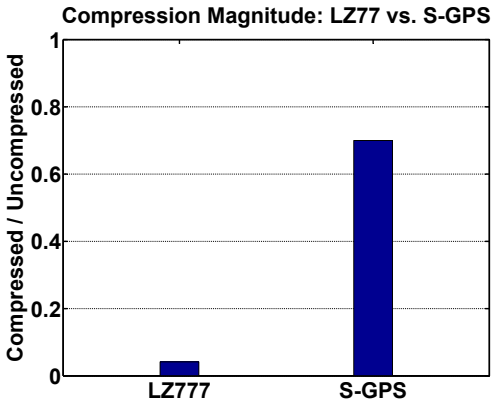


Fig. 12: **Compressibility performance.** *LZ77 vs. S-GPS.*

Road-map. Based on the results of acquisition quality, location accuracy and energy cost, a compression factor of 0.30 and K=10 (i.e., a combined data length of $10\,\text{ms}$) is a reliable configuration for S-GPS.

C. Compression Performance

Fig. 12 compares the compression quality of S-GPS against the popular LZ77-based algorithm 'gzip'. The result shows that raw GPS signals are not much compressible ($< 10\%$) using traditional methods such as LZ77 (since they appear like random noise by design); while the same signals can be compressed by 70% with S-GPS. Besides, the LZ77 lossless nature of compression makes it nonrobust to information loss (packet drops) during data transmission that are common in low-power sensor networks. On the other hand, the S-GPS would offer a graceful performance degradation under similar circumstances as it can still recover the results, but with larger errors; and has the same performance as compressing with a smaller α.

IV. Related Work

Location sensing, by itself, is a rich area of research; and performing it with GPS is an active topic of investigation. S-GPS draws on prior work in multiple areas; *mainly* public infrastructure based outdoor mobile sensing, signal synchronization, cloud offloading based GPS, and sparse approximation.

The use of public infrastructure (such as GPS [23], A-GPS [24], WiFi access points [25], FM radio stations [26], Cellular towers, etc.,) is an appealing direction as it relaxes the need to deploy systems (such as wireless sensor nodes [27] and other embedded platforms [28]) for localization assistance, and can cater to a wider range of outdoor (real-time and delay-tolerant [7]) applications. The mobile receivers simply needs to synchronize their operation with the respective infrastructure system counterpart. Although the problem of signal synchronization applies to many of the above wireless technologies, synchronizing with GPS satellites is processing-intensive and energy consuming; due to its nature of operation (i.e., two-dimension search using cross-correlation) and very low ($-20\,\text{dB}$) SNR [4]. WiFi and Cellular systems operate at much higher levels of SNR than GPS; and hence, the receivers can synchronize with relatively low overhead by merely detecting an increase in received power [29].

Several techniques have been proposed to address the problem of energy consumption in GPS receiving. Many of these solutions are designed with a focus on real-time applications, and also consider GPS as a potential black-box. They tradeoff energy expense by adaptively using it with other sensors [3], turning it on when significant motion is detected [2], or by combining location requests from multiple applications with a single GPS reading [30]. Although these solutions can also be used for delay-tolerant applications, they do not improvise on the time-flexible nature to optimize the de facto GPS post-processing chain and/or aggressively duty-cycle.

In pursuit of a mechanism to reduce the active GPS time, LEAP [8] and CO-GPS [9] solutions combine these benefits for applications of delay-tolerant nature. LEAP performs GPS acquisition on the receiver locally, and transfers the resulting information to the cloud. The location is computed on the cloud server with the CTN technique, wherein the preliminary acquisition results are combined with a known location of a nearby object (e.g., the cell tower in case of a phone). As this solution cannot be used for embedded platforms without cellular connection (for providing landmarks) and limited computation resources (for performing acquisition); CO-GPS transfers the required raw signal samples for acquisition to the cloud, and uses its vast computing resources to guess candidate landmarks and maintain the ephemeris database for positioning. S-GPS adopts the sample-and-process approach of CO-GPS, but reduces the energy cost of data offloading.

S-GPS, based on the theory of sparse approximation, takes advantage of information sparsity in the GPS acquisition process. The information is efficiently embedded without much loss (which serves the purpose of storage and transmission), and is subsequently recovered from an underdetermined system by ℓ_1-minimization. Sparsity aware solutions offer an efficient sampling strategy; and so, are an active field of applied research in resource constrained sensor networks [12], [31]. Although S-GPS falls into the category of sparse approximation based range-finders as previously investigated by Misra et al. [12], the scope of our problem is completely different. We design a new dictionary and related suite of signal processing techniques for GPS acquisition that estimate the propagation delay from the satellites to the receiver; while compensating for Doppler effect and very low SNR levels. QuickSync [10] belongs to the class of real-time solutions, which aim to reduce the GPS synchronization time on the receiver itself by sparse Fourier transform. Therefore, there are fundamental differences in theory and application scope; although, the idea of sparsity is central to both S-GPS and QuickSync.

V. Conclusion

For delay-tolerant applications, offloading GPS processing to the cloud is possible. S-GPS is a GPS sensing approach that is aimed to limit the associated energy costs in this transfer operation. The sparse representation based GPS acquisition technique can efficiently capture and embed information in a lower dimensional space (by random ensembles); and subsequently, recover it from an underdetermined system. Such an approach has several merits. It provides a simple dimensionality reduction mechanism to condense the dataset. As the data compressibility is proportional to its information level,

sparse (information) signals can be compressed significantly. It requires transferring and processing a significantly smaller datasets to obtain accuracies comparable to the state-of-the-art detection technique. At the local device end, the simplicity of this operation translates into appreciable energy savings. By empirical evaluations, we showed that S-GPS is 2 times more energy efficient than offloading uncompressed data, and has 5-10 times lower energy costs than a standalone GPS; with a median positioning accuracy of 40 m.

S-GPS would further benefit by mechanisms to enhance the received SNR; which was, by far, the most daunting task in realizing this solution. The initial success of S-GPS has motivated us to explore efficient hardware designs for compression on the target platform. In addition, we are also focusing on techniques to optimize the energy expenditure and execution time on the server side.

VI. Acknowledgement

This work was performed by P. Misra during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Programme. The research leading to these results have received funding from the *European Union* Seventh Framework Programme(FP7/2007-2013) under *grant agreement* no. 246016. We would like to thank the anonymous reviewers, and our shepherd Dr. Niki Trigoni (University of Oxford) for their helpful comments.

References

[1] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *MobiSys '10*. New York, NY, USA: ACM, 2010, pp. 285–298. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814462

[2] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *MobiSys '10*. New York, NY, USA: ACM, 2010, pp. 299–314. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814463

[3] R. Jurdak, P. Corke, D. Dharman, and G. Salagnac, "Adaptive GPS duty cycling and radio ranging for energy-efficient localization," in *SenSys '10*. New York, NY, USA: ACM, 2010, pp. 57–70. [Online]. Available: http://doi.acm.org/10.1145/1869983.1869990

[4] D. Plausinaitis, "GPS receiver technology mm8," http://kom.aau.dk.

[5] T. Wark, C. Crossman, W. Hu, Y. Guo, P. Valencia, P. Sikka, P. Corke, C. Lee, J. Henshall, K. Prayaga, J. O'Grady, M. Reed, and A. Fisher, "The design and evaluation of a mobile sensor/actuator network for autonomous animal control," in *IPSN '07*. New York, NY, USA: ACM, 2007, pp. 206–215. [Online]. Available: http://doi.acm.org/10.1145/1236360.1236389

[6] D. Anthony, W. P. Bennett, M. C. Vuran, M. B. Dwyer, S. Elbaum, A. Lacy, M. Engels, and W. Wehtje, "Sensing through the continent: towards monitoring migratory birds using cellular sensor networks," in *IPSN '12*. New York, NY, USA: ACM, 2012, pp. 329–340. [Online]. Available: http://doi.acm.org/10.1145/2185677.2185747

[7] R. Jurdak, P. Sommer, B. Kusy, N. Kottege, C. Crossman, A. Mckeown, and D. Westcott, "Camazotz: multimodal activity-based GPS sampling," in *IPSN '13*. New York, NY, USA: ACM, 2013, pp. 67–78. [Online]. Available: http://doi.acm.org/10.1145/2461381.2461393

[8] H. S. Ramos, T. Zhang, J. Liu, N. B. Priyantha, and A. Kansal, "Leap: a low energy assisted GPS for trajectory-based services," in *UbiComp '11*. New York, NY, USA: ACM, 2011, pp. 335–344. [Online]. Available: http://doi.acm.org/10.1145/2030112.2030158

[9] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. F. Loureiro, and Q. Wang, "Energy efficient GPS sensing with cloud offloading," in *SenSys '12*. New York, NY, USA: ACM, 2012, pp. 85–98. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426666

[10] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk, "Faster GPS via the sparse fourier transform," in *Mobicom '12*. New York, NY, USA: ACM, 2012, pp. 353–364. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348587

[11] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006. [Online]. Available: http://dx.doi.org/10.1002/cpa.20132

[12] P. Misra, W. Hu, M. Yang, and S. Jha, "Efficient cross-correlation via sparse representation in sensor networks," in *IPSN '12*. New York, NY, USA: ACM, 2012, pp. 13–24. [Online]. Available: http://doi.acm.org/10.1145/2185677.2185680

[13] "MAX-6 series: compact u-blox 6 GPS modules," http://www.u-blox.com/en/gps-modules/pvt-modules/max-6.html.

[14] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. Jensen, *A software-defined GPS and Galileo receiver: A single-frequency approach.* Birkhauser, 2006.

[15] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transaction on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[16] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, pp. 253–263, 2008.

[17] Y. Shen, W. Hu, J. Liu, M. Yang, B. Wei, and C. Chou, "Efficient background subtraction for real-time tracking in embedded camera networks," in *SenSys '12*. New York, NY, USA: ACM, 2012, pp. 295–308. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426686

[18] "SiGe GN3S Sampler v3," https://www.sparkfun.com/products/10981.

[19] Z. Charbiwala, P. Martin, and M. Srivastava, "CapMux: a scalable analog front end for low power compressed sensing," in *IGCC '12*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1109/IGCC.2012.6322255

[20] F. Chen, F. Lim, O. Abari, A. Chandrakasan, and V. Stojanovic, "Energy-aware design of compressed sensing systems for wireless sensors under performance and reliability constraints," *IEEE Transactions on Circuits and Systems I*, vol. 60, no. 3, pp. 650–661, 2013.

[21] P. Sommer, B. Kusy, A. McKeown, and R. Jurdak, "The big night out: Experiences from tracking flying foxes with delay-tolerant wireless networking," in *REALWSN '13*, 2013.

[22] M. Afanasyev, D. OŔourke, B. Kusy, and W. Hu, "Heterogeneous traffic performance comparison for 6lowpan enabled low-power transceivers," in *HotEmNets '10*. New York, NY, USA: ACM, 2010, pp. 10:1–10:5. [Online]. Available: http://doi.acm.org/10.1145/1978642.1978655

[23] B. Hoffmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, 4th ed. NY: Springer-Verlag, 1997.

[24] J. LaMance, J. DeSalas, and J. Jarvinen, *Assisted GPS: A Low-Infrastructure Approach*. GPS World, 2002.

[25] "Skyhook," http://www.skyhookwireless.com.

[26] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *MobiSys '12*. New York, NY, USA: ACM, 2012, pp. 169–182. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307653

[27] B. Kusý, J. Sallai, G. Balogh, A. Ledeczi, V. Protopopescu, J. Tolliver, F. DeNap, and M. Parang, "Radio interferometric tracking of mobile wireless nodes," in *MobiSys '07*. New York, NY, USA: ACM, 2007, pp. 139–151. [Online]. Available: http://doi.acm.org/10.1145/1247660.1247678

[28] L. Girod, M. Lukac, V. Trifa, and D. Estrin, "The design and implementation of a self-calibrating distributed acoustic sensing platform," in *SenSys '06*. New York, NY, USA: ACM, 2006, pp. 71–84. [Online]. Available: http://doi.acm.org/10.1145/1182807.1182815

[29] M. Karim and M. Sarraf, *W-CDMA and CDMA2000 for 3G Mobile Networks*. McGraw-Hill, 2002.

[30] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *MobiSys '10*. New York, NY, USA: ACM, 2010, pp. 315–330. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814464

[31] X. Wu and M. Liu, "In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing," in *IPSN '12*. New York, NY, USA: ACM, 2012, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/2185677.2185679