Using Crowdsourced Satellite SNR Measurements for 3D Mapping and Real-time GNSS Positioning Improvement

Andrew T. Irish
Dept. of Electrical and Computer Engineering
UC Santa Barbara
Santa Barbara, CA 93106
andrewirish@ece.ucsb.edu

Daniel Iland
Dept. of Computer Science
UC Santa Barbara
Santa Barbara, CA 93106
iland@cs.ucsb.edu

Jason T. Isaacs
Dept. of Electrical and Computer Engineering
UC Santa Barbara
Santa Barbara, CA 93106
jtisaacs@ece.ucsb.edu

João P. Hespanha
Dept. of Electrical and Computer Engineering
UC Santa Barbara
Santa Barbara, CA 93106
hespanha@ece.ucsb.edu

Elizabeth M. Belding
Dept. of Computer Science
UC Santa Barbara
Santa Barbara, CA 93106
ebelding@cs.ucsb.edu

Upamanyu Madhow
Dept. of Electrical and Computer Engineering
UC Santa Barbara
Santa Barbara, CA 93106
madhow@ece.ucsb.edu

ABSTRACT

Geopositioning using Global Navigation Satellite Systems (GNSS), such as the Global Positioning System (GPS), is inaccurate in urban environments due to frequent non-line-of-sight (NLOS) signal reception. This poses a major problem for mobile services that benefit from accurate urban localization, such as navigation, hyperlocal advertising, and geofencing applications. However, urban NLOS signal reception can be exploited in two ways. First, one can use satellite signal-to-noise ratio (SNR) measurements crowdsourced from mobile devices to create 3D environment maps. This is possible because, for example, the SNR of signals obstructed by buildings is lower on average than that of line-of-sight (LOS) signals. Second, in a sort of reverse process called Shadow Matching, SNR readings from a particular device at an instant in time can be compared to 3D maps to provide real-time localization improvement. In this paper we give a brief overview of how such a system works and describe a scalable, low-cost, software-only architecture that implements it.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

GNSS; GPS; Crowdsourcing; 3D Mapping; Localization Improvement; Shadow Matching

S3'14, September 7, 2014, Maui, Hawaii, USA.

ACM 978-1-4503-3073-2/14/09 ...$15.00.
http://dx.doi.org/10.1145/2645884.2645890.

1. INTRODUCTION

While many mobile applications require accurate geolocalization outdoors, it is an unfortunate fact that in dense urban environments positioning accuracy using the Global Positioning System (GPS) degrades significantly, with errors on the order of tens of meters [6]. The main culprit for this performance bottleneck is that in large cities the line-of-sight (LOS) to various satellites becomes occluded by buildings, leading to non-light-of-sight (NLOS) and multipath signal reception. As a result, the only satellites useful for trilateration come from a narrow area in the sky, yielding poor satellite geometries and positioning accuracy, particularly in the cross-street direction. The underlying geometry problem is not solved even as additional constellations of Global Navigation Satellite Systems (GNSS) – such as the Russian GLONASS, European Galileo, and Chinese Beidou – become available and supported by mobile devices.

One promising method to address the aforementioned satellite *Shadowing Problem* is Shadow Matching (SM) [7]. In SM, urban 3D map databases combined with real-time satellite coordinates can be used to compute the shadows of buildings with respect to various satellites. Then, low (or high) satellite signal-to-noise ratio (SNR) measurements can be used to match the device's location to areas inside (or outside) various shadows, thereby reducing positioning uncertainty. Since, for example, any GNSS-capable Android smartphone or tablet can provide via the Location Application Programming Interface (API) its estimated position with uncertainty, as well as the satellite coordinates and SNRs, SM can be done entirely in software without any additional infrastructure. A major problem with SM, however, is that update-to-date 3D maps of urban environments are not always available, and even if they are they can be expensive to obtain. Fortunately, as shown first in [4] and as we further elaborated on in [1, 2], large amounts of satellite SNR data can be used to *create* 3D maps. Intuitively, this is possible by assigning many crisscrossing receiver-satellite rays likelihoods of blockage based on measured SNRs, and then stitching these rays together into 3D maps. If the data is crowdsourced from many GNSS devices and cloud-based computation is leveraged, building such 3D maps can be

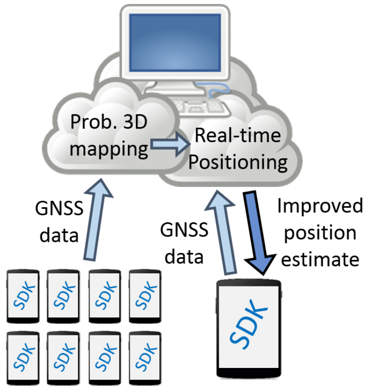

Figure 1: Proposed system architecture.

done cheaply and scalably, enabling SM-based positioning improvement anywhere GNSS data is regularly collected.

2. SYSTEM OVERVIEW

A simplified schematic representing a version of the system we proposed in our earlier conference paper [3] is shown in Figure 1. In this system, a Software Development Kit (SDK) is distributed among many mobile devices which allows for crowdsourcing of GNSS data. For each device, this GNSS data includes its estimated latitude and longitude coordinates with an uncertainty value from various time instants, along with the azimuth, elevation, and SNR of each satellite in view. Cloud-based machine learning routines are then used to process large amounts of this data into probabilistic 3D maps of the environment; these maps are continually updated as additional data becomes available. We give a brief overview the algorithms used for mapping in Section 3. Once a probabilistic estimate of the 3D environment is available in a given area, GNSS data from a single mobile device can be streamed via the same SDK to a cloud-based Bayesian filter which performs SM. Finally, revised position estimates are fed back to the device in real-time; we give a summary of the localization improvement filter in Section 4. As previously mentioned, the above system can be implemented entirely in software at the application level, although it should be acknowledged that certain aspects of the positioning improvement portion could be integrated directly with the GNSS receiver, for example as proposed in [5], with potentially additional localization gains.

3. PROBABILISTIC 3D MAPPING

The considered mapping problem can be succinctly described as using noisy GNSS position and SNR measurements, y and z, to estimate a 3D map m. In our previous works [1–3] we proposed several different methods to compute probabilistic estimates of the map. A common thread among all three approaches is the choice of an *Occupancy Grid* (OG) model, where the environment is partitioned into a 3D grid of cube-shaped "voxels" (or "cells") m_i, each of which can either be empty ($m_i = 0$) or occupied ($m_i = 1$). For the sake of brevity, in this paper we skip a lengthy discussion justifying this choice of a map model. Assuming this representation, a natural question is then the following: Given all of the measurement data, what is the likelihood that each voxel is occupied (or empty)? Mathematically,

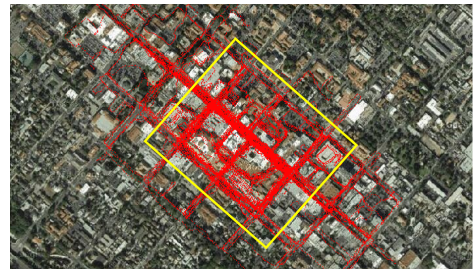

Figure 2: Google Maps aerial view of downtown Santa Barbara, with GNSS traces in red and mapped region outlined in yellow.

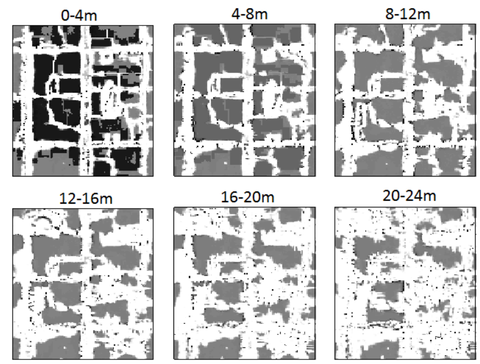

Figure 3: Horizontal layers of the generated occupancy map of downtown Santa Barbara. White/black corresponds to areas identified as empty/occupied, with shades of grey in between.

this is equivalent to determining the marginal posterior distributions, $p(m_i|y, z)$, for all i. However, because the *exact* paths of the devices x are unknown, to arrive at a solution for the map it turns out one *also* must estimate quantities of the form $p(x_t^j|y, z)$, where x_t^j is the position of a particular device j at time t. In the robotics community, this is referred to as the Simultaneous Localization and Mapping (SLAM) problem, although in this context it is SLAM for the purposes of mapping, not localization.

The major difference between the works [1–3] is that each strikes a different balance between computational efficiency and mapping accuracy. At one extreme is [3] which describes a lightweight algorithm to recursively estimate the map and receiver path given sequential GNSS measurements. However, in that work (as in virtually all recursive OG mapping techniques), we rely on cell independence assumptions which, although vastly simplifying, are known to be flatly incorrect and lead to overconfident results. In [2] we propose to explicitly model the dependencies of the mapping problem using a Bayesian network, and apply a scalable version of Loopy Belief Propagation, a graphical machine learning algorithm, for inference purposes. In that work, though, we make the simplifying assumption that all GNSS position fixes are error-free, i.e., $x = y$. In the third paper [1], we relax this assumption and again tackle the SLAM problem using a similar but more complex graphical framework. Example experimental results of this last approach, leveraging OpenStreetMap data as a-priori information on the first two layers, can be seen Figures 2 and 3, which show the traces

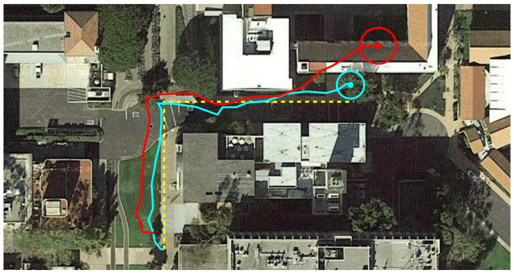

Figure 4: Positioning improvement at UCSB: true path in yellow, standalone GNSS output in red, and corrected path in light blue.

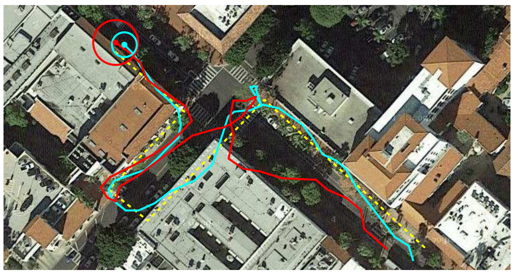

Figure 5: Positioning improvement in downtown Santa Barbara: true path in yellow, a-GNSS output in red, and corrected path in light blue.

and the generated OG map for about 25 hours of input data from 4 Android devices in downtown Santa Barbara.

4. REAL-TIME POSITIONING

The localization improvement problem can be described as follows: Given a stream of noisy GNSS SNR and location data for device j, denoted z^j and y^j, along with a noisy estimate of the map m, what is the best estimate of the device's current location x_t^j, and how confident are we in that estimate? Denoting the occupancy probabilities of the map cells around the device as o^j, and treating these as additional measurements, the quantity we are then interested in is $p(x_t^j|z^j, y^j, o^j)$. In [3] we describe a particle filtering approach which allows one to apply SM against the occupancy map in a recursive, real-time fashion, and purely in software at the mobile application level. However in this work we ignored the fact that the errors in successive GNSS position fixes are highly temporally correlated, owing especially (but not only) to the fact that these fixes are taken from the output of an Android device's navigation filter. Recently, we have developed methods to more accurately handle these correlated errors, resulting in significantly better localization, particularly where native device positioning becomes overconfident and inaccurate.

Some example results of this approach can be seen Figures 4 and 5, which show Google Maps aerial views of positioning improvement for two different mobile devices on the UCSB campus and in downtown Santa Barbara. For the interested reader, the maps used to enable this location improvement at UCSB are displayed in [1]. In the UCSB experiment, a Samsung Galaxy Tab 2 was used with only GPS+GLONASS enabled to arrive at the original position fixes, whereas in downtown Santa Barbara a Motorola Moto X smartphone with cellular assisted GPS+GLONASS (a-GNSS) was employed. It should be noted that the positioning algorithms discussed can generally supplement any assisting technologies (such as Wi-Fi, cellular, inertial, etc.) running natively on the device. Developing additional experimental results for such scenarios is an important area of future work.

5. CONCLUSION

The urban satellite shadowing problem leads to severely degraded GNSS performance, negatively impacting many mobile services which benefit from or rely on accurate geolocalization outdoors. However, it is possible to confront the source of these errors by using crowdsourced GNSS data to create 3D models of the urban environment, and then matching against these models in real-time to produce localization enhancement. Furthermore, such a system can be cheaply and scalably implemented in application level software without any additional infrastructure required. Experimentally demonstrating the scalability of the proposed mapping algorithms and the efficacy of the real-time positioning filters is a major focus of ongoing and future work.

6. ACKNOWLEDGMENTS

This work was supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

7. REFERENCES

[1] A. T. Irish, J. T. Isaacs, F. Quitin, J. P. Hespanha, and U. Madhow. Belief Propagation based localization and mapping using sparsely sampled GNSS SNR measurements. In *Proc. of IEEE International Conference on Robotics and Automation*, 2014.

[2] A. T. Irish, J. T. Isaacs, F. Quitin, J. P. Hespanha, and U. Madhow. Probabilistic 3D mapping based on GNSS SNR measurements. In *Proc. of IEEE International Conference on Acoustics and Signal Processing*, 2014.

[3] J. T. Isaacs, A. T. Irish, F. Quitin, U. Madhow, and J. P. Hespanha. Bayesian localization and mapping using GNSS SNR measurements. In *Proc. of IEEE Position Location and Navigation Symposium*, 2014.

[4] K. Kim, J. Summet, T. Starner, D. Ashbrook, M. Kapade, and I. Essa. Localization and 3D reconstruction of urban scenes using GPS. In *Proc. of IEEE International Symposium on Wearable Computers.*, pages 11–14, 2008.

[5] M. Obst, S. Bauer, P. Reisdorf, and G. Wanielik. Multipath detection with 3D digital maps for robust multi-constellation GNSS/INS vehicle localization in urban areas. In *Proc. of IEEE Intelligent Vehicles Symposium.*, pages 184–190, 2012.

[6] M. Park, H. Kim, S. Lee, and K. Bae. Performance evaluation of Android location service at the urban canyon. In *Proc. of IEEE International Conference on Advanced Communication Technology*, pages 662–665, 2014.

[7] L. Wang, P. D. Groves, and M. K. Ziebart. Shadow matching: Improving smartphone GNSS positioning in urban environments. In *Proc. of the China Satellite Navigation Conference.*, pages 613–621, 2013.