Poster Abstract: VeLoc: Finding Your Car in the Parking Lot

Mingmin Zhao, Ruipeng Gao, Jiaxu Zhu, Tao Ye, Fan Ye, Yizhou Wang, Kaigui Bian, Ming Zhang
Department of Electrical Engineering and Computer Science
Peking University, Beijing, China
{zhaomingmin, gaoruipeng, zhujiaxu, yetao, yefan, Yizhou.Wang, bkg, mzhang}@pku.edu.cn

Abstract

We present VeLoc, a smartphone-based vehicle localization approach that tracks the vehicle's parking location using the embedded accelerometer and gyroscope sensors. It harnesses constraints imposed by the map and landmarks (e.g., speed bumps) recognized from inertial data, employs a Bayesian filtering framework to estimate the location of the vehicle. We have conducted experiments in 3 parking lots of different sizes and structures, using 3 vehicles and 3 kinds of driving styles. We find that VeLoc can always localize the vehicle within $10m$, which is sufficient for the driver to trigger a honk using the car key.

Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Microprocessor/microcomputer applications; I.5.4 [**Applications**]: Signal Processing; H.1.2 [**User/Machine Systems**]: Human information processing; H.4.m [**Miscellaneous**]

Keywords

Vehicle Localization, Indoor Localization, Inertial Tracking, Robotic Navigation, Virtual Landmarks

1 Introduction

Remembering where a vehicle was parked has proven a hassle in large parking structures. Existing RF signature based indoor localization technology is not applicable where such signals may not be available such as at underground parking lots. Instrumenting additional sensors may solve the problem but at the cost of significant overheads in time, money and human efforts.

VeLoc is a vehicle localization system that utilizes accelerometer and gyroscope sensors in the smartphone to provide accurate vehicle localization. It does not rely on GPS or RF signals, neither require any additional sensors to instrument the parking ground.

Realizing such an inertial-based solution, however, involves non-trivial challenges. First, the driver may place the phone in arbitrary positions and road conditions may jolt the phone to change its position on the course. How to estimate the pose (i.e., the relative orientation of the smartphone to the vehicle) despite all such uncertainty and disturbances? Second, inferring the vehicle's traveling distance or even trajectory is still difficult due to the lack of periodic acceleration patterns, which are fundamental in step-counting techniques [10, 13]. Also, identifying which types of landmarks can be reliably recognized despite different parking lots, vehicles and driving styles, and how, remain open questions. Finally, since both the pose and landmark detection results contain errors, how can one track the location of a vehicle accurately, such that the driver can always find the vehicle?

As shown in Figure 1, VeLoc contains three components to deal with the above challenges. A *pose estimation* module estimates the pose of the smartphone inside the vehicle. A *landmark detection* module finds unique patterns corresponding to different types of landmarks and classify them reliably. A *location estimation* module determines the vehicle's final location from the map and detected landmarks.

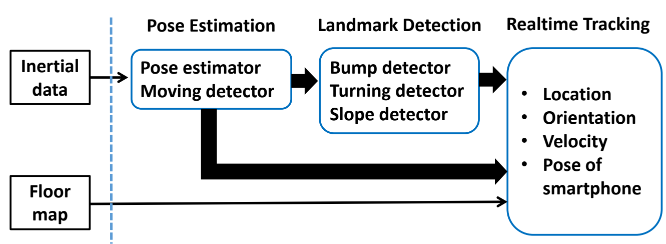
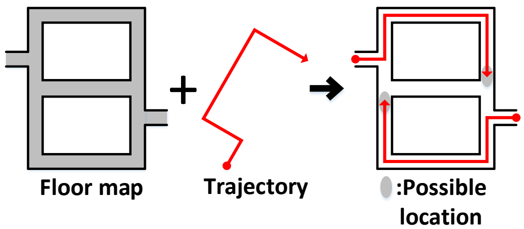


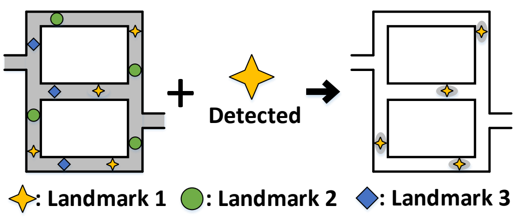
Figure 1. Three components of VeLoc. Inertial data is used to compute the smartphone's pose in the vehicle, then it is further processed to detect certain landmarks during driving, and the augmented particle filter harnesses constraints from landmark detection and the floor map for vehicle localization.

SenSys'14, November 3–6, 2014, Memphis, TN, USA.

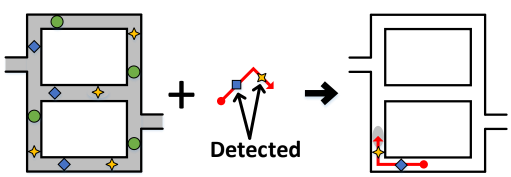
The intuition behind VeLoc is that although a noisy trajectory does not, by itself, reveal a vehicle's location, using the constraints imposed by the parking structures map and detected landmarks is able to help vehicle tracking(shown in Figure 2).



(a) Localization using constraints imposed by the map.



(b) Localization using detected landmarks.



(c) Localization using both map constraints and detected landmarks.

Figure 2. Intuition of localization. (a) trajectories can be used for localization since only a few paths on the map could accommodate the trajectory. (b) once a landmark is detected, only a few positions marked with the same kind of landmark are possible. (c) using both map constraints and detected landmark could narrow down the uncertainty more quickly.

2 Experiments

2.1 Methodology

We use smartphones to collect motion sensor readings on a vehicle in three underground parking lots, and their size are $250m \times 90m$, $80m \times 90m$, and $180m \times 50m$, respectively. Each parking lot has one entrance and one exit, and there are 298, 68, 79 parking spots, 19, 7, 12 bumps, 10, 14, 11 turns and 4, 2, 2 slopes in each parking lot, respectively. During experiments, we conduct 20 vehicle traces in each parking lot with 4 iPhones with different poses to simultaneously collect inertial sensor data during the driving.

As to evaluate our system's robustness, we invite three volunteers to drive their own cars in one parking lot, and their car's cost is around 10 thousand, 20 thousand and 30 thousand dollars, respectively. During each driving, we start the data collection app developed by us on 4 phones to record their accelerometer and gyroscope readings before vehicle enters a parking lot, and ends app after vehicle stops at a parking spot.

We measure each pose's position in the mould as ground truth for the pose estimation algorithm. The landmarks come across during the drive are recorded as ground truth for the landmark detection algorithm. Also, the final position where the vehicle stops is recorded as ground truth for the localization algorithm.

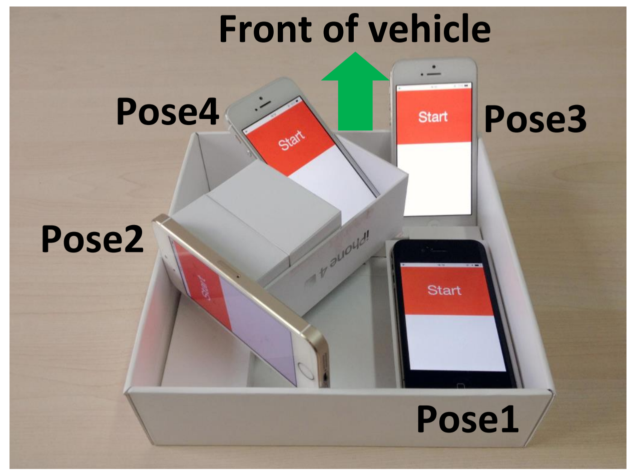


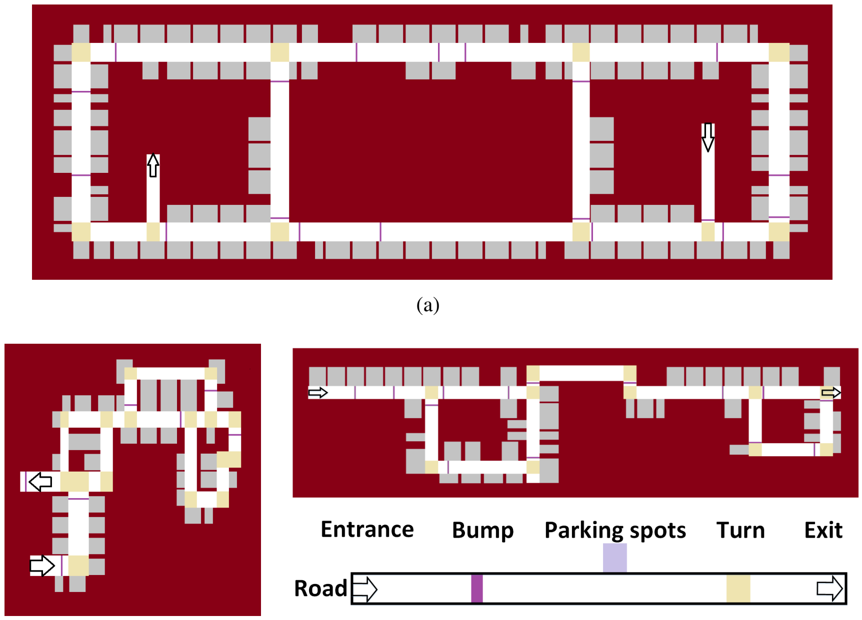
Figure 4. Mould is with 4 iPhones.

2.2 Evaluation of Pose Estimation

Here we evaluate the performance of pose estimation algorithms, namely the accuracy of estimated smartphone's pose inside vehicle. We measure the orientation error between vehicle's estimated orientation and its ground truth orientation both in smartphone's coordinate system. We use the control variate method to evaluate the effect of different smartphone poses, driving styles and parking lots.

Figure 5(a) shows the orientation error with different smartphone poses. We could observe that the 90-percentile error is around 9 degrees, and the largest error is less than 40 degrees. Figure 5(b) presents the effect of driving styles, namely drivers, on our pose estimation. We could observe that despite different drivers and performance of cars, we achieve similar pose estimation accuracy, also around 9 degrees at 90-percentile. Figure 5(c) illustrates that our algorithm works well in different parking lots, they all achieve 13 degrees accuracy at 90-percentile, and Parking lot 2 has the largest error of 16 degrees, due to its short straight road which we use to compute the vehicle's orientation.

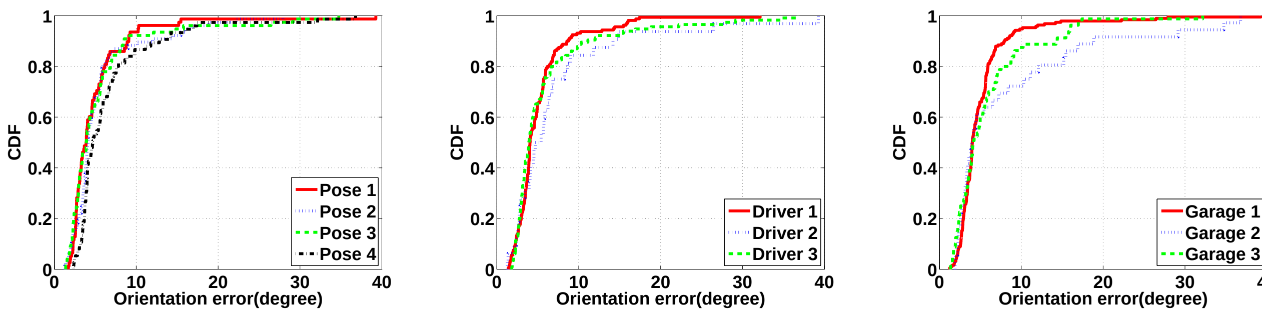
In Section 3.4, we estimated the probability of the forward and backward directions of the vehicle, $p(\theta = \theta_2)$ and $p(\theta = \theta_2)$. And we also know which one is the real forward direction as we have recorded the ground truth. Then we can calculate the distribution of estimated probability of the



(a)

(b) (c)

Figure 3. Floor map of three underground parking lots: (a) parking lot 1: $250m \times 90m$ **with 298 parking spots, 19 bumps and 10 turns. (b) parking lot 2:** $80m \times 90m$ **with 68 parking spots, 7 bumps and 14 turns. (c) parking lot 3:** $180m \times 50m$ **with 79 parking spots, 12 bumps and 11 turns.**



(a) (b) (c)

Figure 5. Pose estimation error in different scenarios: (a) 4 different poses. (b) 3 different vehicles and drivers in one parking lot. (c) 3 different parking lots.

backward direction,i.e., the estimation error. As showed in Figure 6, it achieve more than 99% accuracy at 90 percentile and a maximum error of 23%.

Figure 7 shows the pose estimation accuracy with different time windows. We could see that the 90-percentile pose estimation error is reduced when increasing the time window for pose estimation algorithms. Additionally, the localization error stays stable when time window is larger than 0.5s, which reflects realtime performance of our system.

2.3 Evaluation of Landmark Detection

Here we evaluate the performance of landmark detection using the precision and recall as metrics. As to measure the precision and recall of landmark detection, we set breakpoints where a certain landmark is announced to be detected, and we check whether it corresponds to a correct landmark on the floor map at that time stamp. We also compute how many landmarks the vehicle goes through from floor map as its ground truth number of landmarks.

Since our calibration of vehicle localization mainly relies on the landmark detection, we prefer higher precision rather than recall. This is because if we omit a certain landmark, we'll just miss a chance for recalibration. On the contrary, the detection of a nonexistent landmark will lead our particles to somewhere unknown.

Table 1 shows the recall and precision for different landmarks. We observe that turn detection has detected all the turns correctly. Bump detecion has the lowest precision of 87%, and recall of 83%. This is because gyroscope sensor is much more precise than accelerometer sensor on smartphone, and there are many activities can be confused with

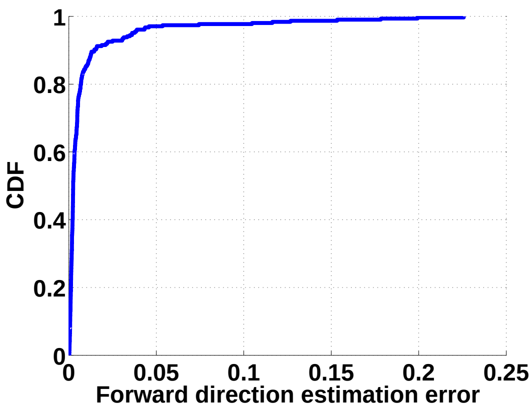


Figure 6. The distribution of estimated probability of the backward direction.

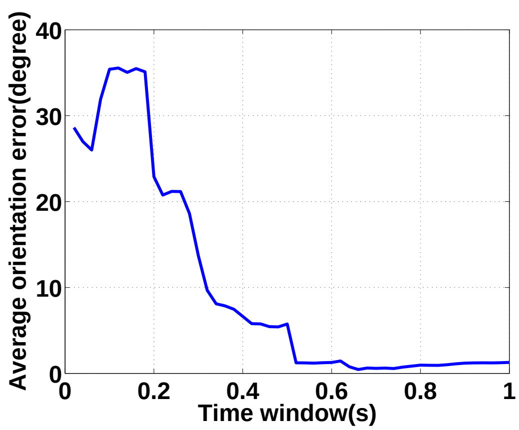


Figure 7. Orientation error of pose estimation for different time windows.

bumps.

Table 1. Landmark detection performance for different landmarks

	Bump	Turn	Slope
Precision	87%	100%	97%
Recall	83%	100%	95%

Table 2 shows the precisions with different poses of smartphone, we can observe that all precisions are quite high, while twos are relatively lower. This is because that in some positions, the smartphones are also sensitive to the jolting of the car, which may falsely be detected as a bump. Table 3 presents the effect of driving styles on our landmark detection. We achieve similar landmark detection precision, around 92%. Table 4 illustrates that we achieve quite different recall in different parking lots since the recall is effected by the property of landmarks.

2.4 Evaluation of tracking

Figure 8 shows the vehicle localization accuracy in different scenarios. The 90-percentile localization error is around 10m for all 4 poses, and the maximum errors are about 30m, shown in Figure 8(a). Additionally, as Figure 8(b) shows, different drive styles achieve localization accuracy around 10m at 90-percentile. Figure 8(c) shows the localization error in 3 parking lots, which are different since those parking lots have different shape and size.

Additionally, we evaluate the performance of the localization algorithm with different particle numbers. As we can see in the Figure 9, localization errors shrink along with the increasing particle number and almost converge when then its number exceeds a certain threshold, namely 2000 particles.

3 Related Work

We present a brief introduction of the related work below to distinguish components of Veloc from existing technologies for estimating the phone pose, detecting the landmarks, monitoring the states of a robot, a pedestrian or a vehicle, etc.

Phone pose estimation. It is impractical to assume that the smartphone inside a vehicle has a known position or it is placed stationary. It is necessary to periodically estimate the phone pose in the vehicle. Wang et al. found it possible to align the smartphone's and vehicle's coordinate systems using gravity, accelerometer and gyroscope sensors [21]. Specifically, it aligns z-axis of the vehicle with its gravity direction; gyroscope is used to determine whether the vehicle is driving straight, then accelerometer readings are extracted as vehicle driving direction, namely y-axis of the vehicle. However, this approach fails to consider the case when the vehicle runs on a slope, and simply extracting accelerometer readings as vehicle direction is not robust.

Virtual landmark detection. In robot localization system, a robot is assumed to be exploring the space of interest that has various landmarks (e.g., barcode pasted on walls or a particular pattern painted on the ceiling). The equipped

Table 2. Performance of landmark detection with different poses

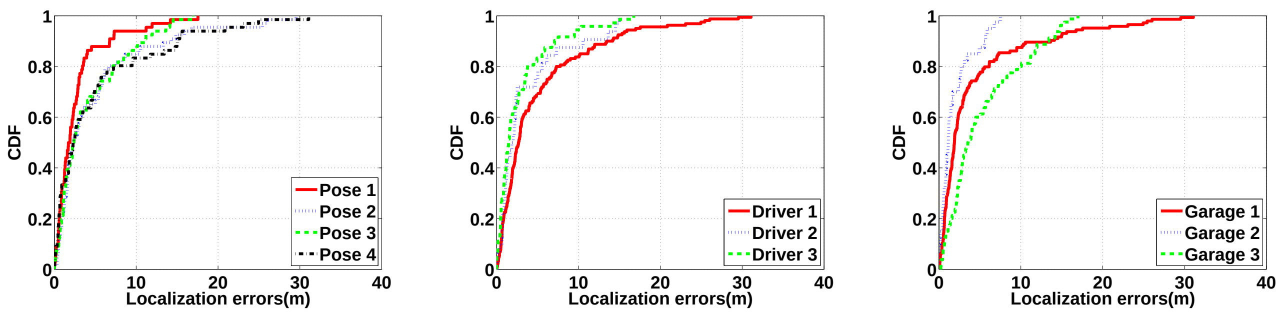
	Pose 1	Pose 2	Pose 3	Pose 4
Precision	93%	88%	85%	93%
Recall	88%	89%	87%	85%

Table 3. Performance of landmark detection with different drivers

	Driver 1	Driver 2	Driver 3
Precision	93%	92%	91%
Recall	89%	90%	87%

Table 4. Performance of landmark detection with different garages

	Garage 1	Garage 2	Garage 3
Precision	94%	90%	92%
Recall	78%	93%	91%



(a) (b) (c)

Figure 8. Vehicle localization errors in different scenarios: (a) 4 different poses. (b) 3 different cars and drivers in one parking lot. (c) 3 different parking lots.

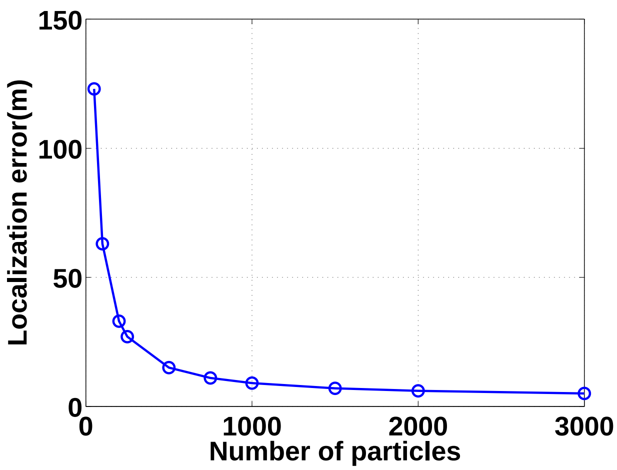


Figure 9. Performance of different particle numbers.

sensors on a robot, such as laser-based ranging and cameras, are used to detect these artificially placed landmarks. However, for smartphone-based indoor localization, it is the smartphone that is carried around by a user to explore the space of interest. It is impossible to have robot sensors (e.g., laser ranging) on a commodity phone. Thus, researchers refer to virtual landmarks which are essentially ambient signatures or recognized patterns/activities that are perceivable by smartphone sensors [1, 16], and UnLoc [20] is believed to be the first to apply virtual landmarks towards deadreckoning. In VeLoc, we use sensor measurements to detect all kinds of road anomaly and turnings as landmarks. In addition, sensor measurements also provide cues for estimating the state of vehicles. For instance, different patterns between a immobile vehicle and a moving one can be view as a measurement of the velocity of the vehicle.

Robotic localization. SLAM is a popular technique in robotics which allows the robot to acquire a map of its environment while simultaneously localizing itself relative to this map [12]. Recently, WiFi-SLAM [7] was proposed to utilize the WiFi signal strength as the input of SLAM. Unlike SLAM, VeLoc assumes the availability of a map and the problem to be addressed is equivalent to the robot localization problem of determining the pose of a robot relative to a given map of the environment [18].

The early work on robot localization problem used Kalman Filters which is thought to be the earliest tractable implementations of the Bayes filter for continuous spaces. Subsequent work has been based on Markov localization, which is a better match in practice since it allows the robot's position to be modeled as multi-modal and non-Gaussian probability density functions. Of particular interest to us is the Monte Carlo Localization (MCL), or particle filtering based approach [8, 19]. Instead of representing the distribution by a parametric form, particle filters represent a distribution by a set of samples drawn from this distribution. Those particles are then evolved based on the action model and the measurements [18]. Again, robot localization typically depends on using explicit environment sensors, such as laser range finders and cameras. Moreover, the rotation of the robot wheels offer a precise computation of displacement.

Unlike robot localization systems, VeLoc is independent of laser ranging or cameras, but it uses smartphone sensors to compute the displacement and direction of vehicles and detect the virtual landmarks that are only specifically available in parking lots.

Dead-reckoning. Dead reckoning using inertial sensors is a well explored approach to monitor the states of a moving object or a pedestrian. However, conventional sensors used in these applications are very expensive. Recently, it is attractive to use smartphone sensors in indoor environments [4], since consumer mobile devices are increasingly being equipped with sensors such as accelerometer, gyroscope, magnetometer and barometer. However, directly applying this approach in indoor environments is non-trivial since many factors cause fluctuations in acceleration, resulting in erroneous displacements.

Many methods have attempted to mitigate the accumulation of error. Foot-mounted sensors have been shown effective in reducing the error [15, 23]. However, the accumulation of error remains when a smartphone pose is unknown.

Outdoor localization schemes like CompAcc [5] employ a periodic GPS measurement to recalibrate the users location. UnLoc [20] provides an option to replace GPS with virtual indoor landmarks that can be detected using existing sensing modalities for calibration. Dead-reckoning techniques have been widely used in mobile computing community with the purpose of addressing the indoor localization problem [14].

To prevent the error accumulation, VeLoc simultaneously harnesses constraints imposed by the map and environment sensing. The only external input for VeLoc is a map of the indoor space of interest. Since the map of a place do not change for several months or years, no repeated manual efforts for calibration are required by VeLoc. In addition, the need for special-purpose hardware and infrastructure is avoided to make VeLoc more practical for the real-world use.

Estimation of vehicle states. There have been many active research efforts in using smartphones' embedded sensors (1) to monitor the states of vehicles (e.g. dangerous driving alert [10], car speaker [24] and CarSafe [25]); (2) to inspect the road anomaly/condition (e.g., Pothole Patrol [6] and Nericell [11]); and (3) to detect traffic accidents (Nericell [11] and WreckWatch [22]).

The vehicle speed is a critical input for implementing these applications. It is easy to calculate the outdoor vehicle speed by using the phone GPS [9, 17], while the GPS signal is weak or even unavailable at indoor parking lots. Some alternative solutions leverage the phone's signal strength to estimate the vehicle speed [2, 3].

4 Discussion

The start of the trace. It is possible that the car is not stationary when the VeLoc application starts, or that it does not start exactly at the entrance of the parking lot. In these cases, VeLoc can employ two heuristics to determine what time should be considered the start of the trace by examining when: (1) the smartphone loses its GPS signal, which is usually the time it enters a parking structure such as an underground lot; and (2) the first (or certain) landmark is detected, which can be used to back trace the time when the car enters the structure (as illustrated in Section **??**).

Complex parking structures. There are many other types of complex parking structures such as a multi-storey parking lot with up/down-slope parking spaces. VeLoc can support vehicle tracking in these complex parking structures: (1) the turn and slope detection algorithms can jointly determine which storey the vehicle is located; (2) the pose estimation module can tell whether the vehicle is parking on the slope and calibrate the coordinate systems accordingly.

Other extreme cases. The smartphone may keep jittering when the vehicle is on a jerky road, or the driver puts the phone in his pocket. These extreme cases require VeLoc to estimate the phone pose in real time. To address this problem, VeLoc employs a sliding time window technique, which shows that it can track the pose quite accurately in just one second (Figure 7). This avoids large errors that may arise due to the latency in pose estimation.

5 Conclusion and Future Work

In this paper we describe VeLoc that can track the vehicle's movements and estimate the final parking location using the smartphone's inertial sensor data only. It does not depend on GPS or WiFi signals which may be be available in environments such as underground parking lots, or require additional sensors to instrument the environment. VeLoc first estimate the pose of the smartphone relative to the vehicle, so that inertial sensor data can be transformed into the coordinate system of the vehicle. It then detects landmarks such as speed bumps, turns and slopes, and combine them with the map information to estimate the vehicle's location using a probabilistic model. Experiments in three parking structures have shown that VeLoc can track the parking locations to within 4 parking spaces, which is enough for the driver to trigger a honk using the car key.

Currently VeLoc depends on accurate parking structure maps to reduce the uncertainty in the vehicle location. Since such maps are not always available, we plan to study how to obtain the map information, and track the vehicle when only incomplete and/or inaccurate map is available. This further extend VeLoc's capability in the real world.

6 References

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: Mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, pages 261–272, New York, NY, USA, 2009. ACM.

[2] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. Martin, J. Yang, and Y. Chen. Tracking vehicular speed variations by warping mobile phone signal strengths. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 213–221, March 2011.

[3] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen. Vehicular speed estimation using received signal strength from mobile phones. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, pages 237–240, New York, NY, USA, 2010. ACM.

[4] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see bob?: Human localization using mobile phones. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 149–160, New York, NY, USA, 2010. ACM.

[5] I. Constandache, R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, 2010.

[6] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 29–39, New York, NY, USA, 2008. ACM.

[7] B. Ferris, D. Fox, and N. D. Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007.

[8] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999:343–349, 1999.

[9] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines

for distributed privacy-preserving traffic monitoring. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 15–28, New York, NY, USA, 2008. ACM.

[10] J. Lindqvist and J. Hong. Undistracted driving: A mobile phone that doesn't distract. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, HotMobile '11, pages 70–75, New York, NY, USA, 2011. ACM.

[11] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: Using mobile smartphones for rich monitoring of road and traffic conditions. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 357–358, New York, NY, USA, 2008. ACM.

[12] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, 2002.

[13] S. Nawaz, C. Efstratiou, and C. Mascolo. Parksense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom '13, pages 75–86, New York, NY, USA, 2013. ACM.

[14] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 293–304, New York, NY, USA, 2012. ACM.

[15] P. Robertson, M. Angermann, and B. Krach. Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors. In *Proceedings of the 11th International Conference on Ubiquitous Computing*, Ubicomp '09, pages 93–96, New York, NY, USA, 2009. ACM.

[16] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 155–168, New York, NY, USA, 2011. ACM.

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

[18] S. Thrun, W. Burgard, D. Fox, et al. *Probabilistic robotics*, volume 1. MIT press Cambridge, 2005.

[19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1):99–141, 2001.

[20] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 197–210, New York, NY, USA, 2012. ACM.

[21] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin. Sensing vehicle dynamics for determining driver phone use. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 41–54, New York, NY, USA, 2013. ACM.

[22] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt. Wreckwatch: Automatic traffic accident detection and notification with smartphones. *Mob. Netw. Appl.*, 16(3):285–303, June 2011.

[23] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pages 114–123, New York, NY, USA, 2008. ACM.

[24] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin. Detecting driver phone use leveraging car speakers. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 97–108, New York, NY, USA, 2011. ACM.

[25] C.-W. You, N. D. Lane, F. Chen, R. Wang, Z. Chen, T. J. Bao, M. Montes-de Oca, Y. Cheng, M. Lin, L. Torresani, and A. T. Campbell. Carsafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 461–462, New York, NY, USA, 2013. ACM.