Location Detection for Navigation Using IMUs with a Map Through Coarse-Grained Machine Learning

J. Jose Gonzalez E.*, Chen Luo*,
Anshumali Shrivastava, Krishna Palem
Rice University
Houston, Texas, USA
Email: {jjg11, cl67, anshumali, kpalem}@rice.edu

Yongshik Moon, Soonhyun Noh,
Daedong Park, Seongsoo Hong
Seoul National University
Seoul, Korea.
Email: {ysmoon, shnoh, ddpark, sshong}@redwood.snu.ac.kr

Abstract*—Location detection or localization supporting navigation has assumed significant importance in the recent past. In particular, techniques that exploit cheap inertial measurement units (IMU), the gyroscope and the accelerometer, have garnered attention, especially in an embedded computing context. However, these sensors measurements are quite unreliable, and it is widely believed that these sensors by themselves are too noisy for localization with acceptable accuracy. Consequently, several lines of work embody other costly alternatives to lower the impact of accumulated errors associated with IMU based approaches, invariably leading to very high energy costs resulting in lowered battery life. In this paper, we show that IMUs are sufficient by themselves if we augment them with known structural or geographical information about the physical area being explored by the user. By using the *map* of the region being explored and the fact that humans typically walk in a structured manner, our approach sidesteps the challenges created by noise and concomitant accumulation of error. Specifically, we show that a simple coarse-grained machine learning approach mitigates the effect of the noisy perturbations in the information from our IMUs, provided we have accurate maps. Throughout, we rely on the principle of inexactness in an overarching manner and relax the need for absolute accuracy in return for significant lowering of resource (energy) costs. Notably, our approach is completely independent of any external guidance from sources including GPS, Bluetooth or WiFi support, and is this privacy preserving. Specifically, we show through experimental results that by relying on gyroscope and accelerometer data alone, we can correctly identify the path-segment where the user is walking/running on a known map, as well as the position within the path with an accuracy of 4.3 meters on the average using 0.44 Joules. This is a factor of 27X cheaper in energy lower than the "gold standard" that one could consider based on GPS support which, surprisingly, has an associated error of 8.7 meters on the average.

I. Introduction

Navigation technology is expected to be a 4 billion dollar industry in itself by 2018 [1]–[11]. Increased demand for accurate indoor navigation is a combined effect of an increase in venue-based marketing and government initiatives in developing positioning systems for public safety and urban security segments. GPS (Global Positioning Systems) are widely considered prohibitive because of their energy requirements in the indoor environments [11].

* Indicate Equal Contribution.

In recent years, low-cost inertial sensor based on accelerators and gyroscopes have emerged as a well-known solution for navigation in indoor or hybrid indoor-outdoor environments. The hardware required for inertial navigation is tiny, light-weight and is very frugal in resource usage. However, measurements from these sensors are extremely noisy, and therefore, existing technology utilizing these sensors have limited applicability due to significant errors.

Since sensor technologies are based on the physical theories of acceleration, most popular inference techniques with IMUs (Inertial Measurement Units) are based on numerical integration, in some form or the other. Numerical integration typically suffers from accumulated error, including Abbe error [12]. Because the guidance system is continually adding detected changes to its previously-calculated positions (see dead reckoning [6]–[10], [13]), any errors in measurement are accumulated from point to point. This accumulation leads to "drift", or an ever-increasing difference between where the system "thinks" it is located, and the actual location. Furthermore, IMUs always work with averages. So if an accelerometer is able to retrieve acceleration once per second, based on sampling the device will have to work as if it experienced the same acceleration throughout that second, although the acceleration could have varied drastically in that time period. Due to integration, a constant error in acceleration results in a linear error in velocity and a quadratic error growth in position. A constant error in altitude rate (gyroscope) results in a quadratic error in velocity and a cubic error growth in position [12], [13]. Consequently, it is widely believed that only accelerometer and gyroscope measurements are insufficient for accurate navigation.

Due to the significance of the problem and its usefulness in practice, there has been a flurry of work trying to correct accumulated errors from sensor measurements by utilizing additional information from new and independent sensors. A standard line of thought is to try to integrate additional sensor information, leading to an information fusion problem, and rely on Kalman filters [14] or other smoothing models to reduce errors. However, often, new sensors introduce noise, and therefore, information fusion using smoothing techniques fails to work at low signal to noise (SNR) ratio. A common alternative is to periodically query an independent and accurate

978-3-9815370-8-6/17/$31.00

positioning mechanism to correct these errors. Such positioning mechanisms are typically GPS based or used other popular WiFi based positioning [15], both of which are again often inaccurate in indoor environments. Communication-based positioning is further well-known to be expensive from energy perspective [11], and are further prone to privacy breaches.

In this paper, we argue that trying to rely on noisy IMU sensors to infer position and movements, with all possibilities of motions such as arbitrary angle of turns, rotations, and others, is solving a much harder problem than needed, and is also unnecessary from a practical perspective. The world that we live in a quite structured, and imposing this structure is enough to solve, often mathematically intractable, inverse problems. We have seen that imposing such real-world structural priors has led to several breakthroughs in our capabilities to understand language (text) [16] and images (vision) [17], which are otherwise very ill-posed and hard mathematical problems. We believe this is true of navigation and location detection or localization as well. For example, many of our environments as well as our walking paths, have a grid-like structure consisting of straight lines and turns. It is seldom the case that we walk on a sinusoidal path. For example, we have the map of the navigating environment, such as a shopping mall beforehand. In this work, we heavily leverage this structure to obtain an accurate position estimation algorithm relying only on gyroscope and accelerometer measurements only.

We show that given the maps of the indoor environment, indicating all the navigable straight line paths and turns, the noisy accelerometer and gyroscope measurements are sufficient for accurate navigation. The key observation we exploit is that although IMU measurements are very noisy for complete inference by themselves, they are very accurate for coarse-grained discrete decisions such as detecting turns. These coarse-grained decisions with crude distance estimation over short intervals when combined with a map provided an accurate position estimation algorithm. Surprisingly, we note that our inexact approach has quality or error comparable to GPS, while the later is more expensive by a factor of 27X in energy.

Our contributions:

- We provide the first algorithm which combines accelerometer and gyroscope measurements with cheap coarse-grained machine learning along with the map of the environment, for accurate navigation. We leverage the prior grid-like structure of the navigating environment to correct the noisy measurements, without requiring any additional sensor data. Additional sensory information is often expensive, which as we show can be completely avoided.
- The knowledge of the environment can be leveraged to avoid costly and inaccurate numerical integration, and instead, we rely on coarse-grained machine learning based inference such as detecting the type of turns and distance traveled in short intervals. We further show that such coarse-grained inference is remarkably accurate with machine learning algorithms despite high measurement errors.
- We provide evaluation using datasets having gold standard position measurements over 5.35 km with completely accurate measured distances.
- Our algorithm does not require any communication with the external environment and hence preserves users privacy.
- Using the principle of *inexactness*, we relax the need of exact estimations but instead, one willing to accept a good enough solution. In return, we see a significant reduction in resource lost, -energy consumption in this case. We contrast our approach with the best known approach that is widely-used, GPS, in Table I. With noisy measurements, often using very precision data is misleading. In such scenarios, coarse-grained information is more reliable. Similar phenomena occur in signal processing [18], and is the main reason why inexact computing shows significant advantages in the context of machine learning [19]–[23].

TABLE I: Energy and accuracy of our method vs GPS on the map described in Fig. 1

Method	Average Error	Energy
GPS	8.7m	11.85J
Our Method	4.3m	0.44

II. Overview of The Proposed Approach and Mode

We assume that we have the map of the environment which is modeled as a variant of a ”Manhattan Grid” of Fig. 1c. Such a map is almost always available beforehand. For example, it is common to have map (or plans) of shopping malls, large buildings, and campuses. We assume that the walking paths are straight lines, with turns, not necessarily right angles, including u-turns. This is not an unrealistic assumption, as humans usually walk along paths which can be accurately approximated with a union of reasonably small straight line segments. Using this scenario, our method allows us to eliminate costly and inaccurate numerical integration. Our guess of final position estimation to perform localization, given the starting point, can be broken down into two major task: 1) distance estimation along a straight line and 2) turn estimation. We briefly describe the two tasks.

1.Distance traveled within a path segment: For accurate position estimation, we should be able to determine the distance during sufficiently short time intervals of time. Instead of using numerical integration, *we cast this as a regression problem given the accelerometer measurements over the specified time interval in a feature space.* We show that coarse-grained regression over short time intervals is significantly more accurate than numerical integration. (See section IV for details.)

2. Turn determination: We need to infer, within a reasonably short time window, whether there was a turn. If there was a turn, we also want to infer the type and angle of the turn, including about u-turns. We formulate this as a classification problem given the gyroscope readings over a reasonably large time interval. We were able to achieve a remarkable accuracy of around 90-95% in classifying turns using non-linear machine learning algorithms. (See section V for details)

It should be noted that these two tasks with perfect accuracy are sufficient for position estimation in a Manhattan-like environment. To estimate distances and turns in that window, we implement ”sliding window” over time. In a grid like environments, knowing the distances and the turns accurately will help locate the final position. However, perfect accuracy is far from being realistic in the two elemental steps locally. Moreover, since incorrect estimation gets accumulated, the error aggregation leads to blow up in errors over time.

The good news: our results show that machine learning algorithms, especially non-linear classifiers such as random forests, are significantly accurate for turn estimation with an accuracy of around 90-95%. Such high accuracy is achievable because, even though gyroscope measurements are quite noisy, during turns, there are drastic changes in the signs of angular acceleration, which can be easily identified by machine learning to discriminate between different types and angles of turns. In other words, while the gyroscope measurements are too noisy for accurate estimation of angular acceleration there is enough information to discriminate between various coarse-grained turns.

For distance estimation over a fixed short interval of time, we observe that machine learning algorithms that operate in the frequency domain using the Fourier spectrum of accelerometer data are significantly more accurate than those techniques including numerical integration that operate in the time domain. Our results demonstrate that we can achieve a mean error of 7.2m over 15sec time interval compared to 22m during the same time interval achieved by numerical integration (see Section IV for analysis). However, distance estimation, as expected, suffers from the problem of error accumulation over time as it involves aggregating estimates over small windows. Without periodic corrections, this error will amplify over longer paths. We show that a simple methodology of combining the information in the map indicating the feasible walking paths and locations of turns, we were able to correct these errors with remarkable accuracy. The moment we detect a turn, which matches with the map, with near perfect accuracy, the error on the user's location is corrected to zero. We found that using this simple correction, our final accumulated average end-to-end position estimation error is on average 4.3meters.

Organization: We start by describing the environment and datasets used for evaluations and training machine learning in Sections III. In Section IV and V, we describe the two basic pillars of distance and turn determination respectively. Section VI details our final procedure that integrates the two approaches with the information from the map for obtaining the final position.

TABLE II: Distances of Rice University's Academic Quad.

Dimension	Edges - Distance			
Horizontal	1-12 78m	12-11 16m	11-10 19m	10-9 50m
Vertical	1-2 31m	2-3 31m	NA	NA

III. Data Collection

Our approach relies on supervised machine learning for distance and turns estimation. We, therefore collected labeled datasets for training our models from two independent walking experiments. We describe these datasets next.

A. Source 1: GPS Labeled Data for Distance Estimation

Distance estimation approached by us as a regression problem requires adequate supervision and comprehensively labeled data. To automatically generate the ground truth labels for extensive data we relied on GPS measurement in an outdoor environment. Fig. 1a describes the place where the data were gathered. The dataset collected have seven components: 3-axis accelerometer readings, location based on GPS coordinates (3 values), and the timestamp indicating the time of each reading. The sampling frequency is around 51Hz. The data was generated by a single user who walked/ran for 9 hours on 12 different paths in different modes. The data was taken with an iPhone6s. The walking conditions were varied. This dataset was primarily used for training the distance estimation procedure.

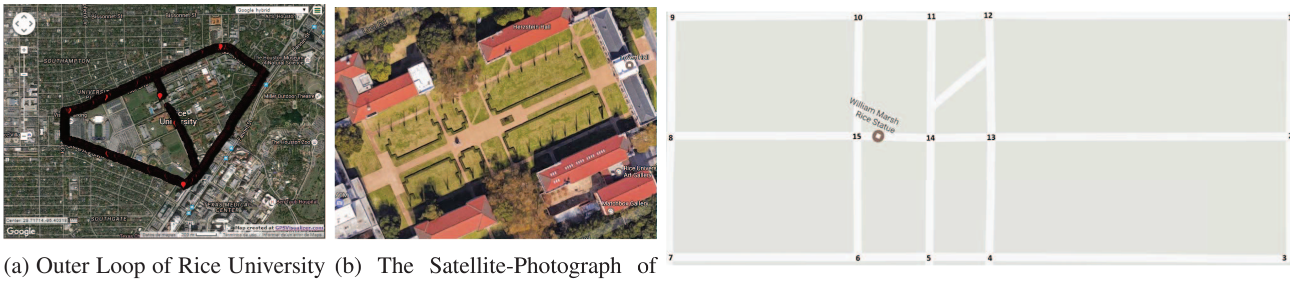
We also characterized the error inherent to GPS. For this purpose, 50 minutes of data were taken in the place mentioned in Fig 1a. GPS measurements were taken with two iPhones fixed on the waist. The difference between the two distances was 4.86 meters on average with 15 seconds interval.

B. Source 2: Data with Exact Labels

We chose the Academic Quad of Rice University as is shown in Fig. 1. The dataset collected has ten components, 3-axis gyroscope, 3-axis accelerometer, location based on GPS coordinates and the timestamps. The sampling frequency is around 51Hz. Data was collected by two users who walked for 1 hour and 15 minutes in different modes: fast, slow, run. The data was taken with two iPhone6s, one for each user. The dataset contains 100 turns over 45 minutes on the first day, and 50 turns on the second day. The types of turns include U-turns and any turn possible on the map. For this case, we do not rely on GPS-based location labels. Our ground truth was the actual distance of the grid which is described in Table II. To exactly label positions and distances, the grid was sampled with numerous fixed checkpoints. Given the starting point, the timestamp to reach different checkpoints was recorded to label the gyroscope and accelerometer data with actual coordinates of the checkpoints. All our validations, including the final validation of our end to end algorithm, was performed on this dataset. Since this dataset has sufficient turn information, it was also used for training our turn classification ML algorithm described in Section V.

IV. Distance Estimation in straight line

In this section, we describe our process for distance estimation along the straight line. For straight line distance, we used



(a) Outer Loop of Rice University Campus

(b) The Satellite-Photograph of the Academic Quad.

(c) The Topographic Map of the Academic Quad

Fig. 1: Maps for the experiment of the work.

TABLE III: Distance Estimation using different machine learning algorithms and different feature types.

Method	Percentage Error Rate (Error in Meters)	
	Frequency Domain	Time Domain
Nearest Neighbors	26.34% (8.18m)	67.68%(20.77m)
Ridge Regression	27.08% (8.39m)	71.23%(22.01m)
Random Forest Regression	23.5% (7.2m)	66.32%(20.46m)
Numerical Integration	NA	70% (22m)

accelerometer data. Given accelerometer data over a given interval of time, typically between turns, we want to estimate the distance traveled. In theory, this can be easily calculated using the physics of acceleration, which is a simple numerical integration. However, noise and sampling approximation from the accelerometer leads to error aggregation over time. This noise accumulation leads to very poor accuracy. We take a different approach and cast it as a machine learning problem over coarse-grained intervals of time.

The accelerometer data leads to a 3-D time series, which are the features. This time series is affected by noise. To reduce noise, we only consider frequencies below 15Hz where more of the energy from human gait can be found as shown in [24]. We develop features by transforming this data into the frequency spectrum. To obtain enough resolutions of signal frequency, we define k frequency bands on the interval [0, 15Hz], where k is one of the parameters of our model.

We perform a comprehensive study of different machine learning algorithms including linear and non-linear models for regression over the data. In our case, we considered near neighbor, ridge regression, and the random forest. To understand the effect of noise in both time and frequency domains we analyzed both cases.

Nearest Neighbor Regression [25] uses the nearest value in historical data set to the given data as the prediction value. Ridge Regression [26] build a linear model with an L2 constraint on the data set and uses the linear model to do the further prediction for the given data. Random Forests [27] partitions the data space into smaller regions, and do the simple regression in each region. Random Forest is powerful for dealing with nonlinear predictive models.

To test the effectiveness of the machine learning methods, we use numerical integration as our baseline algorithm using 15 seconds interval as well. The acceleration obtained from the accelerometer is filtered to reduce noise. Next, we integrate the data to obtain the average speed during the interval.

Table III provides a summary of the results. We can clearly see that in the time domain, all the ML methods including numerical integration performs significantly worse, compared to methods using Fourier coefficients as features in the frequency domain. This is expected because of the noisy nature of the accelerometer data. Walking introduces different kind of spurious accelerations which are usually noise, and affects the distance estimation. Noise is easier to identify in the high-frequency domain. For example, truncating high-frequency signals does result in significantly better performance. From Table III, among machine learning models, we can see that nearest neighbors method and ridge regression performed worse, whereas random forest regression performed best. Even in the time domain, machine learning models are superior to integration method, although by a smaller margin. Overall, the best we can obtain is mean error of 7.2m (23.5% relative) which is significantly superior to standard numerical integration error of 22m (70% relative). We note that our results exploit the stability of working in the frequency domain while numerical integration is a time domain method.

V. ESTIMATING TURNS.

Another critical component of our navigation framework is the estimation of turns. In particular, at each time, given the current accelerometers and gyroscope data, estimate (1) whether the person takes a turn or not, (2) if the person takes a turn, identify the (coarse-grained) type of the turn: Right Turn, Left Turn, or U-Turn.

We use gyroscope data for calculating the coarse-grained turn information. Although, gyroscope data can be used to exactly compute the angle of the turn with high precision, the noise in each measurement renders exact estimation is quite poor. Techniques relying on accurate angle estimation of turns will also suffer from accumulated errors which grow cubically with the amount of gyroscope data [28]. We argue that such fine precision information is not necessary and in fact is detrimental. We also show that there is enough discriminative information in gyroscope measurement for near perfect discrimination between coarse-grained turn categories. We further show that this coarse-grained information is sufficient for our task. In particular, we cast the turn estimation problem as classifying (discriminative) categories of turns given gyroscope measurements as features. The core problem is to find a function, for mapping the noisy gyroscope data to a turning class (No Turn, Right Turn, Left Turn, U-Turn).

TABLE IV: The accuracy for Turning detection and Turning classification. Both results comes from using Frequency domain features and Time domain Features. NN (Nearest Neighbor Method), RF (Random Forest Classification), and SVM (Support Vector Machines) are used. RF got the best accuracy on turning detection and turning classification tasks.

Method	Turn Detection Accuracy		Turn Classification Accuracy	
	Frequency	Time	Frequency	Time
NN	52%	54%	33%	31%
RF	67%	83%	65%	81%
SVM	72%	**95%**	69%	**90%**

Our datasets show that every turn takes around four seconds to finish. We, therefore, break the time series data into overlapping intervals of four seconds. Increasing the length of this interval does not affect the results. Now, the classification problem of turn estimation is formulated as: given gyroscope measurements, as 3-D time series over 4 second time intervals, classify whether a turn is present or not. If a turn is present, then classify whether it is left turn, right turn or a U-turn.

Having time series gyroscope data, we again use the time and frequency domain features, with high-frequency features removed, as described in Section IV.

We tested three popular classification algorithms: nearest neighbor, support vector machine, and random forest classification on the data set. In the previous section, we already briefly introduced nearest neighbors method, random forest methods. SVM (or support vector machines) [29] uses a kernel function to map the training data point into a high-dimensional kernel space so that in the kernel space the data is separable by a hyperplane. In this paper we used Gaussian kernel.

The results are shown in the Table IV. We make the turning estimation into two steps: Turn Detection and Turn Classification. (1) in the turn detection step, we detect whether or not there is a turn, (2) if there is a turn, then we do turn classification, that classifies the turn into three classes.

From Table IV, we can see that for turn detection step (turn or no-turn), the SVM (support vector machines) methods can achieve an average of 95% accuracy by using the time domain gyroscope data. And from the Table IV, we can see that for turning classification step, SVM (support vector machines) can achieve an average of 90% accuracy. Thus, coarse-grained turn estimation can be performed with very high accuracy.

Note: with turn estimation, we found that time domain features works better. This is expected because the turning time is usually short, and hence the frequency domain does not have sufficient resolution.

VI. Leveraging the Map to Correct Errors: Final Algorithm

In this section, we discuss our approach to combine these two methods with the map information to correct the accumulated errors yield to accurate localization algorithm.

For example, distance estimation in a short interval of 15 seconds has an average error of around 7 meters. A single turn estimation, although nearly perfect, still can be wrong with around 0.05 probability. These errors can accumulate over time. Without map based correction, the average error was 28.72m, the maximum error is 99m, and the minimum is 0.222m. This high error is expected without correction as errors accumulate. On the other hand a simple map based correction, as we show next, drastically reduces this error.

However, it is possible to correct these errors using the map information. Define each intersection point on the map as a vertex and each path between vertices as an edge. We use the following simple strategy to correct these errors.

Step1 Set the initial position.

Step2 Keep a running window (overlapping) of 4sec time interval. Use the turn classification algorithm in Section V over these windows. Report all the turns on the gyroscope data series.

Step3 Do distance estimation between two turns using random forests as shown in Section IV. Accumulate distance estimates with 15sec non-overlapping windows.

Step4 Based on the map information, estimate whether a turn is a valid turn. Given the distance estimate, determine if an actual turn is in within the error of a predicted turn. If we do not find an intersection in the vicinity, then skip this turn. If a turn is found, then correct the current position (distance estimate correction) to the point of the intersection.

We tested our method on 200 paths having different variations as mentioned in section III. The path length ranges from 3 to 60 meters, with an average path length of 357 meters. The average number of turns in each path is around 6.

Result. For the 200 random paths, all the final edges were correctly identified in all cases. 100% accuracy for path identification is quite remarkable. We believe that this is partly due the fact that coarse-grained turn estimation has near perfect accuracy with random forest. For the position estimate, we get an average error of 4.3 meters. The best case error is 0.05 meters, and the worst case error is 11.5 meters. In Fig. 2, we highlight some results for the final edge of two Random Paths.

A. Energy Consumption compared to GPS

Error correction based on costly sensors, such as GPS, is common in practice. However, in addition to being inaccurate, GPS sensors consumes a significantly larger energy which drains the battery life of the device. We compare the energy consumption of our proposed method with GPS. We use a 15s data to test on both methods. The proposed method requires mere 0.44J whereas GPS signal in such a short interval needs 11.85J. Therefore, our proposed method consumes 27x less energy than GPS. In this estimate, we included both the energy associated with sensors, i.e. accelerometer and gyroscope, as well as the energy associated with computation. The energy consumed by the sensors is 0.271311J (61.7% of the total energy) and the rest (0.3834%) is because of the calculations involved with our method.

VII. Conclusion and Future Work

There are three directions to pursue a possible future work. First, we propose more extensive testing in terms of the

Fig. 2: (a) Example Random Path 1: **1–14–1–2–15**. Final Position (Red Circle) on the edge between 15 and 2. (2) Example Random Path 2: **3–4–3–2–3**. Final Position(Red Circle) on the edge between 4 and 3. (3)Position Localization for Random Path 1 and Random Path 2. The black circle in the figure means the final position estimated by the proposed method. The red circle in the figure shows the ground truth location. The error for Random Path 1 is 0.8 meters and the error for Random Path 2 is 4.3 meters.

number of paths and their combinations. Second, we intend to provide a detailed analysis of the sources of error. Our plan is to lower the amount of error with 3 meters being our goal. Finally, we plan to compare energy usage and accuracy with other competing approaches including WiFi and Bluetooth, beyond GPS.

Acknowledgment

We thanked Professor John Hennessy for discussions and for recommending the distance estimation and navigation question studied in this paper. This work was supported in part by DARPA Grant FA8750-16-2-0004.

References

[1] R. Reimann, A. Bestmann, and M. Ernst, *Locating Technology for AAL Applications with Direction Finding and Distance Measurement by Narrow Bandwidth Phase Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 52–62.

[2] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. Seattle, WA: USENIX Association, Apr. 2014, pp. 331–343. [Online]. Available: https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/li

[3] C. Yang and H. r. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, March 2015.

[4] P. Lazik and A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: ACM, 2012, pp. 99–112. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426667

[5] N. U. Hassan, A. Naeem, M. A. Pasha, T. Jadoon, and C. Yuen, "Indoor positioning using visible led lights: A survey," *ACM Comput. Surv.*, vol. 48, no. 2, pp. 20:1–20:32, Nov. 2015. [Online]. Available: http://doi.acm.org/10.1145/2835376

[6] J. Racko, P. Brida, A. Perttula, J. Parviainen, and J. Collin, "Pedestrian dead reckoning with particle filter for handheld smartphone," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2016, pp. 1–7.

[7] Q. Tian, Z. Salcic, K. I. K. Wang, and Y. Pan, "An enhanced pedestrian dead reckoning approach for pedestrian tracking using smartphones," in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, April 2015, pp. 1–6.

[8] Y. Sun, Y. Zhao, and J. Schiller, "An autonomic indoor positioning application based on smartphone," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 3326–3331.

[9] A. Perttula, H. Leppkoski, M. Kirkko-Jaakkola, P. Davidson, J. Collin, and J. Takala, "Distributed indoor positioning system with inertial measurements and map matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2682–2695, Nov 2014.

[10] J. Ruppelt, N. Kronenwett, and G. F. Trommer, "A novel finite state machine based step detection technique for pedestrian navigation systems," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2015, pp. 1–7.

[11] Y. Moon, C. Luo, A. Shrivastava, and K. Palem, "Capsule: A camera-based positioning system using learning."

[12] G. Zhang, "A study on the abbe principle and abbe error," *CIRP Annals-Manufacturing Technology*, vol. 38, no. 1, pp. 525–528, 1989.

[13] R. W. Levi and T. Judd, "Dead reckoning navigational system using accelerometer to measure foot impacts," Dec. 10 1996, uS Patent 5,583,776.

[14] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on automatic control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[15] Y. Masumoto, "Global positioning system," May 11 1993, uS Patent 5,210,540.

[16] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit." in *ACL (System Demonstrations)*, 2014, pp. 55–60.

[17] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[18] S. S. Bhattacharyya, E. F. Deprettere, R. Leupers, and J. Takala, *Handbook of signal processing systems*. Springer Science & Business Media, 2013.

[19] K. Palem and A. Lingamneni, "Ten years of building broken chips: the physics and engineering of inexact computing," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 2s, p. 87, 2013.

[20] K. V. Palem, "Inexactness and a future of computing," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2018, 2014. [Online]. Available: http://rsta.royalsocietypublishing.org/content/372/2018/20130281

[21] J. Markoff, "Beyond silicon: Squeezing more out of chips," 2016. [Online]. Available: http://www.nytimes.com/2016/10/31/technology/beyond-silicon-squeezing-more-out-of-chips.html

[22] K. V. Palem, *Computational Proof as Experiment: Probabilistic Algorithms from a Thermodynamic Perspective*. Springer Berlin Heidelberg, 2003, pp. 524–547.

[23] ——, "Energy aware computing through probabilistic switching: A study of limits," *IEEE Trans. Comput.*, vol. 54, no. 9, 2005.

[24] E. K. Antonsson and R. W. Mann, "The frequency content of gait," *Journal of Biomechanics*, vol. 18, no. 1, pp. 39 – 47, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021929085900430

[25] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[26] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[27] M. Xu, P. Watanachaturaporn, P. K. Varshney, and M. K. Arora, "Decision tree regression for soft classification of remote sensing data," *Remote Sensing of Environment*, vol. 97, no. 3, pp. 322–336, 2005.

[28] D. Yuan, X. Ma, Y. Liu, S. Yan, and C. Hao, "Statistical modeling of additive noise and random drift for triaxial rate gyros and accelerometers," in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, June 2016, pp. 308–313.

[29] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.