Finding a Parked Car Location in a Multi-Storey Building Without GPS Service

https://doi.org/10.3991/ijim.v14i10.14385

Heba Fathy (✉), Reda Elbasiony, Mohamed Talaat Faheem
Tanta University, Tanta, Egypt
heba.fathi@f-eng.tanta.edu.eg

Abstract—This paper introduces a new approach named 'Find My Parked Car' to provide the user with the return way for his parked car location in the indoor garage of a multi-storey building where GPS is denied using the acquisition data obtained from the inertial sensors on smart devices such as smartphones. We build up a system that employs Pedestrian Dead-Reckoning Technique (PDR) to track the user inside the building. This approach harness the environment landmarks for floors separation and the PDR paths' landmarks to enhance them by applying the Kalman Filter. Then we prunes the paths to remove the redundant parts of paths. Finally, we display the paths that guide the user back to his car via an android application. The feature of our approach is to help users returning back to parked car location based only on the available ubiquitous sensors in the new smartphones without any need to have any map or any additional infrastructure support.

Keywords—Ubiquitous sensors; Pedestrian dead-reckoning; Kalman filter, GPS.

1 Introduction

Recently, the trend of building indoor garages with multi-storey buildings has been established to save much more parking locations for expected increasing number of cars. It is noted that the user spends more time inside the building wandering from one place to another, he needs for some sign to know where he parked his car inside the indoor garage of the building. To overcome this difficulty of remembering parked location, he needs to use some new means to help him getting his car back.

Google maps added a new feature called 'Save your parking' that enables the user to register his parking location in the open parking areas because it depends on its GPS for locating the parked car. In case of a user has parked his car inside the indoor garage where GPS signal is not available, and in case it's the first time that user visit this building, it is not easy to obtain the assigned car's location.

Our work, what we call Find My Parked Car system that leverage the ubiquitous sensors which are available in the new smartphones in order to provide the user with the return way to his car location in the indoor garage of a multi-storey building.

The main idea is to simulate the work of GPS work but in the indoor buildings. Our Find My Parked Car system consists of a mobile application that collects the readings of the sensors in the user's phone from the moment the application is activated by user or the application detects that the car was parked in parking the garage to the moment the user requests the application to guide him to his car. Through the sensor readings, the forward trajectories are configured using Pedestrian Dead-Reckoning (PDR) and the landmarks are extracted that are divided into two types; the first type used to improve the forward trajectory generated by PDR (Trajectory enhancement landmarks) and the other type used to separate the floors so that the user can see the path he will follow for each floor separately (Floors separation landmarks). Then enhanced the PDR trajectories as they suffer from a lot of noise because they depend on a noisy sensor of smartphones by applying the Kalman Filter on the landmarks and PDR trajectories. After we have the enhanced forward trajectories, we conduct the pruning process to remove the trajectories' redundant parts to make the trajectories from the parked car location to the user current location directly not including paths inside shops and rooms or looping segments. Our system last step is to reverse all the forward trajectories (the trajectory for each floor) to constitute the whole trajectory in which the starting point is the user's current location and the destination is the parked car location inside the garage also reversing directions, angles and turns along the path and display them using an android mobile application.

Implementing this idea on the ground is faced with many challenges: (1) A highly deviated trajectories from the real trajectories created by Dead-Reckoning suffer from terrible impurities because they rely on inexpensive noisy mobile sensors. (2) As far as we know, there is still no way to find the accurate moving angle for the user through the mobile sensors. (3) To detect the locations of the landmarks fairly accurately to be used in separating the path of each floor and in enhancing the resulting path of Dead-Reckoning without prior knowledge of the building map. (4) Pruning the resulting path to provide the user with shortest paths over his forward paths to relieve him of passing by the unrequired places.

To evaluate our system, we implement it on different Android phones, and evaluate by 3 persons who follow our case scenario; first they parked their cars in university building indoor garage, then wandering inside the building and finally request our application to show them the return way to their parked cars. Our result show that can generate an accurate return trajectory from user's current location to his parked car with an error of 3.2m.

In summary, our main contribution is presenting "Find My Parked Car" system that provides the user with a way back to his car parked in the indoor garage of the building using only the available sensors in smartphones with no need to have any map or any additional infrastructure support.

The rest of the paper is organized as follows: Section 2 presents the related work. We give system details in sections 3. Section 4 provides the performance results for Find My Parked Car system. Finally, sections 5 conclude the paper.

2 Related Work

Many researchers focused their works on solving this problem. Parkloc [1] has integrated vehicular localization module with pedestrian navigation system. They focused on vehicular localization module which adopts GraphSLAM and consider well-known pedestrian localization schema (PlaceLab [2], Zee [3], and Unlock [4]).

The ubiquitous indoor localization proposals available almost classified into WiFi-based or dead reckoning based. WiFi-based techniques, e.g., [5–11], require calibration to create a prior wireless map for the building. While the calibration is a ponderous, time consuming, and requires recurring updates process; which led to the appearance of calibration-free techniques [12] but still there is a problem of the unavailability or extreme attenuation of WiFi signals in most indoor parking garages.

Dead-Reckoning based localization techniques, e.g., [13], [4], used the inertial sensors on smart phones which suffer from its severity affected by noise. Which leads to accumulation of error to deviate the resulting dead reckoning path from the real path at a high rate. There are many techniques that have been proposed to reset the dead-reckoning accumulated error using environment landmarks like elevator and stairs [14], [15].

Most of the previous techniques are still suffering from many problems. The WiFi-based indoor localization techniques have the problem of the WiFi signal unavailability and extremely attenuated in most indoor parking garages. And the Dead-Reckoning based indoor localization techniques produce a very deviant paths from the actual paths, as a result of their dependence on the distorted sensors which are embedded in smartphones.

Accordingly, we introduce a new vision to overcome these problems, Find My Parked Car system applies the pedestrian dead reckoning and landmarks concepts but in different ways as we will explain in the upcoming sections of paper, as it provides accurate, energy-efficient localization without prior information about the map of the building or requiring any additional infrastructure support.

3 Proposed System Overview and Architecture

Our problem is concerning with the difficulty for the user to return to his car's parking location in the indoor garages of multi-storey buildings as the nature of multi-storey buildings forced the user to go up and down and move between floors which leads to confusion of the user's mind wondering how many floors hove gone up?, how many floors have gone down?, in which floor the indoor garage starts? And on which floor of the garage I stopped my car? The problem is further complicated with the lack of GPS as it is an indoor area where GPS signal is not available in addition to the unavailability of the floor plan for most buildings. There are many mobile applications that enable the user to register his car's parking location but the question here is how to reach from the use's current location inside the building to the registered car's location which recorded by the mobile application?! And what if this the first visit of the user to the building?!

He has no idea of the roads inside the building, his car's location may be somewhere in front of him but he doesn't know how to get to this place.

Our system seeks to carry out the same task of GPS but inside buildings, as shown in Figure 1 that shows our system architecture. Our architecture consists of six modules:

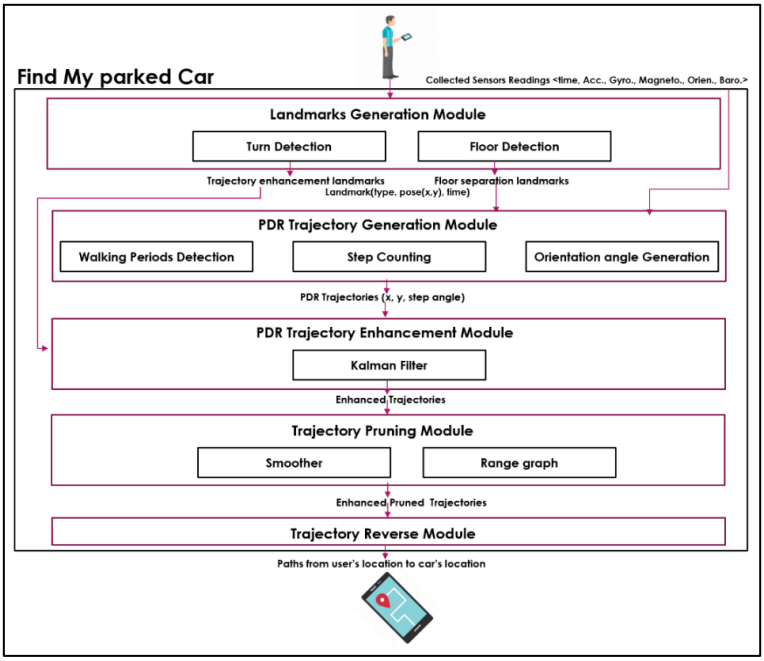

Fig. 1. Find My Parked Car System Architecture

3.1 Sensors data recording mobile application

As we seek to generate an energy-efficient tracking technique, we will depend only on the ubiquitous sensors which are available in the new smartphones as they don't consume any extra energy because they running all the time for mobile orientation detection. We implemented an android application that records the sensors reading; accelerometer, gyroscope, magnetometer, barometer, and orientation sensors that are already embedded in the new smartphones. The application starts recording once the car parked by one of two options; first option the recording will start manually by user when he taps on start recording button, or automatically by using on of this methods that detect that the car had parked [16], [17] to identify the parked car location as the final destination and start recording sensor readings. Once the user stops and requests the application to guide him to return back to his car through stop recording button, all the sensors readings are to be processed to generate the target return way by pressing on guide me to car button.

3.2 Landmarks generation module

In this module we leveraging the collected sensors reading to generate two types of landmarks.

Trajectory enhancement landmarks: We use gyroscope reading to generate landmarks that will be used to enhance the trajectory. We observe that when the user turns right in an angle, a negative difference occurs in the resulting angle of the gyro by the turn angle, and if he turns left, a positive difference occurs in the resulting angle of the gyro by the turn angle.

We monitored the angle resulting from the gyro along the user's path and when noticing a difference in the angle we record the time, location, and the difference occurred in gyro that represent the turn angle and we define it as a turn made by the user to be used in enhancing the trajectory as we will see in the trajectory enhancement section.

Floor separation landmarks: For floor level change detection we leveraged the barometer sensor which records the atmospheric pressure to separate the path for every floor and to tell the user in which elevation semantic he will transfer to next floor just to be a mark for the end of that floor and to assure the user that he is on the right track. For every 1.0m level change, there will be a 0.12 hPa difference in the atmospheric pressure; negative 0.12hPa difference for going up and positive 0.12hPa difference for going down. Often the typical floor height is 3.0m, so we considered a 0.36hPa threshold on the atmospheric pressure value to change floor level by one floor.

Once we detect this level change, we will record its time as the end of this floor trajectory and the start of the next floor trajectory. We also determine the elevation's means; elevators, escalators, half-landing stairs, and straight stairs by employing Elevation Change Semantics Detection in [18] that applied a decision tree classifier on the sensor's readings based on the observations of semantics usage scenarios and their physical structures.

As the elevator generates a distinctive pattern which can be easily distinguished that consists of walking period, waiting for the elevator, walking inside the elevator, changing direction to face the elevator door in case of single door elevator while in the two-door elevators users don't have to change their direction to face the elevator door, standing for a while, followed by a level change up or down when it starts to move. We use the accelerometer, gyroscope and barometer sensors readings to detect the elevator pattern. All states must be detected to consider the elevation semantic is an elevator.

If the elevator is excluded, we will proceed to the escalators (Standing) test in which the variance of the acceleration signal will still small unlike climbing stairs or escalators. If the variance of the acceleration is high, the user either is climbing a stair or an escalator due to his vertical motion.

If the escalators (Standing) is excluded, we will proceed to the Half-landing Stairs (Climbing) test in which there is a sudden rush in gyroscope readings due the direction change in the middle of the elevation change period because the half-landing staircases have a turn in the middle that force the user to change his direction.

If there is no direction change, it's either the user is climbing a straight stair or climbing an escalator. The difference between them can be only created from the magnetic field variance because the escalator constant speed motors that resulting a high

magnetic field variance compared to climbing a straight stair. If all other elevation change semantics are excluded, we can consider the elevation semantic is straight stairs.

3.3 PDR trajectory generation module

In this section our target was to track the user's path from the moment he parked his car until he requests the application to return him back to his car generating the trajectory as (x,y, theta) using the pedestrian dead-reckoning algorithm (PDR) technique [19] which calculate the user's current position from his previous Known position as equations:

$$Xnew = Xold + SL * \cos(theta) \quad (1)$$

$$Ynew = Yold + SL * \sin(theta) \quad (2)$$

Where Xnew, Ynew are the coordinates of the user's current position, Xold, Yold are the coordinates of the user's previous position, SL is the user stride length which represents the distance covered in the step, and theta is the orientation of the step taken by user.

So, to generate this trajectory, we followed the next steps.

Walking periods detection: First we needed to determine the walking periods along the whole path, the [20] provides a survey of all walking detection algorithms that have been previously submitted. We employ the walking detection algorithm in [3] that distinguishes the user is walking or not by using combination of both standard deviation in the magnitude of acceleration $\sigma||a||$ and the maximum normalized auto-correlation Ψ depending on that the standard deviation in the magnitude of acceleration would be a good indicator of any movement as when the user is idle, the smartphone's accelerometer sensor will register little acceleration and the maximum normalized auto-correlation to ascertain that the user is actually walking or idle with a large acceleration due to a sudden movements made by him depending on the repetitive pattern arises on the acceleration signal because of the rhythmic nature of walking with a sequence of two steps (one left and one right) constituting the period. Zee summarized the transition between walking and stationary states as follow:

Algorithm 1 Walking Detection Algorithm

```
for all Acceleration reading a do
     if  std (||a||) < 0.01 then
          state ← "STATIONARY"
     else if  Ψ (a) > 0.7 then
          state ← "WALKING"
     else
         state ← "NO CHANGE"
      end if
```

Step counting: After we determined the walking and stationary periods, we will count the steps in all walking periods to calculate the position coordinate resulting from each step. We employ the step detection algorithm in [21] that built a FSM-based step detection algorithm based on the magnitude of acceleration as its signal goes to a positive peak value followed by a negative peak value during each step taking advantage of the effect of walking on the magnitude of acceleration is independent from the phone orientation. UPTIME use four thresholds to ensure that the magnitude of acceleration signal forms a positive peak value followed by a negative peak value completely that forming the distinctive pattern for a step. We executed this algorithm on the whole magnitude of acceleration along the forward trajectories generating the total number of steps taken by user and time of each step.

Orientation angle generation: We assumed that the user holding his smartphone in his hand. To get the orientation angle for each step we leveraged the orientation sensor reading from the new android smartphones. We noticed from experiment that when we used the orientation angle generated from the orientation sensor as the step angle this resulted in a large deviation in the direction of the trajectory, and it reached to reverse the trajectory at an angle of 180˚ due its noisy nature. So, we performed the next steps:

- We get the sin and cosine values for the orientation angle resulted from orientation sensor.
- We applied moving average filter with 15 window size on the sine and cosine values resulting from pervious step to remove the noise from the orientation sensor.
- Then we computed the atan2 to the sine and cosine after the filter in order to get the angles values confined between -180˚ to 180˚.
- Finally, we added a 10˚ offset difference angle to the pervious angle generated from the atan2 step because by experiment we found that there is a difference between the true north and the north generated by the orientation sensor about 10˚. The resulting angles considered as the orientation angles for each step taken by the user.

Stride length calculation: Stride Length represents the distance covered by user within the step. There are researches that used a constant value for the average step size 70.74m, while others have proposed different approaches for stride length estimation like [21], [22]. In our experiment we pinpointed the steps on the ground truth trajectory according to the average step size for our participants as we will explain in system implementation section. But in our future work we will employ stride length estimation technique because the stride length changes from user to user according to the physical characteristics and changes for the same user according to the type and speed of gait.

PDR trajectories building: Finally, we can apply the equations (1), (2) to get all coordinates of steps taken by user in forward direction configuring the PDR trajectories as we had the start and the end for each trajectory of a specific floor from the Floor Separation Landmarks section. Then we will use these trajectories as inputs for next module PDR Trajectory Enhancement Module (Kalman Filter).

3.4 PDR trajectory enhancement module (Kalman filter)

The PDR will provide a very smooth estimate of the user's position, but it will drift over time as small errors accumulate. As Kalman filter deals effectively with the uncertainty due to noisy sensor data, so our target is to enhance the noisy PDR Trajectories using Kalman Filter [23] with the aid of trajectory enhancement landmarks and that is the turns taken by user along the PDR Trajectory obtained from Trajectory Enhancement Landmarks Module.

We considered that each pose on the trajectory has two landmarks; the turn before that pose and the turn after it and in cases where there are no turns before or after the pose, will have a single landmark. In our system The Kalman filter considers the PDR Laws of motion are the system's dynamic model, the distance traveled within the step (e.g. user's stride length) and the orientation step angle are the control inputs to that system, and the distance from the pose to its corresponding landmarks and the angle between them are the multiple sequential measurements to produce an estimate of the state of the system as an average of the system's predicted state and of the new measurement using a weighted average that is better than the estimate obtained by using only one PDR alone, with more weight being given to estimates with higher certainty. The weights are calculated from the covariance, a measure of the estimated uncertainty of the prediction of the system's state. The result of the weighted average is a new state estimate that lies between the predicted and measured state, and has a better estimated uncertainty than either alone. This process is repeated at every time step detected, with the new estimate and its covariance informing the prediction used in the following iteration using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

The Kalman Filter works in two distinct phases: predict and update. In the prediction phase, the user's old position will be modified according to the PDR laws of motion. Not only will a new position estimate be calculated, but also a new covariance will be calculated as well. Next, in the update phase, a measurement of the user's position is calculated from the distance between the position and its corresponding landmarks and the angle between them. Along with this measurement comes some amount of uncertainty, and its covariance relative to that of the prediction from the previous phase determines how much the new measurement will affect the updated prediction. As the PDR tend to drift away from the real position, the landmarks measurement should pull the position estimate back towards the real position but not disturb it to the point of becoming noisy and rapidly jumping.

3.5 Trajectory pruning module

After we got the user's enhanced forward trajectories from the previous module, now we want to shorten these trajectories, meaning that we want to truncate the unwanted trajectory segments, such as those where the user wanders inside the shops, in case the building was a shopping mall or any looping segments in order to provide the user with his way back to car direct with no need to enter shops or take loops again. The Trajectory Pruning Module consists of two steps:

- **Smoother:** In this phase starting from the floor enhanced trajectory start point and for each point by comparing each point with its predecessor, we will consider the point that shorten the trajectory to the end point while neglecting the points that extend this trajectory in order to prevent points from going back.
- **Range graph:** Now we will make a graph for each point on the smoothed floor enhanced trajectory with all the following points within a certain range which we chose it 5 meters, and then we choose from each graph closest point to the end point of that trajectory in order to prevent loops forming.

3.6 Trajectory reverse module

Last step in our system is to show the final paths in a reverse direction trough the android application including the total number of steps, directions and total floors. The android application will display the path of the current floor indicating its start by the user's current location and its end by the elevation mean that will take him to the next floor whether it's up or down and so on until he reaches to the indoor garage floor in which his car is parked.

4 Experimental Results

Find My Parked Car is implemented on Google Nexus 5 and Samsung A5 (2017) smartphones using JAVA as the programming platform. Find My Parked Car collects 5 sensors readings; accelerometer, gyroscope, magnetometer, barometer, and orientation sensors at sample rate 50Hz, then these readings sent to Find My Parked Car server which is written using JAVA and Python and implements landmarks detection, PDR, PDR enhancement, and pruning algorithms but In a real- life deployment of Find My Parked Car isn't mandatory since our all algorithms can be executed on the phone.

We design real-life experiments with 3 different users in Faculty of Engineering Tanta university building 3. We covered 2500m² approximately in that building. Each user walked around in the building carried 2 phones in their hand with the screen facing up. We created multiple ground truth trajectories covering multiple floors by putting phosphorous adhesives for each pose on the trajectory to be followed by user. We put the adhesives at predefined distances to enable us to create the ground truth trajectory to be compared with our resultant trajectory on the building map to find the error rate. We used the building map only for testing purpose but in a real- life deployment of Find My Parked Car we assume that the user does not have a prior knowledge about the building map and this is what gives an advantage to Find My Parked Car.

The following figures show what we observe during our experiments. Figure 2 shows the angles resulted from the gyroscope readings over the time, which are observed to produce a negative difference when the user turns right and this difference is approximately equal to the turn angle and vice versa. In Figure 2, the part enclosed in the red rectangle shows that the user turned right approximately at an angle of 90 degrees, and the part surrounded by the green rectangle shows that the user turned left at

an angle of 90 degrees. These turns points used as landmarks to enhance the PDR trajectories by means of the Kalman filter.

Figure 3 shows barometer readings over the time, which are observed to produce a 0.36hPa negative difference when the user goes one floor up. When this difference arises, we record this point as the transition from one floor to another, which we use to separate the trajectory of each floor from another.

Figure 4 shows the readings from the gyroscope, the accelerometer and the barometer over the time which used to detect the pattern generated from using a two-door elevator that consists of normal walking period, waiting for the elevator, walking inside the elevator, standing for a while, followed by a level change up or down when it starts to move. Once all states detected it's considered that the elevation semantic is an elevator.

Figure 5 shows the steps taken to perform the pruning process. Firstly, we conduct the smoother phase to only consider the points that shorten the path to the destination dismissing other points. Then we formed a graph for each point on the smoothed trajectory with all its following points within a 5 meters range. Finally, to cut any loop in the trajectory, we choose from each graph closest point to the end point of that trajectory.

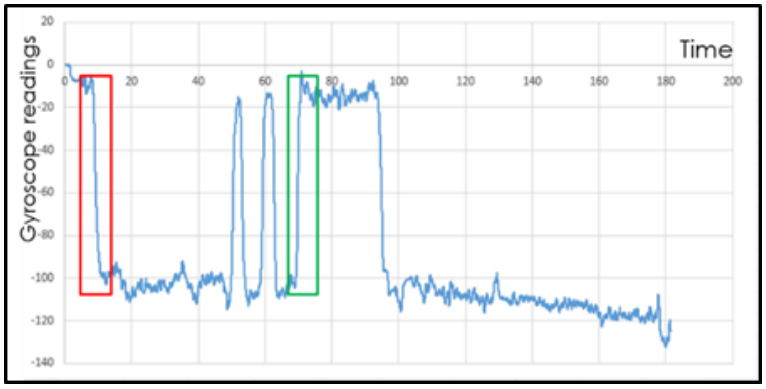

Fig. 2. Turns effect in gyroscope readings, red rectangle exhibit about 90˚ negative difference because the user turns right with angle 90˚ counter to the green rectangle that shows the turn left effect because the difference is positive.

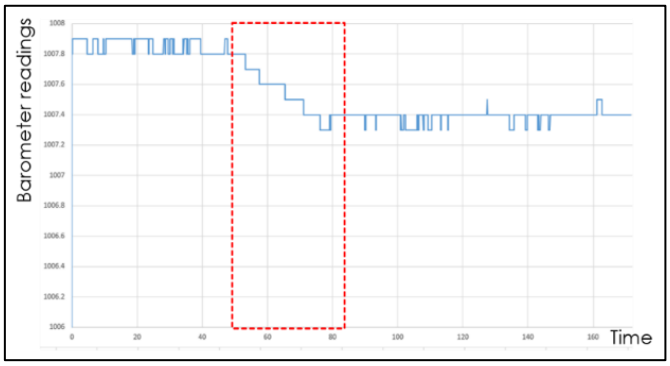

Fig. 3. Barometer difference 0.36hPa, negative difference for going one floor up.

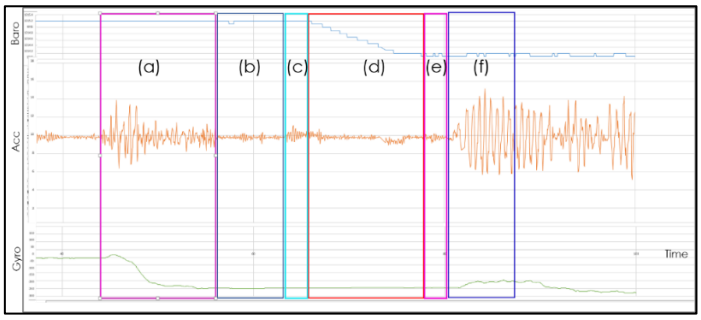

Fig. 4. Elevator pattern for a two-door elevator: (a) walking period, (b) stationary period waiting for elevator coming, (c) walking into elevator, (d) level change up; barometer reading changed and the acceleration reading shows an accelerate-stationary-decelerate emerging from the start and stop of the elevator, (e) stopping, and (f) walking out from the elevator

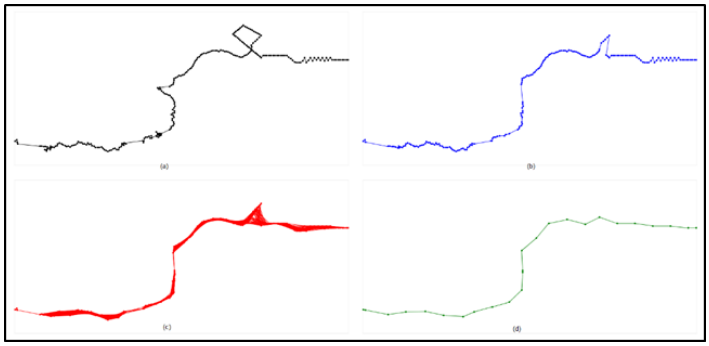

Fig. 5. Illustration of our Trajectory Pruning Algorithm: (a) trajectory before pruning, (b) smoothing phase for considering the points that shorten the path to the destination, (c) range graph formation for each point and its corresponds points which are 5 meter away from it, (d) trajectory after pruning not contains any loops.

We present our system accuracy via showing the comparison between the ground truth path, PDR path, enhanced path and the pruned path. Figure 6 shows a segment of a user's trace in the first floor inside a building with dimensions 30m×166m. The sold line shows the ground truth path that we defined for our experiment, the dotted line shows the PDR path, the dashed line shows the enhanced PDR path (Kalman filter path) and the dash-dot line shows the pruned path. We calculated the accuracy as the average Euclidean distance between the estimated position [24] and the actual position. So, the resulting distance error between the ground truth path and the enhanced path was 3.2 meters. And by visual evaluation we found that the resulting path leads the user to his car accurately as target was to guide the user to his parked car through reversing his forward tracked path inside the building after removing the unwanted parts of his path, so if there is a simple difference between the original and resulting path, it will be accepted as long as the resulting path finally guide the user to his parked car. Figure 7 shows how the curves are correlated together which proves that our goal has been achieved.

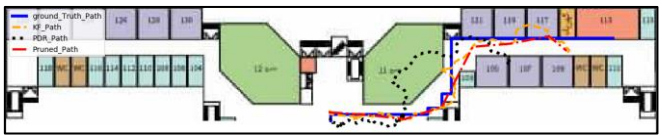

Fig. 6. Experiment trace of the first floor inside the university building.

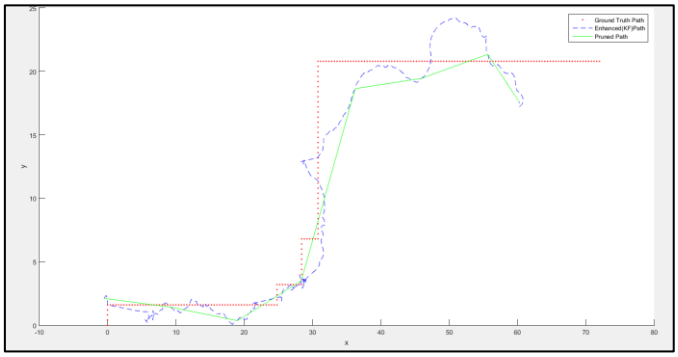

Fig. 7. The correlation between the ground truth path, enhanced path, and the pruned final path.

5 Conclusion

We presented the Find My Parked Car system for providing the user with the return way for his parked car location in the indoor garage of a multi-storey building where GPS service is denied via tracking his path by Pedestrian Dead-Reckoning technique based on leveraging the smartphones' inertial sensors for its energy efficiency on enhancing the tracking trajectories by applying Kalman filter, then pruning that

trajectories to get the shorten paths to the car which is displayed via an android application. We implemented Find My Parked Car using different smartphones that run the Android operating system and evaluated it at our university building. Our result shows that Find My Parked Car can generate an accurate return trajectory from user's location to his parked car with an error of 3.2m without having any map or any additional infrastructure support. Finally, Find My Parked Car can be generalized over various buildings running by different users guiding them to different objects not only their parked cars.

Currently, we are expanding our system in different directions including covering of other cases of mobile phones, such as being in the user's pocket or being in his bag, increasing the accuracy of locating the landmarks, and generating an accurate online tracking module to track the user on his way back to make sure that he is following the same path that is supposed to be followed to reach his car and if he deviates from the return path, it will supply him how to return to the return path again.

6 Acknowledgement

We sincerely pray to God to have mercy on Dr. Mustafa ELhamshary and to reward him with good for his participation in the beginning of work on this paper.

7 References

[1] J. Cherian, J. Luo, S. S. Ho (2018), "ParkLoc: Light-weight Graph-based Vehicular Localization in Parking Garages", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(3), 1-23. https://doi.org/10.1145/3264909

[2] Y. C. Cheng, Y. Chawathe, A. LaMarca, J. Krumm (2005), "Accuracy characterization for metropolitan-scale Wi-Fi localization", In Proceedings of the 3rd international conference on Mobile systems, applications, and services (pp. 233-245). https://doi.org/10.1145/1067170.1067195

[3] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, R. Sen (2012)," Zee: Zero-effort crowdsourcing for indoor localization", In Proceedings of the 18th annual international conference on Mobile computing and networking (pp. 293-304). https://doi.org/10.1145/2348543.2348580

[4] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, R. R. Choudhury (2012), "No need to war-drive: Unsupervised indoor localization", In Proceedings of the 10th international conference on Mobile systems, applications, and services (pp. 197-210). https://doi.org/10.1145/2307636.2307655

[5] K. El-Kafrawy, M. Youssef, A. El-Keyi, A. Naguib (2010)," Propagation modeling for accurate indoor WLAN RSS-based localization", In 2010 IEEE 72nd Vehicular Technology Conference-Fall (pp. 1-5). IEEE. https://doi.org/10.1109/vetecf.2010.5594108

[6] A. Eleryan, M. Elsabagh, M. Youssef (2011), "Synthetic generation of radio maps for device-free passive localization", In 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011 (pp. 1-5). IEEE. https://doi.org/10.1109/glocom.2011.6134052

[7] M. Ibrahim, M. Youssef (2011), "A hidden markov model for localization using low-end GSM cell phones", In 2011 IEEE International Conference on Communications (ICC) (pp. 1-5). IEEE. https://doi.org/10.1109/icc.2011.5962993

[8] M. Youssef, M. Abdallah, A. Agrawala (2005), "Multivariate analysis for probabilistic WLAN location determination systems", In the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (pp. 353-362). IEEE. https://doi.org/10.1109/mobiquitous.2005.41

[9] M. Youssef, A. Agrawala (2005), "The Horus WLAN location determination system", In Proceedings of the 3rd international conference on Mobile systems, applications, and services (pp. 205-218). https://doi.org/10.1145/1067170.1067193

[10] M. Youssef, A. Agrawala (2006), "Location-clustering techniques for WLAN location determination systems", International Journal of Computers and Applications, 28(3), 278-284. https://doi.org/10.1080/1206212x.2006.11441813

[11] M. Youssef, A. Agrawala (2008), "The Horus location determination system", Wireless Networks, 14(3), 357-374. https://doi.org/10.1007/s11276-006-0725-7

[12] R. Elbakly, M. Youssef (2016), "A robust zero-calibration RF-based localization system for realistic environments", In 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) (pp. 1-9). IEEE. https://doi.org/10.1109/sahcn.2016.7732967

[13] Y. Jin, H. S. Toh, W. S. Soh, W. C. Wong (2011), "A robust dead-reckoning pedestrian tracking system with low cost sensors", In 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom) (pp. 222-230). IEEE. https://doi.org/10.1109/percom.2011.5767590

[14] H. Abdelnasser R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, M. Youssef (2015), "SemanticSLAM: Using environment landmarks for unsupervised indoor localization", IEEE Transactions on Mobile Computing, 15(7), 1770-1782. https://doi.org/10.1109/tmc.2015.2478451

[15] H. Aly, A. Basalamah, M. M. Youssef, "A crowd-sensing system for automatic map semantics identification", In Proceedings of the Eleventh IEEE International Conference on Sensing, Communication, and Networking (SECON), Singapore (Vol. 30). https://doi.org/10.1109/sahcn.2014.6990340

[16] J. Cherian, J. Luo, H. Guo, S. S. Ho, R. Wisbrun (2016), "Parkgauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics", In 2016 17th IEEE International Conference on Mobile Data Management (MDM) (Vol. 1, pp. 92-101). IEEE. https://doi.org/10.1109/mdm.2016.26

[17] P. E. Carnelli, J. Yeh, M. Sooriyabandara, A. Khan (2017), "Parkus: A novel vehicle parking detection system", In Twenty-Ninth IAAI Conference.

[18] M. Elhamshary, M. Youssef, A. Uchiyama, H. Yamaguchi, T. Higashino (2016), "TransitLabel: A crowd-sensing system for automatic labeling of transit stations semantics", In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (pp. 193-206). https://doi.org/10.1145/2906388.2906395

[19] L. Ojeda, J. Borenstein (2007), "Personal dead-reckoning system for GPS-denied environments", In 2007 IEEE International Workshop on Safety, Security and Rescue Robotics (pp. 1-6). IEEE. https://doi.org/10.1109/ssrr.2007.4381271

[20] A. Brajdic, R. Harle (2013), "Walk detection and step counting on unconstrained smartphones", In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (pp. 225-234). https://doi.org/10.1145/2493432.2493449

[21] M. Alzantot, M. Youssef (2012), "UPTIME: Ubiquitous pedestrian tracking using mobile phones", In 2012 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 3204-3209). IEEE. https://doi.org/10.1109/wcnc.2012.6214359

[22] V. Renaudin, M. Susi, G. Lachapelle (2012), "Step length estimation using handheld inertial sensors", Sensors, 12(7), 8507-8525. https://doi.org/10.3390/s120708507

[23] S. Thrun, W. Burgard, D. Fox (2006), "Probalistic robotics", Kybernetes.
[24] H. Liu, H. Darabi, P. Banerjee, J. Liu (2007), "Survey of wireless indoor positioning techniques and systems", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37(6), 1067-1080. https://doi.org/10.1109/tsmcc.2007.905750

8 Authors

Heba Fathy is an Egyptian Computer Engineering Master's degree student and a Teaching Assistant at Computer & Automatic Control Engineering Department - Faculty of Engineering - Tanta University - Tanta - Egypt.

Reda Elbasiony is an Egyptian computer scientist, engineering educator, and an Assistant Professor at Computer & Automatic Control Engineering Department - Faculty of Engineering - Tanta University - Tanta - Egypt.

Mohamed Talaat Faheem is an Egyptian computer scientist, engineering educator, and a Professor Emeritus at Computer & Automatic Control Engineering Department - Faculty of Engineering - Tanta University - Tanta - Egypt.

Article submitted 2020-03-24. Resubmitted 2020-04-27. Final acceptance 2020-04-28. Final version published as submitted by the authors.