Floor Number Detection for Smartphone-based Pedestrian Dead Reckoning Applications

1st Cedric De Cock
Dept. of Information Technology
imec-WAVES/Ghent University
Ghent, Belgium
cedric.decock@ugent.be

2nd Wout Joseph
Dept. of Information Technology
imec-WAVES/Ghent University
Ghent, Belgium
wout.joseph@ugent.be

3rd Luc Martens
Dept. of Information Technology
imec-WAVES/Ghent University
Ghent, Belgium
luc1.martens@ugent.be

4th David Plets
Dept. of Information Technology
imec-WAVES/Ghent University
Ghent, Belgium
david.plets@ugent.be

***Abstract*—We present a new floor number detection algorithm for use in smartphone-based indoor localisation systems. It is designed to complement any pedestrian dead reckoning (PDR) algorithm able to detect steps and estimate a 2D trajectory from data of the smartphone's inertial measurement unit.**

Our proposed method is based on the Viterbi algorithm, fusing data from an off-the-shelf smartphone's accelerometer, barometer and wifi received signal strength (RSS) measurements. The accelerometer is used to detect accelerating elevators, while the barometer is used to detect stair climbing. This is combined with model-based wifi RSS fingerprinting, enabling accurate floor number detection. Our system is tested in an office environment with 7 41 m x 27 m floors, each of which has 2 pre-existing wifi access points. The algorithm is evaluated with a total of 116 minutes of recorded data, in which the floor number changed 76 times and a distance of 4.8 km was travelled. Since the Viterbi algorithm allows to easily correct past states (i.e. floor numbers) based on new information, it is evaluated in real-time and batch mode. Our proposed algorithm achieves a floor number detection accuracy of 99.1% (real-time) and 99.7% (batch), while using only RSS measurements resulted in 91% accuracy.

***Index Terms*—indoor localisation, fingerprinting, wifi, received signal strength, barometer, accelerometer, Viterbi, activity recognition**

I. Introduction

While satellite navigation (e.g. GPS) is the standard for outdoor localisation, signal blockage by buildings makes these systems unusable indoors. Dedicated indoor localisation systems have been developed in recent years [1], [2]. These systems make use of wireless technologies, such as ultra-wideband (UWB) [3], [4], visible light communication [5], [6] or wifi [7], [8]. Generally, a human or object is equipped with a tag that is localized relative to a fixed infrastructure of anchor nodes. A drawback is dependency on the anchor nodes [2], which are expensive, require manual work to setup and their placement and quantity affects the accuracy. Furthermore, these systems have to deal with challenges which depend on the environment, e.g. non-line-of-sight (NLOS), multipath, human body effects, ...

Alternatively, Inertial Navigation Systems (INS) [9] use inertial measurement units (IMU) to enable localisation without any dedicated infrastructure. However, an INS does not provide absolute positioning and is prone to drift. Hybrid systems fuse both methods which combines their advantages [10]. Most smartphones are equipped with an IMU and can be used for Pedestrian Dead Reckoning (PDR). In this method, data of the IMU's sensors (i.e. accelerometers, gyroscopes and/or magnetometers) are fused to detect steps and estimate their length and heading. While dedicated IMU's strapped to the leg provide higher accuracy [11], smartphone-based PDR requires no additional hardware and offers good user comfort since people already carry their phone with them. Smartphones are also equipped with WiFi and Bluetooth chipsets, which allows fusion with WiFi/Bluetooth localisation systems [10], [12], [13].

A challenge in smartphone-based PDR is the extension to multi-floor localisation. One approach is using WiFi fingerprinting, but the detection accuracy depends on the environment and the amount of APs [14]. Another approach is to detect the floor transitions and estimate the height change using the on-board sensors. Since the pedestrian is either on a staircase or in an elevator during a floor transition, detecting these transition intervals is also useful for improving 2D localisation [15]. Classifiers have been designed to detect floor transitions in accelerometer and gyroscope data [16], [17]. Also, the pressure measured by the barometer can be converted to height [18]. [19], [20] use the first pressure measurement as a reference to estimate the height difference during the rest of the trajectory. However, these methods require the initial floor to be known and are prone to drift caused by false detections or changing pressure due to the weather, opening doors/windows, ... [20], [21]. Fusion of both approaches allows absolute floor detection and accurate floor transition detection [12], [22], [23]. In

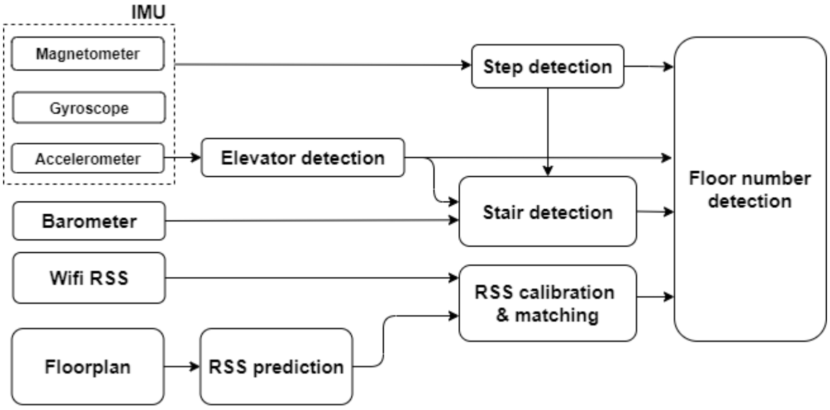

Fig. 1. High level flowgraph of the floor number detection algorithm. This algorithm consists of three phases. First, stairs and elevator usage are detected and the height change is estimated. Then, RSS measurements are matched with the model-based radiomaps and finally, the output of the first 2 phases is combined to estimate the floor number.

[23], a probabilistic model decides if recent pressure change was caused by a floor change. Therefore, a floor number change was only detected when the transition was (almost) finished. Reference [22] used a moving average to detect floor transitions. However, the floor number is detected with a pressure look-up table, which needs frequent recalibration during which the user has to stand still. Reference [12] used WiFi and Bluetooth fingerprinting and detected floor transitions with the barometer and gyroscope. However, their method depends on the placement of many APs (i.e., 42 beacons). Lastly, a reference barometer at a known floor is used in [21], [24] to compensate for pressure drift.

Because many systems depend on prior knowledge, reference devices, dedicated infrastructure and/or do not incorporate elevator usage, we have designed and implemented an accurate smartphone-based floor detection algorithm, which detects both stairs and elevator usage and corresponding height change. It is based on the Viterbi algorithm and does not need additional/dedicated hardware. The only requirement is that some wifi APs are present, which is the case for public buildings and office environments.

II. Method

Figure 1 shows a flowgraph of the proposed floor number detection algorithm. This three-phase algorithm performs floor number detection with model-based fingerprinting (MBF) of wifi RSS and floor transition detection with accelerometer and barometer data. In the first phase, the accelerometer and barometer data are used to detect elevator and stair usage. In the second phase, available RSS values from the wifi APs are matched to the model-based fingerprint databases of each floor separately, using the Euclidean distance metric. In the third phase an algorithm based on the Viterbi-algorithm [25] is proposed that fuses the output of the previous phases to estimate the most likely floor number. The algorithm's purpose is to complement PDR algorithms, thus an output is calculated after each step detection. The step detection algorithm used for this paper is reproduced from [26]. In real-time mode, this algorithm estimates the most likely floor based on all previous transition detections and fingerprint matches. In batch mode, it incorporates all detections and matches of the trajectory to estimate the most likely sequence of floors at once by backtracking with a Trellis diagram. Lastly, both the step detection and proposed algorithms are independent of the orientation and carrying mode of the smartphone.

A. Elevator detection with accelerometer

The elevator detection algorithm is based on the elevator acceleration sensing principle of [27].

In [27], a 1 second wide 'window' is moved over the acceleration norm after filtering out the DC-component (i.e. gravity). An accelerating elevator is detected if the values within the window are all positive (*hill*) or all negative (*valley*) and the absolute values lie within a specified interval. A valley following a hill means a rising elevator and vice versa. Our algorithm adds a low-pass (LP) filter with cut-off frequency of 1.5 Hz to the acceleration norm to remove high frequency noise and reduce the influence of hand trembling. A maximum time offset between a candidate hill and valley is set by measuring the time the elevator needs to traverse all floors. In case of an elevator detection, the latest step is labeled *elevator* (algorithm 1). The height change can be estimated by double integration of acceleration, but this is not reliable [27]. Instead, the floor change n is estimated with eq. 1.

$$n = \frac{v_{max} * (T_{total}^{elev} - T_{acce}^{elev})}{h_{floor}} \tag{1}$$

T_{total}^{elev} is the time interval of the whole elevator transition and h_{floor} the estimated floor height. We estimated the maximum speed v_{max} and acceleration/deceleration time interval of the elevator T_{acce}^{elev} by averaging these two parameters from detected elevator transitions in the training data.

B. Stair detection with barometer

The pressure is first converted to a height change [18], but no initial height is assumed (i.e. initial height is $0m$). Stair transitions are detected by thresholding the height change within a long window of W_l and a short window of $W_s < W_l$ past step detections. Slow height changes (long window) indicate stair usage, while fast height changes (short window) indicate either elevator usage or noise. The latter can be caused by opening/closing doors or windows [21].

The current step is labeled *noise* when fast height change is detected and none of the steps in the short window have been labeled *elevator* (II-A). The current step is labeled *stairs* when slow height change is detected and none of the steps in the long window are labeled *elevator* or as *noise*. This prevents simultaneous elevator and stair detections, and false stair or elevator detections due to sudden pressure changes not caused by a floor transition. If the stair transition is initially detected, the height difference over the whole window is estimated. If there was a stair detection during one or more steps of the detection window, the height change until the previous stair detection is added to the change between the previous and current stair detection.

The algorithm uses three parameters, i.e. the length of the long window W_l and short window W_s and the height change threshold th_h. th_h and W_l are adapted to the device, because different devices produces different amounts of noise. Assuming the barometer sensor noise is Gaussian and the height measurements (converted from pressure measurements) have a standard deviation σ_h, height difference is also Gaussian: $N(\Delta h, \sigma^2_{\Delta h})$ with Δh the real height difference and $\sigma_{\Delta h} = \sqrt{2}\sigma_h$. We impose $|\Delta h| = 3\sigma_{\Delta h}$ and assuming a stair rise of 0.15m [28], the stair detection window should be at least $20\sigma_{\Delta h}$ steps long. We choose $th_h = \sigma_{\Delta h}$, which makes false positives more likely than false negatives. However, a false positive stair detection is not as bad as a false negative in this context, because the detected height change must be at least 50% of the height between two floors to trigger a floor number change.

C. Model-based fingerprinting

RSS measurements since the previous step are used as one RSS vector when a new step is detected. The RSS values are calibrated and then compared to model-based radiomaps with eq. 2.

$$\widetilde{d}^{t,i}_{RSS} = \frac{d^{t,i}_{RSS}}{\sum_{j=0}^{n-1} d^{t,j}_{RSS}} \tag{2}$$

$d^{t,j}_{RSS}$ is the smallest Euclidean distance between the RSS vector of the t-th step and the fingerprints of the j-th of n floors. A seperate KD-tree of the radiomaps for each floor is used to quickly find the smallest distance. Only radiomaps for pre-existing wifi APs are used, because the intention is to be independent of a dedicated localisation infrastructure.

The radiomaps are made with the WiCA Heuristic Indoor Propagation Prediction (WHIPP) tool [29]. The user can draw the floorplan and enter the building materials of each wall as well as the location and other parameters of the wifi AP. The path loss (PL) for each AP is estimated for each coordinate on a grid, by a PL model which incorporates the distance to the AP and the cumulated losses due to wall penetrations and direction changes of the signal [30].

The RSS values are finally calculated by subtracting the PL (in dB) from the transmitted power P. P also accounts for the antenna gains of the AP and smartphone, which are often not known exactly. This causes a bias of several dB between the predicted and the real RSS values. The bias is compensated with self-calibration per AP. This method estimates the bias by mapping the cumulative distribution functions (CDFs) of measured RSS values to CDFs of estimated RSS values and is fully explained in [31].

D. Viterbi-based floor number detection

Finally, we propose a Viterbi based algorithm to combine the detected stairs and elevator transitions with wifi RSS-MBF, to enable more accurate floor number detection. The original Viterbi algorithm [32] uses transition and emission probability matrices to calculate the probability of transitioning to a certain state at a given time and finds the sequence with highest probability. Contrary, our algorithm uses a cost function (eq. 3) to calculate the cost of transitioning between two states, i.e. two floors, and finds the sequence with lowest cost.

$$\begin{aligned} C(S^i_t|S^j_{t-1}) = {} & C(S^j_{t-1}) \\ & + c_{trans}.|i - j + acc\_transition[S^j_{t-1}] - n_t| \\ & + c_{obs}.\widetilde{d}^{t,i}_{RSS} \end{aligned} \tag{3}$$

S^i_t is the i-th floor during the t-th step detection and $C(S^i_t)$ is the cost up until the current step, given that the pedestrian is at the i-th floor. $C(S^i_t)$ is calculated with eq. 4.

$$C(S^i_t) = \begin{cases} Min[C(S^i_t|S^j_{t-1})]\,\forall j \in [0, n-1], & t > 0 \\ c_{obs}.\widetilde{d}^{t,i}_{RSS}, & t = 0 \end{cases} \tag{4}$$

The initial cost for each floor is zero, i.e. no assumptions about the initial floor are made. n_t is the number of floor changes detected at step t. $type_t$ is the transition type at step t. $acc\_transition[S^j_{t-1}]$ is the floor change accumulated by the Viterbi algorithm during a stair transition for the j-th floor up until the previous step. If the measured RSS vector does not contain a value for a certain AP, then a default value of -100 dBm is inserted for that AP. If there are no RSS measurements available at step t, then $\widetilde{d}^{t,j}_{RSS}$ is set to zero for all floors. The transition cost c_{trans} and observation cost c_{obs} are coefficients that depend on accuracy of the transition detection and fingerprint matching respectively. Finally, the sequence with the smallest cost is chosen as the most likely sequence for batch processing. For real-time processing, the floor with the smallest cost is chosen at each iteration.

TABLE I

Confusion matrix for batch floor transition detection. The numbers are percentages.

	True Activity				
Detected Activity	**W**	**S-U**	**S-D**	**E-U**	**E-D**
W	**89**	5	7	3	6
S-U	6	**95**	0	0	0
S-D	3	0	**92**	0	0
E-U	1	0	0	**97**	0
E-D	1	0	0	0	**94**

W: Walking, S: Stairs, E: Elevator, U: Up, D: Down

TABLE II

Confusion matrix for real-time floor transition detection. The numbers are percentages.

	True Activity				
Detected Activity	**W**	**S-U**	**S-D**	**E-U**	**E-D**
W	**89**	35	33	3	6
S-U	6	**65**	0	0	0
S-D	3	0	**67**	0	0
E-U	1	0	0	**97**	0
E-D	1	0	0	0	**94**

W: Walking, S: Stairs, E: Elevator, U: Up, D: Down

E. Evaluation configurations

The proposed algorithm is evaluated with data from real measurements. The considered environment is located at floors 4-11 of a 41 m x 27 m (1107m^2) office building. Figure 2 shows a floorplan of the 5th floor of this building, where most of the measurements were performed. The office floors have three elevators (E), three staircases (S) and two wifi APs, and are 3.5 m high. The APs are indicated with blue dots. The PL model of section II-C incorporates all elements of Figure 2, except for the open staircase (S3) when predicting the RSS fingerprints. Note that the center of the floor consists of thick concrete walls, where the smartphones often can't detect the APs' signals. Plots of the other floors are omitted because they are almost identical.

The smartphones used for evaluation are a Samsung Galaxy S5 (2014) and a Samsung Galaxy S7 (2016). Both have a 9-DOF IMU, barometer and wifi chipset. Random trajectories were travelled on each floor using both phones, for RSS self-calibration (II-C) and to determine the barometer noise parameter σ_h for each device (II-B). To calibrate the elevator detection algorithm, the elevator was used several times, rising/descending a different amount of floors each time. Similarly, a staircase was climbed several times to choose the coefficients c_{trans} and c_{obs}, which are chosen to be 4 and 1 respectively. W_l is 10 (Galaxy S7) and 16 (Galaxy S5).

Almost two hours (116 minutes) of evaluation data were recorded while roaming through the building, of which twelve minutes were spent using the elevator and 23 minutes climbing the stairs. 76 floor transition events were performed, of which 28 were elevator transitions. The smartphone was held in front of the body as if the user were watching the screen. The data were logged with the GetSensorDataApp [33]. The interface of this app has a button which saves a timestamp. The user presses the button each time when entering or leaving a staircase or elevator during a pre-defined trajectory, in order to correctly evaluate the floor (transition) detection algorithm.

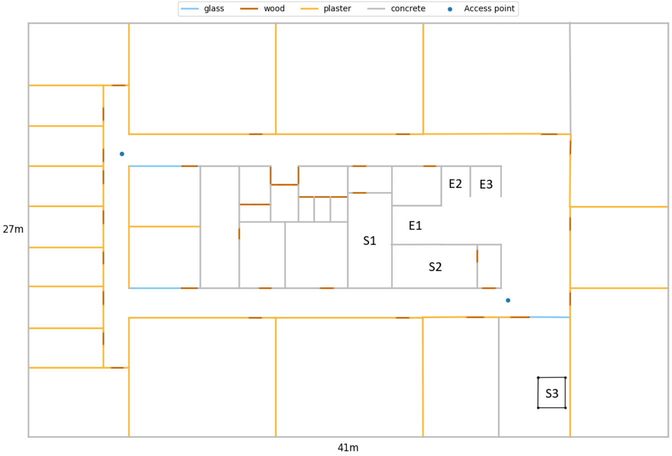

Fig. 2. Floorplan of the fifth floor of the office building. E1-3 are elevators and S1-2 are closed concrete staircases which connect all other floors. S3 is an open metal staircase which connects to the sixth floor. The two blue dots in the corridors are access points. The other floors are almost identical.

III. Results

Tables I and II show confusion matrices of the classification results for batch and real-time floor transition detection. The global detection accuracy of these intervals is 90.9% (table I) for batch and 84.6% (table II) for real-time mode. These results are biased, since most of the evaluation data consists of regular walking and always choosing *walking* would result in 69.5% accuracy. The unbiased global accuracies are 92.6% (batch) and 81.6% (realtime). In real-time, most errors are false negative stair detections at the start of a stair transition. The first steps are labeled as regular walking due the length of the stair detection window. This is a design choice: the long detection window allows the algorithm to ignore noisy measurements. Indeed, the amount of false negative stair detections is drastically reduced in batch mode, because we know at the start of a stair transition that the past steps in the detection window happened on a staircase too. Of the remaining stair detection errors, all false negatives and most of the false positives happen when the user is entering or leaving a staircase, because of the delay introduced by the LP filter on the barometer data. These errors have no impact on the floor number detection or localisation algorithms. Some false stair detections happened when the user was not near a staircase. Most of these did not cause a floor number change, since the detected height change must be at least 50% of the floor height to trigger a floor number change. However, false floor transitions can still be disadvantageous if they are used to improve 2D localisation [17]. When a false detection did trigger a floor number change, the algorithm added an extra

TABLE III
FLOOR DETECTION ERRORS FOR DIFFERENT CONFIGURATIONS AS DISTANCE IN FLOOR NUMBERS TO THE TRUE FLOOR. THE NUMBERS ARE PERCENTAGES.

Algorithm configuration	Floor number distance							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
RSS only (Euclidean distance metric)	**91.6**	6.8	0.2	0	1.0	0.1	0.3	0.1
Viterbi: RSS (realtime)	**85.3**	8.4	3.5	0	2.1	0	0.5	0.2
Viterbi: RSS (batch)	**94.1**	3.8	1.0	0	0.7	0	0.4	0
Viterbi: RSS Floor transition detection (realtime)	**99.1**	0.9	0	0	0	0	0	0
Viterbi: RSS Floor transition detection (batch)	**99.7**	0.3	0	0	0	0	0	0

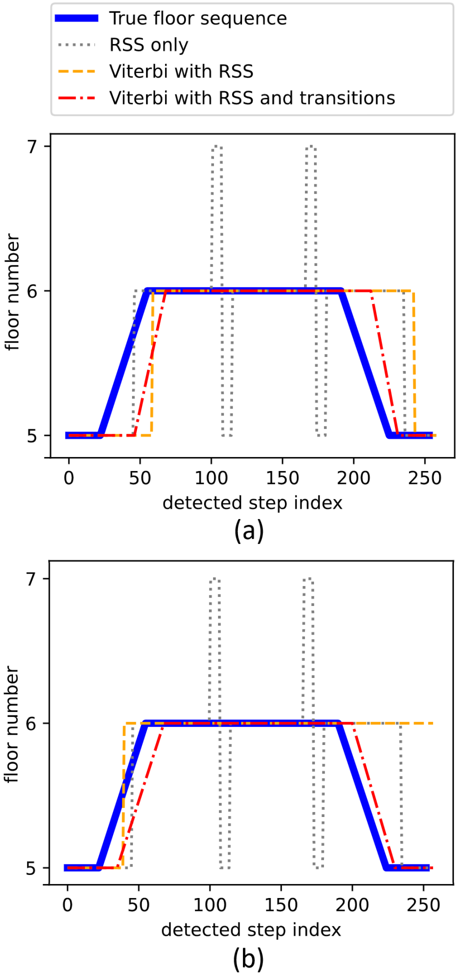

Fig. 3. Plots of the estimated floor number in function of the detected step index for variations of the floor number detection algorithm for a slice of the evaluation data, in real-time (a) and batch (b) mode.

transition back to the correct floor after a few seconds, because the RSS measurements favoured the correct floor.

Table III lists a comparison of variations of the floor number detection algorithm. These variations include pure RSS matching with the Euclidean distance metric, Viterbi with only WiFi RSS, and fusion of both WiFi RSS and floor transitions with Viterbi, in realtime and batch. Only the intervals where the real activity is *walking* are used for evaluation of the floor number detection. As expected, the highest accuracy is achieved by (f)using both transition detection and RSS measurements, with batch (99.7%) providing slightly better accuracy than realtime mode (99.1%). Surprisingly, RSS only (91.6%) performs better than Viterbi with RSS in realtime (85.3%), while Viterbi with RSS in batch mode (94.1%) lies in the middle.

Figure 3 (a) shows the real-time output for a slice of the evaluation data. The user started at the fifth floor and went to the sixth floor by stairs. He then returned to the fifth floor by stairs after some roaming. While the output of the Viterbi algorithm is always an integer floor number, the floor number is interpolated during transitions for visualisation of these intervals. Figure 3 (b) shows the batch output for the same evaluation data. These plots confirm the explained results. The wrong floor number changes caused by bad RSS matching (grey) are correctly ignored when RSS measurements are fused with floor transition detections, even in realtime. However, the stair detection is delayed in real-time mode because of the long detection window.

With over 99% floor detection accuracy, our algorithm is comparable to other recent works [19], [20]. However, these require the initial floor to be known. The floor transition detection in [20] was designed for a specific type of staircase, where each floor transition consists of “two staircases and a transition area between them” (p. 8). 100% accuracy was achieved in [19], but a newer high-end smartphone (Iphone X) provides a clear advantage in detecting the correct floor number. Other systems, e.g. [12], [17], are difficult to compare because the floor number detection is part of a complete PDR system and is not seperately evaluated.

IV. DISCUSSION AND FUTURE WORK

We proposed a new smartphone-based floor transition and floor number detection algorithm, which fuses data from the accelerometer, barometer and WiFi RSS. We first implemented an elevator detection algorithm based on the work in [27]. Then, we proposed a new adaptive stair detection algorithm, which adapts its parameters according to the noise produced by the barometer sensor. Our stair detection algorithm handles the pressure drift problem and is able to ignore fast pressure changes, which can be caused by opening/closing doors or windows [21]. RSS fingerprinting provides absolute floor number estimation and is fused with floor transitions for accurate floor number detection. The purpose of this algorithm is to complement PDR algorithms, in two ways. First, detecting the floor number enables tracking a pedestrian across multiple floors of a building. Second, recognition of activities such as climbing stairs and taking an elevator can also improve 2D localisation.

The evaluation dataset consists of 116 minutes of recorded data, during which the actors changed floors 76 times. The proposed algorithm achieves 99.1% accuracy in real-time and 99.7% in batch mode. Only pre-existing APs were used, of which two were available on each 41 m x 27 m floor. Since the radiomaps are model-based, expensive measurement campaigns are not needed. Combined with the small and easily obtained amount of necessary training data, this algorithm can be easily deployed and is independent of a dedicated localisation infrastructure.

When the pedestrian is walking regularly (i.e. not climbing stairs), 8% of the time the activity is labeled as climbing the stairs. While most of these wrong classifications happen when the user is leaving a staircase, some happen when the pedestrian is nowhere near a staircase. Future work will consist of reducing these false stair detections by incorporating data from both the accelerometer and gyroscope, as proposed in other works [12], [34]. Consequently, more accurate stair detection (and using a more modern smartphone) will allow us to shorten the detection window, reducing the delay in real-time mode.

REFERENCES

[1] Athanasios Gkelias Faheem Zafari. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2597, 2019.

[2] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Alhadhrami, Mai Al-Ammar, and Hend Al-Khalifa. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16:1–36, 05 2016.

[3] C. Zhang, M. Kuhn, B. Merkl, M. Mahfouz, and A. E. Fathy. Development of an uwb indoor 3d positioning radar with millimeter accuracy. In *2006 IEEE MTT-S International Microwave Symposium Digest*, pages 106–109, 2006.

[4] G. Schroeer. A real-time uwb multi-channel indoor positioning system for industrial scenarios. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–5, 2018.

[5] J. Luo, L. Fan, and H. Li. Indoor positioning systems based on visible light communication: State of the art. *IEEE Communications Surveys Tutorials*, 19(4):2871–2893, 2017.

[6] David Plets, Yousef Almadani, Sander Bastiaens, Muhammad Ijaz, Luc Martens, and Wout Joseph. Efficient 3d trilateration algorithm for visible light positioning. *Journal of Optics*, 21, 03 2019.

[7] J. b. Xu, H. q. Zhang, and J. l. Zhang. Self-adapting multi-fingerprints joint indoor positioning algorithm in wlan based on database of ap id. In *Proceedings of the 33rd Chinese Control Conference*, pages 534–538, 2014.

[8] O. Costilla-Reyes and K. Namuduri. Dynamic wi-fi fingerprinting indoor positioning system. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 271–280, 2014.

[9] F. W. Gobana. Survey of inertial/magnetic sensors based pedestrian dead reckoning by multi-sensor fusion method. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1327–1334, 2018.

[10] Alwin Poulose, Jihun Kim, and Dong Seog Han. A sensor fusion framework for indoor localization using smartphone sensors and Wi-Fi RSSI measurements. *Applied Sciences (Switzerland)*, 9(20), 2019.

[11] 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)*, pages 1–7, 2016.

[12] Toni Fetzer, Frank Ebner, Markus Bullmann, Frank Deinzer, and Marcin Grzegorzek. Smartphone-based indoor localization within a 13th century historic building. *Sensors*, 18(12), 2018.

[13] J. C. Aguilar Herrera, P. G. Plöger, A. Hinkenjann, J. Maiero, M. Flores, and A. Ramos. Pedestrian indoor positioning using smartphone multi-sensing, radio beacons, user positions probability map and indoor osm floor plan representation. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 636–645, 2014.

[14] S. Gansemer, S. Hakobyan, S. Püschel, and U. Großmann. 3d wlan indoor positioning in multi-storey buildings. In *2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 669–672, 2009.

[15] Dominik Gusenbauer, Carsten Isert, and Jens Krösche. Self-contained indoor positioning on off-the-shelf mobile devices. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–9, 2010.

[16] Boyuan Wang, Xuelin Liu, Baoguo Yu, Ruicai Jia, and Xingli Gan. Pedestrian dead reckoning based on motion mode recognition using a smartphone. *Sensors*, 18(6):1811, Jun 2018.

[17] G. Pipelidis, N. Tsiamitros, C. Gentner, D. B. Ahmed, and C. Prehofer. A novel lightweight particle filter for indoor localization. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2019.

[18] Binghao Li, B. Harvey, and T. Gallagher. Using barometers to determine the height for indoor positioning. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7, 2013.

[19] Hong-Yu Zhao, Wanli Cheng, Ning Yang, Sen Qiu, ZHELONG Wang, and Jianjun Wang. Smartphone-based 3d indoor pedestrian positioning through multi-modal data fusion. *Sensors*, 19:4554, 10 2019.

[20] J. Yan, G. He, A. Basiri, and C. Hancock. 3-d passive-vision-aided pedestrian dead reckoning for indoor positioning. *IEEE Transactions on Instrumentation and Measurement*, 69(4):1370–1386, 2020.

[21] M. Tanigawa, H. Luinge, L. Schipper, and P. Slycke. Drift-free dynamic height sensor using mems imu aided by mems pressure sensor. In *2008 5th Workshop on Positioning, Navigation and Communication*, pages 191–196, 2008.

[22] Khanh Nguyen-Huu and Seon Woo Lee. A Multi-Floor Indoor Pedestrian Localization Method Using Landmarks Detection for Different Holding Styles. *Mobile Information Systems*, 2021, 2021.

[23] F. Ebner, T. Fetzer, F. Deinzer, L. Köping, and M. Grzegorzek. Multi sensor 3d indoor localisation. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–11, 2015.

[24] Seong sik Kim. Floor detection using a barometer sensor in a smartphone. 2017.

[25] George Slade. The viterbi algorithm demystified. 03 2013.

[26] W. Kang and Y. Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal*, 15(5):2906–2916, 2015.

[27] T. Yang, K. Kaji, and N. Kawaguchi. Elevator acceleration sensing: Design and estimation recognition algorithm using crowdsourcing. In *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*, pages 534–539, 2013.

[28] Standards and jurisdiction. https://www.escaliers-echelle-europeenne.com/en/tips/standards-jurisdiction/. Accessed: 2021-04-16.

[29] Ghent Univerity IMEC. Wica heuristic indoor propagation prediction tool (whipp). https://www.waves.intec.ugent.be/exposure-tool/expert-edition, 2020.

[30] David Plets, Wout Joseph, Kris Vanhecke, Emmeric Tanghe, and Luc Martens. Coverage prediction and optimization algorithms for indoor environments. *EURASIP Journal on Wireless Communications and Networking*, 2012, 12 2012.

[31] Christos Laoudias, Robert Piché, and Christos Panayiotou. Device signal strength self-calibration using histograms. In *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, pages 1–8, 11 2012.

[32] G.D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[33] A. R. Jiménez, F. Seco, and J. Torres-Sospedra. Tools for smartphone multi-sensor data registration and gt mapping for positioning applications. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2019.

[34] Akram Bayat, Marc Pomplun, and Duc Tran. A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, 34:450–457, 12 2014.