Sampling Rate Dependency in Pedestrian Walking Speed Estimation using DualCNN-LSTM

Takuto Yoshida
takuto@ucl.nuee.nagoya-u.ac.jp
Nagoya University
Nagoya, Japan

Junto Nozaki
nozaki@ucl.nuee.nagoya-u.ac.jp
Nagoya University
Nagoya, Japan

Kenta Urano
vrano@ucl.nuee.nagoya-u.ac.jp
Nagoya University
Nagoya, Japan

Kei Hiroi
k.hiroi@ucl.nuee.nagoya-u.ac.jp
Nagoya University
Nagoya, Japan

Katsuhiko Kaji
kaji@aitech.ac.jp
Aichi Institute of Technology
Toyota, Japan

Takuro Yonezawa
yonezawa.takuro@f.mbox.nagoya-u.ac.jp
Nagoya University
Nagoya, Japan

Nobuo Kawaguchi
kawaguti@nagoya-u.jp
Nagoya University
Nagoya, Japan

ABSTRACT

There are many inertial sensor based indoor localization methods for smartphone, for example, SINS and PDR. However, most of the MEMS sensors of smartphones are not precise enough for these methods. We proposed end-to-end walking speed estimation method using deep learning to perform robust walking speed estimation with a low-precision sensor. Currently, we use the input data with a fixed format of 200 samples at 100 Hz. However, the sampling rate and sequence length should be changed appropriately depending on the required accuracy and terminal performance. They are critical factors when using our method for a long time on a terminal because continuous processing of a large amount of data leads to shorter battery life. In this paper, we evaluate the accuracy of the estimated speed by our method when changing the sampling rate and sequence length. As a result, using 5 patterns of combinations, the estimation accuracy hardly changed.

UbiComp/ISWC '19 Adjunct, September 9–13, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6869-8/19/09…$15.00
https://doi.org/10.1145/3341162.3343765

CCS CONCEPTS

• **Information systems** → **Location based services**; • **Computing methodologies** → *Supervised learning by regression.*

KEYWORDS

indoor positioning, deep neural networks, PDR

ACM Reference Format:
Takuto Yoshida, Junto Nozaki, Kenta Urano, Kei Hiroi, Katsuhiko Kaji, Takuro Yonezawa, and Nobuo Kawaguchi. 2019. Sampling Rate Dependency in Pedestrian Walking Speed Estimation using DualCNN-LSTM. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), September 9–13, 2019, London, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3341162.3343765

1 INTRODUCTION

Most of SINS(Strapdown Inertial Navigation System) use accelerameter and gyrocope snsor. These senor is necessarily to be high precision. SINS estimates the position by double integration of acceleration and the attitude by integration of angular rate [21, 22]. PDR(Pedestrian Dead Reckoning) is one of the promising technology for indoor localization. Most of the current PDR techniques can be categorized as the strapdown algorithm and the step-and-heading algorithm[5, 9, 20]. The strapdown algorithms require high precision sensor devices to accurately estimate position. These methods require high precision sensor to estimate position accurately. However, most of the current smartphones equipped with MEMS sensors do not have enough precision to estimate position accurately, it causes a time-cumulative drift-error. On the other hand, the step-and-heading PDR algorithm has the major difficulty in robust estimation of the step length and the step detection. Step length depends on several parameters such as a person's height, walking speed and type of gait. So, in the conventional method[2, 3, 10, 11, 14], it is difficult to estimate step length without using user-dependent information. For the step detection, "stamp" with usual "walk" is very difficult. Therefore, we proposed end-to-end walking estimation method using deep learning to overcome these problems. The proposed method can estimate robustly for various data such as the gaits[24].

Currently, we use the fixed input data at 100 Hz and 200 sequence lengths. However, the sampling rate and sequence length of the data should be changed adaptively depending on the situation, the required accuracy, and the terminal specification. The data sampling rate and the sequence length is a critical factor when using our proposed method on a terminal for a long time because the large data at a high-frequency sampling rate and the intermittent short length data consumes a lot of power for processing. On the other hand, changing the sampling rate and sequence length of the data have negative aspects, for example, the decreased accuracy and the delay of the estimated speed. In this paper, we evaluate the accuracy of estimated speed by a proposed method when changing the sampling rate and sequence length of the data. We use the data with a sampling rate of 25, 50, 100 Hz and a sequence length of 50, 100, 200. As a result of the evaluation using 5 patterns of combinations of sampling rate and sequence length, the estimation accuracy hardly changed. The difference in the accuracy among all data is about 0.5-2.5%.

2 RELATED WORK

Pedestrian Localization Systems

There is a large amount of study which handles pedestrian localization systems[5, 9, 20]. One of the successful PDR is based on ZUPT(Zero Velocity Updates)[25] method which uses fixed sensors on the foot[6, 12]. But this method cannot utilize smartphone because it requires to fix the sensors on the foot. Most of smartphone PDR researches use a step-and-heading algorithm. For the step detection, Alzantot[2] utilize finite automaton with peak detection. Also, there are several PDR competitions[11, 14] which collects several algorithms to evaluate them under the same condition. In addition, there is a step-length estimation method which utilizes stacked autoencoders[7]. These works challenged to increase the accuracy of PDR. however, still not achieved enough accuracy for real-world deployment.

Table 1: Collected dataset

Number of Subjects	9 subjects (20's male)
Terminal Position	Hand, Left/Right Waist Pocket
Type of Gait	walk, stamp, skip
Total Routes	112 routes
Average walking time	79.6 sec, SD: 53.1sec
Average route length	51.6m, SD:31.9m
Device	PHAB2 Pro (Android 6.0.1) Xperia G8342(Android 8.0.0) Nexus 6 (Android 6.0.1)
Software	Google Tango HASC Logger for Android
Sensor data	Acceleration Angular rate GPS Magnetic Pressure WiFi

End-to-end Machine Learning System

Recent advancement of deep learning technology enables end-to-end machine learning on different domains[13, 17, 23]. We obtain various technical hints from these researches. One of the most famous end-to-end machine learning systems is "Deep Speech"[8] which enables end-to-end speech recognition. By utilizing a fully connected layer and bi-directional Recurrent Neural Network, they enabled learning from the unaligned transcribed audio dataset. In the indoor positioning field, there are deep learning based methods. Chen[4] propose IONet which is neural network framework using inertial sensor data to estimate indoor position.

3 END-TO-END WALKING SPEED ESTIMATION METHOD

We proposed end-to-end walking speed estimation method using deep learning to overcome the problems of threshold-based PDR. Our proposed method does not have to estimate stride and detect step. In our method, we need to collect acceleration, gyro and correct speed data and train DNN model using these data. The ground truth is 2D speed data converted by 3D trajectory of the terminal position form Google Tango. We employ DualCNN-LSTM network model shown in Fig. 4 which is the integration of CNN-LSTM and fusion layer as walking speed estimation model.

Data Collection

End-to-end machine learning of PDR requires ground truth data of the precise terminal location with sensor inputs. In this paper, we employ Google Tango enabled smartphone

(a) Route from side view **(b) Route from upper view** **(c) Participant**

Figure 1: Data collection method

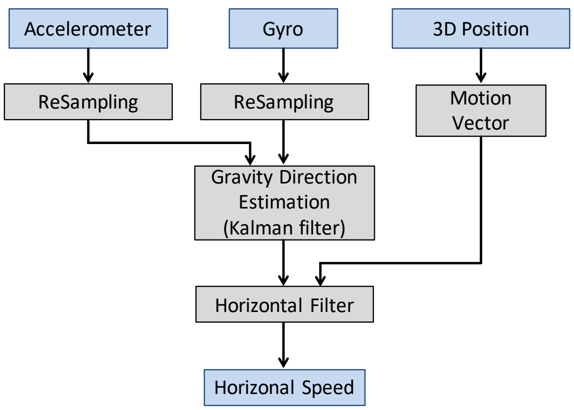


Figure 2: Extracting horizontal speed input for the learning phase

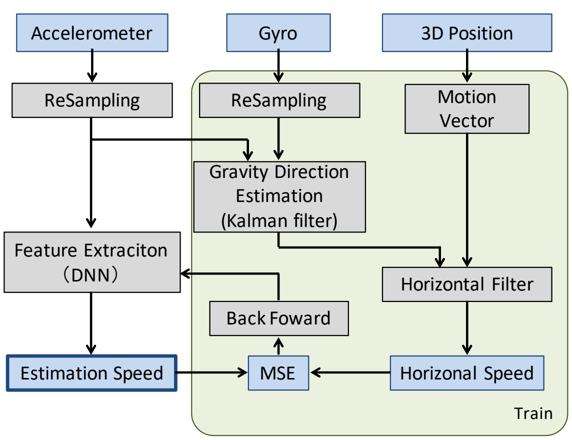


Figure 3: The flowchart of speed estimation

(Lenovo PHAB2 Pro) with original location data logger software and HASC Logger[15]. Google Tango utilizes vision tracking called "VSLAM" with sensor fusion technology. By using Google Tango, we can obtain 3D trajectory of the terminal position. The location measurement error of Google Tango in our pre-experiment is less than 30cm, and also in the evaluation literature. So we use Google Tango tracking data as a ground truth data of the terminal location. We have collected 112 different routes by 9 subjects who is equipped with 3 smartphones simultaneously. Subjects walk the route shown in Fig. 1b clockwise and counterclockwise. Fig. 1c shows one of the subjects collecting data. As you can see, Fig. 1 shows we collect data outdoors because we can collect various data such as GPS other than inertial sensor data to enhance the dataset. In this paper, we do not use the data other than time and inertial sensor data. In our data collection, subjects are ordered to perform a different type of gaits such as fast walk, normal walk, slow walk, stamp and skip. Details of the collected PDR dataset is shown in Table 1.

Based on the collected data, we have to estimate the pedestrian's speed. In this paper, we focus on two-dimensional trajectory. However, the speed vector which is calculated from Tango's location data cannot be used directly because Tango exports data which includes 3D data. Therefore, we applied Karman filter based method[16] to estimate and remove data of gravity direction with considering noise reduction[3]. We calculate 2D moving vector by using gravity direction vector g which is estimated from Karman filter as following: $v_h = v - \frac{g \cdot v}{|g|^2} v$. Fig. 2 shows the overview of the process of extracting horizontal speed. Fig 3 shows the flowchart of speed estimation. Horizontal speed extracted form 3D position data is used as an argument of the loss function.

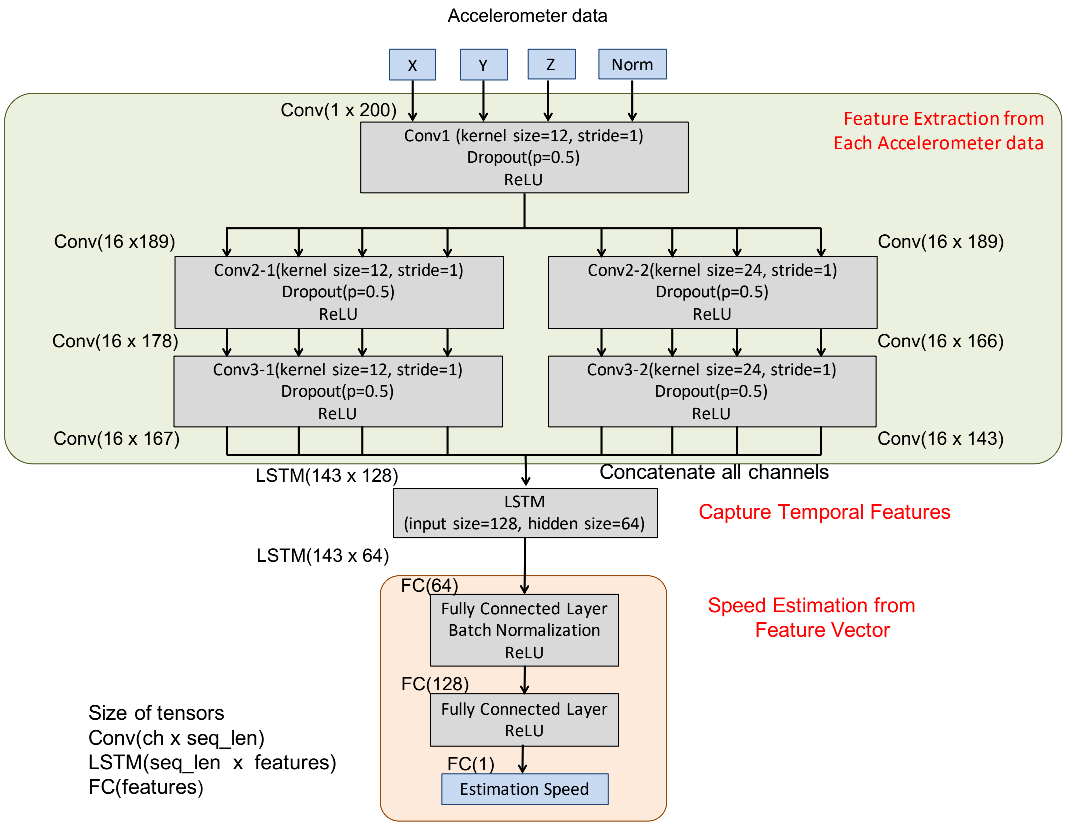


Figure 4: DualCNN-LSTM network model for End-to-End Walking Speed Estimation (kernel size: 12)

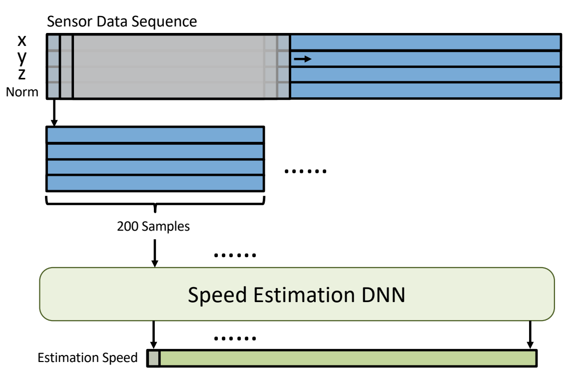


Figure 5: Sensor data input and estimated speed output of DualCNN-LSTM

Deep Neural Network

To model the walking speed, we employ CNN-LSTM[18] which is successfully used for activity recognition and other temporal signal processing methods. Additionally, we use the fusion layer to capture short and long term features of walking activities. Detail of the structure and tensor sizes of DualCNN-LSTM network is shown in Fig. 4. We utilize dropout(p=0.5), and ReLU for activation function. For the learning phase of DualCNN-LSTM, we use horizontal walking-speed for simplicity.

Fig. 5 shows the data flow of the sensor data input and the estimated horizontal speed output of DualCNN-LSTM. We input sequence data into a convolutional layer of the DualCNN-LSTM network. Inside of the network, short term feature and long term feature are extracted and combined into LSTM. We use PyTorch[19] as a deep learning platform.

4 EVALUATION

We evaluate the proposed method with data whose sampling rate is 25, 50, 100 Hz and a sequence length is 50, 100, 200 samples. Optimal model parameters depend on sampling rate and sequence length. Therefore, we perform a grid search to find the best model for each data. We employ evaluation metrics as PIEM(Path Independent Evaluation Metrics)[1].

Table 2: Evaluation Dataset(SR: sampling rate[Hz])

	sequence length		
SR	200	100	50
100	2sec	-	-
50	4sec	2sec	-
25	8sec	-	2sec

Table 3: Hyper Parameter

kernel size	2, 4, 8, 10, 12, 16, 20, 24
batch size	64, 128, 256, 512, 1024
learning rate	0.001, 0.002, 0.003

Table 4: The best hyperparameters

input data		batch size	learning rate	kernel size
SR	sample			
100	200	1024	0.002	8
50	200	256	0.002	20
50	100	1024	0.002	8
25	200	64	0.002	4
25	50	1024	0.002	2

Generating Low-frequency Data

Input data consists of 5 patterns shown in Table 2. It is difficult to collect different sampling rate data accurately because Android sampling rate is unstable and if we collect 3 differet sampling rates data, we need to have 9 terminals(3 each for hand, light/left waist pocket) at the same time. Therefore, we resample the collected data at 100, 50, 25 Hz and mold into 200, 100, 50 sequence length. The processed data can be divided into 2 types. The one corresponds to sequence data fixed at 200 samples, and the other corresponds to sequence data fixed at 2 seconds which is calculated as following: $t\,(\text{sec}) = \frac{SL}{SR}(SL : sequence\ length, SR : sampling\ rate)$.

Hyperprameter

Optimal model parameters depend on sampling rate and sequence length. It is necessary to change the kernel size of the CNN layer because the time series features of the input data depending on the sampling rate and sequence length. The batch size and learning rate also should be changed because these parameters is important when training model. The candidates of hyperparameters are shown in table 3. The best hyperparameter is the one when the evaluation metric AMDE(Average moving distance error; described later) is the smallest. The result of the best parameters is shown in table 4.

Table 5: Evaluation with dataset fixed at 200 samples (SR: sampling rate[Hz])

SR[Hz]	AMDE[m]	MDEM[%]	MDES[%]
100	5.79	**8.74**	5.96
50	**5.34**	9.92	**5.05**
25	6.72	11.14	5.35

Table 6: Evaluation with dataset fixed at 2 sec (SR: sampling rate[Hz])

SR[Hz]	AMDE[m]	MDEM[%]	MDES[%]
100	**5.79**	**8.74**	5.96
50	6.13	11.12	**5.64**
25	6.34	11.02	6.15

Evaluation with dataset

We first use our PDR dataset for evaluation. We divide the 9 subjects PDR dataset into 7 subjects for learning, and 1 subject for validation, 1 subject for test, which results in 265 learning files and 45 test files. For the evaluation metrics, we employ the following metrics called PIEM(Path Independent Evaluation Metrics)[1]. Then we obtain the error rate from the slope of the line regressed by the least square estimate method.

(1) Average moving distance error (AMDE)
(2) Moving distance error rate for each meter (MDEM)
(3) Moving distance error rate for each second (MDES)

For AMDE, we calculate total distance error by using estimated walking speed and elapsed time. For MDEM and MDES, we first create a scatter plot from moving distance error and ground truth distance, or elapsed time. The smaller MDEM is, the smaller the estimation error in long distances movement. On the other hand, the smaller MDES is, the smaller the estimation error in long time movement.

Result of the evaluation with dataset fixed at 200 samples is shown in Table 5. Evaluation metrics shows the difference of the accuracy among all sampling rates is about 1-2.5% in MDEM and MDES and the input data which gives the best evaluation by each evaluation metric is different. Fig. 6 shows the speed estimation results of 100, 50, 25Hz data. This shows that as the sampling rate decreases, the delay of estimated speed increases, and estimated speed becomes smoother. Result of the evaluation with dataset fixed at 2 sec is shown in Table 6. Evaluation Metrics shows the difference of the accuracy among all sampling rates is about 0.5-2.5% in MDEM and MDES and the input data which gives the best evaluation by each evaluation metric is different. Fig. 7 is examples of the speed estimation results of 100, 50, 25Hz data. The waveform of 25 Hz is more undulating than the other data.

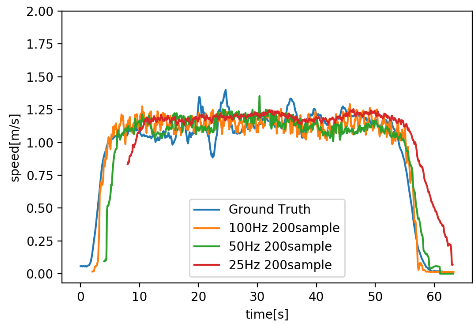


Figure 6: Speed estimation by input data fixed at 200 samples

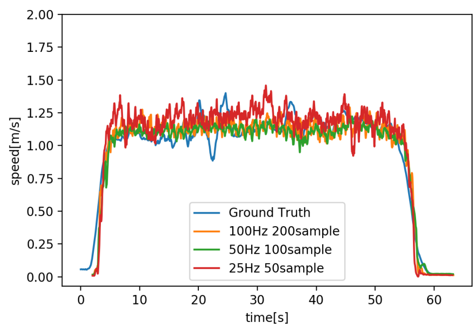


Figure 7: Speed estimation by input data fixed at 2 sec

Discussion

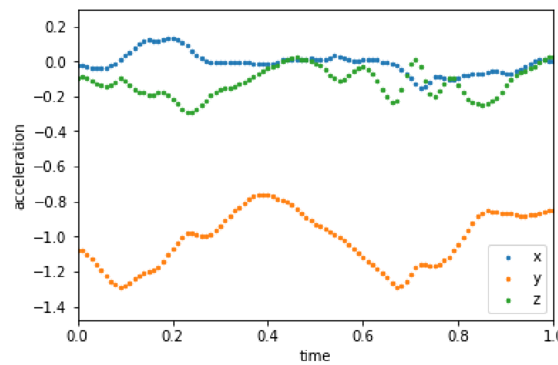
At first, we estimate that dropping sampling rate cause dropping accuracy of speed estimation. However, the result(table 5, 6) shows that there is only few difference in estimation accuracy among sampling rates(100, 50, 25 Hz). Fig. 8 shows the acceleration during walking resampled to 100 Hz, 50 Hz, and 25 Hz. Dropping the sampling rate to 25 Hz does not lose waveform features. Therefore, it is assumed that the DNN can capture the features of acceleration regardless of sampling rates(100, 50, 25Hz).

In this evaluation, we can comfirm the interesting evaluation results between 100Hz and 50Hz. Table 5, 6 show 2 common features:

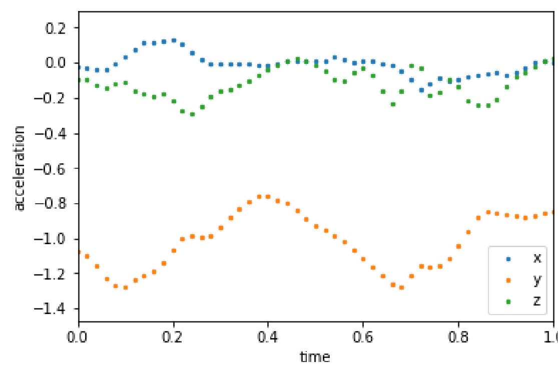
- In 100 Hz, MDEM is smaller than in 50 Hz
- In 50 Hz, MDES is smaller than in 100 Hz

Small MDEM and large MDES means that the accuracy of estimated speed in fast movement is high. On the other hand, large MDEM and small MDES means that the accuracy of estimated speed in slow movement is high[24]. This results means that:

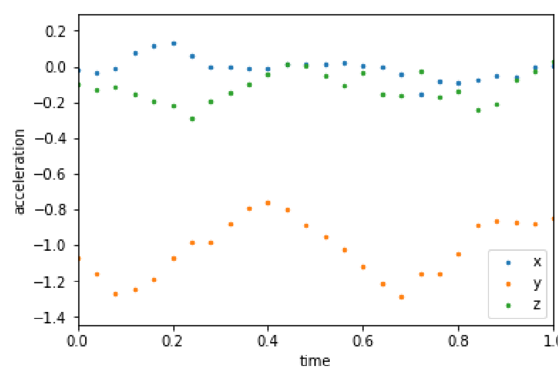
- If we use 100 Hz data, we can estimate speed in fast movement like a skipping more accurately than we use 50 Hz data



(a) 100 Hz



(b) 50 Hz



(c) 25 Hz

Figure 8: Resampled acceleration

- If we use 50 Hz data, we can estimate speed in slow movement like a stampping more accurately than we use 100 Hz data

The difference in sampling rates affects the accuracy in estimated speed depending on gaits.

5 CONCLUSION

In this paper, we evaluated the accuarcy of estimated speed using deep learning when changing the sampling rate and sequence length of the data. The estimation accuracy hardly changed at sampling rates of 100 Hz, 50 Hz and 25 Hz and 200, 100, 50 sequence length. It turned out that dropping the sampling (to 25 Hz) rate hardly affect the accuracy of speed estimation because data around 25 Hz do not lose the feature of the acceleration waveform, and the difference in sampling rates affects the accuracy in estimated speed depending on gaits.

In this evaluation, there is a difference in the evaluation results among all input data. However, it is difficult to prove that the result is general because the dataset we use is too small to prove generality completely. Therefore, we need to collect extended dataset(various gaits, more subjects).

ACKNOWLEDGMENTS

This work is supported by JSPS KAKENHI Grant Number JP17H01762.

REFERENCES

[1] M. Abe, K. Kaji, K. Hiroi, and N. Kawaguchi. 2016. PIEM: Path Independent Evaluation Metric for Relative Localization. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 1–8.

[2] M. Alzantot and M. Youssef. 2012. UPTIME: Ubiquitous pedestrian tracking using mobile phones. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. 3204–3209.

[3] R. Ban, K. Kaji, K. Hiroi, and N. Kawaguchi. 2015. Indoor positioning method integrating pedestrian Dead Reckoning with magnetic field and WiFi fingerprints. In *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. 167–172.

[4] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. 2018. IONet: Learning to Cure the Curse of Drift in Inertial Odometry. *CoRR* abs/1802.02209 (2018). arXiv:1802.02209 http://arxiv.org/abs/1802.02209

[5] Alejandro Correa Vila, Marc Barceló, Antoni Morell, and Jose Lopez Vicario. 2017. A Review of Pedestrian Indoor Positioning Systems for Mass Market Applications. *Sensors (Switzerland)* 17 (08 2017).

[6] S Godha and G Lachapelle. 2008. Foot mounted inertial system for pedestrian navigation. *Measurement Science and Technology* 19, 7 (may 2008), 075202.

[7] F. Gu, K. Khoshelham, C. Yu, and J. Shang. 2018. Accurate Step Length Estimation for Pedestrian Dead Reckoning Localization Using Stacked Autoencoders. *IEEE Transactions on Instrumentation and Measurement* (2018), 1–9.

[8] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. DeepSpeech: Scaling up end-to-end speech recognition. (12 2014).

[9] R. Harle. 2013. A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys Tutorials* 15, 3 (Third 2013), 1281–1293.

[10] Baoqi Huang, Guodong Qi, Xiaokun Yang, Long Zhao, and Han Zou. 2016. Exploiting Cyclic Features of Walking for Pedestrian Dead Reckoning with Unconstrained Smartphones. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 374–385.

[11] Ryosuke Ichikari, Ryo Shimomura, Masakatsu Kourogi, Takashi Okuma, and Takeshi Kurata. 2018. Review of PDR Challenge in Warehouse Picking and Advancing to xDR Challenge. *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (2018), 1–8.

[12] A. R. Jiménez, F. Seco, J. C. Prieto, and J. Guevara. 2010. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In *2010 7th Workshop on Positioning, Navigation and Communication*. 135–143.

[13] Kai Wang, B. Babenko, and S. Belongie. 2011. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*. 1457–1464.

[14] K. Kaji, K. Kanagu, K. Murao, N. Nishio, K. Urano, H. Iida, and N. Kawaguchi. 2016. Multi-algorithm on-site evaluation system for PDR challenge. In *2016 Ninth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. 1–6.

[15] Nobuo Kawaguchi, Hodaka Watanabe, Tianhui Yang, Nobuhiro Ogawa, Yohei Iwasaki, Katsuhiko Kaji, Tsutomu Terada, Kazuya Murao, Hisakazu Hada, Sozo Inoue, Yasuyuki Sumi, Yoshihiro Kawahara, and Nobuhiko Nishio. 2012. HASC2012corpus: Large Scale Human Activity Corpus and Its Application.

[16] M. Kourogi, T. Ishikawa, and T. Kurata. 2010. A method of pedestrian dead reckoning using action recognition. In *IEEE/ION Position, Location and Navigation Symposium*. 85–89.

[17] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end Training of Deep Visuomotor Policies. *J. Mach. Learn. Res.* 17, 1 (Jan. 2016), 1334–1373. http://dl.acm.org/citation.cfm?id=2946645.2946684

[18] Francisco Javier Ord ' o nez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* 16, 1 (2016), 115.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[20] Wilson Sakpere, Michael Adeyeye Oshin, and Nlanhla Mlitwa. 2017. A State-of-the-Art Survey of Indoor Positioning and Navigation Systems and Technologies. *South African Computer Journal* 29, 3 (2017). http://sacj.cs.uct.ac.za/index.php/sacj/article/view/452

[21] Paul G. Savage. 1998. Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms. *Journal of Guidance, Control, and Dynamics* 21, 1 (jan 1998), 19–28.

[22] Paul G. Savage. 1998. Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms. *Journal of Guidance, Control, and Dynamics* 21, 2 (mar 1998), 208–221.

[23] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. 2017. End-to-End Learning of Driving Models from Large-Scale Video Datasets. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 3530–3538.

[24] Takuto Yoshida, Junto Nozaki, Kenta Urano, Kei Hiroi, Takuro Yonezawa, and Nobuo Kawaguchi. 2019. Gait Dependency of Smartphone Walking Speed Estimation Using Deep Learning. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19)*. ACM, New York, NY, USA, 641–642.

[25] W. Zhang, X. Li, D. Wei, X. Ji, and H. Yuan. 2017. A foot-mounted PDR system based on IMU/EKF+HMM+ZUPT+ZARU+HDR+compass algorithm. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 1–5.