Title

Step Length Is a More Reliable Measurement Than Walking Speed for Pedestrian Dead-Reckoning*

ISBN
979-8-3503-2011-4

Authors
Elyasi, Fatemeh
Manduchi, Roberto

Publication Date
2023-09-01

DOI
10.1109/ipin57070.2023.10332483

Step Length Is a More Reliable Measurement Than Walking Speed for Pedestrian Dead-Reckoning*

Fatemeh Elyasi
Department of Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, USA
felyasi@ucsc.edu

Roberto Manduchi
Department of Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, USA
manduchi@ucsc.edu

***Abstract*—Pedestrian dead reckoning (PDR) relies on the estimation of the length of each step taken by the walker in a path from inertial data (e.g. as recorded by a smartphone). Existing algorithms either estimate step lengths directly, or predict walking speed, which can then be integrated over a step period to obtain step length. We present an analysis, using a common architecture formed by an LSTM followed by four fully connected layers, of the quality of reconstruction when predicting step length vs. walking speed. Our experiments, conducted on a data set collected by twelve participants, strongly suggest that step length can be predicted more reliably than average walking speed over each step.**

***Index Terms*—Pedestrian dead reckoning (PDR), Smartphone inertial data, Step length estimation, Walking speed prediction**

I. Introduction

Self-localization is a critical component of applications that provide guidance and location-based information or that monitor the movement of individuals (e.g., patients in a care facility). In this paper, we focus on pedestrian self-localization via inertial odometry, using the sensors commonly found in commodity smartphones.

Strapdown inertial odometry is achieved through some form of dead-reckoning from sensor data [1]. Standard dead-reckoning involves double integration of data from the accelerometers, after subtraction of the gravity vector. The direction of gravity, expressed in the sensors' frame, is tracked by integrating data from the gyros. Normally, an Extended Kalman Filter (EKF) is used to track velocity, position and orientation (as well as sensor biases) through time [2]. It is well known that noise and residual bias contribute to drift, which manifests itself in an error in orientation and length of the reconstructed trajectory that increases over time.

In order to improve the performance of these systems, one could leverage the specific characteristics of human locomotion. Walking is a succession of steps, where the gait cycle at each step is formed by a stance phase (with the foot approximately static on the ground) followed by a swing phase. By using inertial sensors attached to the walker's feet or ankles, it is possible to reset the estimated velocity during stance phases, thus greatly reducing drift (*zero-velocity updates* or ZUPT [3]). While ZUPT cannot be applied on data from smartphones worn elsewhere on one's body, foot-mounted sensors are often used to obtain reliable ground-truth measurements (as we do in Sec. III-A). Modern approaches to dead-reckoning use neural networks to learn the dynamic characteristics of the inertial data recorded by a human walker, to produce odometry results with less drift, and unaffected by the phone's orientation with respect to the walker's body [4]–[7].

A different approach (often called *pedestrian dead-reckoning*, or PDR) measures distances traversed while walking by counting the number of steps taken, and adding together the estimated length of each step. The heading orientation is computed by integrating information from the gyros and the accelerometers. Part of the appeal of PDR is in its simplicity and robustness: counting steps from inertial data is relatively simple, and any errors in step length determination will have a linear effect on the computed location (whereas an uncompensated accelerometer bias has a quadratic effect due to double integration). Several machine learning methods for the estimation of step lengths from inertial data have been proposed in the literature [8]–[11].

In this article, we study methods to improve the accuracy of distance measured via PDR. We argue that PDR does not differ fundamentally from other dead-reckoning algorithms. For example, RoNIN [5] computes the walker's velocity, which is integrated through time to obtain location. By choosing the period of time between two heel strikes (step period) as integration interval, one obtains a vector representing the displacement between two heel strikes, with magnitude equal to the step length. Similarly, given an estimate of step length, the average velocity during a step (*walking speed*) can be obtained by dividing the step length by the step period. Hence, provided that one can reliably detect individual steps, step length and walking speed are easily interchangeable. The main contribution of this work is an analysis of the ability of a standard machine learning system (an LSTM followed by four fully connected layers) to predict step length and average walking speed during a step. We collected a data set from twelve participants, who walked on different paths and with different step lengths, while inertial data was collected

*Research reported in this article was supported by the National Eye Institute of the National Institutes of Health under award number R01EY029260-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

by two smartphones worn on their bodies. Ground-truth step length measurements were obtained from foot-mounted inertial sensors and EKF-based odometry integrated with zero-velocity updates. Experimental results show that our network can predict step lengths more accurately than walking speed over all metrics considered. This result can have practical significance when designing a PDR system, and suggests that machine learning odometry systems designed to measure the walker's velocity (e.g., RoNIN [5]) may not be optimal for what concerns distance measurements.

II. RELATED WORK

Estimating step length plays a key role in the operation of PDR system. It is a challenging task as gait patterns can vary with gender, height, weight, age, and health condition [12]. Traditional step length estimation methods rely on simple models, whose coefficients are normally regressed through a calibration process. For instance, the Weinberg method exploits the relationship between the stride length and the difference of max and min values of the vertical acceleration within the stride [13]. Kim et al. [14] measured step lengths based on the average of the acceleration norm, while Ladetto [15] utilized the local variance of the acceleration signal. Linear models for step length estimation, as proposed in [16], [17], combine step frequency and the user's height (which must be measured beforehand).

Machine learning-based methods for step length estimation have been proposed in recent years. The model described in [8] is based on autoencoders. Useful features from the accelerometer and gyroscope readings are learned through stacked autoencoders in a greedy layer-wise training manner. Step length is predicted through a final regression layer using learned features. A similar approach was employed in [11], with deep belief network (DBN) for feature learning in place of autoencoders. StepNet [10] combines a traditional method with a learning-based model. In this scheme, higher-level features extracted from raw inertial data, along with the smartphone location on the body (e.g., pocket, swing, texting), are plugged into a model based on a convolutional neural network (CNN) to estimate the Weinberg gain coefficient and measure step length. LSTM in conjunction with an autoencoder model was used to predict step lengths in [9]. In this study, temporal dependencies and features embedded in raw inertial data are first extracted by an LSTM network. Subsequently, learned features along with traditional features are fed to an autoencoder to train a noise-robust encoder. Finally, a regression layer is applied to predict the step length.

III. METHOD

A. Data Collection

In this section, we describe our methodology for collecting a representative data set, and for obtaining ground-truth annotations. We designed the study so as to record inertial data from a variety of stride lengths, where the length of each stride is accurately measured. Note that the common practice (e.g., [8], [10], [18]) of measuring step lengths by dividing a path length by the number of steps taken in that path, may produce inaccurate results, considering that, during "natural" walking, a person's step length has non-negligible variations (standard deviation of step length between 3% and 7% of its average as measured in [18]).

Twelve participants (6 female and 6 male, average age: 35.6) walked over four paths (except for participant P10 who walked over three paths only) in an office building. Each path was divided into a number of sub-paths, beginning and ending at marked locations. For each sub-path, one of three possible categories of stride length was prescribed: "natural", shorter than natural, or longer than natural. No other directions were given to the participants, who were free to choose their walking pace for a prescribed stride length category. The sub-paths were chosen such that the overall length traversed was the same across stride length categories (equal to 236 meters, for a total traversal length of 709 meters per participant.) The number of steps taken by the participants varied from 789 (P10) to 1344 (P9).

Each participant carried two inertial IMU packages (Xsens DOT), each tied to either shoe using an elastic band. The IMUs produced data from 3 accelerometers and 3 gyros at the rate of 120 samples/s with 16 bits resolution. In addition, each participant carried two smartphones (iPhone 13 Pro and iPhone X), one tucked in a pants back pocket, the other held by hand at about chest height, as if the participant were looking at the smartphone's screen. The smartphones ran an app that collects time-stamped inertial data at a rate of 120 samples/s. The IMU packages and the smartphones were synchronized to a common time scale.

The inertial data recorded from each foot sensor was processed using an EKF-based dead-reckoning algorithm with zero-velocity updates to obtain accurate measurements of *stride length*, defined as the path traversed between two consecutive heel strikes of the same foot. The EKF odometry algorithm tracked sensor biases along with attitude errors, velocity errors, and position errors [19], [20]. We applied the ZUPT algorithm [3] along with HDR correction [21] to reduce gyro drift. In prior experiments, we verified that this algorithm produced distance errors over long paths that were consistently less than 1% of the total traversed distance in the path. In order to measure stride lengths, we first detected each heel strike as the highest peak of the accelerometer magnitude within a window of 0.5 s around the beginning of a stance period. Stance periods were identified based on three conditions (magnitude of acceleration and of gyro readings, local acceleration variance) as described in [20]. Strides were measured by the path traversed between two consecutive heel strikes (shown as circles in Fig. 1 (a)). Note that the ZUPT algorithm performs a "correction" of the estimated location during a stance period.

Fig. 1 (b) shows (in pink) the distribution of stride lengths measured for all participants in our data collection (standard deviation: $\sigma = 0.38$ m). The same figure shows (in green) the distribution of stride lengths in the data set described in [9], also collected using foot-mounted sensors. Note this prior data

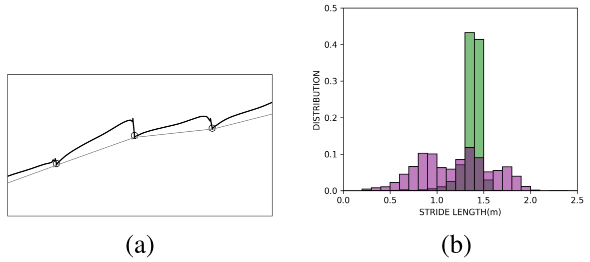


Fig. 1. (a) An example of trajectory reconstructed for one foot-mounted sensor (black line). Note that the ZUPT algorithm applies a correction at each detected stance phase. The circles represent heel strike times, which are used to compute individual stride lengths (shown by grey arrows). (b) Distribution of stride lengths over all participants of our data collection (pink bars), shown together with the distribution of stride lengths for the data set of [9] (green bars).

was built from walkers adopting a rather uniform stride length ($\sigma = 0.12$ m). We believe that a wide distribution of stride lengths is paramount for a genuine assessment of an odometry algorithm.

While foot-mounted sensors are ideally suited to measure stride lengths, when processing inertial data from a smartphone it is more convenient to measure step lengths, where a *step* is the path traversed by the user's body between two consecutive heel strikes from opposite feet, as the data recorded by the phone is approximately periodic over steps. During a step, one of the two feet is in stance phase, while the other is in swing phase. We define the ground-truth step length $\hat{l}_i$ by taking the mean of the two overlapping stride lengths (one per foot), and dividing the result by 2. Along with the step length $\hat{l}_i$, we measure the (average) walking speed during the same step as $\hat{v}_i = \hat{l}_i/\hat{T}_i$, where $\hat{T}_i$ is step period (time between two consecutive heel strikes from opposite feet).

Fig. 2 shows scatterplots of step period $\hat{T}_i$ vs. step lengths $\hat{l}_i$ for six of our participants. The figure also shows loci of constant walking speed. Note how these distributions are different across participants. For example, P1 and P9 kept an almost constant step period for different step lengths (resulting in large variations of walking speed). P4 and P12 adopted different step periods for different sub-paths with prescribed "shorter than normal" stride lengths, such that the walking speed was very different for the same step length.

B. Algorithms

The goal of our system is to estimate either the length l_i or the walking speed v_i during each step, based on inertial data recorded by each smartphone. We used exactly the same architecture for both estimations (l_i and v_i), and compared results using similar metrics.

Following [9], we used a 1-layer LSTM network [22], with 64 hidden units, followed by four fully connected layers with ReLU activation (see Fig. 3 (a)). A recurrent network appears to be the most natural choice for this type of quasi-periodic signal. The input to the network at each time is the vector formed by the 3 accelerometer and 3 gyro measurements. Inspired by [5], we normalize the orientation of these vectors

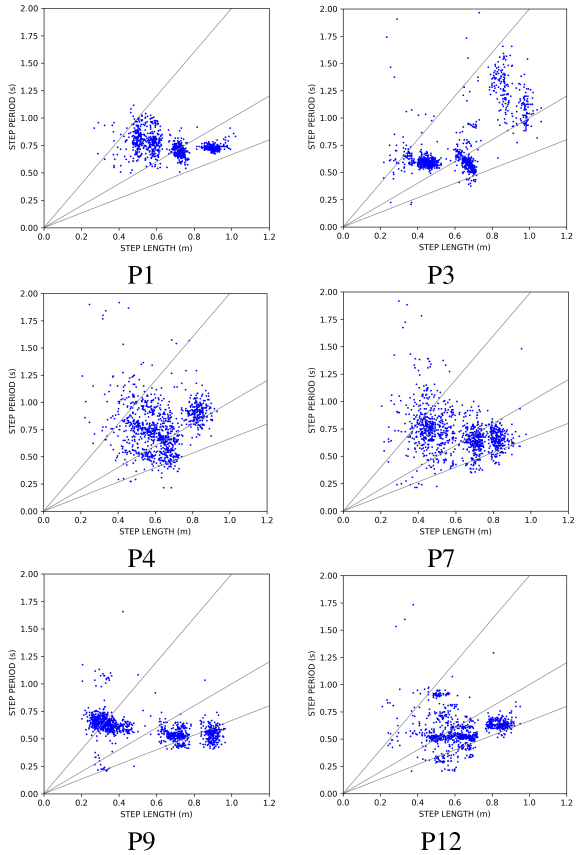


Fig. 2. Step period vs. step length for six participants in our study. Loci of constant walking speed (0.5 m/s, 1 m/s, and 1.5 m/s) are shown by gray lines.

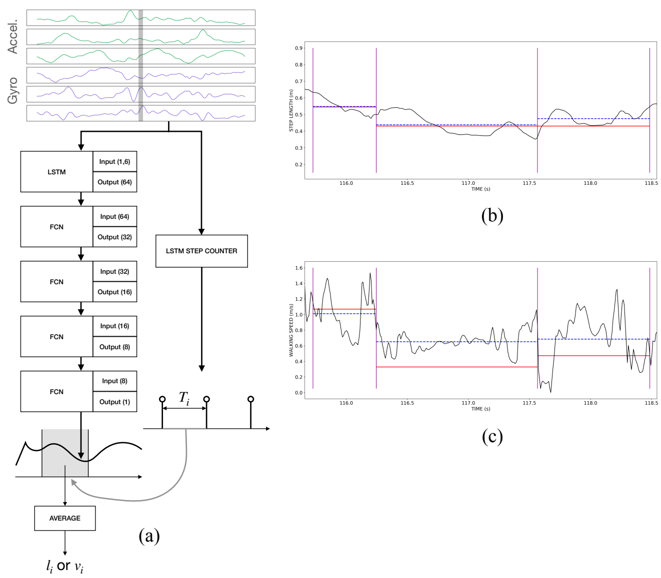


Fig. 3. (a) The architecture of the network predicting step length l_i or walking speed v_i. (b),(c): Black line: Output of the network predicting step length (b) or walking speed (c). Vertical lines: Detected heel strikes. Red segments: ground-truth values. Blue dashed segments: Average output values in a step period. Note that the participant was taking a turn in the path, resulting in significantly reduced walking speed during the second and third steps.

by pre-multiplying them by the inverse of the attitude matrix (which is provided by the iPhone API). Heel strike times (signaling the beginning and end of each step) are detected by another LSTM network [23], which is the uni-directional version of the one proposed in [24]. Unlike [9], we don't reset the state of the LSTM at the beginning of each step, but let it run continuously. Arguably, this may allow the network to better adapt to the periodic variation of the input data. During training, we fed the LSTM with segments of fixed length of 240 samples. Unlike [9], we don't restrict these segments to be contained within individual strides (which may require zero-padding for short stride periods), but sample these segments from anywhere in the signal. Specifically, we segment the input signal into intervals of 240 samples with overlap of 120 samples. For each segment, a quadratic loss is defined on the difference between the last output value produced by the network, and the ground-truth step length $\hat{l}_i$ or walking speed $\hat{v}_i$ associated with the step that contains the end point of the input segment. We found that this approach provides consistently better results than by selecting input segments only from within individual step periods. Training was performed in Keras, with batch size of 128 over a total of 500 epochs. To prevent overfitting, training was stopped if no loss reduction was measured over 50 epochs.

At deployment, the network produces one output sample per input sample. Fig. 3 shows examples of the output of the network computing step length (b) and walking speed (c), along with detected heel strike times. The plots also show the ground-truth values. Note from the figure that the network output has large variations from sample to sample. Rather than picking one value from the output (e.g., at heel strike times), we compute the average value over each step period to produce the quantity of interest (step length or walking speed, as shown by blue dashed segments in Fig. 3).

IV. RESULTS

We ran experiments using the leave-one-person-out modality: for each participant being tested, the network was trained with data from the other 11 participants. In order to evaluate the role of phone placement, we considered four scenarios: (1) train and test with data from in-hand phone ($H \rightarrow H$); (2) train and test with data from in-pocket phone ($P \rightarrow P$); (3) train with data from both phone placements, test with data from in-hand phone ($HP \rightarrow H$); (4) train with data from both phone placements, test with data from in-pocket phone ($HP \rightarrow P$). In each case, we trained two networks: one to estimate step lengths $\{l_i\}$, and one to estimate walking speeds $\{v_i\}$ at each step. The step periods T_i are computed by an LSTM step counter [23], and are identical for the two measurements.

A. Error Metrics

Remember that $\hat{l}_i$ represents the ground-truth length of the i-th step in the test set for a certain participant. We define the following error metrics for the estimated step lengths (where the first and third metrics are from [9]):

- $E_d = \frac{|\sum_{i=1}^N l_i - \sum_{i=1}^N \hat{l}_i|}{\sum_{i=1}^N \hat{l}_i}$
- $E_s = \frac{1}{N} \sum_{i=1}^N |l_i - \hat{l}_i|$
- $E_{sr} = \frac{1}{N} \sum_{i=1}^N \frac{|l_i - \hat{l}_i|}{\hat{l}_i}$
- $R^2 = 1 - \frac{\text{RMSE}^2}{\sigma^2}$, where $\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (l_i - \hat{l}_i)^2}{N}}$ and σ^2 is the variance of the set of ground-truth step lengths $\{\hat{l}_i\}$.

E_d, the relative distance error, is relevant for long paths, where step-to-step error fluctuations cancel out. E_s is the average absolute error at each step, while E_{sr} normalizes errors with the ground-truth step length. R^2, the *coefficient of determination*, is a number that is ≤ 1 (it reaches 1 only in case of zero error). A negative R^2 means that using a constant value, equal to the average step length, would yield a lower RMSE error than the predictions $\{l_i\}$.

Tab. I shows the errors measured for the network predicting step lengths l_i. For these and other measurements, each error metric is computed for each participant, then averaged over all participants. We also report standard deviations computed across participants. The lowest errors are obtained for the $H \rightarrow H$ phone placement configuration. Errors increase for the $P \rightarrow P$ configuration. Training the network with data from both phones ($HP \rightarrow H$ and $HP \rightarrow P$) is shown to decrease performance further. Importantly, the coefficient of determination R^2 is always positive, and reaches a value of 0.76 for the $H \rightarrow H$ configuration.

TABLE I
ERROR METRICS COMPUTED FOR ALL PHONE PLACEMENT CONFIGURATIONS FOR THE NETWORK PREDICTING STEP LENGTHS l_i.

	Step Length			
	E_d	E_s (m)	E_{sr}	R^2
$H \rightarrow H$	0.02 ± 0.02	0.06 ± 0.01	0.10 ± 0.02	0.76 ± 0.12
$P \rightarrow P$	0.05 ± 0.06	0.07 ± 0.03	0.12 ± 0.04	0.61 ± 0.32
$HP \rightarrow H$	0.05 ± 0.03	0.07 ± 0.02	0.12 ± 0.03	0.68 ± 0.19
$HP \rightarrow P$	0.06 ± 0.06	0.08 ± 0.03	0.13 ± 0.04	0.59 ± 0.31

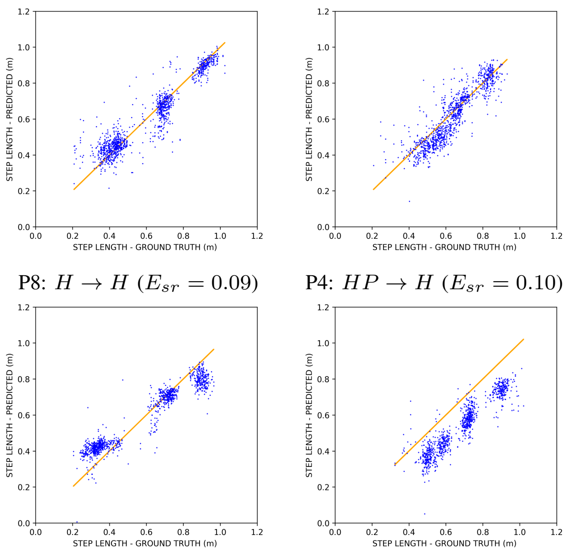
TABLE II
ERROR METRICS COMPUTED FOR ALL PHONE PLACEMENT CONFIGURATIONS FOR THE NETWORK PREDICTING WALKING SPEED v_i.

	Equivalent Step Length From Walking Speed				Walking Speed	
	E_d	E_s (m)	E_{sr}	R^2	E_s (m/s)	R^2
$H \rightarrow H$	0.04 ± 0.02	0.11 ± 0.02	0.20 ± 0.03	0.05 ± 0.39	0.25 ± 0.12	0.24 ± 0.27
$P \rightarrow P$	0.09 ± 0.05	0.13 ± 0.04	0.22 ± 0.08	-0.38 ± 1.22	0.23 ± 0.05	0.22 ± 0.26
$HP \rightarrow H$	0.09 ± 0.04	0.12 ± 0.02	0.22 ± 0.04	-0.14 ± 0.71	0.27 ± 0.12	0.14 ± 0.29
$HP \rightarrow P$	0.08 ± 0.07	0.13 ± 0.04	0.23 ± 0.06	-0.50 ± 1.22	0.26 ± 0.10	0.12 ± 0.25

For the network predicting walking speed v_i, we report in Tab. II error measures computed on *equivalent step lengths* $l_i = v_i \cdot T_i$ (where T_i is computed by the step counting network). In other words, equivalent step lengths are obtained by integrating the predicted walking speed over a step period. In addition, we report error metrics E_s and R^2 for the walking speed v_i itself when compared against the ground-truth $\hat{v}_i = \hat{l}_i / \hat{T}_i$. The results are substantially worse than when predicting step length directly. For example, the relative distance error E_d for equivalent step length increased by 80% ($HP \rightarrow H$), compared to predicting step length directly. The coefficient of determination R^2 for equivalent step length is in most cases negative, implying that taking the mean value would give a better prediction in terms of mean square error. Paired t-tests revealed that, for all metrics considered, there was a significant difference in error between predicted step length and equivalent step length ($p < 0.02$).

Since equivalent step lengths are computed by integrating predicted walking speed over measured step period (which is computed by an LSTM step counter [23]), it is possible, in principle, that errors in equivalent step length may be due, at least in part, to inaccurate step period computation, rather than inaccurate walking speed prediction. To test this hypothesis, we re-computed the equivalent step lengths by integrating the predicted walking speed over ground-truth step period (as determined by the foot sensors). The corresponding error values were not significantly different from those obtained using step periods from the LSTM step counter.

Examples of step length predictions are shown in the scatter-plots of Fig. 4 for different participants and phone placement configurations. Fig. 5 shows, for the same input data, results of walking speed estimation v_i.



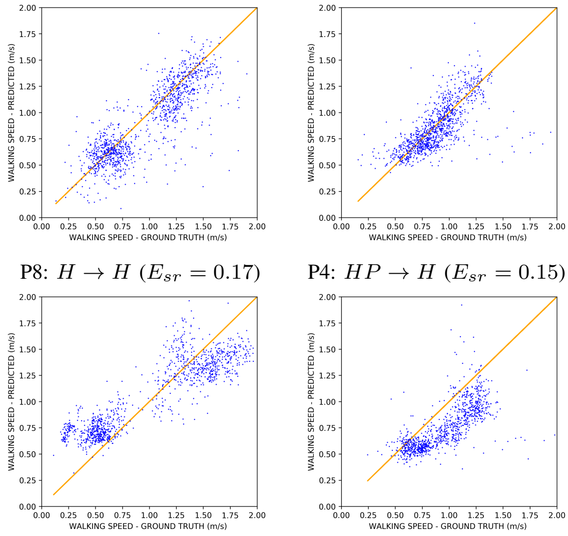
P8: $H \rightarrow H$ ($E_{sr} = 0.09$) P4: $HP \rightarrow H$ ($E_{sr} = 0.10$)

P9: $P \rightarrow P$ ($E_{sr} = 0.16$) P1: $HP \rightarrow P$ ($E_{sr} = 0.22$)

Fig. 4. Examples of step length prediction, plotted against their ground-truth values.

B. Comparison with RoNIN

In order to assess the quality of our LSTM-based algorithm, we compared its result against those obtained using RoNIN [5], a state-of-the-art pedestrian odometry system, when applied to the same iPhone sensor data. RoNIN is designed to compute the user's walking speed with respect to a fixed reference frame. We integrated the walking speed data from RoNIN over each step period as measured by our LSTM step counter model [23]. We then extracted the length of the resulting vector, and compared it with the ground-truth length from the foot sensor. We utilized the authors' open-source implementation (https://github.com/Sachini/ronin) and opted



P8: $H \rightarrow H$ ($E_{sr} = 0.17$) P4: $HP \rightarrow H$ ($E_{sr} = 0.15$)

P9: $P \rightarrow P$ ($E_{sr} = 0.39$) P1: $HP \rightarrow P$ ($E_{sr} = 0.23$)

Fig. 5. Examples of walking speed predictions, plotted against their ground-truth values.

for the RoNIN resnet18 architecture. Data was up-sampled from the original acquisition rate of 120 Hz to 200 Hz (the sampling rate used for RoNIN design). Following [23], we regressed a scale factor α by minimizing the mean squared error between the step length from RoNIN, multiplied by α, and the ground-truth step length ($\alpha = 1.15$ and 1.24 for in-hand and in-pocket phones data, respectively). As shown in Tab. III, our system produced comparable errors to RoNIN for walking speed. Predicted step length from our system had substantially lower error when compared with equivalent step length from RoNIN.

TABLE III
ERROR METRICS COMPUTED FOR TWO PHONE PLACEMENT CONFIGURATIONS USING RONIN

	Equivalent Step Length From Walking Speed				Walking Speed	
	E_d	E_s (m)	E_{sr}	R^2	E_s (m/s)	R^2
H	0.08 ± 0.06	0.12 ± 0.03	0.23 ± 0.08	-0.05 ± 0.37	0.27 ± 0.14	0.16 ± 0.31
P	0.08 ± 0.06	0.11 ± 0.05	0.19 ± 0.07	-0.27 ± 1.06	0.23 ± 0.11	0.22 ± 0.31

C. Model Performance on a Different Data Set

For comparative assessment, we re-trained and tested our step length prediction algorithm on the data set of [9], using 10-fold cross-validation. We obtained almost identical results to [9] for what concerns the mean E_{sr} error rate, and a substantial reduction in the mean E_d error for both models considered in [9] (LSTM and LSTM-DAE). Specifically, our system led to a reduction of E_d from 0.05 to 0.01 when compared to their LSTM model, and from 0.04 to 0.01 when compared to LSTM-DAE.

V. DISCUSSION AND CONCLUSIONS

PDR systems for the reconstruction of odometry from inertial data from a smartphone measure the distance traversed in a path by detecting individual steps and estimating the length of each step. Assuming that steps can be detected reliably, one may choose to predict, for each step, the actual step length, or the average walking speed. In the literature, these two tasks have been investigated using different approaches: step length prediction is typically performed with data from individual steps, while velocity prediction (e.g. [5]) is normally computed at a higher rate, then integrated over time to obtain position. In this contribution, we use the same computational architecture to measure step length and average walking speed over each step. We created a carefully annotated data set from participants walking with a wide variety of stride lengths, then trained and tested the system using a leave-one-person-out strategy. Our results strongly suggest that predicting step length provides more reliable results than predicting average walking speed.

More research is needed to provide a clear explanation of this phenomenon. One may argue that the body dynamics involved in walking (as measured by the inertial sensors of the smartphone) may have a stronger dependence on the length of strides than on the pace rate. The distribution of step length and step period may also have a role in the prediction results. Some insight can be obtained by analyzing the *coefficient of variation (CV)*, that is, the standard deviation divided by the mean, for the quantities considered (step length, step period, and their ratio, that is, the average walking speed in each step). We computed values of CV for each participant and for each prescribed stride length, then averaged them over participants. Note that we only considered ground-truth data for this analysis. We found that the CV values for step lengths (equal to 0.12, 0.09, and 0.08 for short, natural, and long strides, respectively) are substantially lower than those obtained for step periods (0.35, 0.42, 0.29) and, consequently, for walking speed (0.46, 0.54, 0.25). Note that the CV values for step length are slightly higher than those reported in [18], probably owing to the different experimental settings. Arguably, a recurrent network tasked with tracking a quantity that is locally "stable" (step length) may have an easier job than one predicting a quantity that varies more widely from step to step (walking speed).

Our results also showed (Tab. I) that better prediction is obtained when training and testing on data from a phone placed in the same location on one's body. This information could be useful when a mechanism to detect the phone's placement (or *context*) is implemented (e.g. [25]). Knowledge of the phone's current placement could be used to select the appropriate step length predictor at each time.

We should stress that our analysis has only considered the *length*, not the direction, of steps taken while walking. Existing learning-based approaches (e.g., [4]–[6]) can be used to robustly estimate heading direction.

REFERENCES

[1] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Computer Laboratory, Tech. Rep., 2007.

[2] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "Inertial odometry on handheld smartphones," in *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, pp. 1–5.

[3] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer graphics and applications*, vol. 25, no. 6, pp. 38–46, 2005.

[4] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to cure the curse of drift in inertial odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[5] S. Herath, H. Yan, and Y. Furukawa, "RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3146–3152.

[6] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "TLIO: Tight learned inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.

[7] S. Sun, D. Melamed, and K. Kitani, "IDOL: Inertial deep orientation-estimation and localization," *arXiv preprint arXiv:2102.04024*, 2021.

[8] F. Gu, K. Khoshelham, C. Yu, and J. Shang, "Accurate step length estimation for pedestrian dead reckoning localization using stacked autoencoders," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 8, pp. 2705–2713, 2018.

[9] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and Y. Huang, "Pedestrian stride-length estimation based on LSTM and denoising autoencoders," *Sensors*, vol. 19, no. 4, p. 840, 2019.

[10] I. Klein and O. Asraf, "StepNet—Deep learning approaches for step length estimation," *IEEE Access*, vol. 8, pp. 85 706–85 713, 2020.

[11] D. Yan, C. Shi, and T. Li, "An improved PDR system with accurate heading and step length estimation using handheld smartphone," *The Journal of Navigation*, pp. 1–19, 2021.

[12] S.-u. Ko, M. I. Tolea, J. M. Hausdorff, and L. Ferrucci, "Sex-specific differences in gait patterns of healthy older adults: results from the Baltimore Longitudinal Study of Aging," *Journal of biomechanics*, vol. 44, no. 10, pp. 1974–1979, 2011.

[13] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," *Analog Devices AN-602 application note*, vol. 2, no. 2, pp. 1–6, 2002.

[14] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *Journal of Global Positioning Systems*, vol. 3, no. 1-2, pp. 273–279, 2004.

[15] Q. Ladetto, "On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," in *Proc. ION GPS 2000*.

[16] R. Chen, L. Pei, and Y. Chen, "A smart phone based PDR solution for indoor navigation," in *Proc. ION GNSS 2011), year=2011*.

[17] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.

[18] N. Sekiya, H. Nagasaki, H. Ito, and T. Furuna, "Optimal walking in terms of variability in step length," *Journal of Orthopaedic & Sports Physical Therapy*, vol. 26, no. 5, pp. 266–272, 1997.

[19] C. Fischer, P. T. Sukumar, and M. Hazas, "Tutorial: Implementing a pedestrian tracker using inertial sensors," *IEEE pervasive computing*, vol. 12, no. 2, pp. 17–27, 2012.

[20] A. R. Jiménez, F. Seco, J. C. Prieto, and J. Guevara, "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*. IEEE, 2010, pp. 135–143.

[21] J. Borenstein, L. Ojeda, and S. Kwanmuang, "Heuristic reduction of gyro drift for personnel tracking systems," *The Journal of navigation*, vol. 62, no. 1, pp. 41–58, 2009.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] R. Peng, F. Elyasi, and R. Manduchi, "Smartphone-Based Inertial Odometry for Blind Walkers," *Sensors*, vol. 21, 06/2021 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/4033/pdf

[24] M. Edel and E. Köppe, "An advanced method for pedestrian dead reckoning using BLSTM-RNNs," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2015, pp. 1–6.

[25] E. Miluzzo, M. Papandrea, N. D. Lane, H. Lu, and A. T. Campbell, "Pocket, bag, hand, etc.-Automatically detecting phone context through discovery," *Proc. PhoneSense 2010*, pp. 21–25, 2010.