A Survey on Artificial Intelligence for Pedestrian Navigation with Wearable Inertial Sensors

Hanyuan Fu, Yacouba Kone, Valérie Renaudin, Ni Zhu

A Survey on Artificial Intelligence for Pedestrian Navigation with Wearable Inertial Sensors

Hanyuan Fu, Yacouba Kone, Valerie Renaudin, Ni Zhu
Univ. Gustave Eiffel, AME-GEOLOC
F-44344 Bouguenais, France
{hanyuan.fu, yacouba.kone, valerie.renaudin, ni.zhu}@univ-eiffel.fr

***Abstract*—Miniaturized IMU (inertial measurement units) are widely integrated in wearable devices, promoting the versatile and low cost pedestrian inertial navigation technology, especially for indoor environment. In recent years, AI (Artificial Intelligence) is applied to improve the performance of this technology. AI methods work with data samples, thus it is important to select a suitable process for segmenting the inertial data sequences. This survey classifies A I m ethods f or p edestrian i nertial navigation into two categories, namely human gait driven methods and sampling frequency driven methods, according to their data segmentation process. Human gait driven methods segment the inertial measurement sequence by gait (step or stride) events and learn to infer a gait vector (step/stride length and direction) given a gait segment. Sampling frequency driven methods learn to infer the user's velocity or change in position given a fixed-length segment of inertial measurements. The survey studies the underlying assumptions and their validity of the two categories of AI methods. Two methods (SELDA and RoNIN), each from a category, are chosen for evaluation and comparison, on three testing tracks totaling 770m, covering indoor and outdoor environment, including stairs. The experiments highlight the two methods' advantages and limitations, supporting the theoretical analyses. The selected methods achieve 7m and 12m positioning errors, respectively.**

***Index Terms*—Indoor positioning, inertial sensors, pedestrian navigation, dead reckoning, Machine Learning, Deep Learning**

I. Introduction

Pedestrian positioning technology with inertial sensors is used in many daily life applications such as navigation apps and Augmented Reality games. This choice results from the lack of GNSS (Global Navigation Satellite System) signals indoors and degraded performance of GNSS positioning even in urban canyons. Indoor positioning technologies can also be based on radio beacon networks deployed in buildings. But this technology faces many problems such as high layout cost, changing indoor radio propagation context, crowd effect, human body tissue absorption, and so on. Inertial navigation systems have become more and more popular because they are infrastructure-free solutions and low cost. But their performance is degraded by inertial signals drift during a long period, lack of robustness when dealing with variable pedestrian motion modes (slow/normal/fast walking, going up or down stairs, stationary, etc.), device poses (in swinging hand, "texting", in pocket, etc.) and individual walking characteristics. Many researchers apply Artificial Intelligence (AI), namely traditional machine learning and deep learning to mitigate these issues and improve the accuracy and robustness of their models. In this context, this paper surveys AI-based approaches for pedestrian navigation with wearable inertial sensors. It shows that these approaches can be classified into the 2 following classes depending on how the inertial measurement data is aggregated to gather the most of useful information.

1) Human gait driven AI methods: Inspired by the nature of human walking, inertial signals are segmented by the user's gait (step or stride) events. Each segment can be utilized to estimate the user's step vector: length and direction.

2) Sampling Frequency driven AI methods: Inherited from the acceleration double integration approach, inertial signals are segmented into fixed length sequences, overlapping or not. AI methods are trained to infer the user's velocity or change in position given a fixed length of inertial measurements. The length of a segment partly depends on the measurement's sampling frequency.

This survey details the two-categories proposed to classify existing AI based pedestrian navigation methods with wearable inertial sensors. It analyses their principal hypotheses, advantages and limitations, both at theoretical and experimental levels. Theoretical analysis identifies the assumptions underlying the learning methods and reviews their validity. Experimental assessment is conducted in each category with one representative method selected in each category. The experimental assessment is conducted on 3 testing tracks, each is about 250 meters, including both indoor and outdoor environment, stairs, with the user holding the device in front of his chest.

Section II and III present the AI based pedestrian navigation state-of-the art methods: the human gait driven and the sampling frequency driven AI methods, respectively. The two methods selected in the aforementioned categories are detailed in section IV. Section V is dedicated to the experimental assessment, including the evaluation and comparison of the two selected methods on pedestrian tracks. Section VI concludes the survey.

II. Human gait driven AI methods

Human gait driven AI methods are tightly related to the Step-and-Heading approach: triggered by the user's gait (step

or stride) events and estimate the user's current step length and heading. Wearable inertial sensor records are segmented by gait events, which can be detected using the cyclic patterns. Detecting gait cycles from inertial signals collected in different device carrying modes (swinging, texting, in pocket, on armband,...) is challenging, since the signals of different modes have different shapes. [1] considers that a peak in the acceleration norm or a valley in the angular velocity norm represents a step instant, for all carrying modes. Thus they propose 2 parallel Histogram Gradient Boosting Decision Trees (GBDT) [2] based models. One detects acceleration peaks and the other angular velocity valleys. A decision process is designed to combine the output of the 2 models. Tested for normal and visually impaired gaits, it achieves 97% correct step detection with 4 different device carrying modes.

Once the IMU measurement signals are segmented by gait events, each segment can be utilized to estimate the current step/stride length and its direction. In the literature so far, step length and direction are estimated separately.

A. A Two-step Estimation Process: step/stride length and walking direction

1) Step or Stride length Estimation: Step length varies from person to person and even for the same person. It depends on the motion modes (normal/fast walking, going up or down stairs). Even for the same person and same motion, different device poses on the user's body result in different signal patterns. When the physical modeling becomes too complicated, Artificial Intelligence comes in handy.

One common approach is to extract features manually from a step/stride segment. Wang and al. [3] classify relevant features into statistical (mean, standard deviation, maximum, etc.), time-domain (number of peaks, zero-crossing ratio, etc.), frequency domain (dominant frequencies, spectrum energy, etc.) and higher-level features (based on empirical models). The extracted features are utilized in an ensemble regression model combining 6 regressors: Extreme Gradient Boost (XGBoost) [4], LightGBM [2], K-Nearest Neighbor (KNN) [5], Decision Tree (DT) [6], AdaBoost [7] and Support Vector Regression (SVR) [8]. Klein and al. [9] leverage feature selection to accelerate the deep learning process. They extract 60 candidate features from the input sequence, perform feature selection with Monte-Carlo runs and ridge regression, then the selected features are fed to a convolutional neural network to regress step length. Inspired by the Weinberg Model [10], Zhang et al. [11] express the acceleration in the global frame via device attitude tracking, remove the gravity component, filtered it with moving average and take the differences of peak and valley of vertical acceleration as input features to an OS-ELM (Online Sequential Extreme Learning Machine) [12] to regress step length.

An alternative to feature engineering is end-to-end regression with the step or stride sequence. Gu et al. [13] use 2 stacked autoencoders and a dense layer to learn step length from a step sequence. Yang et al. [14]'s step length model is a Deep Believe Network (DBN) built with multiple Gaussian Bernoulli Restricted Boltzmann Machines (RBM, for feature extraction) [15] and a regression layer on top. Wang et al. [16] combine the stride sequence with other empirical features as their model's input. Their model is detailed in section IV.

2) Heading estimation: In earlier PDR frameworks, heading is usually estimated by Extended Kalman Filter fusing the device's angular velocity and magnetometer readings, assuming that the offset between the device's and the user's pointing directions is approximately constant, which is the case of "texting" or "calling" modes. This assumption is not true in the "swinging" or "pocket" modes. More recently deep learning methods are applied to infer the user's heading from sequences of the IMU measurements.

Zhang et al. [11] found recognizable patterns in the sequences of the device's azimuth angle and magnetometer readings when the user changes direction. Thus, they select them as input features to their OS-ELM network to regress the user's heading angle. Wang et al. [17] proposes a hierarchical LSTM [18] architecture associated with a spatial transformer [19] to regress walking direction of each step made by a pedestrian, taking as input 3D acceleration, angular rate and magnetometer readings over a complete trajectory, leveraging the strong correlation between the headings of the adjacent steps.

B. Analysis of Underlying Assumptions

The main interest of this approach is that it's based on physical modeling of human gait. AI methods driven by realistic gait changes are close to the physical facts and thus more explainable. Gait driven AI methods are based on the following hypotheses, proven or empirical.

1) Human walking is cyclic: In [20], walking locomotion is described as a process in which the erect, moving body is supported by first one leg, and then the other. As the moving body passes over the supporting leg, the other leg swings forward to prepare for its next supporting phase.

2) The gait cycles and the movements of the upper and lower body limbs' movements are correlated: It allows gait event detection with wearable sensors. According to [21], during normal walking, head and trunk travel as a unit and move "up and down" as the center of gravity follows the mechanics of the limbs. During each stride, the arms reciprocally flex and extent.

3) Human paces are regular and constrained: According to a statistical study presented by [3], within their dataset containing 13.5km and 10145 strides of gait measurements collected by a foot mounted device, including fast, normal and slow walking, 99.5% of strides were within 1.55m and no stride exceeds 1.75m. The mean and standard deviation of stride length are 1.33m and 0.18m.

4) Step/stride length and inertial signals collected from different body parts are correlated: Empirical models are developed based on foot mounted sensors, for example Kim [22] found correlation between mean acceleration norm and stride length and Ladetto [23] found correlation between acceleration variance and stride length. The hypothesis may be reasonable

considering the correlation between foot movements and the those of the rest of the body.

Hypothesis 1 and 3 are proven in regular walking situations. Exceptions exist though, for example an elderly user may lose balance and fall down. In such case the cyclic and regular pattern of gait is corrupted. Hypothesis 2 is a simplification of the reality, especially for the arms. Users can move their arms freely while walking and detecting step under irrelevant arm/hand movements remains challenging. On the other hand, human gait driven AI models rely vitally on the quality of gait event detection. Poor gait segmentation during the training phase can result in overfitting since it prevents the model from learning relevant representations. For hypothesis 4, even if this correlation exists, it will differ for different device carrying modes (handheld, pocket, armband,...) and different user motions (normal walking, stair climbing, running, ...). Step length estimation taking into account several carrying modes is challenging. Most of the current research works either consider a single device pose ([16], [14]) or necessitate device pose classification ([3], [9]). [13] was a first attempt to estimate step length with a step segment of IMU measurements for 2 carrying modes: swinging and pocket, without classification. However, they only presented walking distance error (sometimes errors may compensate for themselves if we add them up) and they didn't show convincing generalization ability on unseen users.

III. Sampling Frequency driven AI method

Sampling Frequency driven AI methods learn to infer the user's velocity or change in position given a fixed-length inertial measurement sequence, regardless of the gait events. Inherited from the acceleration double integration approach, this branch of methods remedy the drift by piece-wise estimation or correction to stop the error propagation. In the literature, the duration of a segment is usually between 1 and 2 seconds.

A. Existing Approaches

RIDI (Robust IMU Double Integration [24]), considered as the "ancestor" of this branch, tracks the user via the device. RIDI regresses a horizontal velocity of the device given a sequence (200 frames at 200Hz) of acceleration and angular velocity expressed in a reference frame whose y-axis is aligned with the gravity. The regressed velocity is utilized to correct low frequency errors in the acceleration, such that the integration of the corrected acceleration matches the predicted velocities. The device's position is estimated by double integrating the corrected acceleration. The same group of authors later proposed RoNIN (Robust Neural Inertial Navigation [25]), that we will detail in section IV.

IONet [26], another pioneer of sampling Frequency driven AI methods, considers the true acceleration and angular rate of the user as latent variables of IMU raw measurements. The remedy proposed by IONet to reduce drift is to "break it down": inertial measurements are segmented into fixed length time windows (200 frames at 100Hz) with a stride of 10 frames, each segment is "pseudo independent" of others if we consider that the user's initial velocity of the segment can be roughly estimated from the signal's frequency, considering the regularity of human movement. Then the following mapping is possible:

$$(\boldsymbol{a}, \boldsymbol{w})_{200*6} \xrightarrow{f_\theta} (\Delta l, \Delta \psi) \tag{1}$$

where $\boldsymbol{a}$ and $\boldsymbol{w}$ are the triaxial acceleration and angular rate measurements segments, Δl is the distance traveled and $\Delta \psi$ is the user's change in heading over the same time window. the mapping f_θ is done by a 2- layer Bi-directional LSTM network, selected for its ability to handle time dependencies.

Feigl et al. [27] map a sequence (128 frames at 100Hz) of acceleration magnitude and angular rate magnitude (they call them SMV: signal magnitude vectors) to the user's velocity magnitude within the same time window, using a CNN for feature extraction and Bi-directional LSTM for regression.

B. Analysis of Underlying Assumptions

A benefit from this category is that the sampling frequency driven AI methods don't need gait analysis, which is already a complex challenge. Deep learning models always require constant input length, thus, fixed length segments are naturally suitable for deep learning. In contrast, gait driven methods need interpolation or padding to uniformize the number of measurements contained in a gait segment. They consider the following two assumptions.

1) The true kinematic of the user's center of mass is continuous and can be recovered from inertial measurements collected from different body parts: This branch of methods consider this assumption to be true.

2) Each fixed length segment is independent of the others: This means that each window of inertial signals contains sufficient information to yield a velocity or displacement estimation over the same time window.

Hypothesis 2 implies that a segment is able to yield velocity estimation without knowledge of the initial velocity, this is approximately true only if we consider a cyclic and regular movement that the velocity is correlated to the signal's frequency. Thus, the inferred velocity may be sensitive to noises, and the fixed segment length may not be ideal to handle different walking dynamics or less regular walking patterns, for example, extreme slow walking, or some elderly people who frequently lose balance. The same hypothesis also implies that a segment contains sufficient information to infer heading change over the same time window. Combettes et al. [28] did a survey on traditional methods to estimate the angular misalignment between the unconstrained device's pointing direction and the user's walking direction, namely Principal Component Analysis (PCA), [29], [30], Forward and Lateral Accelerations Modeling (FLAM) [31] and Frequency analysis of Inertial Signals (FIS) [32]. All these methods assume that the walking heading is observable with handheld inertial sensors during one step/stride, In another word, a segment of signal cut within gait cycles does not provide enough information to recover a walking heading.

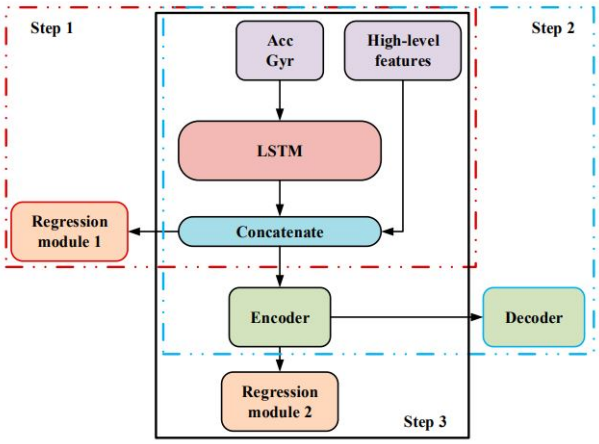


Fig. 1: Stride length model overview [16]

IV. Selected Methods for the Experimental Assessment: SELDA and RONIN

The method [16]: pedestrian Stride-length Estimation based on LSTM and Denoising Autoencoders (titled SELDA in the rest of the paper) is selected among the gait driven AI methods. The later is representative of the category with sufficient implementation and data processing details, along with a benchmarking dataset. RoNIN [25] is selected among the sampling frequency driven AI methods since the authors published their implementation, trained model weights and a part of their dataset.

A. SELDA: pedestrian Stride-length Estimation based on LSTM and Denoising Autoencoders

[16] presents a deep learning stride length estimation model taking as input stride event segmented acceleration and angular velocity readings, collected by a smartphone IMU.

1) SELDA Dataset: is publicly available [33], collected by 5 volunteers of different genders, ages and heights, using a Huawei mate 9 smartphone. Throughout the recordings, the users hold the phone horizontally in right hand in front of their chest. The stride length ground truth is provided by a foot mounted IMU module.

The dataset contained more than 22 km, 10000 strides of gait measurements in natural motions such as fast walking, normal walking, slow walking, running, jumping. It covers indoor and outdoor environment including stairs, escalators, elevators, office environments, shopping mall, streets and metro station.

2) Data preprocessing: The raw triaxial accelerometer and gyroscope readings at 100Hz are segmented by stride events. The stride segments are zero-padded to reach a uniform length of 300. 4 higher-level stride length features given by empirical models (Weinberg [10], Kim [22], Ladetto [23], Scarlet [34]) are also computed within a stride segment:

$$Weinberg = K_w \times \sqrt[4]{a_{max} - a_{min}} \tag{2}$$

where a_{max} and a_{min} are the maximum and the minimum *vertical accelerations*.

$$Kim = K_k \times \sqrt[3]{\frac{\sum_{i=1}^{N} |a_i|}{N}} \tag{3}$$

where $|a_i|$ represents the acceleration magnitude of the i-th sample.

$$Ladetto = \alpha \times f + \beta \times v + \gamma \tag{4}$$

where f is the stride frequency and v is the variance of acceleration magnitude.

$$Scarlett = K_s \times \frac{\frac{\sum_{i=1}^{N} |a_i|}{N} - a_{min}}{a_{max} - a_{min}} \tag{5}$$

where $\frac{\sum_{i=1}^{N} |a_i|}{N}$ is the average acceleration magnitude and a_{max} and a_{min} are the maximum and the minimum acceleration magnitudes.
We estimate the parameters K_w, K_k, K_s, α, β and γ by lineair regression on the whole training set.

3) Stride Length Estimation: The algorithm comprises 3 steps (Fig. 1).

Feature extraction by LSTM. The input is fed to an LSTM based networks containing 2 parallel LSTM layers, one processes the accelerometer sequence and the other, gyroscope sequence. Feature extracted from accelerations and angular velocities as well as higher-level stride length features are concatenated and fed to 4 consecutive dense layers to regress a stride length. The goal of the first phase is to obtain a LSTM feature extractor.

Denoising the features by autoencoder. Once the LSTM network is trained, We only keep the feature extracting layers and discard the regression layers. To address the poor signal quality of the smartphone sensors, an autoencoder is built. The concatenated features extracted by LSTM is fed to a dropout layer to get a corrupted version of it, then the corrupted input goes through the encoder, the decoder reconstructs the input features from the encoder's output. We chose dropout rate = 0.3. The autoencoder is trained in an unsupervised manner minimizing the error between the reconstructed input and the original input.

Stride length regression. Once the autoencoder is trained, we retrieve features extracted by its encoder, as the input for the final stride length regression. The stride length is regressed by 3 consecutive dense layers. All layers are fine-tuned with the regression.

4) Adaptation for experimental assessment: We implement the SELDA model according to [16]. 35 higher-level features are utilized in the article, but definitions of only 4 among them are available, thus we decide to use only 4 higher-level features. We train the SELDA model with SELDA dataset.

To obtain the user's trajectory, we pile up 3 modules namely step detection, stride length estimation (SELDA) and heading estimation to build a complete pedestrian dead reckoning (PDR) algorithm. Since we are only interested in SELDA's

performance, we use the stride detection result and the user's heading at each stride event, all provided by the reference foot mounted tracker. More detail about our foot mounted tracker can be found in section V.

B. RoNIN: Robust Neural Inertial Navigation

RoNIN is a state-of-the-art pedestrian positioning algorithm using smartphone IMU measurements. Both RoNIN dataset, model implementation and trained model weights are publicly available [35].

1) RoNIN dataset: contains 42.7 hours of IMU data collected in 3 buildings, by 100 participants and 3 android devices. Usual device carrying mode and human activities are considered (smartphone in a bag, in pocket, handheld, walking, sitting). A sample track contains synchronised accelerometer and gyroscope readings, game rotation vector provided by android API and ground truth user trajectories and orientations, provided by visual-inertial SLAM using a tango phone, placed on the participant's chest. All measurements are synchronized and sampled at 200Hz. A json file contains sensor biases, scale factors and necessary information for the spatial alignment procedure.

2) Data pre-processing: To use the RoNIN networks, acceleration and angular rate sequences must first be expressed in a navigation frame.

At the beginning of each recording, an "alignment" procedure is performed between the IMU device and the tango phone, by attaching them together screen to screen during a static period. The alignment procedure allows to estimate the rotation quaternion from the device's body frame to the tango phone's body frame at the beginning of the recording: $q_{imu\_to\_tango}$. This quaternion is available in the json files of RoNIN dataset as part of the input. We also need the tango phone's orientation (w.r.t the navigation frame) at the beginning of the sequence $q_{init\_tango\_ori}$. Using these 2 quaternions, we can estimate the initial orientation of the IMU device w.r.t the global frame $q_{init\_imu\_ori}$.

$$q_{init\_imu\_ori} = q_{imu\_to\_tango} \otimes q_{init\_tango\_ori} \tag{6}$$

For the rest of the sequence, the device's orientation will be estimated from its game rotation vector (GRV) [36] q_{grv}. The game rotation vector, provided by android API, estimated only using accelerometer and gyroscope readings, indicates the device's orientation w.r.t some reference coordinate frame whose z axis is aligned with the gravity, and whose horizontal axes are not necessarily aligned with the axes of the navigation frame. If this unknown reference frame is almost fixed (if we ignore the drift), the angular offset between the navigation frame and the game rotation vector's reference frame, denoted q_{init_rotor}, can be estimated at the beginning of the sequence:

$$q_{init\_rotor} = q_{init\_imu\_ori} \otimes q *_{grv} (t = 0) \tag{7}$$

Using this angular offset and the game rotation vector, we can express the IMU device's orientation quaternion in real time:

$$q_{imu\_ori} = q_{init\_rotor} \otimes q_{grv} \tag{8}$$

We can now express acceleration and angular rate measurements in the navigation frame, using the orientation quaternion of the IMU device:

$$q_{nav\_acc} = q_{imu\_ori} \otimes q_{acc} \otimes q*_{imu\_ori} \tag{9}$$

$$q_{nav\_gyro} = q_{imu\_ori} \otimes q_{gyro} \otimes q*_{imu\_ori} \tag{10}$$

where q_{acc} and q_{gyro} are pure quaternions, their vector parts are 3D acceleration or angular rate measurements.

3) Importance of the Game Rotation Vector: Neither published paper nor implementation about the game rotation vector was available. To understand it, we compare the attitude (roll-pitch-yaw w.r.t the local North-East-Down) computed by MAGYQ ([37]), an EKF based algorithm, fusing acceleration, angular rate and magnetometer readings, even under magnetic disturbances, and the ones given by the game rotation vectors. We run the following experiment: ULISS (see section V) and a Xiaomi Mi8 phone are attached rigidly on a aluminium plate, oriented in such way that their z axes are parallel (see Fig 2). We consider 2 scenarios.

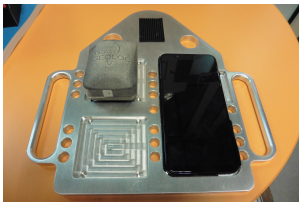


Fig. 2: experimental setup for studying the game rotation vector

- Track 1: slow and steady rotations.
- Track 2: random rotations during walking.

In Fig 3a and Fig 3b respectively, the top figure shows Euler angles of the smartphone computed from android game rotation vector, and the middle figure shows Euler angles of ULISS, computed by MAGYQ. We observe that the game rotation vector's roll angle and ULISS's pitch angle seem to have the same pattern, same observation for game rotation vector's pitch angle and ULISS roll angle. In the bottom figure, we plot the 3 curves, representing the variations of angular offsets between Euler angles given by the 2 algorithms. The under-script "a" stands for android and "u" stands for ULISS.

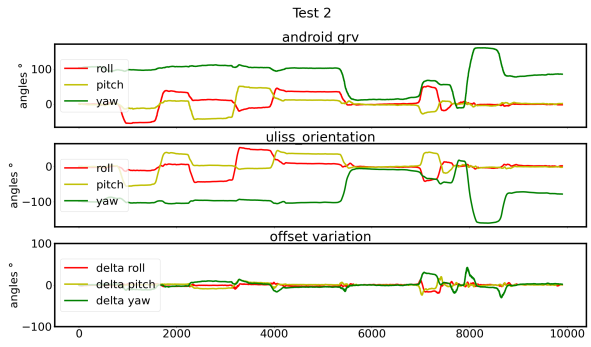
$$\Delta_{roll} = roll_u - roll_u(t=0) - (pitch_a - pitch_a(t=0)) \tag{11}$$

$$\Delta_{pitch} = pitch_u - pitch_u(t=0) - (roll_a - roll_a(t=0)) \tag{12}$$

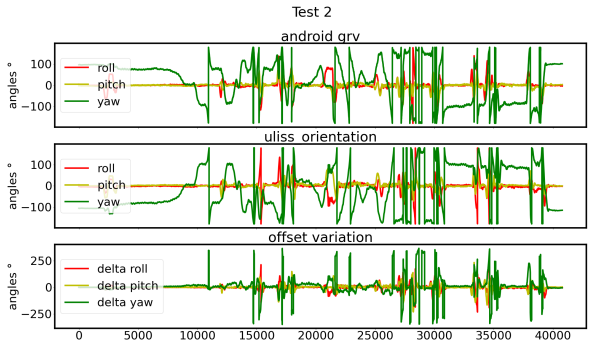
$$\Delta_{yaw} = yaw_u - yaw_u(t=0) + (yaw_a - yaw_a(t=0)) \tag{13}$$

The figures show that the game rotation vector is good at estimating roll and pitch (related to the gravity), there is no huge difference between game rotation vector and MAGYQ result. The offset between game rotation yaw and ULISS yaw is almost constant during several minutes of recordings.

We observe punctual peaks in the offset variation plot, which are due to the singularity points in both game rotation vector and ULISS attitude.



(a) Track 1: Euler angles for slow and steady rotations



(b) Track 2: Euler angles during walking

Fig. 3: Comparison of attitude angles estimated by android game rotation vector and MAGYQ for two scenarios

We can conclude that the offset between the game rotation vector and the device's orientation w.r.t to North-East-down frame is approximately constant during several minutes.

4) Model: The authors compares 3 variants based on different deep learning models: ResNet [38], LSTM [18] and Temporal Convolutional Network (TCN) [39] to estimate the user's position. We only assess RoNIN ResNet, since it yields the best results in the article.

RoNIN ResNet regresses 2D velocity vectors (Vx, Vy). The loss function aims at minimizing the error between the regressed velocity and the difference of positions in the ground truth over a window of 200 frames (1s). At test time, the network makes prediction with a stride of 5 frames and integrate the inferred velocities to estimate trajectories.

5) Adaptation for experimental assessment: We use the article's published model implementation and trained model weights (RoNIN ResNet) for experimental assessment, but we couldn't follow the alignment procedure described in the article, instead we rotate the predicted trajectories horizontally to match the user's true initial heading.

To make the algorithm more transparent, we estimate the device's orientation with MAGYQ instead of using the game rotation vector, since the offset between them is approximately constant during several minutes.

(a) (b)

Fig. 4: (a) ULISS sensor; (b) experimental setup: one ULISS sensor is held in the user's right hand and the other on the user's right foot

V. Experimental Assessment

A. Experimental setup

We use 2 ULISS sensors [40], one as a wearable device to collect IMU and magnetometer readings, the other is attached to the user's foot to obtain stride instants, ground truth stride length and trajectories. ULISS (Fig 4(a)) is a state-of-the-art Inertial Navigation System containing an Xsens Mit-7 IMU-Mag sensor and a GNSS receiver, providing triaxial accelerator, gyroscope and magnetometer readings at 200Hz, GNSS signal at 5Hz, using GPS timestamps.

During each recording, the user holds the device horizontally (in such way the z axis points to the sky, Fig 4(b)), and walks naturally. This configuration is required by SELDA. In contrast, RoNIN can operate under less constrained conditions. All recordings started in an outdoor environment.

The test user is a 1.66m tall healthy man and 3 testing tracks are recorded. Test 1 and 3 correspond to 251m and 288m respectively. The user walks in both outdoor and indoor environment, including stairs. Test 2 is 230m long. The user walks on an outdoor horizontal plan.

B. Performance evaluation

We chose 3 performance metrics to evaluate the horizontal trajectories estimated by the selected methods: scale factor (SF), Endpoint error rate (EPR) and Root Mean Square Error (RMSE). The scale factor is the ratio of the total length of estimated trajectory l_{es} to the total length of the ground truth trajectory l_{gt}. We expect the ratio to be close to 1.

$$SF = \frac{l_{es}}{l_{gt}} \tag{14}$$

The endpoint error is the difference between estimated position $(\hat{x_{end}}, \hat{y_{end}})$ and the ground truth position (x_{end}, y_{end}) at the end of the trajectory. The endpoint error rate is the ratio of endpoint error to the ground truth trajectory's total length.

$$endpoint\_error = \sqrt{[(x_{end} - \hat{x_{end}})^2 + (y_{end} - \hat{y_{end}})^2]} \tag{15}$$

$$ERP = \frac{endpoint\_error}{l_{gt}} \tag{16}$$

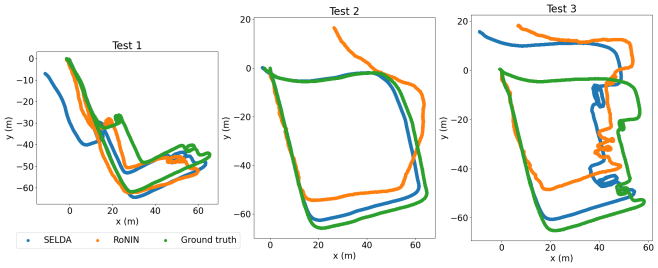


Fig. 5: Estimated trajectories for SELDA (blue), RONIN (orange) and the ground truth (green)

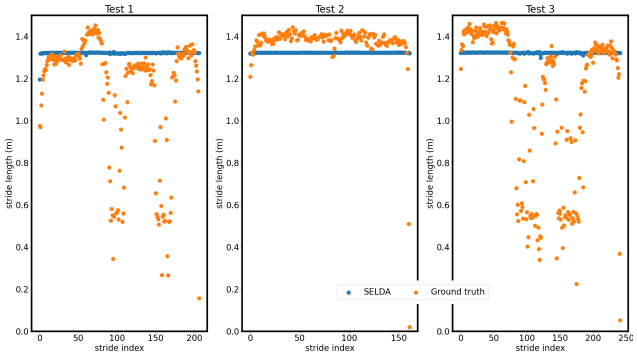


Fig. 6: Stride length predicted by SELDA (blue) and the ground truth (orange)

RMSE mesures the standard deviation on horizontal positions.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[(x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2]} \qquad (17)$$

where n is the sequence's length, (x_i, y_i) is the user's ground truth position at time step i and $(\hat{x_i}, \hat{y_i})$ is the predicted one. The experimental results are reported in table I. Estimated and ground truth trajectories are shown in Figure 5.

	SELDA			RoNIN		
	SF (unitless)	EPR (%)	RMSE (m)	SF (unitless)	EPR (%)	RMSE (m)
test 1	1.08	4.85	7.12	0.97	0.25	5.30
test 2	0.92	0.40	2.63	0.82	14.4	12.96
test 3	1.10	6.03	11.66	0.86	6.64	18.09
average	1.03	3.76	7.14	0.88	7.10	12.12

TABLE I: Performance evaluation of SELDA based PDR and RoNIN

C. Analysis

Scale factor (SF) is the most important among the 3 chosen metrics, since SELDA only estimates stride length. SELDA overestimates twice and underestimate once the walking distance, with an average scale factor of 1.03. RoNIN always underestimates the walking distance, with an average scale factor of 0.88. The standard deviations of their scale factors are 0.083 and 0.067 respectively, which means that RoNIN better

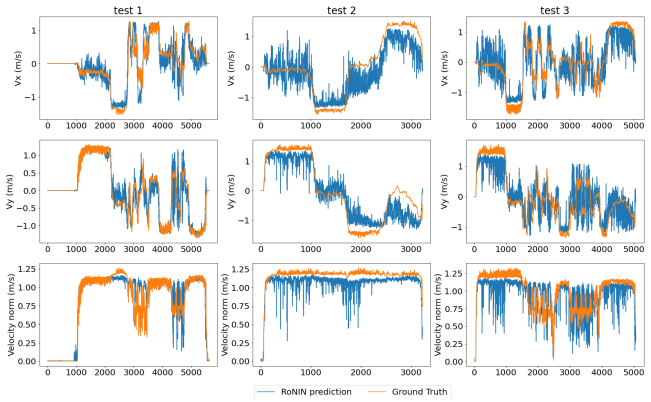


Fig. 7: Velocity predicted by RoNIN (blue) and the ground truth velocity (orange)

tracks the variations of the user's dynamics. On the other hand, important drifts are observed in RoNIN trajectories for test 2 and 3. To better understand these observations, we plot the stride lengths estimated by SELDA against the ground truth in Fig 6, and the predicted velocity against the ground truth in Fig 7. Fig. 6 shows that SELDA is not able to capture the variations in the user's stride length. Especially, when going up and down stairs (smaller strides), its predictions are very close to 1.32m with extremely few variations. Despite the almost constant stride length estimation, the SELDA based PDR's walking distance error is within 10%, thanks to the fact that human walking is regular and constrained.

The velocity norm plot in Fig. 7 shows that RoNIN better tracks the user's dynamical changes as compared to SELDA. Special motions (start and stop, going up/down stairs) are predicted in the velocities, though RoNIN tends to underestimate. On the other hand, velocities predicted by RoNIN are much noisier than the ground truth, especially in test 2 and 3, which explains the important drift observed in the estimated trajectories. As explained in the theoretical analysis, ignoring gait events can result in noisy predictions.

Globally, the two methods show completely opposite behaviours: SELDA yields almost constant predictions corresponding to the user's "nominal" stride length, with very few variation. In contrast, RoNIN captures well the variations of the user's dynamic but may be too sensitive to noises.

VI. Conclusion

This survey proposes two main categories to classify existing AI methods for pedestrian navigation using wearable inertial sensors. (1) Human gait driven AI methods use gait event segmented signal sequence to infer gait vectors and (2) sampling frequency driven AI methods use fixed length signal sequence to infer the user's velocity or change in position. Gait driven methods are based on physical modeling of human walking and yield reasonable predictions thanks to the fact that human locomotion is regular and constrained. However, it is a simplification of the reality and can fail to capture the variation of the user's dynamics. Sampling frequency driven methods

don't need the complex gait analysis and are able to capture the changes in the user's dynamics, but ignoring the gait events can result in noisy inferences. Experiments comparing two methods, one in each category, confirm the theoretical analyses and show their complementary behaviours. The methods selected for category (1) and (2) achieve 7m and 12m average positioning errors respectively, on 3 indoor/outdoor testing tracks, totaling 770m and including stairs.

Both categories are facing challenges. Gait driven AI methods need to improve their robustness to deal with different device poses and user motions. Sampling frequency driven AI methods need to reduce noises in their predictions. Future research could fuse the 2 approaches.

REFERENCES

[1] N. Al Abiad, Y. Kone, V. Renaudin, and T. Robert, "Smartphone inertial sensors based step detection driven by human gait learning," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2021, p. 8. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9662513

[2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[3] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and C. Ou, "Pedestrian walking distance estimation based on smartphone mode recognition," *Remote Sensing*, vol. 11, no. 9, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/9/1140

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[5] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[6] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[7] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *IN PROCEEDINGS OF THE THIRTEENTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING*. Morgan Kaufmann, 1996, pp. 148–156.

[8] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

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

[10] H. Weinberg, "Using the adxl 202 in pedometer and personal navigation applications," 2002.

[11] M. Zhang, Y. Wen, J. Chen, X. Yang, R. Gao, and H. Zhao, "Pedestrian dead-reckoning indoor localization based on os-elm," *IEEE Access*, vol. 6, pp. 6116–6129, 2018.

[12] N.-Y. Liang, G. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, pp. 1411–1423, 2006.

[13] 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, 2019.

[14] D. Yan, C. Shi, and T. Li, "An improved pdr system with accurate heading and step length estimation using handheld smartphone," *Journal of Navigation*, vol. 75, no. 1, p. 141–159, 2022.

[15] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[16] 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, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/4/840

[17] Q. Wang, H. Luo, L. Ye, A. Men, F. Zhao, Y. Huang, and C. Ou, "Pedestrian heading estimation based on spatial transformer networks and hierarchical lstm," *IEEE Access*, vol. 7, pp. 162 309–162 322, 2019.

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

[19] R. Pelossof, I. Singh, J. Yang, M. Weirauch, T. Hughes, and C. Leslie, "Affinity regression predicts the recognition code of nucleic acid-binding proteins," *Nature biotechnology*, vol. 33, 11 2015.

[20] J. Rose, J. G. Gamble, and J. M. Adams, "Human walking." Philadelphia: Lippincott Williams and Wilkins, 2006, p. 2.

[21] J. Perry and J. M. Burnfield, "Gait analysis." SLACK Incorporated, 2012, ch. 7.

[22] J. Kim, H. 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, pp. 273–279, 12 2004.

[23] Q. Ladetto, "On foot navigation: Continuous step calibration using both complementary recursive prediction and adaptive kalman filtering," *Proceedings of ION GPS*, 01 2000.

[24] H. Yan, Q. Shan, and Y. Furukawa, "Ridi: Robust imu double integration," 12 2017.

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

[26] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," 01 2018.

[27] T. Feigl, S. Kram, P. Woller, R. H. Siddiqui, M. Philippsen, and C. Mutschler, "A bidirectional lstm for estimating dynamic human velocities from a single imu," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1–8.

[28] C. Combettes and V. Renaudin, "Comparison of misalignment estimation techniques between handheld device and walking directions," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–8.

[29] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am i facing: Inferring horizontal device orientation from an accelerometer signal," in *2009 International SymposiuEnhancing the Performance of Pedometers Using a Single Accelerometerm on Wearable Computers*, 2009, pp. 149–150.

[30] Z.-A. Deng, G. Wang, Y. Hu, and D. Wu, "Heading estimation for indoor pedestrian navigation using a smartphone in the pocket," *Sensors*, vol. 15, no. 9, pp. 21 518–21 536, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/9/21518

[31] M. Chowdhary, M. Sharma, Noida, A. Kumar, S. Dayal, and M. Jain, "Method and apparatus for determining walking direction for a pedestrian dead reckoning process," 2014.

[32] M. Kourogi and T. Kurata, "A method of pedestrian dead reckoning for smartphones using frequency domain analysis on patterns of acceleration and angular velocity," in *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, 2014, pp. 164–168.

[33] Archeries, "Selda dataset," Available: https://github.com/Archeries/StrideLengthEstimation, Accessed Apr. 25, 2022 [Online], Jan. 11 2019.

[34] J. Scarlett, "Enhancing the performance of pedometers using a single accelerometer," Available: https://www.analog.com/en/analog-dialogue/articles/enhancing-pedometers-using-single-accelerometer.html. Accessed Apr. 25, 2022 [Online], Mar. 2007.

[35] S. Herath, H. Yan, and Y. Furukawa, "Ronin implementation and dataset," Available: https://github.com/Sachini/ronin, Accessed Apr. 25, 2022 [Online], Jan. 13 2022.

[36] "Android game rotation vector documentation," https://developer.android.com/reference/android/hardware/Sensor#TYPE_GAME_ROTATION_VECTOR. Accessed Apr. 25, 2022 [Online].

[37] V. Renaudin and C. Combettes, "Magnetic, acceleration fields and gyroscope quaternion (magyq)-based attitude estimation with smartphone sensors for indoor pedestrian navigation," *Sensors*, vol. 14, no. 12, pp. 22 864–22 890, 2014. [Online]. Available: https://www.mdpi.com/1424-8220/14/12/22864

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[39] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271

[40] M. Ortiz, M. De Sousa, and V. Renaudin, "A new pdr navigation device for challenging urban environments," *Journal of Sensors*, vol. 2017, pp. 1–11, 2017.