Unsupervised pedestrian trajectory reconstruction from IMU sensors

Stéphane Derrode, Haoyu Li, Lamia Benyoussef, Wojciech Pieczynski

Unsupervised Pedestrian Trajectory Reconstruction from IMU Sensors

Haoyu Li[1], Stéphane Derrode[1], Lamia Benyoussef[2], et Wojciech Pieczynski[3]

[1] École Centrale de Lyon, Univ. de Lyon, LIRIS, CNRS UMR 5105, haoyuli1990@gmail.com
[2] EPITA Lyon, lamia.derrode@epita.fr
[3] Telecom SudParis, Univ. Paris-Saclay, SAMOVAR, CNRS UMR 5157, wojciech.pieczynski@telecom-sudparis.eu

Abstract This paper presents a pedestrian navigation algorithm based on a foot-mounted 9DOF Inertial Measurement Unit, which provides accelerations, angular rates and magnetics along 3-axis during the motion. Most of algorithms used worldwide are based on stance detection to reduce the tremendous integration errors, from acceleration to displacement. As the crucial part is to detect stance phase precisely, we introduced a cyclic left-to-right style Hidden Markov Model that is able to appropriately model the periodic nature of signals. Stance detection is then made unsupervised by using a suited learning algorithm. Then, assisted by a simplified error-state Kalman filter, trajectory can be reconstructed. Experimental results show that the proposed algorithm can provide more accurate location, compared to competitive algorithms, w.r.t. ground-truth obtained from OpenStreet Map.

Key words Pedestrian Navigation, Stance Detection, Inertial Sensor, HMMs.

1 Introduction

In recent years, Pedestrian Navigation System (PNS) has gained a lot of attention and been investigated extensively with various kinds of sensors like Inertial Measurement Units (IMUs), camera-based systems and WIFI-based ones. Among these sensors, IMUs have great advantages as they are small and can be wore on the body. They also do not need to pre-install devices like cameras or WIFI systems, and can be used both indoor and outdoor. With the kinematics information acquired from IMUs, it is theoretically possible to transfer the signals from sensor frame to earth frame, or called global frame, based on the sensor orientation, and then to compute velocity and displacement of the motion. Therefore, the exactness of IMUs-based PNS algorithms highly depends on the accuracy of orientation estimation and displacement computation.

When a person is walking, his foot swings in the air and does not move when attached to the ground, alternately. Consequently, one step can be broken down into four phases [7]: stance, push-up, swing and step-down phases. Stance phase is also called zero-velocity state, as the foot is not moving. The most common way to detect stance phase is to set thresholds for both acceleration and angular rate. To avoid manually setting thresholds, unsupervised learning methods can be used. J. Taborri *et al.* proposed an HMM-based distributed classifier for rehabilitation application [2]. H. Pham *et al.* [8] introduced a LLE-HMM framework and use EMG signals for gait recognition. A foot-mounted gyroscope for stance detection is implemented in [7], however because of the weak initialization, parameter learning fails at some time.

In order to reconstruct the trajectory, an algorithm called Pedestrian Dead-Reckoning (PDR) has been proposed, which computes the displacement by estimating and integrating each step length and heading. PDR is very easy to be implemented and used in many applications, but its error particularly depends on the sensor employed. E. Foxlin [4] firstly proposed a PNS algorithm that applies Extended Kalman Filter (EKF) to estimate the error and uses ZUPT approach to reduce the large integration error from acceleration to velocity and then to displacement. He called it INS-EKF-ZUPT (IEZ). Then, different strategies were proposed according to different measurements during the stance phase. S. Rajagopal [9] supposes the angular rate during stance phase to be zero and proposes a Zero Angular Rate Update (ZARU) algorithm to compute trajectory. The orientation error, particularly the yaw error, can also be estimated and added into the EKF measurement with the help of digital compass [1,5].

In this paper, we propose an adaptive stance detection algorithm that uses unsupervised parameter learning algorithm, and emploies a simplified Error-state Kalman Filter for PNS trajectory reconstruction. The remaining of the paper is organized as follows. First, a left-to-right HMM is presented to detect stance with a specific initialization algorithm for unsupervised parameter learning. Then a simplified error-state Kalman filter is developed to compensate for integration errors. An experiment is conducted on true pedestrian data and the proposed algorithm is compared with other algorithms.

2 Stance Detection with Left-to-Right HMM

Precise stance detection plays a critical role in ZUPT algorithm, since if the detection result is wrong (*i.e.* the swing phase is regarded as a stance phase), then the velocity will be wrongly compensated to zero while the foot is still moving.

Let start assuming a hidden Markov chain model with observations $\boldsymbol{Y} = \{Y_1, \ldots, Y_N\}$, each $Y_n \in \mathbb{R}$, and with unknown states $\boldsymbol{X} = \{X_1, \ldots, X_N\}$, each $X_n = k \in \Omega = \{1, \ldots, 4\}$, where Ω represents the stance, push-up, swing and step-down phases respectively. Assuming a discrete time independent Markov process, $\boldsymbol{X}$ can be parameterized by an initial probabilities vector $\boldsymbol{\pi} = p(x_1)$ and a transition matrix $\boldsymbol{A} = p(x_2|x_1)$. In a cyclic Left-to-Right HMM (LR-HMM), this transition matrix has the following particular shape, only allowing to switch from one class to the next:

$$\boldsymbol{A} = \begin{bmatrix} 1-\mathit{\Delta}_2 & \mathit{\Delta}_2 & 0 & 0 \\ 0 & 1-\mathit{\Delta}_3 & \mathit{\Delta}_3 & 0 \\ 0 & 0 & 1-\mathit{\Delta}_4 & \mathit{\Delta}_4 \\ \mathit{\Delta}_1 & 0 & 0 & 1-\mathit{\Delta}_1 \end{bmatrix}, \tag{1}$$

with $\mathit{\Delta}_k = p(x_n = k | x_{n-1})$ the transition probability from state $k-1$ to state $x_n = k$. Here, the N observations represent gyroscope measurements $y_n = \left[{}^s\omega_{xn}\ {}^s\omega_{yn}\ {}^s\omega_{zn}\right]$, which represent the angular rate of the sensor projected in sensor frame. The distributions of observations conditional to classes are assumed to be Gaussian

$$p\left(y_n \mid x_n = k\right) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2}$$

where $\boldsymbol{\mu}_k$ (3×1 vector) and $\boldsymbol{\Sigma}_k$ (3×3 matrix) are the mean and co-variance of observations corresponding to state k. So that the LR-HMM model we deal with is parametrized by the following set of parameters $\boldsymbol{\Theta} = \{\pi_k, \mathit{\Delta}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k \in \Omega}$. All the parameters can be learned using the well-known Baum-Welch algorithm, which is based on the EM principle (Expectation-Maximization) for finding the maximum likelihood iteratively, starting from an initial guess $\Theta^{(0)}$ of parameters and stopping after a criterion or a maximum number of iterations is reached.

Commonly, the initialization is performed using Kmeans algorithm. However, in LR-HMM the state transition has a specific transition order and structure (see eq. (1)) that Kmeans is not able to provide since it does not takes into account past observations. As an illustration, Fig. 1 shows the gyro observations. The observation values close to zero indicate the stance phase. It can be seen that angular rate goes across zero at transition between two states, then Kmeans wrongly classifies the state as stance phase.

To find the true transition order form Kmeans classification results, we propose the algorithm sketched in the diagram 2. Firstly, we filter the angular rate by using a low-pass Butterworth filter, and select the most significant axis, *i.e.* the axis whose signal has the largest magnitude, like the blue dashed line in Fig. 1. Secondly, we search for the movement durations between every stance phase that lasts longer than a threshold, *e.g.* $0.3s$ for example, and then we find the states corresponding to all the peaks and valleys in every movement duration, the peaks and valleys indicate the states of non-stance phases. Thirdly, we sort these states by time and only keep the first one if one state repeats twice or more. Thus, a list of non-stance states order in every movement duration can be acquired. At last, we count the most firstly appeared state in the order list, which means the push-up phase, the state of swing phase and step-down phase can be derived in the same way. In our example, after the disposal of Kmeans results the re-ordered state transition is $1 \to 2 \to 3 \to 4$ (Orange line in Fig. 1). This guess of states is then used to initialize EM for parameter learning.

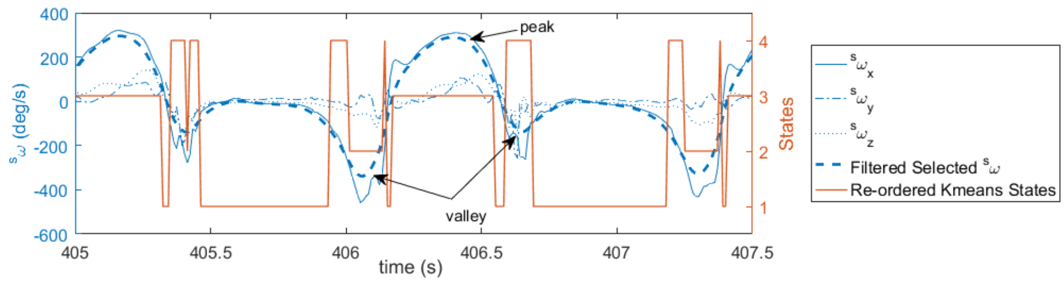


Figure 1: Re-ordered initial state sequence: 1→2→3→4.

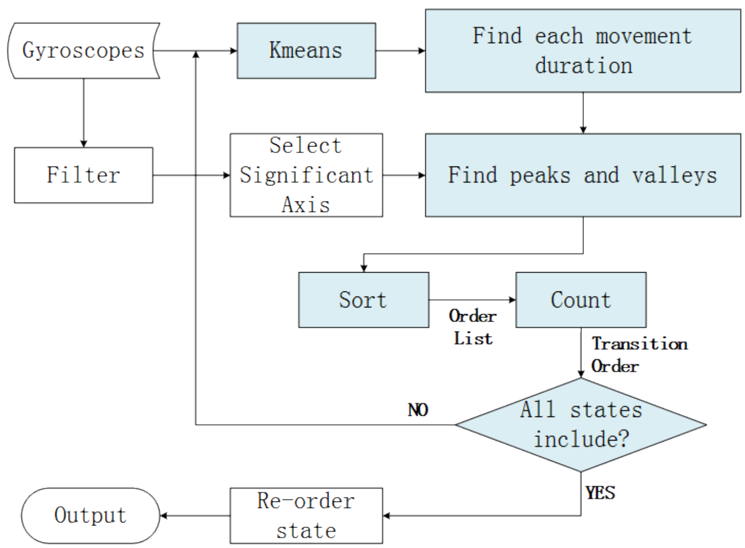


Figure 2: Diagram of state transition re-ordering from Kmeans classification results.

3 Error-state Kalman Filter

By knowing the stance phases, velocity is assumed to be zero and the velocity integration error can be acquired easily, but it is not able to obtain displacement integration error directly. Therefore the displacement integration error should be estimated appropriately, since it derives from velocity error and the correlation between velocity and displacement is determined by the integration function (5), thus it can be estimated by an appropriate way. Error-state Kalman filter is firstly introduced in [4] and gives a strategy to estimate the displacement integration error. In this work, We employ a simplified error-state Kalman filter (9 dimensional), since orientation estimation is performed independently by a gradient descent algorithm based quaternion method [6]. The error-state only takes into account the acceleration

($\boldsymbol{a}$), velocity ($\boldsymbol{v}$) and displacement ($\boldsymbol{r}$):

$$\delta\boldsymbol{\eta}_n = \delta\boldsymbol{\eta}_{n|n} = \left[\delta^e\boldsymbol{r}_n{}^\intercal, \delta^e\boldsymbol{v}_n{}^\intercal, \delta^s\boldsymbol{a}_n{}^\intercal\right]^\intercal \tag{3}$$

The error-state transition equation writes

$$\delta\boldsymbol{\eta}_{n|n-1} = \boldsymbol{\Phi}_n\delta\boldsymbol{\eta}_{n-1|n-1} + \boldsymbol{w}_{n-1}, \tag{4}$$

Where the superscripts e and s represent the earth frame and sensor frame respectively. $\boldsymbol{w}_n$ represents the process noise with covariance matrix $\boldsymbol{Q}_n = E\left(\boldsymbol{w}_n\boldsymbol{w}_n^\intercal\right)$ and where the error-state transition matrix $\boldsymbol{\Phi}_n$ is a 9×9 matrix given by

$$\boldsymbol{\Phi}_n = \begin{bmatrix} I_{3\times3} & \Delta t \cdot I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & \Delta t \cdot {}^e_sC_n \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}. \tag{5}$$

Matrix $\boldsymbol{\Phi}_n$ is time-variant and depends on the value of ${}^e_sC_n$, which represents the rotation matrix required to convert vectors from sensor frame to earth frame at time n. It is derived from the estimated quaternion[11].

Now, the measurement equation writes

$$\boldsymbol{z}_n = \boldsymbol{H}\delta\boldsymbol{\eta}_{n|n} + \boldsymbol{v}_n, \tag{6}$$

where $\boldsymbol{z}_n$ is the measurement from sensor, $\boldsymbol{H} = [0_{3\times3}, I_{3\times3}, 0_{3\times3}]$ is the measurement matrix, and $\boldsymbol{v}_n$ is the measurement noise, assumed Gaussian with covariance $\boldsymbol{R}_n$.

Because error-state measurement is only available when $\hat{x_n}$ is detected as stance phase, the error-state is only updated during this period. The error-state measurement in stance phase is $\boldsymbol{z}_n = {}^e\boldsymbol{v}_n - [0,0,0]^T$ (zero represents the real velocity), thus by the prediction and estimation in Kalman Filter [10], the velocity and displacement can be compensated by ${}^e\boldsymbol{v}_n - \delta^e\boldsymbol{v}_n$ and ${}^e\boldsymbol{r}_n - \delta^e\boldsymbol{r}_n$ respectively, $\delta^e\boldsymbol{r}_n$ and $\delta^e\boldsymbol{v}_n$ in error-state $\delta\boldsymbol{\eta}_n$ should be reset to zero after the compensation.

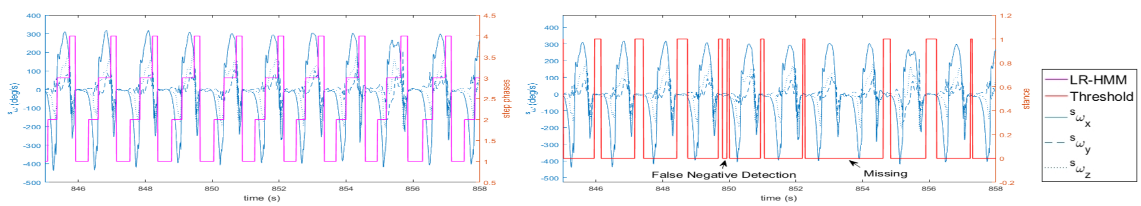
4 Experimental Results

An experiment was conducted on a road nearby the campus of École Centrale de Lyon, Ecully, France. The ground truth is obtained from Openstreet Map , the total travel distance is $1075m$, the data was stored in the IMU embedded SD card. The sampling rate was set to 100 Hz, the range of accelerometers, gyroscopes and magnetometers were set to 8**g**, $1000deg/s$ and $2.5Ga$ respectively [1].

1. more details can be found at manufacturer's site `http://www.shimmersensing.com/images/uploads/docs/ConsensysPRO_Spec_Sheet_v1.1.0.pdf`

Figure 3: Shimmer3 sensor and placement on the shoe.

Before starting to walk, a short standing time period without motion is necessary for initializing the quaternion corresponding to the earth frame (North-West-Up coordinate system), the magnetic declination at Lyon is 1.2°. The LR-HMM method we propose is tested and compared to another threshold based stance detection method detailed in [3]. Parameters of both algorithms are learned or tuned for getting the best result.



(a) LR-HMM. (b) Threshold.

Figure 4: Stance detection, step #1 represents the stance.

Table 1: Stance Detection Error.

Algorithm	Steps Number	Missing Number	False Negative (rate in %)
LR-HMM	735	0	6 (0.823)
Threshold	724	17	14 (1.920)

The total steps number in experiment is 1458, so the steps number of one foot is 729. Compared with the threshold based stance detection method, LR-HMM obtains a more regular stance pattern, rarely makes a false negative detection or misses one step (Fig. 4). Table. 1 reports the missing number and false negative detection number of both algorithms.

Trajectory reconstruction is done by different algorithms, including the proposed algorithm in this paper and a commonly used IEZ algorithm (15 dimensional error-state Kalman Filter). In Fig. 5, the proposed algorithm (HMM+GDA) makes a travel distance of 1077.7m, the relative travel distance error is 0.25%, the End-to-End error is 23.3m. From Tab. 2, in spite of the fact that the End-to-End error of HMM+IEZ is a little smaller than the proposed algorithm, the Dynamic Time Warping (DTW) distance of the proposed algorithm is smaller than HMM+IEZ. This interesting point means that the trajectory derived from the proposed algorithm is closer to the ground truth. And furthermore, when tuning parameters for all algorithms, HMM+GDA has the most robust performance for trajectory reconstruction.

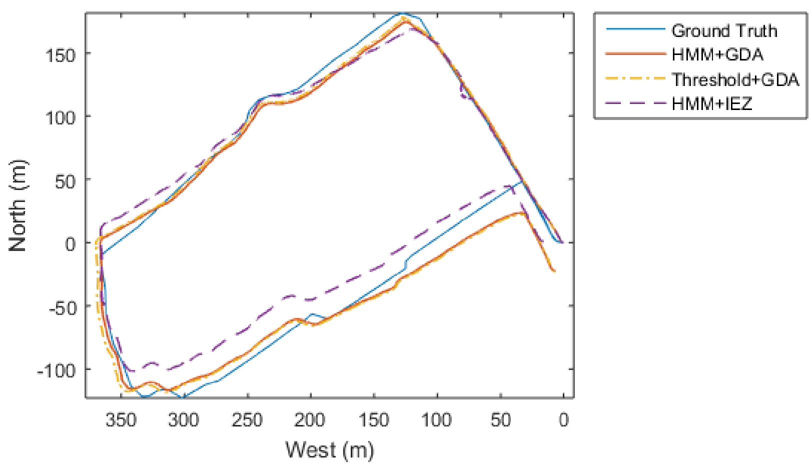


Figure 5: Trajectory of different algorithms compared with the ground truth.

Table 2: PNS Trajectory Error.

	HMM+GDA	Threshold+GDA	HMM+IEZ	Ground Truth
End-to-End Error(m)	23.31	24.76	16.11	0
End-to-End Positioning Accuracy(%)	2.16	2.30	1.49	0
Travel Distance(m)	1077.69	1131.19	1078.52	1075
Relative Error of Travel Distance(%)	0.25	5.22	0.32	0
DTW Distance	8784	8617	12809	—

5 Conclusion

We present an algorithm for trajectory reconstruction from a foot-mounted IMU sensor. The basic difficulty of such an algorithm mainly relies on minimizing the double integration required to calculate the displacement (earth frame) from the observed kinematic signals (sensor frame). We propose an algorithm that is mainly unsupervised and relies on a cyclic Left-Right HMM to mimic the periodicity of the step phases during a walk.

It seems that our algorithm performs better than the state-of-the-art methods, but we still need to experiment further the algorithm, to study large-scale motion, and also to study the performance with respect to the elevation when the terrain is not plane. We also plan to investigate if using two sensors together on both two feet can improve trajectory reconstruction results. In that case we have to study possible interference between the two sensors, and their influence on the reconstructed trajectory.

References

1. A. Norrdine et al. Step detection for ZUPT-Aided inertial pedestrian navigation system using foot-mounted permanent magnet. *IEEE Sensors Journal*, 16(17):6766–6773, 2016.
2. J. Taborri et al. A novel HMM distributed classifier for the detection of gait phases by means of a wearable inertial sensor network. *Sensors*, 14(9):16212–16234, 2014.
3. S. Qiu et al. Inertial/magnetic sensors based pedestrian dead-reckoning by means of multi-sensor fusion. *Information Fusion*, 39(Supplement C):108 – 119, 2018.
4. E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer graphics and applications*, 25(6):38–46, 2005.
5. 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 *Workshop on Positioning Navigation and Communication (WPNC)*, pages 135–143, 2010.
6. S. OH Madgwick, A. JL Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *IEEE Int. Conf. on Rehabilitation Robotics (ICORR)*, pages 1–7, 2011.
7. A. Mannini and A. M. Sabatini. A hidden Markov model-based technique for gait segmentation using a foot-mounted gyroscope. In *Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 4369–4373, Boston, MA, USA, Aug 2011.
8. H. Pham, M. Kawanishi, and T. Narikiyo. A LLE-HMM-based framework for recognizing human gait movement from EMG. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 2997–3002, Washington, USA, May 2015.
9. S. Rajagopal. Personal dead reckoning system with shoe mounted inertial sensors. Master's thesis, Royal Institute of Technology (KTH), Sweeden, 2008.
10. Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, chapter 2, pages 123–138. John Wiley & Sons, 2006.
11. Li Wang, Zheng Zhang, and Ping Sun. Quaternion-based Kalman filter for AHRS using an adaptive-step gradient descent algorithm. *International Journal of Advanced Robotic Systems*, 12(9):131, 2015.