Probabilistic Context-aware Step Length Estimation for Pedestrian Dead Reckoning

Alessio Martinelli, Han Gao, Paul D. Groves and Simone Morosi

***Abstract*—This paper introduces a weighted context-based step length estimation algorithm for pedestrian dead reckoning. Six pedestrian contexts are considered: stationary, walking, walking sideways, climbing and descending stairs, and running. Instead of computing the step length based on a single context, the step lengths computed for different contexts are weighted by the context probabilities. This provides more robust performance when the context is uncertain. The proposed step length estimation algorithm is part of a pedestrian dead reckoning system which includes the procedures of step detection and context classification. The step detection algorithm detects the step time boundaries using continuous wavelet transform analysis, while the context classification algorithm determines the pedestrian context probabilities using a relevance vector machine. In order to assess the performance of the pedestrian dead reckoning system, a dataset of pedestrian activities and actions has been collected. Fifteen subjects have been equipped with a waist-belt smartphone and traveled along a predefined path. Acceleration, angular rate and magnetic field data were recorded. The results show that the traveled distance is more accurate using step lengths weighted by the context probabilities compared to using step lengths based on the highest probability context.**

***Index Terms*—Step length estimation, context detection, step detection, Pedestrian dead reckoning.**

I. Introduction

The rapid growth of location-based services (LBS) over the past few years [1] has led positioning and navigation to assume a significant role in people's daily lives. LBS mainly involve sectors like military, emergency services and commercial [2], and imply the use of mobile computing devices.

Pedestrian navigation applications are crucial for providing LBS to mobile devices carried or worn by pedestrians. Pedestrian navigation is one of the most challenging applications of navigation technology. It must work in environments where coverage of global navigation satellite systems (GNSS) and most other radio signals used for navigation is poor. Inertial sensors can be used to measure pedestrian motion by dead reckoning, and micro electro-mechanical systems (MEMS) sensors are particularly suited for pedestrian navigation purposes: they are small, light and low cost [3]. Moreover, MEMS sensors are built into smartphones, which can be exploited for pedestrian navigation scenarios. However, MEMS sensors provide poor inertial navigation performance stand alone, while the combination of low dynamics and high vibration limits the calibration obtainable from GNSS or other positioning systems [3].

A. Martinelli and S. Morosi are with the Department of Information Engineering, University of Florence, Florence, Italy (e-mail: alessio.martinelli@unifi.it; simone.morosi@unifi.it).

Han Gao and Paul D. Groves are with the Department of Civil, Environmental and Geomatic Engineering, University College London, London WC1E6BT (e-mail: han.gao.14@ucl.ac.uk, p.groves@ucl.ac.uk).

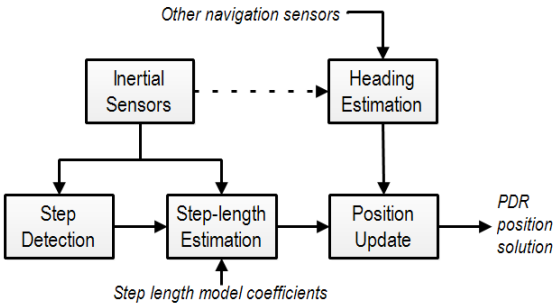

Fig. 1. Pedestrian dead reckoning processing [3].

Pedestrian dead reckoning (PDR) is a promising solution for pedestrian navigation using inertial sensors [4], [5], [6]. A PDR system comprises three phases: step detection, step length estimation (SLE) and position-solution update, as shown in Fig.1. Step detection routine aims to identify when steps occur, while the SLE algorithm determines the length of each detected step using sensor data. SLE works relatively well when walking in a straight line. However, real paths include turns, sidesteps, stairs, variations in speed, and various actions performed by the subject. These all affect the step length and must be considered to provide an accurate estimate of the distance traveled during daily activities as well as laboratory tests.

This work proposes a weighted context-based step length estimation (WC-SLE) algorithm in which the step lengths computed for different pedestrian contexts are weighted by the context probabilities. Six pedestrian contexts are considered: stationary, walking, walking sideways, climbing and descending stairs, and running, as shown in Table I. The proposed SLE algorithm is compared with the highest context-based step length estimation (HC-SLE) method, in which the step length is determined based on the highest probability context. HC-SLE is the union between two state-of-the-art methods:

- highest probability context determination [7];
- regression-based step length estimation[6], [8].

For each considered pedestrian context, the single step length is determined based on the assumption that the step length can be expressed as a linear combination of a constant, the step frequency and the specific force variance.

The performance of the presented WC-SLE method is comparable to those techniques proposed for step length estimation with body fixed and handheld sensors in a walking context only. [9][10][11]. The SLE procedures are part of a PDR system which includes the step detection and the context

classification algorithms, as illustrated in Fig.2. The step detection algorithm aims to detect the step time boundaries and is based on continuous wavelet transform (CWT) analysis. The context classification routine exploits a relevance vector machine (RVM) [12] to determine the pedestrian context probabilities from multiple epochs of data over a sliding time window which may include several contexts [7].
In order to assess the PDR system performance, a dataset of pedestrian activities and actions has been collected. Fifteen subjects, which have been equipped with a waist-belt smartphone, traveled along a predefined path, as shown in Fig.3. Acceleration, angular rate and magnetic field data were recorded at a 100 Hz sample rate by the relevant 3-axis MEMS sensors, which are integrated into the smartphone.
Section II reports the background and related work, while Section III describes the proposed approach and the dataset collection in detail. Sections IV and V respectively characterize the step detection and the context classification techniques. Section VI defines the SLE algorithms and Section VII reports the results obtained for the context classification technique and compares the total distances computed by each SLE algorithms. Finally, Section VIII presents the conclusions.

II. Backgound and Related Work

In the PDR systems, the step detection procedure is commonly based on specific force data processing [13],[14], while angular rate information is less frequently used [15]; moreover, since the pedestrian motion is usually not aligned with the sensor-body coordinate frame, the signal magnitude provided by the sensor is considered. In particular, when focusing on step detection through the specific force signal analysis, the steps are revealed through detection techniques which depend on the on-body sensor location. In the case of foot-mounted sensors, step detection can easily be performed by identifying the stance and swing phases of the foot corresponding to zero velocity periods (ZVPs) [16],[17]. The latter approach cannot be applied when dealing with handheld, wrist- or waist-mounted sensors [16], [11], [13], since these configurations do not lead to a zero velocity period occurrence; in these cases, zero-crossing (ZDT) or peak detection (PDT) can be adopted for the step detection routine. However, both PDT and ZDT may lead to revealing multiple steps when actually only one step occurs [16], leading to detection errors which can affect the resulting PDR navigation solution.
Step mode classification aims to recognize the type of motion activity the subject is performing, e.g. walking or running. The classification routine extracts features [18],[19] from the signals provided by body-worn sensors, e.g. accelerometers, gyroscopes, magnetometers and the barometer. Step mode classification is also dependent on the sensor location: the same movement may generate different signals for different on-body sensor locations. This is the reason that step mode and sensor location classification are frequently tied together [16][15]. When dealing with PDR on smartphones, the sensor location may also be related to the device pose which relates to particular actions the subject can perform: texting, calling, swinging or simply keeping the device in a pocket. The classified step modes can be exploited to get landmarks on building maps in order to calibrate the PDR routines particularly in locations such as stairs or elevators [20].
In order to reveal a particular activity in a certain time interval, the algorithms based on pedestrian activity classification use sensor data from multiple epoch over a sliding time window which is typically a few seconds long [21]. The selection of an appropriate window length is critical. A long time window may incur delayed response of the PDR system and mix multiple activities in a single window, but a too short window may not capture sufficient characteristic of the activity for classification.
The supervised learning field provides a number of classification algorithms adopted for step mode and sensor location classification: decision tree (DT), support vector machine (SVM), k-nearest Neighbour (KNN) [19], [22]; each of which presents different characteristics in terms of memory usage, fitting speed, prediction speed, and predictive accuracy [23]. Furthermore, the relevance vector machine (RVM) is used for pedestrian context classification, in order to provide the context probabilities for each considered context class [7]. These are two examples of context-adaptive and context-aided navigation [24], [25], respectively.
Step length estimation has a significant impact on the performances of PDR systems and can be implemented through several approaches: regression-based [26],[27], biomechanical models [10], [28], or empirical relationships [29], [10]. Regression-based methods aims to model the step length as the combination of a constant, the step frequency, and the specific force variance during the step interval. A Biomechanical model estimator relates the step length to the specific force measurement at the center of mass (COM) of the user. The vertical displacement of the COM during one step is calculated via double integration of the vertical acceleration while the COM is ascending. Empirical relationships determine the step length estimates through mathematical expressions, which generally relate the step length to particular specific force parameters, which commonly correspond to the maximum and minimum acceleration values measured within the step interval.

III. Proposed method and Dataset

The present paper proposes a weighted context-based step length estimation (WC-SLE) algorithm, in which the step lengths computed for different pedestrian contexts are weighted by the context probabilities. A context corresponds to a particular pedestrian motion activity; in particular, six pedestrian contexts are considered: stationary, walking, walking sideways, climbing and descending stairs and running, as illustrated in Table I.
The WC-SLE algorithm works over a 2-seconds long and 50% overlapped sliding time window, and consists of two parts: the first one aims to determine the distance traveled under each context by exploiting the relation in which the step length can be expressed as a linear combination of a constant, the step

TABLE I
PEDESTRIAN CONTEXTS

Context	C_1	C_2	C_3	C_4	C_5	C_6
Type	Stationary	Walking	Walking upstairs	Walking downstairs	Walking sideways	Running

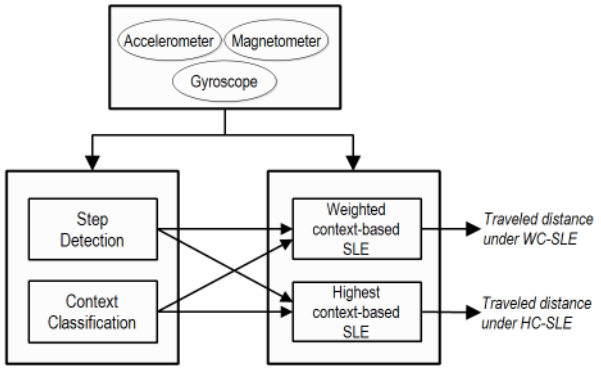

Fig. 2. PDR framework.

frequency, and the specific force variance [26],[27]; the second one performs the weighting between the distances computed in the first part and the context probabilities determined over the considered time window.

The proposed WC-SLE algorithm is compared with the highest context-based SLE (HC-SLE) algorithm, which only considers the traveled distance corresponding to the context with the highest probability.

The SLE routines are part of a PDR framework, which also defines the procedures of step detection and context classification, as shown in Fig.2 The step detection algorithm aims to identify the step time boundaries by performing the continuous wavelet transform (CWT) analysis of the specific force signal [30], [31]. The context classification algorithm uses the RVM method to classify a fixed portion of data, which generally may include several steps, into several contexts, each with an associated probabilities.

In order to assess the performance of the proposed PDR system in a real scenario, a dataset of pedestrian activities has been collected. Fifteen subjects traveled along a predefined path, shown in Fig.3. Subjects with different age and gender have been considered: 7 women and 8 men, both between 20 and 50 years old. The subjects have been equipped with a waist-belt smartphone Samsung Galaxy S4, as shown in Fig.4. Acceleration, angular rate and magnetic field data were recorded at a 100 Hz sample rate by the relevant 3-axis MEMS sensors, which are integrated into the smartphone. Six pedestrian activities, which correspond to the contexts shown in Table I, have been considered. Moreover, the subjects performed specific actions during the test, in order to yield a realistic daily life scenario. As shown in Fig.3, the actions are numbered from 1 to 5 and comprise:

1) Watching a notice board (Fig.5.1).
2) Entering an office and sitting down for a while (Fig.5.2).
3) Opening a door (Fig.5.3).
4) Lacing up shoes (Fig.5.4).
5) Taking a bottle of water from an automatic vending machine (Fig.5.5).

During the data recordings, the path has been identified through a coloured scotch tape arranged on the ground: different tape colours have been used for highlighting different pedestrian contexts. The subjects were asked to travel the path and perform the movements and actions as natural as possible without any conditioning. The true path length is 331 meters: it has been determined through a tape measure by measuring the length of the tape attached on the ground. The dataset has been collected at the School of Engineering of the University of Florence, as shown in Fig 3.

IV. STEP DETECTION

In PDR navigation, the accelerometer signal pattern is generally exploited to determine the presence of steps over time [6], [16]. The specific force signal provided by body-worn sensors during pedestrian activities like walking, shows a periodic pattern which depends on the principal frequency of the movement, as illustrated in Fig.6. The purpose of the step detection routine is to reveal the step time indexes which correspond to the step boundaries, in order to separate consecutive steps [13],[14]. Since the step detection is the primary stage in a PDR process, either false or missed step detections can strongly affect the estimation of the traveled distance.

In this work, the step detection algorithm exploits the continuous wavelet transform (CWT) analysis [30] and follows the block diagram illustrated in Fig.7. The specific force is used from multiple epoch over a 50% overlapped sliding time window of 1-second duration. In order to perform a detection independently of the sensor orientation, the magnitude of the specific force is computed:

$$||f|| = \sqrt{f_x^2 + f_y^2 + f_z^2} \tag{1}$$

where f_x, f_y, and f_z are the components of the specific force expressed in the body coordinate frame. Then, a band-pass filtering operation is performed, in order to filter the signal noise at the higher frequencies and to obtain a zero-mean specific force magnitude to provide as input to the CWT block. The latter aims to identify the local maxima which correspond to the step boundaries, while discarding the local maxima and minima due to the signal irregularities, as illustrated in Fig.8.

The CWT processing brings out specific time events in the signal by decomposing the signal over dilated and translated wavelets [32]. A wavelet is a short waveform generally named $\psi \in L^2(\Re)$: $|\psi| = 1$ and centered in the neighbour of $t = 0$ such that:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0. \tag{2}$$

Suppose that the ψ is a real wavelet, the corresponding Real Wavelet Transform of the function f is given as:

$$W\big[f(u,s)\big] = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{s}}\psi^*\left(\frac{t-u}{s}\right)dt. \tag{3}$$

This operator measures the variation of the function f in a neighbour of u proportional to s. In our case, the variable

Fig. 3. Path and pedestrian activity description. The subject was asked to travel the highlighted path by performing five pedestrian activities and several specific actions. The path length is 331 meters.

u represents the time translation variable which enables the wavelet to scan the entire time window. On the other hand, the variable s represents the scale which permits to dilate the wavelet in order to bring out particular events coming up in a certain time within the analysed signal waveform. The expression (3) represents the inner product between the considered wavelet and the given function f which in our case represents the band-pass filtered version of the specific force magnitude. The higher the correlation between the function and the appropriately dilated wavelet, the higher the resulting inner product value. High values of the correlation between the filtered version of the specific force magnitude and the fine scales of the wavelet are expected at the local maxima which refer to the step boundaries. By knowing that the Local Lipschiz regularity [32] of a function f at a particular time v depends on the decay at fine scale of $|W[f(u,s)]|$ in the neighbour of v, the local maxima in the filtered specific force magnitude can be detected from the local maxima values of $|Wf[(u,s)]|$ [30][32]. In this work, a wavelet scale value is equal to 16 and a time window overlapping factor is equal to 50% have been chosen in order to provide the best trade-off between the step detection accuracy and the computational load.

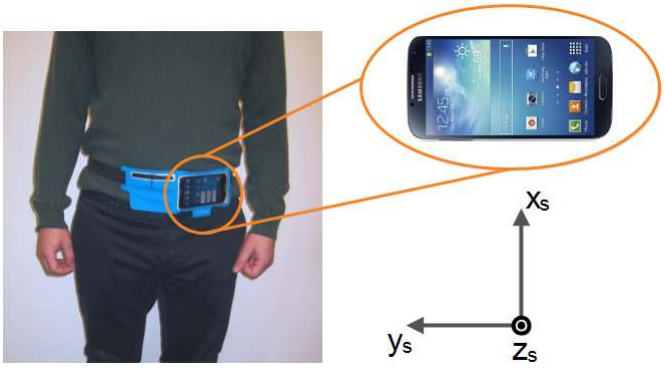

Fig. 4. On-body smartphone location and sensor body frame.

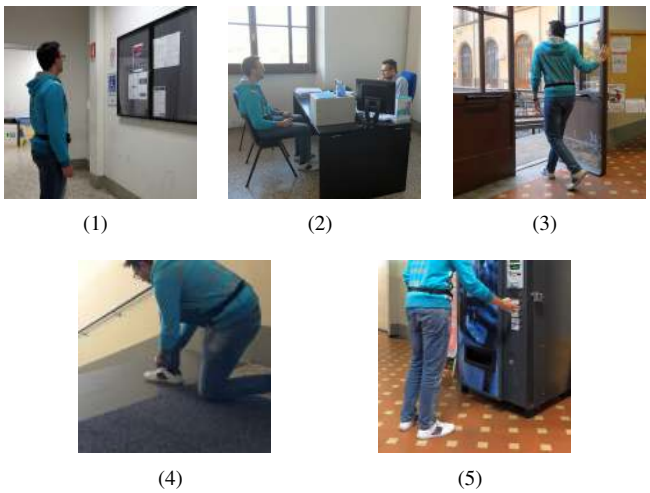

Fig. 5. Illustration of the considered pedestrian actions: watching a notice board (1), entering an office and sitting down for a while (2), opening a door (3), lacing up shoes (4), taking a bottle of water from an automatic vending machine (5).

V. Context Classification

The purpose of the context classification subsystem is to predict the pedestrian contexts based on the features that are

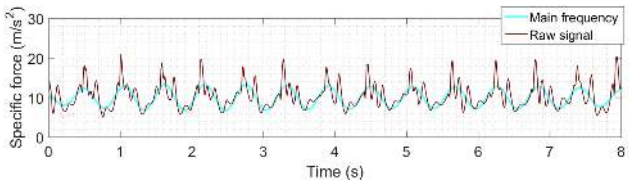

Fig. 6. Pattern of the specific force magnitude provided by the waist-worn accelerometer during a walking activity.

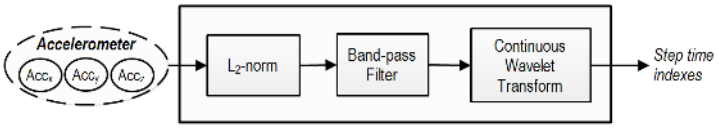

Fig. 7. Step detection processing.

extracted from the sensor measurements. To this aim, the RVM algorithm is exploited and six context categories are considered: stationary, walking, walking sideways, climbing and descending stairs and running.
The context classification technique is illustrated in Fig. 9.

A. Feature Extraction

Feature extraction is a crucial operation in classification problems. A good set of feature can often provide accurate and comprehensive descriptions of patterns from which the differences between context categories are easily discerned. In this work, both time-domain and frequency-domain features are considered, in order to capture either temporal variations or periodic characteristics of motion [7]. Range, standard deviation, skewness, kurtosis, energy, and zero-crossing rate are extracted over time-domain, while the maximum peak and the relative frequency index are extracted over frequency-domain, as summarized in Table II[1]. The aforementioned features are extracted from the magnitude signal of all available sensors: accelerometers, gyroscopes and magnetometers. The effectiveness of features similar to these for pedestrian activity classification have been shown in different studies [18],[19],[33],[34].

B. Feature Selection

In order to assess the usefulness and identify the most relevant features for distinguishing different activities, feature selection techniques are used and described. To explore the best combination of features, the Sequential Forward Floating Selection (SFFS) algorithm [35] is investigated in this paper. The number of misclassified observations has been used as the criterion to determine which feature has to be added next in the current feature set. The SFFS algorithm aims to minimize the misclassified observations over all feasible feature subsets, in order to obtain better classification performance. The SFFS

[1]In Table II, "a" refers to the specific force signal, "g" refers to the angular rate signal, "m" refers to the magnetometer signal, LPS refers to the low-pass filtering operation, DFT refers to the discrete Fourier function operation, ZCR is the zero-crossing rate, N is the length of the sample window, and $I(.)$ refers to the indicator function which is 1 if its argument is true and 0 otherwise.

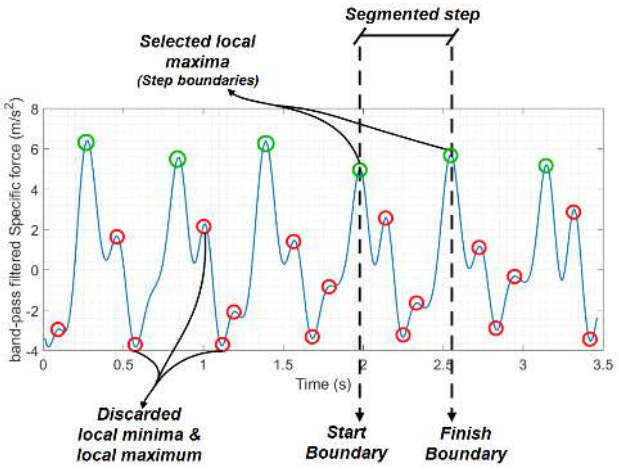

Fig. 8. Step detection performed through specific force signal analysis.

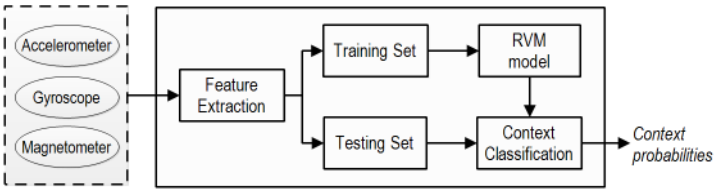

Fig. 9. Context classification processing.

TABLE II
FEATURE DESCRIPTION

F1	$range_{acc} = max(\|a\{t\}\|) - min(\|a\{t\}\|)$
F2	$range_{acc_f} = max(LPF(\|m\{t\}\|)) - min(LPF(\|m\{t\}\|))$
F3	$range_{gyro} = max(\|g\{t\}\|) - min(\|g\{t\}\|)$
F4	$range_{magn} = max(\|m\{t\}\|) - min(\|m\{t\}\|)$
F5	$\sigma_{acc} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|a_t\| - \overline{\|a\|})^2}$
F6	$\sigma_{acc_f} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(LPF(\|a_t\|) - \overline{\|a\|})^2}$
F7	$\sigma_{gyro} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|g_t\| - \overline{\|g\|})^2}$
F8	$\sigma_{magn} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|m_t\| - \overline{\|m\|})^2}$
F9	$skewness_{acc} = \frac{1}{N\sigma^3}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|a_t\| - \overline{\|a\|})^3}$
F10	$skewness_{acc_f} = \frac{1}{N\sigma^3}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(LPF(\|a_t\|) - \overline{\|a\|})^3}$
F11	$skewness_{gyro} = \frac{1}{N\sigma^3}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|g_t\| - \overline{\|g\|})^3}$
F12	$skewness_{magn} = \frac{1}{N\sigma^3}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|m_t\| - \overline{\|m\|})^3}$
F13	$kurtosis_{acc} = \frac{1}{N\sigma^4}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|a_t\| - \overline{\|a\|})^4}$
F14	$kurtosis_{acc_f} = \frac{1}{N\sigma^4}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(LPF(\|a_t\|) - \overline{\|a\|})^4}$
F15	$kurtosis_{gyro} = \frac{1}{N\sigma^4}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|g_t\| - \overline{\|g\|})^4}$
F16	$kurtosis_{magn} = \frac{1}{N\sigma^4}\sqrt{\frac{1}{N}\sum_{t=1}^{N}(\|m_t\| - \overline{\|m\|})^4}$
F17	$Energy_{acc} = \sum_{t=1}^{N}\|a_t\|^2$
F18	$Energy_{acc_f} = \sum_{t=1}^{N} LPF(\|a_t\|^2)$
F19	$Energy_{gyro} = \sum_{t=1}^{N}\|g_t\|^2$
F20	$Energy_{magn} = \sum_{t=1}^{N}\|m_t\|^2$
F21	$max\|DFT(\|a_t\|)\|$
F22	$max\|DFT(LPF\|a_t\|)\|$
F23	$max\|DFT(\|g_t\|)\|$
F24	$frequency\ index\ of\ F21$
F25	$frequency\ index\ of\ F22$
F26	$frequency\ index\ of\ F23$
F27	$ZCR_{acc} = \frac{1}{N-1}\sum_{n=1}^{N-1} I\{\|a_t\|\ \|a_{t+1}\| < 0\}$
F28	$ZCR_{acc_f} = \frac{1}{N-1}\sum_{n=1}^{N-1} I\{LPF(\|a_t\|)\ LPF(\|a_{t+1}\|) < 0\}$

consists of two parts:

- a new feature is added to the current feature subset if better classification performance is achieved;
- a conditional exclusion is then applied to the new feature subset, from which the least significant feature is determined. If the least significant feature is the last one added, the algorithm goes back to select a new feature. Otherwise the least significant feature is excluded and moved back to the available feature subsets and conditional exclusion is continued.

This cycle is repeated until there is no further improvement of classification performance. The advantage of this method is that it takes feature redundancy into consideration and the discarded features can be selected again in the inclusion and exclusion procedure. Fig.10 shows the average classification accuracy as a function of the number of features selected by SFFS, ranging from 1 to 28 (full feature set). The shadow area in the figure indicates the standard deviation using 10-fold cross validation. The results show that the classification performance best with 18 features included, achieving a 91.1% classification accuracy on average. If we pick some more features beyond this number, the performance degrades gradually. The corresponding 18 features are F1, F2, F4, F5, F6, F9, F10, F14, F16, F17, F19, F20, F21, F22, F23, F24, F25 and F27 in Table II.

C. Relevance Vector Machine Model

RVM is a Bayesian sparse kernel technique [36] for regression or classification, sharing many of the main characteristics of SVM. The advantage of RVM beyond boolean classifiers, like SVM, is that it can provide probabilistic classification results for each category. Thus the subsequent mechanism can adapt different strategies based on the uncertainty of the classification decisions. Note that in the algorithm described in this section, we assume that there are $L = 6$ possible pedestrian contexts $C = \{C_k | k = 1, 2, \ldots, L\}$, as illustrated in Table I. Given a training dataset $X = \{X_{i,j} | i = 1, 2, \ldots, N; j = 1, 2, \ldots, M\}$, each sample $X_i = \{X_{i,1}, X_{i,2}, \ldots, X_{i,M}\}$ is assigned to a target value $y_i \in \boldsymbol{C}$. M is the number of features and N is the number of the samples in the dataset. Fundamentally, RVM is a binary classifier ($y \in \{0, 1\}$) under a Bayesian probabilistic framework [36]. The relationship of the input vector and their real-valued predictions $t(\boldsymbol{X}_i)$ are modelled by a linearly weighted function

$$t(\boldsymbol{X_i}; \boldsymbol{w}) = \sum_{i=1}^{N} w_i \phi(\boldsymbol{X_i}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{X_i}), \tag{4}$$

where $\boldsymbol{w}$ denotes the weights of samples and $\boldsymbol{\phi}(\boldsymbol{X}_i)$ is a nonlinear basis function. The input data samples $\boldsymbol{X}_i$ are classified according to the sign of $t(\boldsymbol{X}_i)$. To infer the function $t(\boldsymbol{X}_i)$, we need to define the basis function and to estimate the weights as well. In here, the radial basis kernel function is used, so that:

$$\boldsymbol{\Phi_{ij}} = \boldsymbol{\phi}^T(\boldsymbol{X_i})\boldsymbol{\phi}(\boldsymbol{X_j}) = exp\left(- \frac{\|\boldsymbol{X_i} - \boldsymbol{X_j}\|^2}{2\sigma_G^2} \right). \tag{5}$$

A Bayesian probabilistic framework infers a distribution over the weights. According to Bayes rule, the posterior probability of $\boldsymbol{w}$ is

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{\alpha}) = \frac{p(\boldsymbol{y}|\boldsymbol{w}, \boldsymbol{\alpha}) p(\boldsymbol{w}|\boldsymbol{\alpha})}{p(\boldsymbol{y}|\boldsymbol{\alpha})}, \tag{6}$$

where $\boldsymbol{y} = (y_1, ..., y_N)^T$, α_i represents the precision of the corresponding parameter w_i, and $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_N)^T$. $p(\boldsymbol{y}|\boldsymbol{w}, \boldsymbol{\alpha})$ is the likelihood of the target values given the training dataset. The conditional prior probability distribution $p(\boldsymbol{w}|\boldsymbol{\alpha})$ in Equation (6) is modelled by a Gaussian function where the parameters w_i are weighted by parameters α_i.

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{M} \frac{\sqrt{\alpha}}{\sqrt{2\pi}} exp\left(- \frac{\alpha_i w_i^2}{2} \right). \tag{7}$$

Because $y \in \{0, 1\}$ is a binary variable, the likelihood function can be described by a Bernoulli distribution:

$$p(\boldsymbol{y}|\boldsymbol{w}, \boldsymbol{\alpha}) = \prod_{i=1}^{M} [\sigma_B(t(\boldsymbol{X_i}; \boldsymbol{w}))]^{y_i} [1 - \sigma_B(t(\boldsymbol{X_i}; \boldsymbol{w}))]^{1-y_i}], \tag{8}$$

where $\sigma(y) = 1/(1 + e^{-y})$ is the logistic sigmoid link function. Equation (6) with the probability densities given by (7) and (6) cannot be solved analytically. Therefore, a numerical method, the Laplacian approximation, is proposed to find the maximum a posterior (MAP) weights $\boldsymbol{w}^*$ based on the training dataset,

$$\begin{aligned} &ln\{p(\boldsymbol{y}|\boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{\alpha})\} \\ &= \sum_{i=1}^{N} [y_i \; ln(t_i) + (1 - y_i) ln(1 - t_i)] - \frac{1}{2} \boldsymbol{w}^T \boldsymbol{A} \boldsymbol{w}, \end{aligned} \tag{9}$$

where $\boldsymbol{A} = diag(\boldsymbol{\alpha})$. By computing the maximum value of (9) with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{y}$, the mean $\boldsymbol{w}^*$ and covariance $\boldsymbol{\Delta}$ of the Laplacian approximation are obtained:

$$\begin{aligned} \boldsymbol{w}^* &= \boldsymbol{B} \boldsymbol{\Delta} \boldsymbol{\Phi}^T \boldsymbol{y} \\ \boldsymbol{\Delta} &= (\boldsymbol{\Phi}^T \boldsymbol{B} \boldsymbol{\Phi} + \boldsymbol{A})^{-1}, \end{aligned} \tag{10}$$

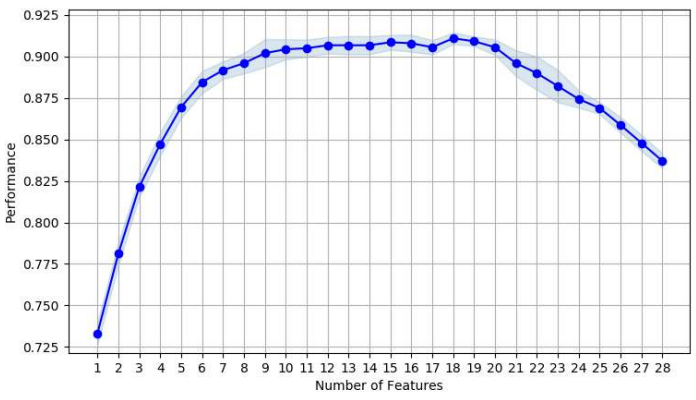

Fig. 10. Feature selection performance of the SFFS algorithm.

where $\boldsymbol{B} = diag(\beta_1, \beta_2, \ldots, \beta_N)$ is a diagonal matrix with $\beta_i = \sigma(y_i)[1 - \sigma(y_i)]$. After obtaining $\boldsymbol{w}^*$, the parameters are iteratively updated using

$$\alpha_i^{new} = \frac{1 - \alpha_i^{old}\Delta_{ii}}{\mu_i^2}, \tag{11}$$

where μ_i is the $i\text{-}th$ posterior mean weight w_i^* and Δ_{ii} is the $i\text{-}th$ diagonal element of the covariance. The procedure is repeated until it converges to a fixed value or the maximum number of iterations is reached. In order to tackle multiclass situations using the RVM method, two possible strategies could be used [36]. The first one is the one-against-all strategy. L binary classifiers will be created for an L-class classification and each classifier is trained to separate one class from the others. The second strategy is one-versus-one. There are $L(L-1)/2$ binary classifiers created to separate every two classes. In this study, the first method is adopted as it is more computational efficient. The context classification use the accelerometer, gyroscope and magnetometer data, from multiple epoch over a 50% overlapped sliding time window of 2-second duration.

VI. Step Length Estimation

A. Step Length k-parameters Calibration

According to the state of the art [26],[37], the step length can be expressed as a linear combination of the step frequency f_s and the specific force variance σ_f^2:

$$SL = k_0 + k_1 f_s + k_2 \sigma_f^2, \tag{12}$$

where k_1 and k_2 are respectively the coefficients of the step frequency and the specific force variance, while k_0 is individually computed for each subject.
In this work, the k_0,k_1 and k_2 parameters are determined for six contexts: stationary, walking, walking sideways, climbing stairs, descending stairs and running. Considering the path illustrated in Fig. 3, the segments which correspond to walking, walking sideways and running, have been fitted out with benchmarks arranged every 10 meters: the subject has been followed with a video camera along the path and the video recordings have enabled the sensor data collected within the physical benchmarks to be time tagged. The collected dataset have been divided into two groups: *Group A* considers the walking, walking sideways and running contexts, and consists of subgroups of 10-meter segments for each subject, accordingly with the segmentation by benchmarks; while *Group B* considers the walking upstairs and downstairs contexts, and consists of subgroups of single step segments for each subject. The k_0, k_1 and k_2 coefficients are determined for each context by solving for the linear system of equations

$$\boldsymbol{H}\boldsymbol{k}_{0,1,2} = \boldsymbol{d}, \tag{13}$$

where,

$$H = \begin{bmatrix} n_{1,1} & 0 & \ldots & 0 & F_{1,1} & \Sigma_{1,1}^2 \\ 0 & n_{1,2} & \ddots & \vdots & F_{1,2} & \Sigma_{1,2}^2 \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots \\ 0 & \ldots & 0 & n_{1,15} & F_{1,15} & \Sigma_{1,15}^2 \\ n_{2,1} & 0 & \ldots & 0 & F_{2,1} & \Sigma_{2,1}^2 \\ 0 & n_{2,2} & \ddots & \vdots & F_{2,2} & \Sigma_{2,2}^2 \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots \\ 0 & \ldots & 0 & n_{2,15} & F_{2,15} & \Sigma_{2,15}^2 \\ & & & \vdots & & \\ & & & \vdots & & \\ n_{s,1} & 0 & \ldots & 0 & F_{s,1} & \Sigma_{s,1}^2 \\ 0 & n_{s,2} & \ddots & \vdots & F_{s,2} & \Sigma_{s,2}^2 \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots \\ 0 & \ldots & 0 & n_{s,15} & F_{s,15} & \Sigma_{s,15}^2 \end{bmatrix}, \tag{14}$$

$$\begin{aligned} \boldsymbol{k}_{0,1,2} &= \begin{bmatrix} k_{0,1},\ k_{0,2},\ \ldots,\ k_{0,15},\ k_1,\ k_2 \end{bmatrix}^T, \\ \boldsymbol{d} &= \begin{bmatrix} d_{1,1},\ d_{1,2},\ \ldots,\ d_{s,15} \end{bmatrix}^T, \\ F_{i,j} &= \sum_{p=1}^{N} f_{s,p}, \quad \Sigma_{i,j}^2 = \sum_{p=1}^{N} \sigma_{f,p}^2. \end{aligned} \tag{15}$$

$k_{0,j}$ is the constant term which is individually computed for the $j\text{-}th$ subject, and $d_{i,j}$ is the length corresponding to the $i\text{-}th$ data segment and the $j\text{-}th$ subject. In the case of *Group A*, $n_{i,j}$ is the total number of steps detected in the $i\text{-}th$ data segment which refers to the $j\text{-}th$ subject; $\boldsymbol{d}$ is the vector which comprises the lengths corresponding to the data segments, i.e. 10 meters; $F_{i,j}$ and $\Sigma_{i,j}^2$ are respectively the summations of the step frequencies and specific force variances of the N steps detected in the $i\text{-}th$ data segment which refers to the $j\text{-}th$ subject; the s index refers to the last data segment for each subject.
In the case of *Group B*, $n_{i,j}$ refers to a single step and is equal to 1; $\boldsymbol{d}$ is the stair length, which is equal to $0.29\ m$ in the case of climbing stairs or $0.43\ m$ in the case of descending stairs. The least squares solution for the (13) is

$$\boldsymbol{k}_{0,1,2} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{d}. \tag{16}$$

B. Step Length Estimation Methods

This Section describes the HC-SLE and the WC-SLE methods, both of which exploit the k-parameters estimated during the calibration process, in order to determine the step length for each pedestrian context by linearly combine a constant, the step frequency and the acceleration variance, as described in (12). In particular, the HC-SLE algorithm selects the step length which corresponds to the context with the highest probability, while in the WC-SLE, the step length determined for each context are weighted by the context

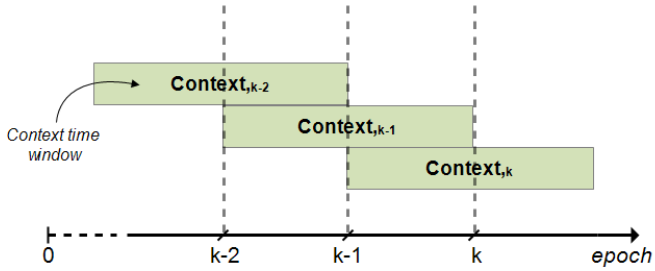

Fig. 11. Time framework of the step length estimation system.

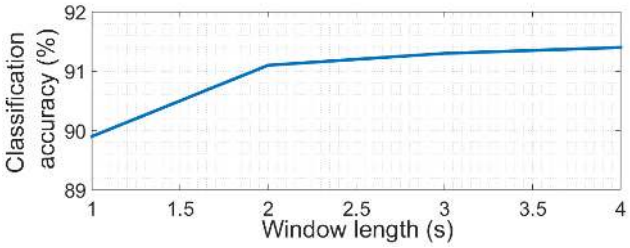

Fig. 12. Context RVM classification accuracy on average against the context window length.

probabilities.

Referring to the time diagram illustrated in Fig. 11, both methods update the traveled distance at each k-th epoch, which corresponds to the end boundary of the $(k\text{-}1)th$ context time window. The latter is 50% overlapped and its duration is 2 seconds. This particular duration maximizes the trade-off between a real-time analysis and the minimum information needed to provide an effective context information, as shown in Fig.12. The improvement on the RVM classification accuracy by using a time window longer than 2 seconds is negligible. A too short window does not enable the context determination process to effectively evaluate the traveled distance along path segments where the context is uncertain.

1) HC-SLE method: assuming the epoch k as the current one (see Fig. 11), the HC-SLE algorithm defines

$$d_{k,C_n} = \sum_{p=1}^{M} k_0^{C_{n,p}} + k_1^{C_{n,p}} f_{s,p} + k_2^{C_{n,p}} \sigma_{f,p}^2, \qquad (17)$$

$$\boldsymbol{d}_k = \begin{bmatrix} d_{k,C_1} \\ d_{k,C_2} \\ \vdots \\ d_{k,C_6} \end{bmatrix}^T, \boldsymbol{Context}_{,k-1} = \begin{bmatrix} Pr\{C_{k-1,1}\} \\ Pr\{C_{k-1,2}\} \\ \vdots \\ Pr\{C_{k-1,6}\} \end{bmatrix}, \qquad (18)$$

$$j = \underset{i}{\text{argmax}} \Big\{ Pr\{C_{k-1,i}\} \in \boldsymbol{Context}_{,k-1} \Big\}, \qquad (19)$$

$$\boldsymbol{Selection}_{,k-1} = \Big[\delta_{j,1},\ \delta_{j,2},\ \ldots,\ \delta_{j,6} \Big]^T. \qquad (20)$$

The d_{k,C_n} term in (17) represents the distance traveled between the epochs $(k\text{-}1)$ and k, when the pedestrian context C_n is considered: in particular, $k_0^{C_{n,p}}$, $k_1^{C_{n,p}}$ and $k_2^{C_{n,p}}$ are the step length k-parameters referred to the p-th detected step and the pedestrian context C_n; $f_{s,p}$ and $\sigma_{f,p}^2$ are respectively the step frequency and the specific force variance of the p-th detected step, and M is the total number of steps detected within the epochs $(k\text{-}1)$ and k.

In (18), the $\boldsymbol{d}_k$ row vector includes the distances traveled between the epochs $(k\text{-}1)$ and k for each context C_n, with $n = 1, .., 6$ (see Table I); while the $\boldsymbol{Context}_{,k-1}$ vector comprises the context probabilities for each context C_n within the context window $(k\text{-}1)$.

The $\boldsymbol{Selection}_{,k-1}$ vector in (20) refers to the context window $(k\text{-}1)$ and comprises the Dirac delta functions [38] which are all equal to zero except the one corresponding to the context with the highest probability.

The algorithm 1 describes the HC-SLE routine operations performed at the epoch k.

Algorithm 1 Highest Context-based SLE Algorithm

$$pd_k^{HC} = \boldsymbol{d}_k \cdot \boldsymbol{Selection}_{,k-1}$$

$$d_{k-1}^{HC} = \boldsymbol{d}_{k-1} \cdot \boldsymbol{Selection}_{,k-1}$$

$$D_{k-1}^{HC} = D_{k-2}^{HC} + \tfrac{1}{2}(d_{k-1}^{HC} + pd_{k-1}^{HC})$$

$$PD_k^{HC} = PD_{k-1}^{HC} + \tfrac{1}{2}(d_{k-1}^{HC} - pd_{k-1}^{HC}) + pd_k^{HC}$$

The distances pd_k^{HC}, d_{k-1}^{HC}, D_{k-1}^{HC}, PD_k^{HC} are sequentially determined.

pd_k^{HC} is the provisional distance traveled between the epochs $(k\text{-}1)$ and k: the scalar product between $\boldsymbol{d}_k$ and $\boldsymbol{Selection}_{,k-1}$ selects the distance d_{k,C_j}, whose j-index identifies the context C_j with the highest probability, as defined in (19). The use of the "provisional" term is because the context probability information within the context window k is not available at the current epoch k; therefore, the information within the context window $(k\text{-}1)$ is exploited.

d_{k-1}^{HC} is the final distance traveled between the epoch $(k\text{-}2)$ and $(k\text{-}1)$. In this case, the "final" term specifies that the context probability information within the context window $(k\text{-}1)$ is available at the current epoch k.

D_{k-1}^{HC} is the total final distance traveled between the epochs 0 and $(k\text{-}1)$: the total final distance D_{k-2}^{HC}, traveled between the epochs 0 and $(k\text{-}2)$, is added up to the arithmetic mean between the final and the provisional distances traveled between the epochs $(k\text{-}2)$ and $(k\text{-}1)$.

Finally, PD_k^{HC} is the total provisional distance traveled between the epochs 0 and k: the total provisional distance traveled between the epochs 0 and $(k\text{-}1)$ is added up to the arithmetic mean between the final and the provisional distances traveled between the epochs $(k\text{-}2)$ and $(k\text{-}1)$, and the provisional distance traveled between the epochs $(k\text{-}1)$ and k.

2) WC-SLE Method: assuming the epoch k as the current one (see Fig. 11), the WC-SLE algorithm defines the terms

$\boldsymbol{d_k}$ and $\boldsymbol{Context_{,k-1}}$, as expressed in (18) for the HC-SLE algorithm.
The algorithm 2 describes the WC-SLE routine operations performed at the epoch k.

Algorithm 2 Weighted Context-based SLE Algorithm

if $\frac{max_2(\boldsymbol{Context}_{,k-1})}{max(\boldsymbol{Context}_{,k-1})} < 0.8$ **then**

go to HC-SLE Algorithm
else

$$pd_k^{WC} = \boldsymbol{d}_k \cdot \boldsymbol{Context}_{,k-1}$$

$$d_{k-1}^{WC} = \boldsymbol{d}_{k-1} \cdot \boldsymbol{Context}_{,k-1}$$

$$D_{k-1}^{WC} = D_{k-2}^{WC} + \tfrac{1}{2}(d_{k-1}^{WC} + pd_{k-1}^{WC})$$

$$PD_k^{WC} = PD_{k-1}^{WC} + \tfrac{1}{2}(d_{k-1}^{WC} - pd_{k-1}^{WC}) + pd_k^{WC}$$

end if

Given the $\boldsymbol{Context}_{,k-1}$ vector as input, the $max(-)$ function outputs the highest context probability, while the $max_2(-)$ function outputs the second highest one. If the ratio between the second highest and the highest context probability is lower than a threshold is equal to 0.8, the context with the highest probability is considered predominant with respect to the other ones; therefore, the HC-SLE algorithm is performed. Otherwise the distances pd_k^{WC}, d_{k-1}^{WC}, D_{k-1}^{WC}, PD_k^{WC} are sequentially determined.
As for the threshold value, it has been set to 0.8 to minimize the error in the estimated traveled distance either when the context is clearly determined or when the context is uncertain, e.g. during context transitions.
pd_k^{WC} is the provisional distance traveled between the epochs (k-1) and k: the distances determined for each context C_n and included in the row vector $\boldsymbol{d}_k$, are weighted by the context probabilities included in the vector $\boldsymbol{Context}_{,k-1}$.
d_{k-1}^{WC} is the final distance traveled between the epochs (k-2) and (k-1). In contrast with the provisional distance computation, the distances included in the row vector $\boldsymbol{d_{k-1}}$ refer to the distances traveled for each context in the previous interval, i.e. between the epochs (k-2) and (k-1). D_{k-1}^{WC} and PD_k^{HC} are respectively the total final distance traveled between the epochs 0 and (k-1), and the total provisional distance traveled between the epochs 0 and k: they are both determined as for the HC-SLE method.

VII. Results and Discussion

A. Context Classification

The context classification model is trained with sensor data recorded from fifteen subjects; in particular, the features in Table II are determined by the specific force, angular rate, and magnetic field sensors, and are given as input to the classification model. The latter employs a 10-fold cross-validation technique for the model validation.

TABLE III
Confusion matrix of the RVM model. Pedestrian contexts: stationary (S), walking (W), walking sideways (WS), walking upstairs (WU), walking downstairs (WD), running (R).

		Predicted context					
		S	W	WS	WU	WD	R
True context	S	93.7%	0.9%	4.9%	0.5%	0%	0%
	W	0%	85.2%	4.8%	9%	1%	0%
	WS	0%	6%	90%	2.2%	1.8%	0%
	WU	0%	9.8%	3.4%	82.5%	4.3%	0%
	WD	0%	1%	0.8%	3%	95.2%	0%
	R	0%	0%	0%	0%	0%	100%

The particular sequence and length of the context segments illustrated in Fig.3, have been considered only for the assessment of the step length estimation routines. In order to avoid the class imbalance, the context classification RVM model has been trained considering the same number of movements for each context. Different short path segments traveled twice have been considered for each context. For each subject, 40 movements relevant to each context have been selected: in total, 600 movements have been evaluated for each context. Considering all pedestrian contexts, the dataset used for context classification consists of 3600 movements.
The RVM method used for context classification achieves a predictive accuracy of 91.1% and its confusion matrix is illustrated in Table.III: walking and climbing stairs are the most confused contexts and produce the main contribution to the overall misclassification rate.

B. Step Length Estimation

The results of the SLE algorithms are divided according to three subgroups of testing subjects: for each subgroup, the data collected from 5 subjects is used to test the SLE algorithms, while the data collected from the remaining 10 subjects is used to train the context classification RVM model.
The total final distances D^{HC} and D^{WC}, which respectively refer to the HC-SLE and WC-SLE algorithms described in Section VI, are determined at the end of the path illustrated in Fig.3. These distances are compared and the results are shown in Fig.13 for each subject and for each SLE method. The root-mean-square percentage error (%RMSE) is computed by comparing the actual path distance with the total final distances determined by the SLE algorithms. Table IV shows the %RMSE obtained for each subgroup of 5 testing subjects and across all 15 subjects.
The WC-SLE algorithm exhibits the best %RMSE performance for the subgroups of testing subjects 6-10 and 11-15, while the HC-SLE algorithm reveals the best %RMSE performance for the subgroup of subjects 1-5. However, the weighted context-based SLE algorithm confirms the best %RMSE performance across all 15 subjects. By analysing in depth the percentage error on total final distance traveled by each subject, described in Table V, it possible to note that the HC-SLE algorithm clearly provides better performance than

TABLE IV
%RMSE ON TOTAL FINAL DISTANCE TRAVELED BY EACH SUBGROUP OF TESTING SUBJECTS AND ACROSS ALL 15 SUBJECTS.

	Subgroup of subjects 1-5		**Subgroup of subjects 6-10**		**Subgroup of subjects 11-15**		**Across all 15 subjects**	
SLE method	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE
%RMSE	4.5%	5.7%	6.2%	5.9%	11.5%	3.6%	7.9%	5.2%

TABLE V
%ERROR ON TOTAL FINAL DISTANCE TRAVELED BY EACH SUBJECT. THE RESULTS ARE DIVIDED IN THREE SUBGROUPS OF TESTING SUBJECTS.

	Subgroup of subjects 1-5									
Subject index	1		2		3		4		5	
SLE method	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE
%Error on total traveled ditance	0.6%	5.1%	-3.3%	1.2%	-5.7%	4.5%	-3%	-3.3%	-6.9%	-10.3%

	Subgroup of subjects 6-10									
Subject index	6		7		8		9		10	
SLE method	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE
%Error on total traveled ditance	-7.8%	-9%	-3.6%	0%	-6%	-7.5%	-4.5%	-0.3%	-7.5%	-6%

	Subgroup of subjects 11-15									
Subject index	11		12		13		14		15	
SLE method	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE
%Error on total traveled ditance	-13.6%	-3%	-13.5%	-3.9%	-7.8%	1.8%	-7.5%	1.2%	-13.1%	-6.1%

TABLE VI
%ERROR ON DISTANCE TRAVELED UNDER DIFFERENT PEDESTRIAN CONTEXTS, AVERAGED ACROSS ALL 15 SUBJECTS.

Context	Walking (209.9 meters)		Walking upstairs (20.6 meters)		Walking downstairs (30.5 meters)		Walking sideways (20 meters)		Running (50 meters)	
SLE method	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE	HC-SLE	WC-SLE
% Error on traveled distance	-8.5%	-5.1%	6.2%	3.4%	-5.5%	-4.2%	-5.9%	-2.5%	-4.8%	-3.3%

the WC-SLE one only for the subjects 1 and 5 (13% out of all subjects). For the subjects 3, 4, 6, 8, and 10 (33% out of all subjects), both the SLE algorithms have similar performance. Whereas, for the subjects 2, 7, 9, 11, 12, 13, 14, and 15 (54% out of all subjects), the proposed WC-SLE algorithm presents an evident better result with respect to the HC-SLE algorithm. The results in Table V can assume a positive or negative sign depending on whether the total traveled distance computed by the SLE algorithms is overestimated or underestimated with respect to the actual path length.

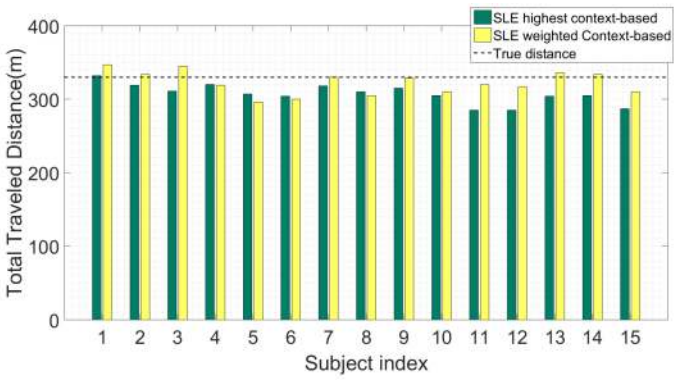

Fig. 13. Total final distances traveled by the subjects and computed for each considered SLE algorithm.

Table VI shows the percentage error on distance traveled under different pedestrian contexts, averaged across all 15 subjects. While the distance traveled under the walking upstairs context is overestimated, all the other ones are underestimated. The proposed WC-SLE algorithm permits to improve the accuracy on the estimated traveled distance under all contexts: the weighted approach achieves better performance when coping with critical path segments where the context is uncertain, e.g. when context transitions occur.

The threshold introduced in the WC-SLE algorithm (algorithm 2 described in Section VI.B) allows the optimization of the step length estimation process. By varying the threshold from 1 to 0, the WC-SLE algorithm would pass from the HC-SLE baseline to a fully WC-SLE approach, as shown in Fig.14. If the threshold is set to 1, the conditional expression in the Algorithm 2 would always be true, leading the WC-SLE to work as the HC-SLE algorithm. This approach could work fine when the pedestrian context is always clear. On the other hand, if the threshold is set to 0, the conditional expression in the Algorithm 2 would always be false, leading to a fully WC-

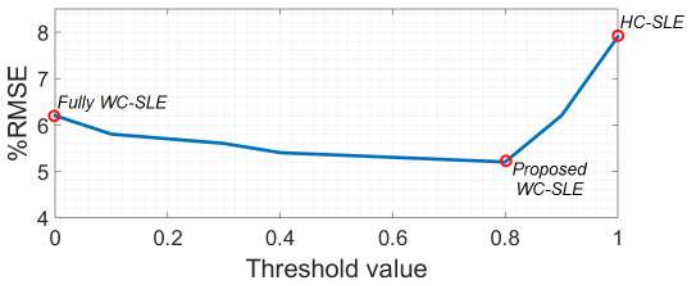

Fig. 14. Performance of the Algorithm 2 with a variable threshold.

SLE version, in which the estimated step lengths are always weighted by the context probabilities, even when the context is quite clear. This approach could work fine when the context is always uncertain. In a real daily life scenario, both clear and uncertain contexts are present; in particular, the presence of a clear context is more likely during a steady context, while the presence of an uncertain context is more likely during the transition between one context to another. The threshold value set to 0.8 leads the WC-SLE algorithm to achieve the best performance in terms of %RMSE. The computationally complexity of the WC-SLE algorithm grows linearly with respect to the number of the considered pedestrian contexts. In particular, both the provisional distance and the final distance defined over each context window, are achieved as the results of scalar products between a row vector containing the step lengths associated to the pedestrian contexts and the vector which includes the pedestrian context probabilities. Note that both the total provisional distance and the final distance do not depend on the number of considered contexts.

VIII. Conclusion

In this work, a weighted context-based SLE algorithm has been demonstrated: the step lengths determined as a linear combination of a constant, the step frequency and the specific force variance, are weighted by the pedestrian context probabilities. The proposed SLE algorithm is compared with the highest context-based method, which considers the step lengths corresponding to the context with the highest probability.

The SLE algorithms are part of a PDR system which defines the step detection and the context classification routines. The step detection aims to detect the step time boundaries and is based on the continuous wavelet transform analysis. The context classification routine exploits the relevance vector machine method to determine the pedestrian context probabilities. Moreover, a dataset of pedestrian activities and actions has been collected. Fifteen subjects, which have been equipped with a waist-belt smartphone, are asked to travel along a predefined path, shown in Fig.3. Six pedestrian contexts have been considered: stationary, walking, walking sideways, climbing and descending stairs, and running.

The RVM method used for context classification has achieved a predictive accuracy of 91.1% and defines walking and climbing stairs as the most confused contexts which produce the main contribution to the overall misclassification rate. Either the total distance traveled by the subjects or the distance traveled under separated contexts, have been determined for each SLE methods. The %RMSE has been computed by comparing the actual path distance with the total distances determined by the SLE algorithms. The weighted context-based SLE algorithm has exhibited an %RMSE of 5.2% across all 15 subjects providing significantly better performance than the highest context-based SLE algorithm which exhibited an %RMSE of 7.9%.

The percentage error on distance traveled under different contexts has been determined. While the distance traveled under the walking upstairs context is overestimated, all the other ones are underestimated. The proposed WC-SLE algorithm permits to improve the accuracy on the estimated distance traveled under all contexts: the weighted approach achieves better performance when coping with critical path segments where the context is uncertain.

References

[1] "Location Based Services (LBS) and Real–Time Location Systems (RTLS) Market – Global Forecast to 2020," *marketsandmarkets.com*, vol. TC 2371, 2015.

[2] A. Küpper, "Location-Based Services: Fundamentals and Operation," *marketsandmarkets.com*, vol. TC 2371, 2005.

[3] P. Groves, "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems," *2nd Edition, Artech House*, 2013.

[4] C. J. Mather, P. D. Groves, and M. R. Carter, "A Man Motion Navigation System Using High Sensitivity GPS, MEMS IMU and Auxiliary Sensors," *Proc. ION GNSS 2006, Fort Worth, TX, September*, pp. 2704–2714, 2006.

[5] J. Käppi, J. Syrjärinne, and J. Saarinen, "MEMS-IMU Based Pedestrian Navigator for Hand.held Devices," *Proc. ION GPS 2001, Salt Lake City, UT, September*, pp. 1369–1373, 2001.

[6] T. Judd, "A Personal Dead Reckoning Module," *Proc. ION GPS 1997, Kansas, MO, September*, pp. 47–51, 1997.

[7] H. Gao and P. D. Groves, "Context Determination for Adaptive Navigation using Multiple Sensors on a Smartphone," *Proc. ION GNSS+ 2016, Portland, Oregon, Sept. 12-16*, 2016.

[8] Q. Ladetto, "On Foot Navigation: Continuous Step Calibration Using both Complementary Recursive Prediction and Adaptive Kalman Filtering," *Proc. ION GPS 2000, Salt Lake City, UT, September*, pp. 1735–1740, 2000.

[9] V. Renaudin, M. Susi, and G. Lachapelle, "Step Length Estimation Using Handheld Inertial Sensors," *Sensors*, vol. 12, pp. 8507–8525, 2012.

[10] J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. G. Boronat, "Comparison and Evaluation of Acceleration Based Step Length Estimators for Handheld Devices," *Proc. IPIN 2010, Zürich, Switzerland,15-17 September*, 2010.

[11] D. Alvarez, R. C. Gonzalez, A. Lopez, and J. C. Alvarez, "Comparison of step length estimators from weareable accelerometer devices," *Proc. 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), New York, NY, USA, August-September*, pp. 5964–5967, 2006.

[12] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[13] J. Qian, L. Pei, R. Ying, X. Chen, D. Zou, P. Liu, and W. Yu, "Continuous Motion Recognition for Natural Pedestrian Dead Reckoning Using Smartphone Sensors," *Proc. ION GNSS+ 2014, Tampa, Florida, September 8-12*, pp. 1796–1801, 2014.

[14] D. Loh, S. Zihajehzadeh, R. Hoskinson, H. Abdollahi, and E. J. Park, "Pedestrian Dead Reckoning With Smartglasses and Smartwatch," *IEEE Sensors Journal*, vol. 16, no. 22, pp. 1421–1429, 2016.

[15] M. Susi, V. Renaudin, and G. Lachapelle, "Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users," *IEEE Sensors Journal*, vol. 13, pp. 1539–1562, 2015.

[16] H. Zhang, W. Yuan, Q. Shen, T. Li, and H. Chang, "A handheld inertial pedestrian navigation system with accurate step modes and device poses recognition," *IEEE Sensors*, vol. 15, no. 3, pp. 1421–1429, 2015.

[17] A. Norrdine, Z. Kasmi, and J. Blankenbach, "Step Detection for ZUPT-Aided Inertial Pedestrian Navigation System Using Foot-Mounted Permanent Magnet," *IEEE Sensors*, vol. 16, no. 17, pp. 6766–6773, 2016.
[18] Ayrulu-Erdem and B. Barshan, "Leg motion classification with artificial neural networks using wavelet-based features of gyroscope signals," *Sensors*, vol. 11, pp. 1721–1743, 2011.
[19] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1780–1786, 2014.
[20] A. R. Pratama and H. R. Widyawan, "Smartphone-based pedestrian dead reckoning as an indoor positioning system," *Proc. International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, September*, 2012.
[21] M. Susi, D. Borio, and G. Lachapelle, "Accelerometer signal features and classification algorithms for positioning applications," *Proc. International Technical Meeting of the Institute of Navigation (ION ITM),San Diego, USA*, pp. 158–169, 2011.
[22] S. Zhang, P. McCullagh, C. Nugent, and H. Zheng, "Activity monitoring using a smart phone's accelerometer with hierarchical classification," *Proc. IEEE 6th International Conference on the Intelligent Environments (IE), Kuala Lumpur, Malaysia, July*, 2010.
[23] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.
[24] P. Groves, H. F. S. Martin, K. Voutsis, D. J. Walter, and L. Wang, "Context Detection, Categorization and Connectivity for Advanced Adaptive Integrated Navigation," *Proc ION GNSS 2013*, pp. 2704–2714, 2013.
[25] J. Prieto, S. Mazuelasd, and M. Z. Win, "Context-Aided Inertial Navigation via Belief Condensation," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 64, no. 12, Jube 15, 2016.
[26] S. H. Shin and C. G. Park, "Adaptive step length estimation algorithm using optimal parameters and movement status awareness," *Medical Engineering and Physics*, pp. 1064–1071, 2011.
[27] L. H. Lee, B. Shin, C. Kim, laehun Kim, S. Lees, and T. Lee, "Real Time Adaptive Step Length Estimation for Smartphone User," *Proc. International Conference on Control, Automation and Systems (ICCAS)*, 2013.
[28] L. Pepa, G. Marangoni, L. C. Matteo Di Nicola, F. Verdini, L. Spalazzi, and S. Longhi, "Real Time Step Length Estimation on Smartphone," *Proc. ICCE 2016*, 2016.
[29] N. Ho, P. H. Truong, and G. M. Jeong, "Step-Detection and Adaptive Step-Length Estimation for Pedestrian Dead-Reckoning at Various Walking Speeds Using a Smartphone," *Sensors*, vol. 16, 2016.
[30] A. Martinelli, S. Morosi, and E. D. Re, "Daily Living Movement Recognition for Pedestrian Dead Reckoning Applications," *Mobile Information Systems*, vol. 2016, 2016.
[31] A. Martinelli, S. Morosi, and E. D. Re, "Daily movement recognition for Dead Reckoning applications," *Proc. IPIN, Banff, Canada*, 2015.
[32] S. Mallat, "A Wavelet Tour of Signal Processing: The Sparse Way," *3rd Edition, Elsevier, Amsterdam, Netherlands,*, 2009.
[33] M. B. D. Rosario, S. J. Redmond, and N. H. Lovell, "Tracking the evolution of smartphone sensing for monitoring human movement," *Sensors*, vol. 15, pp. 18 901–18 933, 2015.
[34] W. He, Y. Guo, C. Gao, and X. Li, "Recognition of human activities with wearable sensors," *Journal on Advances in Signal Processing*, vol. 12, pp. 704–714, 2012.
[35] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.
[36] C. M. Bishop, "Pattern recognition and machine learning," *Company New York*, 2006.
[37] S. H. Shin, C. G. Park, J. W. Kim, H. S. Hong, and J. M. Lee, "Adaptive Step Length Estimation Algorithm Using Low-Cost MEMS Inertial Sensors," *Proc. IEEE Sensors Applications Symposium San Diego, California USA, 6-8 February*, pp. 1064–1071, 2007.
[38] H. Sadri, "Mathematical Methods: For Students of Physics and Related Fields," *Springer New York*, pp. 139–170, 2009.

Alessio Martinelli received his Ph.D. degree in information engineering from the University of Florence in April 2017. He is currently a research fellow at the Department of Information Engineering of the same university. His research activity is mainly focused on positioning and navigation solutions: cooperative GNSS positioning, GNSS/INS integration techniques, and pedestrian dead Reckoning navigation.

Han Gao is a PhD student at University College London (UCL) in the Engineering Faculty's Space Geodesy and Navigation Laboratory (SGNL). He received a Bachelor's degree in Aerospace Engineering from Shanghai Jiao Tong University (SJTU) in 2014. He is interested in multi-sensor contextual navigation and positioning techniques.

Paul Groves is a Senior Lecturer (associate professor) at UCL, where he leads a program of research into robust positioning and navigation within the Space Geodesy and Navigation Laboratory (SGNL). He joined in 2009, after 12 years of navigation systems research at DERA and QinetiQ. He is interested in all aspects of navigation and positioning, including improving GNSS performance under challenging reception conditions, advanced multisensor integrated navigation, and novel positioning techniques. He is an author of more than 80 technical publications, including the book Principles of GNSS, Inertial and Multi-Sensor Integrated Navigation Systems, now in its second edition. He is the recipient of the 2016 Institute of Navigation Thurlow Award and a Fellow of the Royal Institute of Navigation. He holds a bachelor's degree and doctorate in physics from the University of Oxford.

Simone Morosi received his Ph. D. degree in information and telecommunication engineering from the University of Florence in 2000. He is currently an assistant professor at the same university. His present research interests focus on communication systems, energy-efficient wireless communications and localization techniques. He is an author of more than 100 papers in international journals and conference proceedings.