Local HMM for indoor positioning based on fingerprinting and displacement ranging

Ayong Ye\textsuperscript{1}, Jianfei Shao\textsuperscript{1}, Li Xu\textsuperscript{1}, Jianwei Chen\textsuperscript{1}, Jinbo Xiong\textsuperscript{1}

\textsuperscript{1}Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, People's Republic of China

Abstract: Received signal strength (RSS) in wireless networks is widely adopted for indoor positioning purpose because of its low cost and open access properties. However due to the sophisticated propagation of radio signals, the RSS shows a significant variation during pedestrian walking, which introduces critical errors in deterministic indoor positioning. To solve this problem, the authors present a novel method to improve the indoor pedestrian positioning accuracy by modelling fingerprinting and information on the movement into a hidden Markov models (HMMs). They divide the whole continuous positioning process into specified-size sub-processes, which could efficiently reduce the accumulative and resonance error caused by iterative estimation. They use a accelerometer sensor to provide the information on the movement distance to calculate the transition probability of the HMMs. In their experiments, they demonstrate that, compared with the deterministic pattern matching algorithm, the proposed method greatly improves the positioning accuracy and shows robust environmental adaptability.

1 Introduction

With the increasing demand for location-based services, indoor location estimation has attracted considerable attention during the last few years [1]. Examples of such services include smart marketing, precise advertisement, indoor navigation, stuff monitoring and so forth. In many indoor environments, knowing the user's location with high accuracy is the key to providing location based services (LBS) guaranteed by quality of experience.

Global positioning system (GPS) is currently the most widely used technology for location tracking in outdoor environments. However, for indoor localisation, GPS signals cannot reach the receivers and the positioning accuracy is too low to support indoor services. Therefore, a great number of researches with different technologies (like WLAN/Wi-Fi, RFID, ZigBee, UWB, and Bluetooth) have been done to address the indoor localisation problem [2–4]. Among these, the WLAN fingerprint positioning technology has become a research hotspot in the indoor localisation field on account of the popularity of WLAN signals in indoor environments. Many fingerprint-based localisation algorithms, utilising the received signal strength (RSS) provided by the off-the-shelf wireless devices, have been proposed to meet the changes of indoor environment [5–7]. However due to the sophisticated propagation of radio signals, the RSS shows a significant variation during pedestrian walking, which introduces critical errors in deterministic indoor positioning.

Due to the proliferation of Smartphones with various sensors, such as accelerometer, gyroscope, magnetometer and so forth, dead reckoning [8] is modified and applied to indoor positioning. For the outdoor positioning, dead reckoning approach is a good solution for the substitution under the circumstances in which GPS signal is unavailable, for example, the car is driving into a tunnel. This positioning method is based on the variation of displacement and heading from the inertial sensors embedded in cars. However, such technique is not highly reliable in predicting the location since localisation errors in dead reckoning scheme get cumulated over time.

Hidden Markov models (HMMs) [9] are powerful probabilistic tools for modelling sequential data, and have been applied with success to natural language processing, speech recognition and so forth. In recent years, HMMs are applied to indoor positioning tentatively for combining RSSI fingerprinting method with inertial sensors, and make some contributions [10–12]. In that case, positioning is considered as from an isolated location estimating process to a sequential locations transition process, and the observations (RSSI) and the states (locations) transition are usually modelled as Gaussian distributions over a discrete location. Then the positioning problem is transformed to the prediction problem in HMMs. However, because positioning is considered as a continuous process, iterative estimation may increase the accumulative and resonance error. For instance, once given or estimating a large error, the latter positioning may also make a large error. For this case, we propose a local HMM to localise the continuous positioning process for modelling partial processes. In this way, the whole process will be divided into specified-size sub-processes, and each of them is independent. So the accumulative and resonance error could be reduced efficiently.

Compared with existing indoor localisation systems, our main contributions are as follows:

(i) A local HMM is introduced into indoor positioning by dividing the whole continuous process into independent sub-processes. In this case, accumulative and resonance error could be reduced, and the robustness to fluctuation of RSSI would be improved.

(ii) In our approach, the transition probability distribution is modelled by displacement ranging. An integrated algorithm based on step counting and stride length estimating is presented, in which step counting is based on acceleration peak detection and stride length is determined by a learning method of acceleration mean without user intervention.

(iii) To solve the computationally expensive problem of predication task in a large HMM state space, we utilise k-nearest to reduce the Viterbi algorithm computation. Meanwhile because some irrational locations are eliminated, the positioning error has also been reduced to some extent.

(iv) We implement a prototype on a Smartphone and conduct real-world evaluation in a building. The results show that our approach greatly decreases the positioning error and improves the robustness to RSSI fluctuation.

The rest of the paper is organised as follows. We review related works in Section 2. Section 3 introduces the idea of the proposed approach by legend and formula derivation, and the differences between the local HMM and typical HMMs are presented. Section 4 is about how to model the parameters probability distribution and how to predict the location by a modified Viterbi algorithm. In
Section 5, some experimental results and the comparisons are discussed on our developed prototype. Finally, we draw some conclusions and point to future work.

2 Related works

Recently, in order to increase localisation accuracy and reduce the overhead, several works were proposed to utilise new sensing modalities to achieve high accuracy localisation system. Bahl et al. [13] proposed a classic continuous positioning approach without inertial sensors, which is called Viterbi-like. By the captured multiple RSSI fingerprints and the given start location, the positioning result will be obtained by iteratively computing the shortest path. However, because the location estimating depends on the previously estimated location and existing estimating error, the accumulative error and the resonance error may increase in an iterative positioning process. Besides, some isolated locations could be eliminated due to the shortest distance principle, and the weights of reference points are not considered too.

There are many approaches combining fingerprinting method with inertial sensors have been proposed for indoor localisation in recent years. In [14], a maximum likelihood-based fusion algorithm that integrates a typical Wi-Fi indoor positioning system with a pedestrian dead reckoning system (PDR) is proposed, in which the strength of the PDR system can eliminate the weakness of the Wi-Fi positioning system and vice versa. In [15], Hoang et al. present a system for indoor navigation based on RSSI of access point (AP) and relative position estimates. The relative position is gathered from inertial smartphone sensors using step detection and orientation estimate. These approaches above consider the positioning process from an isolated point to a continuous motion sequence. The irrational results returned by fingerprinting method will be corrected by the transition between locations, so the positioning error decreases and the robustness to RSSI fluctuation is improved. However, because of the iterative estimation, accumulative error may increase over time.

The HMMs have been used in some approaches for the integration of fingerprinting positioning and inertial navigation systems [10–12, 16–17]. In [10], Ni et al. present a pedestrian positioning method using a HMM with a fuzzy pattern recognition algorithm in a WLAN fingerprint system. The fuzzy pattern recognition algorithm follows the rule that the RSSI fading has a positive correlation to the distance between the measuring point and the AP location even during a dynamic positioning measurement. However, to get better localisation accuracy, a larger number of particles and computations are needed. In [11], Seitz et al. present an algorithm for pedestrian navigation based on a HMM that combines Wi-Fi positioning and dead reckoning. The hidden states are the positions of the Wi-Fi fingerprints in the database.

The state transition includes dead reckoning based on step length estimation from acceleration measurements and compass heading calculated from magnetic field measurements. However, this method assumed that the fluctuating RSSI followed a Gaussian distribution. It did not consider that there is a mismatch between RSSI measurements in movement and the fingerprint database. In [12], Ma et al. present a robust HMM-based indoor localisation system, which regards these distinctive geomagnetic anomalies as fingerprints for high precision. In order to improve the localisation accuracy, this method proposes a backward sequences matching algorithm to optimise the HMM and vectorises the backward consecutive geomagnetic signals to fingerprint sequences with the help of PDR. In [16], to reduce the calibration work of the positioning process, Liu et al. employed a Weibull distribution to model the distribution of the RSSI and used an accelerometer sensor to provide the movement distance for calculating the transition probability of the HMM. However, the Weibull distribution cannot accommodate the fluctuating RSSI samples, and the transition probability hardly reflects the real state transition process.

3 Pedestrian positioning method based on HMM

In this section, we will introduce our designed system in detail. First, we briefly describe the system architecture shown in Fig. 1. Two phases including testing and training are adopted. In the training phase, we collect the Wi-Fi signals using mobile devices taken by participants. The Wi-Fi fingerprints are then stored in the fingerprint database according to each interested location in the surrounding environment. After collection, a local HMM training algorithm is adopted to calculate the locations transition probability distribution \( P(s'|s) \) and the RSSI-location dependency probability distribution \( P(o|s) \), respectively, in which \( s \) and \( o \) are the reference location and observation (including RSSI and acceleration). In the testing phase, a user estimates the moving distance and samples the RSSI when he/she enters into a WLAN environment with terminals. We use a displacement ranging module to estimate the distance at each positioning point. First, the step counting submodule uses the commonly used accelerometer of the smartphone to detect and track user steps. Second, it utilises a stride length estimating model to infer the user’s step length. After obtaining the observation space and a settled HMM, the user locations are predicted by utilising the Viterbi algorithm.

3.1 Mathematical model of pedestrian positioning

The HMM is a finite state automaton with stochastic state transitions and observations, which has been successfully applied to speech recognition, acoustic treatment and biological

© The Institution of Engineering and Technology 2018
information. A Markov model is called hidden if it contains an underlying stochastic process that is not directly observable, but can be observed through another stochastic process [18]. Fig. 2 shows the basic architecture of the proposed HMM, where the term $s_i$ is the hidden state at time $t$, and $o_t$ is the observation at time $t$. $P(s_i|s_{i-1})$ is the transition probability from $s_{i-1}$ to $s_i$, $P(o_t|s_i)$ is the conditional probability when the observation is $o_t$ in the hidden state $s_i$. The HMMs also define a joint probability distribution $P(s,o)$ consisting of transition probability distribution $P(s_i|s_{i-1})$ as in

$$P(s_t,o_t) = \prod_{i=1}^{t} P(s_i|s_{i-1})P(o_t|s_i)$$

where $t$ is the last observation time.

In our proposed HMM, the hidden state is referred to the discrete positions, and the observable states are indicated by the RSS measure and the acceleration data. The hidden state transition process is a Markov process, and the observation states are generated by hidden Markov processes according to a certain probability distribution [19]. Given a HMM model and an observation sequence, the location of a client is iteratively estimated by the following HMM process:

$$\arg \max_{s_1, \ldots, s_t} P(s_1, o_1, \ldots, o_t) = \arg \max_{s_1} \frac{P(o_t|s_t)P(s_t)}{P(o_1)}$$

$$= \arg \max_{s_1} P(s_t, o_t)$$

$$= \arg \max_{s_1} P(s_t|o_t)$$

3.2 Pedestrian positioning using Viterbi algorithm

The Viterbi algorithm is proposed on the basis of dynamic programming for finding the optimal hidden state sequence [20], which is discussed in detail in Section 4. Given an observation sequence and a HMM model, the hidden location sequence of user can be more accurately estimated by employing the Viterbi algorithm. However, positioning based on typical HMM is also considered as a continuous process. The pedestrian position is estimated by the HMM algorithm iteratively. At each positioning of a user, the location update is accomplished by adding the current estimated displacement to the previously estimated location. Therefore, the positioning error in typical HMM scheme will get cumulated over time. To decrease the accuracy loss caused by the accumulative error of pedestrian positioning, we divide the whole positioning procedure into many independent sub-process and model a single positioning as an independent HMM process, which is irrelevant to the previous positioning results. In this paper, we refer to a HMM algorithm running for an independent positioning by a local HMM. The idea behind positioning based on a local HMM is precise to only use information from the recent past to come up with a better guess of a user's location.

Fig. 3 shows the differences between a local HMM and the typical HMM. As shown in Fig. 3, given a set observations sequence $O = \{o_1, o_2, o_3, o_4, o_5\}$, a HMM algorithm is performed to guess the location of the pedestrian. The location corresponding to the mid-point of the path is guessed to be the user's location. The weight of an edge between vertices $a$ and $f$ is $d_{af}$ the Euclidean distance between the corresponding locations. In this paper, we define the number of observation samples for a local HMM as the window size. It is worth noting that the initial location is not needed in a local HMM positioning process. The local optimal trajectory determined by computing the probabilities of all probable paths consist of the candidate sets. Then the last location in the determined trajectory will be the estimated location of this positioning process. The transition probability between locations and emission probability are detailed in Section 4.

The computation complexity of the Viterbi algorithm is relative to the number of trajectories. Therefore, to find the most likely of the sequence of the location within an acceptable time, we first sort the reference locations according to the emission probability, and then select only the top 4 locations with the highest probability to combine the candidate trajectories. As shown in Fig. 3, when the number of observations equals to 4 and $k = 4$, the observation sequence of four fingerprints $(R_1, R_2, R_3, R_4)$ and three displacements $(d_1, d_2, d_3)$ are gathered in a local HMM positioning process. We can get the emission probability of real-time fingerprints on reference locations and the transition probability of different locations according to the displacements.

The robustness to RSSI fluctuation works on the overlap effect of transition and emission. Suppose the fingerprints $R_2$ and $R_4$ gathered that have large fluctuation, and $(a, f, h, n)$ is the true trajectory. The positioning results of typical single-point positioning algorithm MAP are $(a, d, h, f)$, but the second and the fourth results are in large error. However, after introducing a HMM, weighting the displacement to the distance of reference locations and emission probabilities, the optimal trajectory by using Viterbi algorithm $(a, f, h, n)$ are the determined results.

4 Implement and algorithm

4.1 Mathematical model

The HMM used in an indoor positioning algorithm, which fuses an inertial navigation system and a WLAN fingerprint positioning system, can be mathematically defined as a quintuple $\lambda = (S, O, \Pi, A, B)$, which includes three probability matrices, and two sets of states $(S, O)$ in our pedestrian positioning method.

$S$ is the reference locations space with its fingerprints databases. $O$ is the set of the real-time fingerprints and acceleration data acquired in localisation stage. $\Pi$ is a matrix which denotes the probabilities of reference points becoming the first location, and the probabilities are assumed as equal to each other without any prior knowledge. $A$ is the set of transition probability distributions which models according to the layout of positioning environment and distance of reference locations. $B$ is the set of emission probability distributions which models according to the training RSSI fingerprints at corresponding reference locations.
4.2 Distance estimation module

Theoretically, the walking distance of the user can be obtained by using the acceleration data provided by smartphone. However, the acceleration for an indoor pedestrian is too small to resist the sensor noises. An alternative approach is to detect the walking step and estimate the distance at each step. Hence, the moving distance can be estimated as num_steps × step_length, where num_steps represents the number of steps and step_length represents the length of each walking step. To estimate the walking distance of user, we present both a peak detection method for step counting and a learning method for stride length. The accelerometer delivers the three-dimensional acceleration vector $a$. We calculate the absolute acceleration value $||a||$ and subtract the gravity constant $g$ (about 9.8 $\text{m/s}^2$) for analysis. The displacement ranging could be divided into two parts: step counting and stride length estimating.

A. Step counting: Since the vertical acceleration fluctuates periodically due to human motion, a peak detection algorithm used to estimate the value of num_steps. The advantages of peak detection step counting are the adaption to the speed, length of a step and noise. The acceleration data is different to different steps, for example, the acceleration peak is small in low-speed walking, but large in high speed. Conventional step counting setting a fixed threshold to detect step counts is not sensitive to these features. Acceleration noise is mainly composed by the accelerometer itself and the shake of device in user walking. The noise caused by accelerometer itself is depended on its performance and stability; therefore we could set a fixed noise threshold according to the acceleration data in stationary status. However, the device shake noise in walking could not just be compared to a set threshold, because the shake acceleration could be beyond the set threshold. However, the peak detection could provide the dynamic acceleration threshold to detect the step counts, and the accuracy of step counts will be improved. The solid line in Fig. 4a shows the dynamic threshold of peak detection step counting, and the dashed fixed threshold is to filter the noise of accelerometer itself. The dynamic threshold is the product of peak mean and a coefficient, and the coefficient normally set as 0.6 according to the results of experiments.

B. Stride length estimating: The value of step length can be easily set to a fixed value according to weight and height of user or estimated by counting the number of walking steps required for the user to traverse a specified distance. However, those approaches above are inaccurate for indoor positioning system, since stride length of a particular user had great variation and vary widely from different scenarios. To improve the effect of the step length estimation, we estimate the step length from the acceleration. A dynamic method based on mean peak value detection is used in this paper to estimate the stride lengths. As can be observed from Fig. 4b, accelerations of users with different stride lengths exhibit considerable different characteristics, such as the mean peak value, and evince relevance between stride lengths and acceleration peak. Therefore, the stride length could be adjusted by calculating the peak mean of a period.

Fig. 4 Distance estimation module
(a) Step counting, (b) Acceleration patterns of users with different strides
4.3 Transition probability distribution

Transition probability is the most critical parameter in a HMM based on positioning, since the positioning accuracy of the Viterbi algorithm depends on the precision of the transition probability. The transition probability in indoor positioning denotes the probability of user movement between reference locations. To obtain the transition probability which can reflect the pedestrian walking pattern in indoor environment, we utilise a Dijkstra algorithm instead of spatial straight-line distance to calculate the walking distances of user, since the user movement is restricted to the indoor layout and structure. During training phase, we create a distance matrix of reference locations by employing Dijkstra’s shortest path algorithm based on the undirected connected graph of indoor the environment [21]. The distance matrix can be defined as \( D = \{d_{ij}|i,j \in S\}, \quad 1 \leq i, j \leq N \), \( d_{ij} \) is the pairwise shortest distance between location \( s_i \) and location \( s_j \). \( N \) is the total number of reference points.

As mentioned before, the user movement \( m \) could be computed according to the captured step counts and adjusted stride length, and \( \sigma_m \) is the average error of displacement ranging which could be adjusted from the interval of collecting the acceleration data once and practical effects. The distribution of displacement to the distance of locations is regarded as Gaussian normally

\[
P'(s_j|s_i, m) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{(m-d_{ij})^2}{2\sigma_m^2}\right) \tag{3}
\]

Thus, the transition probability distributions at different reference locations could be obtained

\[
a_{ij} = P(s_j|s_i, m) = \frac{1}{Z(m, \sigma)} \exp\left(-\frac{(m-d_{ij})^2}{2\sigma_m^2}\right) \tag{4}
\]

where \( Z(m, \sigma) \) is the normalisation factor

\[
Z(m, s_j) = \sum_{i \in S} \exp\left(-\frac{(m-d_{ij})^2}{2\sigma_m^2}\right) \tag{5}
\]

4.4 Emission probability distribution

The emission probability is the conditional probability of real-time fingerprints at the reference locations. Reference fingerprints and real-time fingerprints are defined as follows:

\[
F_i = ((f_{i1}, \sigma_{i1}), (f_{i2}, \sigma_{i2}), (f_{i3}, \sigma_{i3}), \ldots, (f_{in}, \sigma_{in}))
\]

\[
R = (r_1, r_2, r_3, \ldots, r_q)
\]

where \( F_i \) is the reference fingerprint at location \( i \), \( i \in \{1, 2, \ldots, n\} \), \( q \) is the number of access points, \( f_{iq} \) is the average RSSI received from access point \( q \) at reference location \( i \), \( \sigma \) is the standard deviation of the corresponding RSSI, \( R \) is the real-time fingerprint and \( r_2 \) is the real-time RSSI received from access point \( q \). The RSSI from different access points are assumed as independent. The probability can be calculated by two classes of approaches. One is the kernel methods choosing usually Gaussian based on kernel functions as an approximation of the distribution density function of the RSSI [14]. The other is the histogram methods usually requiring a number, hundreds or thousands at least, of samples on each reference location to generate the probability distribution approximating the real situation [15]. To reduce the labour costs in offline stage, we choose the Gaussian kernel method to calculate the emission probability, which needs relatively small samples set. Thus, the emission probability distributions at different reference locations could be obtained

\[
P(R|s_i) = \frac{1}{Z(R, s_i)} \prod_{i=1}^{n} \exp\left(-\frac{(r_i-f_{ij})^2}{2\sigma_{ij}^2}\right) \tag{6}
\]

where \( Z(R, s_i) \) is the normalisation factor

\[
Z(R, s_i) = \sum_{i \in S} \prod_{i=1}^{n} \exp\left(-\frac{(r_i-f_{ij})^2}{2\sigma_{ij}^2}\right) \tag{7}
\]

4.5 Modified Viterbi algorithm

The estimated location can be carried out by the Viterbi algorithm, but the real-time computational cost will increase rapidly with the increasing of reference locations. Therefore, to reduce the real-time cost, candidate locations sets are created first, and only the sequence composed by the locations in candidate sets are calculated. The candidate sets can be built by sorting emission probability mentioned in (6). The Viterbi algorithm for HMMs fills in a dynamic programming table with forward probabilities \( \delta(i) \) defined as the probability of producing the observation sequence up to time \( t \) and being in location \( s_i \) at time \( t \).

**Algorithm description:**

(i) The candidate locations sets are built by calculating the emission probability, \( T \) is the cycle (observation time) of a local HMM positioning process, e.g. \( T=4 \) means there are four observation samples in an observation cycle. The locations in sets are the top \( k \) reference locations of emission probabilities corresponding to the real-time fingerprints.

(ii) When \( t=1 \), the trajectory to some location does not exist yet. Then the forward probability can be replaced with the product of initial probability in that location and emission probability of corresponding fingerprint

\[
\delta(i) = \pi_i \cdot P(R|s_i) \tag{8}
\]

(iii) When \( t>1 \), the probability at \( t \) can be calculated from the forward probability of location at \( t-1 \). The trajectory to some location at \( t \) must pass through the partial optimal trajectory, which means the optimal trajectory at \( t \) must consist of the partial optimal trajectory at \( t-1 \) and some locations at \( t \). To find the estimated location at \( t \), optimal probability, (8) is extended

\[
\delta(j) = \max (\delta(i) \cdot P(s_i|s_j, m) \cdot P(R|s_j)) \tag{9}
\]

(iv) The location \( j \) which makes \( \delta(j) \) maximum is the estimation.

5 Positioning experiments

To evaluate the proposed method and algorithm, we develop a positioning prototype system based on Wi-Fi platform and Android system. In addition, we also perform some detailed experiments in our laboratory at Fujian Normal University. This experiment was conducted in a public space where we follow a casual route. During this experiment we use a local HMM to integrate the fingerprint positioning and the inertial sensors to calibrate the user final location.

5.1 Experiment setup

We use the Huawei C8813Q smartphone for implementing the proposal positioning client application, in which the hardware and application programming interface (API) support sampling the accelerometer sensor with 1000 Hz frequency and scanning the RSSI with an interval of 1 s. In addition, we use NETGEAR WNAP210 as an AP, which can provide a more stable Wi-Fi signal. The layout of our experiment environment is shown in Fig. 5. For the purposes of simplifying the experimental procedure, we only choose the public space, where we follow a casual route (red line). There are a total of five APs on this area, three of them are deployed in the ceiling of corridor, and two of them are deployed in the corner of different rooms. Due to high-density deployment of APs on the floor, almost all the localisation area can receive five signals from different APs. The whole positioning test area has a dimension of 33.5 m × 13 m for localisation and has been divided into uniform 1.5 m × 1.5 m grids. The reference points are set in the centre of each grid. Due to obstacles, such as tables, chairs and platform, some areas are not set up reference
points. Therefore, a total of 100 reference points are set in the positioning area. The positioning map was created as an undirected connected graph, in which the solid points indicate the reference locations and the dotted lines show the connectivity of it.

In the training stage, we first built the Wi-Fi fingerprint map of positioning area by collecting the RSSI value at each reference points into a database. Many measures have been employed to improve the accuracy of the RSSI reference location in each direction, and then record the mean values into the fingerprint database. In addition, we estimated the stride length of the experimenter by calculating the mean of the acceleration peak value during a test period. At last, we created the emission probability distributions of reference location fingerprints and the transition probability distributions according to the coordinates of locations.

In the test stage, the positioning app is running to gather the real-time RSSI and acceleration data. The positioning scenarios are divided into two phases: the one is static positioning and the other one is dynamic positioning. The static positioning is that the users stay in a location for some time, and the dynamic positioning is that the users are walking naturally in a specified path so that the true location and positioning results are available for the purpose of comprehensive evaluation and analysis.

5.2 Performance evaluation

5.2.1 Displacement ranging module: The measurement accuracy of the displacement ranging module depends on the precision of both the step counting algorithm and the stride length estimating algorithm.

In the experiment, we selected ten volunteers with different height as the experimenter, half of whom were boys and half were girls. The measurements are collected by each experimenter walking in the indoor environment 100 m at their normal speed. We used the cumulative distribution function of error (CDF) to calculate the step counting error. Fig. 6 shows the experimental results of the peak mean detection algorithm. Obviously, more than 80% samples are counted precisely of less than one step error and 95% samples are precisely of less than two step errors, when using peak mean coefficient as 0.6 in our experiment. Furthermore, we compared the performance of the peak mean detection algorithm with fixed threshold algorithm. Actually, the performance of the peak mean detection algorithm is better than that of the fixed threshold algorithm. This is because that the peak detection provides a dynamic acceleration threshold to detect the step counts when shaking of device happens. As displacement ranging is processing in 1 s period, the step counting error is from 0 to 2 steps normally, and the stride length error is from 0 to 20 cm. So the distance error of displacement ranging would not beyond 1 m from our experiment, and the $\sigma_m$ is set as 0.5 m by experiment.

5.2.2 Optimum of windows size and $k$: We conducted a simple experiment to find the optimal value of windows size $w$ and top $k$ value. The windows size denotes the number of observations including RSSI and acceleration sampling in a positioning process. When windows size $= 1$, the algorithm is equal to MAP algorithm [16], in which the location of maximum posterior is the estimation. When windows size is a big value, then the algorithm is equal to typical HMM algorithm [10]. The parameter $k$ in Viterbi algorithm is used to reduce the computational cost and eliminate irrational results. The measurements are collected by a user walking alone the planed path at normal walking speed. The planned paths are shown in Fig. 5 as thick line. The estimated positions of use in experiment are roughly shown in Fig. 7a. From Fig. 7a, we can clearly see that the performance of local HMM with $w = 4$ is better than that of other values.

In Fig. 7b, we show how our algorithm performs for different value of the averaging window and $k$. We see that $w = 4$ minimise the degree to which state inference is erroneous, despite suffering from a large number of false transitions. Obviously, when $T < 3$ or $T > 6$, the error of algorithm is equal to MAP algorithm. When $T$ is small, the information of transition is not enough to contribute to the positioning precision. When $T$ is large, the continuity of a process is weak to improve the precision and robustness to RSSI fluctuation. Weighting the positioning precision and the computational costs, the value of $T$ is usually set from 3 to 5. The parameter $k$ denotes the number of reference points in candidate set, which is to reduce the cost of Viterbi algorithm. When $k = 1$, the algorithm is equal to MAP algorithm. The number of candidate points in a set will grow with the increasing of $k$. The error of different $k$ in Fig. 7a shows that the positioning error with growing of $k$ will be improved when $k < 5$. When $k > 5$, the error does not decrease, but the computation overhead will increase. Considering the error reduced by $k$ and the increasing computational costs, the value of $k$ will be set from 3 to 5 normally.

IET Commun., 2018, Vol. 12 Iss. 10, pp. 1163-1170 © The Institution of Engineering and Technology 2018
5.2.3 Positioning accuracy: To determine the localisation performance, we first estimate the path by our positioning system while we follow the route in Fig. 5 for ten times. Fig. 8a shows the CDF of the overall estimation error for ten-time experiment. After this experiment, we found estimation error is <1 m for 50% of the time. Fig. 8b shows the CDF of the estimation while we follow the route in Fig. 5 at different speed. In this experiment, we found estimation error is <1 m for 50% of the times for different speed. These results proof the feasibility and the efficiency of our positioning system under different mobility conditions of the user.

To determine the localisation performance, we estimate localisation error which is the Euclidian distance between the estimated location and the corresponding real location. Two scenarios are set up to evaluate the preformation of our algorithm. The first scenarios are static localisation which positioning for users at rest, and the second scenarios are dynamic localisation which positioning for moving users. The initial location in HMMs is given accurately. We also implement Ni [10], MAP [22] and WicLoc [23] to compare their performance with our system over the same experimental data. Figs. 8c and d show the experimental results of the CDF of the four approaches in static and dynamic scenarios, respectively. In static positioning, the averaged localisation error of our algorithm is 1.9 m, which is smaller than that of Ni (3.1 m), MAP (3.8 m) and WicLoc (4.65 m). In our algorithm, we can find the estimation error of 30% positioning is <1 m for different scenarios while those of 50% is <2 m. As the mechanisms of MAP and WicLoc are independent of movement information, the results would be affected easily by the fluctuation of RSSI. Our approach and Ni utilise the acceleration in static status, so that algorithm knows user is staying in a location, and provides accurate previous location information to the estimation.

In dynamic positioning, the accuracy of all approaches is lower than in static, and the distribution of Ni is clearly worse than local HMM, because the movement in positioning lead to the increasing of RSSI fluctuation and the error of displacement ranging. The HMMs estimate location iteratively, so the fluctuation and error will affect the latter estimation. However, the local HMMs are independent of previous positioning results, so the accuracy is

---

**Fig. 7** Estimated positions by different window size w
(a) Positioning instance using the proposed approach, (b) Positioning error as a function of the T or k

**Fig. 8** Positioning accuracy
(a) CDF of the overall estimation error for following the path in Fig. 5 for ten times, (b) CDF of the estimation error for following the path in Fig. 5 at different speeds, (c) CDF of the estimation error of static positioning, (d) CDF of the estimation error of dynamic positioning
similar than in static positioning, and the effect of fluctuation could be offset to some extent by the transition information. Thus, higher accuracy and robustness can be achieved.

6 Conclusion
In this paper, we propose an indoor positioning approach to robust locate a user using a local HMM that combines fingerprinting with displacement ranging. Through the introduction of the local model instead of the whole process, the positioning accuracy and robustness could be improved significantly, and the accumulative and resonance error would be controlled efficiently. We develop a prototype to evaluate the effectiveness of our approach. Considering the high computational cost especially in positioning stage, the parameters values of improvements in our algorithm have been offered reasonably from the experimental results. In the future, we will concentrate on the problem of high human resource costs in offline stage, especially the application scenarios of large area.

7 Acknowledgments
This work was supported partially by the National Natural Science Foundation of China (61771140), the Natural Science Foundation of Fujian Province (2018J01780), the Key Natural Science Foundation of Fujian Educational Bureau for Young Scholars (JZ160430), the 2015 Science and Technology Projects of Fuzhou University (2015-G-51) and the Joint Funds of the National Natural Science Foundation of China (U1405255).

8 References