Probabilistic Infrastructureless Positioning in the Pocket

Le T. Nguyen and Ying Zhang

Carnegie Mellon University,
NASA Research Park Building 23,
Moffett Field, CA 95035, USA
{le.nguyen,joy.zhang}@sv.cmu.edu

Abstract. With the increasing popularity of smart phones, knowing the accurate position of users has become critical to many context-aware applications. In this paper, we introduce a novel Probabilistic Infrastructureless Navigation (ProbIN) system for GPS-challenging environments. ProbIN uses inertial and magnetic sensors in mobile phones to derive users' current location. Instead of relying on basic laws of physics (e.g. double integral of acceleration equals to displacement) ProbIN uses a statistical model for estimating the position of users. This statistical model is built based on the user's data by applying machine learning techniques from the statistical machine translation field. Thus, ProbIN can capture the user's specific walking patterns and is, therefore, more robust against noisy sensor readings. In the evaluation of our approach we focused on the most common daily scenarios. We conducted experiments with a user walking and carrying the phone in different settings such as in the hand or in the pocket. The results of the experiments show that even though the mobile phone was not mounted to the user's body, ProbIN outperforms the state-of-the-art dead reckoning approaches.

Keywords: Inertial positioning, low-cost inertial sensors, Dead Reckoning, Bayes' theorem, Expectation Maximization.

1 Introduction

With the increasing popularity of smart phones, knowing the accurate position of a user has become critical to many context-aware applications. In outdoor environments, standardized Global Positioning System (GPS) is often used. However, for indoor environments such as airports, hospitals or shopping malls GPS signals are usually unavailable or unreliable.

Most of the existing indoor positioning solutions try to address this problem by utilizing existing infrastructures such as Wi-Fi access points or Bluetooth beacons [12]. In cases when an infrastructure is not available, self-contained systems provide a more flexible solution. These systems use sensors such as accelerometers, gyroscopes and magnetometers. In order to derive a user's current location, the movement of the user is tracked by the continuous logging of sensor readings.

J.Y. Zhang et al. (Eds.): MobiCASE 2011, LNICST 95, pp. 311–330, 2012.

Since this technique does not reply on an external infrastructure, theoretically it can be used in any environment. The main drawback of self-contained positioning approaches is error accumulation. Since the sensors utilized are noisy by nature, the error of position estimation grows with time and distance traveled. Moreover, the noise portion in sensor measurements is significantly higher when the phone is held in the hand versus mounting it on certain parts of the body. Positioning in the hand accelerates the accumulation of error and causes a substantial decrease of estimation accuracy.

The purpose of our work is to deliver a system providing positioning and navigation functionality for consumer mobile devices in GPS-challenging environments. The main contribution of this paper is to introduce a novel probabilistic approach of self-contained positioning providing a user's current location. In order to overcome the problem with noisy sensor readings of consumer mobile devices, a statistical model well-known in the field of the statistical machine translation (SMT) is utilized for the positioning purposes.

In our work, the positioning problem is framed as a noisy-channel problem, where we try to recover the actual user's position from the distorted sensor inputs. To recover the user's position, we use a statistical model to *map* the sensor readings directly to the displacement. This is fundamentally different from state-of-the-art dead reckoning approaches. In these approaches the sensor readings are interpreted by their actual physical meanings, i.e., the accelerometer readings are considered as being the actual acceleration of the device. Thus, theoretically based on the laws of physics the travelled displacement can be obtained by double integrating the acceleration.

In ProbIN the sensor readings are interpreted as observed "signals" which are directly mapped to the corresponding displacement based on a statistical model. The statistical model is trained by using SMT techniques adjusted for positioning purposes. During the training phase, ProbIN builds statistical models from the user's data. These models capture the user's walking patterns and adapts to the sensor errors of the mobile device. Thus, although the sensors on the mobile devices are noisy, ProbIN can still estimate a user's current position at a much higher accuracy rate than state-of-the-art dead reckoning approaches.

This paper extends our previous work [13] where there were several limitations in the way how the phone can be positioned. Since the smart phone used for the evaluation was not equipped with a gyroscope the pitch and roll angles of the device were unknown. Therefore, in the previous experiments the phone was positioned on a moving cart. While lying on the cart the pitch and roll angles remain constant. Thus, we were able to evaluate the ProbIN approach by using a phone without a gyroscope.

In this paper, we are focusing on more realistic scenarios. By utilizing iPhone 4 equipped with a gyroscope and by introducing the integration of the gyroscope readings into ProbIN we were able to achieve promising results even when the phone was not mounted on users' body. In order to evaluate the new approach we conducted experiments with a user walking and carrying the phone in different settings such as in the hand or in the pocket.

2 Related Works

Self-contained systems are also called infrastructureless systems, since they can locate a mobile user without any external infrastructure. These systems can provide finer-granular position estimation than infrastructure based systems. Thus, they can be deployed in combination with infrastructure-based systems in order to achieve higher estimation accuracy.

Infrastructureless positioning is typically based on the dead reckoning principle that describes an iterative process of position estimation. According to this principle, a user's current position is determined based on the previous estimated position and the current sensor reading. The main drawback of the dead reckoning is the error accumulation. Since sensor readings are noisy by nature, they are the main source of the error in each positioning step. Due to the dead reckoning principle error generated in one step is carry out to the next step. Thus, the accumulated error increases with time and traveled distance.

Many research projects try to minimize the estimation error by employing high-quality sensors. For example, a ring laser gyroscope [6] or high-quality IMU [14] can be utilized in order to reduce the error in the measurement of rotation rate. However, these hardware components are usually very large and expensive. Therefore, they are not appropriate for daily use.

The noise can be also partially extracted by using a Kalman filter [18] which estimates true values of the observed sensor measurements based on the parameter settings.

In many research projects [6, 2] step-based techniques are applied in order to estimate user's position. The idea is based on counting the number of steps and estimating the length of each step from the accelerometer reading. The heading information is provided by fusion of magnetometer and gyroscope readings. The best approaches were able to achieve an accumulated error of about 2% of the total distance travelled [2]. However, the conducted experiment reveals many drawbacks of these techniques. As mentioned above the heading is derived from the magnetometer and gyroscope readings. However, this heading does not always correspond to the direction of the user's movement. Let us assume that the sensors are mounted on a helmet [2]. If the user walks straight forward and will look straightforward then the estimated trajectory will be correct. However, when the user starts looking around during the walk, the estimations will be incorrect, since the heading determines the estimated trajectory direction. Therefore, in order to achieve the good results the user has to look always into the direction of the walk. Also, it is very difficult to detect side steps, walking backwards or walking up/down the stairs.

Since the step-based techniques are not very practical due to the above mentioned drawbacks an alternative solution using the basic physics seems to be more promising [3, 14, 8]. Instead of utilizing the acceleration magnitude the actual values of the acceleration in all three x-, y- and z-axis can be used. In theory the integral of acceleration over time equals to velocity. When we calculate the integral of the velocity we obtain the displacement. By summing up the calculated displacements during the walk we can estimate a user's current position.

Due to the noisy sensor readings and the integration process, each estimation of the velocity and displacement will contain some error. Since the current velocity and position is always calculated based on the previous values, the accumulated error will grow significantly over time.

By mounting the sensors on a foot, the above-mentioned issue can be partially addressed. The idea is based on the analysis of the human locomotion. During one gait cycle, each foot will go through the two basic phases: swinging in the air and stance on the ground. In the stance phase, the foot velocity is zero. Also in this phase the most accurate orientation can be calculated from the sensor readings. Therefore, by identifying the stance phase based on the sensor readings the Zero Velocity Update (ZUPT) and Zero-Attitude Rate Updates (ZARUs) can be applied [1].

Several research projects [14] reported achieving an error rate lower than 1% by utilizing the physics-based techniques. However, these results seem to be rather an exception. The average reported error rate was around 3% to 5% of the travelled distance [14, 8].

Physics-based techniques benefits from the application of ZUPT and ZARU, which can significantly prevent the accumulation of the estimation error. These techniques also do not suffer from the drawbacks of the step-based techniques, since the direction of the movement is extracted not only from magnetometer and gyroscope but also from the accelerometer, side steps, walking backwards and walking up/down the stairs can all be detected. On the other hand, all sensors need to be mounted on the foot, which in many situations is not possible or practical. When we want to utilize the sensors in mobile device, we cannot assume that the users will mount their phone on the foot every time they use the indoor navigation system.

Besides mounting sensors on the foot, sensors can also be mounted to other parts of the body such as head, waist or shank. However, by selecting these mounting options the ZUPT and ZARU cannot be applied. Therefore, for these mounting settings only the step-based techniques can be utilized.

The more loosely mounting options was investigated in [15]. The sensors were carried in a backpack or mounted on a PDA that was hold in the hand. The experiments showed that the results of the loosely mounting options are significantly worse than the ones acquired by mounting the sensors on the person's body.

3 Probabilistic Infrastructureless Navigation

Probabilistic Infrastructureless Navigation (ProbIN) is a system providing the positioning and navigation functionality for GPS-challenging environment. In this paper, we will focus on the positioning part of ProbIN, which allows tracking of a person.

ProbIN is intended for daily applications such as assisting a user in shopping malls or at airports. Therefore, it needs to fulfill three essential requirements:

1. *Scalability*: ProbIN delivers positioning functionality even without an existing external infrastructure.
2. *Affordability*: ProbIN can be run on consumer mobile devices with relatively low-quality sensors.
3. *Usability*: The user should be able to hold the devices in the hand or in a pocket without degrading delivered functionality.

In order to fulfill the first requirement ProbIN essentially utilizes the inertial sensors. As such, a self-contained positioning system can be delivered, which does not rely on any external infrastructure. ProbIN can be also improved by utilizing a magnetometer and/or digital maps in order to achieve higher positioning accuracy. The advantage for this case is that the system will remain self-contained. A viable extension would be the integration of modules utilizing external infrastructures such as GPS or a Wi-Fi network. These modules would be activated only when an infrastructure is available. The scope of this paper is to develop a system providing the positioning based on inertial sensors and extended by the magnetometer readings.

Due to the second and the third requirements, a novel approach of positioning needs to developed in order to address high error rates of the sensors. It is known that when utilizing low-cost sensors the measurements are typically very inaccurate, especially when the sensors are not mounted to the user's body. In this case, the traditional physics-based positioning performs badly, since an error in the sensor reading causes an error in the estimated displacement. ProbIN addresses this issue by learning a mapping between the sensor readings and the actual true displacements based on training data. Thus, when ProbIN is deployed, even a noisy sensor reading can be mapped to the correct displacement. The problem of minimizing the error rate is thereby transformed into a machine learning problem. In this paper, we will present a solution to this problem by utilizing machine learning techniques well-known to the field of the statistical machine translation (SMT) [4].

As mentioned above, ProbIN utilizes a machine learning technique that is divided into training and testing phases. First, the sensor readings with corresponding true displacements are collected in the training phase. The relationship between the measurements and the displacements are used for creating a statistical model. Then in the testing phase, the statistical model is employed for mapping the sensor readings of the tracked person into a trajectory. The result of the testing phase is typically used for evaluating the approach.

In the following, we will present our approach in a reverse order. First, we will introduce how a user's trajectory is estimated based on the statistical model in order to give an intuition behind ProbIN. Then we will describe the process of training the statistical model in more detail.

4 Statistical Dead Reckoning of Mapping Sensor Readings to Displacements

ProbIN's positioning is a statistical dead reckoning approach allowing erroneous sensor readings to be mapped to correct displacements. This is achieved by framing the problem as a noisy-channel problem [16] where we try to recover the actual position of a user from the distorted sensor inputs. Noisy-channel models have been widely used in fields such as Statistical Machine Translation (SMT), Automatic Speech Recognition or Optical Character Recognition.

The noisy-channel describes the communication process in which a transmitter sends a message through a noisy channel and the receiver receives a corrupted or ambiguous message. The aim is to find the original message produced by the transmitter by analyzing the message observed at the receiver.

In SMT the transmitter produces a sentence in a familiar language, let us say English. On the other end of the channel the receiver observes a sentence in a foreign language, let us say German. The aim is to find an English translation of the German sentence. This issue is addressed by applying Bayes' theorem, which utilizes both the likelihood of translating from a foreign language word to a familiar language (*translation model*) and also a priori knowledge about the "properness" of sentences in the familiar language (*language model*). The translation model and the language model form together the *statistical model* used for finding the translation of the sentence observed by the receiver.

Similarly to SMT, ProbIN uses also a statistical model for finding the most likely trajectory for the given sequence of sensor readings. The observed sensor readings correspond to the German sentence in the above example. The aim is to find the hidden displacements of the trajectory, which correspond to the English sentence.

4.1 Quantizing Sensor Readings and Displacements

Unlike SMT where the vocabulary of a language is limited, the number of possible sensor measurements and displacements are unlimited. It is possible but too complicated to train a reliable continuous statistical model to map raw sensor readings into a displacement in the real value space. Therefore, in ProbIN we first quantize the sensor readings using the K-means clustering [11] which converts the real-valued sensor data to discrete values. Each cluster is then labeled with a *motion label* m, which will represent all sensor readings belonging to this cluster. Thus, the vocabulary size of ProbIN is limited to the number of motion labels. In that case, the statistical model can be efficiently trained for the discrete finite vocabulary space.

Figure 1 shows an example of employing the K-means clustering to quantize sensor readings into motion labels. In ProbIN each sensor reading corresponds to an acceleration and an orientation, which is derived from the gyroscope and magnetometer reading. After the clustering is processed, each sensor reading is represented by a motion label m. Thus, the sequence of t sensor readings collected during a user's movement is represented by a sequence of motion labels $M = m_1, m_2, \dots, m_t$ where m_i denotes a motion label of the sensor reading at timestamp i.

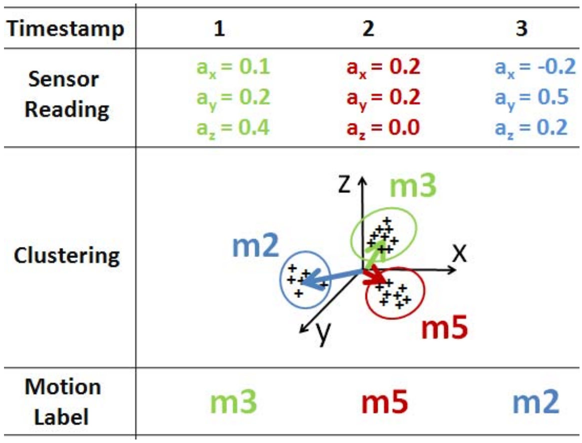


Fig. 1. Example of using k-means clustering in order to define the motion labels

The continuous space of displacements is also quantized in ProbIN. The *displacement label* d is assigned to each displacement cluster. Thus, a sequence of traveled displacements is represented by a sequence of displacement labels $D = d_1, d_2, \ldots, d_t$.

After limiting the size of vocabulary by quantizing the sensor reading space and displacement space, the statistical model can be similarly applied as in SMT. For a sequence of motion labels M, ProbIN searches for the optimal sequence of displacement labels D^* such that:

$$\begin{aligned} D^* &= \arg\max_D P(D|M) \\ &= \arg\max_D P(M|D) \cdot P(D) \end{aligned} \tag{1}$$

To express the above idea in terms of the noisy-channel model, the receiver observes a sequence of motion labels, which might be ambiguous or noisy. The aim is to find the translation in form of a sequence of displacement labels, which was originally produced by the transmitter.

Mathematically, $\arg\max P(D|M)$ returns a sequence of displacement labels D^* that maximizes the probability $P(D|M)$. The term $P(M|D)$ is provided by the translation model and $P(D)$ is provided by the trajectory model, which corresponds to the language model in SMT. The noise channel model can be confusing here as conceptually we are trying to "translate" from motion labels (sensor readings) to displacement labels to estimate a user's position. Yet by applying the Bayes rule (to incorporate the *a priori* knowledge of the trajectory model, the translation model uses conditional probability of "translating" from displacement to motion labels.

The statistical model of ProbIN consists of two parts: the translation model and the trajectory model.

4.2 Translation Model

The *translation model* estimates the likelihood of mapping a sequence of motion label M to a sequence of displacement label D. Assuming the translation of a displacement label to its corresponding motion label is independent of other pairs, we can write:

$$P(M|D) = \prod_{i=1}^{t} P(m_i|d_i) \tag{2}$$

$P(m_i|d_i)$ is the likelihood of "translating" a displacement d_i back to the motion label m_i. This value is extracted from the training data during the training phase.

4.3 Trajectory Model

The *trajectory model* in ProbIN works similarly to the language model used in SMT. A language model estimates how likely a sequence of words is a meaningful English sentence. For example, the sentence "Tomorrow I will go shopping" should have higher language model probability than sentence "Morning I will go shopping", since the former is more likely to be a correct English sentence.

The intuition of utilizing the trajectory model in ProbIN is also based on the idea that not all trajectories are meaningful. It is obvious that some trajectories are physically impossible to achieve. For example, a trajectory with a length of 10 meters after 1 second collected from a person walking with constant speed is somehow suspicious. Moreover, when a user is walking forward, his trajectory is most likely to be a sequence of forward moving displacements rather than a sequence of forward-backward-forward-backward. The information about "meaningfulness" of the trajectories can be extracted from the training data in advance. The probability of a trajectory D is calculated as:

$$\begin{aligned} P(D) &= \prod_{i=1}^{t} P(d_i|d_1^{i-1}) \\ &= P(d_1) \cdot P(d_2|d_1) \cdot \ldots \cdot P(d_i|d_1, \ldots, d_{i-1}) \\ &\quad \cdot \ldots \cdot P(d_t|d_1, d_2, \ldots, d_{t-1}) \end{aligned} \tag{3}$$

Under the Markov assumption that a displacement label d_i only depends on the immediate $n-1$ displacement labels in history, $P(d_i|d_1, \ldots, d_{i-1})$ can be estimated as $P(d_i|d_{i-n+1}, \ldots, d_{i-1})$. This is equivalent to the n-gram language model approached use in SMT.

4.4 Decoder

Given the input sensor readings, which are now quantized as motion labels, a so-called *decoder* applies the translation and trajectory models to search for an optimal displacement label sequence. In our work the sensor reading sequence is called *source sentence* M. First, the translation model generates different translation options D for the whole source sentence M. These translation options are called *hypotheses* as in SMT. Second, for each hypothesis the values of $P(M|D)$ and $P(D)$ are calculated based on the information from translation and trajectory models. The hypothesis with the highest probability $P(M|D) \cdot P(D)$ is selected as the optimal translation for M.

Decoding Process Figure 2 illustrates an example of the decoding process. At each timestamp, one new sensor reading is collected. The decoder generates different hypotheses for the sensor readings observed up to this point. A hypothesis is represented as a path from the root the leave of the updated *decoder tree*. To avoid the explosion of hypothesis due to combinations of different translations, we prune out hypotheses with lower probabilities. Pruning is likely to terminate some hypotheses prematurely as it is based on the incomplete information during decoding. Practice in SMT has shown that pruning is necessary and the degradation on performance is acceptable. By the end of the decoding, the optimal hypothesis with the highest probability is output as the user's estimated trajectory (shown as the red path in the Figure 2).

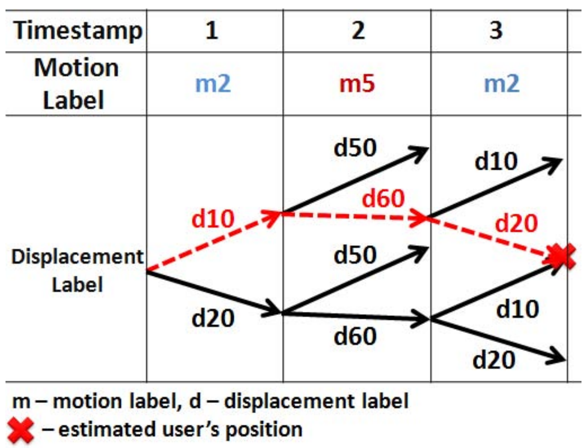


Fig. 2. Example of generated hypotheses. The red path corresponds to the hypothesis being the most probable translation for the given source sentence.

Process Optimizations As it can be observed from the above example, the complete hypothesis space is exponential to input length N and the number of translation alternatives. If each motion label can be mapped to r different displacement labels, the total number of possible trajectories is r^N for an input sequence of N sensor readings. Let us assume that each word in the sentence "Morgen gehe ich einkaufen" would have three possible translations. Thus, there are 81 ($= 3^4$) possible hypotheses such as "Tomorrow I will go shopping", "Morning I will go shopping", "Tomorrow I go buy", "Morning I go buy", etc. In a real natural language, each word can have 10 translations (including different variations). A typical sentence size is around 10 to 15 words. In that case, the number of hypotheses can grow to 10^{15}. Therefore, it is computationally infeasible to search the complete hypothesis space for the global optimal trajectory, especially in case of high N.

On the other hand, a greedy search of keeping only one best hypothesis at time t is likely to end up in a local optimum. In order to approximate the globally optimal translation, we use a multi-stack based decoding technique [10]. The multiple stack data structure stores multiple partial hypotheses that have the highest probabilities up to time t. Thus, not the whole hypotheses space needs to be explored. On the other hand, the number of analyzed hypothesis is sufficient high for finding a translation close to the globally optimum.

The above-described optimization causes that the hypotheses stored in the stack tend to be similar to each other. Therefore, the decoder applies additionally the *hypotheses recombination* technique to merge hypotheses that have the same trajectory model endings and thus cannot be distinguished by future hypothesis expansion. For those partial hypotheses that have the same trajectory model endings, only the one with highest probability will survive [17].

5 Training the Statistical Model

In the previous section we described how the statistical model of ProbIN could be used for estimating a user's current position. This model is created in advance based on the training data set. We assume that the model can be trained based on the user's past trajectory data, in order to deliver the high accurate position estimation. Thus, each user would train his or her own model, which captures the motion patterns of the specific person. However, the initial model can be also created from the trajectory data of different persons in order to capture characteristics common for all users. Once the initial model is created, a user would be able to use the system for the positioning purposes. While the system is used, the data from a specific user can be collected in order to adapt the model to the particular person. In the following we will present the approach of creating the model based on an available training data set.

5.1 Training the Translation and Trajectory Model

In order to extract the translation model probability $P(M|D)$ we need a training data set where correspondences between motion labels and their displacement label are known. To estimate the trajectory model probability $P(D)$, we need a training data with known trajectories of a user in the form of sequences of displacement labels.

Table 1 shows an example of the training data. The training data set should contain sequences of (motion label, displacement label)-pairs. These pairs provide information about the correspondences between motion labels and their reference displacement label. To train the trajectory model, we use just the displacement part from the training data, for example, sequence (d54, d67, d45, d12, . . .).

Table 1. Example of sequences of training data,which consist of (motion label, displacement label)-pairs

#	Sequence of training data
1	(m3, d54), (m2, d67) (m4, d45), (m7, d12) ...
2	(m15, d13), (m6, d45) (m10, d43), (m30, d11) ...
	⋮

Based on the frequency of (motion label, displacement label)-pairs, the translation probability can be estimated through the Maximum Likelihood Estimation (MLE) [5]:

$$P_{MLE}(m|d) = \frac{count(m,d)}{count(d)}. \tag{4}$$

where count(x) returns a number of occurrences of the given x.

The maximum likelihood estimation of the n-gram trajectory model is based on the displacement label only:

$$P_{MLE}(d_i|d_{i-n+1}, d_{i-n+2}, \ldots, d_{i-1}) = \frac{count(d_{i-n+1}, d_{i-n+2}, \ldots, d_{i-1}, d_i)}{count(d_{i-n+1}, d_{i-n+2}, \ldots, d_{i-1})}$$

5.2 Expectation Maximization Algorithm

Motivation In practice, obtaining data in the format illustrated in Table 1 is infeasible. Getting the accurate displacement for each motion label would require for example high speed motion capturing devices. These devices would capture very fine-granular movements and transform them into reference displacements. Since the sampling rate of inertial sensors in mobile phones is 50-100 Hz, the motion capture device needs to record the motions at similar high sampling rate.

Since our work is intended for daily applications, we can not assume users have access to such devices to collect data for training. However, it is reasonable to assume that we can obtain less detailed reference displacement information either from GPS (Table 2) during outdoor training sessions or from user's input overlaid on floor plans. ProbIN uses this coarse-grained data to "estimate" the reference displacement for each sensor reading.

In Table 2 the data is collected during several walks. Besides the sequence of motion labels, each walk also contains information of the GPS coordinates of the starting and the ending positions for the whole walk. However, the reference displacements for each motion label cannot be collected (Figure 3). In ProbIN we apply the Expectation Maximization (EM) to estimate this reference displacement information.

Table 2. Information available for training the models

Walks	Sequence of motion labels	Start	End
Walk 1	m3, m2, m4, m7, ...	(0,0)	(4,10)
Walk 2	m15, m6, m10, m30, ...	(4,10)	(7,15)
	⋮		

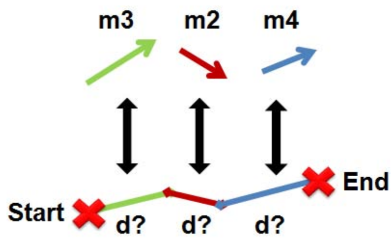


Fig. 3. Missing displacement information

EM Algorithm Expectation Maximization (EM) algorithm [7] is an iterative method for finding hidden parameters of a statistical model. In ProbIN the displacement label is considered as the hidden information. Therefore, the EM algorithm is applied in order to iteratively estimate the hidden displacements for sequences of motion labels. Each EM iteration consists of two steps: Estimation and Maximization. The goal of the *estimation* step is to estimate the hidden displacement using the current statistical model. Based on the estimation the *maximization* step updates the statistical model. In the first iteration the statistical model has not been created yet. Therefore, for the estimation we use the traditional inertial positioning technique based on the laws of physics to bootstrap the model (Figure 4).

The training input contains K instances of "walks":$W_1, W_2, \ldots, W_K$ shown in the Table 2. Each walk W_k has a sequence of motion label M_k, the reference starting position s_k and the reference ending position e_k. In the estimation step of each iteration, we use the current model to find the estimated hidden trajectory D_k for each walk W_k.

In the first iteration the trajectory is estimated by using a traditional physics-based positioning technique. First, the orientation of the phone is calculated by fusing the inertial measurements with magnetic sensor readings through a Kalman filter. This orientation is then used for rotating the raw acceleration readings from the device frame into the global frame. By applying double integral on acceleration readings in the global frame we obtain a user's estimated trajectory.

Since the phone is not mounted on a foot, the Zero Velocity Update (ZUPT) cannot be applied. This is the main concern for the traditional physics-based positioning approaches. Without being able to reset the velocity after each step

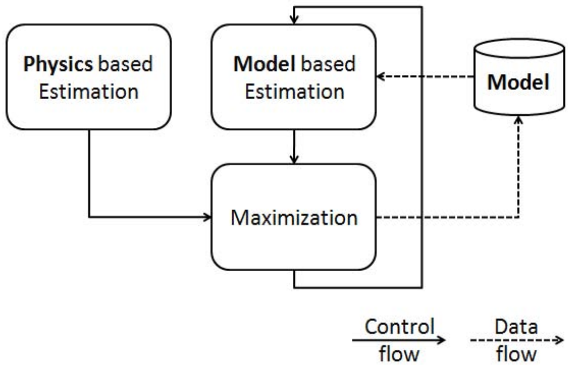


Fig. 4. The EM Algorithm starts with trajectory estimation based on the basic physics and in each iteration updates the statistical model

the error accumulates rapidly over the time. In ProbIN we address this issue by using only the walks with short distances and short time durations in the training process. Thus, the error rate of the trajectory estimates for these walks are minimal.

In the next EM iterations, the physics-based positioning is replaced by the statistical model created at the first iteration. Thus, the statistical model is used to decode the motion label sequence M_k to get the estimated trajectory D_k^*.

Most likely D_k^* is incorrect and does not end at e_k. We assume that the correct trajectory has the similar "shape" as D_k^*. Thus, we can stretch and rotate D_k^* such that it ends at e_k as shown in Figure 5. The resulting trajectory D_k' is the newly estimated trajectory for M_k in this iteration. In other words when D_k does not end up at e_k, we know this trajectory is incorrect. Since we do not know which segments caused the error, we distribute the error to each of the t segments in D_k through the stretch and rotation operations.

The stretching factor is calculated as:

$$f_{stretch} = \frac{distance(s_k, e_k)}{distance(s_k, e_k^*)} \tag{5}$$

where s_k is the starting position of walk W_k and e_k is the ending position. e_k^* is the ending position based on the estimated trajectory D_k^*. $distance(x, y)$ returns a Euclidean distance between two points.

The rotation angle is calculated as:

$$f_{rotate} = cos^{-1}(\frac{e_k^* \circ e_k}{\|e_k^*\| \, \|e_k\|}) \tag{6}$$

where $\circ$ represents scalar product and $\|x\|$ represents Euclidean norm.

In order to correct the estimated trajectory we correct each d_t by stretching it with the $f_{stretch}$ and rotating it by angle f_{rotate}. The trajectory calculated from the corrected d_t^{cor} should end in the position e_k.

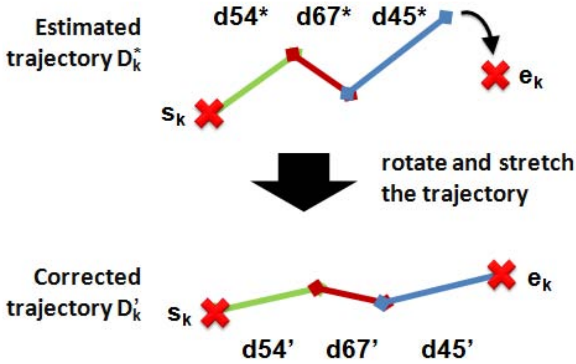


Fig. 5. Correcting the estimated trajectory

In the maximization step the translation and trajectory models are updated based on the corrected estimation of the underlining trajectories D'_k. Thus, the probability of the entire training data can be maximized. The pseudo code of the EM algorithm is described in Algorithm 1.

6 Experiments

Our approach is evaluated on the iPhone 4, the first phone on the market equipped with an additional gyroscope besides an accelerometer and a magnetometer. The gyroscope provides information about the rotation rate of the phone. This information allows our system to accurately calculate the phone's orientation even when the phone is in motion. This is necessary for the process of position estimation in ProbIN and also in any state-of-the-art physics-based positioning system.

In order to evaluate ProbIN approach we conducted experiments on a square-shaped basketball court. For training the statistical model we used sensor readings of straight walks with a length of 4 meters as shown in Figure 6(a). For testing the statistical model we use two types of trajectories. The first type has an L-shape (shown in Figure 6(b)) and the second type has a square-shape (shown in Figure 6(c)). Since each side of the basketball court was 15 meters long, the walks used for testing were 30 meters resp. 60 meters long.

We conducted experiments to evaluate whether the accuracy of the estimations are dependent on the way the user holds the phone. We collect sensor data while the phone is in one of three different settings. First, the user holds the phone in the hand in front of him as if he would be looking to the display. Second, the user is holding the phone loosely in the hand, but he is not looking at the display during the walk. Finally, the phone is carried in the pocket.

Data: Training data: a set of K *walks*: $W_1, W_2, \ldots, W_K$. $W_k = (M_k, s_k, e_k)$
iteration $\leftarrow 0$;
while *not converge* **do**
  for $k \leftarrow 1$ **to** K **do**
    (Estimation step for walk W_k) ;
    if *iteration==0* **then**
      Use basic physics (double integral of acceleration) to estimate D_k^*
    end
    else
      Find D_k^* for M_k given the current statistical models
    end
    if D_k^* *does not end at* e_k **then**
      stretch and *rotate* D_k^* to create D_k' such that D_k' ends in e_k
    end
    else
      $D_k' \leftarrow D_k^*$
    end
    foreach (m, d) *pairs in* (M_k, D_k') **do** increase the count of (m, d) ;
    foreach *n-gram in* D_k' **do** increase the count of the trajectory n-gram ;
  end
  (Maximization step);
  Update the translation model from counts of (m, d);
  Update the trajectory model from counts of trajectory n-grams;
  iteration++;
end

Algorithm 1: EM algorithm used to training the statistical model.

For evaluating our approach we used the relative error rate as the evaluation metric. The relative error of an estimation is calculated for 2 dimensional trajectory, i.e., the altitude is omitted in our experiment:

$$error_{rel} = \frac{error_{abs}}{trajectory_{ref}} \tag{7}$$

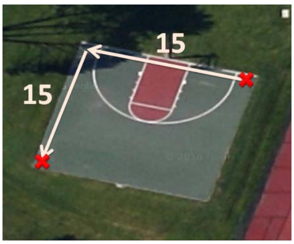
The absolute error $error_{abs}$ corresponds to the distance between the user's position estimated by a positioning system e^* and the user's true position e:

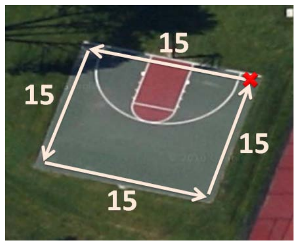
$$error_{abs} = distance(e^*, e) \tag{8}$$

The $trajectory_{ref}$ is length of the reference trajectory. Thus, the relative error grows with the increasing absolute error. However, the relative error is additionally normalized by the length of the reference trajectory in order to be able to compare the results of different experiments with different length of walks.

In the first experiment we focus on the cases when the user holds the phone in the hand in front of him or her as if she would be looking to the display. The results (shown in Table 3) are grouped based on the type of the test data: L-shape and Square-shape. We also evaluate the estimation accuracy for different speed of the walks: slow, normal and fast. In our experiments we use the traditional

(a)





(b) (c)

Fig. 6. Basketball court used for the experiments. The figures show the walks used for training the model (a) and evaluating ProbIN approach (b), (c)

(a) (b)

Fig. 7. The phone can be carried in the hand or in a pocket. While carrying the phone in the hand the user can be occasionally looking at the display.

physics-based positioning system (described in Section 5.2) as a baseline. The error rate of this system is displayed in the column *Physics*. The following columns contain results of the estimations using ProbIN's statistical model generated in the EM iterations 1, 2 and 3.

For training ProbIN's statistical model we used sensor data from 60 straight walks. Each walk was 4-meters long. For testing of each configuration (e.g., square-shape fast walk) we used 5 long walks and calculate an average error rate over 5 estimated trajectories.

Table 3. Average error rate of estimated position of the physics-based approach vs. ProbIN using models trained after EM iteration 1, 2 and 3. The phone is carried in the hand in front the body.

Type	Speed	Physics	EM1	EM2	EM3
L	slow	348.7%	18.3%	17.2%	16.3%
L	normal	194.6%	23.0%	22.1%	22.3%
L	fast	118.2%	26.7%	24.1%	25.6%
S	slow	2158.3%	4.0%	3.9%	4.4%
S	normal	800.3%	3.6%	3.6%	4.4%
S	fast	463.9%	1.8%	1.9%	2.0%

As the results show the error rate of the physics-based positioning approach is significantly higher than the error rate of ProbIN's approach. The reason is that the mobile phone is held in the hand and not mounted on the foot. Therefore, ZUPT cannot be applied. Thus, errors in the physics-based approach rapidly accumulate over time whereas ProbIN bypasses the step of double integrating the acceleration and is, therefore, more robust against sensor noise.

Similar to the findings reported by [9], the main source of error in ProbIN comes from the incorrect heading information. The low-cost sensors from the mobile phone are less stable and they are more likely to be interfered by local magnetic fields. Therefore, the magnetometer readings are easily biased from their true headings. In our experiments, the basketball court seems to have a stable local magnetic field which gives the magnetometer a constant bias. For example, reporting 12 degree north-east while the phone is heading north and always have +12 degrees to the true headings later on. This explains why the error rates of the square-shaped walks are lower than the error rates of L-shaped walks. In L-shaped walks a bias in the heading causes that the distance between the user's estimated ending point and the user's reference ending point increases. Thus, the error rate for L-shaped walks also increases. In square-shaped walks, the user travels in a close loop round trip which results in canceling out the error caused by the magnetometer bias. This finding applies to all round trip trajectories which are usually used for evaluating positioning systems as reported in the literature. Based on our analysis, we strongly argue against evaluating a self-contained positioning system only with round trip trajectories as this design might be blind of heading errors and can be misleading.

Table 4. The phone is carried loosely in the hand and in the pocket

Type	Position	Physics	EM1	EM2	EM3
L	hand	4435.3%	25.1%	25.4%	34.9%
L	pocket	2900.0%	30.6%	25.2%	21.9%
S	hand	19993.5%	16.5%	34.4%	25.5%
S	pocket	13435.8%	8.1%	9.4%	13.4%

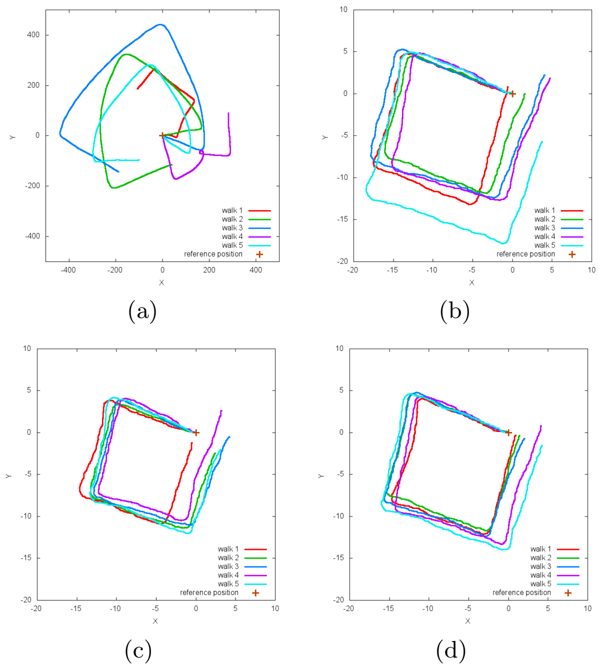


Fig. 8. The trajectories calculated based on the physics-based positioning (a) and on ProbIN statistical model in EM iteration 1 (b), 2 (c) and 3 (d)

Figure 8(a) shows trajectories estimated by the physics-based approach on square-shaped walks in normal speed. It differs notably from the true trajectory shown in Figure 6(c). On the other hand, the trajectories in Figures 8(b) - 8(d) confirmed our expectation that the error rate can be significantly reduced by utilizing ProbIN statistical model.

In the second experiment we explore the remaining two settings of the phone (Table 4). First, the user holds the phone loosely in the hand and is not looking at the display. Second, the phone resides in the pocket. In both cases the sensors of the phone sense the oscillating motions during a walk as shown in Figure 7(b). This motion generates additional noise which causes the increase of the estimation error.

7 Conclusion

In this paper we introduce ProbIN, a novel statistical dead-reckoning approach of mapping sensor readings to user's position for indoor positioning applications. Our approach relies solely on the sensors provided by the consumer mobile devices. Thus, our positioning system can be deployed in any environment. Furthermore, ProbIN adapts its statistical models to the characteristics of the sensor and the user's walking patterns. Thus, even with the noisy sensor readings the estimated position can be relatively accurate. In addition, ProbIN does not require the phone to be mounted to the body. Users can just hold the phone in the hand or put it in the pocket which is more casual for real life applications.

For future work, we plan to conduct experiments with different type of movements (e.g., running, jumping, crawling, etc.) and with longer distances. This will allow the usage of the system in more challenging scenarios such as running in a forest or crawling in a tunnel. Moreover, the calculated error will also include the altitude in order to optimize the system for trajectories taking place on different floors.

Additionally, our system will be extended by utilizing external infrastructures such as GPS or Wi-Fi when they available. The existing indoor maps can be used for implementing the point-to-point navigation will be implemented. Moreover, the estimated trajectories can be corrected by employing state-of-the-art map-matching techniques.

Acknowledgements. This research was supported by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273 from the Army Research Office.

References

[1] Beauregard, S.: Infrastructureless pedestrian positioning. PhD thesis, University of Bremen (2009)
[2] Beauregard, S., Haas, H.: Pedestrian dead reckoning: A basis for personal positioning. In: Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC 2006), vol. 06, pp. 27–35 (2006)
[3] Beauregard, S.: Omnidirectional pedestrian navigation for first responders. In: Workshop on Positioning, Navigation and Communication, pp. 33–36. IEEE (March 2007)
[4] Brown, P.F., Cocke, J., Pietra, S.A.D., Pietra, V.J.D., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics 16(2), 79–85 (1990)
[5] Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19(2), 263–311 (1993)
[6] Collin, J., Mezentsev, O., Lachapelle, G.: Indoor positioning system using accelerometry and high accuracy heading sensors. In: Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS/GNSS 2003, pp. 1–7 (2003)
[7] Dempster, A.P., Laird, N.M., Rubin, D.B., et al.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
[8] Feliz, R., Zalama, E., García-Bermejo, J.G.: Pedestrian tracking using inertial sensors. Journal of Physical Agents 3(1), 35–42 (2009)
[9] Fischer, C., Muthukrishnan, K., Hazas, M., Gellersen, H.: Ultrasound-aided pedestrian dead reckoning for indoor navigation. In: Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT 2008), vol. 31 (2008)

[10] Germann, U., Jahr, M., Knight, K., Marcu, D., Yamada, K.: Fast decoding and optimal decoding for machine translation. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, vol. 39, pp. 228–235. Association for Computational Linguistics (2001)

[11] Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis, 9th edn. WileyBlackwell (March 2005)

[12] Lin, H., Zhang, Y., Griss, M., Landa, I.: WASP: an enhanced indoor locationing algorithm for a congested Wi-Fi environment. In: Proceedings of the Workshop on Mobile Entity Localization and Tracking in GPS-less Environnments, pp. 183–196 (2009)

[13] Nguyen, T.-L., Zhang, Y., Griss, M.: Probin: Probabilistic inertial navigation. In: 2010 IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS), pp. 650 –657 (November 2010)

[14] Ojeda, L., Borenstein, J.: Non-GPS navigation for security personnel and first responders. The Journal of Navigation 60(3), 391–407 (2007)

[15] Randell, C., Djiallis, C., Muller, H.: Personal position measurement using dead reckoning. In: Proceedings of Seventh IEEE International Symposium on Wearable Computers, pp. 166–173 (2003)

[16] Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review 5(1), 55 (2001)

[17] Vogel, S., Zhang, Y., Huang, F., Tribble, A., Venugopal, A., Zhao, B., Waibel, A.: The CMU statistical machine translation system. In: Proceedings of MT Summit, vol. 9 (2003)

[18] Welch, G., Bishop, G.: An introduction to the Kalman filter, pp. 1–16. University of North Carolina at Chapel Hill, Chapel (1995)