Simple approach for indoor mapping using low-cost accelerometer and gyroscope sensors *

G. Trein
University of Toronto
Toronto, Canada
trein@cs.toronto.edu

N. Singh
University of Toronto
Toronto, Canada
navjot@cs.toronto.edu

P. Maddila
University of Toronto
Toronto, Canada
priyamsa@cs.toronto.edu

ABSTRACT

Open map systems are viable today due to the rise of crowdsourcing. Open map systems combine two major sources of data: geographic information system (GIS) databases and motion trajectories contributed by the crowd. Almost any person, powered by his/her mobile phone, can record his/her own trajectory and send it in to help an open map system. However, this approach does not work when the GPS signal is poor or not available, for example, within buildings. The objective of this project is to explore ways to crowdsource indoor maps without relying on GPS. Instead, we address this problem by using the mobile device's sensor data and evaluate its effectiveness considering aspects such as accuracy and complexity. Our solution involves two main phases: detection of the number of steps performed by the user and their direction. Regardless the mobile phone position or orientation, our step detection algorithm is able to detect steps with an accuracy of 98.95%. Additionally, our heading compensating algorithm was able to reduce the heading error from 20.61% to 5.61%, making it usable in an inertial navigation system. Finally, we demonstrate that crowdsourced inertial information is able to provide a relatively accurate mapping of an indoor environment, differentiating free from blocked space.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Experimentation, Measurement, Performance

Keywords

Indoor mapping, mobile sensors, accelerometer, compass, gyroscope, crowdsourcing

*A full version of this paper and related source code is available at `http://www.cs.toronto.edu/~trein/csc2228/index.html`

1. INTRODUCTION

Global Navigation Satellite Systems (GNSSs) have truly revolutionized the way people move around, enabling outdoor navigation. Similarly, indoor navigation have many potential applications which remain underexploited: indoor navigation systems can be used to assist a visually challenged person by differentiating the free space and blocked space, they can provide navigation inside huge structures (e. g. airports and industries) or even help a user to locate his favorite products in a shopping mall.

Unavailability of GPS signal in indoor spaces makes it to realize the potential applications of indoor localization. To make them possible, there are two major existing approaches namely indoor localization techniques with fingerprinting and indoor localization techniques without fingerprinting. Concerning fingerprint-based techniques, the most challenging problem is to generate discriminative signatures. One of the most popular approaches is to make use of WiFi signals in the form of Received Signal Strength Indicator (RSSI) values [2] as WiFi access points are widely available indoors and almost all smartphones have WiFi receivers. The limitations of WiFi RSSI fingerprinting is that they are susceptible to human intervention and may cause localization errors because of RSSI values' high variation over time. Both indoor localization techniques with fingerprinting [2] and without fingerprinting [7, 4, 3, 6] do not use crowdsourcing and hence they track only the path of a single user that does not provide any useful information about the free or blocked space inside a place.

We address some of these problems by adopting a crowdsourced approach without any sort of fingerprinting. Crowdsourcing is a distributed problem-solving model in which a group of people is engaged to solve a complex problem through an open call. Our approach uses crowdsourcing to generate indoor maps by collecting data from the 3-axis accelerometer and digital compass sensors available in the majority of mobile devices. It consumes the continuously changing accelerometer and digital compass data as a user of a mobile device walks around an indoor space with his device. This data is then aggregated on a Web Service at a frequency of 20Hz and is used to calculate the coordinates of the user. The coordinates are further mapped to the available space in the room. This approach is crowdsourced by gathering enough data from multiple users to accurately project the available and blocked space inside a room on a map.

The rest of the paper is organized as follows; Section 2 describes the related work; Section 3 presents an overview of architecture implemented; Section 4 explains the algorithms for step detection and mapping of trajectories; Section 5 summarizes and discusses the results; Section 6 highlights some possible improvements and suggests next steps to the proposed solution.

2. RELATED WORK

Previous approaches to indoor mapping fall into two categories i) fingerprinting-based techniques and ii) step detection-based techniques. The approaches that use fingerprinting, such as [2], include WiFi RSSI and FM broadcast radio signal fingerprinting to map indoor environments. As WiFi access points are widely available indoors and all the smartphones have WiFi and radio signal receivers, these signals are collected in the form of Received Signal Strength Indicator (RSSI) values to map indoor locations. Major limitations of WiFi RSSI fingerprinting include susceptibility, multi-path, fading and temporal variation. The techniques that do not involve fingerprinting adopt dead reckoning to compute the current user coordinates based on previously known values. In FootPath [4], for instance, based on the accelerometer and gyroscope data, the authors employ dead reckoning to calculate user coordinates. There is another approach which presents indoor positioning based on Global Navigation Satellite Systems (GNSSs) combined with Pedestrian Dead Reckoning in [3]. The indoor positioning system in [3] uses activity classification to classify invariants in user locomotion such as walking and running. Similar to our approach, it also utilizes sliding window to extract relevant features from accelerometer sensor data. Moreover, an approach which explains indoor localization with dead reckoning, 2D barcodes and Sensor data is adopted in [6]. Unlike our approach, it already has a map and its main goal is to track the user location. Dead reckoning is also used in [7] which tracks only a single user trajectory.

Our work differs from the above approaches in the following ways. First, none of the above approaches employ crowdsourcing of sensor data to generate indoor maps. We crowdsourced sensor data in order to map the available free space in room. Second, all the above approaches concentrate on the accurate position of the user. We are interested in the higher level goal of using as much information we can to provide an accurate map of an indoor location. Third, our approach doesn't use any sort of signal fingerprinting, it is purely based on the users mobile phone sensor information.

3. ARCHITECTURE OVERVIEW

The proposed mapping system is composed of two main modules: mobile clients and a central server. In this section, we describe briefly the overall organization of the implemented system as well as try to clarify some choices and assumptions made.

3.1 Web Service

The application's back-end simplifies the development of new mobile clients by concentrating all the mapping processing and computation. As it encapsulates the business logic to perform indoor mapping, the complexity of the mobile client's applications is reduced, becoming a simple active agent which sends raw sensors' information. The main advantage of this approach is scalability. Although in this work we focused on iOS platform, this architecture enables us to quickly extend the support to other platforms as Android, Windows Phone or Blackberry.

We chose Java as the language for the back-end implementation because its J2EE stack, which provides an extensive support for intersystem communication (Web Services), asynchronous processing and object relational mapping. Indeed, we use the implementation of several Java Specification Requests (JSR) to speed up the development and deliver a robust solution for indoor mapping processing.

Briefly describing the components: our Web Service endpoint is implemented as a JAX-RS endpoint using RESTEasy framework. This interface was carefully engineered as it is shared across all mobile clients and is the point-of-contact between the systems. The Web Service provides mainly two services: it is the indoor mapping data entry point and it can be consulted in order to obtain mapping information of a given environment. After a request containing mobile client sensor data is received, it is validated and delegated to the mapping algorithm, which will be described in the next sections. The algorithm will output the processed coordinates of the user and then delegate to the repository component, which will perform the persistence of the data in the database. The Figure 1 illustrates the communication between the main components.

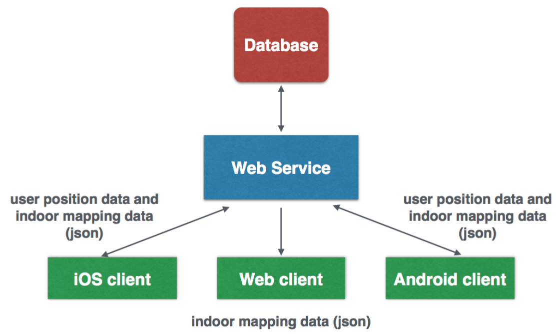

Figure 1: System's overall architecture representation.

3.2 Mobile Client

We use the combination of several inertial sensors in order to provide enough information for indoor mapping estimation. A 3-axis accelerometer is used to detect user's steps through acceleration variations in x, y, z axis. The orientation or heading of the phone is determined by using a magnetometer sensor along with gyroscope sensor.

We consider the mobile client as a simple sensor hub, which collects several parameters from its sensors and send them over the network to the Web Service. A mapping session starts when the user inputs some basic information, such as mapped location identification and his height, and initializes the data acquisition process. The location identification is a unique identification of an indoor location and it allows us to aggregate information about a given space from several different users. The height information is needed in order to estimate the length of each step taken by the user (as will be explained in section 4). The sensor data is acquired with a sampling rate of 20Hz and stored in a local database during the duration of the session. As soon as the mapping session is finished, an asynchronous task retrieves all the stored data from the database, converts it in a JSON format and transfers through the network to the Web Service. In our work, we developed an iOS and Android client, but our evaluations focus on the iOS platform for sensor accuracy reasons.

3.2.1 Heading calibrations and compensation

The majority of mobile SDKs use only the magnetic heading of the compass, which generates sufficiently accurate values for general purpose applications. However, if the compass values are meant be used in inertial navigation applications, often this accuracy is not enough and other approaches have to be adopted in order to compensate drifts caused by electromagnetic perturbations.

In our work, to provide an accurate value for the user heading, we combined the magnetic heading values with gyroscope sensor values. At each change of the magnetic heading value, we acquire the current gyroscope value and compare it with previously stored values. If the gyroscope values also confirm an angle variation, the magnetic heading value is compensated and promoted as a correct measurement, otherwise, it its simply ignored and the last value is maintained.

4. ALGORITHM

As we can see from the literature, step-based Pedestrian Dead Reckoning (PDR) algorithms consist of three phases: step detection, step length estimation and step heading estimation. Each time a step is detected, its estimated length and heading are used to update the position of the tracked user. To be able to perform an indoor mapping estimation, we append to the algorithm two additional phases: data filtering and data conversion. Moreover, in order to simplify the overall algorithm, we fix step length at a value that is proportional to each user's height. This approach will give a reasonable level of performance when the user is walking at a constant speed, but will perform poorly otherwise. This is because step lengths for a given person can vary by up to ±50% depending on walking speed, with a typical person taking smaller steps when walking slowly compared to when they walk at a brisk pace [5].

4.1 Data Filtering

The raw data sent by the sensors is extremely noisy and cannot be used directly in all phases of our algorithm. Low-pass filters provide a smoother form of a signal, removing the short-term fluctuations by attenuating signal components with frequencies higher than the cutoff frequency. The actual amount of attenuation for each frequency varies depending on specific filter design. Low-pass filters exist in many different forms and, in our approach, we use a first-order discrete version of a standard low-pass filter. This form of filter can be easily implemented in any programing language given the simplicity of its recurrence equation (Equation 1).

$$\begin{aligned} \alpha &= \frac{dt}{dt + RC} \\ y_t &= y_{t-1} + \alpha(x_t - y_{t-1}) \end{aligned} \tag{1}$$

In Equation 1, dt is the sensor sampling period and RC the time constant (or also known as $1/RC$ the cutoff frequency). As the mobile sensor sampling rate was set to 20Hz and since we would like to penalize variations greater then 15Hz, we set $dt = 1/20s$ and $RC = 1/15s$. The Figure 2 shows a comparison between the raw and filtered signal obtained after the implemented low-pass filter.

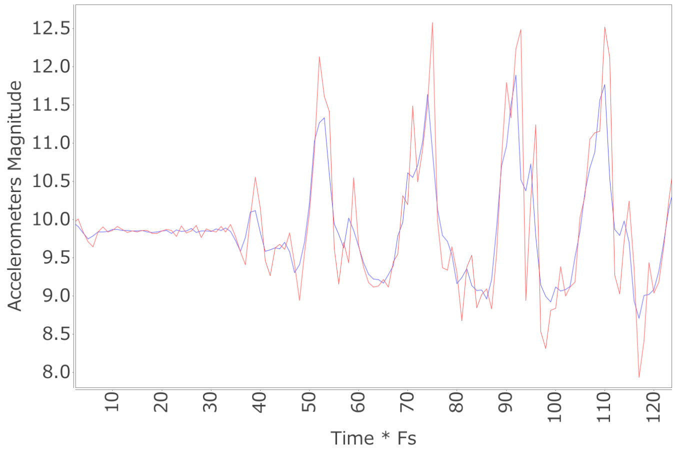

Figure 2: Raw (red) and filtered (blue) signal by low-pass filter. The y-axis represent the 3-axis accelerometers magnitude values combined and the x-axis the sample number. We can notice the filtered signal (blue) is smoother and is less noisy than the raw signal (red).

4.2 Step Detection Algorithm

This algorithm is applied on the accelerometer magnitude values to detect a user step. It is governed by the fact that a sharp peak in accelerometer values is observed as the user takes a step. We use an approach similar to [4] for step detection. We use a sliding window of 13 readings over accelerometer data in this approach. The number of readings to be used in a sliding window is experimentally determined. This value is critical for accurate detection of steps. If the window is too wide, the number of steps determined would be less than the actual number of steps. On the other hand if the window is too narrow, noisy data can be wrongly detected as a step. This results in number of steps detected to be more than the actual number of steps taken by the user. Each window is analyzed for the occurrence of a step. A window signifies a step if the following conditions are satisfied.

1. Standard deviation of the window is greater than the threshold value σ.
2. The median of window has the greatest magnitude.
3. Two steps are separated by certain threshold duration in time t_s.

The raw accelerometer data is noisy and cannot be used as such for step detection. A filter is applied on the accelerometer values to clearly observe the variance. The Figure 3 shows the graph plotted from the filtered accelerometer values and the main parameters evaluated in the algorithm. The peaks observed in this graph correspond to a step taken by the user.

4.3 Conversion Algorithm

A simple approach, as depicted in Figure 4, can be adopted to determine user position in a room in terms of (x, y) coordinates.

We assume the initial position of the user to be a fixed location inside the room defined as origin. As the user moves

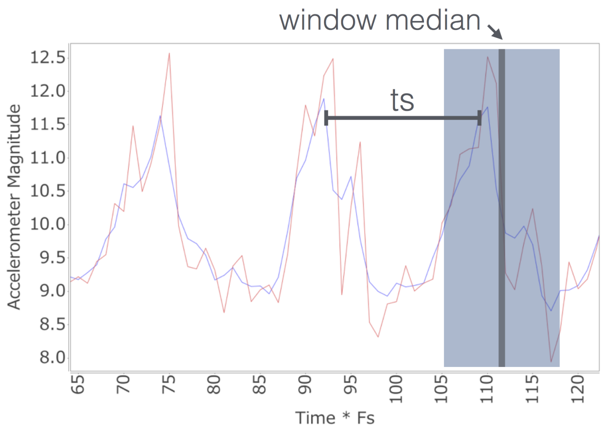

Figure 3: Relevant parameters for step detection algorithm.

along a specific direction, we determine the distance travelled by the user by counting the number of steps. A step is detected by a sharp peak observed in the accelerometer sensor value and the direction is determined from the magnetometer sensor value. As the user moves from origin to point a, his position is defined in terms of the distance travelled by the user d_1 and the direction in which the user is moving θ. A dead reckoning approach is applied to determine the successive positions of the user. For example, the position of the user at point b is defined in terms of point a as stated in Equation 2.

$$\begin{aligned} x_1 &= d_1 \ cos\theta \\ y_1 &= d_1 \ sin\theta \\ x_2 &= x_1 + d_1 \ cos\alpha \\ y_2 &= y_1 + d_1 \ sin\alpha \end{aligned} \tag{2}$$

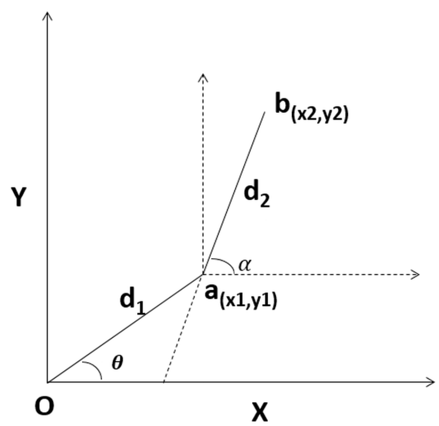

Figure 4: Dead reckoning approach for coordinates conversion.

5. RESULTS

To evaluate the proposed approach we conducted three main experiments. As the precision of the heading and step detection algorithm play an extremely important role in our work, in the first two experiments, we focus on evaluating their accuracy. In the third one, we use the whole stack of algorithms in order to study the indoor mapping solution.

5.1 Heading compensation

The heading values acquired from the compass sensor are extremely important in our inertial navigation system to discover the direction in which the user is walking. However, in environments subjected to intense magnetic fields, these values can get very inaccurate. In order to evaluate the accuracy of our compensating algorithm, we take repeated measurements of heading values in an area having high magnetic interference. We collected raw heading values as well as the compensated values (Figure 5).

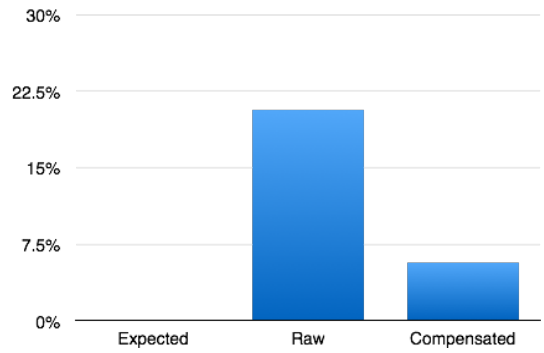

Figure 5: Heading compensation algorithm average errors. The compensating algorithm was able to reduce the average error to 5.67%.

As we can notice, without compensation, the heading values can generate an average error of 20.61%. This error corresponds to a variation of 79.2 degrees in the measured heading values, which will be detected incorrectly by our navigation system as a 90 degrees turn performed by the user. Applying our compensating algorithm, we achieved an average error of 5.67%, which corresponds to a variation of 20.41% degrees in the measured values and will not be interpreted as a 90 degrees turn.

5.2 Step Detection

By performing 15 experiments, we evaluate each phase of the step detection algorithm implemented. Three different persons simulate an indoor mapping session walking straight across a hallway counting their steps. The resulting sensor data is then sent to the Web Service, where it is analyzed and the number of detected steps is compared with the real values. The results are shown in Figure 6.

The peak detection algorithm produces an error of 105.46% on the raw data. The error gets reduced to about 1.055% when the algorithm is applied on the filtered data with additional constraints on standard variation within a window and frequency of steps. This makes us realize the importance of these constraints.

5.3 Indoor Mapping

In this section, we focus on the overall evaluation of our proposed solution. map a partial area in Bahen's seventh floor hallway. The hallway has a rectangular shape. Its mapping is performed by three different users, whose information is combined later to build the final mapping. Figure 7 illustrates the resulting mapping.

We can notice that, although the algorithm was not able to precisely detect the complete path of the user, the resulting mapping follows the hallway expected shape. Additionally,

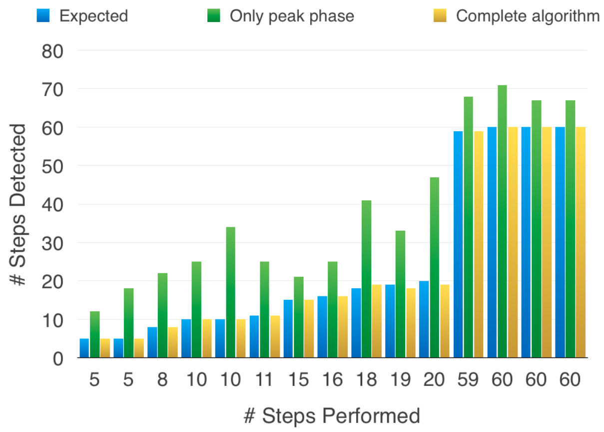

Figure 6: Step detection experiments' results. In blue is the number of performed steps and in green and yellow the number of detected steps using i) only peak detection phase and ii) complete algorithm.

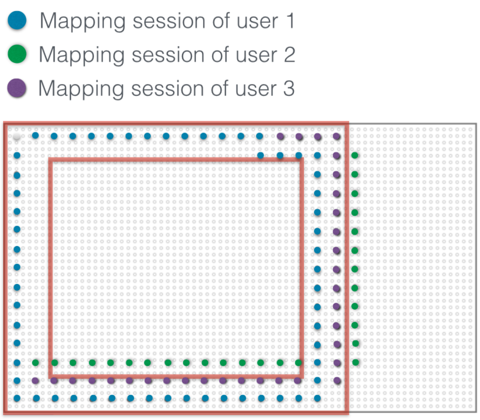

Figure 7: Resulting mapping for a rectangle hallway in Bahen building.

as more users engage to the mapping process, more precise the resulting map is.

6. FUTURE WORK

In this project, we studied the feasibility of using sensor data from mobile devices to map an indoor space. The step detection algorithm achieved a high accuracy which translates to higher precision in terms of dimensions. However, the azimuth values obtained from a mobile device are not very stable. This instability is primarily caused by high sensitivity of magnetometer sensor to external interference. More research is required to further improve its accuracy. We have thought of some approaches to improve the accuracy which need further exploration. One approach can be to combine accelerometer data along with the magnetometer and gyroscope data to determine the direction of the user. The relative orientation of the mobile device with respect to the user should be constant in this approach. As we use a dead reckoning approach for calculation of user coordinates, the results are subject to accumulation of errors. If an indoor space has bluetooth based beacons [1] at well known location, readings from a mobile device can be periodically synchronized with these beacons and accumulation of errors can be prevented.

7. CONCLUSIONS

Whilst satellite positioning systems allow devices to position themselves anywhere in the world when outdoors, they do not work well in indoor environments. This has led to the development of indoor positioning technologies, however the infrastructure requirements of existing indoor location systems make them expensive and time consuming to deploy and maintain.

We presented and evaluated a simple crowdsource-based approach to map indoor environments using inertial information of mobile devices. Although extremely simplified, our solution was able to detect user steps with an accuracy of 98.95%. Moreover, we developed an algorithm that combines data from gyroscope and magnetometers to compensate drift on the users' heading measurements, reducing the heading value error from 20.61% to 5.61%. Finally, we demonstrate how the combination of data from several different users is able to provide a relative accurate mapping of an indoor environment, differentiating free from blocked space. It is important to notice that our proposal is just a starting point for the idea of using crowdsourcing in indoor mapping approaches. We believe that several improvements are possible, as stated on future work section, possibly resulting in a solution which could be easily combined with other mapping approaches to provide an extremely accurate indoor localization system.

8. REFERENCES

[1] S. S. Chawathe. Low-latency indoor localization using bluetooth beacons.

[2] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 169–182, New York, NY, USA, 2012. ACM.

[3] D. Gusenbauer, C. Isert, and J. Krösche. Self-contained indoor positioning on off-the-shelf mobile devices. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–9, 2010.

[4] J. Link, P. Smith, N. Viol, and K. Wehrle. Footpath: Accurate map-based indoor navigation using smartphones. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–8, 2011.

[5] I. Lütkebohle. BWorld Robot Control Software. http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/, 2008. [Online; accessed 19-July-2008].

[6] A. Serra, D. Carboni, and V. Marotto. Indoor pedestrian navigation system using a modern smartphone. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pages 397–398, New York, NY, USA, 2010. ACM.

[7] Y. Xuan, R. Sengupta, and Y. Fallah. Making indoor maps with portable accelerometer and magnetometer. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, pages 1–7, 2010.