

Indoornavigation for a library

Janek Stoeck and Harald Sternberg

HafenCity University Hamburg

Abstract. To find the optimal way to a destination without the help of appropriate tools is sometimes not particularly easy or even impossible. This is also the case in the HafenCity University Hamburg (HCU) library in Hamburg. Due to the inhomogeneous signatures of the books, it is not possible to make clear assignments to specific subject areas here. This is due to the fact that the current collection has been brought together from several old collections.

To simplify this situation, an indoor navigation system based on smartphone sensors is currently being developed. Acceleration sensors are used to realize a step counter. In combination with gyroscope data a dead reckoning can be carried out. The height information is derived of barometric measurements. For additional support, routing graphs are used to constrain the walking direction.

Keywords: indoor navigation, indoor localization, dead reckoning, routing, topological support

1 Introduction

Automatically locating people in real time inside buildings is a challenging task. There are many technologies that can be used to enable real-time localization (WLAN, Bluetooth, Ultra-Wide-Band etc.). In order to realize complex applications, custom developments with infrastructure-based positioning methods are more complicated to realize, especially when it comes to linking several disciplines. This is where systems that the user already owns, such as the widely used low-cost multi-sensor system smartphone, are ideal. It combines sensor technology for tracking objects, as well as many advantages of modern computers in a portable handy device.

Such a complex application is also presented in the following project. In this project a navigation from the current location to a searched book in the library of HCU shall be offered. The reason for this is an inhomogeneous signature indication as well as a not intuitive description of the location of the book.

For this purpose, the book search is to take place on the smartphone with the website offered by the library. Afterwards the way to the book is to be represented on a map. Here, the presentation of a website, extraction of desired data, linking to a database containing information between semantic and topological information of the book, localization and tracking of the smartphone and presentation of the route and the environment on a map are used.

It is much easier for the user to use his own smartphone because he is familiar with it and does not have to find his way into a custom development. In addition, a custom development would result in a limited number of simultaneous users due to limited quantities of the system.

In the following section 2 related works are shown. Afterwards in section 3 the application with regard to localization and in section 4 the work-flow are described. In section 5 first results are shown, whereupon in section 6 a summary and an outlook into the following work is given.

2 Related Work

In [5] a algorithm based on micro electro mechanical systems (MEMS) was presented. A fusion of sensor data and map data for position estimation was performed using particle filters. The used sensors are accelerometer, gyroscope and barometer of a smartphone. The accuracy of this algorithm depends strongly on the number of particles. The more particles are used the more accurate the estimation is, so each particle influences the accuracy but also the computation time which is why it is not particularly suitable for a runtime application on the smartphone with today's technology.

Reference [3] presents an algorithm that uses smartphone MEMS sensors to estimate the position of a smartphone in a building using a particle filter. Inertial measuring units (IMU) and barometers are also used here. Furthermore, a possibility is presented to use additional sensors to limit the search range of the particles. Magnetic signals are used for this purpose. Since particle density is a critical factor as mentioned before and additional infrastructure has to be provided, this form is not suitable for the planned project.

Another system is presented in [1]. In addition to the smartphone, a smartwatch is also used, which is located at the foot ankle. The smartphone is used for a dead reckoning (DR), which is similar to the one presented in this project. The smartwatch is used to get the trajectory over double integration of the accelerations to completed path. By the property that there are phases during the step where the IMU records accelerations close to zero, zero velocity updates can be performed to minimize sensor drift. The results of both approaches are combined into a single trajectory. Although the data is promising, the use of additional hardware is not practical.

Another unpublished project at HCU also deals with indoor navigation, only on a larger scale. In this case, an indoor navigation for the building of the HCU has to be set up. The purpose of this is to make it easier for the user to find the room on the basis of the room number or the person sitting there. Here, too, there are problems in finding the right room by grouping the rooms in combination zones, without displaying the room numbers on signs.

3 System setup

The following section provides an overview of the prerequisites and an insight into the algorithms used to implement the indoor navigation system.

3.1 Prerequisites

For an indoor navigation system to certain rooms, an accuracy in the class between 2-5 m is usually sufficient to reach the desired destination. In the case of this project, this is not enough, because the shelves in which the books are located are about 0.5 m apart. The shelves are marked at the front with the signature area, so that an accuracy of 1-2 m is sufficient to detect the correct shelf. Since the shelves are loaded from both sides, the correct side of the shelf should also be displayed.

3.2 Sensors and algorithms for indoor navigation

The algorithms used were already developed in the above-mentioned project and were adopted for this project. The algorithms were also strongly based on [5].

Fig. 1 shows the principle of the inertial navigation using smartphones, which is realized by the IMU (accelerometer, gyroscope) and a barometer. The accelerometer is used to implement a step counter like in [5]. In Fig. 1 this and the step length estimation are grouped as virtual sensors, as they are based on another sensors data and further are no physical entities. The step counter and the step length estimation are used together with integrated angular velocity of the gyroscope to perform a step based DR. The barometer is used to derive height changes to identify the floor where the user is located. It is assumed that the user is holding the smartphone in his hand within an angle of roughly 45° to the ground.

First of all, the axes of the acceleration and angular rate sensors are planed in the Z direction. This is done by calculating the inclination angles from acceleration using (1) and rotate the acceleration and angular rates by them.

$$\begin{aligned}\alpha &= \arctan\left(\frac{a_y}{a_z}\right) \\ \beta &= \arctan\left(\frac{a_x}{a_z}\right)\end{aligned}\tag{1}$$

The step counter is based on the accelerometer z-axis and has two conditions to be fulfilled. First, an initial maximal threshold has to be passed followed by passing a lower threshold. If these conditions are true, a step is detected.

If the actual accelerometer z-reading passes the thresholds value including a scaling factor, the threshold value increases to the actual z-reading.

This is the advantage of this step counter, because it ensures, that it adopts the walking behavior of the user. Further the biases of the accelerometer have no impact on this method, because the thresholds are dependent to the mean value of the acceleration in z.

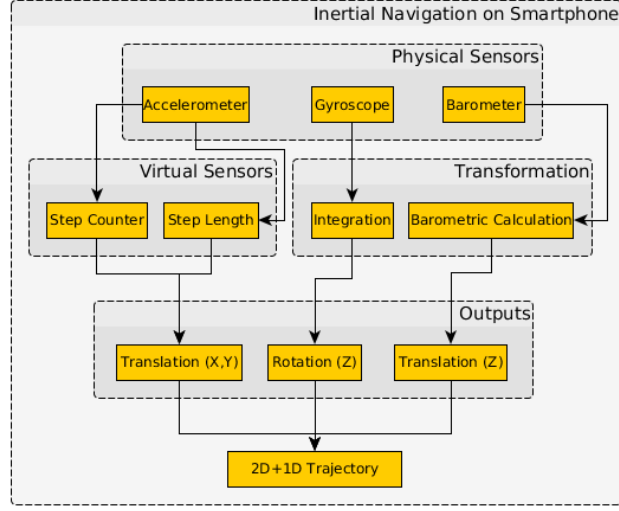


Fig. 1: System setup for inertial navigation using smartphone sensors

The step length is estimated using the accelerations as well. For this the algorithm of [4] is used with a K of 0.54. A_{max} is the highest value in the z direction and A_{min} is the lowest value during a recognized step.

$$l_i = K * \sqrt[4]{A_{max} - A_{min}} \quad (2)$$

$$r_i = r_{i-1} + \omega_i * \Delta t \quad (3)$$

The angular rates are integrated over time with (3). The actual angle r_i depends on the previous angle r_{i-1} and the actual angular rate ω_i multiplied with the time difference Δt between actual and previous reading.

Combining the information that a step is taken, the integrated gyroscope readings and the step length DR can be performed. The formula can be seen in (4), where \mathbf{x}_i is the actual position, \mathbf{x}_{i-1} is the previous position and R and t are the rotation and translation between both. R contains the angular reading around the z -axis and t includes the estimated step length. For a more detailed described implementation of the algorithm for DR and the step counter see [5].

$$\mathbf{x}_i = \mathbf{x}_{i-1} + R * t \quad (4)$$

$$h_b = \frac{288.15K - 288.15K * \sqrt[5.255]{\frac{p(h)}{1013.25hPa}}}{0.0065 \frac{K}{m}} \quad (5)$$

The height information is obtained by deriving the height from barometer data using the general height formula (5). Inaccuracies based on changing air conditions do not have a big impact, because the height information is used to identify the

floor, where the differences between them are more than 4 m. Together with the data of the DR a 2D+1D trajectory is formed.

As the integration of angular velocities leads to a time dependent drift, the system is supported by the addition of routing data. In doing so the algorithm 1 is used. If the estimated position is close to the routing graph \mathfrak{X}_P and the orientation Ψ is about 7.5° , the orientation Ψ is set to the orientation of the graph \mathfrak{X}_P and the position is set to the base point BP , which is calculated from the triangle between start and end node of the routing graph and the estimated position. In this way, the user moves on the graph. If the orientation of the smartphone deviates over 7.5° , the routing graph is left again.

Algorithm 1 Snap Position to Graph

```

if  $\overline{P(x, y), BP(x, y)} < 0.3m$  then
  if  $|\Psi - \mathfrak{X}_{graph}| < 7.5^\circ$  then
     $\Psi \leftarrow \mathfrak{X}_{graph}$ 
     $P \leftarrow BP$ 
  end if
end if

```

Another topological support is implemented through the delimitation of the areas in which the stride length changes to a given measure. This happens here when stairs are used. It is necessary because climbing stairs does not differ significantly from normal walking and therefore no correct step length can be estimated. In addition, such topological area checks can also be used to determine whether the user is at the lower or upper end of the staircase. This is particularly useful if the user's smartphone does not have a barometer. Here, the height can be estimated so that the height change corresponds to a stair step height, if the user is on the stairs.

4 Workflow

The user searches for the desired book via a web interface which is integrated in the application. Using a JavaScript code, the signatures of the borrowable books are now selected. Once a signature has been identified, the appropriate routing node is searched for using an SQL query. The user is then asked to initialize his starting point. This is done by scanning a QR code with a camera. This contains the location information (position and elevation coordinates and orientation of the smartphone).

A routing is now calculated with the help of the two nodes. For this case the Dijkstra algorithm is used, where each following node after the current get a weighting, based on its distance to the current node. The next node picked is the one with the lowest weight. This continues until the destination is found. As soon as the routing is known, the user is shown the corresponding route in a map

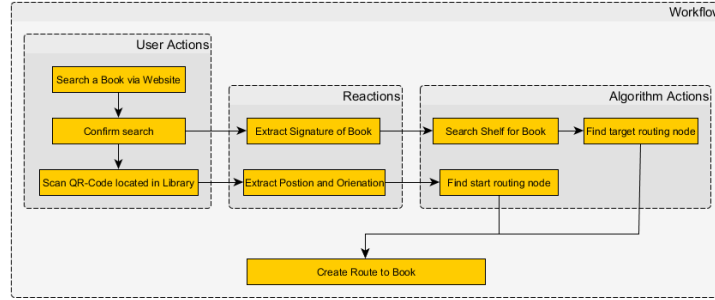


Fig. 2: Workflow of the developed android application

view. Now the position of the user is estimated using the algorithm described in section 3. Each new position updates the map so that the progress is visible.

5 Experimental Results

To investigate the inertial navigation system, several trajectories were recorded in the building of the HCU, each of which took the same course with minor deviations. In this early phase of the project the tests were done with only one person and used a LSM6DL from STMicroelectronics which is built in a Samsung Galaxy S8.

The start and end points are identical so that a loop-closure error can be calculated. This should provide information about the accuracy of the position estimation. In addition, the trajectories run over three floors, with the changes being made via stairs and elevator rides. The distance covered is 340 m. The results are shown in Fig. 3. Similar to the work flow described in the previous section, the test person had to scan a QR-Code to initialize the position and orientation in order to get an absolute position on the maps coordinate system.

The upper part of the figure shows the course of a total of four trajectories on the fourth floor of HCU. In the lower part the heights calculated from barometer data are displayed. The mean error of the four trajectories is 1.80 ± 0.25 m and thus represents a solid result of less than 1% of the traveled distance. The sensor drift could clearly get rid of with the described algorithms in section 3. Since the respective floor data are selected over the height, an accuracy of the height estimation to one meter is required. This could also be achieved.

Another test should show that the algorithms also work in smaller areas, such as the library. The library is a room of HCU, which has the dimensions of about 40 m x 20 m and extends over three floors.

For this purpose the application was used as described in chapter 4, whereby the same book was searched for in each of the thirteen tests carried out. The start was on the ground floor in front of the stairs, where a QR code was located. The destination was a book on the first floor. The test person always went from

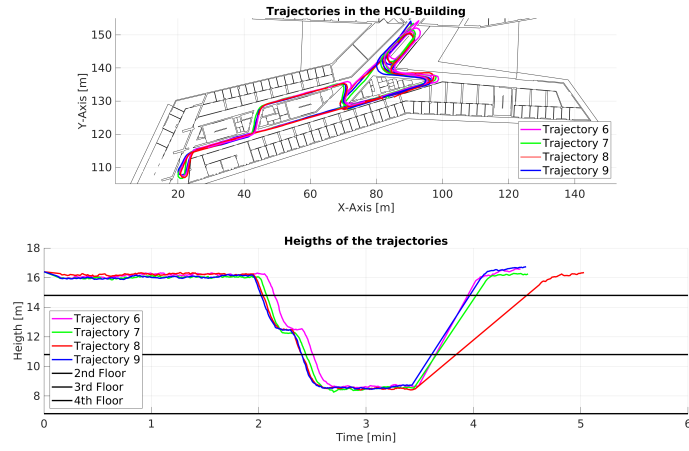


Fig. 3: Several trajectories estimated with the described setup

the starting node to the routing node in the passage between the shelves in order to determine the deviation from target to actual coordinate. As in the test before this was done by the same person with the same smartphone.

In Fig. 4 the results are shown visually. There the target node of the routing graph and the shelf in which the book is located are shown in red. The last estimated position is colored blue and the ground truth trajectory is colored green. To get the accuracy of the system the deviation of each last estimated

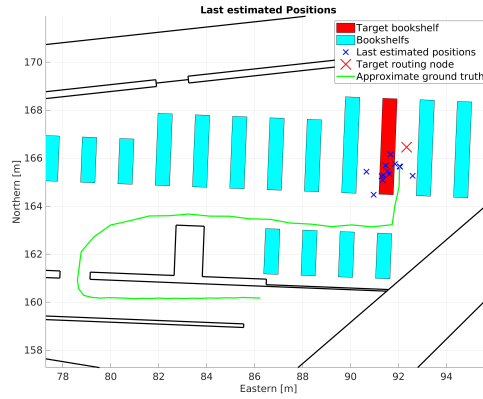


Fig. 4: Results of the position estimated in the library

position to the targeted routing node were calculated. It can be seen that the

results scatter within a range of two meters. The mean deviation is 1.35 ± 0.5 m. This is a accuracy of 4 % of traveled distance. The effect, that all deviations from last estimated position to the targeted routing node are negative, indicates that the step counter did not recognize the correct amount of steps.

6 Conclusion and Outlook

This paper presents an application for finding a book in a non-homogeneously structured library. The stock of this library was brought together from different stocks, whereby due to the expenditure a new cataloguing was not possible.

Algorithms are used which allow for the spatially accurate estimation of the position (2-5 m). By adding topological support, an attempt was made to achieve a more accurate result, since the shelves wanted are approximately 0.5 m apart.

The results are all in the range of less than two meters, which is a good result for the localization of people indoors. This also satisfies the requirement for an accuracy of less than 2 m for this project. However, reaching the right side of the shelf has not yet been achieved on the basis of the results. To achieve this, the parameters of the step counter can be adjusted or additional sensors can be used. For example, the camera could be used again to identify the correct shelf on the basis of the signature on the front of the shelf and determine the deviation from the current position.

While particle filters are out of the question due to the runtime of the algorithms, the implementation of a Kalman filter for support could be tested in the future. However, in contrast to older IMUs, the IMUs used here show a low drift. Accordingly, it will hardly be very significant in such short distances as in this application.

One can claim that a step length estimation according to [4] does not make sense for widespread use, since the K factor is a value that depends on the user. In order to avoid this we will implement and test algorithms in the future, as they are presented in [2], since they rely on the estimation of the K-factor and show a very small error according to own statements.

References

1. Gunia, M., Wu, Y., Joram, N., Ellinger, F.: Building up an inertial navigation system using standard mobile devices. *Journal of Electrical Engineering* 5, 299–320 (2017)
2. Ho, N.H., Truong, P., Jeong, G.M.: Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone. *Sensors* 16(9), 1423 (2016)
3. Real Ehrlich, C., Blankenbach, J., Sieprath, A.: Towards a robust smartphone-based 2, 5d pedestrian localization. In: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN). pp. 1–8. IEEE (2016)
4. Weinberg, H.: Using the adxl202 in pedometer and personal navigation applications. Analog Devices AN-602 application note 2(2), 1–6 (2002)
5. Willemsen, T.: Fusionsalgorithmus zur autonomen Positionsschätzung im Gebäude, basierend auf MEMS-Inertialsensoren im Smartphone (2016)