Indoor Orientation

Trevor Clack (1298022)

May 2023

1 Introduction

GPS is a modern marvel of navigation. With non-military grade receivers, we can navigate wherever we need on earth. GPS's achilles heel however is it requires a clear view to the skyline. Outside, tall buildings can interfere with navigation and indoors, it's almost useless. The goal of this project is to expand upon the previously created pedometer in order to trace out a path using a phone's accelerometer, gyroscope and magnetometer.

2 Methods

2.1 Data Collection through PhyPhox

For the original pedometer project, data was collected using Matlab's mobile app which interfaced with the phone sensor systems. This was simple to use when just extracting the accelerometer data, however when gathering data from multiple sensors at once, it became a pain to take the Matlab file object (some nested Table) and convert it to a .csv file. So an alternative application was used, Phyphox. Phyphox provides access to same 6 sensors along with several built-in extensions. It's easy to export this data to a .csv file, excel file, etc. Another reason for this switch was because of the difference in the amount of data collected. It seems to be a common error with Android devices, but although data is collected simultaneously, neither application (Matlab or PhyPhox) collected all data simultaneously. The difference was greater in Matlab wherein while 3300 points were collected from the accelerometer, just over 600 were collected from the magnetometer. In contrast, Phyphox collected 500 points from the acceleromter and gyroscope and 300 from the magnetometer.a 20% difference was favored over an 80% difference making the analysis more accurate.

2.2 Interpolation

Data again was analyzed using Python. First each of the collected data files were converted to a pandas (a python library common for data applications) dataframe indexed by time. Next the data was merged together using interpolation. As mentioned already, data was not all collected simultaneously, so to merge the data into one dataframe, interpolation was used over using time method over a linear method which would have assumed equally spaced datapoints. It also made sense to "merge down" i.e. interpolate the larger sample collections to fit the smaller ones so that fewer total data points need to be estimated. Qualitatively, there is little difference, but this choice seems safer, given the data being collected has a lot of variation.

2.3 Heading Calculation

Following interpolation we needed to estimate the heading. This is the key portion of navigation was to determine the heading. This is simply the angle of travel. Two independent data sources were used then a 3rd was calculated via a Kalman filter. A strong constraint is assumed for both methods. The phone was to be held approximately flat (screen up and top forward). In this configuration, the z-component of the accelerometer was used to track turns. A plot of all three axes are shown for a run with a single 90° counterclockwise turn then a 180° clockwise turn. To convert this to a heading, it was just necessary to numerically integrate under a simple first order approximation to the derivative using:

$$\theta_G = \omega_z \times dt \tag{1}$$

where θ_G is the gryscope's heading, dt is the time difference and ω_z is the measured angular acceleration in the z direction (positive aligned with the screen's face).

The magnetometer tracks the earth's magnetic field in units of μ Teslas. If facing magnetic north, the B_y component is approximately maximum. As one turns the y and x values will change but z will stay approximately constant. For this, we can use:

$$\theta_B = \cot^{-1}\left(\frac{B_y}{B_x}\right) \tag{2}$$

where θ_B is the heading from the magnetometer. B_x and B_y are the x and y components respectively of the magnetic field. Either the magnetometer or the gyroscope has to be negated, because as someone holding the phone turns left, the gyroscope will register a positive change in angle while a static magnetic field such as the earth's will move to the right. Additionally, the magnetometer will be in radians and needs to be scaled to degrees, but essentially these act as independent methods for determining a change in relative heading. Ultimately the gyroscope has no way to independently to determine global heading and so the magnetometer was used to determine that.

The magnetometer heading comes from measuring earth's magnetic field which is aligned approximately north. So the initial heading calculated to the magnetometer was added to the gyroscope as the gyroscope will always begin with a 0° heading. It's important to note that 0° lies along the positive x direction and headings are given such that North is 0°, so 90 degrees need to be added to the calculated headings. Finally Magnetic north is not the same as geographic north. All measurements were taken in Melbourne which at the time of being taken has a + 11° and 51 minutes declination (11.85°). This correction was added to the final measurement.

2.4 Kalman Filter

Each method was determined from data with noise and uncertainty so neither can be expected to be more trustworthy. Instead we can implement a Kalman filter to fuse measurements from the magnetometer and gyroscope, improving the accuracy of the heading calculation. The filter takes into account noise and uncertainty in the measurements to produce a more reliable estimate of the true heading. As python was used for data analysis, a convenient library called Filterpy has a Kalman filter class. First a state vector of the measurements of interest was created (Heading from both the magnetometer and gyroscope), then four matrices were created:

1. A state transition matrix which defines how the state vector evolves over time. As the heading will evolve based on angular velocity of the gyroscope, this was chosen to be defined as [1 dt; 0 1] where dt is the time step between measurements.

2. A measurement matrix which defines how the state vector is related to the measured values. As the magnetometer data is used to measure heading directly (as opposed to indirectly with the gyroscope), a good choice is [1 0].

3/4. Process noise matrix and measurement matrix. This is uncertainty in the state and measurement respectively. It was uncertain how to gauge this in the specific use case, so a generic set was used. The filter was initialized with the initial state vector. Then the filter class has built-in predict and update methods which use the state transition matrix and process noise matrix to predict the next state. In the update step, the difference between the predicted value in the previous step and the actual measured values are found. These predict and update steps are completed for the duration of the length of the data collected.

Several variations were tried on the setup of the Kalman filter. Initially headings were calculated using the gyroscope and magnetometer data, then the Kalman filter was setup using these two parameters to make predictions from. Results of the two-parameter processing are shown in figure 1.

Later, a three-parameter setup of the Kalman filter was used, this time using the relevant raw data, namely the magnetic field measured in the X and Y directions and the angular frequency about the Z axis. The other measurements were not considered relevant as they aren't used in calculating heading. The results of these performed on the same data is shown in Figure 2.

3 Results

Figures 1 and 2 show the result of using a Kalman filter using two and three parameters respectively. The three parameter Kalman filter runs (red) qualitatively appear to more closely match the actual paths (green). Overall the gyroscope in this particular device has performed qualitatively quite well in terms of making paths that line up with the actual path apart from initial heading which was determined from the more variable magnetometer data.

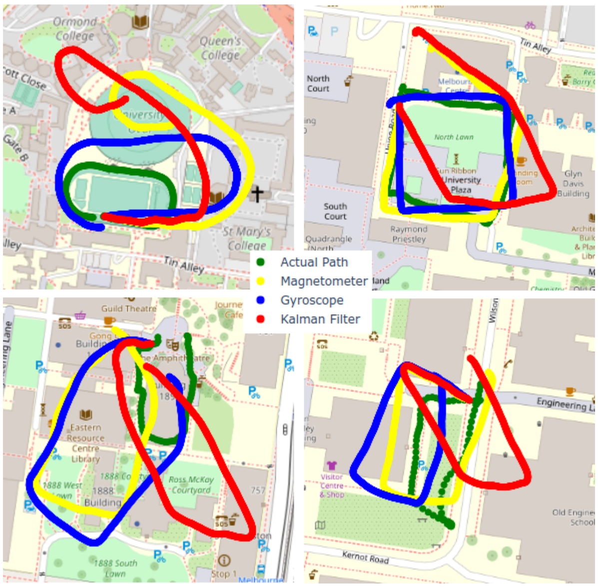

Figure 1: four runs showing the actual path (green from GPS data), and the paths according to turns from the gyroscope (blue), magnetometer (yellow) and a two-parameter Kalman filtered result (red).

4 Discussion

4.1 Improving the Kalman filter implementation

The green paths represents the actual route taken and they are the same in figure 1 and 2. Figure 1 demonstrates that with equal weight between the magnetometer and gyroscope and two parameters that the Kalman filter doesn't match up as well. Compare this to figure 2.

Figure 2 differs from figure 1 from the red-path represents a three-parameter Kalman filter, where the values being predicted and updated was the raw data itself, rather than the heading calculated from the raw data. This path is also less influenced by the magnetometer's variance because weightings were assigned

Figure 2: four runs showing the actual path (green from GPS data), and the paths according to turns from the gyroscope (blue), magnetometer (yellow) and a three-parameter, weighted Kalman filtered result (red).

such that the more stable gyroscope data with its smaller variance tends to give a path more closely aligned to the actual path taken. Note: initially the only change to be made here was the three parameters, the weighting was an idea that provided further improvement based on variance of the measured data, with more time and research, it's likely that an improved version of this can still be found.

In every case, the gyroscope had the cleanest (qualitative) path often forming closed loops and no noticeable drift that more closely reflect the actual paths taken. The magnetometer in the device used has more variation and drift (particularly noticeable in the upper-left runs around the track and field path) It is for this reason that the parameters of the Kalman filter were tweaked to favor the gyroscope by a factor of 12 (calculated from the average magnetometer variation collected when the device is laying still).

4.2 Difficulties and Lessons Learned

There's a calibration step normally required for magnetometer data to account for hard iron and soft iron error. Phyphox has a checkbox (enabled by default) that accounts for this error, however, it's not a simple task and in order to take advantage of this automated process, the phone needs to be moved in a "figure of 8" pattern. This saves time in processing the data afterwards as well.

At various points, collected data implied very strange behavior such as sudden reversals. For instance a collection taken along the track-and-field course would suddenly hook right as if the poles reversed. After some of these observed events, a check was performed in which a full circle was made. Expected behavior was the X and Y components would vary between some positive and maximum value centered at zero. But instead the value may be centered at some other position and all be positive, which doesn't make physical sense. However doing a "figure of 8" movement several times would allow the sensors to calibrate itself against magnetic interference. Figure 3 shows such a routine where some circles were performed, followed by a "figure of 8" pattern, followed by more circles after which better measurements were yielded. It also shows a run with unexpected multiple signals. Phyphox providing live plots allow for better assessment of the data in real time without the need to process it first on a computer making it a far more preferred tool over Matlab's API.

To briefly describe what this does, there's a known "soft iron" and "hard iron" effects. Moving the phone in a "figure of 8" samples points in all directions. Ideally this would map to a sphere as the net magnetic field is approximately constant in one region. But these hard and soft iron will distort this and yield a "squashed sphere". The automated step performs a least squares ellipse fit which can be shifted and scaled to give the expected sphere. Many samples of this code can be found online, however, as it's provided by Phyphox, it's important merely to understand how it works and be aware of the source of this error.

Another difficulty encountered is that there are several independent steps in this indoor positioning which must line up for an accurate working product. The pedometer must be accurate. When creating the pedometer alone, a threshold was selected that worked best between a dozen samples collected while running, walking, with the phone in the hand, in the pocket, etc. As all the runs made here relies on the phone being held horizontal, a different threshold was selected however some of the runs had accurate counts, and some were inaccurate. Sufficient time was not able to be allocated in improving the step count at this time. But it is one important step in an accurate positioning system when can be returned to later.

the Magnetometer (apart from being properly calibrated) must consistently indicate the correct direction, this is critical to properly orient the more accurate gyroscope heading data. Longer runs tend to yield poorer results as the magnetometer seems to still drift. The upper runs in Figures 1 and 2 are over 500 and over 300 steps. Drifting is very clear in the track-and-field track and begins near the end of the square around the North lawn.

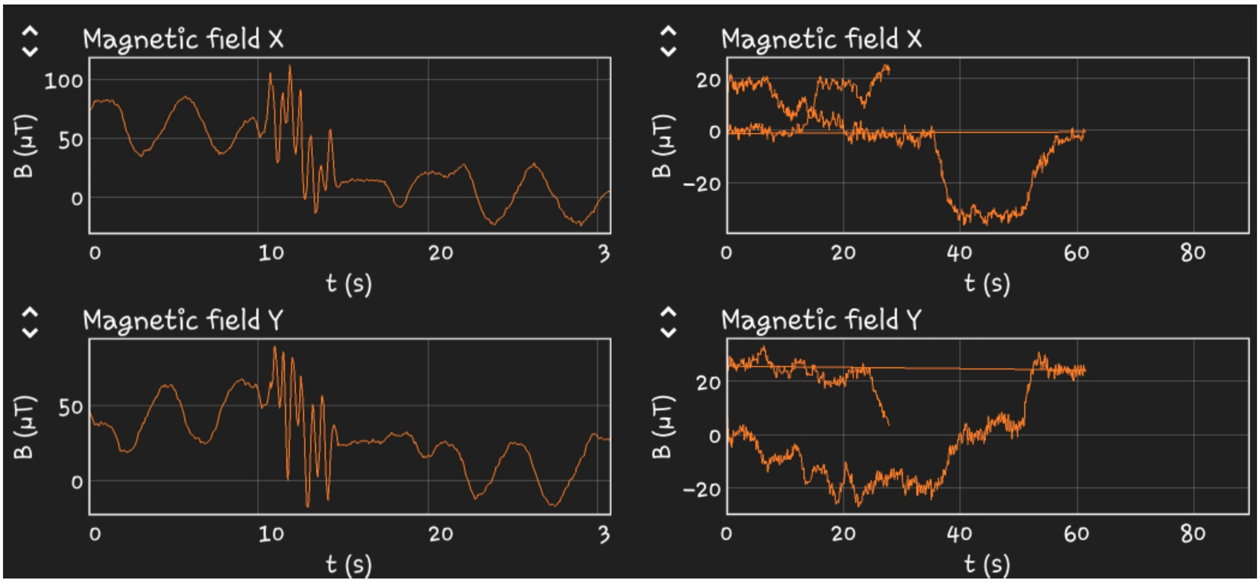

Figure 3: Two examples of bad magnetometer data. Left: A common issue where the magnetometer over a full circle show be centered over zero. A procedure of moving in a "figure of 8" pattern is an effective strategy to calibrate prior to collection. Right: Inexplicable interference which has been be observed in real-time, which can save time by flagging issues immediately

5 References

Magnetic declination in Melbourne is +11 degrees 51 minutes:
https://www.magnetic-declination.com/

Latitude and longitude conversion:
https://www.usgs.gov/faqs/how-much-distance-does-a-degree-minute-and-second-cover-your-maps

Kalman Filter insight:
The Kalman filter class used in python and learning how to setup parameters are from here:
https://filterpy.readthedocs.io/en/latest/kalman/KalmanFilter.html