Dead Reckoning, a location tracking app for Android™ smartphones

Nisarg Patel
Mentored by Adam Schofield and Michael Caporellie

Introduction

Many modern conveniences rely on the ability to pinpoint accurate location. GPS has made this job significantly easier, however the system is not perfect. There are many places where GPS signal is restricted, from underground tunnels, to inside large buildings, to outdoors if surrounded by large structures. The Dead Reckoning app offers a solution by using smartphone inertial sensors to track location, making the tracking process self-sufficient and non-reliant on GPS.

Methods

The app was developed using the Android Software Development Kit on the Android Studio Integrated Development Environment and tested on a Google Nexus 5 smartphone.

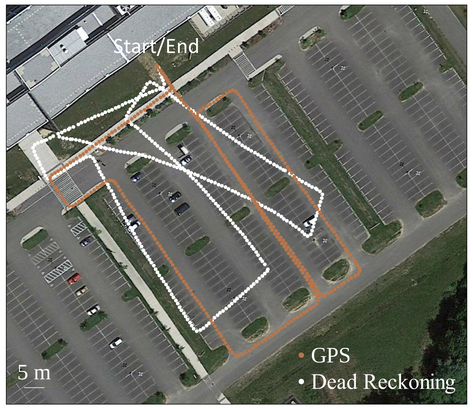
The process of building the fully functional location tracker was split up into multiple parts. During each part, standalone app components were built to be later incorporated into the main project. These components include a step counter, a compass, a data collector, a QR code scanner, a calibration mode, and an interactive graph. The app also features a user interface that allows the user to create profiles and store tracking preferences such as stride length and step counter sensitivity.

When the app is started, it determines the user's initial orientation relative to Earth by analyzing magnetic field and gravity data via a Direction Cosine Matrix (DCM) (Freescale Semiconductor, 2012). This initial orientation serves as the origin to which future changes in location are added. The step detector begins counting steps by finding peaks in the accelerometer data, where each step taken produces a spike. When a step is detected, distance traveled is calculated by applying the stride length of the user to the step taken. The gyroscope continuously monitors angular rotations via another DCM (University of Cambridge, 2007) to track changes in heading. Together, the change in heading and distance traveled define a position vector which is applied to the last recorded location to calculate the current location. The current position is plotted to an on-screen graph and stored in various data files for later use.

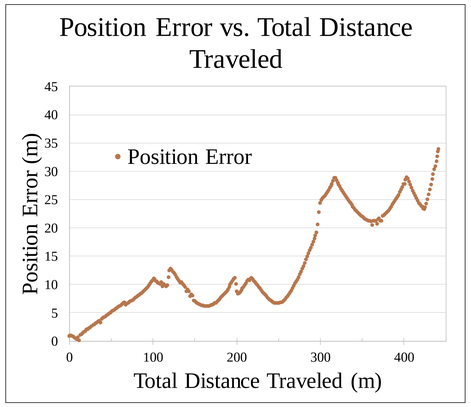
Results

The app was tested outside against a NovAtel® ProPak6™ Triple-Frequency GNSS Receiver, which is accurate to 20 cm. The GNSS truth location was compared to the tracked location.

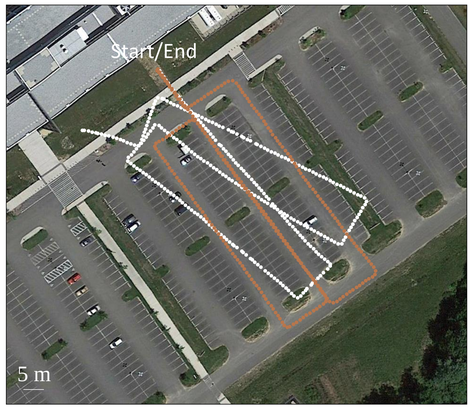
Results (cont.)



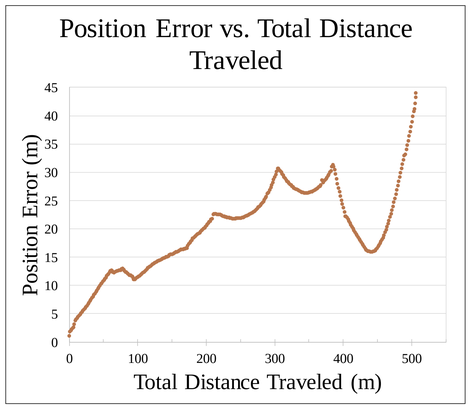
Picture 1: Walk 1 – 450 m traveled



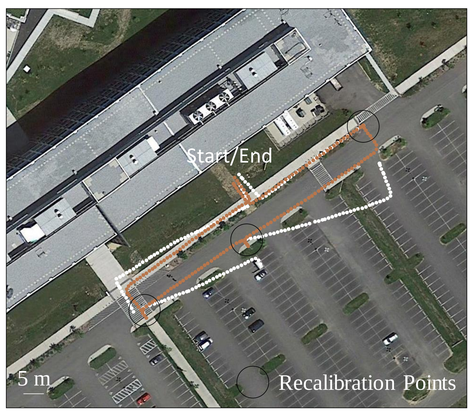
Graph 1: Walk 1 – 450 m traveled



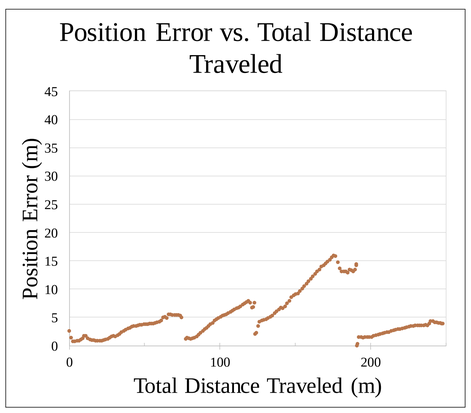
Picture 2: Walk 2 – 550 m traveled



Graph 2: Walk 2 – 550 m traveled



Picture 3: Walk 3 – 250 m traveled



Graph 3: Walk 3 – 250 m traveled

Results (cont.)

The first walk covered a distance of 450 m (Picture 1), with the error accumulating to 35 m (Graph 1). The second walk was longer, covering a distance of 550 m (Picture 2), with an error accumulation of 45 m (Graph 2). The third walk covered a distance of 250 m in 4 separate segments (Picture 3), calibrating between each walk. The error started to accumulate as the walk processed, and reset when the phone was recalibrated (Graph 3). Without recalibration, error increased linearly with distance traveled. The recalibration was akin to a waypoint system, whereby calibrating at each waypoint would reaffirm the correct location, and allow the error to be reduced to minimal levels.

Conclusion

For the app to be viable in navigation, it needed to have a maximum error margin of 3 m at any given time. Although the app tracked accurately initially, it started to quickly accumulate drift, which ranged from as little as 1 m to larger than 45 m depending on the distance traveled. A solution for the error was to recalibrate the app between walks, which reduced the error down to minimal levels, below 3 m, with every calibration. For the best accuracy, calibration would have to be repeated every few meters to consistently control error. Although the app may not be suitable for navigation in its current state due to the accumulating error, it is an excellent first step and proof of concept. In the future, accuracy may be increased by using better sensors than those available on smartphones, filtering sensor data to reduce noise, fusing sensor data from additional sensors, and the use of visual aids in addition to inertial sensors.

References

Freescale Semiconductor. (2012). Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors (AN4248). Texas: Ozyagcilar, T.

University of Cambridge. (2007). An introduction to inertial navigation (1476-2986). England, UK: Woodman, O.

Acknowledgements

Thank you to Adam Schofield and Michael Caporellie, my mentors, for guiding me in this project. They have provided me with countless resources to build the app from bottom up and get it working. Thank you to Gary Davis, my faculty advisor, for keeping the project focused and on track. This project would not have been achievable without the assistance of these individuals.