

Visualizing Mobile Phone Sensor Data in an R Environment

Riccardo Miccini
Technical University of Denmark - DTU

Objectives

The aim of the project is the application of methods for visualizing mobile phone (Android, iPhone) sensor measurements in an R environment, using the *Google Cloud* as buffer. The system has to be able to:

- Collect GPS positioning data from mobile phones and store them remotely.
- Read the content of the spreadsheet from an R environment in real-time, and visualize spatial data on a map and other information on plots.
- Allow the user to interact with the data - filtering, zooming, scrolling, exporting...

Introduction

The project has been carried out under the supervision of profs. John Aasted Sørensen and Ian Bridgwood, as part of a multidisciplinary project.

The implemented system is composed of three main elements: a series of end users' mobile devices, a remote host, and a *data analyst* station. The former are equipped with a custom-made application capable of submitting GPS data to a remote *Google Sheet* document, which acts as a database and is accessible through the cloud. The data analyst can then visualize the collected data in real-time, using the provided R scripts and a web browser.

All brands, product names, logos, or other trademarks featured or referred to in this document are the property of their respective holders, and their use does not imply endorsement.

Requirements

Here is a summarization of the functional and non-functional *Software Requirements Specification*:

- The collected data shall include device ID, coordinates (latitude, longitude), altitude, speed, and timestamp.
- The mobile phone app shall submit data at a user-defined time interval.
- The R software shall visualize the spatial information using a map and any other data using a chart, in real-time.
- The user should be able to use the R application to filter and export the data.
- All the developed source code should be modular, reusable, and well-documented.

Tools

Android Studio is the official IDE for native Android development. It is distributed freely by Google for most platforms, and it is based on *IntelliJ IDEA*, a proprietary Java development tool. It features:

- Code editor with completion and refactoring
- UIs designer, code generation wizards
- Integrated debugger and emulator

RStudio is a free and open-source IDE for R, a language and software environment for statistical computing, data analysis, and visualization. The software comprises a text editor with code completion and syntax highlighting, an interactive command interpreter with built-in debug, command history, and data viewer.

The capabilities of R can be extended through a system of packages, which may provide graphics tools, statistical and data handling functions, or software bindings and APIs. **Shiny** is a framework for building live web applications in R. It comprises control widgets and graphic outputs, and uses a reactive model for determining which parts of the pages needs updating. Other used packages are:

leaflet Maps service with custom overlays
plotly Interactive and customizable plotting engine
googlesheets Wrapper for the *Google Sheets* APIs
dplyr Functions for manipulating data

☐ Update in real-time

Filter data

Date range

2017-01-30

to

2017-02-05

Device IDs

c262a4e4-6724-417e-9949-25a2f40dd8be

RESET FILTERS

Export data

CURRENT DATA

ALL

CLEAR REMOTE DATA

Implementation

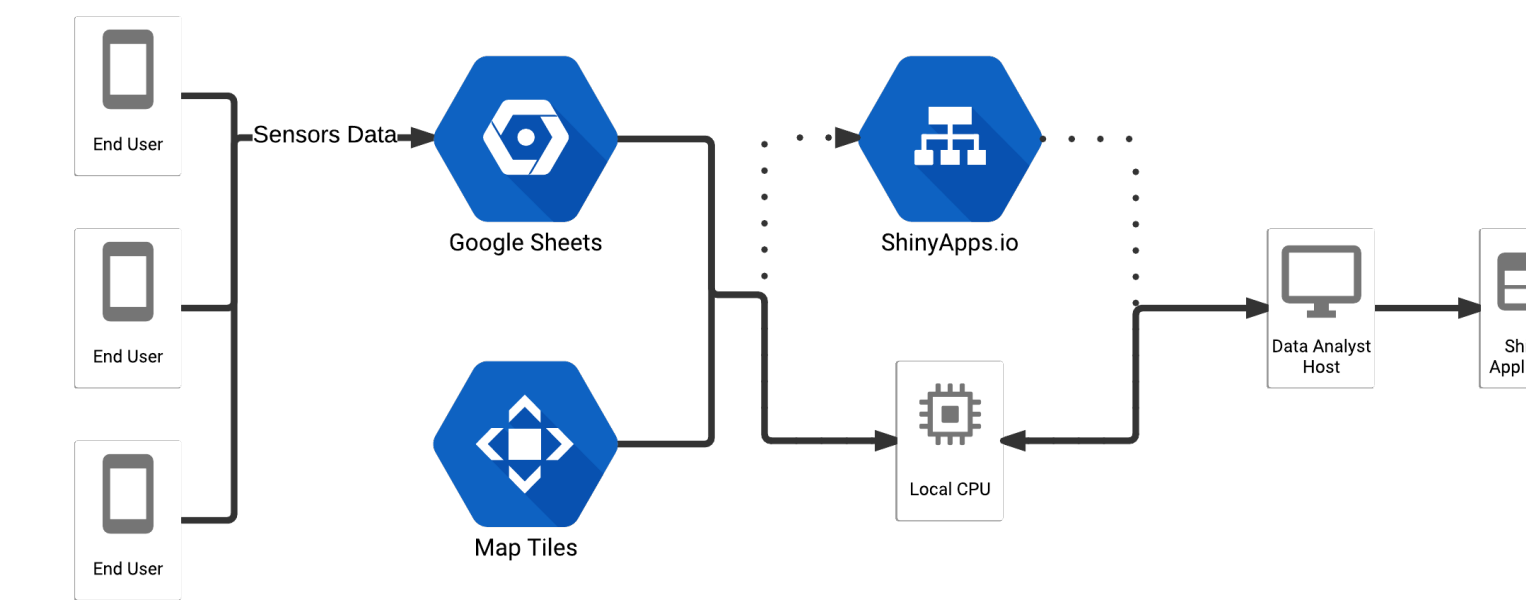
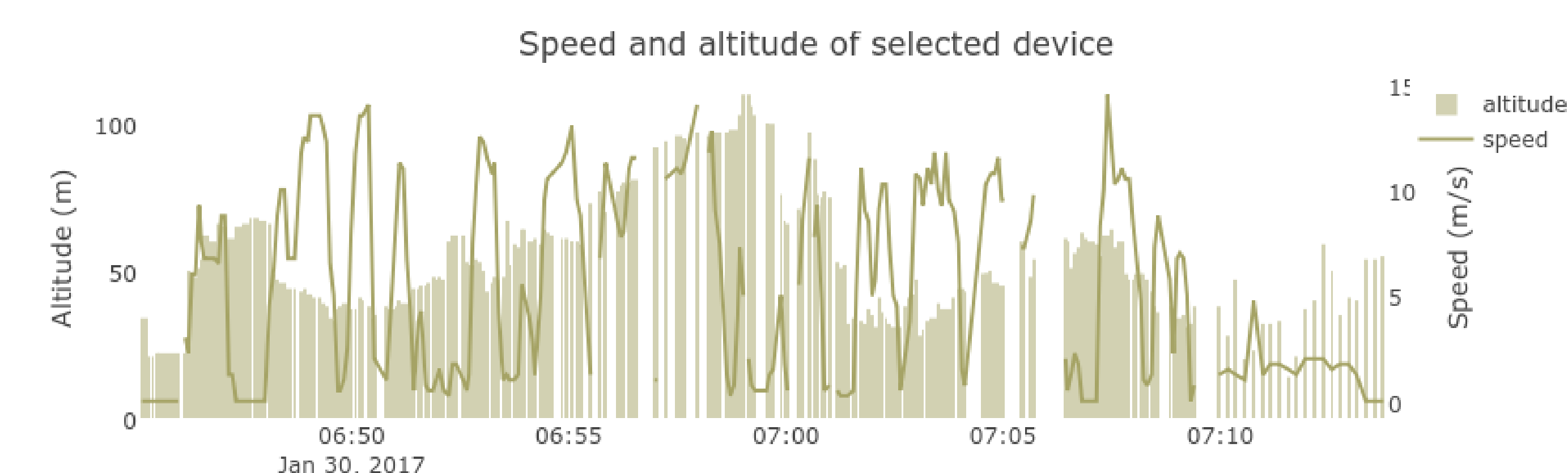
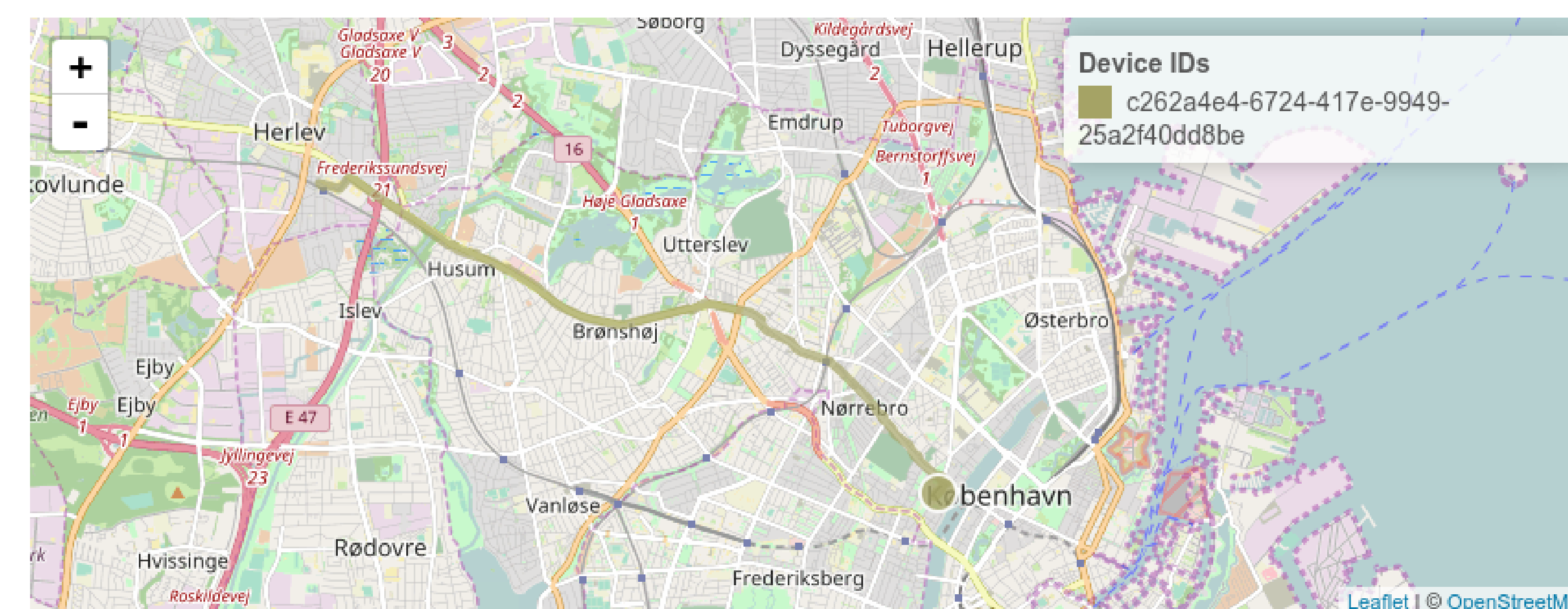


Figure: Overall system infrastructure: network diagram

The **phone app** is responsible for collecting the GPS data and submitting them to the remote storage location, provided by *Google Sheets*. Sheets is accessed through its APIs that allows third-party applications to manipulate and manage documents. The Android version of the app has been developed in JAVA using the *Android Studio* environment and the *Android SDK* libraries. A corresponding *iOS* application has also been developed, performing the same tasks as the Android version.

The **R application** takes care of gathering the data from the spreadsheet, filtering it, and visualizing it on a web UI composed of a map and chart. It is built around the *Shiny* framework, is accessible through a web browser, and can be hosted remotely. The application is composed of a front-end script which specifies the layout and widget elements of the webpage, and a back-end script that periodically queries the Sheet document, filters its data according to the user input, and generates and shows the map overlays and plot.



Verification

The functionality of the developed solution has been verified by conducting *acceptance testing*. The following tests have been performed and passed:

- Phone app
 - Press button ▶
 - Insert time interval, press button ▶
 - Press button ||
 - Verify content of the Google Sheets document
- R Shiny application
 - Load R application
 - Use UI controls to select a single device
 - Press button 📁 *all*, inspect the file
 - Filter data, press button 📁 *current data*, inspect the file

The performances of using Google Sheets as storage back-end has been assessed by determining its network capacity, as well as the computational power required after each update. The devised test consists in an R script repeatedly querying data from the Google Sheets servers, with increasing numbers of rows.

Results and Conclusion

The project has reached a satisfactory level of completion. Nevertheless, the system is far from optimal and could still benefit from further development, in particular:

- Allow app to track data during background operation
- Improve R application responsiveness by limiting the amount of transmitted data
- Adopt a different storage back-end, e.g. Google *BigQuery*

The project has been a compelling opportunity for exploring the Google Services ecosystem as well as the Android and R development environments.

References

- [1] *Google Sheets API*. URL: <https://developers.google.com/sheets/api/>.
- [2] *Introduction to Android*. URL: <https://developer.android.com/guide/>.
- [3] *Leaflet for R*. 2014. URL: <https://rstudio.github.io/leaflet/>.
- [4] *Plotly R Library*. 2014. URL: <https://plot.ly/r/>.
- [5] *Shiny, a web application framework for R*. 2012. URL: <https://shiny.rstudio.com/>.

Contact Information

Riccardo Miccini
 s137345@student.dtu.dk

miccio-dk
 rimiccini