Road Quality Assessment with Cyclists Crowdsourcing

Federico Russo, Gabriele Marcuccetti, Luca Di Giacomo
f.russo50@studenti.unipi.it, g.marcuccetti4@studenti.unipi.it, l.digiacomo3@studenti.unipi.it

ABSTRACT

This paper presents Bike Assistant, an Android application designed to enhance users' cycling experience by providing alerts about areas with poor road quality and offering feedback on speed and total distance traveled. The application employs the Dynamic Comfort Index (DCI) calculation method described in the research paper "RIDE VIBRATIONS: TOWARDS COMFORT-BASED BICYCLE NAVIGATION" [1], with certain modifications tailored to the application's context. GPS data from Open Street Map APIs is utilized to track current and average speed. The GPS trace, along with the associated DCIs, is sent to the backend, where data is accumulated and categorized into zones using a centroid-based clustering technique. The zones, identified by their respective centroids' coordinates, are hierarchically aggregated and stored in Firebase's Firestore. The application's frontend runs on Android devices and consists of three activities: the Main Activity, which serves as a landing page for starting a new ride or accessing the ride history; the New Ride Activity, which displays a map view of the city with a heatmap indicating road quality; and the Profile Activity, designed to visualize past ride information such as date, distance, average speed, and DCI values. The backend follows a serverless architecture utilizing Google Cloud Platform, with Firestore as the database and Google Cloud Functions employing the NodeJS runtime. Experimental results and possible future improvements are discussed, including the introduction of individual road heatmaps, consideration of multiple parameters beyond DCI, and automatic detection of the user's transportation context.

1 Introduction

Bike Assistant is an Android Application developed with the aim of helping users to have a more enjoyable cycling experience by alerting them on zones with a bad overall road quality, but also giving the user some feedback on the speed and the total distance traveled, with a straightforward and intuitive interface.

To achieve the road quality measurement, the procedure stated in the: "RIDE VIBRATIONS: TOWARDS COMFORT-BASED BICYCLE NAVIGATION"[1] paper has been followed for the calculation of the Dynamic Comfort Index (**DCI**) with the introduction of some changes due to the context of the application, in particular:

- A frequency of 200Hz of the z-axis accelerometer has been used for the sampling.
- The initial state of the accelerometer is removed from the measurements to discard the initial gravity acceleration of G~9.8m/s2 by using the rotation matrix to obtain the inclination of the smartphone.
- Accelerations are gathered over a window of 1 second for then being aggregated into a single DCI measurement using an average of the readings, this decision has been taken due to the refreshing ratio of the GUI of the application that had to follow the minimal GPS sampling ratio of 1Hz in order to maintain coherency between what is seen in the application and what is sent to the backend of the application.

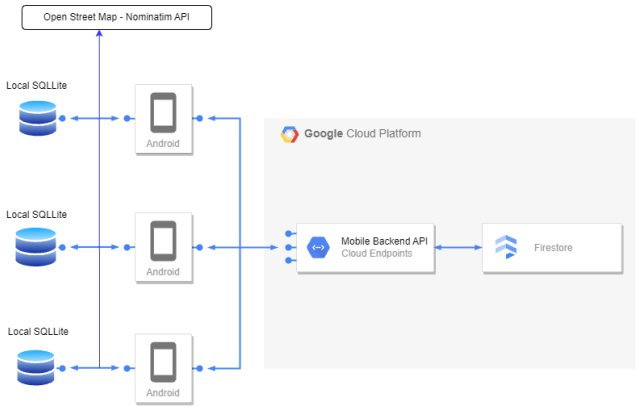
Since data on the location are collected using GPS locations over Open Street Map APIs, they are also used to keep track of the current speed and the average speed, the first calculated as the simple difference in location readings and the second as a cumulative sum of the readings divided by the total number of samples (seconds), then a route is built by aggregating the single location readings with their relative evaluated DCI (average DCI@200Hz over the 1-second window for the refresh rate of the location) in a similar approach observed in the "BikeMate: Bike Riding Behavior Monitoring with Smartphones"[2]. The GPS trace associated with the respective DCIs is sent to the backend where data are collected and cumulated by zone using a centroid-based clustering where the centroid is calculated as the center of a fixed dimension cell of about 250 meters ($0.25km / 2\pi R = \Delta\vartheta° / 360°$, valid for small variation in angles and assuming a constant earth radius, which generically fits with the context of small trips). relatively to the osm tag's center[3] (town, city, suburb, and village tags are monitored). Finally, three different levels of aggregation are stored in Firebase's Firestore: (Town | City) levels holding the data over an entire city and containing multiple sub-levels of aggregations by (Suburb | Village | Neighborhood) holding the single zones' statistics each one having a final level of aggregation which is the squared cell, identified by its centroid's coordinates.

2 Architecture

Each Android device communicates with instances of two main cloud functions (better described in the Backend section), working as middlewares between the application frontend and Firestore.:

- GET /geomap/{city}
- POST /geomap/{city}/{zone}

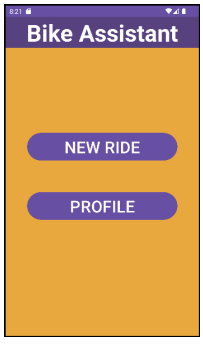
The device additionally uses its own SQLLite storage to cache and save information of the routes locally for both speeding the process of computing new zone's grids dynamically as the user moves in a city, and having local feedback on the previous rides of the user.



2.1 Frontend

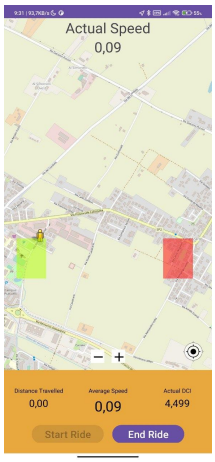
The GUI runs on Android devices and is composed of three activities

2.1.1 Main Activity



The main activity works just as a landing page, allowing whether to start a new ride or to check its profile for past rides.

2.1.2 New Ride Activity



This activity shows a map view of the city where the device is located and displays a colored heatmap of the zones using green color to display cells having an average DCI value close to 1, red for cells with DCI values close to 0 and yellow for in-between values. Moreover, this activity allows one to zoom in and out adapting the displayed heatmap to entire zones and cities as well as data about the current ride

At the instantiation of the activity a request to the GET /geomap/{city} endpoint is sent to get the whole document of the current city so that new requests are only sent when a change in the current city is observed. The document obtained as a result of the request is stored in the device's local memory so that changes in zones are very quickly displayed.

In addition to that, when a change in city or zone is detected a request to the POST /geomap/{city}/{zone} endpoint is delivered to have a better separation of the route's points.

The city and zone changes are detected using Nominatim API that allows converting the location from GeoPoint locations into structured data including the tags for city and zone.

2.1.3 Profile Activity

It is an activity with the purpose of visualizing the history of the past rides, showing information about the date, the distance traveled, the average speed, and information on the DCI of the traveled path

2.2 Google Cloud

The backend follows a serverless approach where the database and the APIs are implemented over Google Cloud Platform by using respectively Firestore and the Google Cloud Functions over NodeJS runtime. For the Cloud Functions two APIs have been developed:

1. GET /geomap/{city} → in order to make as few calls as possible, this API gets a city as url parameter and returns the entire document of that city, formatted as follows:

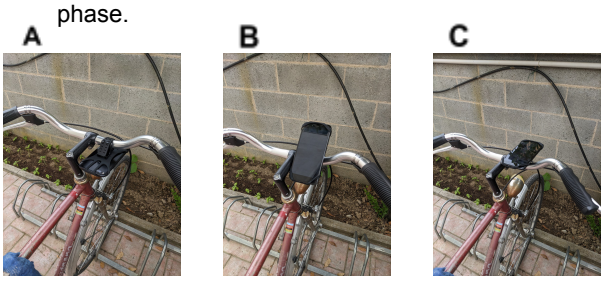
```
type CellDoc ={
  center: GeoCoordinates;
  avgDCI: number;
  nSamples: number;
  cellID: string;
  bbox: GeoCoordinates[];
};

type ZoneDoc = {
  center: GeoCoordinates;
  avgDCI: number;
  cells: CellDoc[];
  name: string;
  nSamples: number;};

type CityDoc = {
  center: GeoCoordinates;
  avgDCI: number;
  zones: ZoneDoc[];
  nSamples: number;
};
```

2. POST /geomap/{city}/{zone} → aggregates data of a ride updating the according city document and the respective zone, by calculating the relative centroid with respect to the zone's center taken from OpenStreetMap's tags

3 Experimental results

Experiments have been carried out to evaluate the impact of multiple aspects over the DCI calculation, with particular focus on the orientation of the smartphone and its position.

The results show the importance of placing the smartphone in a stable manner, while the orientation does not have an impact, thanks to the use of the rotation matrix to extract the degree of tilt of the cellphone in the initial phase.



In the above images, we expose examples of bad positioning (A) or good positioning (B; C)

With a correct setup, we also performed tests on different roads, verifying the ranges of DCIs are not too much impacted by the handle's vibrations and are coherent with the results shown in the paper [1].

Below we show an example of results over a good and a bad quality for streets:

0.390 0.142

0,100 0,112

4 Conclusion

We developed the idea of this application based on the intuition of the DCI, trying to make it usable in practice, in the context of bicycles. The aim was to try to make cycling in the city safer and to collect data on the quality of the roads. In the implementation we tried to adapt the features to common cases during cycling, such as taking into account the angle of the phone as much as possible, trying to discard possible outliers within the measured DCI values, trying to give an idea in real-time of the conditions of the area in which the user is passing and of the possible accessible areas nearby.
Through the application we have given the possibility to have access to information regarding the user's past history, useful to help him improve the experience.
Via the backend, data regarding the quality of the road and the use of certain routes is collected. Thanks to this information the work could be used as follows:
- it would be possible to monitor how many users travel on a given road, as the number of samples collected in the various areas is taken into account. In this way, information could be obtained concerning the use of bicycles in the city and in which areas they are most used, or perhaps which user target uses the bicycle most and in which context or area of the day;
- it would be possible to monitor the road conditions of the various areas of a city in real-time, so as to be able to verify why a certain section is not used and intervene promptly if necessary.

Possible improvements to the work done are:
- the introduction of a heatmap on individual roads. One would no longer look at the road conditions by area of the

city but at the specific road traveled;
- take into consideration several parameters, and not just the DCI;
- automatic detection of the user's context, i.e. being able to understand if the user of the application is on board a bicycle, a car or some other vehicle.

REFERENCES

[1] RIDE VIBRATIONS: TOWARDS COMFORT-BASED BICYCLE NAVIGATION
https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-367-2020
O. Wage, U. Feuerhake, C. Koetsier, A. Ponick, N. Schild, T. Beening, and S. Dare

[2] BikeMate: Bike Riding Behavior Monitoring with Smartphones
https://dl.acm.org/doi/proceedings/10.1145/3144457
Weixi Gu, Zimu Zhou, Yuxun Zhou, Han Zou, Yunxin Liu, Costas Spanos, Lin Zhang

[3] Open Street Map Official Documentation for tags
https://wiki.openstreetmap.org/wiki/Tags