BikeVibes: An App for Crowdsourcing Open Road Quality Data From a Cyclist Perspective

Kai Luedemann
University of Alberta
Canada
kluedema@ualberta.ca

Mario A. Nascimento
University of Alberta
Canada
mario.nascimento@ualberta.ca

ABSTRACT

This paper presents BikeVibes[1], an app that cyclists can use to log data regarding the smoothness of their rides. The main goal of BikeVibes is to facilitate the collection of anonymized open data about road quality that others can download and peruse. A few sample scenarios where having this type of crowdsourced data would be useful are as follows. A city can use the gathered data in order to determine which roads need to be maintained/upgraded since the quality of the road can be perceived very differently when riding a bike compared to driving a car. Likewise, a city can determine paths that are more frequently used by cyclists in order to decide where to build or upgrade dedicated bike lanes and/or how to prioritize maintenance. Also, third-party app developers can use the road quality data to suggest paths to cyclists based on smoothness, as this may be an important attribute for some people, e.g., in the case of parents riding bicycles hauling trailers with children. None of these scenarios could be easily contemplated without the availability of data such as that gathered through BikeVibes.

CCS CONCEPTS

• **Information systems** → Spatial-temporal systems.

KEYWORDS

crowdsensing, open cycling data, road quality monitoring, mobile app

ACM Reference Format:
Kai Luedemann and Mario A. Nascimento. 2022. BikeVibes: An App for Crowdsourcing Open Road Quality Data From a Cyclist Perspective. In *The 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS '22), November 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557991.3567779

[1]https://www.bikevibes.ca. The BikeVibes app, for now only available as an Android OS app, is available for download at https://play.google.com/store/apps/details?id=com.bikevibes.bikeapp. The source code for the app and the website are available on GitHub at https://github.com/kluedemann/bikevibes-app and https://github.com/kluedemann/bikevibes-website, respectively

IWCTS '22, November 1, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9539-7/22/11…$15.00
https://doi.org/10.1145/3557991.3567779

1 INTRODUCTION

Sustainable transportation has become a major topic for cities around the world. This has led many municipalities to place a greater focus on cycling infrastructure, in particular the construction of new bicycle lanes or paths. As with any form of infrastructure, it is important to consider not only the quantity but the quality of these paths. With that in mind, we have developed BikeVibes,a smartphone application that uses the device's accelerometer data to crowdsource information on the roughness (or smoothness) of roads and bicycle paths. Additionally, we have created an accompanying website that allows anyone to view, query, or download the anonymized data collected.

Cities have various methods of monitoring the surface quality of their roads. They may use scanning vehicles or have means for individuals to report major surface imperfections such as cracks or potholes[2]. However, this information is primarily from the perspective of motor vehicles. Whereas cyclists may be able to avoid larger obstacles, they are more affected by smaller imperfections or the presence of debris. The authors of [1] suggest that there is a lack of correlation between the international roughness index, a commonly-used measure of passenger car comfort, and the vibrations experienced by cyclists. Thus, a means to measure road roughness from the perspective of cyclists is crucial to evaluate the quality of a city's cycling infrastructure.

Rather than rely on sophisticated devices, BikeVibes leverages a smartphone's accelerometer to gather road surface information, thus facilitating the crowdsourcing of this type of data. Using smartphones to gather road quality data has been previously explored [2–4], so we focus on the crowdsourcing and sharing of open data. The primary motivation for users to participate is that the open and anonymized data collected may be used to improve not only their own cycling experience but that of other users as well. In short, BikeVibes provides an easy-to-use and cost-free means to (1) crowdsource a city's *infrastructure sensing*, (2) provide insight into *human dynamics analysis* while (3) collecting data that can support decisions related to *smart city operations*, and (4) foster a city's *sustainability* in the sense that, directly or indirectly, it provides an incentive for people to use bicycles more often in their daily lives.

In order to maximize the potential benefits of crowdsensing, the data is made available under the principles of open data. In addition to governments using sensor data to improve bicycle paths, individuals or organizations may use it for different purposes. For example, those particularly sensitive to surface roughness, such as parents with trailers or food-delivery companies, can plan routes

[2]E.g., https://www.edmonton.ca/transportation/report_requests/report-a-pothole

that facilitate smoother rides. As well, information about ridership patterns can be used by governments to plan new bicycle paths.

Next, we discuss the underlying architecture of the app and website. Afterwards, we present a sample demonstration scenario and conclude by offering suggestions for future work.

2 ARCHITECTURE OVERVIEW

BikeVibes consists of (1) a smartphone app that allows users to collect and upload data and (2) a server-side application to store, display and share available data through the BikeVibes website. The mobile app is written in Java and the back-end application is written in Python using the Flask[3] framework and hosted on a cloud-based server. Figure 1 illustrates how the data is collected and processed.

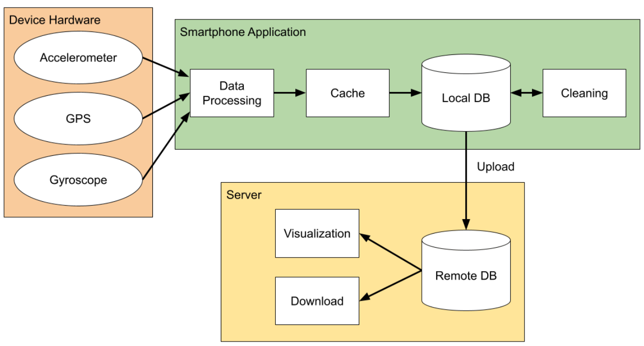

Figure 1: Application data flow

Firstly, the raw data is gathered from the device's hardware. Location information is determined from the phone's GPS receiver as latitude-longitude coordinates. A new location record is obtained once at least 5 seconds have passed and the user has travelled at least 10 metres from the previous location. The phone's accelerometer provides the instantaneous acceleration vector measured in m/s^2 relative to the device's orientation. The app implements a minimum sampling rate of 5 Hz.

We need to ensure that the acceleration data is oriented correctly. If the device has a gyroscope, Android can use this to determine the orientation. Otherwise, we can determine the device's orientation from the direction of gravity. Assuming a fairly stable orientation, the force of gravity is nearly constant; thus, we can apply a low-pass filter to the raw accelerometer data to isolate the low-frequency signal produced by gravity against the higher frequency vibrations. Once the orientation is determined, a rotation is applied to the acceleration vector so that the z-coordinate is perpendicular to the ground. The processed acceleration data and the location data are added to a cache and flushed to a local SQLite[4] database once that cache reaches maximum capacity. These records are stored alongside the timestamp (in milliseconds) at which they were recorded, as well as the index of the trip they were associated with. A new trip is started each time the user begins tracking. This prevents the data from separate rides from being misidentified as the same ride.

[3]https://flask.palletsprojects.com/en/2.2.x/
[4]https://www.sqlite.org/index.html

Each trip is also attributed to a user. While users should be able to view their individual data on the website, others should not be able to trace any data back to those users. Our solution is to store the data pseudonymously. When the data is uploaded, it is attached to a user ID. The user ID is a universally unique identifier[5] that is randomly generated when the user uploads data for the first time. All records produced by the user are stored alongside the user ID in the server's database. Importantly, the user ID is never revealed to anyone and cannot be traced back to an individual or a device. If the user wishes to view or query their own data, they may create a pseudonym (alias)—known only to them—in the application. This is tied to the user ID and can be used on the website to access a specific user's data. Additionally, users can delete their data from the remote server at any time from within the app.

In order to further protect the user's privacy, the app removes data around both the start and end of the trip, thus removing potentially sensitive locations such as the user's place of residence and/or work. We do that by considering a circle of radius r around the start and end points of a trip. We take the first time at which a recording was made outside of the starting circle and remove all data obtained before that time. Similarly, we delete all data from after the time when the last location was recorded outside of the ending circle. The value r can be configured by the user in the app.

The uploading process queries the stored, clean data from completed trips. Then the data is converted into JSON format and the user ID is attached before being sent to the server. When the data reaches the server, it is read from JSON and inserted onto another SQLite database.

Once the data is on the server, it may be viewed or queried from the BikeVibes website (discussed in the next section). When a user applies filters, the server queries the database with the appropriate parameters and returns the data to be displayed in the browser. It retrieves pairs of consecutive location records known as "segments", which can be drawn onto the map. They are coloured according to the root-mean-square (RMS) of vertical acceleration along that segment. According to [1], this is a "suitable key performance indicator of pavement conditions".

3 DEMONSTRATION SCENARIO

In this section, we will show how the mobile app is used and discuss its settings. We also demonstrate how the website is used. The data displayed below has been obtained from sample rides by the authors within Edmonton, Canada.

3.1 User Interface

Figure 2 shows the app's user interface. Beginning in the centre of the screen, we have the map visualization. This displays the data from a single trip, defaulting to the most recent one, as a series of coloured line segments overlaid on a map. Each line segment is coloured based on the root-mean-square (RMS) of vertical acceleration recorded between its endpoints. The colour gradient increases with the RMS from green to red with $0m/s^2$ mapping to green and $3.5m/s^2$ mapping to red. This provides a quick overview of the roughness of the ride to the user. When a trip is loaded, the map zooms in and centres on the trip that is being displayed. The user

[5]https://en.wikipedia.org/wiki/Universally_unique_identifier

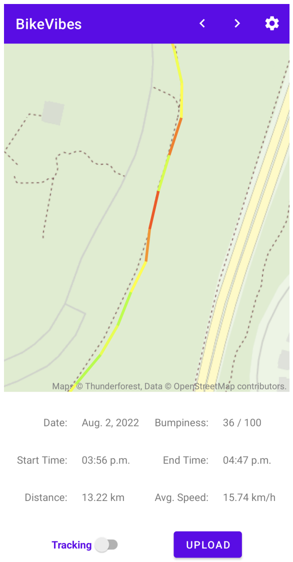

Figure 2: App Main Screen

can then zoom in/out and pan the map as they like. The map uses the OSMdroid[6] package with tiles from Thunderforest[7] and data from OpenStreetMap[8].

Below the map, the user is shown summary statistics for the current trip. This includes the date, start and end times, bumpiness score, distance, and average speed. The bumpiness score is determined by applying a log function to the average roughness (i.e. RMS of vertical acceleration) over all of the segments in the ride. This results in a score from 0 to 100 that can be used to compare rides. This section of the user interface is intended to provide users with information about their ride that they could find in a typical ride-tracking app.

The bottom of the user interface contains the tracking switch and upload button. These allow the user to perform the core functionality of the app. The tracking switch is used to start and stop tracking a ride. The app only records data when the switch is active, and it continues even if the user closes the app or turns off the screen. This allows the user to conserve battery life and open different apps while the trip is ongoing. To ensure the user understands that data is being recorded, even if the app is closed, a notification is displayed whenever tracking is enabled. The upload button sends data from all of the user's previous trips that have not yet been uploaded to the server.

Once a trip is completed, users are prompted to select the primary surface type over which they rode from one of "Pavement", "Dirt", or "Gravel". This is uploaded alongside the other data. This information makes it easier to identify segments that are rough relative to other surfaces of that type.

Finally, the arrows at the top of the screen allow the user to change which trip is currently displayed on the map and in the summary. The gear icon provides access to the settings menu.

[6]https://osmdroid.github.io/osmdroid/
[7]https://www.thunderforest.com/
[8]https://www.openstreetmap.org

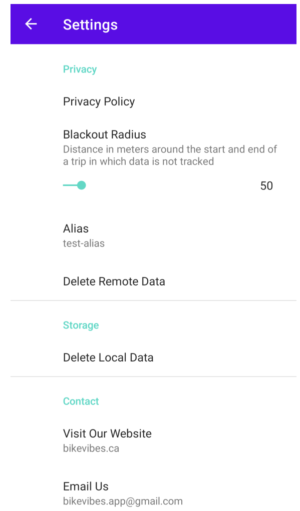

Figure 3: Settings Menu (partial view)

3.2 Settings Menu

Figure 3 displays the app's settings menu. It contains sections for privacy, storage, contact information, and acknowledgements. The privacy section contains the most important settings. The blackout radius slider allows the user to select a value from 0 to 500 metres to use as the blackout radius when obscuring sensitive data after a trip is stopped. This controls how much information around the start- and endpoints of a trip is removed, so that the user's commonly visited (and potentially sensitive) locations are not revealed. Tapping the alias setting allows users to create or change the pseudonym (which is private to the user) that is used to access their data on the website. As well, the privacy section includes a link to the privacy policy and a button that allows users to delete their data from the remote server at any time. Finally, the storage section allows users to free up space by deleting any data kept in the local database.

3.3 BikeVibes website and visualization

Figure 4 showcases the interface for viewing data on the website (acessible by clicking on 'map' at https://www.bikevibes.ca). The page displays the collected segments on a map, coloured by how rough the surface was, based on the RMS of vertical acceleration. The map interface uses the Leaflet[9] library. In the header, users can navigate to the About page or download the data. The sidebar also features a set of filters that allows the user to query the data based on alias, start and end date, time of day, and surface type. Unless an alias is specified, the data is combined from all of the users. (Recall that the user's aliases are not public, i.e., one user should not be able to purposefully see another users' data.) Choosing a surface also changes the color gradient so that it reflects roughness relative to the given surface type. This allows for greater differentiation which makes it easier to identify problem areas.

Figure 5 is a screenshot from the website when zooming in onto portions of two trips on pavement surfaces. The bottom illustrates the user cycling along the sidewalk heading east, whereas the top

[9]https://leafletjs.com/

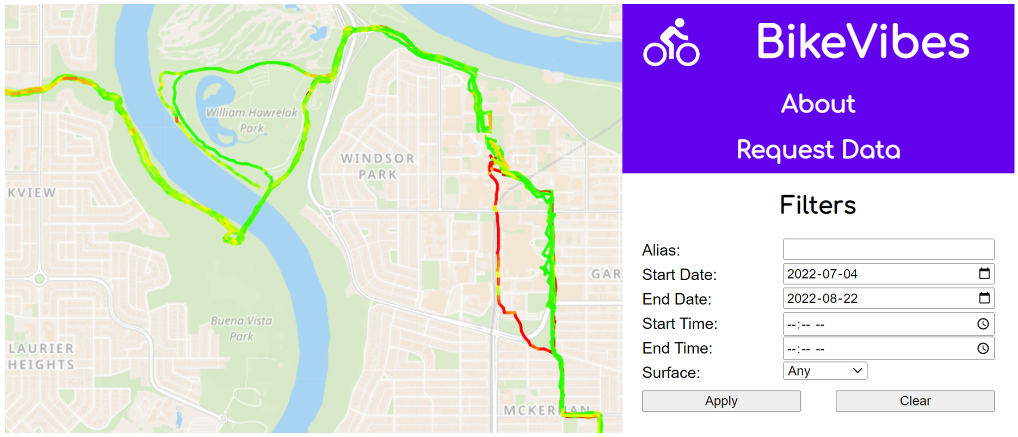

Figure 4: BikeVibes's data visualization on the web (partial view). Note the clear difference in some paths' quality (red vs. green)

was taken from the return trip along a side road heading west. The upper portion contains many segments that are coloured red or orange, indicating a higher degree of roughness, i.e., RMS of vertical acceleration ($1.5m/s^2 - 2.0m/s^2$). This is in stark contrast to the mostly green segments along the sidewalk to the south, which indicate a lower roughness (i.e. RMS between $0m/s^2$ – $0.5m/s^2$). Importantly, this accurately reflects the actual conditions of the pavement from both surfaces. The sidewalk in this scenario has lateral grooves separating the concrete sections which causes some slight roughness, but is otherwise in good condition. However, the side road has been poorly maintained and has many bumps, cracks, and potholes.

Having access to such a visualization allows users to make proactive decisions. For instance, with this information, a cyclist may decide to cross the road at the intersection to the east rather than ride on the rough side road. Alternatively, they may choose to ride on 95th Avenue's pavement when heading west. A city manager may also make conclusions from this information. They may suggest that the side road be repaired or that new bike lanes should be constructed along 95th Avenue. As well, the traffic patterns indicate that cyclists are crossing the street to continue on the opposite side of the road. This may suggest that the cycling conditions on the north side of the road are unsatisfactory, either due to safety or comfort concerns.

Finally, users can download the data to analyze it more thoroughly. The request data button leads to a page where users can select similar filters as on the visualization page to query the data by alias, date, time, or surface type. This is delivered in CSV format containing user ID, accelerometer, location, and surface type data. Note that only the anonymous user ID is attached to the data so that it cannot be attributed to any particular individual.

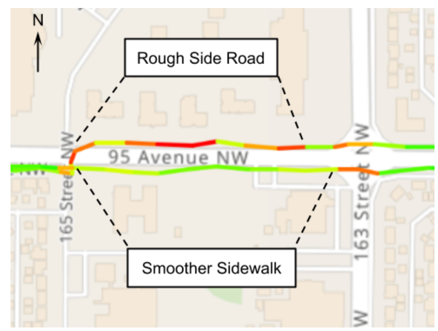

Figure 5: A comparison between two paths

4 CONCLUSIONS

We presented BikeVibes, a mobile application for collecting data about the roughness of paths from bicycle rides. Along with the accompanying website, it enables users to crowdsource data and allows individuals or organizations to use it for improving the quality of cycling infrastructure. There are many potential directions for future work. One example would be the development of smoothest, rather than distance-wise shortest, path algorithms. If such road roughness data is available, such an algorithm can be used to plan routes for cyclists who are sensitive to vibrations or bumps. Another direction would be to explore seasonal usage patterns, i.e., how the behaviour of cyclists varies by season or time of day in order to improve the city's cycling infrastructure in a more targeted manner.

ACKNOWLEDGEMENTS

Research supported by NSERC Canada. The web-server used by BikeVibes is graciously hosted by Cybera (https://www.cybera.ca/).

REFERENCES

[1] Cafiso, S., et al. Urban road pavements monitoring and assessment using bike and e-scooter as probe vehicles. Case Studies in Construction Materials 16 (2022), e00889.

[2] Lekshmipathy, J., Velayudhan, S., and Mathew, S. Effect of combining algorithms in smartphone based pothole detection. International Journal of Pavement Research and Technology 14, 1 (2021), 63–72.

[3] Seraj, F., et al. Rovi: Continuous transport infrastructure monitoring framework for preventive maintenance. In 2017 IEEE Intl. Conf. on pervasive computing and communications (PerCom) (2017), pp. 217–226.

[4] Yeganeh, F., et al. Validation of smartphone-based pavement roughness measures. AUT Journal of Civil Engineering 1, 2 (2017), 135–144.