A research about Acceleration Thresholds of Bumps and Holes on Various Road Surfaces Based on Raspberry Pi System

Abstract

This paper introduces an embedded system which is developed based on Raspberry Pi system. The system has the ability of collecting vibration data and position information by the integrated accelerometer and GPS module and it is developed for detecting bumps and holes in road surface by using bikes. The road surface is divided into 12 classes and 2 types of bike are applied to collect vibration data Also, this paper purposes a differential method to analyze the vibration data collected by the Raspberry Pi system which could minimize the mistake caused by the shifting of this system. According to the data analysis result, this paper obtains 24 thresholds of 24 corresponding combination of various road conditions and bike types which could be used to determine bumps and holes in the road more accurately and efficiently.

Keywords: Embedded System, Public Infrastructure, Sensor Communication

1. Introduction

From the nineteenth century to the present, bicycles have been widely used. It can be used as an environmentally friendly vehicle. Benefits of bikes include flexible mobility, reduced road congestion, reduced emissions, and reduced fuel use. Also, with the development of civilization, the city's infrastructure is aging fast. This aging is more and more affecting the livers' daily life, so it is important to detect and evaluate the city infrastructure conditions like road surface flatness and so on so forth. Nowadays, the general tools to evaluate the road condition are cars integrated with cameras. However, in the real application, it is not only expensive, but also has low adaptability to various types of roads including grass ground, dirt ground and concrete ground. Bikes have the unbeatable adaptability in the field of environment and infrastructure evaluation because they don't have the two main disadvantages of cars and can be more feasible in the real application. Some former researchers developed embedded systems in this field to collect environmental data like PM2.5[4] and public infrastructure data like public bike usage data [3][4][5][6]. According to this collected data, the government and company can make better strategy in environmental policy or public bike distribution in the city. To be more specific, as for the road condition detection, previous researchers developed several types of systems which are based on different platform and combination of other tools like GIS system and google maps.[1][2] These systems collected vibration data and use apply various processing methods designed by their developers to the data to evaluate the road surface conditions. However, these systems don't offer us a table of thresholds which could be used in various road conditions but give a general threshold for all kinds of road. Thus, the system and data processing method we used is designed to obtain this table for the more accurate and efficient evaluation of the road conditions.

Our project is called Instrumented Bike Share project, which has been part of the capstone series for the past two semesters. In fact, some parts of the project has been partially completed by the previous teams. Previous capstone teams have basically formed the basic module of data collecting and processing. Based on the common road types, we (The Smart Wheel team) as a group of 4, have taken up measurements on 4 types of road, 3 types of weather conditions and 2 types of bike. With the idea in mind, we regenerated a relatively effective low-cost system with high accuracy to detect the road conditions of the respective type. Hence, the system formed previously will hopefully be improved and implemented on the public bikes at Northern Arizona University. Although the project is partially completed by previous teams, the Smart Wheel team will work on the improvements and start up an innovation that will help the system run with higher efficiency.

Figure 1: Installation of the embedded system.

The basic features of our system include a Raspberry Pi system which act as the microcontroller, a solar powered battery to power up the system, an accelerometer and GPS which detect the location and road conditions when biking, and a camera which capture images of the road condition (Ex: Bumps, potholes, etc). These little parts combined and form a system device which will be installed on bikes for analysis. Basically, the device is built out of a hardware module for recording and processing data and the Android App to share and display data. All the necessary operations and maintenance of the entire system is highly dependent on the Raspberry Pi itself except for the Android App.

2. System Structure

2.1Wireless Communication and Connection

The Wireless communication system is basically a communication platform between the sensor device and the Raspberry Pi. Hence, the reason the platform is implemented is to allow a strong and smooth transition between the system and the Raspberry Pi system. Zooming in into the device, the GPS module has the role to track down the location during testing phase or operating phase where applicable. Specifically, it allows the system to know the location of the bumps and so that the accelerometer can perform its module by analyzing the accelerating data. The compatibleness of these parts is the formation of the wireless communication platform which is an important factor to our entire project as we need clear information from the devices.

The module of the device is basically a development of a sensor data logger, which allows the tracking of any road conditions and data needed to perform analysis. Hence, the clear communication between all the parts are necessary in order to proceed to the next subsystem.

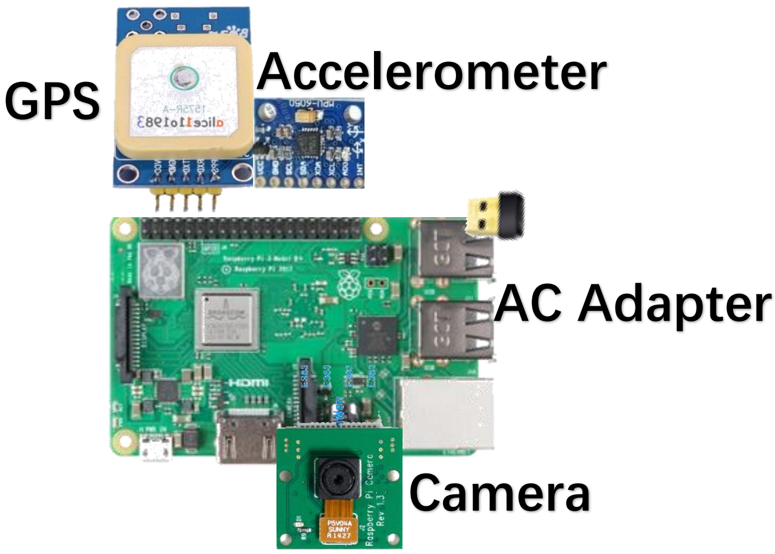


Figure 2: The Raspberry Pi system and sensor device

2.2 Data Collection Module

Our data collection module is basically collecting sets of data to prepare for analysis of the dynamic threshold and terrain algorithm. It is set up in a way that it is based on both road and weather conditions. There are 4 main types of road conditions that we have considered as mentioned before, they are concrete, asphalt, grass, and dirt. On top of that, we take into consideration for each of these conditions with the weather conditions such as dry, wet, and snowy. In this module, the main goal is to collect data based on the weather conditions for all types of road conditions stated. Once the data is all collected, we then analyze them based on the details and eventually set a threshold for the respective conditions.

```
0. 606 0. 32 -111. 656511 35. 178354
1. 611 0. 36 -111. 656511 35. 178354
2. 612 0. 34 -111. 656511 35. 178354
3. 612 0. 48 -111. 656511 35. 178348
4. 608 0. 55 -111. 656510 35. 178339
5. 607 0. 50 -111. 656507 35. 178327
6. 596 0. 59 -111. 656505 35. 178312
7. 596 0. 57 -111. 656506 35. 178300
8. 609 0. 40 -111. 656507 35. 178288
9. 606 0. 56 -111. 656512 35. 178279
10. 612 0. 41 -111. 656511 35. 178264
11. 633 0. 61 -111. 656510 35. 178251
12. 613 0. 41 -111. 656506 35. 178235
13. 620 0. 15 -111. 656505 35. 178220
```

Figure 3: Sample data collected

```
138. 084 0. 96 -111. 657116 35. 178115 Moderate
191. 08 0. 82 -111. 65781 35. 178502 Moderate
372. 092 0. 68 -111. 656302 35. 179919 Moderate
393. 093 0. 73 -111. 656478 35. 180173 Severe
432. 087 1. 19 -111. 656398 35. 180295 Moderate
526. 088 0. 79 -111. 65588 35. 179053 Moderate
```

Figure 4: Analyzed data sample

Based on the figures above, a set of data is collected for the first 13 seconds of testing. The GPS location is shown in an address form together with the vibration data of bumps/potholes. With the data collected, our team were able to analyze the data to eventually decide on the threshold of the bumps/potholes. In fact, the module can determine whether the road conditions at the data is moderate, normal, or severe. Just for your information, these data are collected for two types of bikes which are road and mountain bikes, which is why most of the time spent for the project is on this module.

2.3 Road and Bike Conditions Difference

As we proceed to our respective data collection and analysis modules, we took into consideration and anticipated the different results that will be achieved with two different types of bikes. The glitches and effects of these two bikes on the 4 types of road conditions (Concrete, Asphalt, Grass, Dirt) provide a huge difference in the data analysis module since the weight and the difference in the mechanism of the bikes. As complicated as it sounds, the weather conditions are also one of the factors we consider during our testing phase. Fortunately, some of the conditions such as snow grass road and snow dirt road provide relatively similar data as both surfaces have a similar effect when covered by snow, according to our testing simulation result. Although some of the results can be acquired easily due to their similarities, our team are still required to consider all the 3 weather conditions on 4 road conditions, together with the 2 type of bikes as the mix combination of the result from the requirement can be different once tested.

Figure 5: Road bike display

Figure 6: Mountain bike display

3. Data Analysis

3.1 Classification

By using the system mentioned in the previous part, we collected a large set of data of different conditions of infrastructure. The classes of infrastructure are shown in the table 1 below:

Asphalt		Concrete		Grass		Dirt		Texture
Street	Mountain	Street	Mountain	Street	Mountain	Street	Mountain	Bike / Condition
								Dry
								Wet
								Snowy

Analyzed result — Thresholds.

Table 1-Division of different road conditions, weathers and bike types.

Totally, we have 24 thresholds needed to be determined. These 24 classes include 3 road conditions, which are dry, wet and snowy, 4 road materials, that are asphalt, concrete, grass and dirt, and 2 bike types, that are mountain bike and street bike. The system is applied to collect vibration data in these 24 classes of road and bike. Then we analyze the collected data by using MATLAB program and finally obtain the corresponding thresholds. Figure 7 shows several road surface samples that we have made test on.

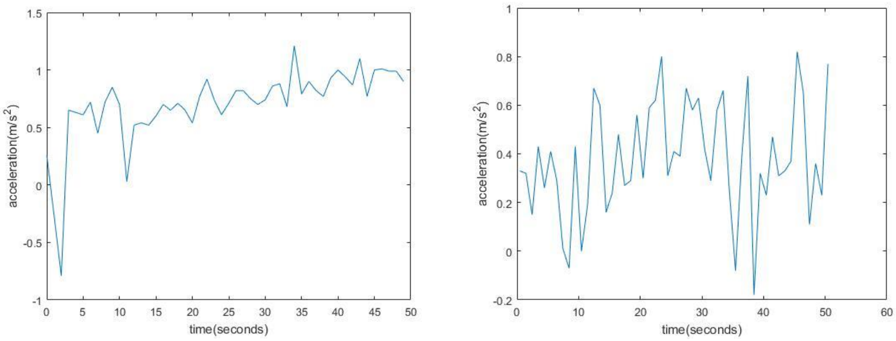
(a) (b) (c) (d)

Figure 7: Images on roads with different conditions: (a) Dry Asphalt (b)Snowy Grass (c) Wet Concrete (d) Dry Dirt

Some scene pictures

3.2 Data Processing and Analysis

Our system was developed 1 year ago and the data collected by it exists shifting and mistakes because of the long storage time. One example of the collected raw data is shown in the figure 8 below.



(a) (b)

Figure 8: Unprocessed vibration data under (a) Mountain Bike-Dry-Asphalt class and (b) Street-Snow-Grass class

According to these two example figures, we can see the vibration clearly not in the large scale but in the small one and the mean of the collected data is changing as time goes by. That is the feature shows the system existing shifting and this shifting is sensitive to the time when the system is working. By analyzing the relation between the shifting and working time, we found that these two factors are not related mathematically and the shifting is random in some degree, but the determined thing is that the mean will change as time goes by. Thus, if we could limit the scale of time to consider the vibration, we could eliminate the mistake caused by the shifting and get more accurate data. Thus, we applied the differential analysis to the data. Differential analysis which is actually a length-2 sliding window method[1] means we don't care about the exact value of acceleration but focus on the difference between two adjacent data. That is,

$$Dif_i = R_i - R_{I-1}$$

,where Dif_i is the ith data point of differential data

R_i is the ith data point of the raw data

Eq.1: Differential Processing Method

By applying this differential analysis, the random shifting's affect is limited in the time period of 1 seconds and the mistake is limited, too. Thus, the processed data can show the vibration's feature more accurately and precisely. Figure 6 is the figure of corresponding processed data of the figure 5 (a).

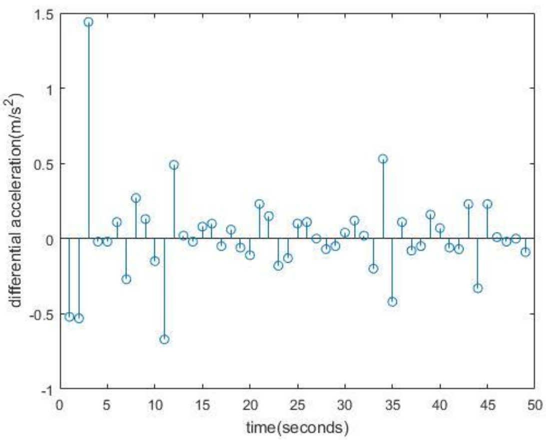


Figure 9: Processed differential data figure under Mountain-Dry-Asphalt class

In figure 9, the data shows the changing of the vibration data, which can be called "vibration of vibration". According to these data points, some outliers are shown clearly.

For each bump and hole, the module of the processed data is predetermined. Both bumps and hole will lead to a maximum and a minimum in the collected data, which separately corresponds to a positive value and a negative value. To be more specific, a bump will have a minimum following a maximum in the collected data and a hole will have a maximum following a minimum in the collected data. In the processed differential data, a bump will have a negative value following a positive value in the collected data and a hole will have a positive value following a negative value in the collected data. On the other hand, generally speaking, every surface of the road is not perfect. That means the accelerometer will always have small vibration data which shows the existence of small bumps and holes. These small bumps and holes will not affect the feeling of biker and will not be counted as obvious bumps or holes. On the other hand, according to the result of Figure 6, some data points have obviously higher absolute value than most of data points. These data points are “obvious bumps and holes” and we could determine the thresholds by considering these data points’ value.

In order to determine the accurate, effective threshold, we need to make sure that the threshold will identify ‘obvious bumps and holes’ accurately. Statistically, expectation or average could show the general feature of the group of data. Thus, we used the average of the all processed differential data as thresholds’ basis and times a factor of 1.5 to make sure the accuracy of the determine threshold. Figure 9 shows the corresponding threshold line of the Mountain-Dry-Asphalt class.

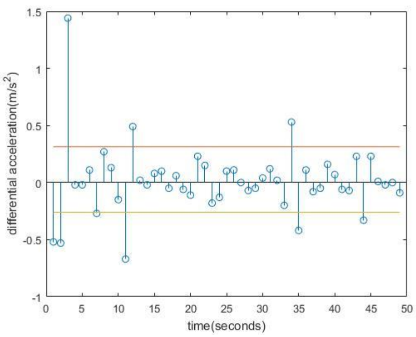


Figure 10: Threshold determination of Processed differential data figure under Mountain-Dry-Asphalt class

In figure 10, two plotted straight lines are separately corresponding positive threshold and negative threshold. These thresholds are obtained by the following Eq.2

$$Positive\ Threshold = 1.5 \times average(dif_i, if\ dif_i > 0)$$
$$Negative\ Threshold = 1.5 \times average(dif_i, if\ dif_i < 0)$$

Eq.2: Thresholds Determination Equations

According to the result of calculation, it is clear in figure 7 that each outlier is identified clearly. Also, there are some data samples that are close to borders but they are not obvious outliers. Actually, in the real application, the corresponding threshold can be softened and the corresponding soften threshold can be used to determine the bumps and holes' degree of severe by calculating the distance from data sample to the threshold. The thresholds given by this method is a representative value. In the real application, it can be utilized optimally and more functorial. The positive and negative thresholds build a safe zone between them. The collected data between these two thresholds are determined as normal road surface and the outliers are bumps and holes on the road.

To make the thresholds more general, the final thresholds' value takes the average of the sum of the absolute values of positive threshold and negative threshold, by the eq. 3:

$$Absolute\ Threshold = average(|Positive\ Threshold| + |Negative\ Threshold|)$$

Eq.3 Final Thresholds Determination Equation

By apply Eq.3, the final absolute thresholds are determined and shown in the table 2:

	Asphalt		Concrete		Dirt		Grass	
	Mountain	Street	Mountain	Street	Mountain	Street	Mountain	Street
Dry	0.2872	0.3194	0.2900	0.3002	0.3600	0.2581	0.3247	0.3836
Wet	0.2564	0.2836	0.2300	0.2572	0.3327	0.2029	0.2501	0.3578
Snowy	0.5864	0.4004	0.4884	0.4425	0.2859	0.2538	0.3659	0.3679

Table 2: Final result of thresholds determination

3.3 Result Analysis

According to the threshold's determination table above, we could make some conclusions. Firstly, we could see that for asphalt and concrete surface, the thresholds under wet condition are generally slightly lower than the thresholds under dry condition. Secondly, the thresholds under snowy condition are obviously higher than the thresholds under dry condition.

According to the table, in Asphalt Concrete and Grass conditions, generally speaking, the thresholds of the mountain bike is lower than the thresholds of the street bike. That can be explained by the difference of the vibration absorption systems and types of tire of two bikes. For the mountain bike, its vibration absorption system is more powerful than the street bike and its tire is also wider than the street bike's tire. These differences will make the mountain bike has less vibration when it meets holes and bumps in the road. On the contrary, in the dirt roads, the mountain bike's data is higher than the street bike's result. Well, since the dirt road has loose structure, during the process of biking, the street bike will originally fall into the surface but the mountain bike will not due to its wide tire. Thus, when the street bike meet bumps and holes in the dirt road, its thin tire will "cut" these bumps and holes in some degree and has lower vibrations.

Here is a summary of general trends we found from the table:

- the thresholds under wet condition are generally slightly lower than the thresholds under dry condition.
- the thresholds under snowy condition are obviously higher than the thresholds under dry condition.
- in Asphalt Concrete and Grass conditions, the thresholds of the mountain bike are lower than the thresholds of the street bike.
- in the dirt roads, the mountain bike's thresholds are higher than the street bike's thresholds.

4. Android Mobile Application

To let the system easier to use we developed an Android application and built a web server. The android application is used to transition data from the Raspberry Pi to the web server, and it also can send orders to the Raspberry Pi to control the system. The web server is used to save all the data we collect and show all the bumps on the map

We applied Android Studio to complete the design of our app design. And the Android application is consisting of three main interfaces, including the main interface, setting interface, and the toolbar.

In the login interface, the user could log in by scan the QR code on each system. In the main interface, the user could see every bump the detected clearly. In the toolbar there are five buttons Map, Connect, Start, Stop and Analyze. Connect, Start, Stop and Analyze is used to send orders to the Raspberry Pi to control the whole system Connect is used to build a connection between Raspberry Pi and Android device, Start is letting system start to collect data, and Stop is letting system stop to collect data. After data has been collected, the Analyze command will let the system to start analyzing data. And the Map button will show the map, which is on our server and marks all the bumps and holes detected by the system. In setting interface, users could receive data from the Raspberry Pi or send the received data to the webserver.

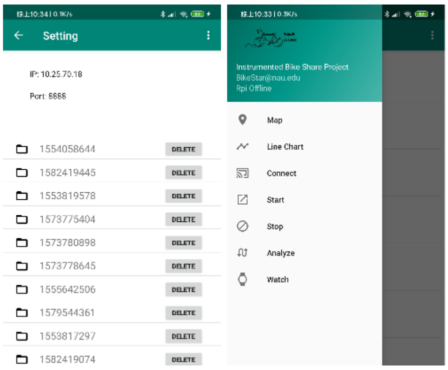


Fig.11 setting interface (left) and toolbar(right)

Here is how the Android application works. After the user starts running the script runPi.py, the device will listen to port 8888 and wait for TCP connection with the smartphone. If the connection is established, then the device will wait for commands to perform corresponding operations. However, all the commands should come in order. First, the device should receive “start” to start the threads of recording video, positions and accelerating values, and the corresponding Python scripts are gyro.py for recording accelerating values, and code for recording the video and positions are already in the entry script runPi.py. Then, if the device receives “stop”, the threads will be terminated. Also, after the device receives “analyze”, the device can run the script extract_pic.py, detect_peaks.py, and label_image.py. After all the processes above, in the folder

“/home/pi/BikeData” on the raspberry pi will show a folder with a timestamp like “1536428326”, a movie file “motion.h264” and a text file “raw.txt”. Inside the time folder, it will show you several extracted pictures and a text file “processed.txt” showing recorded time from the start of running the system, accelerating value, positions and image labels. These folders, as well as files, can be transferred to the phone via FTP after a user tapping “Download” in the setting page of the app, also if the user tapping “Upload” txt file “processed.txt” will upload to the server via FTP. The workflow of the system is shown in the flowchart below.

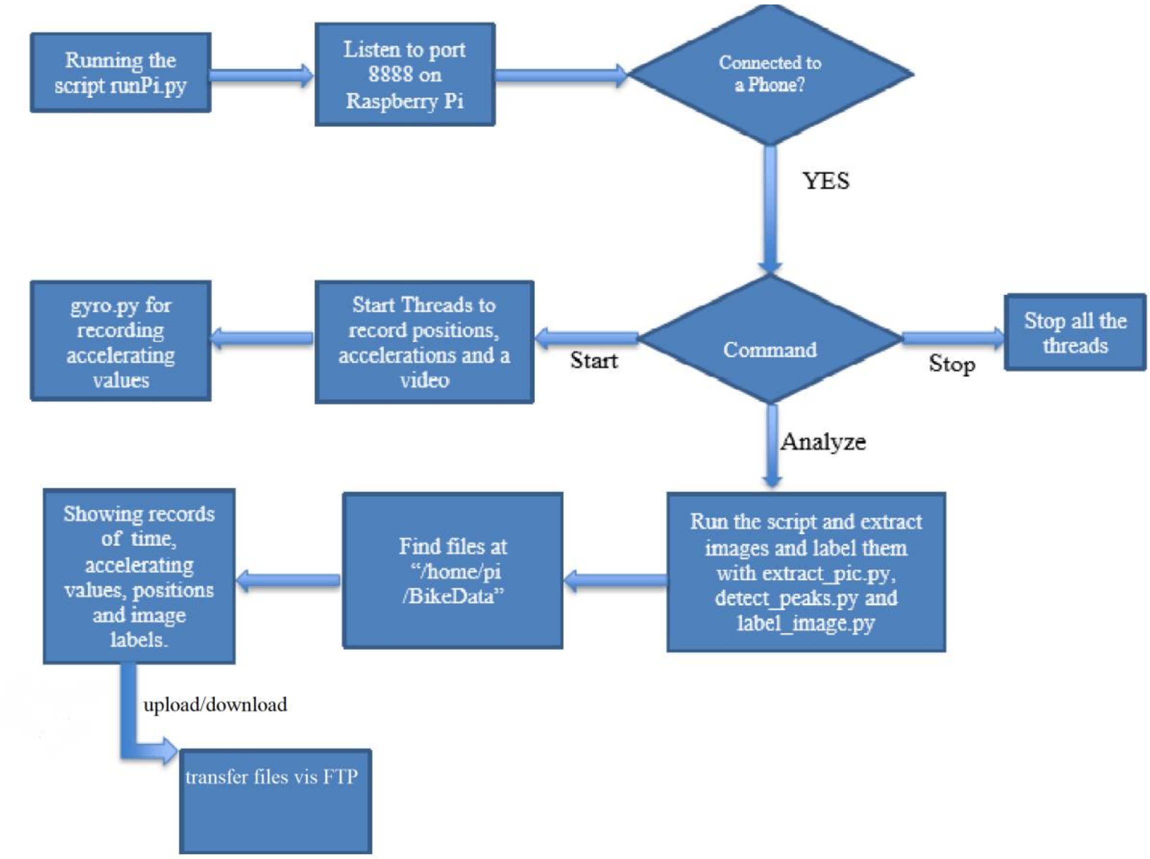


Fig.11 Flowchart

The web servers are used to save the information of the bump point and mark all the bump point on a map. The text file is not a very good file format to store or process geography data, but if the file is GeoJSON format, it will be very easy to use. So, before the data store and process, the txt file will be transferred to GeoJSON format by a Python script. Because the “processed.txt” saved the processed data into five-column: the first is the timestamp, the second column is data of accelerating, the third column is longitude, the fourth column is the latitude and the last one is level of the dump and every column is departed by Space. So, the Python script will intercept the data of accelerating and the level of the dump and store it as "properties" and intercept the data of longitude, latitude store is as "coordinates", then the processed file will save as “processed.js”. Then the web page will process those data easily and mark them on the map.

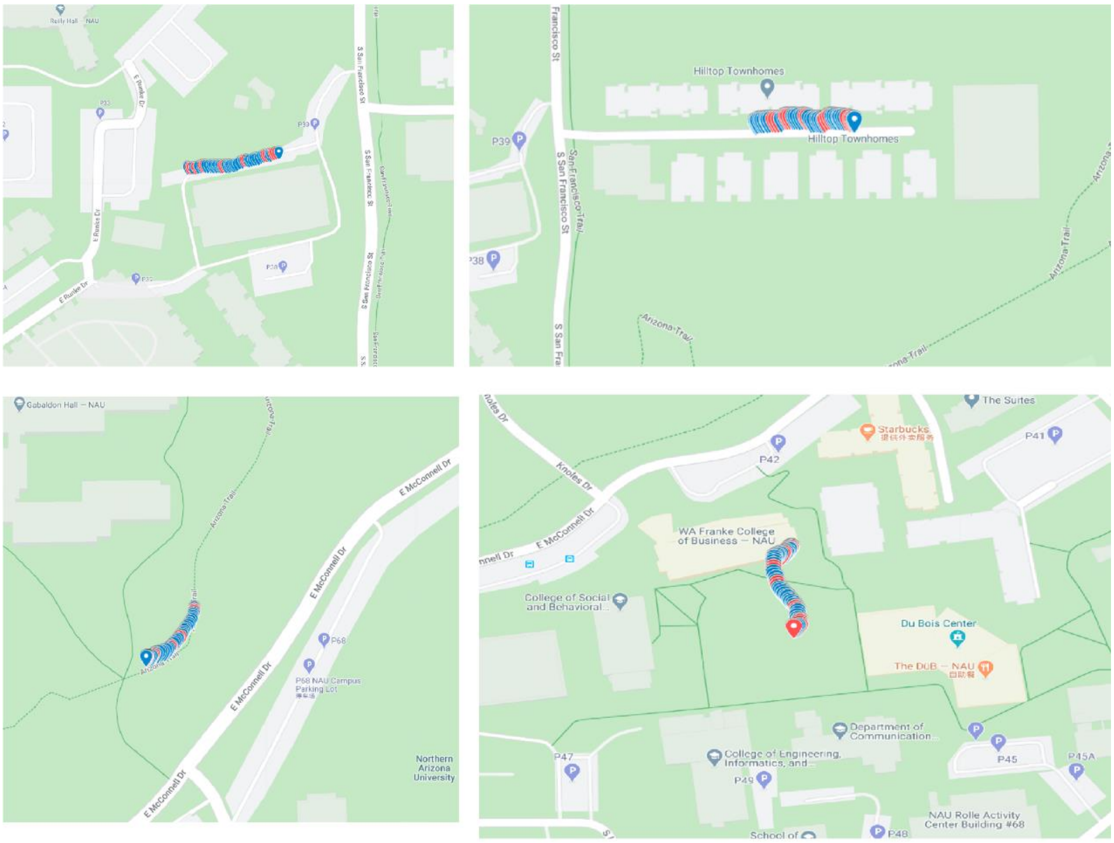


Fig.12 example of the data after process

5. Conclusion

All in all, we utilized an embedded system which is based on raspberry pi and integrates sensors including an accelerometer and a GPS module to collect vibration data of the infrastructure. The infrastructure is divided into 12 different road conditions and we collect the corresponding data for thresholds determination. In data analysis module, we used the differential method to minimize the mistake made by the drift of the sensor. By applying this method, we obtained 24 different positive thresholds and 24 negative thresholds and finally 24 absolute thresholds. These 24 thresholds indicate the difference of various road conditions and types of bike. In the reality, these thresholds can cooperate with the terrain recognition algorithm with machine learning method to identify the bumps and holes in the roads more accurately. In the future, we will develop the corresponding machine learning algorithm and make the system more perfectly.

Reference

[1] P. Qiu, X. Liu, S. Wen, Y. Zhang, K. N. Winfree and C. Ho, "The Development of an IoT Instrumented Bike: for Assessment of Road and Bike Trail Conditions," *2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI)*, Shanghai, 2018, pp. 1-6.

[2] Ho, Chun-Hsing, Slim, Gerjen, Monahan, Shannon, DeGeyter, Jeremy ”Application of Geographic Information Systems and Vibration Mobile Apps in Road Condition Assessment of Bike Trails” Transportation Research Board 95th Annual Meeting 2016, Washington DC, NW, 16-1424.

[3] Wang and C. Wang, "Analyzing Bike Repositioning Strategies Based on Simulations for Public Bike Sharing Systems: Simulating Bike Repositioning Strategies for Bike Sharing Systems," 2013 Second IIAI International Conference on Advanced Applied Informatics, Los Alamitos, CA, 2013, pp. 306-311.

[4] Lai, P., Huang, H., Sheu, M., Wu, C., Le, J. and Chen, T. (2018). Bike Sensor System Design for Safety and Healthy Riding - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8448461

[5] Yi, H. and Nie, Z. (2017). Mobility Innovation through an Efficient Mobile System for Bike Sharing on Campus - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8842744

[6] Luo, R. and Shen, Y. (2009). The Design and Implementation of Public Bike Information System Based on Google Maps - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/5199859