Parking Space Finder App

Chuanglin Cui ccui5@binghamton.edu
Mengqiu Zheng mzheng47@binghamton.edu

ABSTRACT
As the grows of cars, it is increasingly difficult to find a parking place in our campus. Our parking space finer App finds a good solution to this problem by sharing user’s parking result and leaving parking lots’ information to all users, which create a way to show the parking place’s situation to all users, and help them saving time to find a parking lot.

Keywords: Google map, activity detective, GPS, Multiple threads

I. INTRODUCTION
As the number of cars grows, parking spaces are becoming tighter and tighter. It usually takes us a lot of time to find an available parking lot, so we want to design an App to help us drivers saving time to find a parking place.

Our Application’s goal is to show our drivers the situation of the parking lots around him, like full or empty, and then let the driver to make his decision to park where he wants.

The idea of generating parking lots’ situation is based on detecting users’ activity: If a user in this parking lot can find a place or not. If we can find a driver is leaving from the parking lot. And we will update the information we detected from all the users’ activity to our server’s database, then we can generate a map of all parking lots’ situation, and send that back to our users in every period of time.

We get an pretty successful result in our field experiment, and we believe the accuracy of parking map we sent to user would be improve as the number of users increases.

II. ARCHITECTURE DESIGN
The basic architecture of our design is sharing the parking lots situation between users and our server’s database.

Parking lots situation could be full or empty. To decide if a parking lots situation we need to operate a paring lot in to serval points (parking points). And track the users parking route by detect the parking points he passed. (figure 1 shows parking map)

Another information we get from users is their activity. For example, we can detect a person is waking by knowing the accelerate value and speed. Therefore we get that information from smart phone’s senior, and decide the user’s activity with certain specific algorithms.

With the activity we detect we can decide our users’ parking result:
Situation 1:
In the parking area:
IN_VEHICAL -> ON_FOOT
In the parking area:
Result: parking success -> parking place available

Situation 2:
In the parking area:
IN_VEHICAL
Out of the parking area
Result: Parking failed -> parking place unavailable

Situation 2:
In the parking area:
ON_FOOT -> IN_VEHICAL
Out of the parking area
Result: Leave -> parking place available

As we detected user’s route and activity, we need to update what exactly information we detect to server’s database. And at the same time, we need to load the latest parking map from server. (figure 2 show the entire logic works)
III. IMPLEMENTATION

We develop the Application with android Studio. And we create a server to save our database. While the App running, here are Multiple threads Synchronously: Read/Load server thread, Activity recognition and Location change thread.

Read/Load server thread is used for Update, load and add data operations, it serves as parking map renew work and send users’ parking activity result to server.

Location change thread: recognize every step the user move, and with the exact location we need to recognize it to the corresponding parking points or gate points we built.

Recognize activity works every 17 seconds (on foot, in vehicle). To reduce the consumption, we only read small window size data.

We do all the logic Algorithms in our main function.

In the server terminal, we has three sheets, which save the information (location and statue features) of parking points, gates points, and activity.

IV. EVALUATION

As the app start, users will locate to his current location on the map, and all parking points and gate points shows to user.

As our App designed users can find the situation of a parking lot by the color of gate points, red stands for empty and blue stands for full, users can also get more information by touching the gate point, if it is full it will return message “full” (figure3), and return “it may still have space” if empty. (figure4)

V. CONCLUSION

Our app turned out positive result in field experiment, and it can greatly help users to find an available parking place. Besides, we believe as the number of our user grows, the accuracy of parking map could be significant improved.

Honestly, we also need to mention a few problems of our App need to fixed in future work:
As it uses google API to locate user’s location, we need to get a new API-key every time when build it on a new user.
GPS signal does not work well in cloudy weather, and it cause mistake in location detect and activity recognize.
Every time (17 second) we recognize user’s activity, we need to restart the app, because of some unfixable bugs, it make a little bad user experience.