CUbiPark: Sensing Parking Availability

Ali Soong
Cornell University
Ithaca, NY
acs322@cornell.edu

Jonathan Lee
Cornell University
Ithaca, NY
jwl274@cornell.edu

ABSTRACT

Finding parking can be frustrating, enraging, and tedious. We introduce CUbiPark a mobile Android application designed to passively sense user behavior while driving and interpret parking availability within a given parking lot. Using accelerometer, location, and activity recognition data from just one driver, CUbiPark classifies and labels the parking availability within the lot for future drivers. Applied and tested on Cornell University's campus with a select number of parking lots, the application shows promise in accurately providing parking lot availability information to users. Crowdsourced data collection combined with accurate interpretations of sensor data can provide a no-hassle and no-effort means of determining parking availability in a lot.

INTRODUCTION

Everyone has suffered from the pain of having to go from parking lot to parking lot looking for a place to park. Whether it is a sporting event, move-in week, or just any busy time of day, there is always the lingering possibility of not being able to find a parking spot. If there was a way for drivers to know an estimated availability of parking lots around them, they could better make decisions concerning where to find parking as well as alternative parking locations if their desired location was full, saving them the time and effort it takes to physically search on their own. In this paper, we introduce CUbiPark, a mobile Android application designed to accomplish just that.

CUbiPark uses a host of smartphone sensors to collect, analyze, and interpret data concerning a user's activities as they enter and exit a parking lot. Accelerometer, location, and Google's Activity Recognition API are crucial sensor data that ultimately help characterize the availability of parking lots surrounding a user.

This paper serves to illustrate the design considerations, development, and analysis surrounding the CUbiPark application. We first introduce related work surrounding parking sensing as well as initial considerations and motivating factors in the development of the CUbiPark system. Subsequently, we discuss the actual development, application, and decision tree logic, which CUbiPark uses to determine the parking availability within the lot. Lastly, we define initial results, design of future user studies, and evaluation techniques to further improve on the accuracy and to establish the efficacy of the CUbiPark system.

This paper not only serves to discuss the implementation of our application but also addresses the capability of modern smartphone sensing to recognize key activities and features pertinent to parking and driving and how those can be used to estimate availability within a parking lot.

RELATED WORK

While there has been previous work done involving the sensing and monitoring of parking behaviors, the solutions are not fully ubiquitous. Currently, there are physical infrastructure systems for detecting parking lot availability but they are incredibly expensive and require a lot of additional devices to be installed. Other options require users to manually input data, making them time consuming and less desirable [1].

More and more mobile applications are constantly being developed as the ubiquity of smartphones continues to increase. However, many apps that seek to provide useful information for drivers to utilize while parking either just present already publicly available data or do not actually offer parking lot availability in real time.

SPARK is a VANET-based parking scheme that is intended to survey an entire parking lot and provide drivers with real-time parking navigation [2]. However, this requires additional infrastructure, limiting its widespread use.

ParkNet [3] and Parksense [4] are other systems that also attempt to estimate parking availability, but they both focus on street parking rather than parking in designated lots. ParkNet uses a passenger-side-face ultrasonic rangefinder that is attached to a vehicle in order to determine the parking spots' availability. This solution would require equipping numerous cars with the device in order to get accurate parking information. Parksense on the other hand makes use of WiFi beacons to detect and monitor the availability of on street parking, but this limits the range of detection to areas within WiFi range.

MOTIVATING DESIGN AND CONSIDERATIONS

While alleviating the frustration and difficulty of finding parking was our primarily motivation, we had a number of design motivations that also drove the development of our application.

Passive Sensing

A successful implementation of an application for parking availability should not require any user input or effort. Otherwise, the application could simply ask users how many spaces were available upon arriving. CUbiPark does rely on predetermined parking spot locations but does not require any user input to collect, analyze, or update data within the application.

Single Driver Assessment

CUbiPark should only require a single driver's drive through a parking lot to provide an estimation of parking availability. We discuss more in our Future Work section how future drivers can iteratively improve estimations of availability.

However, for our implementation only a single driver is required to provide an estimate of availability.

METHODS

In order to figure out what distinguishes driving behavior under different parking lot availability conditions, we first gathered initial data from several different parking lots under these various conditions. We built a preliminary application to collect GPS, accelerometer, and activity recognition data. This provided us with "ground truth" data to analyze and extract the key features that differentiates how a person drives, and therefore how the car moves, when parking availability varies.

We identified the activity of the user when entering and existing the parking lot, looping, parking distance from the point of interest, and stops as key features for determining availability. These features proved to be easy to track, distinguishable from other driving behaviors, and most deterministic of parking lot availability. Using the information we collected, we built an algorithm to take in the raw data, detect these key features, and then classify the lots' parking spot availability. We then had users drive to and attempt to park in the Sage Hall parking lot on campus at different times of the day to get a range of parking conditions. The Sage Hall parking lot was ultimately chosen because of its simplicity and varying activity throughout the day. Conducting trials and user testing in this parking lot would offer easily collected and testable data.

After collecting this data, we ran our algorithm to see how well our classification matched the users' description of the parking availability. We then made some adjustments and updates to our program before running another round of user tests. Finally we evaluated our results by comparing how well our program detected various features and if our final classification matched users' reports. We go on discuss these findings as well as our iterative phases of testing in further detail later in this paper.

APPLICATION DEVELOPMENT

The CUbiPark application incorporates parking lot markers into a simple Google Map interface in order to display parking lot information to users. From the main screen, users have the capability to individually select parking lots, which display information concerning the name of the parking lot, the total number of spots available in the lot, its status, and the date and time in which it was last updated.

While we developed a prototype for the purpose of user testing, the application and its user experience and interface acted as proof of concept for a future consumer application. The application, picture below, is intuitive and simple for users to use.

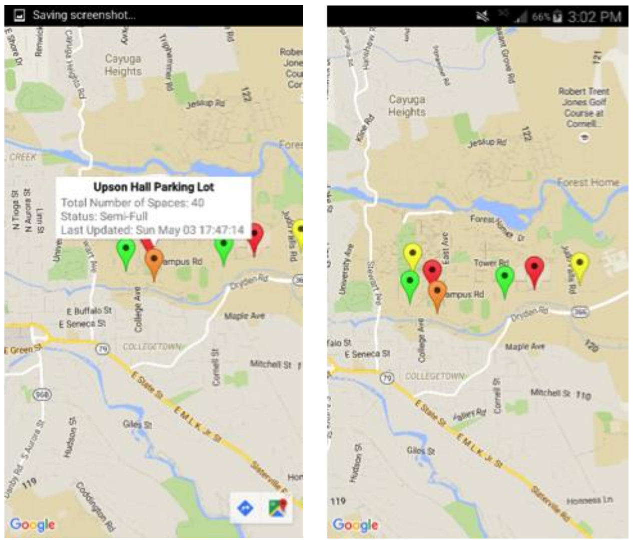


Figure 1: Main Application Screen

A potential future consumer version of CUbiPark would also include a number of additional features such as the ability to manually add parking lots to the map, provide driving directions to a selected parking lot, and have the option to disable the collection and analysis of a users' sensor data to ensure privacy.

Feature Detection

We go on to discuss the specific features we detect within our application. In total, there are five features that the application leverages in order to determine parking lot availability. They all rely on available sensor data from the accelerometer and GPS of an Android smartphone.

Entering and Exiting a Parking Lot

Android provides capabilities to account for users' current location with an awareness of their proximity to areas of interest. Geofences are application specific and defined areas that alert the application when users have entered or exited an area.

CUbiPark leverages Geofence monitoring to passively and with low battery consumption detect the entrance and exit of a parking lot by a user. Every Geofence has a predetermined latitude, longitude, and radius (defined in meters). For the purposes of the initial deployment of our application, parking lot information for Geofence creation were manually inputted into the Android application. A radius of 100 meters was used.

We recognize that this practice of Geofence monitoring as a tool for determining entrance and exit is not foolproof. While this solution works perfectly fine for square parking lots that can be enclosed in a circular perimeter, a long horizontal parking lot for example poses significant issue.

An alternative solution that we have explored for future implementation is to use lists of latitude and longitude to delineate the shape of a parking lot. If a user's location drifts outside this area, the application will know a user has left the parking lot.

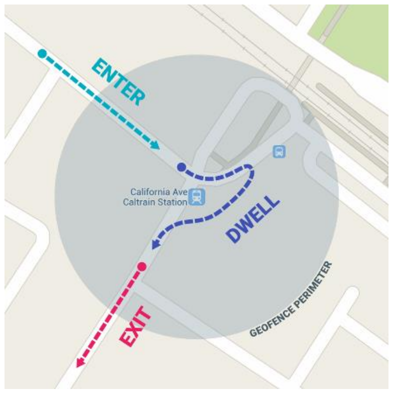


Figure 2: Example Android Geofence Diagram

Geofence monitoring allows CUbiPark to work "quietly" in the background until a user enters a parking lot. Once he or she does, the application knows to begin collecting and processing data.

Number of stops

The number of stops a driver makes while finding a parking spot can help indicate that the driver is actively searching for a spot, meaning that one is not easily available. When observing driving behavior, users frequently slow down and stop to view available parking spaces before continuing again.

To find the number of stops, we look for periods of time where the variance in the magnitude of acceleration drops below a threshold, then rises above the threshold again. The threshold is set to the average variance of each group of 100 acceleration magnitudes. While the car is moving, the acceleration is constantly changing from changes in speed, turns, and bumps in the road. The variance is calculated on a moving window of the last 100-recorded values.

We first loop though each recorded value, calculating the magnitude of the acceleration and store them in a list until it accumulates 100 values, when it starts accumulating the variance of the last 100 magnitudes it has seen. Each time a new value is added past 100, the oldest value is removed from the beginning, so the variance covers the last 100 values.

After the variances are calculated, the threshold is set to the average of all the variances. Then a second loop goes through the data accumulates a list of 100 magnitudes, similarly to the first loop, and starts calculating the variance and removing the oldest entry when it reaches 100 values. If the variance is below the threshold and we have previously detected movement, we increment the stops counter by one. If we did not previously detect movement, we do not count a stop even if the variance is below the threshold. This keeps us from counting stops in the case where the variance in the first recorded values are already above the threshold. The following figures show an example of the differences that can be seen during a trial in an open parking lot (Figure 3) versus a crowded parking lot (Figure 4).

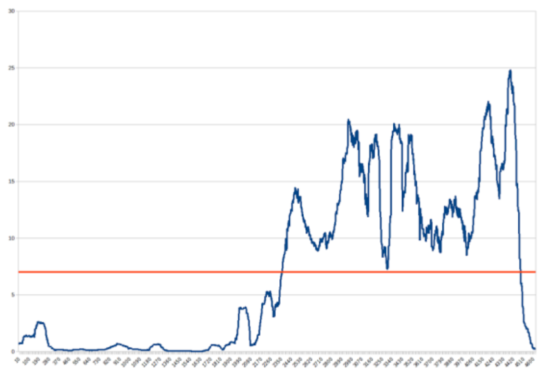


Figure 3: Variance of Crowded Parking Lot

This figure shows the graph of our calculated variance values along with the threshold (average variance). This trial consisted of 5 stops and our algorithm detected 5 stops.

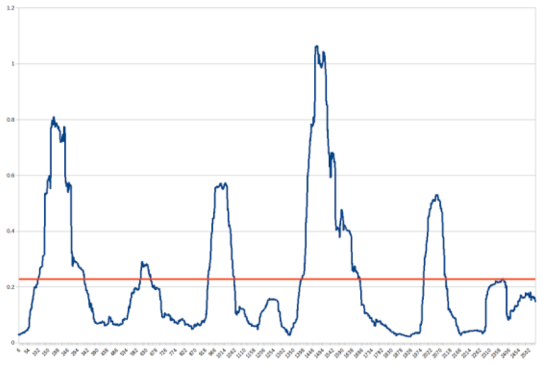


Figure 4: Variance of Empty Parking Lot

This figure shows the graph of our calculated variance values along with the threshold (average variance). This trial consisted of 1 stop and our algorithm detected 1 stop.

Distance from a Point of Interest (POI)

Parking location is extremely pertinent to our determination of parking lot availability. Each parking lot has a predetermined point of interest. For example, in the Sage Hall parking lot we define the point of interest to be the Sage Hall entrance facing the parking lot.

To determine parking location for a given user, the application first determines the timestamp upon parking. We cannot simply account for situations where the location data remains unchanged; a car may be stopped waiting for pedestrians to walk by. Instead, we utilize Google's Activity Recognition API to determine the approximate timestamp when the user parks. This can be determined by first detecting when a user begins walking. Working from that point backwards, we determine the rough time at which the activity recognition service is not supremely confident (< 40% confidence) that the user is still within a vehicle. This timestamp is an accurate reflection of the time at which a user has parked. The application then finds the closest location data point that corresponds to this parked timestamp. We use that point to correspond to the user's parked location.

The distance from the point of interest simply corresponds to the distance between our parked location and our predetermined point of interest.

We recognize that parking lots may have different points of interest. For instance, on opposite sides of a parking lot could be two buildings that may be indicated as points of interest for drivers to that parking lot.

One potential solution for future implementation is to determine the time spent walking before exiting a given parking lot, rather than just determining distance to the hard coded point of interest. A high walk time relative to the size of the parking lot indicates that the user parked far from their point of interest. A low walk time before exiting the parking lot suggests that the user parked relatively close to their point of interest.

Activity Recognition

We leverage Google's Activity Recognition API to determine user behavior at particular moments during the parking process.

Once Google's Activity Recognition API is implemented, the service in periodic intervals provides confidence metrics for a number of detected activities: in vehicle, on bicycle, on foot, walking, running, still, tilting, and unknown. While most power consuming, our application relies on the fastest detection interval possible for the service. This provides the most immediate and precise classification of user behavior at any given moment. CUbiPark relies on activity recognition to identify a number of key features pertinent to our parking lot classification.

Upon recognizing entrance and exit of a parking lot through Geofence monitoring as detailed above, activity recognition provides valuable information concerning the state of the user. If the user is on foot or on a bicycle when entering a parking lot location, CUbiPark knows to dismiss these events, as the user is not in a vehicle seeking parking. However if activity recognition is confident that the user is currently within a vehicle upon entering the parking lot, CUbiPark knows to begin collecting and processing data as to the state of the parking lot. When exiting a parking lot determined by the Geofence solution detailed above, activity recognition determines what state the user is at as they leave the parking lot. If the user is walking, CUbiPark can confidently say that there was parking available when the user entered the parking lot. If the user exits the parking lot within a vehicle, CUbiPark can infer that there was no available parking within the designated lot, causing the user to drive away.

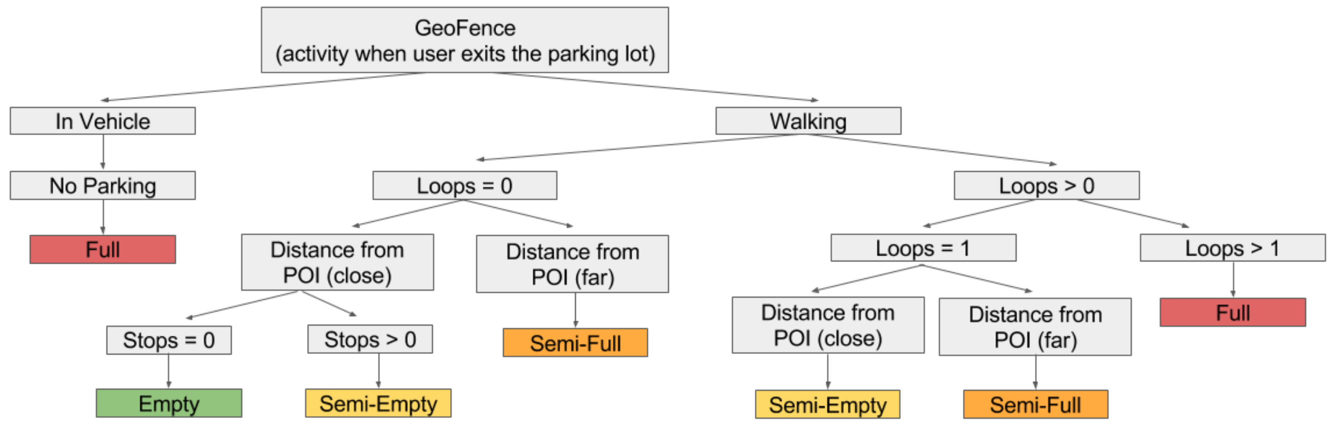
Activity recognition is also valuable in identifying transitions between stages of parking: in vehicle, still, exiting vehicle on foot. By identifying the timestamps of these transitions within our dataset, we can use this data to reconcile with our location and accelerometer data. This is important in distinguishing between stops while in vehicle versus a complete stop because of parking. It also helps distinguish between stopping in vehicle and parking and leaving the vehicle on foot.

Activity recognition is crucial for coordinating between location and accelerometer datasets as well as offering key insight into user behavior in and outside a parking lot.

Time in Lot

We do not formally use this metric as a means of determining availability within a parking lot, but it remains a convenient sanity check for our data. We have also discussed how a time in lot or time walking metric could be viable for more accurate future implementations.

Figure 5: Implemented Rule Tree for Classification



Time in a given parking lot is simply determined by subtracting the time that the program recognized the user entered the Geofence parking lot to the time that the user exited the parking lot. Further specificity can be applied to also find the amount of time spent walking or the amount of time spent driving.

Availability Estimation and Classification

The availability of parking is ultimately determined by a decision tree factoring in the features discussed above. Seen above in Figure 5, the decision tree maps observations of relevant features all the way to an end status concerning availability in the given parking lot.

We classify availability into three categories denoted by color: red, orange, yellow, and green.

Red: Full (parking is not available)
Orange: Semi-Full (limited parking is available)
Yellow: Semi-Empty (parking is largely available)
Green: Empty (parking is completely available)

Ultimately, the role of CUbiPark is to provide meaningful information to users in order to guide their parking behavior and decisions. While we provide rough estimations of capacity based off denoted color, the role of the application is not to specifically identify capacity within a parking lot but to determine ranges of confidence that a parking lot has empty spaces available.

We believe that this four-tiered color system provides the most actionable amount of data for a user. Future work should, however, aim to more precisely denote capacity in the most meaningful way possible for users. This involves determining whether more precise breakdowns can be derived from parking behavior and the actual approximations of availability given parking behavior. We developed our decision tree (Figure 5) to infer parking availability based off the features detected and ranked and prioritized them in the order we felt was most deterministic and significant to a driver's parking behavior.

Feature priority

The decision tree governing CUbiPark's availability estimation relies on the features discussed above. We go on to discuss how each feature governs the decision making process for availability estimation in the order that they are processed in the decision tree.

Activity Recognition

Upon exiting a parking lot, CUbiPark determines user activity behavior. This is the first branch of our decision tree. If the user is still within a vehicle, the application can deduce that there was no parking available. If the user is on foot, CUbiPark can recognize that the user did park and that the application must next estimate how much parking is available.

Number of Loops

The number of loops that a driver makes within a parking lot is a significant indication of the capacity of a parking lot. Looping around a parking lot indicates that there is parking available but that the most optimal spots were not available.

If the number of loops is zero, we can assume two scenarios existed. Either the driver recognized that there were no spots available in the most optimal locations and consequently decided to park far away or that there were optimal spots available and the drive parked there.

If the number of loops is equal to one, either the driver was able to find a newly formed vacancy at an optimal location or the driver had to settle on a subprime location.

If the number of loops is greater than one, it implies that the driver struggled to find parking within the parking lot. We can infer that there is a high probability that the driver took a newly created vacancy after looping multiple times.

Distance from the Point of Interest

Distance from a point of interest is a large indicator of the availability of parking within any given lot. We assume that parking behavior by individuals is motivated by a desire to be as near to a point of interest of parking lot as possible.

While parking extremely near a point of interest is not a guarantee of available parking, combining our knowledge of the distance from a point of interest with other detected features such as the number of loops and stops can provide a keen understanding to the state of the parking lot.

Number of Stops

We found that while a driver is looking for spots in a crowded parking lot, they tend to slow down or even come to a full stop to check if a space is available or if a particular spot is the right size for them. In addition, stops might signify that the driver is waiting for someone to leave a spot, which is also an indication that the lot is relatively full. When a parking lot is more open, drivers can easily see the available spots and usually do not make as many, if any, stops while finding a spot.

However, since there are other unpredictable reasons for drivers to appear to make stops in the parking lot (such as pedestrians crossing, avoiding objects, or making tight turns), we consider this feature last as a lower priority indicator.

In some instances, the number of stops is irrelevant to our decision tree logic since prior features outweigh their significance. For example, if an individual loops around a parking lot multiple times, without knowledge of the number of stops, we can estimate that the parking lot was at capacity and the driver was looping in circles waiting for a parked driver to vacate the parking lot.

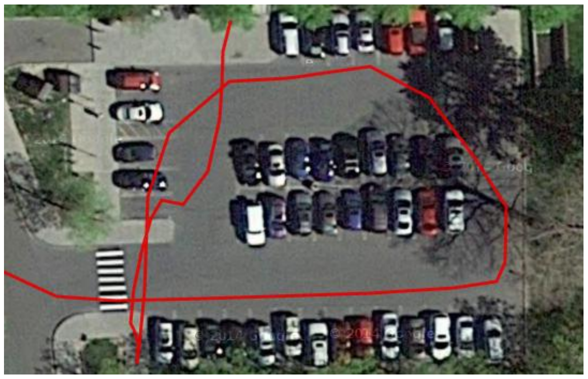


Figure 6: Map of Trial GPS data

This figure shows the user's driving and walking data mapped onto a Google Maps' image. This demonstrates looping, parking, and walking to the users' point of interest.

EVALUATION

Two rounds of user testing were conducted to test driver behavior while parking, test the ability for our application to accurately detect features relevant to driving and parking behavior, and lastly to test the classification of parking lot status based off our initial decision tree.

Round 1

To test our application and decision tree classification, we conducted an initial round of user testing in the Sage Hall parking lot. We had 4 trials where users drove into the lot and attempted to park under our four different conditions, one for each trial (empty, semi-empty, semi-full, and full). We compared the actual data with how our application performed and realized that several aspects of our classification were not as accurate as expected. We therefore decided to adjust and update our model to match better with the actual driver behavior.

Adjusting and Updating Our Application

For our original design, we recorded GPS information as often as possible, giving us jagged data points that jumped all over the map. In order to reduce these inconsistencies and clean up the data, we slowed down the collection of GPS data from the multiple points per second to every 2 seconds instead. This gave us a more accurate and smooth picture of the drivers' path through the parking lot.

We also realized that our threshold for distinguishing the users' parking spot as being "close" versus being "far" from their point of interest was too large. We therefore refined our point of interest classification to include a smaller distance threshold, matching the area of the parking lot better.

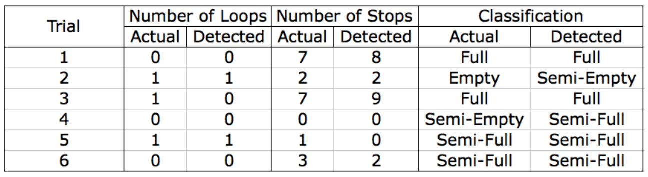
Lastly, we found that the activity recognition was not updating as fast as we thought. For example, we assumed that once a driver parked, the activity recognition would detect the user as being "still". However we noticed that if the user parked and then immediately or relatively quickly exited their vehicle, the activity recognition would go directly from "in vehicle" to "walking" with no detection of "still". Therefore, we updated our feature detection to account for this fact.

Round 2

After making these adjustments to our application, we conducted another round of user testing consisting of 6 trials in the Sage Hall parking lot on campus at different times of the day. For each trial we had a different participant drive to the Sage Hall parking lot and attempt to park as if their desired destination was Sage Hall itself. We manually recorded the number of loops, final parking spot, and number of stops that the driver made during their trial. After attempting to park, we also asked each participant to designate the lot as either empty, semi-empty, semi-full, or full. The following table (Table 1) shows the results of our tests, displaying the actual number of loops, number of stops, and classification compared to what our application detected.

DISCUSSION

While these results show that our application is not 100% accurate, it does demonstrate that this type of sensing is possible. Actual versus detected statistics for the features detected were within a minor degree of error and classification was relatively accurate.



Trial	Number of Loops		Number of Stops		Classification	
	Actual	Detected	Actual	Detected	Actual	Detected
1	0	0	7	8	Full	Full
2	1	1	2	2	Empty	Semi-Empty
3	1	0	7	9	Full	Full
4	0	0	0	0	Semi-Empty	Semi-Full
5	1	1	1	0	Semi-Full	Semi-Full
6	0	0	3	2	Semi-Full	Semi-Full

Table 1: Results from Round 2 of Our User Testing

There is certainly work and improvement to be made in both feature detection as well as classification. For one, it is difficult to completely discern user behavior (e.g pedestrians walking through a crosswalk versus actually stopping for parking purposes). In addition, the delineations in our current system between tiers of parking availability are very slight. An inaccuracy of a number of meters in determining distance from a point of interest shifts our classification from Empty to Semi-Empty. Ultimately however, our results suggest a definitive ability to track driver behavior and infer parking availability based off features that we detect.

LIMITATIONS

There are a number of limitations to our implementation and the applicableness of our application to all parking behaviors and parking lots.

Small Sample Set

For the purposes of developing our model, we only conducted user testing in one parking lot: the Sage Hall parking lot. In addition, we were restricted by limited accessibility to drivers with cars and were therefore only able to complete 6 different trials for our second round of user testing on our finished application. This provided us with only a small sample set in a specific parking lot with a limited number of drivers, making it difficult to determine how significant our results are or if our success would apply to additional parking lots.

Imprecise Sensor Data

The implementation discussed above relies heavily on accurate and frequent sensor data in order to make judgments concerning parking location, activity recognition, and driver behavior within a parking lot. Imprecise sensor data especially by nature of poor GPS location services within the parking lot can impact our recognition of availability within a parking lot.

Likewise, the Google Activity Recognition API used in this project is both inconsistent in its update rate as well as unconfident in its feature recognition.

Unpredictable Driver Behavior

As discussed, CUbiPark relies on particular assumptions concerning driver behavior within parking lots. When driver behavior is irregular or illogical (e.g driving through parking spots to get to a destination), our estimation of availability within the parking lot may be negatively affected.

Setting

Our implementation relies heavily on accurate GPS data, which may become inaccurate when parking in garages or indoor parking structure. Future work should look to leverage other sensors such as barometric pressure coupled with existing accelerometer data to determine rough parking location within an indoor structure.

FUTURE WORK

CUbiPark has shown merit in the ability for a mobile application to determine the availability of parking within a given parking lot. We hope to further develop CUbiPark to take other features into consideration and improve its accuracy in estimating parking availability.

Iterative Estimation

Future work should look to incorporate functionality to address iterative trips to the same parking lot by different users within a short time frame. If a given user travels to a parking lot immediately after another user has already went there, the application should be able to use both sets of data to provide a more accurate estimation of parking availability.

This functionality may be particular useful in situations where many users are searching for parking within a short time frame (e.g. stadium parking for a sporting event)

Account for Additional Driver Behavior

CUbiPark relies on particular driving behavior to estimate parking availability. While we designed the application to account for the most prominent user behavior while driving, there are surely more minute driving behaviors that can help improve availability estimation.

We have identified a number of additional driving behaviors that future work would seek to identify and classify.

Refined Point of Interest Calculation

In previous discussion of feature detection, we noted that a given location could have many point of interests depending on the user. Instead of interpreting the distance to a given point, one option would be to determine the amount of time spent walking relative to the size of the parking lot. This would help more accurately determine parking lot availability given multiple points of interest and could be generalized to determine availability.

Reversing

One behavior that was sometimes exhibited during user testing and observing driving behavior was the act of reversing. Users may find themselves reversing rather than driving around the parking lot again, or may find that a space opened up directly behind them. Identifying reversing driver behavior can help shed further light on driver behavior within parking lots.

Generalize Behavior Across Parking Lots

CUbiPark relies on manually inputted information concerning parking lot information. It requires not only the coordinates for the parking lot itself, but also latitude, longitude coordinates for loop point within a parking lot as well as the coordinates concerning the parking lot entrance to be entirely accurate.

A means of potentially allowing users to denote when they are in a parking lot and interpret the size, number of spots, number of loops without knowledge of the structure of the lot, and location of the parking lot entrance, would be incredibly valuable in scaling the platform for use outside of Cornell University.

CONCLUSION

We have presented CUbiPark, a mobile application that provides users with real-time parking availability estimates. CUbiPark collects data unobtrusively, without any input from users, as long as drivers have their smartphone in the car with them. By analyzing GPS, accelerometer, and activity recognition data to extract key driving behavior features such as the type of activity the user is engaging in when entering or exiting the parking lot, the number of loops the user makes, the distance from the users' parking spot to their point of interest, and the number of stops the user makes, CUbiPark classifies the parking lot into four categories, giving future users the ability to check which parking lots are most likely to have open spots before even heading out onto the road. From our user testing and evaluation, we demonstrated that CUbiPark is capable of closely predicting lot availability. This approach requires no additional infrastructure or equipment as well as no direct interaction from the user.

Continued work on CUbiPark can take a number of variations. One can continue to user test with a larger sample size to more accurately refine the decision tree governing availability as well as further test the accuracy of feature detection. Looking forward in development, we hope to address some of the limitations we encountered and include crowdsourcing capabilities to make the application even more accurate.

REFERENCES

1. Chen, X., Santos-Neto, E., & Ripeanu, M. (2012, October). Crowdsourcing for on-street smart parking. In *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications* (pp. 1-8). ACM.
2. Lu, R., Lin, X., Zhu, H., & Shen, X. S. (2009, April). SPARK: a new VANET-based smart parking scheme for large parking lots. In *INFOCOM 2009, IEEE* (pp. 1413-1421). IEEE.
3. Mathur, S., Jin, T., Kasturirangan, N., Chandrasekaran, J., Xue, W., Gruteser, M., & Trappe, W. (2010, June). Parknet: drive-by sensing of road-side parking statistics. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 123-136). ACM.
4. Nawaz, S., Efstratiou, C., & Mascolo, C. (2013, September). Parksense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th annual international conference on Mobile computing & networking* (pp. 75-86). ACM.