iParking – Real-Time Parking Space Monitor and Guiding System with Cloud Service

Ching-Fei Yang, You-Huei Ju, Chung-Ying Hsieh, Chia-Ying Lin, Meng-Hsun Tsai, Hui-Ling Chang

CSIE, NCKU, No. 1 University Rd., Tainan 701, Taiwan, R.O.C.

Abstract

By the popularization of cars, average number of vehicles owned by one person grows with passing days. However, the number of parking areas is out of proportion. In order to satisfy the requirements of parking space and reduce illegal parking, we propose iParking, a real-time parking space monitoring and guiding system, in this paper. We lay emphasis on roadside parking. The system determines and records empty parking spaces through cloud computing, wireless technology between vehicles, and image analysis. It tells you the nearest location of empty parking space while drivers have requests. We expect the system to cause attention to more people and government while it solves relative problems about parking space.

Keywords: cloud computing, image recognition, parking space management, wireless technology

1. Introduction

Recently, parking problem has become people's harassment. It is shown from statistical data in Ministry of Transportation, Taiwan that the number of registered vehicles is 7,554,319 until December 2014 [1]; However, it is also mentioned that the number of legal parking space is about four million in total. Furthermore, it will cause several problems such as the extremely slow speed while finding parking space, scrambling for roads with scooters, parking temporarily in dangerous part, or driving U-turn illegally. The behavior will not only break the safety and regular of transportation but also make noise and consume resource. It is easy to observe that some vehicles need to find parking space by themselves while roadside parking spaces are not enough.

The situation will bring out arbitrary parking, and it is also the main reason of illegal parking.

Over the last few years, LBS (Location-Based Service) [2] is getting noticed along with the appearance of smartphones. LBS can apply broadly to different area like health, job, daily life, and etc. Thus, how to use LBS to help different users find the appropriate parking space is vital. The usage of monitoring parking space now is to provide roadside parking space's locations at best, but it will not tell drivers where the vacant space is. Thus, we would like to develop a monitor and guiding system focus on roadside parking to provide the information of nearby parking space and help drivers park with the fastest way.

We will introduce the system in four parts. First of all, know what is the demand of our work and compare with other techniques. Next, show the details, especially features and structure of our service. In addition, demonstrate how we implement and design the system. Finally, make a conclusion, discuss more about future work.

2. Related Work

2.1. Demand of parking space

In the statistical table of important indicators from Ministry of Transportation, Taiwan [1], it is pointed out that parking space is one of the important indicators in addition to the number of vehicles. Figure 1 is the comparison chart between the number of vehicles and parking spaces from 2006 to 2014 in Taiwan.

It is shown from Figure 1 that the difference between supply and demand of parking space is about two million. In addition, it is pointed out from trend of line that the growth rate of vehicles and parking spaces is closed. However, it is not simple to add parking spaces because it involves road network planning. In this knotty situation, it becomes vital and urgent to solve the management of parking space in order to make good use of limited resource.

Furthermore, some recent researches verify that people would rather spend more time finding roadside parking space than off-street parking even if there are vacant spaces in the off-street parking lot [3]. Take Tainan City for example, service in parking lot does not meet drivers' expect, such as high parking fee, mess surroundings, or etc. Hence, the situation results in low usage rate

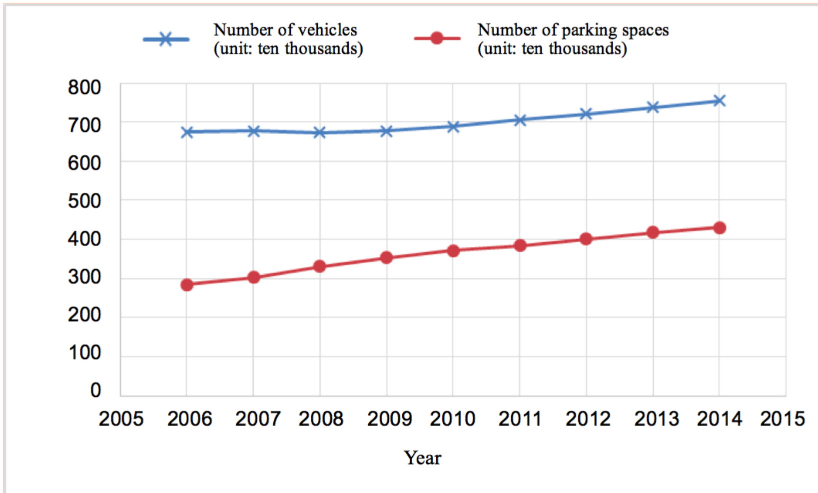


Figure 1: Statistical table of vehicles and parking spaces' number from 2006 to 2014

of parking lot and shortage of roadside parking space. Thus, to solve the problem of roadside parking is necessary.

2.2. Existing parking space monitored technique

All of existing parking space monitored techniques are limited to parking lot and only supported by sensors. For instance, intelligent parking lot uses wireless sensor network, ZIBEE, pressure sensors [4, 5]. They update database by sensors to know if it is empty. The another instance is Eco-Community plan developed by several schools [6]. Its main method is using the sensitivity of the sensor, Octopus II, updating changes to database. Therefore, the changes will tell users information about parking spaces. However, the two techniques have a big constraint when it comes to downtown area. It is a huge challenge to set up sensors for all parking spaces in downtown due to the fee of building and maintenance. That is to say, both of them are not suitable to roadside parking in comparison with our system.

As regards other apps in the market, they are connected to nearby parking lot, offering real-time information. However, only few of them mention payment information about roadside parking. In conclusion, none of the apps in the market provide function to find roadside parking spaces until now.

3. Service and System Structure

3.1. Software platform

We choose smartphone and related device to complete mobility, driving recorder, and Network communication by reason of the target users, people with mobile vehicle. We build the application base on Android, using Java to implement code structure and GUI design.

3.2. Features

3.2.1. Cloud storage and computing

Cloud Storage is an online service which can save data on virtual server through Internet. The service become more and more popular due to the popularity of Internet and the increasing demand of data storage. That is to say, simply save data in actual hard dick is getting insufficient. Therefore, limited storage devices will bring more benefit by Internet and storage virtualization technique.

In order to improve the efficiency of driving records, and reduce the capacity of mobile device. We will refer to existing cloud storage service, analyzing data through servers in cloud, and send the parking information to users who have request.

3.2.2. Static image streaming

Streaming media is a process to compress a series of media data, send through network section, and offer real-time media service on the Internet [7]. By the technique, media data are able to watch without downloading whole media. Therefore, it is called "streaming" because data in the process behaves like running water.

We can say that static image streaming is to connect images, record the event over the next period of time. Under the premise that analyzing driving records accurately, we will use static image streaming to lessen the burden instead of sending whole driving record.

3.2.3. Analyzing vacant parking space

In the reference [8], the author has proposed solutions to detect if parking spaces are vacant. Its technique includes Hough line detection and Canny edge detection, implementing by OpenCV library. The original method has two limitations. The first one is that it can only identify one photo at a time while the another is that only the parking space at bottom right corner can be identified. We breakthrough them by using static image streaming.

3.3. Efficacy

3.3.1. Monitoring parking space

The situation of parking space is different from area, timing, and location. Therefore, the key point is how to monitor the specific parking space immediately. Besides, if there are many people use this service at the same time, it will be fairly accurate with steadily update.

3.3.2. Saving and analyzing driving recorder

In the process of detecting vacant parking space's condition, it is necessary to analyze big data and use large storage. Hence, we use cloud service and client-server model to handle and send all the requests in order to reduce the usage of memory, storage, and workload.

3.3.3. Data transmission

If the goal is to keep high accuracy and immediacy, the system will bring out high Internet usage because it continuing transfers driving recorder. Therefore, we will capture driving image with a fixed distance according to speed of the vehicle. Coordinating with GPS position, it will become image streaming instead of video. That is to say, capacity and the time of data transmission can be saved.

3.4. Structure

We mainly focus on car owners. Besides, we will use our own approach to detect driving record automatically and communicate between vehicles. The application is built on Android, expected to run the program on driving recorders. We will limit to a specific road section while testing and developing the system.

Four steps are supposed to proceed. First, determine the specific road section, and collect data; Next, sort out the collected data. Start to plan the structure of program; Then, begin to develop the program, add GUI, do simulation, and test system. Finally, analyze and present whole the research. Figure 2 is the flow diagram of these steps.

3.4.1. Building cloud service and algorithm design

The main usage of cloud service is to implement each operated function, handle users' request, and provide storage to save driving records. The operated function includes receiving image data from users, analyzing images, building database, searching database, and etc. Hold time is the first concern

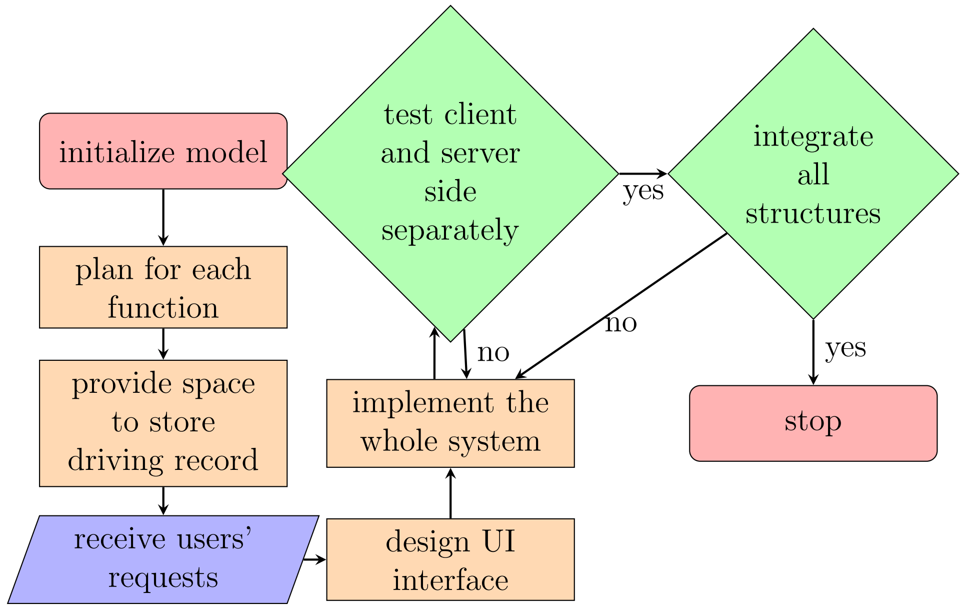


Figure 2: Research flow diagram

because of massive calculation. The other factor is the availability of data, for example, we will not use old data and data which has been analyzed in the same location. This algorithm will help reduce repeated operations, furthermore, it helps us make sure that we are analyzing real-time image in every data.

3.4.2. Program structure in device and UI design

We will plan our service with relation to different characters. For example, it is necessary for users to search nearby parking space, it is required for devices to send image information to servers in cloud, and it is important for servers to analyze and collect data. As for UI design in app, in order to realize convenient searching function and interface, it need to be designed from the view of users.

In Figure 3, we divide the system into three parts, User A, User B, and Cloud server. User A sends images and GPS information through WiFi / 3G / 4G technique; User B sends request for parking, and get the parking space information immediately; And cloud server uses algorithm to know if there are any spaces and tell users the result if they have requests.

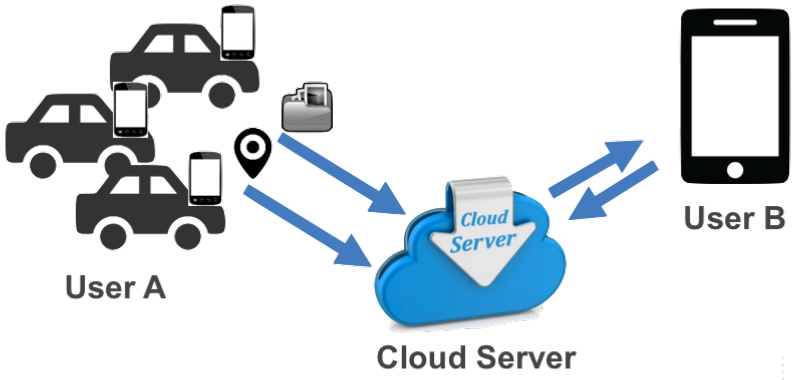


Figure 3: System flow chart

The error of GPS measurement is about 5 to 10 meters. We can say that it is about 1 or 2 roadside parking spaces. In order to enhance the accuracy, we will combine Google Maps API, take the advantage of its navigation and distance matrix service. Besides accessing the speed and distance of vehicles, the API can also help send GPS location to keep loading and operating fast in the device.

When it comes to clients, we will check whether there is anyone else sending the same information of specific location at the same time or not. It will not send information if anyone else is sending the data. However, if there is not anyone else sending the data, we plan to capture image immediately after moving a small and appropriate distance before sending to servers in cloud.

4. Implementation

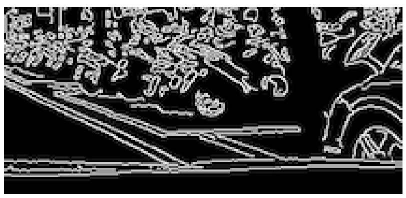
The system is divided into three parts - image recongization, cloud server, and client's application. At first, the three parts will be implemented separately. They will be combined and operate after they all make a certain proportain.

4.1. Image recognition of roadside parking space

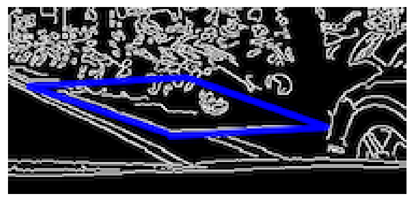
Image recognition and analysis of roadside parking space is implemented by C++ with OpenCV library. The program will return if the image of parking space is vacant after received an image.

(a) Gray scale image

(b) ROI image



(c) Detect each side



(d) Draw parking space

Figure 4: Process of Image recognition

4.1.1. Setting ROI (Region of Interest)

First, as Figure 4(a) and Figure 4(b), the program will convert the image into gray scale. The reason is that the perspective of image will affect the degree of image recognition. Moreover, in order to eliminate the noise and white area such as sky and clouds, the image will be divided into four equal parts. Only the right-bottom part will be reserved because the spaces are usually located in the right hand side (in right-hand-traffic countries).

4.1.2. Sides detection and Noise reduction of Image

The program uses Canny edge detection in OpenCV to find each side of parking space. Afterward it reduces noise by the way, Median Blur. (Figure 4(c))

4.1.3. Find out Lines in the Image

First, we use Hough line detection to detect straight lines by angle and intersection of lines. With standard Hough transform function, a single-channel, grayscale image is sent. Next, the function will return a two-element vector lines (ρ, θ) while ρ is the distance from the coordinate origin $(0, 0)$ and θ is the line rotation angle in radians. After getting lines, we divide lines into three parts, vertical lines ($tan\theta = 0$), horizontal lines ($tan\theta < -5$ or

$tan\theta > 5$) and parking lines (neither vertical lines nor horizontal lines). It is also our approach to distinguish parking spots from roads.

4.1.4. Choose the Correct Lines

The next step is to choose the correct lines in pool composed of several lines with have same direction. Because the particular spot may not look same by different distance, angel, or camera. The program will find the longest length of parking space, which is also the line closed toand parallel to the road.

From the result of Hough transform function, we get vector (ρ, θ). That is to say, we can get start and end points of each line in the same direction. Therefore, we get the parking line that is constructed by the average of all starting points and all end points. Finally, the longest line of parking space is reduced to a specific one.

4.1.5. Get the Appropriate Line Segment

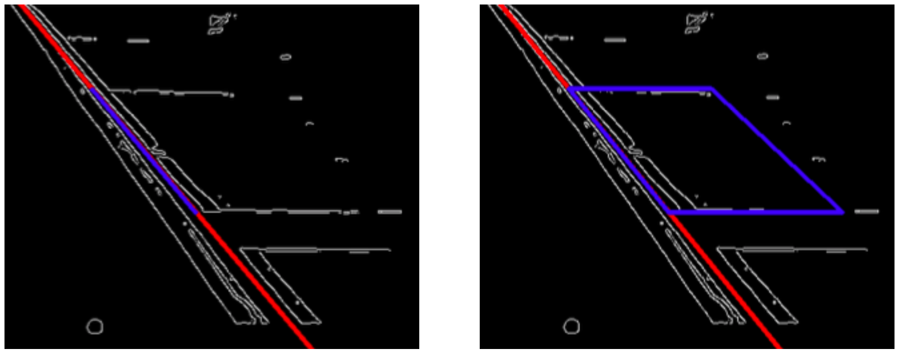
Within the longest line we get from last step, it will have n points of intersection if there are n horizontal lines. Next, the points are sorted from small to large by their y value. And the program subtracts the second y value from the first one between each sorted point. If the result is more than 10, we suppose that the two pionts is not close. Therefore, we store the second point because they are not on the same line. After the process, the program gets several points which are on different horizontal lines. The last but not the least, we find the longest distance between the points. It is also the correct parking space's line segment that is closed to and parallel to the road such as Figure 5(a).

4.1.6. Determine each Line of Parking Space

By angle of view (AOV)[9], if we supposed that the length of straight line we get from last step is m, the length of top horizontal line is $m \times 0.9$, and the length of bottom horizontal line is $m \times 1.1$. Thus, as shown in Figure 5(b), the program will get the parking space by points, length and width.

4.1.7. Detect if the parking space is able to use

If the appropriate space is not found, the program will return the result of no vacant space (equals to occupied by vehicle). However, if the space is found, it will continue to detect if the space is able to use.



(a) Blue line is the line segment (b) Find out all lines

Figure 5: Process of Image recognition

If the ratio between the side of parking space and its shelter is more than a certain number, it means that the space has been occupied. Therefore, the program will return no vacant spaces. On the contrary, it will return there is a vacant parking space.

4.2. Cloud analyzing server

After passing image recognition of roadside parking space testing, the program mentioned above will be moved to cloud service. Moreover, it will coordinate with the open data offered by government. The data will provide the information about roadside parking space. Therefore, we are able to know which road sections do not have spaces, prevent analyze the images from those sections. We implement the server by nodejs action hero framework; In addition to offering API with http, we will provide interface for webpages in order to let users find parking spaces directly. Non-relational database, Mongo, is also used to accelerate access and operated speed.

Servers in cloud will translate longitude and latitude into address information while receiving GPS information and images by users. Next, it will compare the road section with open data to confirm if the section provides parking spaces. After successful analysis, the result with address will update to database for other users.

4.3. Mobile APP (client side)

It is divided into several steps to implement, mainly separated into user (device) and server side. First, we use Android platform with Android Studio and Android SDK, which are based on JAVA, to develop client's application.

(a) Home (b) Record (c) Setting (d) Result (e) Map

Figure 6: User interface

Figure 6 is our main interface and function. Figure 6(a) is the first screen of the system. You can choose the function of both recording and searching. Figure 6(b) is recording screen. It will use device's camera automatically, temporarily save the records in iParking folder. The purpose is to let users check the record, choose if they want to provide it to other users or delete it. After uploading the record, it will be deleted. Figure 6(c) is the setting page. Users can change their sending rate (network flow) and searching range while finding parking space. Figure 6(d) is the result of searching. Users will know how many parking spaces nearby. After choosing the one user prefers, the location will be shown on the map like Figure 6(e). Therefore, user can be navigated to the space combined with Google API.

4.4. Testing

In the part of testing, we will initially test client and server side separately, merge them with UI design. Finally, we will test the integrated system by the following steps. First, a single vehicle. That is to test with different speed, make sure that static image streaming is worked with cloud service. Next, test the efficiency of multiple vehicles. Verify that no images will be in the same location at the same time. The last step is to confirm that client side can get the correct information immediately.

5. Conclusion

The goal of this system is to offer users a practical and useful application. Users are able to find parking spaces while having requests. In addition, traffic problem, air pollution problem, and the behavior of illegal parking will decrease.

We have a simple but completed system until now. The system includes cloud service, the technique to analyze images, and an application for Android. We expect to develop more applications on iPhone and Windows

Phone. Besides, we look forward to doing more research on accuracy of image recognition, server load balance, and Vehicular ad hoc network.

6. Acknowledgments

This work was sponsored in part by Ministry of Science and Technology (MOST), Taiwan, under the contract number MOST 105-2221-E-006-186- and MOST 104-2815-C-006-029-E.

References

[1] M. of Transportation, Statistical chart of important indicators in Taiwan (in Chinese), http://www.motc.gov.tw/uploaddowndoc?file=reference/g004.pdffiledisplay=g004.pdfflag=doc 2014.

[2] A. Kupper, Location-Based Services: Fundamentals and Operation, John Wiley and Sons, 2005.

[3] Z.-T. Huang, Simulating the On-Street Parking Behavior of Commercial Consumer Based on Agent-Based Model, Master's thesis, National Cheng Kung University, 2014.

[4] C.-J. Hsu, Intelligent roadside parking payment system (in Chinese), Urban Traffic 22 (2007) 97–105.

[5] Y. Cui, J. Zhao, Real-Time Location System and Applied Research Report, in: Internet of Vehicles - Safe and Intelligent Mobility, volume 9502 of *Lecture Notes in Computer Science*, pp. 49–57.

[6] C.-W. Yi, Eco-Community: Building an Intelligent Cyber-Physical Community Using Wireless Sensor Networks, Technical Report NSC100-2218-E009-002, Government of Taiwna, 2011.

[7] T. Kanter, R. Rahmani, Y. Li, B. Xiao, Vehicular Network Enabling Large-Scale and Real-Time Immersive Participation, in: Internet of Vehicles - Technologies and Services, volume 8662 of *Lecture Notes in Computer Science*, pp. 66–75.

[8] C.-Y. Lin, J.-T. Su, W.-P. Tsai, M.-H. Tsai, Finding Nearby Available Roadside Parking Spot, in: The Proceeding of IPPR Conference on CVGIP.

[9] T. Dobbert, Matchmoving: The Invisible Art of Camera Tracking, John Wiley and Sons, 2 edition, 2012.