

# iParking - A Real-Time Parking Space Monitoring and Guiding System

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

*tsaimh@csie.ncku.edu.tw*

*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

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

11 parking space by themselves while roadside parking spaces are not enough.  
12 The situation will bring out arbitrary parking, and it is also the main reason  
13 of illegal parking.

14 Over the last few years, LBS (Location-Based Service) [2] is getting no-  
15 ticed along with the appearance of smartphones. LBS can apply broadly to  
16 different area like health, job, daily life, and etc. Thus, how to use LBS to  
17 help different users find the appropriate parking space is vital. The usage  
18 of monitoring parking space now is to provide roadside parking space's lo-  
19 cations at best, but it will not tell drivers where the vacant space is. Thus,  
20 we would like to develop a monitor and guiding system focus on roadside  
21 parking to provide the information of nearby parking space and help drivers  
22 park with the fastest way.

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

## 28 **2. Related Work**

### 29 *2.1. Demand of parking space*

30 In the statistical table of important indicators from Ministry of Trans-  
31 portation, Taiwan [1], it is pointed out that parking space is one of the  
32 important indicators in addition to the number of vehicles. Figure 1 is the  
33 comparison chart between the number of vehicles and parking spaces from  
34 2006 to 2014 in Taiwan.

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

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

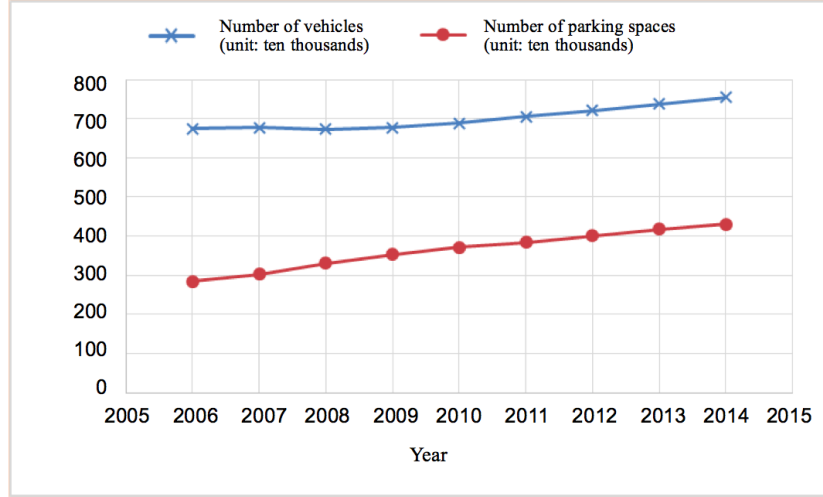


Figure 1: Statistical table of vehicles and parking spaces' number from 2006 to 2014

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.

### 65 3. Service and System Structure

#### 66 3.1. Software platform

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

#### 71 3.2. Features

##### 72 3.2.1. Cloud storage and computing

73 Cloud Storage is an online service which can save data on virtual server  
74 through Internet. The service become more and more popular due to the  
75 popularity of Internet and the increasing demand of data storage. That is  
76 to say, simply save data in actual hard disk is getting insufficient. There-  
77 fore, limited storage devices will bring more benefit by Internet and storage  
78 virtualization technique.

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

##### 83 3.2.2. Static image streaming

84 Streaming media is a process to compress a series of media data, send  
85 through network section, and offer real-time media service on the Internet  
86 [7]. By the technique, media data are able to watch without downloading  
87 whole media. Therefore, it is called “streaming” because data in the process  
88 behaves like running water.

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

##### 93 3.2.3. Analyzing vacant parking space

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

### 100 3.3. Efficacy

#### 101 3.3.1. Monitoring parking space

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

#### 106 3.3.2. Saving and analyzing driving recorder

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

#### 111 3.3.3. Data transmission

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

### 118 3.4. Structure

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

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

#### 129 3.4.1. Building cloud service and algorithm design

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

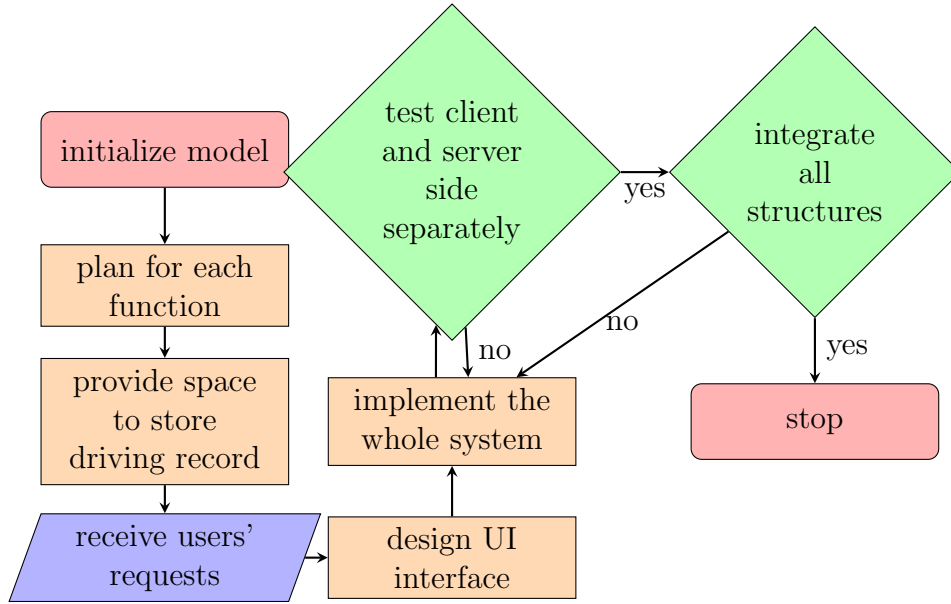


Figure 2: Research flow diagram

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

#### 139 3.4.2. Program structure in device and UI design

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

146 In Figure 3, we divide the system into three parts, User A, User B, and  
 147 Cloud server. User A sends images and GPS information through WiFi / 3G  
 148 / 4G technique; User B sends request for parking, and get the parking space  
 149 information immediately; And cloud server uses algorithm to know if there  
 150 are any spaces and tell users the result if they have requests.

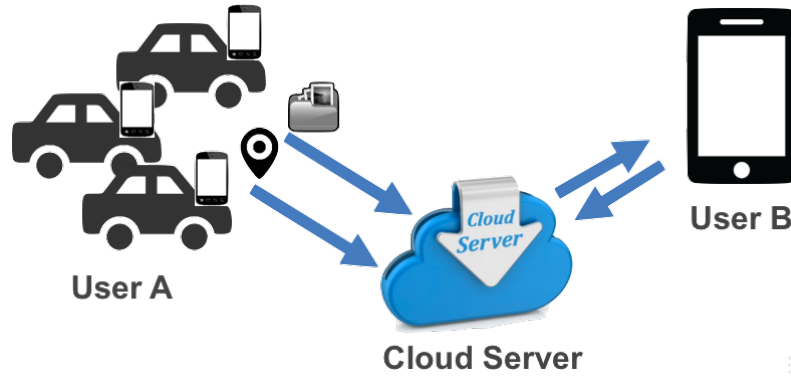


Figure 3: System flow chart

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

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

## 163 4. Implementation

164 The system is divided into three parts - image recognition, cloud server,  
 165 and client's application. At first, the three parts will be implemented sep-  
 166 arately. They will be combined and operate after they all make a certain  
 167 proportion.

### 168 4.1. Image recognition of roadside parking space

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

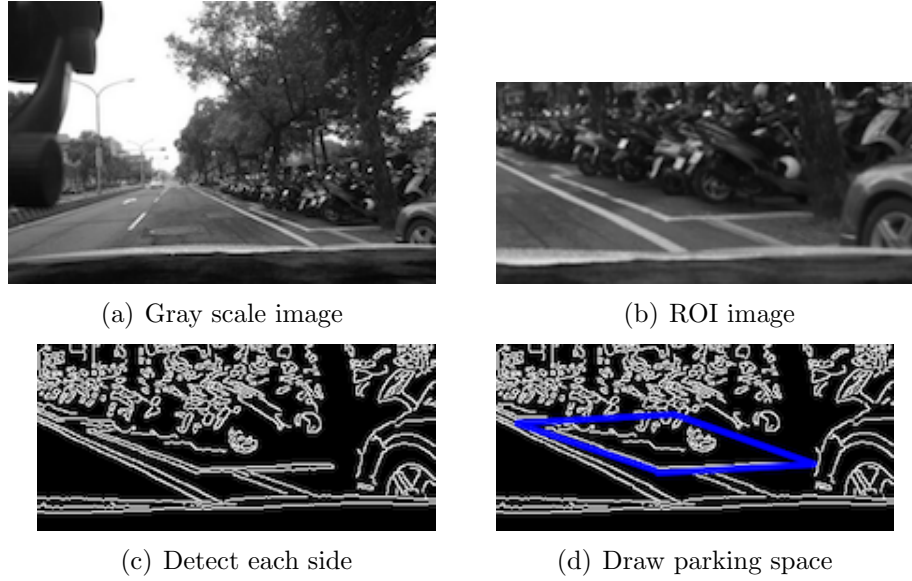


Figure 4: Process of Image recognition

#### 172 4.1.1. Setting ROI (Region of Interest)

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

#### 179 4.1.2. Sides detection and Noise reduction of Image

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

#### 183 4.1.3. Find out Lines in the Image

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



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

#### 192 4.1.4. Choose the Correct Lines

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

198 From the result of Hough transform function, we get vector  $(\rho, \theta)$ . That  
199 is to say, we can get start and end points of each line in the same direction.  
200 Therefore, we get the parking line that is constructed by the average of all  
201 starting points and all end points. Finally, the longest line of parking space  
202 is reduced to a specific one.

#### 203 4.1.5. Get the Appropriate Line Segment

204 Within the longest line we get from last step, it will have  $n$  points of  
205 intersection if there are  $n$  horizontal lines. Next, the points are sorted from  
206 small to large by their y value. And the program subtracts the second y  
207 value from the first one between each sorted point. If the result is more than  
208 10, we suppose that the two points is not close. Therefore, we store the  
209 second point because they are not on the same line. After the process, the  
210 program gets several points which are on different horizontal lines. The last  
211 but not the least, we find the longest distance between the points. It is also  
212 the correct parking space's line segment that is closed to and parallel to the  
213 road such as Figure 5(a).

#### 214 4.1.6. Determine each Line of Parking Space

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

#### 219 4.1.7. Detect if the parking space is able to use

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

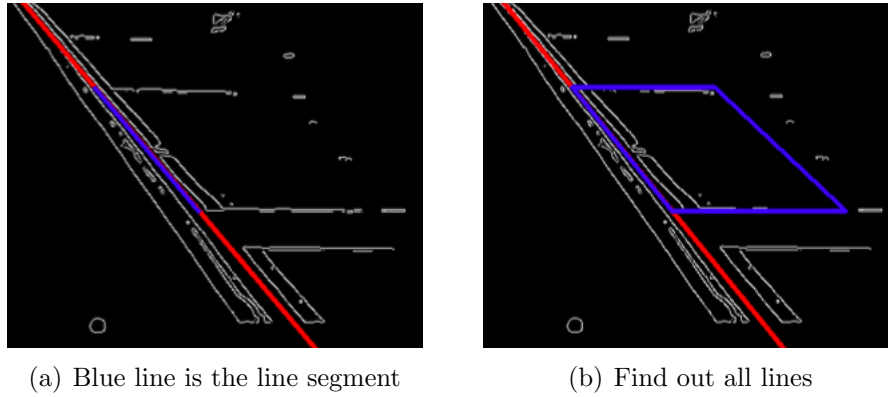


Figure 5: Process of Image recognition

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

#### 227 4.2. Cloud analyzing server

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

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

#### 242 4.3. Mobile APP (client side)

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

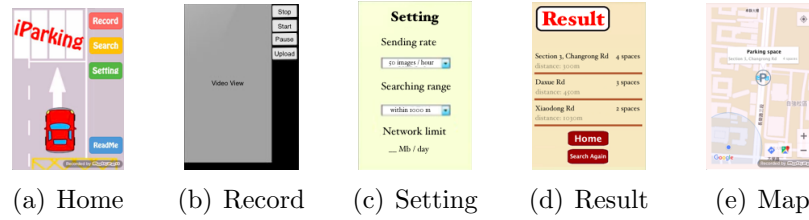


Figure 6: User interface

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

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

## 275 6. Acknowledgments

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

## 279 References

- 280 [1] M. of Transportation, Statistical chart of important indicators in Taiwan  
281 (in Chinese), 2014.
- 282 [2] A. Kupper, Location-Based Services: Fundamentals and Operation, John  
283 Wiley and Sons, 2005.
- 284 [3] Z.-T. Huang, Simulating the On-Street Parking Behavior of Commercial  
285 Consumer Based on Agent-Based Model, Master's thesis, National Cheng  
286 Kung University, 2014.
- 287 [4] C.-J. Hsu, Intelligent roadside parking payment system (in Chinese),  
288 Urban Traffic 22 (2007) 97–105.
- 289 [5] Y. Cui, J. Zhao, Real-Time Location System and Applied Research Re-  
290 port, in: Internet of Vehicles - Safe and Intelligent Mobility, volume 9502  
291 of *Lecture Notes in Computer Science*, pp. 49–57.
- 292 [6] C.-W. Yi, Eco-Community: Building an Intelligent Cyber-Physical Com-  
293 munity Using Wireless Sensor Networks, Technical Report NSC100-2218-  
294 E009-002, Government of Taiwan, 2011.
- 295 [7] T. Kanter, R. Rahmani, Y. Li, B. Xiao, Vehicular Network Enabling  
296 Large-Scale and Real-Time Immersive Participation, in: Internet of Ve-  
297 hicles - Technologies and Services, volume 8662 of *Lecture Notes in Com-  
298 puter Science*, pp. 66–75.
- 299 [8] C.-Y. Lin, J.-T. Su, W.-P. Tsai, M.-H. Tsai, Finding Nearby Available  
300 Roadside Parking Spot, in: The Proceeding of IPPR Conference on  
301 CVGIP.

- 302 [9] T. Dobbert, Matchmoving: The Invisible Art of Camera Tracking, John  
303 Wiley and Sons, 2 edition, 2012.