Identifying Parking Spaces & Detecting Occupancy Using Vision-based IoT Devices

Xiao Ling, Jie Sheng, Orlando Baiocchi, Xing Liu, Matthew E. Tolentino
Intelligent Platforms & Architecture Lab
University of Washington
Tacoma, WA, USA
{*xiaol7, shengj2, baiocchi, xingliu8, metolent*} *@uw.edu*

***Abstract*—The increasing number of vehicles in high density, urban areas is leading to significant parking space shortages. While systems have been developed to enable visibility into parking space vacancies for drivers, most rely on costly, dedicated sensor devices that require high installation costs. The proliferation of inexpensive Internet of Things (IoT) devices enables the use of compute platforms with integrated cameras that could be used to monitor parking space occupancy. However, even with camera-captured images, manual specification of parking space locations is required before such devices can be used by drivers after device installation.**

In this paper, we leverage machine learning techniques to develop a method to dynamically identify parking space topologies based on parked vehicle positions. More specifically, we designed and evaluated an occupation detection model to identify vacant parking spaces. We built a prototype implementation of the whole system using a Raspberry Pi and evaluated it on a real-world urban street near the University of Washington campus. The results show that our clustering-based learning technique coupled with our occupation detection pipeline is able to correctly identify parking spaces and determine occupancy without manual specication of parking space locations with an accuracy of 91%. By dynamically aggregating identied parking spaces from multiple IoT devices using Amazon Cloud Services, we demonstrated how a complete, city-wide parking management system can be quickly deployed at low cost.

***Keywords*-parking management system; IoT; topology learning; occupancy detection; edge device; computer vision**

I. INTRODUCTION

Information and Communication Technologies (ICT) and Internet of Things (IoT) are receiving great interest among city governments to make cities more sustainable, accessible, livable and "green." An urban development vision called Smart City has been proposed in the attempt to utilize innovative solutions to manage public affairs and resources [1]. A key goal of Smart Cities is to enhance the quality of lives of residents and visitors by providing real-time information about how the city is be used. This includes traffic management and parking management.

Within large urban environments, parking management constitutes a significant challenge. According to some studies, between 28% and 41% of traffic in specific times and city areas corresponds to drivers searching for a vacant parking spot [2]. The increasing number of vehicles over the years has caused significant parking issues in public areas, such as shopping malls and downtown area. This constitutes a source of atmospheric and acoustic pollution, as well as stress and wasted time for drivers seeking limited space availability. In an ideal Smart City, urban scale parking systems would be capable of providing real-time information about the availability of on-street parking spots.

A number of approaches for detecting on-street vacant parking spot have been suggested and examined. Generally, the solutions can be categorized into two groups: sensor-based and image-based [3]. Most of the sensor-based solutions entail the deployment of application-focused sensors, such as pneumatic road tubes, and magnetic or infrared sensors. However, the cost of deploying and maintaining these sensors prohibits practical use, except within certain cities [4]. In contrast, camera-based sensors are simple to install and less expensive to maintain [5]; In fact, closed-circuit surveillance cameras have already been widely deployed within urban environments for monitoring traffic activity at scale. While cameras are frequently used for traffic monitoring, their use for tracking urban parking management has been limited.

This paper proposes a camera-based parking space detection system that leverages cloud resources to track parking space availability across a large-scale city. The proposed system uses a single camera connected with an IoT edge device to monitor the status of on-street parking spots within its field of view. We have built this system to automatically recognize available parking spots by tracking different shape and size of occupying vehicles over time and effectively learning the available spaces after being placed in an area of a city.

There are several contributions of this work. First, we have built a complete system to dynamically learn the parking lot locations from parked vehicles using machine learning techniques. Using our system we are able to place our IoT edge device in unfamiliar parking areas and dynamically identify available parking spaces over time. We have developed a technique to check the occupancy of parking lots by using a probabilistic foreground classifier, which generates the probabilities of car occupancy using the average pixel value of the parking region. This approach provides a simple and efficient way for vacant parking space detection. Second, we have built a prototype based on a Raspberry Pi and computer vision techniques to identify parking spaces locally, mini-

978-1-5090-5249-0/17/$31.00 ©2017 IEEE

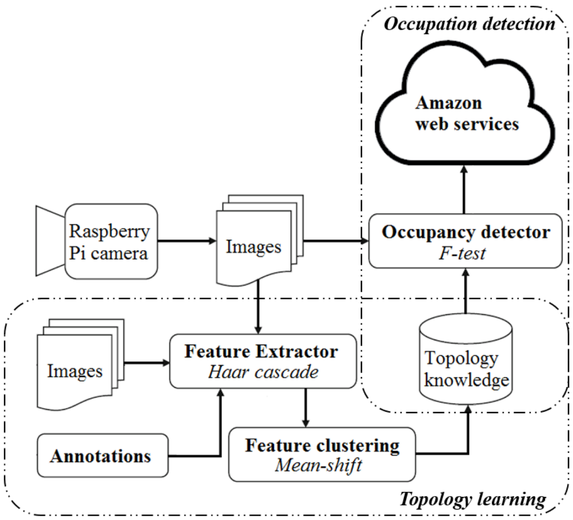


Figure 1: System Architecture

mizing the network overhead of offloading image analysis to a remote server. Although we identify parking spaces dynamically on the local edge device, we have developed a mechanism to aggregate parking space availability across a city by leveraging Amazon Web Services (AWS). Finally, we have developed a prototype AWS service that tracks parking space availability using input from IoT device input.

The remainder of the paper is organized as follows. Section 2 presents and describes the architecture of the proposed system. Section 3 presents the experiments conducted to test the system under different parking conditions, and the conclusions of this work are discussed in Section 4.

II. PARKING SPACE DETECTION SYSTEM

The proposed system is an extension of the image-based approach to vacant parking spot detection. It is designed to learn the topology of parking lot spaces as well as detect occupancy of the identified spaces. We have developed a complete pipeline for both of these primary functions. Our goal is to build a framework that easily and dynamically learns different parking space topologies. For instance, when the device is deployed on a specific unfamiliar location, the parking lot topology learning function immediately begins to recognize the parking regions within the field of camera view. Once the spaces are identified, the occupancy detector is used to generate the likelihood of each parking lots being occupied. This two steps procedure can be run iteratively and in parallel to improve the accuracy of the generated topology map as well as the occupancy prediction. The functional block diagram corresponding to this approach is depicted in figure 1.

A. Parking Lot Topology Learning

The proposed system uses a car-driven approach to recognize the parking spots [6]. Our strategy is to learn the parking spot from the location where parking events occur over time. Hence, a vehicle recognition function is used to drive the image analysis phase of the pipeline. Here we employ the Haar-featured object detection algorithm provided within the OpenCV library.

The Haar-feature classifier is a machine learning based approach where a cascade function is trained from a series of annotated figures [7]. In OpenCV, Haar-cascade training process is implemented in C++ and the best features are obtained using the adaptive boosting algorithm. We used the OpenCV cascade classifier trainer to generate a vehicle classifier from a set of annotated car images. The vehicle classifier then performs as a feature extractor in the system, where it collects the vehicle features from observed images and returns their locations as feature attributes. In order to learn the most frequently parked locations, we applied a feature clustering process based on Mean-shift to the high confidence feature attributes.

Being a centroid based algorithm, Mean-shift clustering updates centroid candidates to be the mean of the samples within a given bandwidth. The centroid candidates are then filtered in a post-processing stage to eliminate near duplicates to generate the final set of centroids [8]. Given a candidate centroid x_i for iteration t, the candidate is updated according to the equation 1:

$$x_i^{t+1} = x_i^t + m(x_i^t) \quad (1)$$

We denote $N(x_i)$ as the neighborhood of samples within a given distance around x_i and denote m as the mean-shift vector that is computed for each centroid that points towards a region of the maximum increase in the density of samples. $m(x_i)$ is computed using the equation 2, updating a centroid to be the mean of the samples within its neighborhood:

$$m(x_i^t) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_i}{\sum_{x_j \in N(x_i)} K(x_j - x_i)} \quad (2)$$

Function $K(x_j - x_i)$ dictates the size of the region to search through. Relying on a parameter $bandwidth$, it multiplies by a weight of 1 or 0 based on the distance of the sample to the centroid. The algorithm stops when the change in centroids is minimized. Subsequent labeling of a sample is performed by finding the nearest centroid for the given sample. This clustering algorithm does not require training and can be recomputed dynamically. The only parameter $bandwidth$ can be tuned during the experiment to fit the camera resolutions, as well as the feature sizes of the observed images.

Finally, centroids that contain the mean values of samples within each cluster are stored in a Topology knowledge file. The attributes of the centroid include the features average x-y coordinates and average size, as well as the features mean gray-scale density and maximum gray-scale standard deviation. These centroids are regarded as the representation of parking lots and they are used as the knowledge base during parking space occupation detection.

Algorithm 1: Algorithmic description of the Topology Learning stage. The parking spots is represented by vector s.

```
Data: Video feed of the parking
Result: Parking spots vector s
bandwidth=InitializeVehicleSize;
for frame=1...end do
    Vehicle=CascadeClassifier(frame);
    if VehicleSize>bandwidth then
        [x,y,h]=append(VehicleCoordinates);
    end
    s=MeanShift(bandwidth,[x,y,h]);
    return s;
end
```

B. Parking Lot Occupation Detection

In the occupation detection stage, the system performs automatic detection of changes in parking space occupancy, which we refer to as parking events. We use the images captured from the embedded camera to determine the likelihood of parking spots occupancy.

Because the computational power is limited on the Raspberry Pi, we used a simple and straightforward way to separate vacant and occupied parking lots. As the topology knowledge base provides location and gray scale information about the regions in the previous training step, we observed that vacant parking lot images typically have a lower gray-scale density variance than an occupied one. This result is intuitively logical as an empty parking lot picture typically consists of only the asphalt pavement, while an occupied parking lot contains a car with more complex shapes and colors. Hence, we applied an F-test to the on the variance of gray-scale density. The hypothesis for the test is that the occupied parking regions will have a similar or greater gray-scale density than the minimal one we observed during the topology training phase. The test statistic F is calculated by the equation 3:

$$F = \frac{s_1^2}{s_2^2} \tag{3}$$

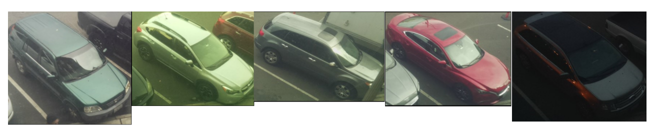
Since the F statistic follows the F distribution, we can calculate a likelihood of each parking lot is being occupied. A threshold can be applied to reject low probabilities. The final occupation result together with the observation data is then uploaded and stored in the AWS database. In a full deployment scenario, the system would continue to execute this detection process to ensure the occupation information in the database is kept up to date.

III. EXPERIMENTS & RESULTS

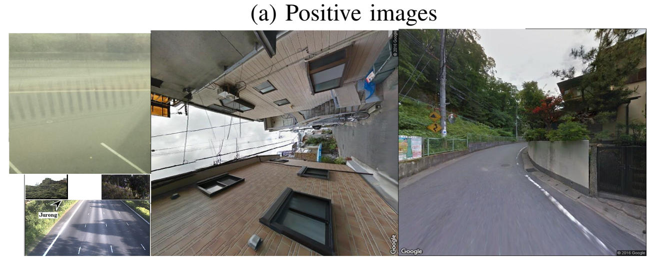
To evaluate the performance of the parking space detection system, we conducted a test on a local street nearby the University of Washington campus area. All the experiments in the test have been conducted on a Raspberry Pi 3 model B board with a 1.2 GHz quad-core ARM Cortex-A53 CPU and 1 GB 900MHz RAM. A Pi version 2 eight mega-pixel camera board is used for both the training and detection experiments.

A. Vehicle Classifier Training

To train a boosted cascade of classifiers on Haar-like features we needed a set of positive samples that contain actual vehicles and a set of negative images which contain no vehicles. To collect the positive samples, we used the Pi camera to take 1024*768 resolution images of cars from a front-right angle. Vehicle pictures are then manually cropped from the images. The negative samples are collected randomly from the Internet. To increase the size of our negative samples we also rotated some sample images horizontally. Finally, during our training process, we included 105 car images as the positive samples and 364 images that did not have vehicles as the negative sample set. Some of our positive and negative samples are shown in figure 2. We used the OpenCV library python wrapper to train the classifier. An XML file that contains the trained positive sample features is generated from our script in Raspberry Pi and is then used for vehicle detection.



(a) Positive images



(b) Negative images

Figure 2: Image examples in vehicle classifier training

B. Image Data Collection and Topology Learning

Parking space occupancy events typically occur at slow speed. Consequently, it is not necessary to stream image frames continuously from the Pi camera. Instead, we used a down-sampling strategy commonly used in previous video-based vacant parking spot detection work [9]. Also, due to

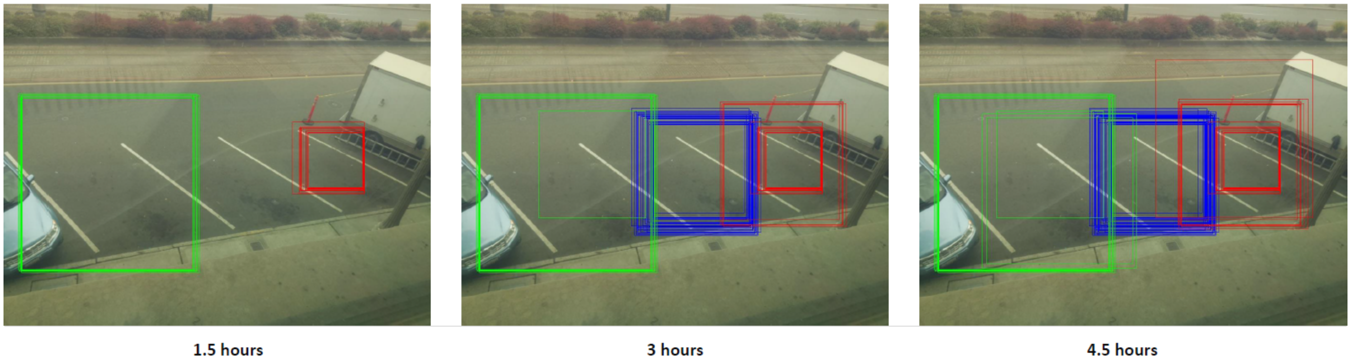


Figure 3: Vehicle sensing and clustering in training phase

the limited computing power of Raspberry Pi, we set our image collection and processing frequency to 0.2 Hz. While this frequency could be tuned, we found this processing frequency was adequate to learn the topology result within a reasonable time frame.

We applied the previously trained Haar-classifier to recognize vehicles within the image stream. Then the identified vehicle features are clustered to represent different parking spaces. The bandwidth is set to be 120 pixels. To test the performance of the approach, we started our system and ran the Topology Learning function for 4.5 hours. The system continually collected vehicle features from the surveillance scenes and clustered the features in the feature clustering step, as described earlier. After training for 1.5 hours, we inspected our collected features and plotted them on a vacant street background image, as shown in figure 3. The color label shows clustering group for each vehicle feature. And as we observe in figure 3, clustering accuracy improves with longer training times. During the initial 1.5 hours of training time the system is only able to identify two feature groups, whereas after 3 hours of training the system had gathered enough data to recognize all three parking spaces. All the features in each cluster are merged into a single average region based on the centroid which represents the particular parking region in view of the device. We call the identification of the parking spaces using this centroid approach the topology knowledge. Figure 4 below shows the three parking regions learned from our experiments. This shows that using this technique, we are able to leverage parking events to train a model to dynamically identify parking spaces. After this initial training period, our proposed system is able to generate a parking space map without manual specification. Although the required training time is limited to a few hours in our experiments, we note that this training time is highly dependent on the appearance of parked vehicles within the field of view of the IoT device.

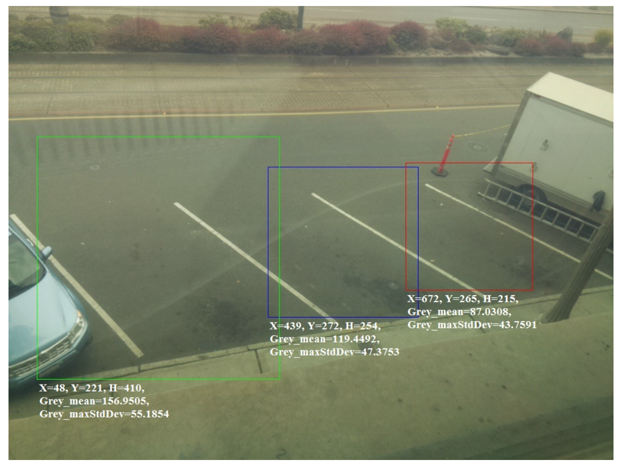


Figure 4: Centroids in topology knowledge

C. AWS Setup and Occupation Detection

Prior to evaluating the performance of the occupation detection, we established a connection between the Raspberry Pi and AWS, so the detection result can be easily restored and observed using AWS DynamoDB. The Raspberry Pi is connected to the AWS cloud through AWS IoT. In this project, we used the AWS IoT Device SDK for JavaScript on the Pi and implemented the system functionality using JavaScript. This also included several image analysis functions written in Python. Using the proper certificates and private key configuration within the script, we registered our Pi device with AWS IoT.

After the device registration, we set up a table named as ”ParkingSpaceState” in AWS DynamoDB to record parking space state over time. The table contains two columns, where column ”Timestamp” stores the time frame when each occupation state update is received. And column ”State” stores the update content from AWS IoT. Finally, we established a connection between AWS IoT to AWS DynamoDB by creating a Rule in AWS IoT. This Rule ingests all whole data received from Raspberry Pi into the ParkingSpaceState table.

Once we established the connection between our Raspberry Pi to AWS IoT to store parking space state data within AWS DynamoDB we conducted an additional experiment to measure the performance of our occupancy detector. Based on the topology knowledge base from previous experiences, we observed the minimal gray-scale density standard deviation to be 30.5232. Consequently, we used this as the threshold of observed density standard deviation, as well as for the denominator of the F-test. We set the alpha level for the F-test to 0.05. As it is time consuming to manually label all the occupation states for the frames captured during this experiment, we randomly examined our result among 30 discontinuous time points. And for those 90 detection results, we observed an accuracy of 91%. The detailed result is summarized in Table I. Figure 5 shows the occupation result in one of the sampled images.

	Predicted vacant	Predicted occupied
Vacant	22	6
Occupied	2	60

Table I: Table to test captions and labels

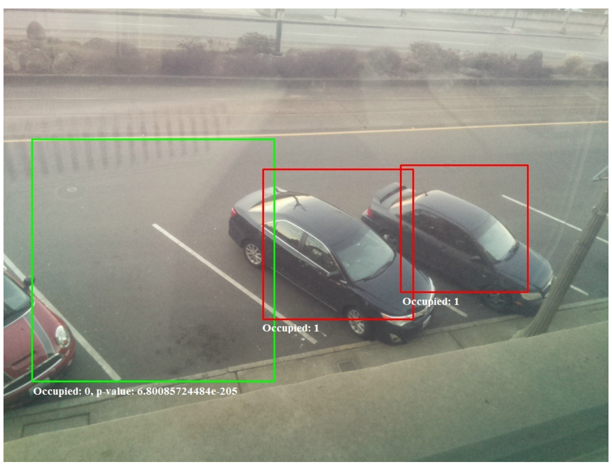


Figure 5: Occupation detection example result

IV. RELATED WORK

The use of image-based approaches for vacant parking space detection has attracted considerable interest from the research community over the years. A review on smart car parking systems is available in [11]. More recently, Huang and Wang proposes a general categorization of the existing vacant parking spot detection methods into two categories: car-driven and space-driven [12]. The car-driven method are based on car detection algorithms, and the detection of vacant parking spots by estimating the distances between detected cars. In contrast, space-driven methods focus on detecting the available parking spaces in a scene, including using a background and foreground classifier or parking lots detection methods.

Most of the vacant parking spot detection systems found in the literature utilize the space-driven approach. In [13], the authors proposed a four-step approach that included (i) parking region detection, (ii) Gaussian ground color model feature extraction, (iii) multi-class SVM training and (iv) Markov Random Field conflict classification correction system between contiguous parking spaces. Another space-driven approach was presented by True [14], which relied on parking space region extraction, followed by color histogram classification and Harris corner detection for vehicle feature identification. Tschentscher et al. proposed an evaluation of a pipeline composed of feature extraction (including color histograms, gradient histograms and Haar features), classification algorithms (including k-nearest neighbour, linear discriminant analysis and support vector machines) and temporal filtering for vacant parking spots detection [15].

Among the car-driven approaches to vacant parking spot detection, Masmoudi et al. [16] proposed their work to deal with occlusions between neighboring parking spots. A surface based model is proposed for parking spot model extraction, performing the detection of vehicles entering or leaving parking spots. Later, the same authors developed a real implementation of a video-based vacant parking spot detection system, proposing the architecture of a multi-agent parking lots management system to detect and localize vacant parking spots at a city level, as well as to detect anomalous behavior of drivers through the parking site [17]. Wang et al. presented a three-stage approach to vacant parking spot detection [18]. In the first stage, foreground and background models were generated for determining parking spot occupancy status. Then the difference between successive video frames was used to determine whether a car is finally parked on a given spot. And third, an adaptive decision threshold is applied to get the final occupancy status.

Compared to the classic car-driven or space-driven classification of vacant parking lot detection approaches, our proposed system differs with the previous ones in several ways. First, most of the space-driven approaches require a manual pre-processing step to identify the parking sites or region of interest (ROI). Our approach uses a robust machine learning method to automatically recognize those regions, eliminating the need for user specification. This enables our system to simply be placed in any location and dynamically identify parking space locations over time. Unlike other methods, our approach utilizes methods with lower computational complexity than most of the car-driven approaches. Consequently, our approach is suitable for deployment on IoT devices. Overall, our work expands the current applicable limit of vacant parking lot detection systems, as well as presents a novel and efficient solution for the parking lot occupancy detection problem.

V. CONCLUSION

In this paper, we have designed and evaluated an IoT-based, computer vision approach using a single camera that can dynamically identify parking spaces and detect occupancy with high accuracy. The proposed system builds on well-established vehicle feature extraction and clustering approaches to perform parking spot topology learning, as well as a statistical occupancy detector to perform the parking spot status detection with low computational complexity.

Our experiments show that with additional training time, we are able to accurately identify parking lot topologies dynamically. Using our occupation detector we are able to process frames in near real-time with an accuracy of 97%. Moreover, it is important to note that these results have been obtained on a real on-street parking area with challenging daylight illumination conditions, highlighting the robustness of our proposed system.

Moreover, we have shown this system can be built with commodity IoT devices deployed at the edge, such as the Raspberry Pi. We have also shown that by leveraging the AWS cloud, we can store, aggregate, and leverage newly deployed IoT devices to build a complete city-wide parking lot management system.

Although our system exploits topology learning and occupation detecting function using pre-trained vehicle classifier and statistical tests, we believe using more robust and more sensitive methods such as the trajectory analysis may prove to reduce the required training time and increase accuracy [10]. We plan to evaluate the integration of such approaches in future work. Additionally, we plan to evolve the capabilities of cloud service to manage multiple IoT devices, which can provide a parking state ensemble for an urban area of a Smart City from the input through different functional IoT detectors.

Finally, it is important to note that using camera-based parking space occupancy systems at night time is challenging. Although our approach performs well during day time experiments, we plan to expand our experiments to night time operation to better characterize the use of alternative, vision-based algorithms.

REFERENCES

[1] Zanella, Andrea, et al. "Internet of things for smart cities." IEEE Internet of Things journal 1.1 (2014): 22-32.

[2] Schaller, Bruce. "Curbing Cars: Shopping, Parking, and Pedestrian Space in SoHo." Transportation Alternatives (2006).

[3] Bong, D. B. L., K. C. Ting, and K. C. Lai. "Integrated approach in the design of car park occupancy information system (COINS)." IAENG International Journal of Computer Science 35.1 (2008): 7-14.

[4] Idris, M. Y. I., et al. " park system: a review of smart parking system and its technology." Inf. Technol. J 8.2 (2009): 101-113.

[5] Wolff, Joerg, et al. "Parking monitor system based on magnetic field senso." Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE. IEEE, 2006.

[6] Mrmol, Elena, and Xavier Sevillano. "QuickSpot: a video analytics solution for on-street vacant parking spot detection." Multimedia Tools and Applications 75.24 (2016): 17711-17743.

[7] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[8] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on pattern analysis and machine intelligence 24.5 (2002): 603-619.

[9] Jermsurawong, Jermsak, et al. "Car parking vacancy detection and its application in 24-hour statistical analysis." Frontiers of Information Technology (FIT), 2012 10th International Conference on. IEEE, 2012.

[10] Ng, Lih Lin, and Hong Siang Chua. "Vision-based activities recognition by trajectory analysis for parking lot surveillance." Circuits and Systems (ICCAS), 2012 IEEE International Conference on. IEEE, 2012.

[11] Idris, M. Y. I., et al. " park system: a review of smart parking system and its technology." Inf. Technol. J 8.2 (2009): 101-113.

[12] Huang, Ching-Chun, and Sheng-Jyh Wang. "A hierarchical bayesian generation framework for vacant parking space detection." IEEE Transactions on Circuits and Systems for Video Technology 20.12 (2010): 1770-1785.

[13] Wu, Qi, et al. "Robust parking space detection considering inter-space correlation." Multimedia and Expo, 2007 IEEE International Conference on. IEEE, 2007.

[14] True, Nicholas. "Vacant parking space detection in static images." University of California, San Diego 17 (2007).

[15] Tschentscher, M., et al. "Comparing image features and machine learning algorithms for real-time parking space classification." Computing in Civil Engineering (2013). 2013. 363-370.

[16] Masmoudi, Imen, Ali Wali, and Adel M. Alimi. "Parking spaces modelling for inter spaces occlusion handling." 22nd Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic. 2014.

[17] Masmoudi, Imen, et al. "Multi agent parking lots modelling for anomalies detection while parking." IET Computer Vision 10.5 (2016): 407-414.

[18] Wang, Ying, et al. "A Vision-Based Method for Parking Space Surveillance and Parking Lot Management." International Conference on Image and Graphics. Springer International Publishing, 2015.