

**AN IMAGE-BASED VACANT PARKING
DETECTION SYSTEM USING HYBRID FEATURES
AND A BAYESIAN CLASSIFIER**

BY

JUNZHAO LIU

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

APRIL 2014

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **JUNZHAO LIU** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

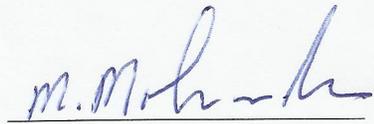


Dr. Ali Al-Shaikhi
Department Chairman



Dr. Salam A. Zummo
Dean of Graduate Studies

5/5/14
Date



Dr. Mohammed Mohandes
(Advisor)



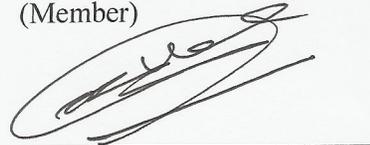
Dr. Mohammed Deriche
(Co-Advisor)



Dr. Adil Balghonaim
(Member)



Dr. Hassan Ragheb
(Member)



Dr. Abdelmalek Zidouri
(Member)

© Junzhao Liu

2014

DEDICATION

Affectionately dedicated to

My parents, my teachers and my friends

ACKNOWLEDGMENTS

The author would like to thank KFUPM (Office of Planning and Quality) for the support of the whole project.

I would like to express my deep appreciation to my thesis advisor Dr. Mohammed Mohandes, and my thesis co-advisor Dr. Mohammed Deriche for their ideas of the starting the smart parking project, their guidance, their supports, their encouragements and consistent help. Thanks are due to my thesis committee members Dr. Adil Balghonaim, Dr. Hassan Ragheb and Dr. Abdelmalek Zidouri for their comments, time and efforts in reviewing this thesis.

Also, I would like to thank my parents for always supporting with my study and research. I would like to thank Dr. Zhenbiao Li, Dr. Huanxin Chen and Dr. Zummo for introducing KFUPM to me. I would like to thank all the professors who taught me for their imparting knowledge in different courses and seminars. I would like to thank my classmates and colleagues for the academic discussions which help me come up with different ideas and innovations, especially Mr. Aliyu for helping me find out a naming convention mistake in Weka, and Mr. Alkhodary for his significant suggestions. I would like to thank all my friends, mostly those who helped me adapt to the life in a new environment and who participated in the database of this work.

TABLE OF CONTENTS

DEDICATION	IV
ACKNOWLEDGMENTS	V
TABLE OF CONTENTS	VI
LIST OF TABLES.....	X
LIST OF FIGURES.....	XI
LIST OF ABBREVIATIONS	XX
ABSTRACT	XXI
ملخص الرسالة.....	XXII
1 CHAPTER 1 INTRODUCTION	1
1.1 Background of the Project.....	1
1.2 Statement of the Problem.....	2
1.3 Contributions.....	2

1.4	Thesis Structure	4
2	CHAPTER 2 LITERATURE REVIEW	5
2.1	Smart Parking System	5
2.2	A Review of Image Features	8
2.2.1	Histogram-based Features.....	10
2.2.2	Co-occurrence Matrix	11
2.2.3	Coarseness	12
2.2.4	Contrast Measure.....	13
2.2.5	Directionality.....	13
2.2.6	Autocorrelation Features.....	15
2.2.7	Law's Texture Energy Measures	15
2.2.8	Gabor Filter	16
2.2.9	Hu's Moments	17
2.2.10	Haralick Texture Features	18
2.3	Texture Segmentation.....	22
2.3.1	Summary	23
3	CHAPTER 3 THE FEATURES USED IN OUR SYSTEM.....	24

3.1	Edge Information	26
3.1.1	Edge Counting Method	26
3.1.2	Boundary Counting Method	28
3.2	Background Information and Foreground Acquirement	29
3.2.1	Background Generating	30
3.2.2	Foreground Acquirement	31
3.3	Summary.....	32
4	CHAPTER 4 OTSU METHOD AND THE BAYESIAN CLASSIFIER	33
4.1	Otsu Method and Threshold Acquirement	33
4.2	The Bayesian Classifier	34
4.3	The Application of Naive Bayesian Classifier using Weka	36
4.4	Summary.....	39
5	CHAPTER 5 EXPERIMENTAL RESULTS AND SYSTEM PERFORMANCE	40
5.1	Vacant Parking Detection for a Fixed Parking Region	42
5.1.1	Vacant Parking Detection for the Case of All-vacant Parking Region	43
5.1.2	Vacant Parking Detection for Mixed-State Parking Region	51
5.1.3	Processing of the Parking Image with no Vacant Parking Space.....	65

5.2	Bayesian Classification Using the Weka Package.....	72
5.2.1	Database Setting up	72
5.2.2	The Performance of Bayesian Classifier Using the Weka Package.....	82
5.3	Extension to Parking Images under Different Conditions	95
5.3.1	Processing of a Parking Image Taken by a Wide Angle Camera.....	96
5.3.2	Processing of an Outdoor Parking Image	103
5.3.3	Processing of the Parking Image Containing a Covered Car	106
5.4	Summary.....	109
6	CHAPTER 6 CONCLUSIONS AND FUTURE WORK	110
	REFERENCES	112
	VITAE	116

LIST OF TABLES

Table 2.1: Summary of Texture Features in different classes	9
Table 3.1: Prewitt operator	26
Table 3.2: Edge counting and object counting techniques for Figure 3.3	29
Table 3.3: Detection results using the foreground/background method for Figure 3.3	31
Table 5.1: Summary of Figure 5.1	50
Table 5.2: Summary of Figure 5.12	57
Table 5.3: Summary of Figure 5.23	64
Table 5.4: Summary of Figure 5.34	71
Table 5.5: Image size information	73
Table 5.6: The entire database used in the project	73
Table 5.7: The seven spaces parking lot detection results of Figure 5.65	102
Table 5.8: Edge pixel counting, object counting, and F/B method of Figure 5.77	105
Table 5.9: Information Summary of Figure 5.82	108

LIST OF FIGURES

Figure 2.1: The 256 level histograms	10
Figure 2.2: The spatial co-occurrence calculations.....	12
Figure 3.1: Block diagram of the proposed system.....	24
Figure 3.2: Flow chat of processing one input image.....	25
Figure 3.3: Input image example 1	27
Figure 3.4: Segmentation of space 2.....	27
Figure 3.5: Edge detection of Figure 3.3	27
Figure 3.6: Edge pixel counting for all the four parking spaces of Figure 3.3.....	28
Figure 3.7: Object counting result of space 1 in Figure 3.3	28
Figure 3.8: Histogram for Figure 3.3 using 25 bins.....	30
Figure 3.9: Foreground information for Figure 3.3.....	31
Figure 4.1: Data Preprocess using Weka.....	38
Figure 4.2: Naive Bayes Classifier.....	38
Figure 5.1: The input image of a fixed parking region with three vacant spaces and zero occupied space	43
Figure 5.2: Edge detection of a fixed parking region with three vacant spaces and zero occupied space	44
Figure 5.3: Edge pixel counting for space 1 of a fixed parking region with three vacant spaces and zero occupied space, space 1 containing 64 edge pixels, detected as a vacant space.....	45

Figure 5.4: Edge pixel counting for space 2 of a fixed parking region with three vacant spaces and zero occupied space, space 2 containing 40 edge pixels, detected as a vacant space.....	45
Figure 5.5: Edge pixel counting for space 3 of a fixed parking region with three vacant spaces and zero occupied space, space 3 containing 94 edge pixels, detected as a vacant space.....	46
Figure 5.6: Object counting for space 1 of a fixed parking region with three vacant spaces and zero occupied space, space 1 containing 14 objects, detected as a vacant space.....	46
Figure 5.7: Object counting for space 2 of a fixed parking region with three vacant spaces and zero occupied space, space 2 containing 7 objects, detected as a vacant space.....	47
Figure 5.8: Object counting for space 3 of a fixed parking region with three vacant spaces and zero occupied space, space 3 containing 14 objects, detected as a vacant space.....	47
Figure 5.9: Histogram of a fixed parking region with three vacant spaces and zero occupied space.....	48
Figure 5.10: Histogram using 25 bins of a fixed parking region with three vacant spaces and zero occupied space.....	48
Figure 5.11: The foreground image generation of a fixed parking region with three vacant spaces and zero occupied space.....	49
Figure 5.12: The input image of a fixed parking region with two vacant spaces and one occupied space.....	51

Figure 5.13: Edge detection of a fixed parking region with two vacant spaces and one occupied space	52
Figure 5.14: Edge pixel counting of a fixed region with two vacant spaces and one occupied space, space 1 containing 1730 edge pixels, detected as an occupied space	52
Figure 5.15: Edge pixel counting for space 2 of region with two vacant spaces and one occupied space, space 2 containing 84 edge pixels, detected as a vacant space	53
Figure 5.16: Edge pixel counting for space 3 of region with two vacant spaces and one occupied space, space 3 containing 84 edge pixels, detected as a vacant space	53
Figure 5.17: Space 1 object numbers of region with two vacant spaces and one occupied space, space 2 containing 83 objects, detected as an occupied space	54
Figure 5.18: Space 2 object numbers of region with two vacant spaces and one occupied space, space 2 containing 12 objects, detected as a vacant space	54
Figure 5.19: Space 3 object numbers of region with two vacant spaces and one occupied space, space 3 containing 11 objects, detected as a vacant space	54
Figure 5.20: Histogram of a fixed parking region with two vacant spaces and one occupied space	55
Figure 5.21: Histogram using 25 bins of a fixed parking region with two vacant spaces and one occupied space	55
Figure 5.22: The foreground image generation of a fixed parking region with two vacant spaces and one occupied space	56

Figure 5.23: The input image of a fixed parking region with one vacant space and two occupied spaces	58
Figure 5.24: Edge information of a fixed parking region with one vacant space and two occupied spaces	59
Figure 5.25: Edge pixel counting for space 1 of a fixed parking region with one vacant space and two occupied spaces, space 1 containing 76 edge pixels, detected as a vacant space.....	59
Figure 5.26: Edge pixel counting for space 2 of a fixed parking region with one vacant space and two occupied spaces, space 2 containing 1116 edge pixels, detected as an occupied space.....	60
Figure 5.27: Edge pixel counting for space 3 of a fixed parking region with one vacant space and two occupied spaces, space 3 containing 1332 edge pixels, detected as an occupied space.....	60
Figure 5.28: Space 1 object numbers of a fixed parking region with one vacant space and two occupied spaces, space 1 containing 6 objects, detected as a vacant space	61
Figure 5.29: Space 2 object numbers of a fixed parking region with one vacant space and two occupied spaces, space 2 containing 71 objects, detected as an occupied space	61
Figure 5.30: Space 3 object numbers of a fixed parking region with one vacant space and two occupied spaces, space 2 containing 83 objects, detected as an occupied space	61

Figure 5.31: Histogram of a fixed parking region with one vacant space and two occupied spaces.....	62
Figure 5.32: Histogram using 25 bins of a fixed parking region with one vacant space and two occupied spaces	62
Figure 5.33: The foreground image generation of a fixed parking region with one vacant space and two occupied spaces.....	63
Figure 5.34: The input image of a fixed parking region with zero vacant space and three occupied spaces	65
Figure 5.35: Edge information of a fixed parking region with zero vacant space and three occupied spaces	66
Figure 5.36: Edge pixel counting for space 1 of a fixed parking region with zero vacant space and three occupied spaces, space 1 containing 1469 edge pixels, detected as an occupied space.....	66
Figure 5.37: Edge pixel counting for space 2 of a fixed parking region with zero vacant space and three occupied spaces, space 2 containing 1058 edge pixels, detected as an occupied space.....	67
Figure 5.38: Edge pixel counting for space 3 of a fixed parking region with zero vacant space and three occupied spaces, space 3 containing 1205 edge pixels, detected as an occupied space.....	67
Figure 5.39: Space 1 object numbers of a fixed parking region with zero vacant space and three occupied spaces, space 1 containing 74 objects, detected as an occupied space	68

Figure 5.40: Space 2 object numbers of a fixed parking region with zero vacant space and three occupied spaces, space 2 containing 82 objects, detected as an occupied space	68
Figure 5.41: Space 3 object numbers of a fixed parking region with zero vacant space and three occupied spaces, space 3 containing 82 objects, detected as an occupied space	68
Figure 5.42: Histogram of a fixed parking region with zero vacant space and three occupied spaces	69
Figure 5.43: Histogram using 25 bins of a fixed parking region with zero vacant space and three occupied spaces	69
Figure 5.44: The foreground generation of a fixed parking region with zero vacant space and three occupied spaces	70
Figure 5.45: Statistic of 303 normal parking spaces	83
Figure 5.46: Output of Weka naive Bayes classifier for the feature of the number of edge pixels	84
Figure 5.47: ROC curve of the output in Figure 5.46	84
Figure 5.48: Output of Weka naive Bayes classifier for the feature of object number	85
Figure 5.49: ROC curve of the output in Figure 5.48	85
Figure 5.50: Output of Weka naive Bayes classifier for the feature of FMPV	86
Figure 5.51: ROC curve of the output in Figure 5.50	86
Figure 5.52: Output of Weka naive Bayes classifier for all three features	87
Figure 5.53: Statistic with special cases	88

Figure 5.54: Weka naive Bayes classifier for the feature of the number of edge pixels (with special cases).....	89
Figure 5.55: ROC curve of the output in Figure 5.54.....	89
Figure 5.56: Weka naive Bayes classifier for the feature of object number (with special cases).....	90
Figure 5.57: ROC curve of the output in Figure 5.56.....	90
Figure 5.58: Weka naive Bayes classifier for the feature of FMPV (with special cases) .	91
Figure 5.59: ROC curve of the output in Figure 5.58.....	91
Figure 5.60: Weka naive Bayes classifier for dynamic background value (with special cases).....	92
Figure 5.61: ROC curve of the output in Figure 5.60.....	92
Figure 5.62: Output of Weka naive Bayes classifier for three main features (with special cases).....	93
Figure 5.63: Output of Weka naive Bayes classifier for three main features, and dynamic background value (with special cases).....	94
Figure 5.64: A raw parking image taken by wide angle camera.....	96
Figure 5.65: The input image from a wide angle camera	97
Figure 5.66: Results of the dynamic parking space selection of the input image from a wide angle camera	97
Figure 5.67: Edge pixel counting for space 1 of the input image from a wide angle camera, containing 2817 edge pixels, detected as an occupied space.....	98
Figure 5.68: Edge pixel counting for space 2 of the input image from a wide angle camera, containing 193 edge pixels, detected as a vacant space	98

Figure 5.69: Edge pixel counting for space 3 of the input image from a wide angle camera, containing 3961 edge pixels, detected as an occupied space.....	98
Figure 5.70: Edge pixel counting for space 4 of the input image from a wide angle camera, containing 4328 edge pixels, detected as an occupied space.....	99
Figure 5.71: Edge pixel counting for space 5 of the input image from a wide angle camera, containing 9603 edge pixels, detected as an occupied space.....	99
Figure 5.72: Edge pixel counting for space 6 of the input image from a wide angle camera, containing 8130 edge pixels, detected as an occupied space.....	99
Figure 5.73: Edge pixel counting for space 7 of the input image from a wide angle camera, containing 6634 edge pixels, detected as an occupied space.....	100
Figure 5.74: Histogram of the input image from a wide angle camera	100
Figure 5.75: The foreground generation of the input image from a wide angle camera .	101
Figure 5.76: Stem figure of doubled FMPV of the input image from a wide angle camera	101
Figure 5.77: Input image (outdoor) with one occupied spots and three vacant spaces ...	103
Figure 5.78: Edge detection of an outdoor parking with one occupied spots and three vacant spaces.....	103
Figure 5.79: Histogram using 25 bins of an outdoor parking with one occupied spots and three vacant spaces	104
Figure 5.80: The foreground information of an outdoor parking with one occupied spots and three vacant spaces	104
Figure 5.81: Stem figure of double FMPV of an outdoor parking with one occupied spots and three vacant spaces	105

Figure 5.82: Input image containing a special case (a five-spaces parking from building 64).....	106
Figure 5.83: Edge information of a parking containing a special case (a five-spaces parking from building 64).....	107
Figure 5.84: Foreground information of a parking containing a special case (a five-spaces parking from building 64).....	107

LIST OF ABBREVIATIONS

FMPV	:	Foreground Mean Pixel Value
ROC	:	Receiving Operating Characteristic
AR	:	autoregressive model
ARMA	:	Autoregressive Moving Average model
DDWT	:	Discrete Dyadic Wavelet Transform
MPD	:	Multi-scale Pyramid Decomposition
QMF	:	Quadrature Mirror Filter
DCT	:	Discrete Cosine Transform
ASM	:	Angular Second Moment
O	:	Occupied Parking Space
V	:	Vacant Parking Space
F/B Method	:	Foreground/Background Method

ABSTRACT

Full Name : Junzhao Liu
Thesis Title : An Image-based Vacant Parking Detection System Using Hybrid
Features and a Bayesian Classifier
Major Field : Signal Processing and Image Processing
Date of Degree : April 2014

Parking has become a major issue of concern in city centers especially around malls, stadiums, transportation centers, exhibition centers, etc. Finding a vacant space to park in in large cities may take more than half an hour. To solve this problem, we are witnessing a huge interest in the development of smart parking systems, which can efficiently manage car parks and hence minimize drivers' time in finding a vacant spot. Smart parking systems have traditionally been categorized into either sensor-based or image-based systems. Sensor based systems require the deployment of a set of sensors at each car park, leading to a cumbersome infrastructure, like wiring, communications, and synchronization. On the other hand, image-based systems require simpler infrastructure as a single camera can monitor several car spots. Furthermore, cameras can transmit acquired images wirelessly for processing. In this study, we present a new image texture feature analysis for vacant parking detection which uses segmentation and texture features to identify vacant car spots. The features considered here include texture, edge information, histogram, and other statistical features. The combination of these features at the decision level provides robust identification performance. The original database of this work comes from the parking areas in KFUPM. Our numerous experiments showed that the proposed system can achieve a recognition accuracy of more than 90% in both indoor and outdoor parking.

ملخص الرسالة

الاسم الكامل: ليو جنزهاو

عنوان الرسالة: نظام الكشف عن مواقف السيارات الفارغة باستخدام العلامات الهجينية ومصنف بايزن

التخصص: معالجة إشارات رقمية – معالجة صور رقمية

تاريخ الدرجة العلمية: أبريل 2014

في الآونة الأخيرة أصبحت مشكلة توفر مواقف للسيارات مثيرة للقلق في مراكز المدن وخصوصاً حول مراكز التسوق و الملاعب و مراكز النقل، و مراكز المعارض ، وما إلى ذلك. العثور على مساحة شاغرة لمواقف سيارة في المدن الكبيرة قد يستغرق أكثر من نصف ساعة. إننا نشهد في الوقت الراهن اهتماما كبيرا في تطوير أنظمة وقوف السيارات الذكية ، وكيفية إدارة مواقف السيارات بكفاءة، و بالتالي تقليل وقت السائقين في إيجاد موقف فارغ.

جرت العادة على تصنيف نظم وقوف السيارات الذكية إما إلى أنظمة تعتمد على المستشعرات أو أنظمة تعتمد على الصور الرقمية. الأنظمة المعتمدة على المستشعرات تتطلب نشر مجموعة من أجهزة الاستشعار في كل موقف للسيارات، مما يؤدي إلى بنية تحتية معقدة، مثل الأسلاك، والاتصالات ، والتزامن. من ناحية أخرى، الأنظمة المعتمدة على الصورة الرقمية تتطلب أبسط بنية تحتية وكاميرا واحدة يمكن رصد العديد من مواقع السيارات. علاوة على ذلك، يمكن للكاميرات نقل الصور لاسلكيا للمعالجة.

في هذه الدراسة، فإننا نقدم تحليلاً جديداً للصورة للكشف عن مواقف شاغرة والتي تستخدم تجزئة ميزات الصورة لتحديد المواقف الشاغرة. يمكن اعتبار مميزات الصورة لتشمل التلوين، و معلومات الحافة، الرسم البياني، والميزات الإحصائية الأخرى. الجمع بين هذه الميزات على مستوى القرار يوفر أداء قويا في الكشف عن المواقف الشاغرة. البيانات والصور المعتمد عليها في هذا العمل تأتي من مواقف للسيارات في جامعة الملك فهد. فقد أظهرت العديد من التجارب أن النظام المقترح يمكن تحقيق دقة كشف تفوق 90 % في كل من وقوف السيارات في الأماكن المغلقة و الهواء الطلق.

CHAPTER 1

INTRODUCTION

1.1 Background of the Project

Parking has become a major issue of concern in city centers especially around malls, stadiums, transportation centers, exhibition centers, etc. Finding a vacant space to park in Riyadh, for example, may take more than half an hour. To solve this problem, we are witnessing a huge interest in the development of smart parking systems, which can efficiently manage car parks and hence minimize drivers' time in finding a vacant spot. Smart parking systems have traditionally been categorized into either sensor-based or image-based systems. Sensor based systems require the deployment of a set of sensors at each car park, leading to a cumbersome infrastructure, like wiring, communications, and synchronization. On the other hand, image-based systems require simpler infrastructure as a single camera can monitor several car spots. Furthermore, cameras can transmit acquired images wirelessly for processing. In this study, we develop an image-based parking system which uses segmentation and texture features to identify vacant car spots. The features considered here include texture, edge information, histogram, and other statistical features. We show that the combination of these features at the decision level provides robust identification performance. Our focus is to develop a system that can be deployed in the different parking areas at KFUPM.

In this work, we first start with some initial experiments in processing parking spot images. With a basic camera, we are able to cover three or four parking spaces. For our advanced experiments, a wide angle camera was used to get images that cover a large number of car spots. Such cameras are usually deployed when used in large parking areas both indoors and outdoors.

1.2 Statement of the Problem

Our main problem is to develop a vacant parking detection system which is robust, cost efficient, accurate and response in a short time. To achieve this goal, numerous experiments using different texture features were implemented.

In particular, the thesis objectives are as follows:

- 1) Developing a set of optimal features for distinguishing vacant and occupied parking spaces.
- 2) Developing a Bayesian framework for classifying parking spaces.
- 3) Intensive testing of the proposed system.
- 4) Deploying the developed system in a practical setup at KFUPM.

1.3 Contributions

In this project, one of the most time consuming tasks was the development of database of parking spaces. Since we could not find publically available database, we had to develop our own database. Over last year, our database kept growing with more images taken from different parking at KFUPM. We tried different types of cameras, such as normal digital camera which can cover four parking spots, wide-angle camera which can cover

seven parking spots, and mobile video cameras. Finally, more than 100 images were collected to constitute our database.

Before the development of our vacant parking detection system, it is necessary to select the features used in our system and related image processing method. We went over early works of image-based vacant parking detection systems to verify the performance of some common used features. Then we reviewed the documents about image features (color feature, texture features and shape feature), image segmentation, image matching and feature extracting. Based on the literature review, three main features are selected. The feature of the number of edge pixels is similar to the work in [1]. Then combine the edge detection method in [2] and the segmentation method in [3], we defined the different feature of object numbers. Inspired from the foreground image in [4] and based on the concept of contrast feature, the FMPV is defined. After comparing the final decision of voting function in [1] and hierarchical Bayesian generation in [5], we selected Bayesian classifier to deal with the entire database. During the testing of the three main features used in our system, we found that some other features also can be used to verify the state of a parking space or increase the recognition accuracy when adding to the naïve Bayes classifier. We take such features as secondary features. After implementing our vacant parking detection system using hybrid features, we applied our system under different conditions and the performance was still good. This procedure helps us to improve our system to a robust and efficient system.

1.4 Thesis Structure

The rest of the thesis consists of five chapters. In Chapter 2, firstly a review of different smart parking system is given. Then an introduction of different features is presented. After that the feature segmentation techniques are described. In Chapter 3, our image based vacant parking detection system using hybrid features is introduced. We show the structure of the system and how to extract the features used in our system. In Chapter 4, a brief introduction of related method is presented, containing Otsu Method and Bayesian classifier. In Chapter 5, experimental results are presented in details. The outcomes contain images of different features, tables of detection summaries, entire database and system performance using the Bayesian classifier. In Chapter 6, the conclusion of the thesis is given with some potential future research directions.

CHAPTER 2

LITERATURE REVIEW

2.1 Smart Parking System

Several methods for image-based parking lot detection have been discussed in the literature. Banerjee et al. developed a system using the Prewitt operator to detect edges of cars at the entrance of the parking, then counts the number of incoming cars by image matching[2]. In their experiment, they used two independent cameras, each covering 10 parking spaces. The drivers are guided to the parking region that has vacancies. The system is easy, simple, and cheap, however, it requires multiple cameras to cover a parking area since the view of one camera is limited. To solve this problem, Chen, et al. developed a system that covers a wide-area parking lot with a four-camera system[6]. The scenes from the different cameras are merged by affine transformation. Then, all the views are integrated together and analyzed. The system integrates the detected objects from different cameras onto a panoramic scene. Each parking space is modeled as an ellipse. Then, an edge based scheme and a color-based model are used to detect whether a parking space is occupied or not. For the edge based scheme, the Canny edge detector [7] was used while for the color based model, the RGB statistical parameters (mean and variance) of each pixel of the vehicle area and the background, were used. The system achieved an accuracy of 94.7% with the color-ellipse method.

It is worth noting that indoor parking lot detection systems [8] are different from outdoor parking detection systems [9]. As the illumination has a major influence on image-based systems deployed outdoors, we have to consider the glaring sun light and dark shadows in the daytime, low light density and back lighting at night. Usually an outdoor detection system could also be used indoor while the reverse is not as accurate. An example of a Car Park Occupancy Information System for outdoors is discussed in [3]. It finds the boundary of the parking space as an initial step, then a feature extraction stage is used before deciding if the parking space is occupied or not. To avoid the false detection caused by the shadow of a vehicle, median filtering and Sobel edge detection are used.

Based on the size of the covered area, either single space [2] or multiple space [10][11] detection techniques have been used. Single space systems are more accurate but costly to deploy in large areas. Lin et al. [4] developed a vision-based parking lot management system that generates an adaptive background image first. Then parking spaces with large pixel differences compared to the adaptive background are declared as occupied. After a shadow detection process, the total number of parked cars is decided and the information is conveyed to drivers. Lixia et al. [1] used image segmentation and local binary patterns (LBP) to detect vacant parking space. If the parking space is vacant, the corresponding image is shown to contain a relatively small number of blocks while an image of a car always contains large number of blocks. The system achieved 97% detection accuracy but since it focused only on single parking slot, it is shown to be expensive to deploy in practical situations. In [5], Huang and Wang developed a 3-D scene model for vacant parking space detection. The detection was based on a solar movement model and the illuminant probability. It determines the sunlight direction first, then uses it to generate a

vehicle and a shadow models. Based on these two, an intensity model is built. Finally the vacant parking spaces of the whole parking are detected by a classifier generated from the intensity model and the illumination model.

Image segmentation has also been used as a popular method for vacant parking detection [12]. In [13], car feature point detection and color histogram classification are used, with the system recognition accuracy above 80% for both day images and night images. In [14], Fabian's algorithm combines the box model, a quad tree decomposition and weight map. Yusnita et al. [15] developed a system that deal with car models. A small circle is placed at the center of each parking space. By tracking the object shape and boundaries, the parking status is decided.

Our analysis of the current literature showed that most proposed systems focus on one set of features. Such features could be either edge-based, objects or blocks-based, or based on statistical features such as pixel difference, etc. In our work, we propose to use a combination of features. Such a combination is expected to provide a more robust vacant parking detection system which can be deployed in a spectrum of situations and environments.

2.2 A Review of Image Features

Before developing the optimal set of features for vacant parking space detection, we start by surveying the different features that can be used for this purpose. In image processing, there are three basic sets of features: Color Features, Texture Features and Shape Features. The feature extraction and matching for Color Features is usually based on color histogram. The feature extraction for Shape Features mainly consists of two branches: contour feature and region feature. For Texture Features, a number of approaches can be used. Texture Features are used as quantifying metrics of perceived texture. Texture features provide information on the arrangement of patterns and colors. [16]

For humans, we have an extended visual system to distinguish the details of what we see. But a machine vision is totally different as it requires models to represent texture. When dealing with texture features, there are 4 major issues that need to be considered: Feature extraction, texture discrimination, texture classification and texture shape.[17]

Texture features include histogram information, co-occurrence matrix, coarseness, contrast, directionality, autocorrelation, power spectrum, frequency, phases, linearity, randomness, image moments, uniformity etc.[18]

Table 2.1 shows the different classes of texture features used in practice. In what follows, we discuss the texture features that can be used for our applications

Table 2.1: Summary of Texture Features in different classes

Statistical Methods	Model Based Methods	Structural Methods	Transform Methods
Gray Level Co-occurrence Matrix[19]	Autoregressive (AR) Model[18]	Syntactic Approach[20]	Gabor Transform [21][22][23][24][25]
Gray Level Difference Statistics[26]	Autoregressive Moving Average (ARMA) model[27]	Morphological Image Processing[28]	Discrete Dyadic Wavelet Transform (DDWT)[29]
			Multiband Wavelet Transform[30]
Local Statistics[31]	Simultaneous Autoregressive Model[32]	Laws Texture Features[33]	Multi-scale Pyramid Decomposition (MPD)[34]
Gray Scale Gradient Matrix[35]			Tree-structured wavelet Decomposition[36]
Haralick Features	Long-range Dependence[37]	Eigen Filter/ Characteristic Filter[38]	Ridgelet Transform & Curvelet Transform[39]
			Radon Transform[40]
Semivariogram[41]	Mosaic Model[42]	Quadrature Mirror Filter (QMF)[43]	Discrete Cosine Transform (DCT)
Texture Spectrum/Histogram [44]	Fractal Model[18]		Local Walsh Transform & Local Fourier Transform[45]
Cross Diagonal Texture matrix[46]	Markov Model[18]	Optimization of FIR Filters[47]	Hadamard Transform[48]
	Gibbs Model[49]		Loop Filter & Wedge Filter[47]

2.2.1 Histogram-based Features

In digital image processing, the whole image is a combination of pixels, each of which has a value function $f(x,y)$, where x usually stands for row and y stands for column. For general images, we use M for the total number of rows and N for the total number of columns. If the image is a gray-scale image, the function $f(x,y)$ takes integer value from 0 to 255 (8 bits). Therefore, we have 256 intensity-levels. Then, the intensity-level histogram of the whole image is simply seen as the probability distribution of different gray levels.

The shape of the histogram provides a lot of information that can be used for image processing. For example, a low-contrast image has a narrow-distributed histogram.

We applied in Figure 2.1 the 256 level histograms of two typical images.

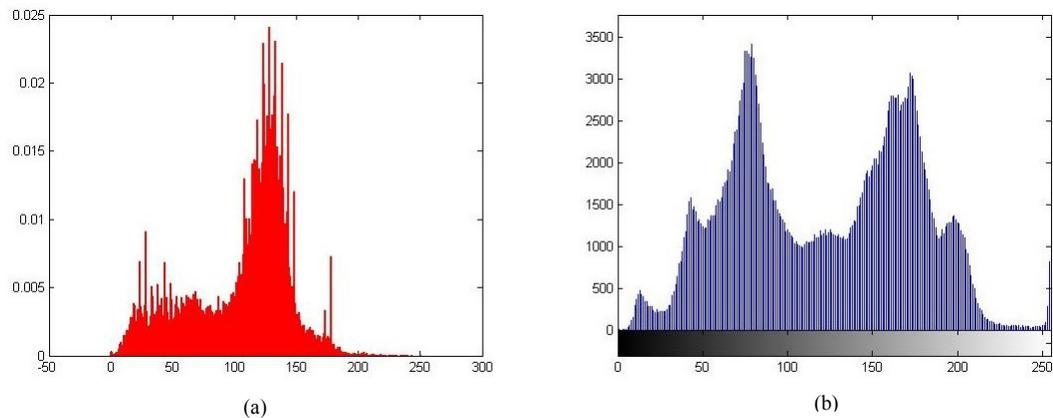


Figure 2.1: The 256 level histograms

(a) shows a low-contrast image histogram (y axis is the probability)

(b) shows a high-contrast image histogram (y axis scaled to the number of pixels)

From pixel gray value histograms one may derive simple measures such as the average gray value, the variance of gray values, percentiles and the texture contrast [50].

2.2.2 Co-occurrence Matrix

The co-occurrence matrix is defined as the joint probability of two different pixels at a given offset. It provides information on the luminance distribution and the distribution of the pixels with the same brightness. It is a second-order statistical measurement. Also, the second-order histogram is defined as the Co-occurrence Matrix.

For a gray-scale image, the co-occurrence matrix provides information on the direction, adjacent interval and amplitude changes. It is a basic measure used for image pattern and pixel arrangements in a certain region.

Given a gray-scale image $f(x,y)$ of M rows and N columns, the co-occurrence matrix is given as $L \times L$ matrix $T[t_{ij}]_{L \times L}$

Where, $L = 256$ in the gray-scale image, $i = 1, 2, \dots, L$ and $j = 1, 2, \dots, L$

$$t_{ij} = \sum_{p=1}^M \sum_{q=1}^N \begin{cases} 1, & \text{if } I(p, q) = i \text{ and } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

Where Δx is the horizontal offset and Δy is the vertical offset.

Usually we do not analyze all the elements since a gray scale image co-occurrence matrix contains 256^2 elements. However, we are able to use the mean of the rows μ_x , the mean of the columns μ_y , the variance of the rows σ_x^2 and the variance of the columns σ_y^2 . The correlation function is:

$$\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{i \cdot j \cdot p(i, j) - \mu_x \cdot \mu_y}{\sigma_x \sigma_y} \quad (2.2)$$

$$P(i, j) = \frac{t_{ij}}{\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} t_{ij}} \quad (2.3)$$

An example of a co-occurrence matrix calculation[51] is illustrated in Figure 2.2,

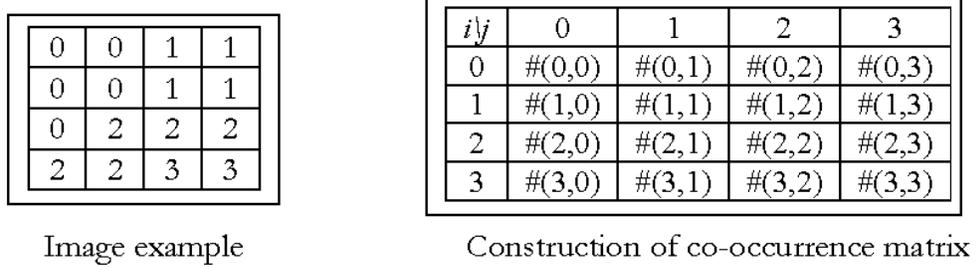


Figure 2.2: The spatial co-occurrence calculations

for $d = 1$ (d is the distance). The classification of fine textures requires small values of d , whereas coarse textures require large values of d . Reduction of the number of intensity levels (by quantizing the image to fewer levels of intensity) helps increase the speed of computation, with some loss of textural information.

2.2.3 Coarseness

Coarseness is a measure of scale in micro texture within the image. A simple procedure was presented that detects the largest size at which repetitive patterns are present[18].

$$C_{cos} = \left(\varepsilon + \sum_{i=0}^{G-1} p_i S(i) \right)^{-1} \quad (2.4)$$

Where G is the element number of A Gray-Tone Difference Matrix [52], ϵ is a small number to prevent the coarseness coefficient becoming infinite and p_i is the estimated probability of the occurrence of the intensity level i , $S(i)$ is the i -th entry of the Gray-Tone Difference Matrix.

2.2.4 Contrast Measure

Contrast is the difference in color that makes the image distinguishable. It describes the luminance of the image while it could be influenced by black and white color or the shading from different gray levels.

Mathematically, contrast can be present as:

$$Contrast = \sum_{\delta} \delta (i, j)^2 P(i, j)_{\delta} \quad (2.5)$$

Where $\delta(i, j)$ is the absolute value between the adjacent pixels; $P(i, j)_{\delta}$ is the pixels with gray level differences equal to δ between the adjacent pixels.

2.2.5 Directionality

Directionality is the gray level direction of the image. To get the directionality there are 4 steps.[53]

- Calculate the gradient of every pixel.

The horizontal and vertical gradients are obtained using:

$$g = \begin{pmatrix} \Delta_h \\ \Delta_v \end{pmatrix}$$

$$\Delta_h = \sum_{k \in \{-1, 0, 1\}} p(i+1, j+k) - p(i-1, j+k) \quad (2.6)$$

$$\Delta_v = \sum_{k \in \{-1, 0, 1\}} p(i+k, j+1) - p(i+k, j-1)$$

Where Δh is horizontal gradient and Δv is the vertical gradient.

- Calculate the polar coordinates of the graduation vector, ϕ is the phase.

$$(|g|, \phi) = \left(\frac{|\Delta_h| + |\Delta_v|}{2}, \tan^{-1} \left(\frac{\Delta_v}{\Delta_h} \right) + \frac{\pi}{2} \right) \quad (2.7)$$

- Calculate the histogram of the inclined vector angle.

Each bar of the histogram represents the ratio between total numbers of the pixels which satisfy

$$\frac{2k-1}{2n} < \frac{\phi}{\pi} < \frac{2k+1}{2n} \pmod{1} \quad (2.8)$$

and the number of pixels with condition (t is the threshold):

$$|g| > t \quad (2.9)$$

- Calculate the sum of the difference around the peak.

2.2.6 Autocorrelation Features

The autocorrelation function provides information on repetitive nature of the placement in an image. Mathematically, the autocorrelation function of a certain image $I(x, y)$ is defined as follows:

$$\rho(x, y) = \frac{\sum_{u=0}^N \sum_{v=0}^N I(u, v) I(u+x, v+y)}{\sum_{u=0}^N \sum_{v=0}^N I^2(u, v)} \quad (2.10)$$

If the relatively large texture primitives exist, the autocorrelation function value decreases slowly while the distance increasing. Inversely, it decreases rapidly when there are only small texture primitives. If the image is consist of periodically texture primitives, the autocorrelation changes periodically with the distance. The directional nature of this texture is reflected in the directional distribution of energy in the power spectrum.

2.2.7 Law's Texture Energy Measures

The Law's texture energy measures use local masks to estimate the textures. There are five masks representing Average Gray Level, Edge, Spots, Ripples and Waves. These masks are generated from the vectors as following:

Level L5 = [1, 4, 6, 4, 1]

Edge E5 = [-1,-2, 0, 2, 1]

Spots S5 = [-1, 0, 2, 0,-1]

Ripples R5 = [1, -4, 6,-4, 1]

Waves W5 = [-1, 2, 0,-2,-1]

By convoluting the 5×5 Law's Mask with the texture image, a feature vector can be generated to describe the texture. For example, by multiplying L5 and E5, we can get a 2D 5×5 mask below:

$$[1, 4, 6, 4, 1]' \times [-1, -2, 0, 2, 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Then using this filter to process the image, we are able to get the feature of each pixel.

2.2.8 Gabor Filter

In order to optimize the partial features, a filter with variable measurement should be used instead of a fixed one. Therefore, Gabor function can be used to extract texture features. A 2-D Gabor function can be present as following [54]:

$$g(x, y) = \left[\frac{1}{2\pi\sigma_x\sigma_y} \right] \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi jWx \right] \quad (2.11)$$

Where W is the frequency modulated by a Gaussian envelope. Therefore, the Fourier transform of $g(x, y)$ can be present as following:

$$G(u, v) = \exp \left\{ -\frac{1}{2} \left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right] \right\} \quad (2.12)$$

$$\sigma_u = 1 / 2\pi\sigma_x \quad (2.13)$$

$$\sigma_v = 1 / 2\pi\sigma_y \quad (2.14)$$

The values σ_x and σ_y are the sizes of the Gaussian envelope in the x and y directions. For a Gabor filter with a particular orientation, the Gabor wavelet can be written as following:

$$g_{mn}(x, y) = a^{-m} g(x', y'); \quad a > 1 \quad (2.15)$$

$$x' = a^{-m}(x \cos \theta + y \sin \theta), \quad y' = a^{-m}(-x \sin \theta + y \cos \theta) \quad (2.16)$$

Based on the linearity of the Fourier transform,

$$u' = a^{-m}(u \cos \theta + v \sin \theta), \quad v' = a^{-m}(-u \sin \theta + v \cos \theta) \quad (2.17)$$

By changing the integer value of m and n , a set of filters with different measurement and orientation can be generated.

2.2.9 Hu's Moments

2-D $(p+q)^{\text{th}}$ order moment is defined as follows:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (2.18)$$

The moments of all orders exist only when the image function $f(x, y)$ is a piecewise continuous bounded function. There is a one-one mapping between each element of the moment sequence $\{m_{pq}\}$ and $f(x, y)$ and when $\{m_{pq}\}$ is known, correspondingly, $f(x, y)$ is uniquely. When $f(x, y)$ changes by translating, rotating or scaling, the moments might be variant. The invariant features can be defined using central moments:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y); \quad p, q = 0, 1, 2, \dots; \quad \bar{x} = \frac{m_{10}}{m_{00}}; \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (2.19)$$

The pixel point is the image centroid. After normalization, scale invariance can be obtained as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}; \gamma = (p+q+2)/2, p+q = 2, 3, \dots \quad (2.20)$$

Based on central moment normalization, Hu introduced seven moment invariants[55]
[56]:

$$\phi_1 = \eta_{20} + \eta_{02} \quad (2.21)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (2.22)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \mu_{03})^2 \quad (2.23)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \mu_{03})^2 \quad (2.24)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} - \eta_{12})[(\eta_{30} + 3\eta_{12})^2 - 3(\eta_{21} + \eta_{03})] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.25)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (2.26)$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{21})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (2.27)$$

The seven moment invariants were proved to be rotational invariance, scaling invariance and translation invariance by Hu in 1962. Usually Hu's moment is used to do the recognition of large size object in the image. Hu's moment can response in a very short time while doing the recognition. However, since Hu's moments are low-order moments based (up to third moment), it is weak to describe the image in details.

2.2.10 Haralick Texture Features

The Haralick texture features can be used for image classification. Those features capture information about the texture patterns. After constructing a co-occurrence matrix, the features are calculated and totally there are 13 features' calculations. Among the 13

features, the contrast, correlation, the entropy measures and the angular second moment are included.

The 13 texture features are listed in as following[57]:

- Angular Second Moment:

$$f_1 = \sum_{i=1}^N \sum_{j=1}^N p(i, j)^2 \quad (2.28)$$

where N is the number of gray levels. When the gray level of an image distribute uniformly, the ASM of the image is a relatively large value.

- Contrast:

$$f = \sum_{n=0}^{N-1} n^2 \left\{ \sum_{i=1}^N \sum_{\substack{j=1 \\ |i-j|=n}}^N P(i, j) \right\} \quad (2.29)$$

Contrast shows a total gray level change. The larger the contrast is, the more distinct for visual perception.

- Correlation

$$f_3 = \frac{\sum_i \sum_j (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.30)$$

where μ_x is the mean of the each row, μ_y is the mean of the each column, σ_x is the standard deviation of the each row, σ_y is the standard deviation of the each column.

- Sum of squares: variance

$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j) \quad (2.31)$$

where μ is the mean value of the whole image.

- Inverse Different Moment

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j) \quad (2.32)$$

This value shows the homogeneity of gray scale in local pixel space.

- Sum Average

$$f_6 = \sum_{i=2}^{2N} iP_{x+y}(i) \quad (2.33)$$

where

$$P_{x+y}(k) = \sum_{\substack{i=1 \\ i+j=k}}^N \sum_{j=1}^N p(i, j) \quad k = 2, 3, \dots, 2N \quad (2.34)$$

- Sum Variance

$$f_7 = \sum_{i=2}^{2N} (i - f_8)^2 P_{x+y}(i) \quad (2.35)$$

- Sum Entropy

$$f_8 = - \sum_{i=2}^{2N} P_{x+y}(i) \log \{P_{x+y}(i)\} \quad (2.36)$$

- Entropy

$$f_9 = -\sum_i \sum_j P(i, j) \log\{P(i, j)\} \quad (2.37)$$

- Difference Variance

$$f_{10} = \text{variance of } p_{x-y}$$

$$P_{x-y} = p_{x-y}(k) = \sum_{\substack{i=1 \\ |i-j|=k}}^N \sum_{j=1}^N p(i, j) \quad k=0,1,\dots,N-1 \quad (2.38)$$

- Difference Entropy

$$f_{11} = -\sum_{i=0}^{N-1} P_{x-y}(i) \log\{P_{x-y}(i)\} \quad (2.39)$$

- Information Measure 1 of Correlation:

$$f_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}} \quad (2.40)$$

where HX is the entropy of p_x and HY is the entropy of p_y , and

$$HXY = -\sum_i \sum_j P(i, j) \log(p(i, j)) \quad (2.41)$$

$$HXY1 = -\sum_i \sum_j P(i, j) \log(p_x(i)p_y(j)) \quad (2.42)$$

- Information Measure 2 of Correlation:

$$f_{13} = \sqrt{1 - e^{-2.0(HXY2 - HXY)}} \quad (2.43)$$

where

$$HXY2 = -\sum_i \sum_j p_x(i) p_y(j) \log(p_x(i) p_y(j)) \quad (2.44)$$

To apply the Haralick texture features, Eizan Miyamoto et al. developed a fast calculation using standard practices of optimizing code such as scalar replacement and unrolling. [58]

2.3 Texture Segmentation

Texture segmentation is complex and difficult because without any image processing experiment, usually it is not easy for the computer or even the engineer to tell how many types of textures are there in an certain image, what kind of textures exist in that image, what is the total number of the textures and which region contain the texture that is easy to be analyzed. Actually, in order to do texture segmentation, there is no need to know exactly which specific textures are there in the image. The easiest way to do the segmentation is using proper methods which make one texture different from another in the same image. To do the texture segmentation, two general approaches, which are popular now, are used: region-based approaches and boundary-based approaches. [54]

- Region-based Texture segmentation

To do the texture segmentation by a region-based approach, we try to identify regions of the image which have a uniform texture. Pixels or small local regions are merged based on the similarity of some texture property. The regions having different textures are then considered to be segmented regions. This method has the advantage that the boundaries of regions are always closed and therefore, the regions with different textures are always well separated. It has the disadvantage, however, that in many region-based segmentation

methods, one has to specify the number of distinct textures present in the image in advance. In addition, thresholds on similarity values are needed.

For example, region growing is a simple region-based method. Firstly, we should choose a set of seed points of the image and we have to determine the range of threshold. Then the regions are grown from these seed points to adjacent points depending on a region membership criterion. We keep examining the adjacent pixels of the seed point. If they belong to the criterion we used, they are classified into seed points. Until all the adjacent points are not belonging to our criterion, the growing stopped.

- Boundary-based Texture segmentation

Detect the differences in texture in adjacent regions. Thus boundaries are detected where there are differences in texture. In this method, one does not need to know the number of textured regions in the image in advance. However, the boundaries may have gaps and two regions with different textures that are not identified as separate closed regions. Strictly speaking, the boundary based methods result in segmentation only if all the boundaries detected form closed curves.

2.3.1 Summary

Based on all the techniques and methods discussed above, in the vacant parking space detection project, since the number of texture regions of the input image is unknown, the Boundary-based texture segmentation approach is used. Among all the texture extraction techniques, histogram and contrast method is already used in our detection (the foreground/background detection). Therefore, the future work is using other texture features to improve the detection result of our vacant parking detection system.

CHAPTER 3

THE FEATURES USED IN OUR SYSTEM

Before deploying the proposed system, an initial installation & calibration stage is needed. For a given image covering a certain parking area (5 to 7 spaces), the camera position is fixed. Thus, the parking space boundaries are also fixed. We can define the region of each parking space, using 4 dots or just a parallelogram as a given parking spot. After that, each parking space is allocated a number, usually from one (representing the left-most space) to the maximum number (representing the right-most space) in a sequence. Once the installation and calibration stages are carried, the system is active and ready for usage. The block diagram of the whole system is shown in Figure 3.1.

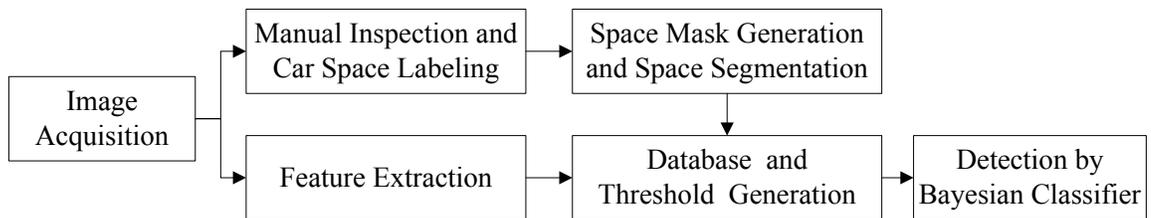


Figure 3.1: Block diagram of the proposed system

We propose three different methods for vacant parking detection. The first detection method is a simple edge detection technique [59] and an edge counting approach [16]. The second technique is based on object counting, while the third method is based on foreground & background detection. Each of these techniques provides one decision independently. Then, we propose to integrate the results from these three methods into a final decision. The flowchart of processing one input image is shown in Figure 3.2.

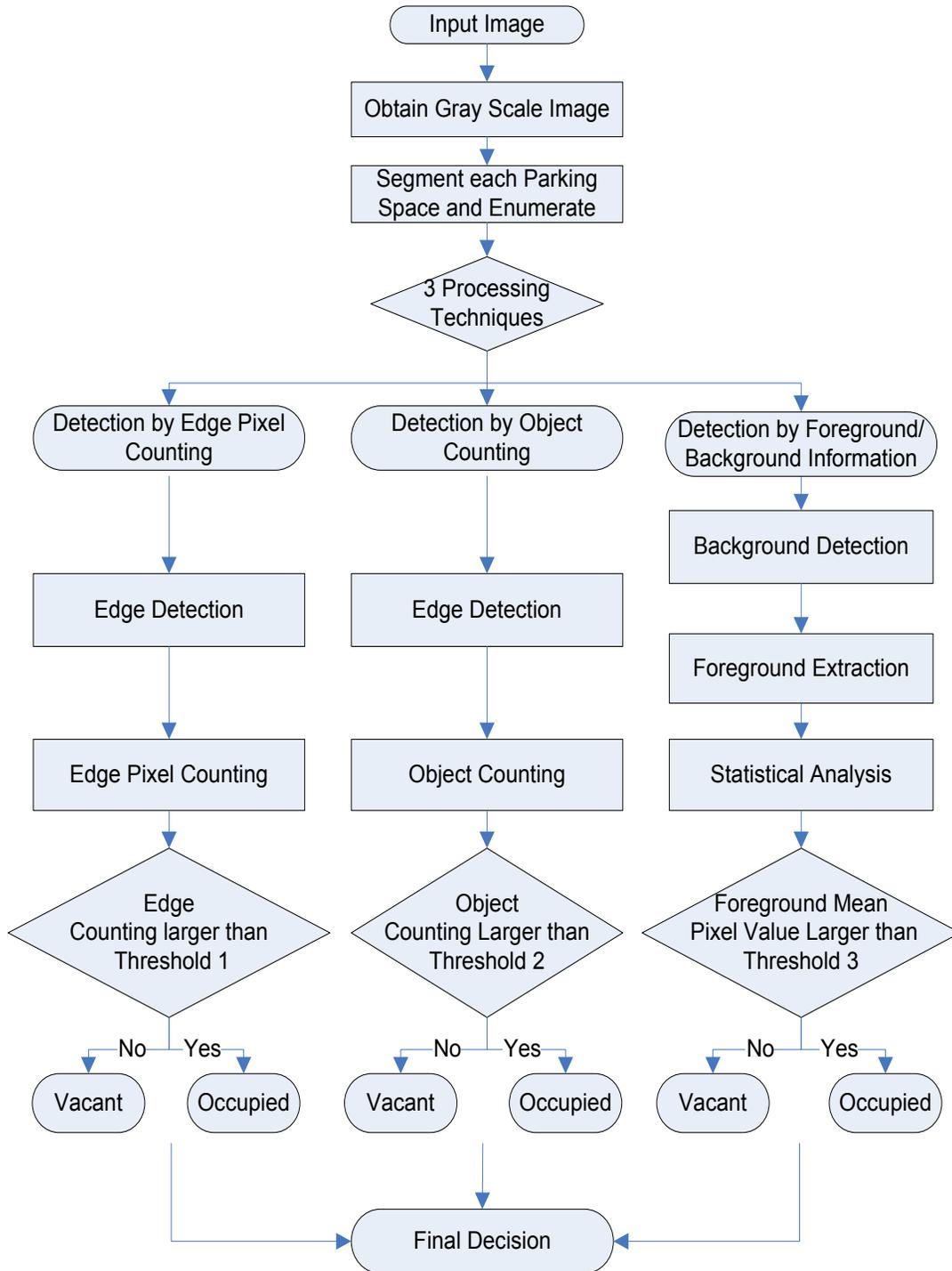


Figure 3.2: Flow chat of processing one input image

3.1 Edge Information

We start with a simple edge detection algorithm using the Prewitt edge operator [59], shown in Table 3.1:

Table 3.1: Prewitt operator

Z1	Z2	Z3	-1	-1	-1	-1	0	1
Z4	Z5	Z6	0	0	0	-1	0	1
Z7	Z8	Z9	1	1	1	-1	0	1

Image Neighborhood Mask for G_x Mask for G_y

The Prewitt operator provides a simple but efficient approximation of the first derivative. For a certain pixel with a gray level $G(i,j)$, after applying domain convolution with the image and masks, the overall potential edge-value is obtained as the max of G_x and G_y . If this value is above a certain threshold it is declared as an edge pixel.

3.1.1 Edge Counting Method

After applying the Prewitt edge detector, we use the parking space mask to extract the edge information for each parking space and count the number of edges. The space filter only covers the edge pixels within the predefined car spots obtained from the training stage. If the total number of edge pixels is less than a certain value, the parking space is declared, as vacant.

An example of an image with no vacant parking space is shown in Figure 3.3.



Figure 3.3: Input image example 1

We display in Figure 3.4 the cropped image showing a given parking space. Such area is used in the calibration and installation stages.



Figure 3.4: Segmentation of space 2

Recall that during training and calibration, the different parking spaces are segmented and numbered. After the edge detection stage, we get the image displayed in Figure 3.5.

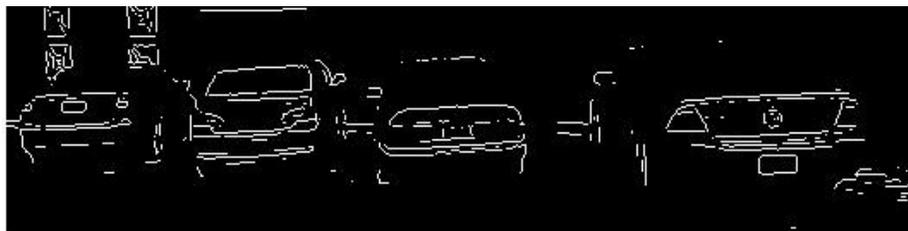


Figure 3.5: Edge detection of Figure 3.3

After multiply the edge image and count the edge pixel number, we get Figure 3.6

space 1 total pixel : 379 ,occupied



Figure 3.6: Edge pixel counting for all the four parking spaces of Figure 3.3

3.1.2 Boundary Counting Method

With the edge information obtained from the Prewitt operator, we determine all the closed contours or objects and count the total number of objects in each parking spot. More specifically, an occupied parking space consists of more closed objects than a vacant parking space. Therefore, when the number of contours is less than a certain threshold, the parking spot is declared as vacant. Figure 3.7 shows the region counting of space 1.

space 1 total region : 49 ,occupied



Figure 3.7: Object counting result of space 1 in Figure 3.3

Using the parking space filter from the training stage, we extract the edge information for each parking space. For the object counting method, we use the same edge information but we count the number of the contours. The small numbers are the marks of each object

(or close contours). Each object is given a label. The results of the edge counting technique and the object counting technique for image 1 are displayed in Table 3.2.

Table 3.2: Edge counting and object counting techniques for Figure 3.3

	space 1	space 2	space 3	space 4
# edge pixels	379	551	480	803
Decision 1	occupied	occupied	occupied	occupied
# objects	49	32	44	61
Decision 2	occupied	occupied	occupied	occupied

The threshold (generated from Otsu Method described in section 4.1 and 4.2) is listed below:

The number of edge pixels= 360;

Region Number=30;

3.2 Background Information and Foreground Acquisition

In this method, we propose to process the image as follows:

- 1) Transform the input color image into a gray scale image.
- 2) Obtain the histogram of the gray scale image.
- 3) Get the histograms using 25 bins with each of these bins covering around 10 pixel values. (We tested using different bins numbers and found out that even reduced the number of bins from 256 to 25, the final result do not change.) The bin with the highest probability is taken as the background bin. All pixel values within the bin, found above,

are declared as background pixels, we use the mean of all these pixels as the relative background values when extracting the foreground information.

4) The difference between the original grey level image and the background image (from step 3) is obtained. Non-zero pixel values provide information on whether a certain spot area contains a car or just consists of a background.

5) The number of non-zero pixels from step 4 for each parking spot is counted and compared to a threshold, if the number is above a certain threshold, the region is declared as occupied, otherwise it is declared as a vacant space.

3.2.1 Background Generating

For the foreground/background segmentation technique, we used the histogram as shown in Figure 3.8 to identify the background pixels.

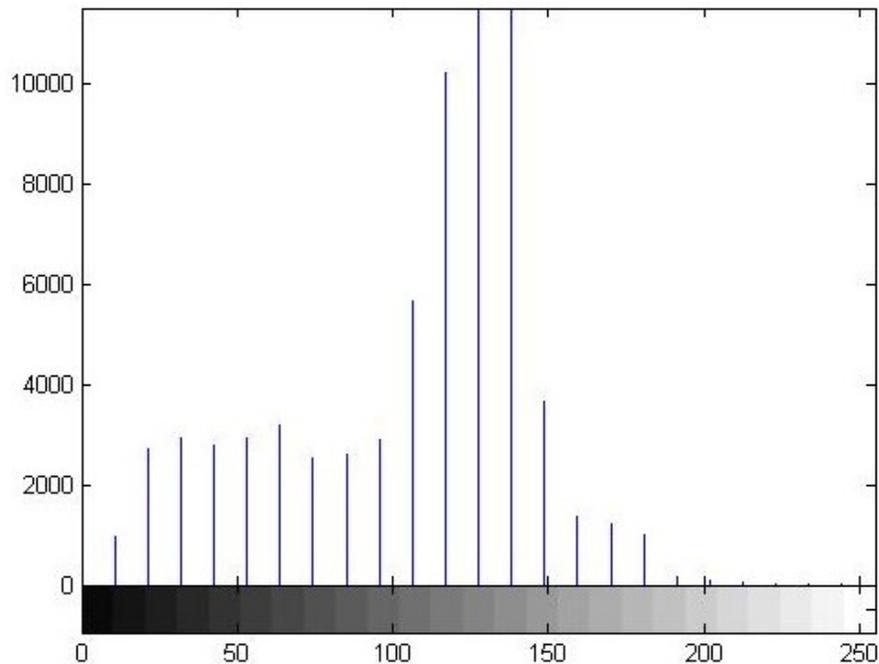


Figure 3.8: Histogram for Figure 3.3 using 25 bins

The maximum of the histogram is obtained at pixel values 122-132. All pixel values within that interval are declared as background pixels. We use the mean of all these pixels as the relative background value.

3.2.2 Foreground Acquirement

Then we use the background information to extract the foreground pixels. The resulting foreground Mean Pixel Value (FMPV) image is shown in Figure 3.9. The results are displayed in Table 3.3.



Figure 3.9: Foreground information for Figure 3.3

The threshold (generated from Otsu Method described in section 4.1 and 4.2) is FMPV=45;

Table 3.3: Detection results using the foreground/background method for Figure 3.3

	Space 1	Space 2	Space 3	Space 4
FMPV	131	101	115	104
Decision 3	occupied	occupied	occupied	occupied

When considering all the images with different numbers of vacant parking spaces in the database, we found that a threshold of 45 is suitable to differentiate between vacant and occupied spaces (again using Otsu's technique).

3.3 Summary

This chapter describes a system using our image-based vacant parking detection method using hybrid features. The system flowchart and one example of each feature are presented. In chapter 5, the experimental results and performance evaluation are based on the system described in this chapter.

CHAPTER 4

OTSU METHOD AND THE BAYESIAN CLASSIFIER

4.1 Otsu Method and Threshold Acquisition

Otsu's method is used for performing clustering-based thresholding automatically. It is applied when the image contains two classes of pixels or bi-model histogram. By minimizing the intra-class variance; Otsu Method calculates the optimum threshold[62]. This method is first introduced by Mr. Otsu in 1979 to solve threshold selection problem from gray level histograms[63].

In Otsu's Method, all the calculation is used to search the threshold that minimizes the variance within the class. The variance within the classes is defined as the intra-class variance:

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t) \quad (4.1)$$

Where t is a selected threshold, $\sigma_1^2(t)$ is the variance of class 1 and $\sigma_2^2(t)$ is the variance of class 2, $w_1(t)$ is the probability of class 1 and $w_2(t)$ is the probability of class 2.

The class probabilities are given by

$$w_1(t) = \sum_{i=1}^t p_i \quad (4.2)$$

$$w_2(t) = \sum_{i=t+1}^L p_i \quad (4.3)$$

Where p_i is the probability of gray level i and L is the total number of gray levels.

Otsu proved that the minimum of the intra-class variance $\sigma_w^2(t)$ is the maximum of the inter-class variance $\sigma_b^2(t)$:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = w_1(t)w_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (4.4)$$

Then the class-mean levels are given by

$$\mu_1(t) = \sum_{i=1}^t \frac{i \cdot p_i}{w_1(t)} \quad (4.5)$$

$$\mu_2(t) = \sum_{i=t}^L \frac{i \cdot p_i}{w_2(t)} \quad (4.6)$$

Thus, in our project, for example, when using the feature FMPV which is related to the gray-scale image, $L = 256$. Then we set up $\mu_1(0)$, $\mu_2(0)$, $w_1(0)$ and $w_2(0)$ to get the value of $\sigma_b^2(0)$. By updating $t = 1, 2, \dots, L$, we can get a group of $\sigma_b^2(t)$. The value t that maximize $\sigma_b^2(t)$ is the threshold we want.

4.2 The Bayesian Classifier

Bayes classifier is widely used in statistical classification. Such classifier minimizes the probability of misclassification.[64] A naïve Bayes classifier, which is based on Bayes'

theorem, is a simple probabilistic model. To apply a naïve Bayes classifier, there are some conditions for the features. Presence and absence of a particular feature should be unrelated to the presence or absence of any other features. That means those features should be independent from each other and contribute independently to the probability. The method of maximum likelihood is used to do the parameter estimation for naïve Bayes classifiers in many cases. It can have a training database in the pre-processing stage or deal with the database directly. Since each feature is independent from others, only the variances for each class are needed to take into consideration instead of the entire covariance matrix.

Given n independent feature variables vectors F_1 to F_n . C is defined as a dependent class variable which can be decided by F_1 to F_n :

$$p(C|F_1, \dots, F_n) = \frac{p(C)p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)} \quad (4.7)$$

or in general:

$$posterior = \frac{prior * likelihood}{evidence} \quad (4.8)$$

In practice, we care about the numerator part of the equation. The values of all the features, F_i , are given so that it could be consider as constant. Using the joint probability model, the numerator can be represent as

$$\begin{aligned} p(C, F_1, \dots, F_n) \\ = p(C)p(F_1, \dots, F_n|C) \end{aligned} \quad (4.9)$$

$$\begin{aligned}
&= p(C)p(F_1|C)p(F_2, \dots, F_n|C) \\
&= p(C)p(F_1|C)p(F_2|C, F_1) \dots p(F_n|C, F_1, \dots, F_{n-1})
\end{aligned}$$

The naïve Bayes classifier has an assumption of all the features are independent which means given the category C , when $i \neq j$, F_i is conditionally independent of F_j . Therefore,

$$p(F_i|C, F_j) = p(F_i|C) \quad (4.10)$$

and

$$p(C|F_1, \dots, F_n) \propto p(C) \prod_{i=1}^n p(F_i|C) \quad (4.11)$$

Under the independence assumption,

$$p(C|F_1, \dots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^n p(F_i|C) \quad (4.12)$$

Where Z is a scaling factor dependent on the features F_1, \dots, F_n . If all the features variables are known, Z is a constant.

In [43], Domingos et al. verified that the Bayesian classifier is able to achieve a quite well performance under different conditions. Empirical evidence is given by dealing with numeric attributes, zero counts and missing values.

4.3 The Application of Naïve Bayesian Classifier using Weka

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter

estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

In our experiment, the given variables are feature vectors generated from different images, containing the size of the image, the size of each parking space, the number of edge pixels, the edge pixel density of the whole image, the edge pixel density of single parking space, region numbers, static/dynamic background pixel value and FMPV.

In section 5.2, we are going to use Weka 3-6-10 [66] software to show the experimental result using the naïve Bayes Classifier, applying three main features which are the number of edge pixels, region numbers and FMPV, plus a minor feature, dynamic background pixel value.

In the Weka explorer, *weka.classifiers.bayes package* is offered. The database should be set up in advance. Edit in the preprocess stage as shown in Figure 4.1

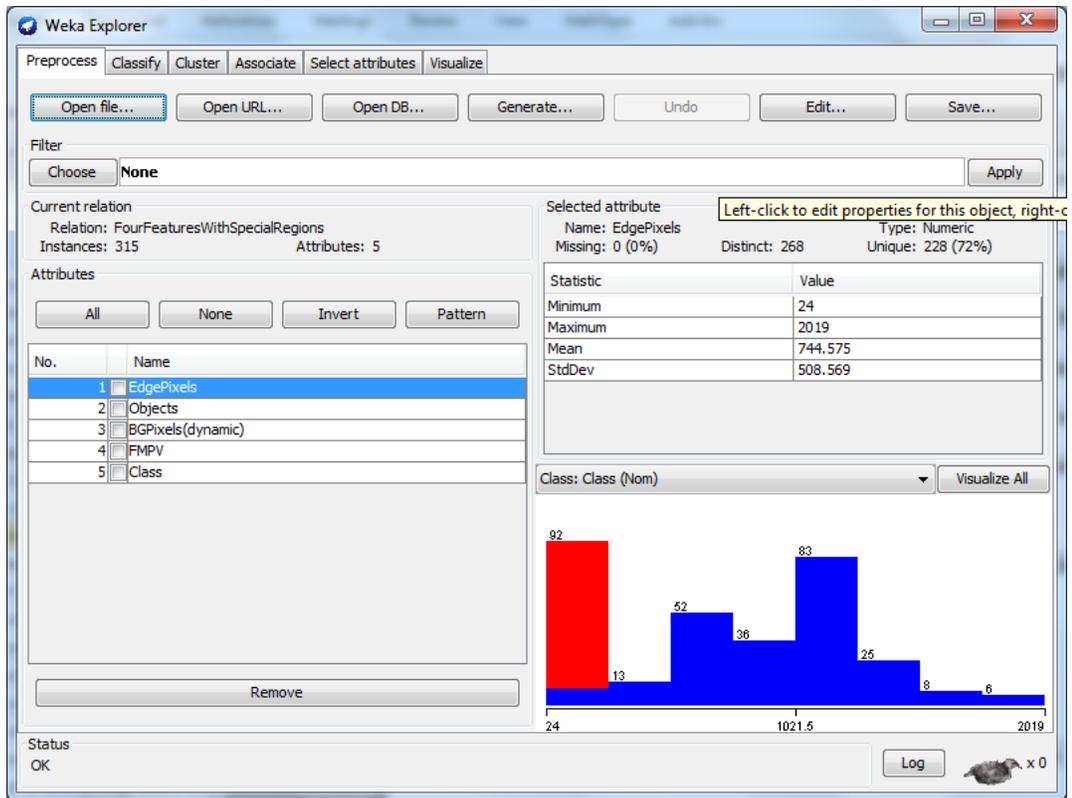


Figure 4.1: Data Preprocess using Weka

After selecting the features, we choose the Naïve Bayes tool as shown in Figure 4.2

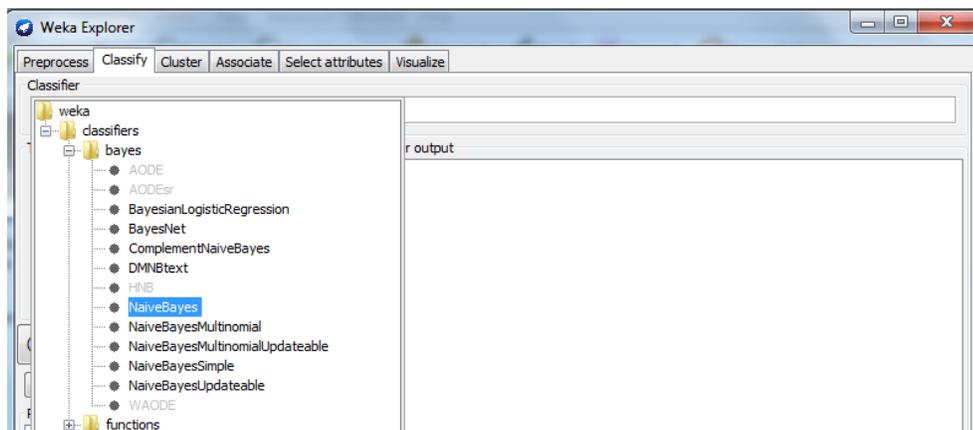


Figure 4.2: Naive Bayes Classifier

4.4 Summary

In this section, a brief introduction of related method used in our system is given. Otsu method is used for threshold acquirement. It will be applied to all the features in the preparation stage. The Bayesian classifier is applied to the whole database and the experimental results will be shown in chapter 5.

CHAPTER 5

EXPERIMENTAL RESULTS AND SYSTEM

PERFORMANCE

This chapter includes the experimental results and the system performance. The main objective is to show that the image-based vacant parking detection system using hybrid features can achieve high recognition accuracy and can be applied under different conditions. In section 5.1 and 5.2, the database used in this project will be described and the overall performance will be presented. It consists of more than 100 different images taken from the same parking lot and shows the detection performance and data analysis using Weka (introduced in section 4.3 “The Application of Naïve Bayesian Classifier using Weka”). The image-based vacant parking detection system using hybrid features and Bayesian classifier will be presented by the outcomes of the image processing and statistical analysis. The hybrid features used in this project are listed below:

- 1) The number of edge pixels (contour feature of the shape feature, acting as a main feature in our system, described in section 3.1.1 “Edge Counting Method”)
- 2) Object Number (Boundary-based texture segmentation and region feature, acting as a main feature in our system, described in section 3.1.2 “Boundary Counting Method”)

- 3) Foreground Mean Pixel Value (Statistical method of Texture Features, related to contrast and histogram, acting as a main feature in our system, described in section 3.2.2 “Foreground Acquirement”)
- 4) Dynamic Background Pixel Value (Statistical method of Texture Features, related to contrast and histogram, acting as a secondary feature in our system, described in section 3.2.1 “Background Generating”)
- 5) Edge Pixel Percentage of the Selected Parking Space (A combination of shape feature and texture feature, acting as a reference feature of the number of edge pixels, related to edge detection and coarseness.)

After applying the naïve Bayes classifier to all main features and secondary features, the number of edge pixels achieves highest recognition accuracy. However, one feature is not enough. Under the ideal conditions, the system is able to achieve 100% recognition accuracy using three features. Therefore, it is necessary to decide how to combine the hybrid features and how many features are enough to achieve an excellent recognition result.

To transform all the images to the feature database, each image is processed to the single image vacant parking system independently. For the parking images taken from a fixed position (located in Building 59, B2 floor, opposite to the elevator entrance, three complete parking spaces within the camera’s view), there are totally three types of parking: all-vacant parking, mixed-state parking, no-vacant parking.

5.1 Vacant Parking Detection for a Fixed Parking Region

In this section, the experimental results from a total of four parking images are presented. Each sub-section contains both the outcomes of image processing and detection summary of one of the six parking images. From section 5.1.1 to section 5.1.3, all four input images come from one fixed parking lot with three parking spot (located in Building 59, B2 floor, opposite to the elevator entrance). All images share the same size 640 by 480 (pixels), which is just the original image size taken by the front camera of iPhone 4. The camera covers three complete parking spaces so that there are 4 different combinations: parking region with three vacant spaces and no cars, parking region with two vacant spaces and one car, parking with one vacant space and two cars, and parking region with three cars. The processing time of one image is 0.49515 seconds.

The threshold (generated from Otsu Method described in section 4.1 and 4.2) for the images from section 5.1.1 to section 5.1.3 is listed below:

The number of edge pixels= 360;

Region Number=30;

FMPV=50;

Edge Pixel Percentage of Single Parking Region=1.6%.

5.1.1 Vacant Parking Detection for the Case of All-vacant Parking Region

In this section, we present the experimental result of a parking with three vacant parking spaces. Image name is given to show the date and time as well as being the index of the whole database. An input raw image with file name “2013-10-11 13.46.36” is shown in Figure 5.1:



Figure 5.1: The input image of a fixed parking region with three vacant spaces and zero occupied space

After transforming the input image to a gray-scale image, edge detection of the whole image is applied using the Prewitt operator as shown in Figure 5.2 :



Figure 5.2: Edge detection of a fixed parking region with three vacant spaces and zero occupied space

Then a fixed parking space filter from the training stage will be applied to extract the edge information for each parking space, using rectangle model extraction. Firstly, the corners of each rectangle region are identified. Secondly, using the defined corners, each parking space is replaced with a rectangle. Thirdly, we set the matrix inside the rectangle to logic 1 (all white) and the remaining to logic 0 (black). Thus, three templates for each of the parking spaces are prepared. Finally we multiply each template and the edge detection image one by one and the extraction of each parking space is generated easily. The additional calculation is counting the number of edge pixels and object number of each parking space. Figures 5.3 to 5.5 show the edge pixel counting results, while Figures 5.6 to 5.8 show the object counting results.

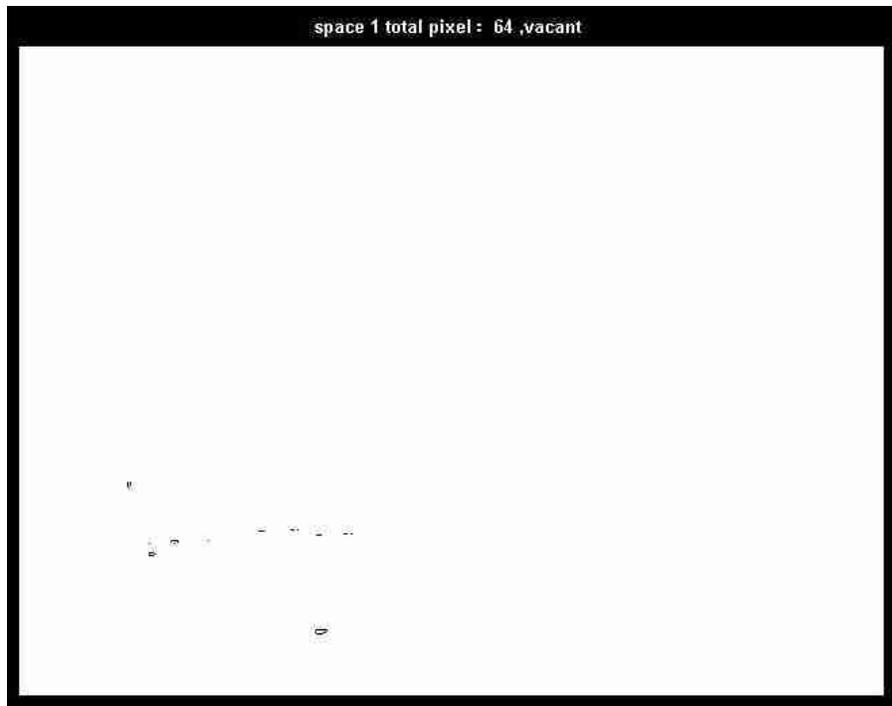


Figure 5.3: Edge pixel counting for space 1 of a fixed parking region with three vacant spaces and zero occupied space, space 1 containing 64 edge pixels, detected as a vacant space

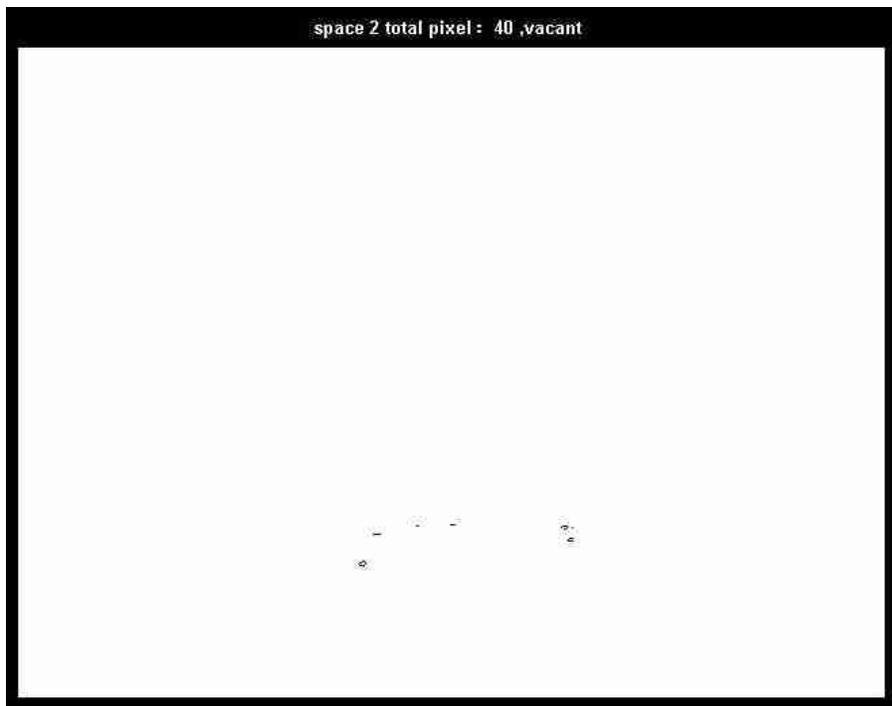


Figure 5.4: Edge pixel counting for space 2 of a fixed parking region with three vacant spaces and zero occupied space, space 2 containing 40 edge pixels, detected as a vacant space

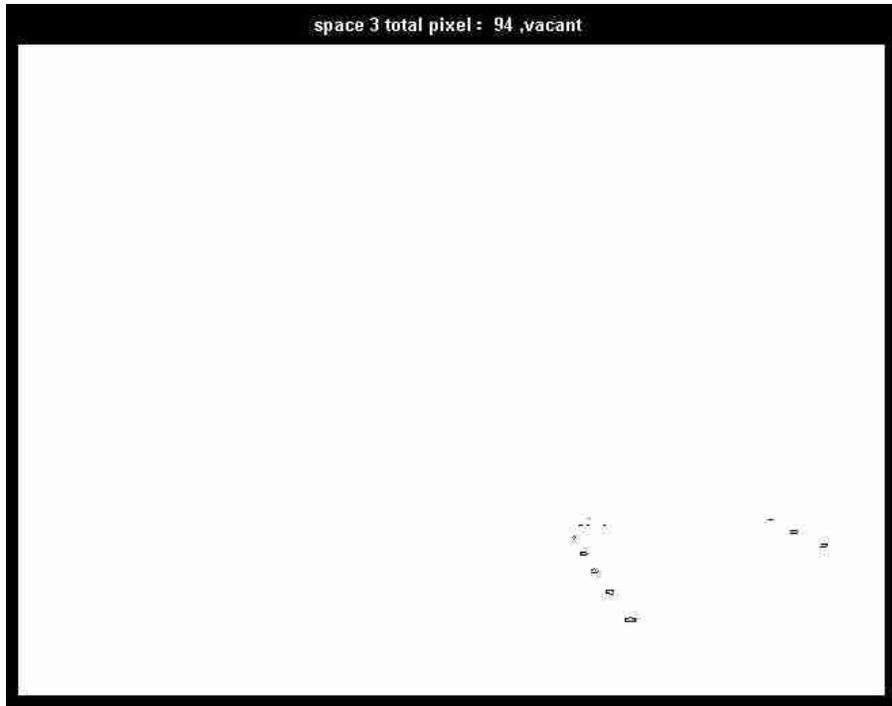


Figure 5.5: Edge pixel counting for space 3 of a fixed parking region with three vacant spaces and zero occupied space, space 3 containing 94 edge pixels, detected as a vacant space

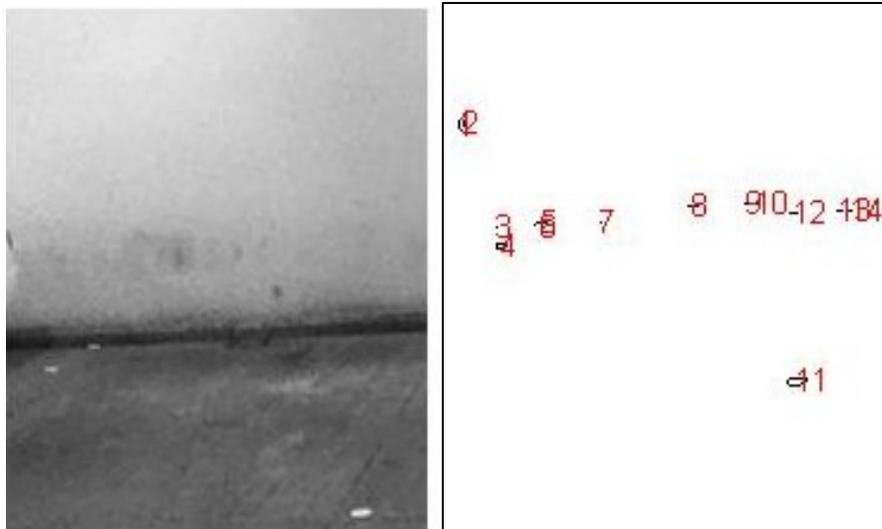


Figure 5.6: Object counting for space 1 of a fixed parking region with three vacant spaces and zero occupied space, space 1 containing 14 objects, detected as a vacant space

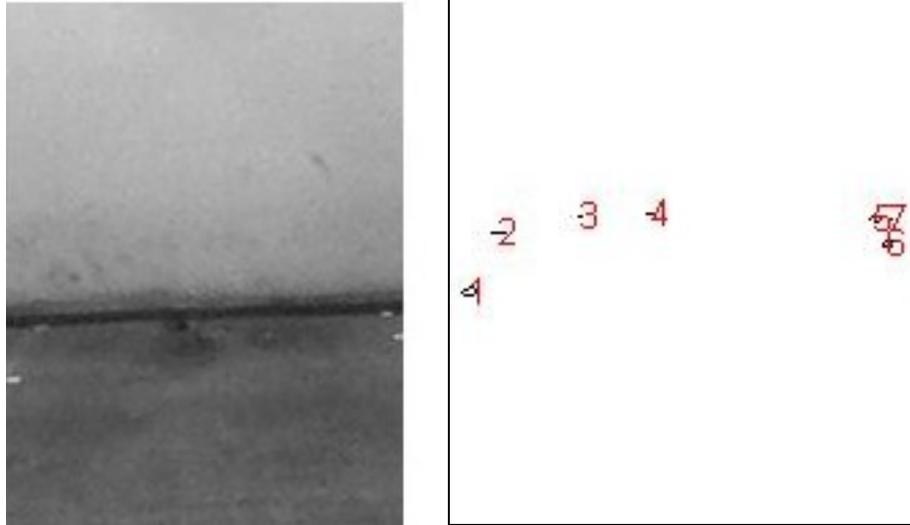


Figure 5.7: Object counting for space 2 of a fixed parking region with three vacant spaces and zero occupied space, space 2 containing 7 objects, detected as a vacant space

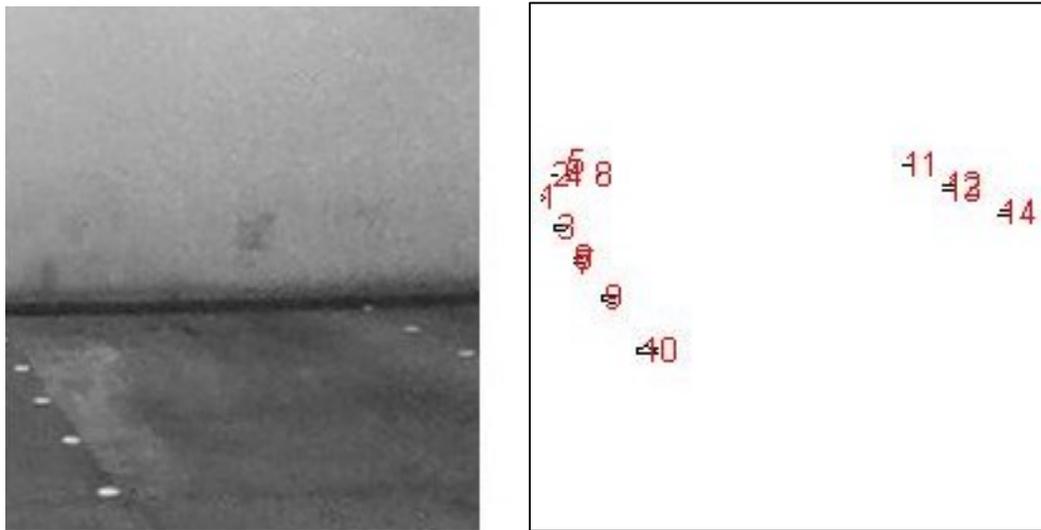


Figure 5.8: Object counting for space 3 of a fixed parking region with three vacant spaces and zero occupied space, space 3 containing 14 objects, detected as a vacant space

For the foreground/background segmentation technique, we used the histogram as shown in Figure 5.9 and Figure 5.10 to identify the background pixels.

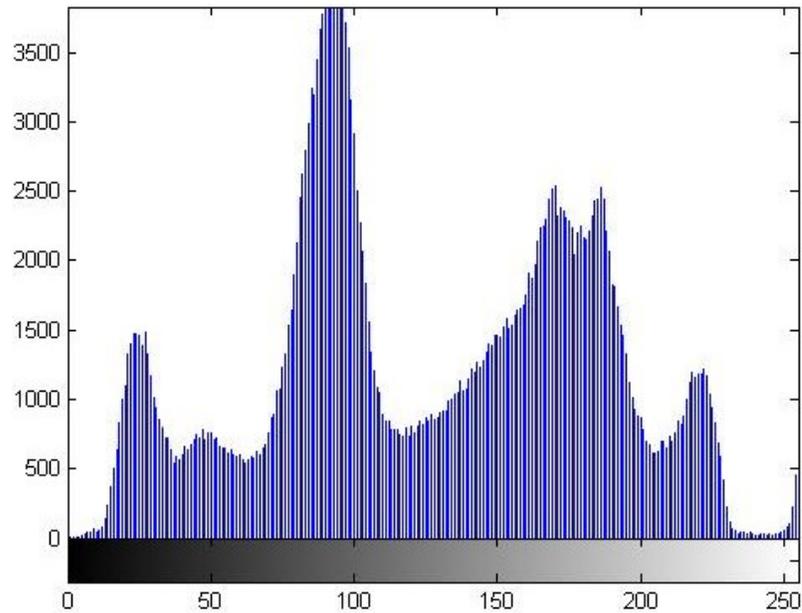


Figure 5.9: Histogram of a fixed parking region with three vacant spaces and zero occupied space

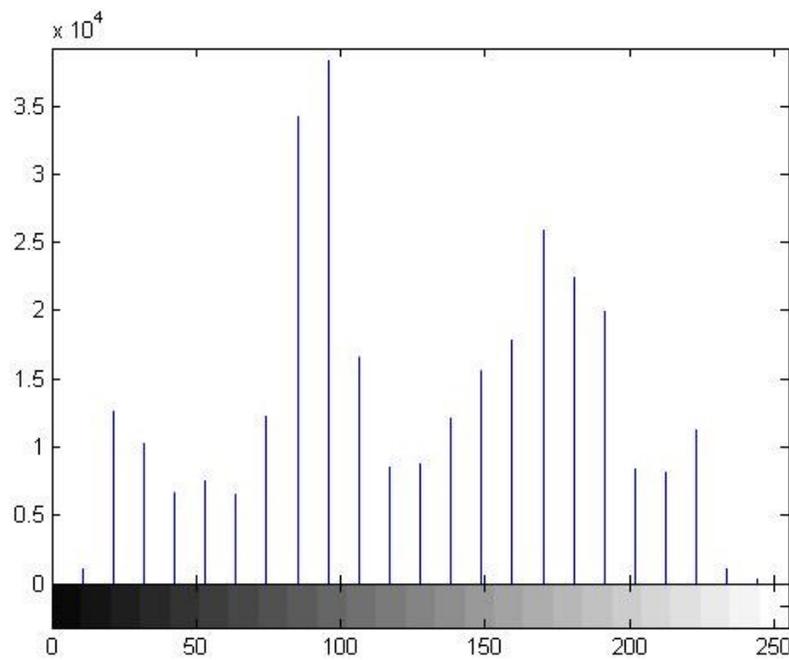


Figure 5.10: Histogram using 25 bins of a fixed parking region with three vacant spaces and zero occupied space

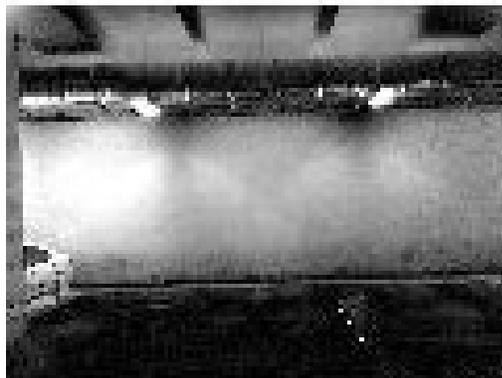
Figure 5.11 shows how the foreground information is generated



(a) The dynamic background pixel value subtracts the gray-scale image



(b) The gray-scale image subtracts the dynamic background pixel value



(c) Integrated image of (a) and (b)

Figure 5.11: The foreground image generation of a fixed parking region with three vacant spaces and zero occupied space

The final result summary of the input image is shown in Table 5.1:

Table 5.1: Summary of Figure 5.1

Image Information	Image Date	2013-10-11 13.46.36		
	Place	Indoor		
	Bayesian Decision	Vacant	Vacant	Vacant
	Spaces	v1	v2	v3
Original Image	x	640		
	y	480		
	Total Pixels	307200		
Edge Based	X Region	210	210	210
	Y Region	170	160	190
	Region Total	35700	33600	39900
	Edge Pixels	64(V)	40(V)	94(V)
	% of the whole image	0.0201	0.0130	0.0306
	% of Region	0.179(V)	0.119(V)	0.236(V)
	Objects	14(V)	7(V)	14(V)
Foreground/Background Method	BG value(dynamic)	95.625		
	FMPV	59(O)	47(V)	40(V)

The above table shows the detection result, where V stands for vacant, O stands for occupied. Since there are no occupied spaces, the all-vacant parking images could be used as a template when applying image-matching method described in other related works.

5.1.2 Vacant Parking Detection for Mixed-State Parking Region

1) Parking Region Containing Two Vacant Spaces and one Occupied Space

In this section, we present the experimental result of a parking with two vacant parking spaces and one occupied parking space. Image name is given to show the date and time as well as being the index of the whole database. An input raw image with file name: 2013-12-22 22.39.49 is shown in Figure 5.12:

Input Image, choose the parking region



Figure 5.12: The input image of a fixed parking region with two vacant spaces and one occupied space

We repeat the whole processing applied in the previous section and show the results in the same order. We get the edge and region information using edge detection as shown in Figure 5.13. After that, Figures 5.14 to 5.16 show the edge pixel counting method while Figures 5.17 to 5.19 show the object counting method.

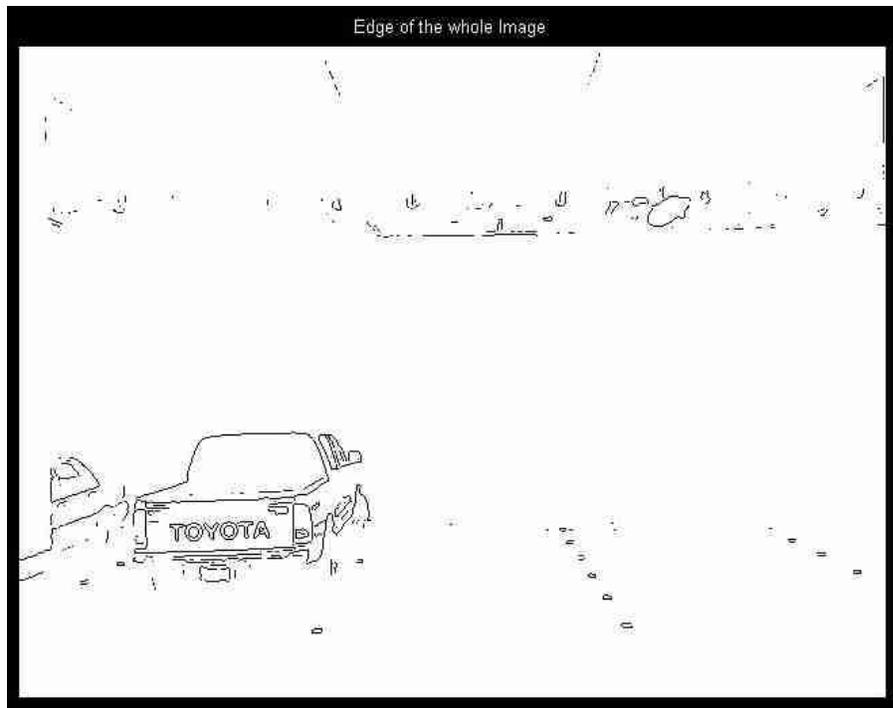


Figure 5.13: Edge detection of a fixed parking region with two vacant spaces and one occupied space

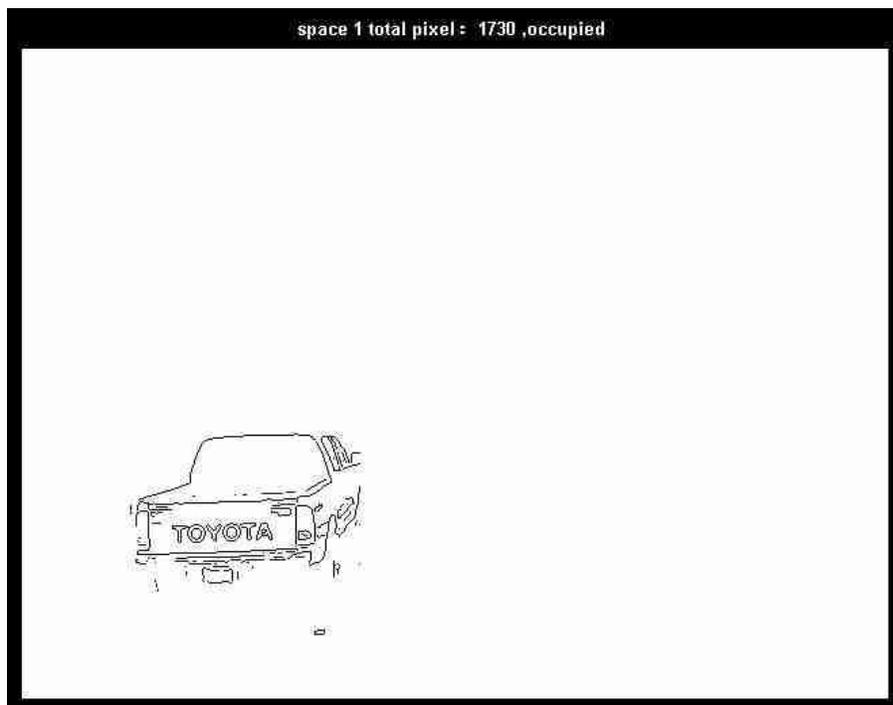


Figure 5.14: Edge pixel counting of a fixed region with two vacant spaces and one occupied space, space 1 containing 1730 edge pixels, detected as an occupied space

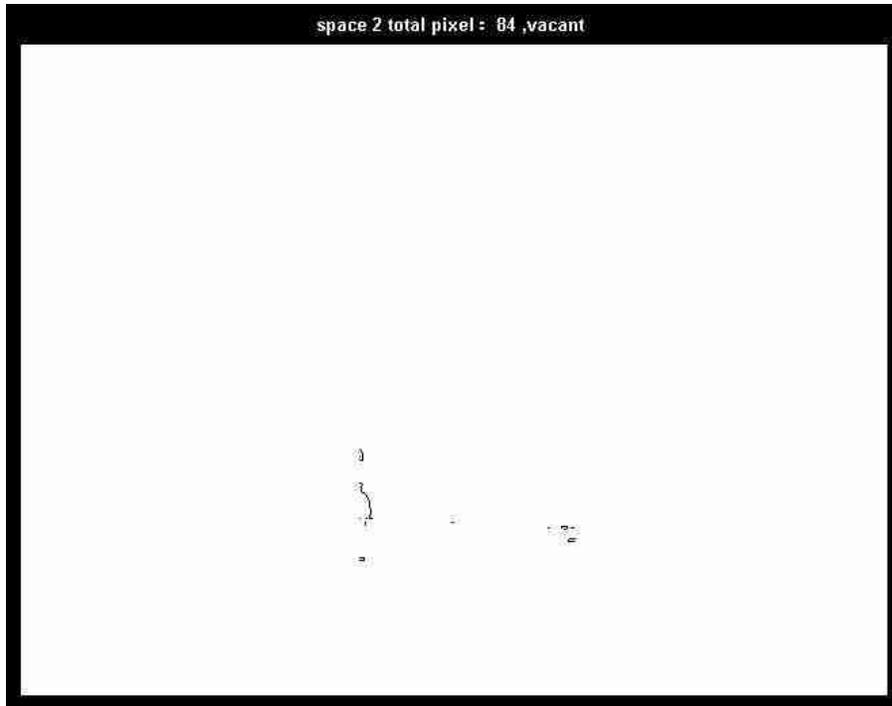


Figure 5.15: Edge pixel counting for space 2 of region with two vacant spaces and one occupied space, space 2 containing 84 edge pixels, detected as a vacant space

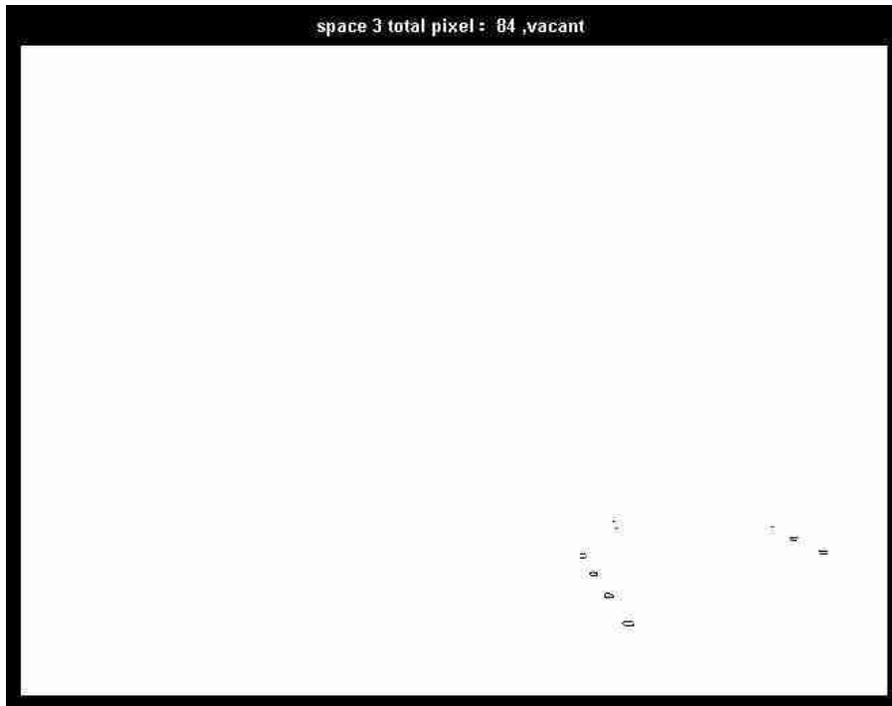


Figure 5.16: Edge pixel counting for space 3 of region with two vacant spaces and one occupied space, space 3 containing 84 edge pixels, detected as a vacant space

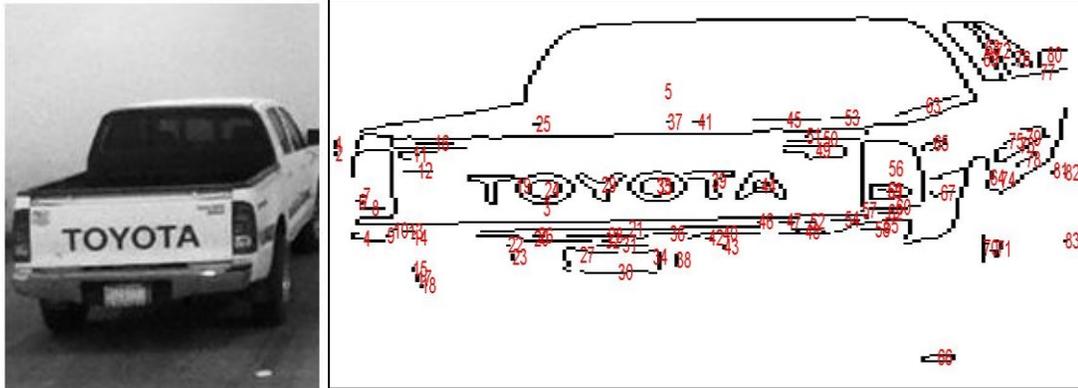


Figure 5.17: Space 1 object numbers of region with two vacant spaces and one occupied space, space 2 containing 83 objects, detected as an occupied space

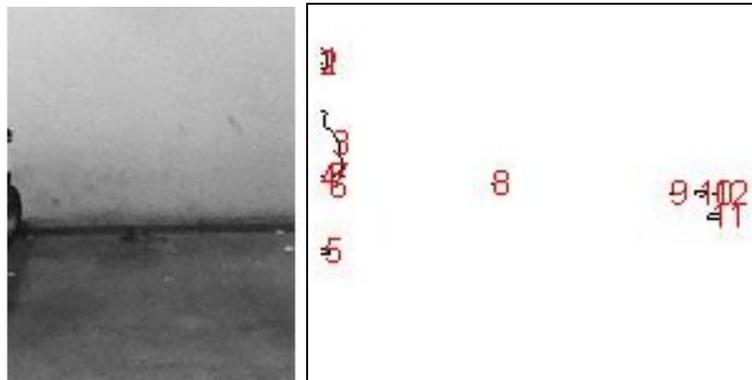


Figure 5.18: Space 2 object numbers of region with two vacant spaces and one occupied space, space 2 containing 12 objects, detected as a vacant space

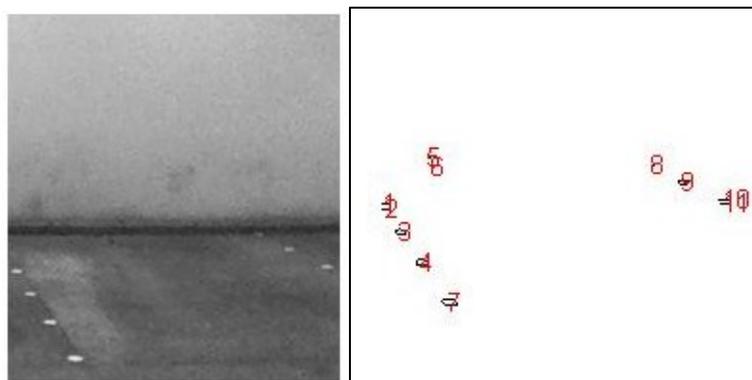


Figure 5.19: Space 3 object numbers of region with two vacant spaces and one occupied space, space 3 containing 11 objects, detected as a vacant space

For the foreground/background segmentation technique, we used the histogram as shown in Figure 5.20 and Figure 5.21 to identify the background pixels. Figure 5.22 shows the foreground information generated from image subtraction of both directions and the integrated value.

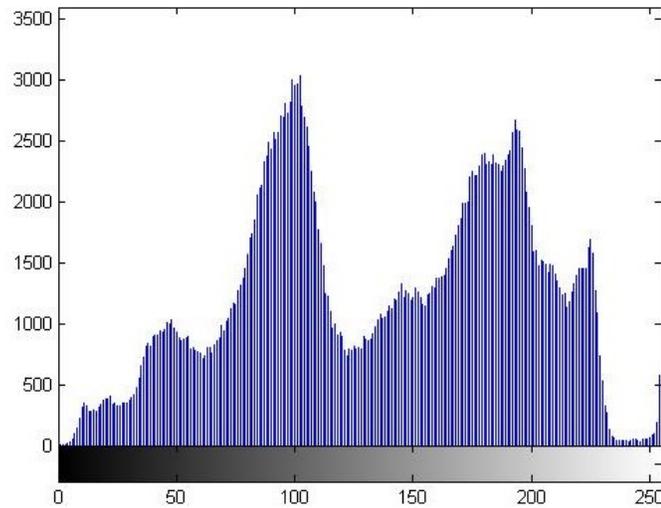


Figure 5.20: Histogram of a fixed parking region with two vacant spaces and one occupied space

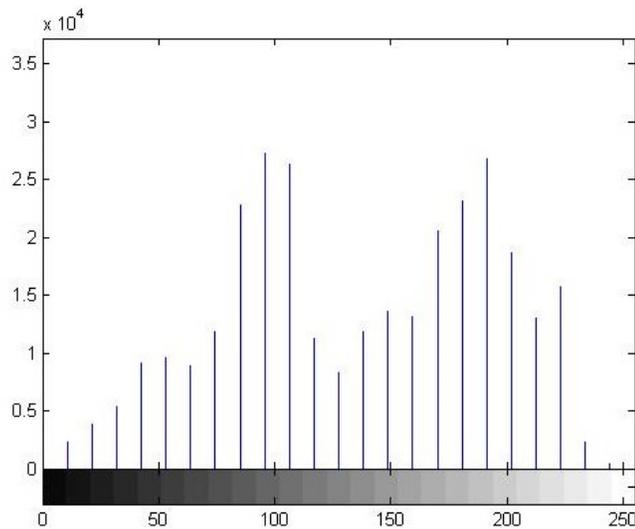


Figure 5.21: Histogram using 25 bins of a fixed parking region with two vacant spaces and one occupied space



(a) The dynamic background pixel value subtracts the gray-scale image



(b) The gray-scale image subtracts the dynamic background pixel value



(c) Integrated image of (a) and (b)

Figure 5.22: The foreground image generation of a fixed parking region with two vacant spaces and one occupied space

The final result summary of Figure 5.12 is shown in Table 5.15:

Table 5.2: Summary of Figure 5.12

Image Information	Image Date	2013-12-22 22.39.49		
	Place	Indoor		
	Bayesian Decision	Occupied	Vacant	Vacant
	Spaces	o1	v2	v3
Original Image	x	640		
	y	480		
	Total Pixels	307200		
Edge Base	X Region	210	210	210
	Y Region	170	160	190
	Region Total	35700	33600	39900
	Edge Pixels	1730(O)	84(V)	84(V)
	% of the whole image	0.563	0.0273	0.0273
	% of Region	4.84(O)	0.25(V)	0.21 (V)
	Objects	83(O)	12(V)	11(V)
Foreground/Background Method	BG value(dynamic)	95.625		
	FMPV	77(O)	51(O)	44(V)

The above table shows the detection result, where V stands for vacant, O stands for occupied.

2) Parking Region Containing one Vacant Space and two Occupied Spaces

In this section, we present the experimental result of a parking with one vacant parking space and two occupied parking spaces. Image name is given to show the date and time as well as being the index of the whole database. An input raw image with file name: 2013-09-27 21.06.38 is shown in Figure 5.23:



Figure 5.23: The input image of a fixed parking region with one vacant space and two occupied spaces

We repeat the whole processing applied in the previous section and show the results in the same order. We get the edge and region information using edge detection as shown in Figure 5.24. After that, Figures 5.25 to 5.27 show the edge pixel counting method while Figures 5.28 to 5.30 show the object counting method.

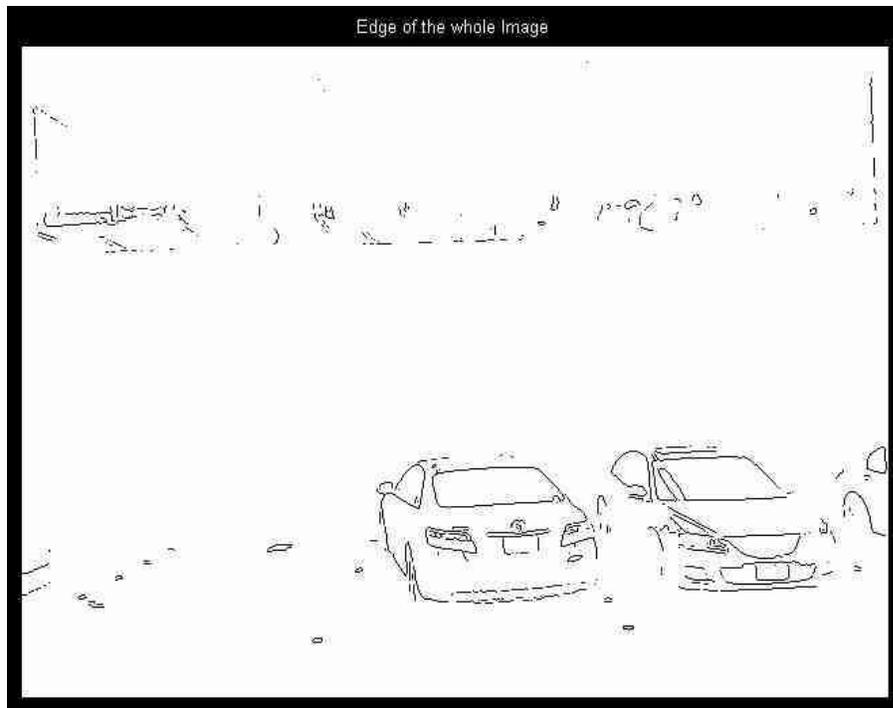


Figure 5.24: Edge information of a fixed parking region with one vacant space and two occupied spaces

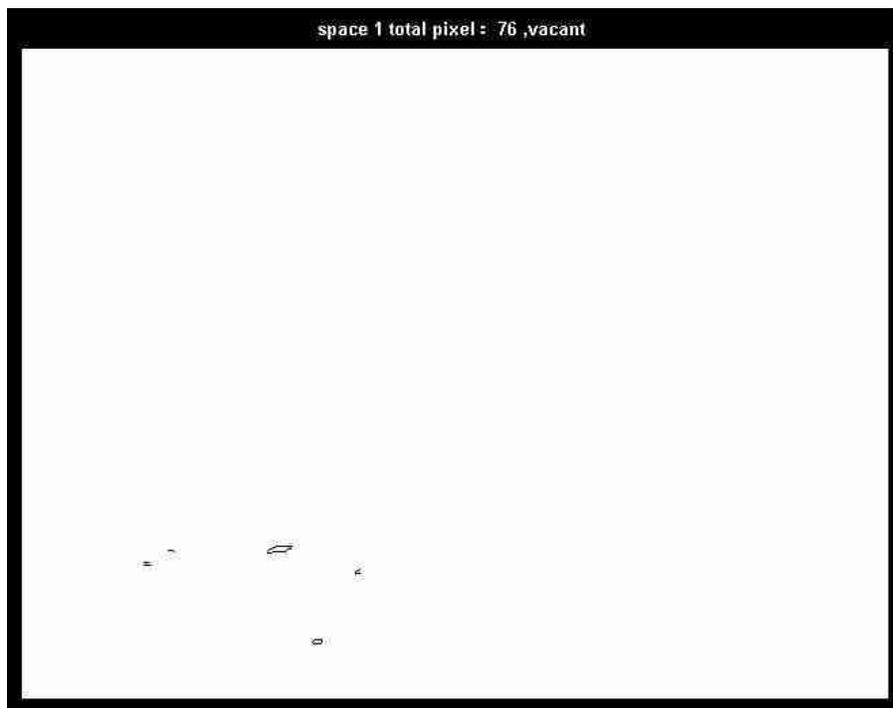


Figure 5.25: Edge pixel counting for space 1 of a fixed parking region with one vacant space and two occupied spaces, space 1 containing 76 edge pixels, detected as a vacant space

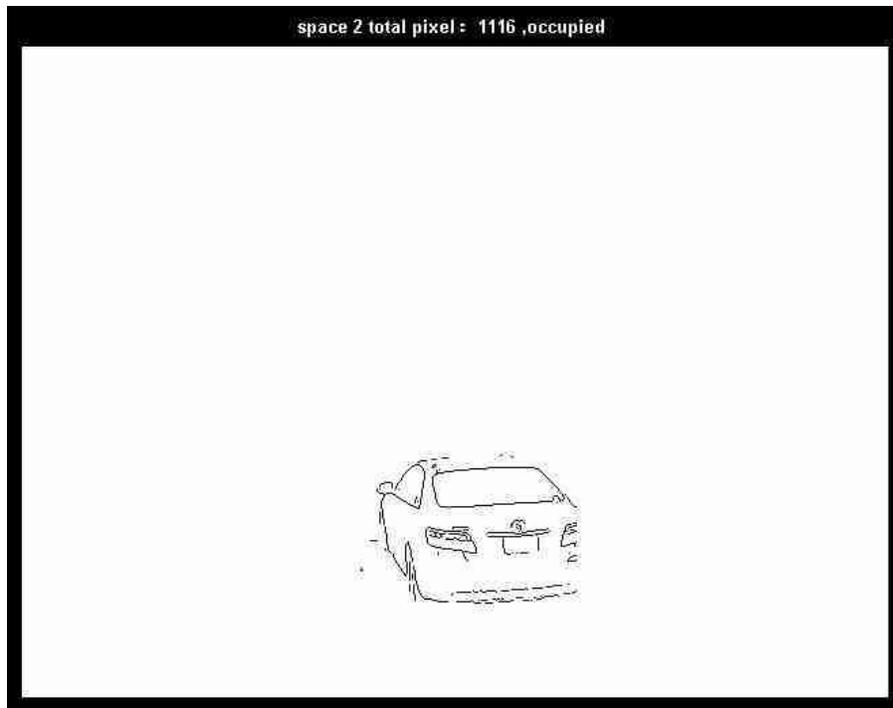


Figure 5.26: Edge pixel counting for space 2 of a fixed parking region with one vacant space and two occupied spaces, space 2 containing 1116 edge pixels, detected as an occupied space

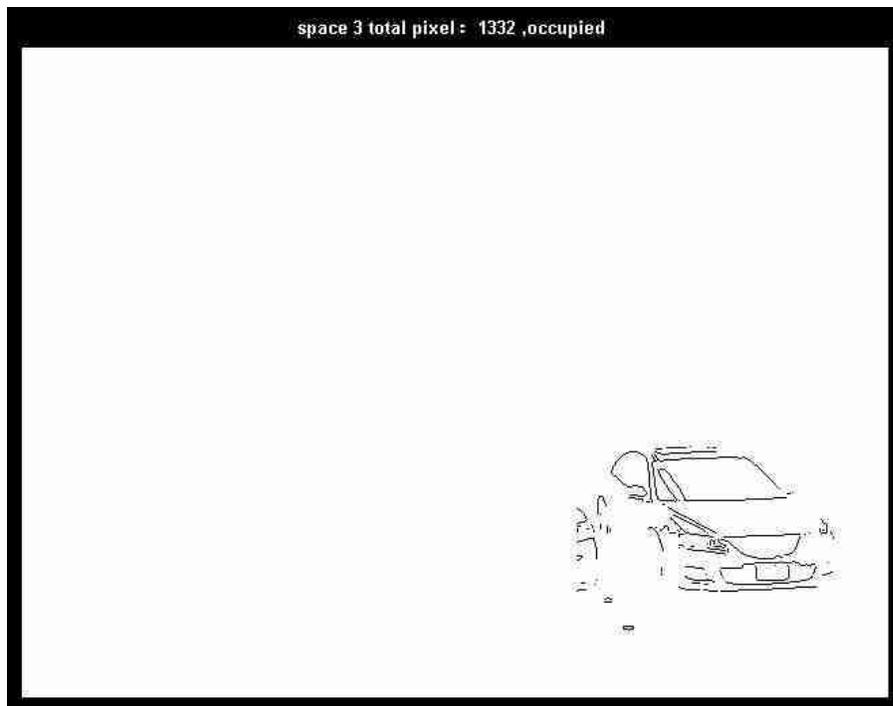


Figure 5.27: Edge pixel counting for space 3 of a fixed parking region with one vacant space and two occupied spaces, space 3 containing 1332 edge pixels, detected as an occupied space

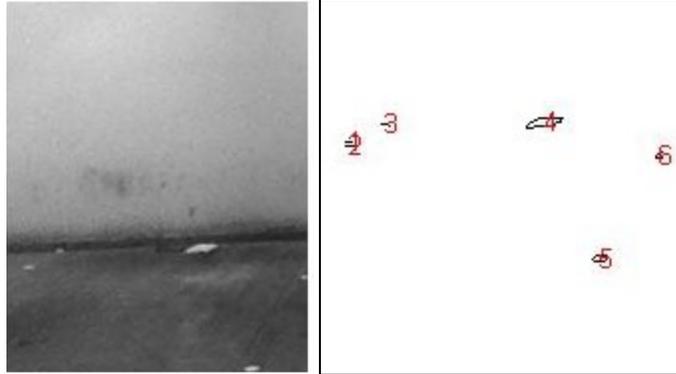


Figure 5.28: Space 1 object numbers of a fixed parking region with one vacant space and two occupied spaces, space 1 containing 6 objects, detected as a vacant space

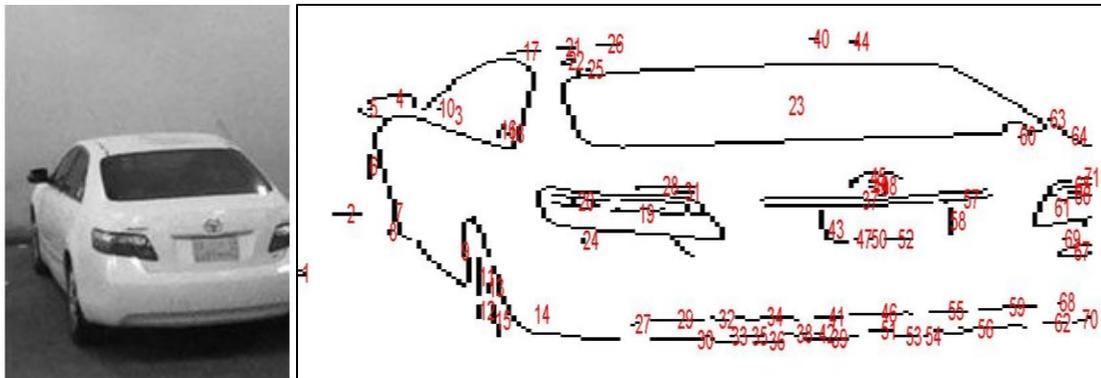


Figure 5.29: Space 2 object numbers of a fixed parking region with one vacant space and two occupied spaces, space 2 containing 71 objects, detected as an occupied space

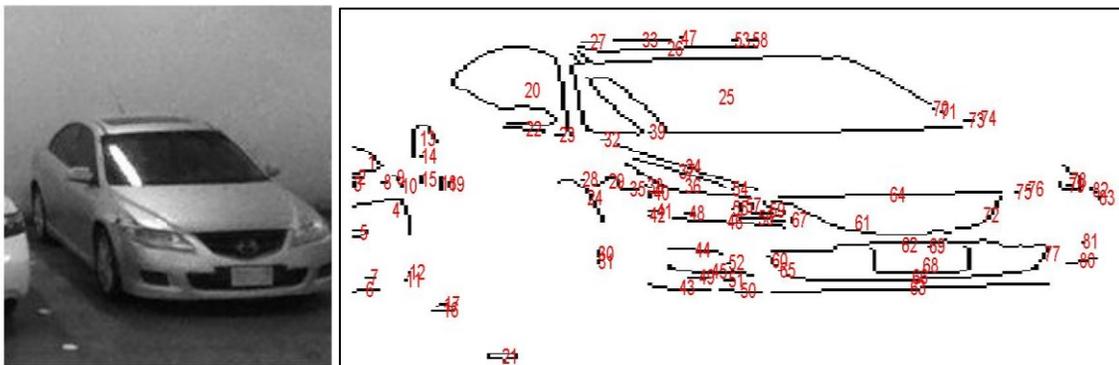


Figure 5.30: Space 3 object numbers of a fixed parking region with one vacant space and two occupied spaces, space 2 containing 83 objects, detected as an occupied space

For the foreground/background segmentation technique, we used the histogram as shown in Figure 5.31 and Figure 5.32 to identify the background pixels. Figure 5.33 shows the foreground information generated from image subtraction of both directions and the integrated value.

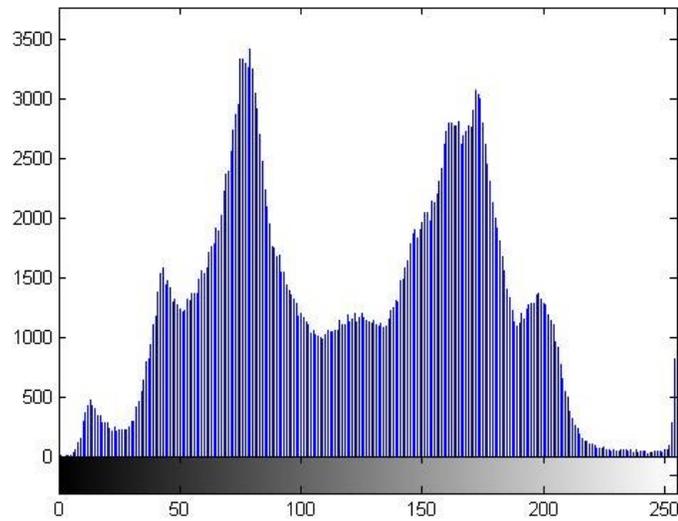


Figure 5.31: Histogram of a fixed parking region with one vacant space and two occupied spaces

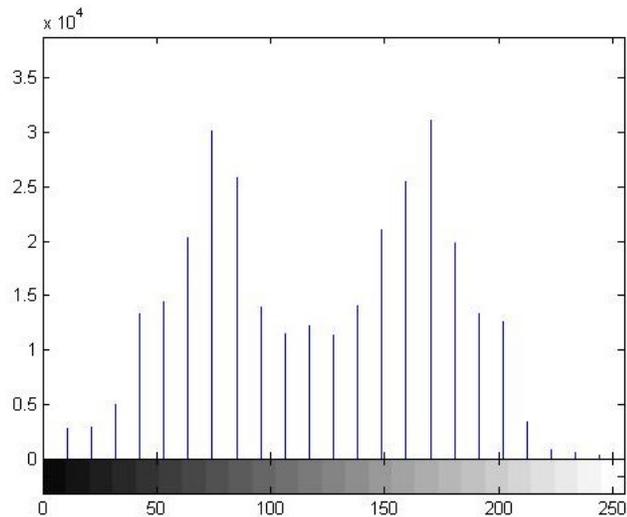


Figure 5.32: Histogram using 25 bins of a fixed parking region with one vacant space and two occupied spaces



(a) The dynamic background pixel value subtracts the gray-scale image



(b) The gray-scale image subtracts the dynamic background pixel value



(c) Integrated image of (a) and (b)

Figure 5.33: The foreground image generation of a fixed parking region with one vacant space and two occupied spaces

The final result summary of Figure 5.23 is shown in Table 5.3:

Table 5.3: Summary of Figure 5.23

Image Information	Image Date	2013-09-27		
	Place	Indoor		
	Bayesian Decision	Vacant	Occupied	Occupied
	Spaces	v1	o2	o3
Original Image	x	640		
	y	480		
	Total Pixels	307200		
Edge Based	X Region	210	210	210
	Y Region	170	160	19
	Region Total	35700	33600	39900
	Edge Pixels	76(V)	1116(O)	1332(O)
	% of the whole image	0.0247	0.3633	0.4336
	% of Region	0.21(V)	3.32(O)	3.34(O)
	Objects	6(V)	71(O)	83(O)
Foreground/Background Method	BG value(dynamic)	170		
	FMPV	49(V)	48(V)	65(O)

The above table shows the detection result, where V stands for vacant, O stands for occupied.

5.1.3 Processing of the Parking Image with no Vacant Parking Space

In this section, we present the experimental result of a parking with no vacant parking spaces and three occupied parking spaces. Image name is given to show the date and time as well as being the index of the whole database. An input raw image with file name: 2013-09-11 10.38.44 is shown as Figure 5.34 .



Figure 5.34: The input image of a fixed parking region with zero vacant space and three occupied spaces

We repeat the whole processing applied in the previous section and show the results in the same order. We get the edge and region information using edge detection as shown in Figure 5.35. After that, Figures 5.36 to 5.38 show the edge pixel counting method while Figures 5.39 to 5.41 show the object counting method.

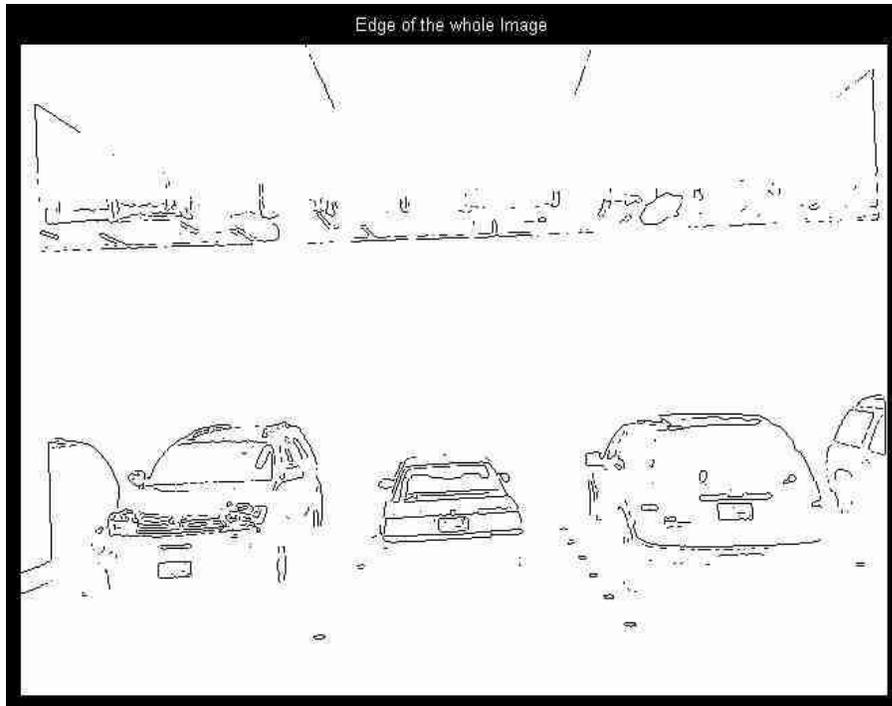


Figure 5.35: Edge information of a fixed parking region with zero vacant space and three occupied spaces

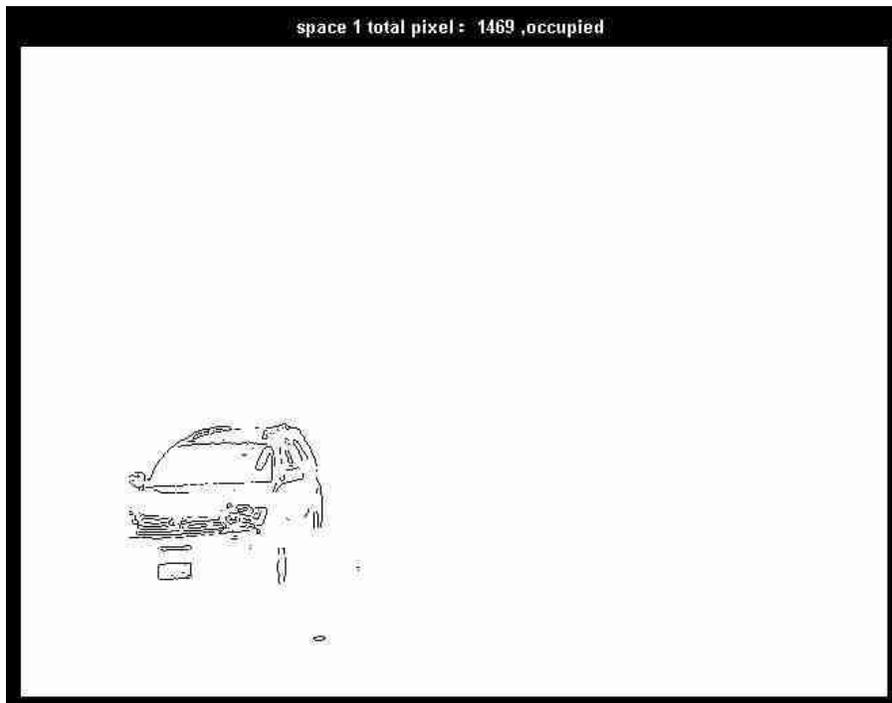


Figure 5.36: Edge pixel counting for space 1 of a fixed parking region with zero vacant space and three occupied spaces, space 1 containing 1469 edge pixels, detected as an occupied space

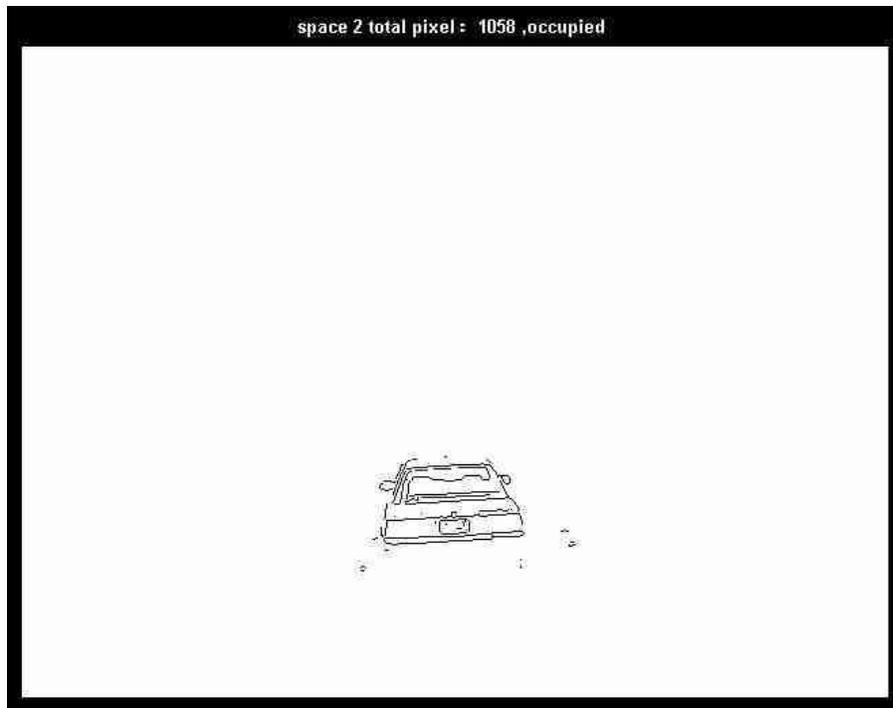


Figure 5.37: Edge pixel counting for space 2 of a fixed parking region with zero vacant space and three occupied spaces, space 2 containing 1058 edge pixels, detected as an occupied space

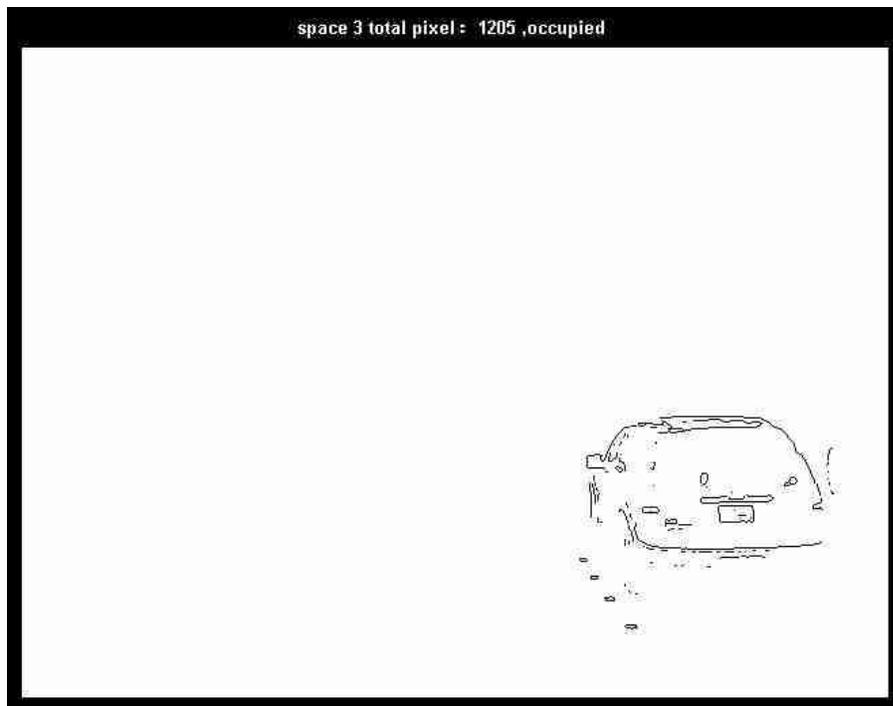


Figure 5.38: Edge pixel counting for space 3 of a fixed parking region with zero vacant space and three occupied spaces, space 3 containing 1205 edge pixels, detected as an occupied space

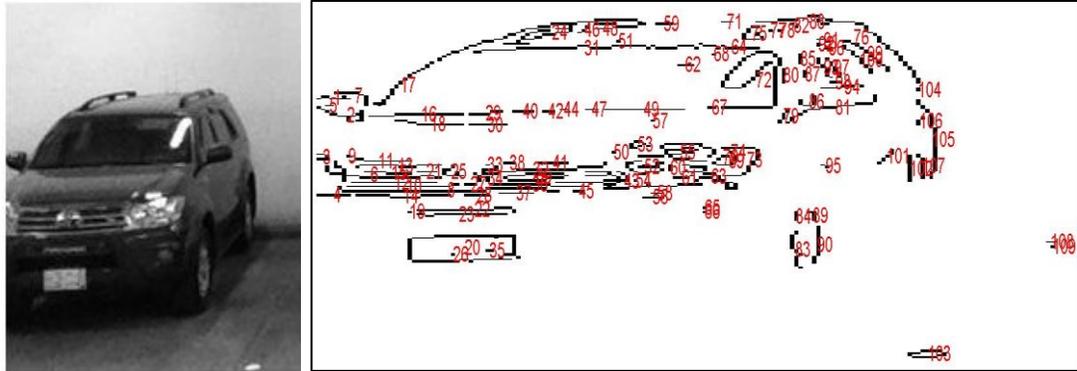


Figure 5.39: Space 1 object numbers of a fixed parking region with zero vacant space and three occupied spaces, space 1 containing 109 objects, detected as an occupied space

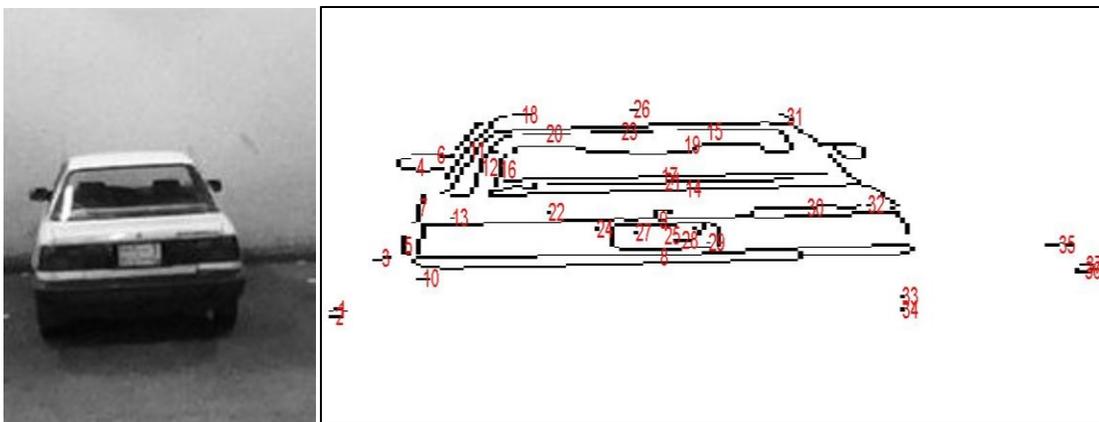


Figure 5.40: Space 2 object numbers of a fixed parking region with zero vacant space and three occupied spaces, space 2 containing 37 objects, detected as an occupied space

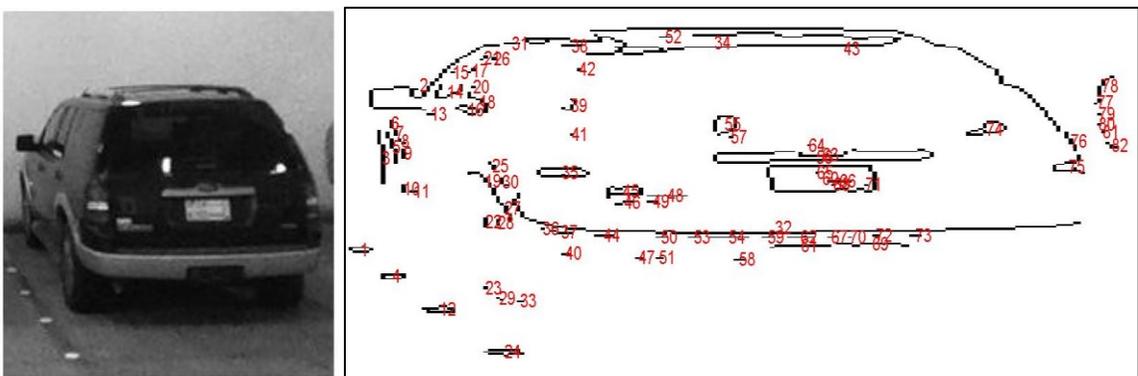


Figure 5.41: Space 3 object numbers of a fixed parking region with zero vacant space and three occupied spaces, space 3 containing 82 objects, detected as an occupied space

For the foreground/background segmentation technique, we used the histogram as shown in Figure 5.42 and Figure 5.43 to identify the background pixels. Figure 5.44 shows the foreground information generated from image subtraction of both directions and the integrated value.

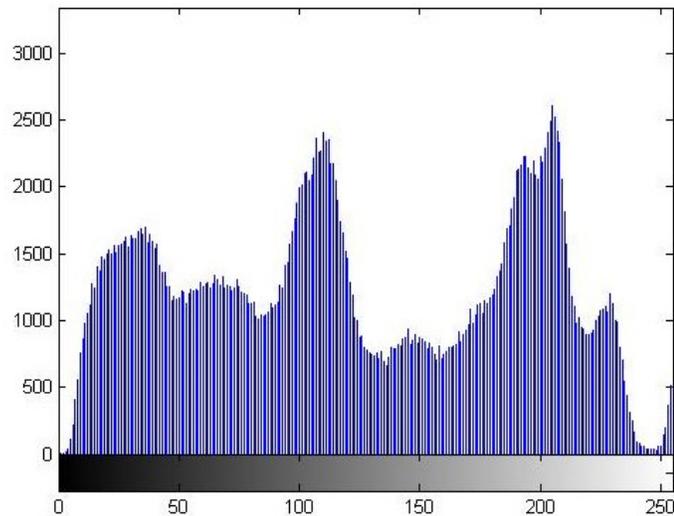


Figure 5.42: Histogram of a fixed parking region with zero vacant space and three occupied spaces

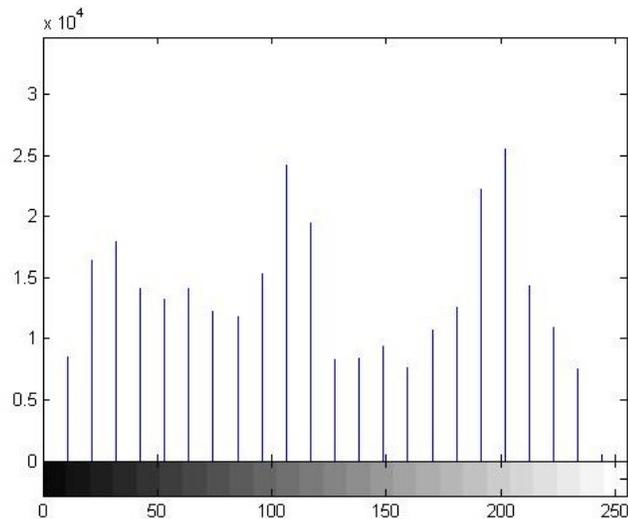
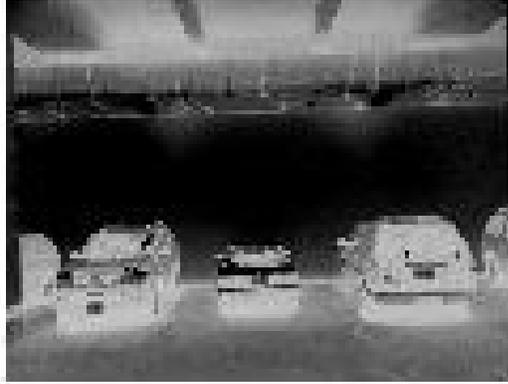


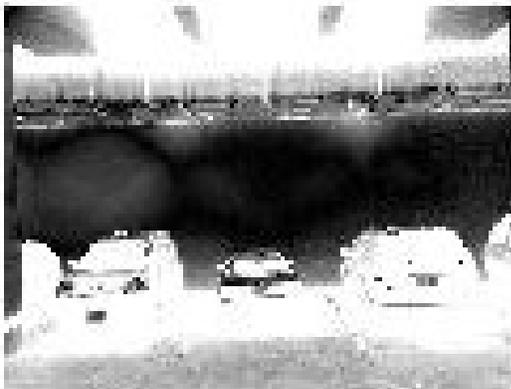
Figure 5.43: Histogram using 25 bins of a fixed parking region with zero vacant space and three occupied spaces



(a) The dynamic background pixel value subtracts the gray-scale image



(b) The gray-scale image subtracts the dynamic background pixel value



(c) Integrated image of (a) and (b)

Figure 5.44: The foreground generation of a fixed parking region with zero vacant space and three occupied spaces

The final result summary of Figure 5.34 is shown in Table 5.4:

Table 5.4: Summary of Figure 5.34

Image Information	Image Date	2013-09-11 10.38.44		
	Place	Indoor		
	Bayesian Decision	Occupied	Occupied	Occupied
	Spaces	o1	o2	o3
Original Image	x	640		
	y	480		
	Total Pixels	307200		
Edge Based	X Region	210	21	210
	Y Region	170	160	190
	Region Total	35700	33600	39900
	Edge Pixels	1469(O)	1058(O)	1205(O)
	% of the whole image	0.4782	0.3444	0.3923
	% of Region	4.11 (O)	3.15(O)	3.02 (O)
	Objects	109(O)	37(O)	82(O)
Foreground/Background Method	BG value(dynamic)	201.875		
	FMPV	86(O)	67 (O)	93(O)

The above table shows the detection result, where V stands for vacant, O stands for occupied. The case of no-vacant parking is also very useful. If we detect that the whole region do not have any vacant spaces, there is no need to do the parking space segmentation and figure out which parking space is available. It reduces the processing time while keeping a high recognition rate.

5.2 Bayesian Classification Using the Weka Package

5.2.1 Database Setting up

To transform the raw images to features used for classification, we use the same method to generate a database of more than 100 images and select 101 images taken at parking lot opposite to elevator exit of Floor B2, Building 59 which contains totally 303 complete parking spots and 101 half-parking spots. The 303 complete parking spots are used in the database as normal condition and several half-parking spots are added as special condition to test the stability of the whole system. The detection and classification will be judged by the naïve Bayes classifier. To test the feasibility of our database, we applied the first 27 images with 81 parking spots using MATLAB. We divided the data into three groups, each of which consists of 27 samples. Therefore, we circularly use one of the three groups as testing matrix and the remaining two groups as training database. If we use all the four features (The number of edge pixels, object numbers, edge pixel percentage and FMPV) together as the criterion, the detection result will be more accurate than use one of the features only. The reason is that when a single criterion is under consideration, there will be an error when the image is taken under some special conditions (e.g. image including noises, some dust and wasted bottles in the parking spaces, car under single color coverage, extreme light conditions).

In Table 5.5, single image size information of the database is presented where parking number 1, 2, 3 are complete parking space and number 4 is half-parking space.

Table 5.5: Image size information

Original Image Size Information			
Bld. 59 Sample	x	y	Image Total
Whole Image	640	480	307200
Region Size Information			
Space Number	xRegion	yRegion	Region Total
1(impact space)	210	170	35700
2(impact space)	210	160	33600
3(impact space)	210	190	39900
4(half-space)	210	40	8400

Then In Table 5.6, the entire database used in our project is presented. It contains all the features that will be considered in our vacant parking detection method. (C is the class item, “1” stand for occupied and “2” stand for vacant. S is the space number item; F/B Method is the brief of Foreground/Background Method).

Table 5.6: The entire database used in the project

Image Information			Edge Based Data				Foreground/ Background Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of Region	Objects	BGPixels (dynamic)	FMPV
2013-09-30 18.52.10	1	1	1185	0.385742188	3.319328	78	85	75.3181231
	1	2	819	0.266601563	2.4375	54	85	56.99368317
	1	3	1362	0.443359375	3.413534	71	85	58.26311083
2013-09-27 21.06.38	2	1	76	0.024739583	0.212885	6	170	48.7669209
	1	2	1116	0.36328125	3.321429	71	170	47.81149352
	1	3	1332	0.43359375	3.338346	83	170	65.24757539
2013-09-26 13.57.38	1	1	1647	0.536132813	4.613445	98	191.25	65.58489744
	1	2	574	0.186848958	1.708333	66	191.25	66.949096
	1	3	1269	0.413085938	3.180451	54	191.25	58.31902025
2013-09-25 14.14.12	1	1	1576	0.513020833	4.414566	99	201.875	70.58513437
	1	2	1084	0.352864583	3.22619	28	201.875	67.64325834
	1	3	693	0.225585938	1.736842	90	201.875	79.76229798
2013-09-19 16.32.33	2	1	89	0.028971354	0.2493	14	95.625	59.01255717

Table 5.6: The entire database used in the project (Con. 1)								
Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	2	2	94	0.030598958	0.279762	13	95.625	50.24437148
	2	3	92	0.029947917	0.230576	11	95.625	43.70325907
2013-09-18 16.56.48	1	1	699	0.227539063	1.957983	55	95.625	70.48564479
	2	2	56	0.018229167	0.166667	13	95.625	49.24799179
	2	3	88	0.028645833	0.220551	16	95.625	42.79482785
2013-09-17 20.06.09	1	1	1146	0.373046875	3.210084	86	95.625	66.55677218
	1	2	1089	0.354492188	3.241071	70	95.625	63.96094636
	2	3	87	0.028320313	0.218045	12	95.625	40.30080322
2013-09-17 10.32.46	1	1	1374	0.447265625	3.848739	117	191.25	87.26660478
	1	2	1064	0.346354167	3.166667	63	191.25	63.79711495
	1	3	917	0.298502604	2.298246	57	191.25	96.34544581
2013-09-11 19.04.39	1	1	1245	0.405273438	3.487395	74	95.625	68.64008203
	1	2	539	0.175455729	1.604167	46	95.625	57.13281931
	2	3	138	0.044921875	0.345865	21	95.625	43.98611076
2013-09-11 10.38.44	1	1	1469	0.478190104	4.114846	109	201.875	85.57640687
	1	2	1058	0.344401042	3.14881	37	201.875	67.32759923
	1	3	1205	0.392252604	3.02005	82	201.875	92.65317926
2013-11-24 18.54.49	1	1	1173	0.381835938	3.285714	75	191.25	63.82772364
	1	2	431	0.140299479	1.282738	28	191.25	74.67184958
	1	3	1214	0.395182292	3.042607	123	191.25	89.49939351
2013-11-21 10.43.41	1	1	720	0.234375	2.016807	61	201.875	75.23258344
	1	2	1194	0.388671875	3.553571	79	201.875	69.191651
	1	3	1370	0.445963542	3.433584	61	201.875	72.20662277
2013-11-21 10.43.38	1	1	720	0.234375	2.016807	63	201.875	75.40703297
	1	2	1192	0.388020833	3.547619	64	201.875	69.27814238
	1	3	1398	0.455078125	3.503759	67	201.875	71.99504988
2013-11-14 17.09.52	2	1	40	0.013020833	0.112045	6	85	57.59604014
	1	2	1046	0.340494792	3.113095	57	85	61.50220981
	2	3	100	0.032552083	0.250627	9	85	41.26052625
2013-11-13 17.40.13	1	1	1185	0.385742188	3.319328	82	95.625	73.95089927
	1	2	757	0.246419271	2.252976	82	95.625	62.40602143
	2	3	126	0.041015625	0.315789	14	95.625	45.60732339
2013-11-09 12.48.59	1	1	1062	0.345703125	2.97479	66	85	67.71380893
	2	2	79	0.025716146	0.235119	6	85	50.92134797
	1	3	981	0.319335938	2.458647	77	85	47.76744539
2013-11-07 17.02.21	1	1	1058	0.344401042	2.963585	89	191.25	74.41794049

Table 5.6: The entire database used in the project (Con. 2)								
Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	1	2	771	0.250976563	2.294643	53	191.25	73.88438696
	2	3	159	0.051757813	0.398496	23	191.25	56.88700479
2013-11-06 17.01.53	1	1	1213	0.394856771	3.397759	84	85	62.56308589
	2	2	31	0.010091146	0.092262	5	85	46.77503083
	1	3	1420	0.462239583	3.558897	102	85	47.92633513
2013-10-31 17.00.11	1	1	828	0.26953125	2.319328	60	85	74.34434465
	1	2	997	0.324544271	2.967262	48	85	62.72428956
	2	3	75	0.024414063	0.18797	9	85	41.3561691
2013-10-31 12.49.00	1	1	1144	0.372395833	3.204482	91	191.25	68.0671006
	1	2	655	0.213216146	1.949405	56	191.25	73.41435541
	1	3	1199	0.390299479	3.005013	72	191.25	68.75186368
2013-10-30 17.45.23	1	1	1051	0.342122396	2.943978	70	191.25	77.35892309
	1	2	556	0.180989583	1.654762	45	191.25	71.30533761
	1	3	597	0.194335938	1.496241	58	191.25	78.62111181
2013-10-02 17.15.19	2	1	108	0.03515625	0.302521	18	95.625	62.11730589
	2	2	52	0.016927083	0.154762	12	95.625	50.59177359
	2	3	92	0.029947917	0.230576	13	95.625	43.40464797
2013-10-03 16.47.06	2	1	115	0.037434896	0.322129	20	191.25	50.4504596
	1	2	1218	0.396484375	3.625	68	191.25	62.98735016
	1	3	1313	0.427408854	3.290727	86	191.25	68.29567782
2013-10-08 10.36.42	1	1	1532	0.498697917	4.291317	126	191.25	87.4380623
	1	2	1248	0.40625	3.714286	66	191.25	65.48786437
	1	3	1331	0.433268229	3.33584	74	191.25	67.98481471
2013-10-08 10.41.56	2	1	58	0.018880208	0.162465	15	95.625	61.86168309
	1	2	1183	0.385091146	3.520833	63	95.625	57.6620701
	1	3	1279	0.416341146	3.205514	60	95.625	58.73795688
2013-10-08 12.05.43	1	1	837	0.272460938	2.344538	74	85	67.13346994
	1	2	1146	0.373046875	3.410714	61	85	56.38318552
	1	3	1248	0.40625	3.12782	67	85	57.08456973
2013-10-11 13.46.36	2	1	64	0.020833333	0.179272	14	95.625	59.34191386
	2	2	40	0.013020833	0.119048	7	95.625	46.81489355
	2	3	94	0.030598958	0.235589	14	95.625	39.64058207
2013-10-23 16.44.07	1	1	1264	0.411458333	3.540616	77	95.625	72.74656244
	1	2	527	0.171549479	1.568452	51	95.625	60.20160349
	2	3	74	0.024088542	0.185464	10	95.625	42.01700887
2013-10-24 17.09.59	1	1	976	0.317708333	2.733894	82	191.25	79.13843534

Table 5.6: The entire database used in the project (Con. 3)

Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	2	2	68	0.022135417	0.202381	12	191.25	56.75634933
	1	3	1075	0.349934896	2.694236	100	191.25	71.20244531
2013-10-27 12.42.19	1	1	1140	0.37109375	3.193277	73	191.25	76.19461171
	1	2	1159	0.377278646	3.449405	80	191.25	60.04185718
	1	3	1007	0.327799479	2.52381	83	191.25	76.54352252
2013-10-27 15.16.32	1	1	1190	0.387369792	3.333333	79	201.875	76.11004691
	1	2	1177	0.383138021	3.502976	75	201.875	60.26863176
	1	3	1240	0.403645833	3.107769	81	201.875	90.75836195
2013-10-28 15.35.49	1	1	981	0.319335938	2.747899	64	85	67.51416183
	1	2	1123	0.365559896	3.342262	64	85	55.6735317
	1	3	1992	0.6484375	4.992481	102	85	63.46009372
2013-10-29 08.43.51	1	1	1035	0.336914063	2.89916	72	106.25	70.41300062
	1	2	954	0.310546875	2.839286	40	106.25	66.54613131
	1	3	1164	0.37890625	2.917293	91	106.25	52.18563327
2013-02-12 17.07.45	1	1	1231	0.400716146	3.448179	102	95.625	68.71916519
	2	2	50	0.016276042	0.14881	11	95.625	48.92447257
	2	3	102	0.033203125	0.255639	13	95.625	43.0372552
2013-02-13 14.19.25	1	1	859	0.279622396	2.406162	72	201.875	91.60419653
	1	2	648	0.2109375	1.928571	68	201.875	75.26574052
	1	3	1241	0.403971354	3.110276	101	201.875	70.73231311
2013-02-14 19.10.23	2	1	73	0.023763021	0.204482	12	85	63.81854759
	2	2	61	0.019856771	0.181548	13	85	52.37678663
	2	3	84	0.02734375	0.210526	11	85	46.0243123
2013-02-16 12.58.18	1	1	1237	0.402669271	3.464986	88	106.25	72.35818105
	1	2	681	0.221679688	2.026786	66	106.25	67.83991954
	1	3	685	0.222981771	1.716792	76	106.25	56.26873616
2013-02-16 14.51.30	1	1	1235	0.402018229	3.459384	79	191.25	61.58826512
	1	2	705	0.229492188	2.098214	65	191.25	74.80772247
	1	3	728	0.236979167	1.824561	90	191.25	78.32481363
2013-02-16 16.23.44	2	1	47	0.015299479	0.131653	9	95.625	60.53813064
	1	2	873	0.284179688	2.598214	51	95.625	64.00829603
	2	3	66	0.021484375	0.165414	9	95.625	41.34713125
2013-02-18 13.13.45	1	1	1171	0.381184896	3.280112	84	191.25	63.64866464
	1	2	1334	0.434244792	3.970238	101	191.25	64.59058553
	2	3	124	0.040364583	0.310777	17	191.25	64.36724325
2013-02-19 18.31.42	1	1	1509	0.491210938	4.226891	77	191.25	55.73717569

Table 5.6: The entire database used in the project (Con. 4)

Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	2	2	158	0.051432292	0.470238	16	191.25	62.47694237
	1	3	1284	0.41796875	3.218045	82	191.25	81.62768102
2013-02-19 23.00.44	2	1	38	0.012369792	0.106443	8	95.625	60.49978693
	1	2	1404	0.45703125	4.178571	68	95.625	63.06128138
	2	3	70	0.022786458	0.175439	10	95.625	42.14723139
2013-02-20 14.12.09	1	1	905	0.294596354	2.535014	62	191.25	77.58887888
	1	2	666	0.216796875	1.982143	58	191.25	75.61462463
	1	3	1868	0.608072917	4.681704	126	191.25	68.21016863
2013-02-23 13.09.55	1	1	1248	0.40625	3.495798	55	191.25	58.61078291
	1	2	665	0.216471354	1.979167	75	191.25	73.54458637
	1	3	794	0.258463542	1.989975	72	191.25	93.50794715
2013-02-25 13.00.54	1	1	918	0.298828125	2.571429	47	170	55.62994311
	1	2	298	0.097005208	0.886905	27	170	71.12129182
	1	3	1079	0.351236979	2.704261	55	170	53.52843095
2013-02-27 14.34.54	2	1	27	0.008789063	0.07563	5	180.625	51.54028797
	1	2	401	0.130533854	1.193452	32	180.625	72.75886557
	1	3	1063	0.346028646	2.66416	64	180.625	54.78521002
2013-03-04 11.21.09	1	1	659	0.214518229	1.845938	50	191.25	89.82186587
	1	2	442	0.143880208	1.315476	43	191.25	74.99607034
	1	3	1407	0.458007813	3.526316	86	191.25	63.37364127
2013-03-05 14.29.02	1	1	1236	0.40234375	3.462185	76	191.25	71.88113266
	1	2	565	0.183919271	1.681548	58	191.25	73.62042334
	1	3	1050	0.341796875	2.631579	79	191.25	65.46754744
2013-03-11 10.56.14	1	1	1246	0.405598958	3.490196	70	95.625	79.04344893
	1	2	578	0.188151042	1.720238	65	95.625	64.1464761
	1	3	991	0.322591146	2.483709	86	95.625	50.34106872
2013-03-16 16.59.01	1	1	1701	0.553710938	4.764706	114	170	58.51685788
	1	2	1218	0.396484375	3.625	74	170	65.13390386
	1	3	482	0.156901042	1.20802	52	170	74.39167722
2013-03-19 16.20.22	1	1	1782	0.580078125	4.991597	106	85	77.33442508
	2	2	108	0.03515625	0.321429	9	85	50.91333091
	2	3	74	0.024088542	0.185464	9	85	43.00641321
2013-08-21 09.41.42	1	1	1115	0.362955729	3.123249	44	106.25	75.80154246
	1	2	1040	0.338541667	3.095238	53	106.25	57.52657713
	1	3	584	0.190104167	1.463659	43	106.25	55.18240619
2013-08-22 18.21.25	2	1	72	0.0234375	0.201681	14	95.625	60.51179259

Table 5.6: The entire database used in the project (Con. 5)								
Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	2	2	61	0.019856771	0.181548	14	95.625	49.12485202
	1	3	667	0.217122396	1.671679	71	95.625	54.92518801
2013-08-26 11.57.25	1	1	991	0.322591146	2.77591	53	201.875	61.93231822
	1	2	772	0.251302083	2.297619	72	201.875	84.42287122
	1	3	708	0.23046875	1.774436	63	201.875	78.70863291
2013-08-27 14.24.19	1	1	1025	0.333658854	2.871148	58	106.25	66.77375362
	1	2	1015	0.330403646	3.020833	34	106.25	63.68802128
	1	3	584	0.190104167	1.463659	53	106.25	55.1271795
2013-08-28 19.42.14	2	1	67	0.021809896	0.187675	14	95.625	62.34528302
	2	2	54	0.017578125	0.160714	13	95.625	49.81820661
	1	3	734	0.238932292	1.839599	81	95.625	54.0386889
2013-08-29 10.56.19	1	1	1151	0.374674479	3.22409	48	95.625	73.76822953
	2	2	38	0.012369792	0.113095	9	95.625	47.03255267
	1	3	647	0.210611979	1.621554	80	95.625	52.45916357
2013-08-29 17.48.55	1	1	706	0.229817708	1.977591	57	95.625	70.61305746
	2	2	83	0.027018229	0.247024	16	95.625	50.85115789
	1	3	700	0.227864583	1.754386	68	95.625	54.22003637
2013-09-01 13.39.56	1	1	498	0.162109375	1.394958	45	106.25	72.48216749
	1	2	996	0.32421875	2.964286	34	106.25	64.9641463
	1	3	571	0.185872396	1.431078	46	106.25	55.89142078
2013-09-01 19.17.50	1	1	487	0.158528646	1.364146	40	201.875	73.36436545
	1	2	1073	0.349283854	3.193452	71	201.875	75.75977849
	1	3	558	0.181640625	1.398496	50	201.875	76.9926491
2013-09-02 14.29.56	1	1	1550	0.504557292	4.341737	106	95.625	66.44003203
	1	2	975	0.317382813	2.901786	44	95.625	60.74398919
	1	3	692	0.225260417	1.734336	62	95.625	53.54230255
2013-09-02 18.23.47	1	1	1127	0.366861979	3.156863	100	95.625	71.12596621
	1	2	987	0.321289063	2.9375	63	95.625	60.38888723
	2	3	104	0.033854167	0.260652	19	95.625	45.09649502
2013-09-06 19.04.21	1	1	1473	0.479492188	4.12605	77	85	75.46811749
	2	2	42	0.013671875	0.125	9	85	51.5352389
	2	3	100	0.032552083	0.250627	12	85	45.05316814
2013-09-10 15.34.43	1	1	691	0.224934896	1.935574	54	191.25	691
	1	2	1301	0.423502604	3.872024	55	191.25	1301
	1	3	1343	0.437174479	3.365915	76	191.25	1343
2013-10-01 10.08.20	1	1	1194	0.388671875	3.344538	58	95.625	71.54151518

Table 5.6: The entire database used in the project (Con. 6)

Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	1	2	517	0.168294271	1.53869	55	95.625	60.37058543
	1	3	1076	0.350260417	2.696742	86	95.625	45.8280773
2013-10-08 10.41.28	2	1	108	0.03515625	0.302521	14	106.25	59.48881553
	1	2	1080	0.3515625	3.214286	60	106.25	58.35954737
	1	3	1186	0.386067708	2.972431	55	106.25	61.89191767
2013-10-08 12.05.45	1	1	875	0.284830729	2.45098	83	85	69.40963872
	1	2	1160	0.377604167	3.452381	68	85	58.52423908
	1	3	1271	0.413736979	3.185464	60	85	58.90502329
2013-10-08 13.00.29	1	1	959	0.312174479	2.686275	84	201.875	63.58035815
	1	2	1179	0.383789063	3.508929	73	201.875	69.42939088
	1	3	1303	0.424153646	3.265664	69	201.875	72.55042441
2013-10-16 14.01.45	2	1	45	0.014648438	0.12605	8	85	59.87409014
	2	2	37	0.012044271	0.110119	8	85	47.05970603
	2	3	63	0.020507813	0.157895	11	85	39.54721803
2013-10-16 16.30.13	2	1	37	0.012044271	0.103641	5	74.375	57.0835645
	2	2	32	0.010416667	0.095238	6	74.375	43.1559846
	2	3	89	0.028971354	0.223058	10	74.375	35.46120623
2013-10-18 19.07.40	2	1	24	0.0078125	0.067227	7	85	65.31219436
	2	2	29	0.009440104	0.08631	7	85	48.24928532
	2	3	87	0.028320313	0.218045	9	85	37.93497901
2013-11-05 15.47.49	2	1	41	0.013346354	0.114846	8	106.25	59.73574893
	1	2	682	0.222005208	2.029762	66	106.25	64.70312453
	1	3	705	0.229492188	1.766917	71	106.25	57.56179211
2013-12-03 12.11.08	1	1	1056	0.34375	2.957983	77	223.125	62.96660322
	1	2	887	0.288736979	2.639881	41	223.125	58.10740358
	1	3	881	0.286783854	2.20802	51	223.125	67.44760948
2013-12-03 19.14.01	2	1	95	0.030924479	0.266106	16	85	64.58368509
	1	2	777	0.252929688	2.3125	85	85	62.95692487
	2	3	109	0.035481771	0.273183	14	85	46.94249274
2013-12-04 12.01.19	1	1	1806	0.587890625	5.058824	94	191.25	69.93254607
	1	2	566	0.184244792	1.684524	61	191.25	79.07897354
	1	3	1231	0.400716146	3.085213	59	191.25	72.19994428
2013-12-04 17.02.54	2	1	53	0.017252604	0.148459	10	85	65.63783646
	2	2	35	0.011393229	0.104167	5	85	52.69515271
	2	3	74	0.024088542	0.185464	9	85	45.94702408
2013-12-05 17.13.09	2	1	43	0.013997396	0.120448	4	85	64.39399949

Table 5.6: The entire database used in the project (Con. 7)								
Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	1	2	852	0.27734375	2.535714	72	85	60.74300616
	2	3	70	0.022786458	0.175439	11	85	45.3235863
2013-12-08 11.25.43	1	1	1180	0.384114583	3.305322	100	191.25	65.08375162
	1	2	1156	0.376302083	3.440476	64	191.25	59.61058259
	1	3	626	0.203776042	1.568922	72	191.25	79.62679063
2013-12-08 14.31.16	1	1	1166	0.379557292	3.266106	99	180.625	64.41454617
	1	2	1095	0.356445313	3.258929	73	180.625	57.73636747
	1	3	662	0.215494792	1.659148	61	180.625	77.38009583
2013-12-09 11.21.02	1	1	1189	0.387044271	3.330532	112	201.875	87.11929997
	1	2	315	0.102539063	0.9375	28	201.875	73.95140275
	1	3	672	0.21875	1.684211	50	201.875	79.17459433
2013-12-12 17.39.57	2	1	52	0.016927083	0.145658	6	85	61.91308072
	2	2	43	0.013997396	0.127976	9	85	50.22248804
	1	3	1098	0.357421875	2.75188	85	85	51.63949381
2013-12-14 18.30.18	1	1	1047	0.340820313	2.932773	54	191.25	54.35729022
	1	2	1042	0.339192708	3.10119	94	191.25	67.31599758
	1	3	1000	0.325520833	2.506266	85	191.25	73.70531639
2013-12-15 20.44.34	1	1	1281	0.416992188	3.588235	66	201.875	62.62155905
	1	2	1120	0.364583333	3.333333	95	201.875	79.30823704
	1	3	1689	0.549804688	4.233083	93	201.875	77.40324624
2013-12-18 15.28.59	1	1	747	0.243164063	2.092437	69	180.625	86.37852446
	1	2	1187	0.386393229	3.532738	51	180.625	59.77414274
	1	3	478	0.155598958	1.197995	41	180.625	80.62821676
2013-12-19 10.45.00	1	1	865	0.281575521	2.422969	94	170	62.58311524
	1	2	1295	0.421549479	3.854167	68	170	57.10346738
	1	3	949	0.308919271	2.378446	101	170	86.6896813
2013-12-22 00.20.06	2	1	41	0.013346354	0.114846	5	85	61.33422445
	2	2	37	0.012044271	0.110119	8	85	48.74513007
	1	3	752	0.244791667	1.884712	57	85	52.34815673
2013-12-22 17.40.15	1	1	583	0.189778646	1.633053	33	191.25	74.35949574
	1	2	1049	0.341471354	3.122024	52	191.25	66.06959896
	2	3	94	0.030598958	0.235589	14	191.25	65.14178333
2013-12-22 22.39.49	1	1	1730	0.563151042	4.845938	83	95.625	76.58818993
	2	2	84	0.02734375	0.25	12	95.625	50.78998918
	2	3	84	0.02734375	0.210526	11	95.625	43.78670304
2013-12-23 17.34.06	1	1	944	0.307291667	2.644258	57	191.25	71.30597162

Table 5.6: The entire database used in the project (Con. 8)

Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	1	2	1209	0.393554688	3.598214	61	191.25	66.5604601
	1	3	1117	0.363606771	2.799499	93	191.25	83.26882496
2013-12-24 14.35.01	1	1	625	0.203450521	1.7507	52	85	55.49673782
	1	2	420	0.13671875	1.25	47	85	53.37226103
	1	3	937	0.305013021	2.348371	84	85	44.90223379
2013-12-26 14.21.40	1	1	708	0.23046875	1.983193	52	95.625	61.01311098
	1	2	914	0.297526042	2.720238	68	95.625	54.8393953
	2	3	75	0.024414063	0.18797	8	95.625	38.49317704
2013-12-29 19.02.18	1	1	1209	0.393554688	3.386555	72	106.25	70.19017551
	2	2	50	0.016276042	0.14881	10	106.25	48.99401018
	1	3	1980	0.64453125	4.962406	113	106.25	57.38043369
2013-12-31 08.14.50	1	1	2019	0.657226563	5.655462	120	191.25	84.12707392
	1	2	603	0.196289063	1.794643	59	191.25	75.16726673
	1	3	1364	0.444010417	3.418546	129	191.25	85.54283726
2013-12-31 16.02.10	1	1	656	0.213541667	1.837535	49	95.625	73.25207737
	2	2	146	0.047526042	0.434524	17	95.625	48.99084661
	1	3	1144	0.372395833	2.867168	68	95.625	52.19995716
2013-12-31 16.02.14-1	1	1	664	0.216145833	1.859944	48	191.25	76.6509077
	2	2	138	0.044921875	0.410714	18	191.25	59.65356647
	1	3	1242	0.404296875	3.112782	69	191.25	69.86932849
2014-01-01 21.59.09	2	1	58	0.018880208	0.162465	12	85	67.00710713
	2	2	195	0.063476563	0.580357	24	85	53.32569928
	2	3	133	0.043294271	0.333333	14	85	45.83368233
2014-01-14 12.20.30	1	1	697	0.226888021	1.952381	62	95.625	63.57497953
	2	2	220	0.071614583	0.654762	20	95.625	45.95929429
	1	3	994	0.323567708	2.491228	88	95.625	46.62678332
2014-01-21 10.13.58	1	1	667	0.217122396	1.868347	58	191.25	76.91870355
	1	2	1106	0.360026042	3.291667	69	191.25	64.98909416
	1	3	1116	0.36328125	2.796992	75	191.25	72.87434588
2014-01-26 07.25.29	1	1	1564	0.509114583	4.380952	117	201.875	87.07237397
	1	2	1114	0.362630208	3.315476	38	201.875	67.06917688
	1	3	784	0.255208333	1.964912	85	201.875	79.66723885
2014-01-26 15.47.45	2	1	80	0.026041667	0.22409	13	106.25	57.20169842
	1	2	1321	0.430013021	3.931548	54	106.25	61.58079933
	1	3	596	0.194010417	1.493734	49	106.25	58.48564281
2014-01-29 13.19.35	1	1	1231	0.400716146	3.448179	58	201.875	65.92409508

Image Information			Edge Based Data				F/B Method	
Image Name Date and Time	C	S	Edge Pixels	Percentage of the Whole	Percentage of the Region	Objects	BGPixels (dynamic)	FMPV
	1	2	862	0.280598958	2.565476	57	201.875	67.73866442
	1	3	1807	0.588216146	4.528822	126	201.875	76.94995092
2014-01-29 13.19.35	1	4	291	0.094726563	3.464286	18	201.875	103.3270744
2014-01-26 07.25.29	1	4	169	0.055012842	2.011905	12	201.875	85.51218929
2013-12-09 11.21.02	1	4	154	0.050129882	1.833333	11	201.875	80.21038806
2013-10-08 13.00.29	1	4	228	0.074218025	2.714286	16	201.875	70.95459427
2013-08-26 11.57.25	1	4	227	0.073892267	2.702381	24	201.875	68.4278714
2013-02-13 14.19.25	1	4	128	0.041665989	1.52381	11	201.875	63.29633401
2013-10-27 15.16.32	1	4	338	0.110023893	4.02381	26	201.875	72.79468833
2013-11-21 10.43.38	1	4	165	0.053709714	1.964286	15	201.875	71.32638806
2013-11-21 10.43.41	1	4	154	0.050128903	1.833333	19	201.875	71.63586437

5.2.2 The Performance of Bayesian Classifier Using the Weka Package

To get more useful information, we applied the data of 101 images using Weka package. There are two classes of each item, class 1 stands for occupied parking space and class 2 means vacant. Three main features used in our system are Edge Pixels Number, Objects Number and FMPV. At the same time, some subordinate features are also taken into consideration. Those subordinate features used as references of main features are: Parking Space Position, Image Length, Image Width, Image Size, Parking Spot Length, Parking Spot Width, Parking Spot Size, Edge Pixel Percentage of the whole image, Edge Pixel Percentage of the Parking Spot, and Dynamic Background Pixel Value. Such features are called the subordinate features because they are either the same when considered with the same spot or influenced by some other features. For example, for the images taken by a fixed camera, the edge pixel percentage is influenced by the number of edge pixels and the parking spot size. The subordinate features are useful when we are going to enhance our work to unfixed position parking images in the future.

Firstly we deal with the normal case which only contains data from complete. Figure 5.45 shows the statistic of the input data:

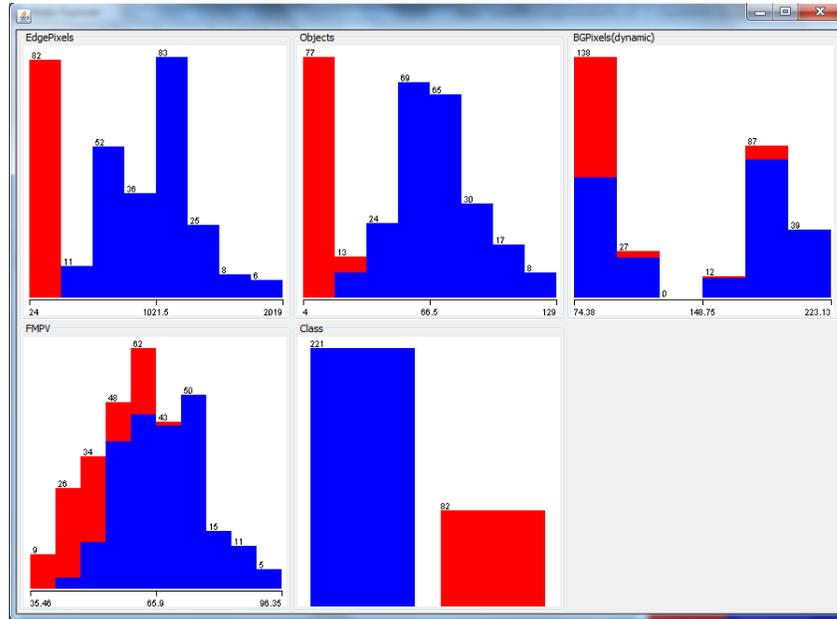


Figure 5.45: Statistic of 303 normal parking spaces

We can see that the data are divided into different intervals for each feature. Data from occupied spaces are marked as blue and data from vacant spaces are marked as red.

Secondly, naïve Bayes classifier is applied to the three main features one by one.

Out of the total 303 normal cases, edge pixel counting method provides 2 errors with ideal ROC curve, object counting method also provides 2 errors with ideal ROC, but foreground/background method gives 101 errors with the ROC area of 0.79. The outputs from the naïve Bayes classifier using three methods one by one with corresponding ROC curve are shown in Figures 5.46 to 5.51:

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      301          99.3399 %
Incorrectly Classified Instances    2            0.6601 %
Kappa statistic                    0.9832
Mean absolute error                0.0084
Root mean squared error            0.0719
Relative absolute error            2.1341 %
Root relative squared error        16.1846 %
Total Number of Instances          303

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                -----  -----  -
                1      0.024   0.991     1      0.995     1      Occupied
                0.976   0       1      0.976   0.988     1      Vacant
Weighted Avg.   0.993   0.018   0.993     0.993   0.993     1

=== Confusion Matrix ===

  a  b  <-- classified as
221  0  |  a = Occupied
  2  80 |  b = Vacant

```

Figure 5.46: Output of Weka naive Bayes classifier for the feature of the number of edge pixels

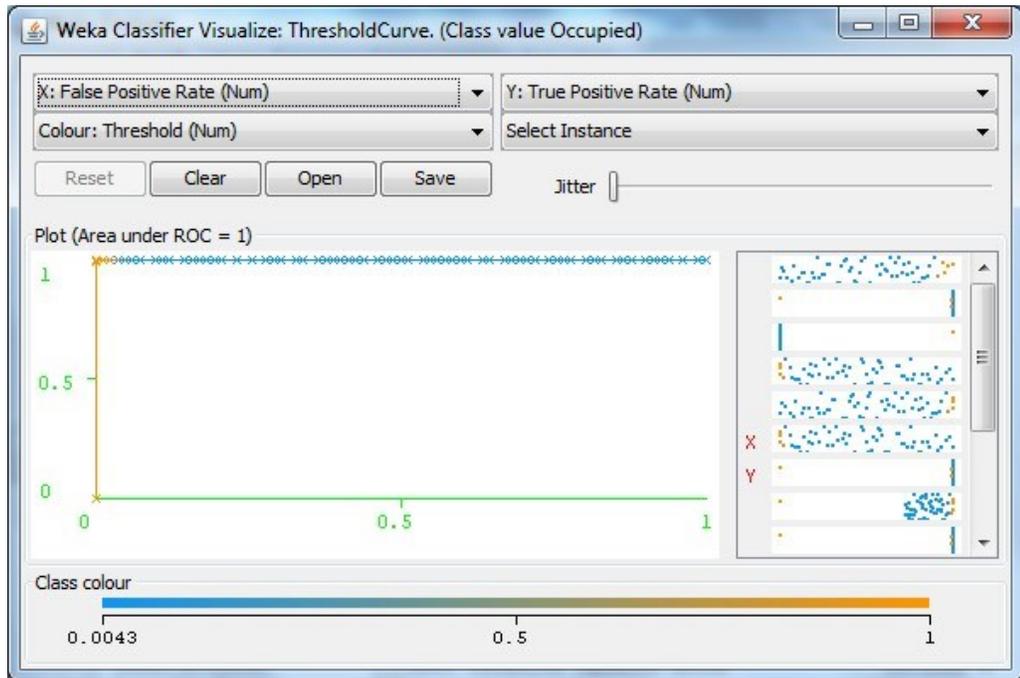


Figure 5.47: ROC curve of the output in Figure 5.46

The recognition accuracy using edge pixel counting is 99.34%. The ROC area is 1.000.

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      301          99.3399 %
Incorrectly Classified Instances    2            0.6601 %
Kappa statistic                    0.9832
Mean absolute error                 0.0098
Root mean squared error            0.0589
Relative absolute error             2.4661 %
Root relative squared error        13.2613 %
Total Number of Instances          303

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          1      0.024   0.991     1      0.995     1        Occupied
          0.976   0       1         0.976  0.988     1        Vacant
Weighted Avg.  0.993   0.018   0.993     0.993  0.993     1

=== Confusion Matrix ===

  a  b  <-- classified as
221  0  |  a = Occupied
  2  80 |  b = Vacant

```

Figure 5.48: Output of Weka naive Bayes classifier for the feature of object number

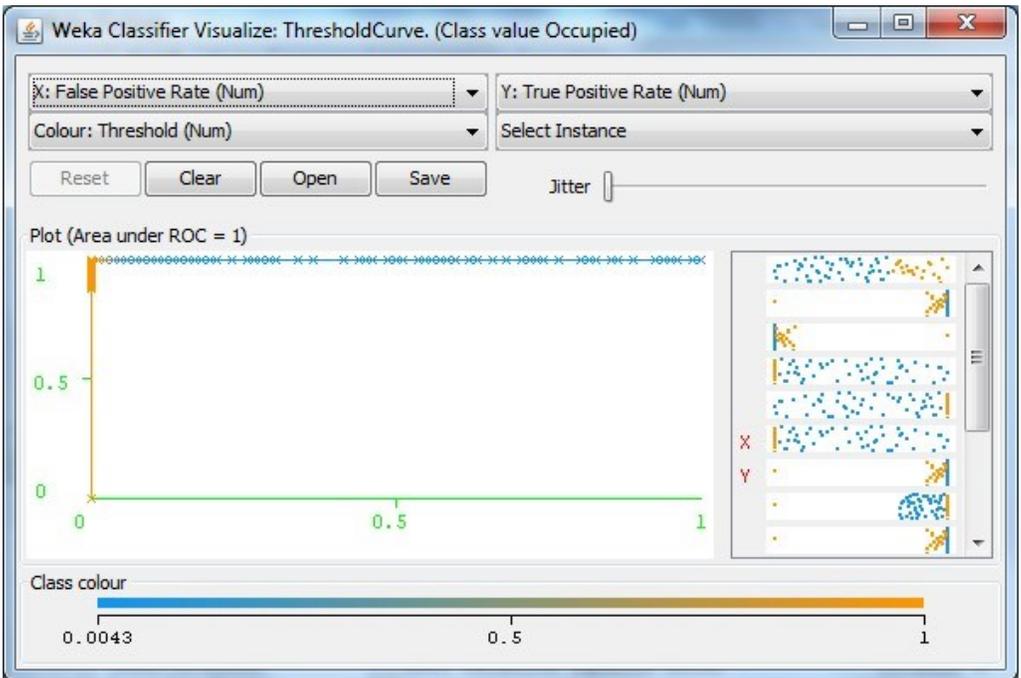


Figure 5.49: ROC curve of the output in Figure 5.48

The recognition accuracy using object counting is 99.34%. The ROC area is 1.000.

```

=== Summary ===

Correctly Classified Instances      257          84.8185 %
Incorrectly Classified Instances    46           15.1815 %
Kappa statistic                    0.6001
Mean absolute error                0.2297
Root mean squared error            0.3415
Relative absolute error             58.0679 %
Root relative squared error        76.8602 %
Total Number of Instances          303

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                0.919   0.341   0.879     0.919   0.898     0.888    Occupied
                0.659   0.081   0.75      0.659   0.701     0.888    Vacant
Weighted Avg.   0.848   0.271   0.844     0.848   0.845     0.888

=== Confusion Matrix ===

  a  b  <-- classified as
203 18 |  a = Occupied
 28 54 |  b = Vacant

```

Figure 5.50: Output of Weka naive Bayes classifier for the feature of FMPV

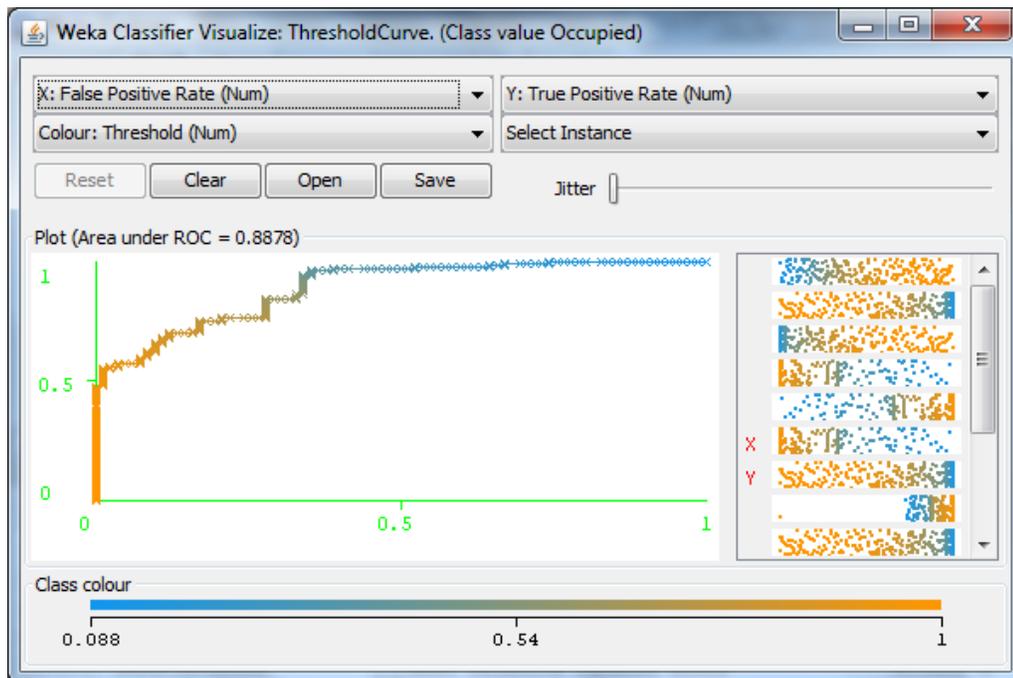


Figure 5.51: ROC curve of the output in Figure 5.50

The recognition accuracy using FMVP is 84.82%. The ROC area is 0.888.

We can see that the first two outputs show excellent results while the third one does not perform very well. This is because dynamic background pixel value is chosen and both white wall and dark road offers some weight to the background pixel value. On the other hand, the cars themselves also contribute to the background pixel value. Especially, when all parking spaces are occupied, the cars offer a relatively large weight to the background pixel value. However, FMVP feature is still very important because when we applied the three features together, the errors reduce to 0 under normal conditions, as shown in Figure 5.52:

```

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      303          100 %
Incorrectly Classified Instances    0            0 %
Kappa statistic                    1
Mean absolute error                 0.0009
Root mean squared error            0.0089
Relative absolute error             0.2373 %
Root relative squared error        2.0112 %
Total Number of Instances          303

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          1      0      1          1      1          1      Occupied
          1      0      1          1      1          1      Vacant
Weighted Avg.  1      0      1          1      1          1

=== Confusion Matrix ===

 a  b  <-- classified as
221  0 | a = Occupied
  0 82 | b = Vacant

```

Figure 5.52: Output of Weka naive Bayes classifier for all three features

We can see that the image-based vacant parking detection system using hybrid features performs very well. When the three main features are taken into consideration under normal condition, the system achieves a recognition accuracy of 100% for all 303 parking spaces.

Thirdly, we deal with dataset with several special cases which contains data from half parking region. Figure 5.53 shows the statistic of the input data:

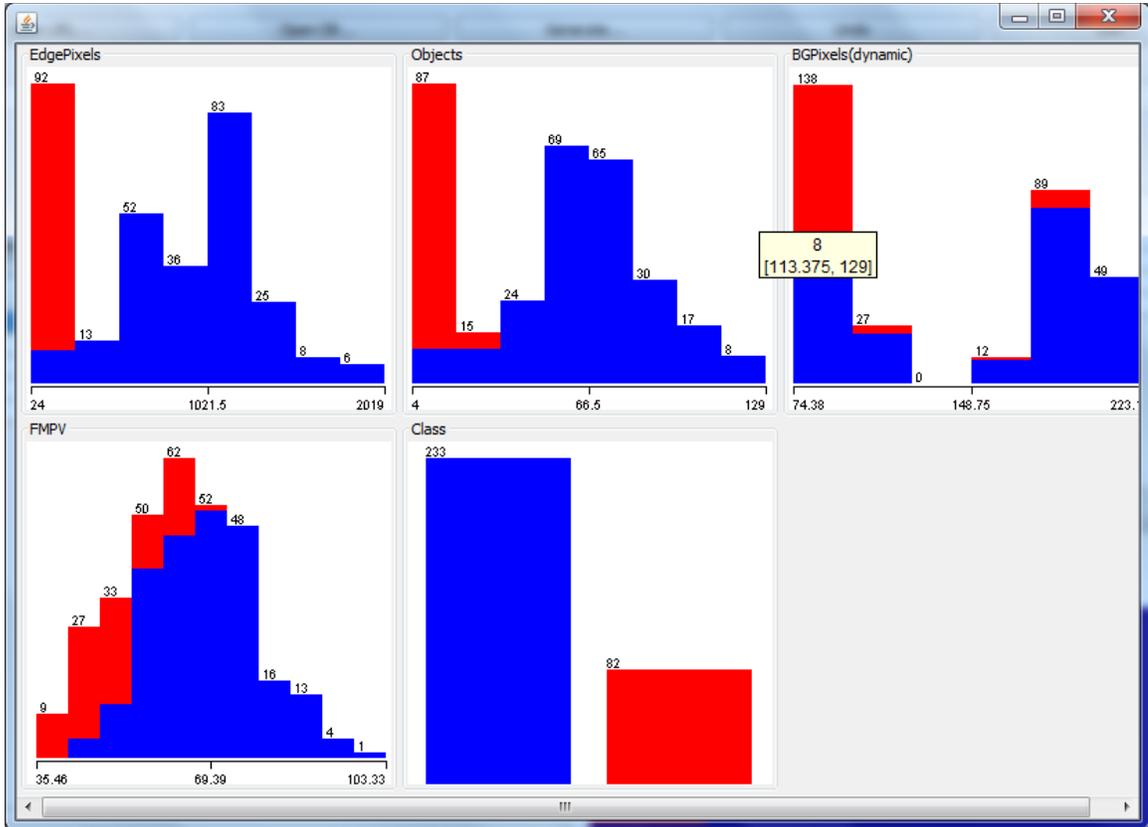


Figure 5.53: Statistic with special cases

We can see that the data are divided into different intervals for each feature. Data from occupied spaces are marked as blue and data from vacant spaces are marked as red.

For the database with special cases, naïve Bayes classifier is applied to the three main features plus a secondary feature dynamic background pixel value.

Also, the outputs from the naïve Bayes classifier using four methods one by one with corresponding ROC curve are shown in Figures 5.54 to 5.61:

```

=== Summary ===
Correctly Classified Instances      305          96.8254 %
Incorrectly Classified Instances    10           3.1746 %
Kappa statistic                    0.9195
Mean absolute error                0.0344
Root mean squared error            0.1583
Relative absolute error             8.9239 %
Root relative squared error        36.079 %
Total Number of Instances          315

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.966   0.024   0.991     0.966   0.978     0.998   Occupied
          0.976   0.034   0.909     0.976   0.941     0.998   Vacant
Weighted Avg.  0.968   0.027   0.97      0.968   0.969     0.998

=== Confusion Matrix ===

  a  b  <-- classified as
225  8  |  a = Occupied
  2 80  |  b = Vacant

```

Figure 5.54: Weka naive Bayes classifier for the feature of the number of edge pixels (with special cases)

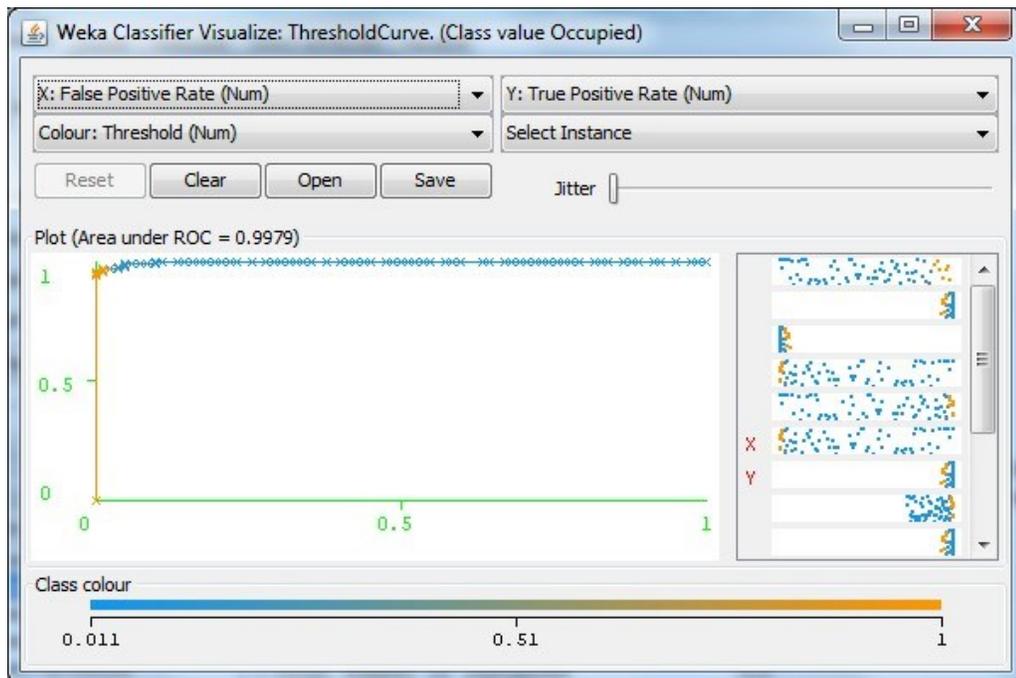


Figure 5.55: ROC curve of the output in Figure 5.54

The recognition accuracy using edge pixel counting is 96.83%. The ROC area is 0.998.

```

=== Summary ===

Correctly Classified Instances      302          95.873 %
Incorrectly Classified Instances    13           4.127 %
Kappa statistic                    0.8957
Mean absolute error                0.0496
Root mean squared error            0.1855
Relative absolute error            12.8641 %
Root relative squared error        42.2596 %
Total Number of Instances          315

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.957   0.037   0.987     0.957   0.972     0.979   Occupied
          0.963   0.043   0.888     0.963   0.924     0.979   Vacant
Weighted Avg.  0.959   0.038   0.961     0.959   0.959     0.979

=== Confusion Matrix ===

  a  b  <-- classified as
223 10 | a = Occupied
  3 79 | b = Vacant

```

Figure 5.56: Weka naive Bayes classifier for the feature of object number (with special cases)

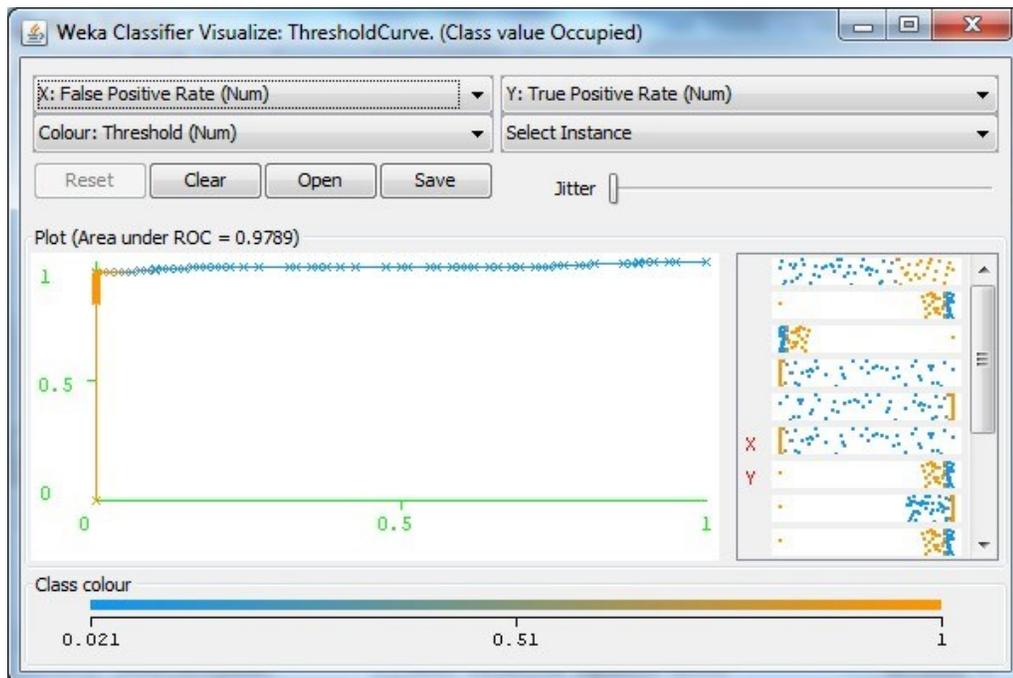


Figure 5.57: ROC curve of the output in Figure 5.56

The recognition accuracy using object counting is 95.87%. The ROC area is 0.979.

```

=== Summary ===

Correctly Classified Instances      269      85.3968 %
Incorrectly Classified Instances    46      14.6032 %
Kappa statistic                    0.6052
Mean absolute error                0.2213
Root mean squared error            0.3345
Relative absolute error            57.3303 %
Root relative squared error        76.2146 %
Total Number of Instances          315

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.923   0.341   0.885     0.923   0.903     0.895   Occupied
          0.659   0.077   0.75      0.659   0.701     0.895   Vacant
Weighted Avg.   0.854   0.273   0.85      0.854   0.851     0.895

=== Confusion Matrix ===

  a  b  <-- classified as
215 18 | a = Occupied
 28 54 | b = Vacant

```

Figure 5.58: Weka naive Bayes classifier for the feature of FMPV (with special cases)

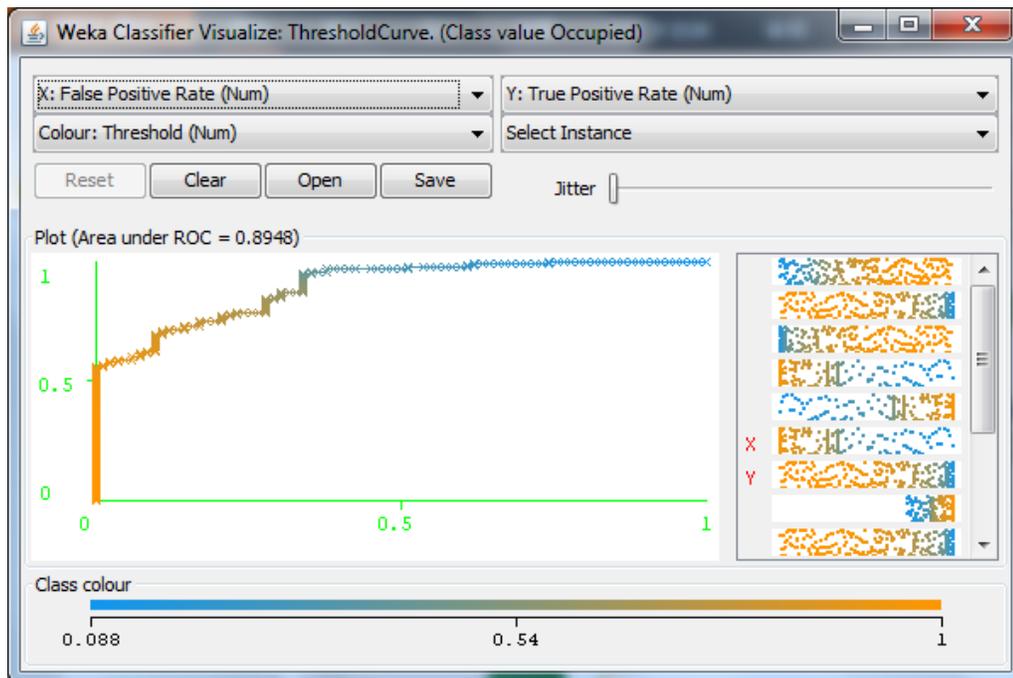


Figure 5.59: ROC curve of the output in Figure 5.58

The recognition accuracy using FMVP is 85.39%. The ROC area is 0.895.

```

=== Summary ===

Correctly Classified Instances      233          73.9683 %
Incorrectly Classified Instances    82           26.0317 %
Kappa statistic                    0.4465
Mean absolute error                 0.2927
Root mean squared error             0.3968
Relative absolute error             75.8437 %
Root relative squared error         90.4203 %
Total Number of Instances          315

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                0.704   0.159   0.927     0.704   0.8         0.788   Occupied
                0.841   0.296   0.5       0.841   0.627     0.788   Vacant
Weighted Avg.   0.74    0.194   0.816     0.74    0.755     0.788

=== Confusion Matrix ===

  a  b  <-- classified as
164 69 | a = Occupied
 13 69 | b = Vacant

```

Figure 5.60: Weka naive Bayes classifier for dynamic background value (with special cases)

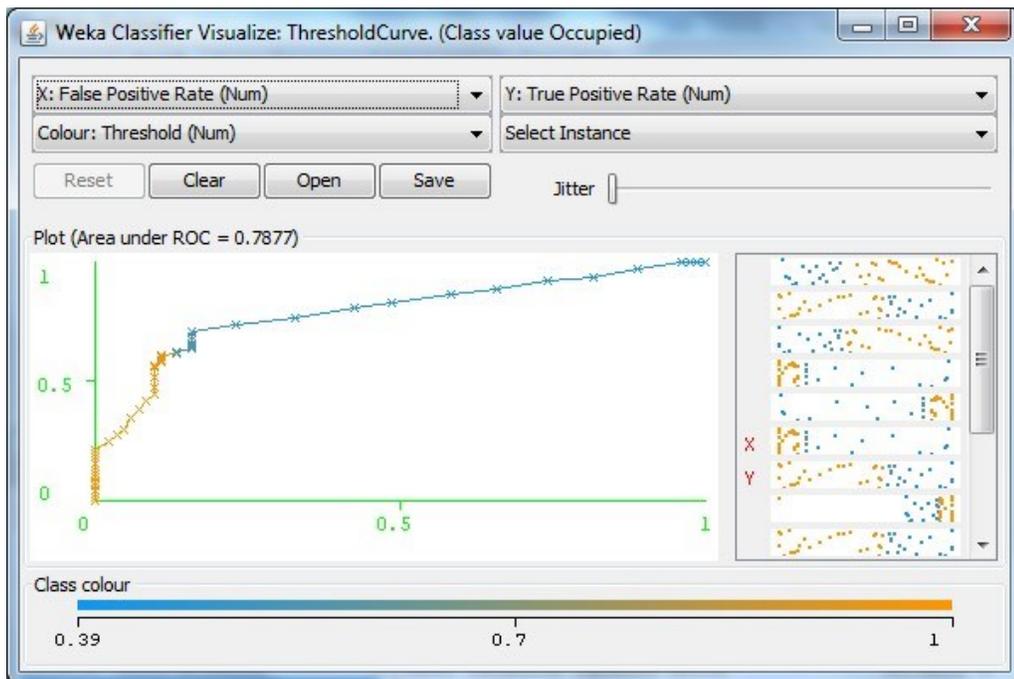


Figure 5.61: ROC curve of the output in Figure 5.60

The recognition accuracy using dynamic background value is 73.97%. The ROC area is 0.788.

The same process is also applied for the database with special cases; we can see that the recognition accuracy increases to 97.12% when using three main features together (as shown in Figure 5.62)

```

=== Summary ===

Correctly Classified Instances      307          97.4603 %
Incorrectly Classified Instances     8           2.5397 %
Kappa statistic                     0.9351
Mean absolute error                 0.0231
Root mean squared error             0.1379
Relative absolute error             5.9828 %
Root relative squared error         31.4346 %
Total Number of Instances          315

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.974   0.024   0.991     0.974   0.983     0.999   Occupied
          0.976   0.026   0.93      0.976   0.952     0.999   Vacant
Weighted Avg.  0.975   0.025   0.975     0.975   0.975     0.999

=== Confusion Matrix ===

  a  b  <-- classified as
227  6  |  a = Occupied
  2 80  |  b = Vacant

```

Figure 5.62: Output of Weka naive Bayes classifier for three main features (with special cases)

We can see that after integrating all three features (The number of edge pixels, object number, FMPV), the recognition rate increases a little bit to 97.46%. However, compared with the detection result using single feature of the number of edge pixels, only one detection error is corrected. This is due to the influence of the special cases. Therefore, it is necessary to add some secondary features to improve the recognition accuracy, although extra detection time is required. The system achieves an accuracy of 98.73% for all the three main features plus a secondary feature, dynamic background pixel value (as shown in Figure 5.63).

```

=== Summary ===
Correctly Classified Instances      311          98.7302 %
Incorrectly Classified Instances    4            1.2698 %
Kappa statistic                    0.967
Mean absolute error                 0.0145
Root mean squared error            0.0985
Relative absolute error            3.7544 %
Root relative squared error        22.4475 %
Total Number of Instances          315

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.991   0.024   0.991     0.991   0.991     0.999   Occupied
          0.976   0.009   0.976     0.976   0.976     0.999   Vacant
Weighted Avg.  0.987   0.02     0.987     0.987   0.987     0.999

=== Confusion Matrix ===

  a  b  <-- classified as
231  2  |  a = Occupied
  2  80 |  b = Vacant

```

Figure 5.63: Output of Weka naive Bayes classifier for three main features, and dynamic background value (with special cases)

We can see that compared with single feature detection, the recognition rate increases to 98.73% and the ROC area is 0.999, nearly ideal. Although some features may not perform well when applied alone, they still can be used to increase the recognition accuracy when applied with other features together.

To increase the recognition accuracy, if response time is not considered, adding more significant features is one choice.

5.3 Extension to Parking Images under Different Conditions

In this section, the experimental results of three independent images under different conditions are presented. Firstly, an entire process of an image from a wide-angle camera is presented. It shows that using the method described in this thesis, the region of detection can be enlarged. Secondly, one sample is taken from an outdoor parking lot is presented. There are some holes at the wall behind the parking region. It shows that using the method described in this thesis, the outdoor parking with some interference objects can also achieve a good detection result. Thirdly, a parking image containing a car beneath single color car cover is given to show that there are special conditions which may result in detection errors using one feature, but still can achieve the right result using multiple features.

The threshold (generated from Otsu method described in section 4.1 and 4.1) for images from section 5.3.1 to section 5.3.3 is listed below:

The number of edge pixels= 360;

Region Number=30;

FMPV=45;

Edge Pixel Percentage of Single Parking Region=1.6%

5.3.1 Processing of a Parking Image Taken by a Wide Angle Camera

This section shows the experimental result of an indoor parking lot (Building 64, the student parking, KFUPM). The parking image came from a wide angle camera which can cover 7 spaces. The original size of the raw image is 5184*3456 (pixels), taken on May 31 2013 (file name: IMG_2442), shown in Figure 5.64:



Figure 5.64: A raw parking image taken by wide angle camera

To reduce the processing time and match the matrix size limit of MATLAB. The raw image is cut into a small size by removing the ceiling region and the opposite parking region. The road region is remained to improve the recognition accuracy using foreground/background method as the ground share the similar gray pixel value. The input image is shown in Figure 5.65:



Figure 5.65: The input image from a wide angle camera

To deal with a single image from a different parking lot, static parking space region selection model applied to the images from section 5.1.1 to section 5.1.3 is no longer suitable. Therefore, dynamic polygon regions selection method is applied in this case.

Figure 5.66 shows the selected quadrangle parking spaces from No.1 to No.7.

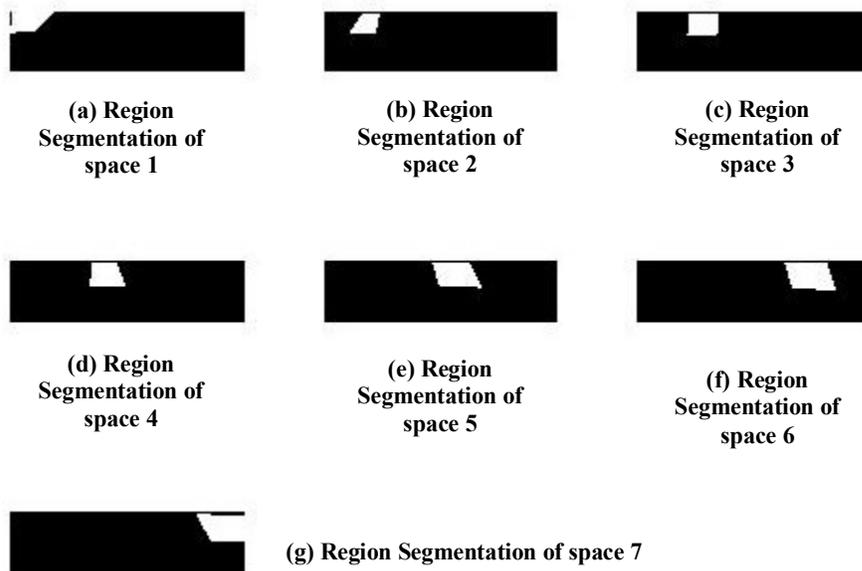


Figure 5.66: Results of the dynamic parking space selection of the input image from a wide angle camera

Figures 5.67 to 5.73 show result of the edge pixel counting method of each parking space.

Based on the same image, we also can get the object number.

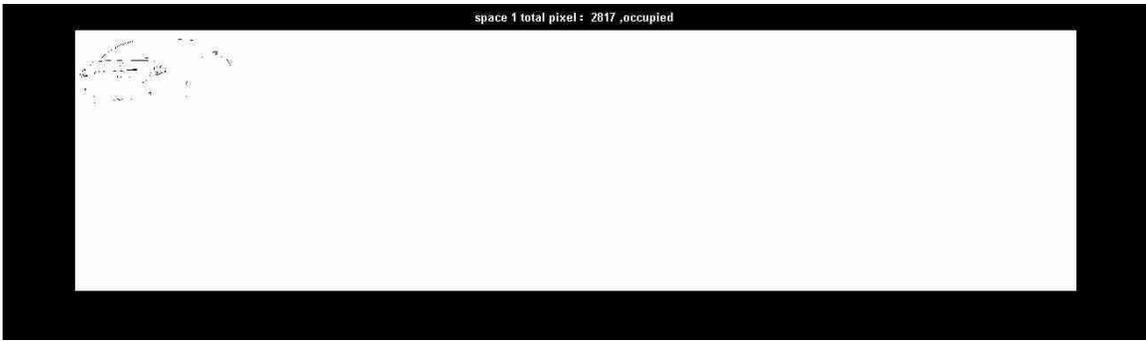


Figure 5.67: Edge pixel counting for space 1 of the input image from a wide angle camera, containing 2817 edge pixels, detected as an occupied space

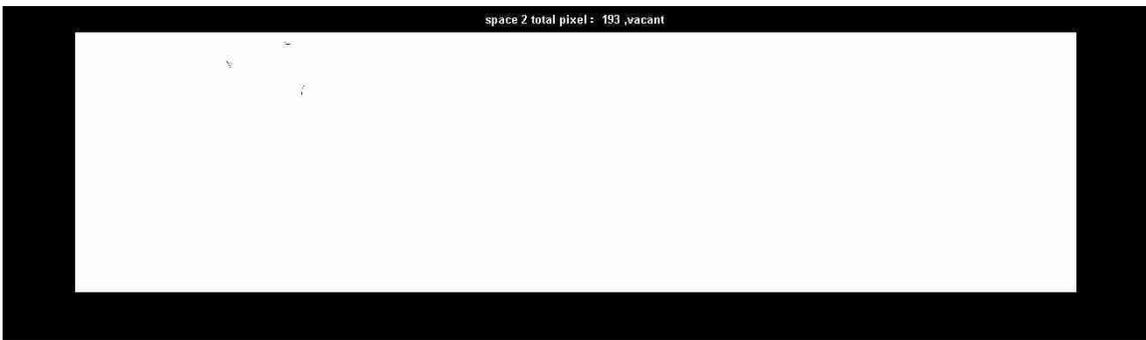


Figure 5.68: Edge pixel counting for space 2 of the input image from a wide angle camera, containing 193 edge pixels, detected as a vacant space

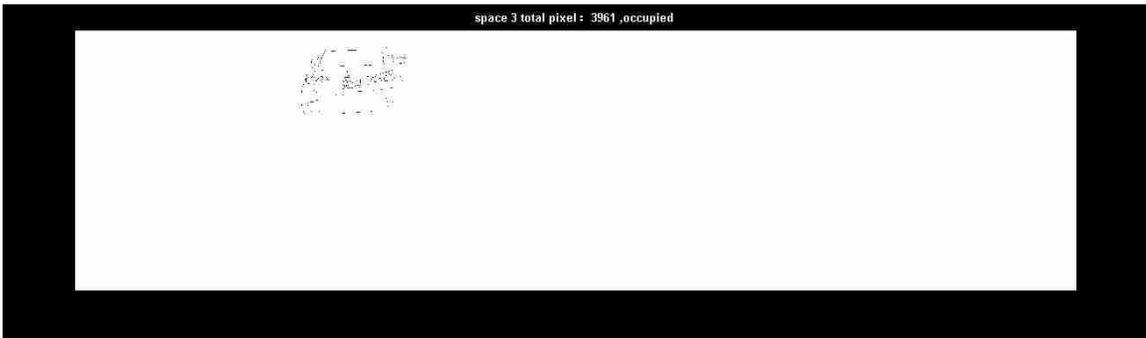


Figure 5.69: Edge pixel counting for space 3 of the input image from a wide angle camera, containing 3961 edge pixels, detected as an occupied space

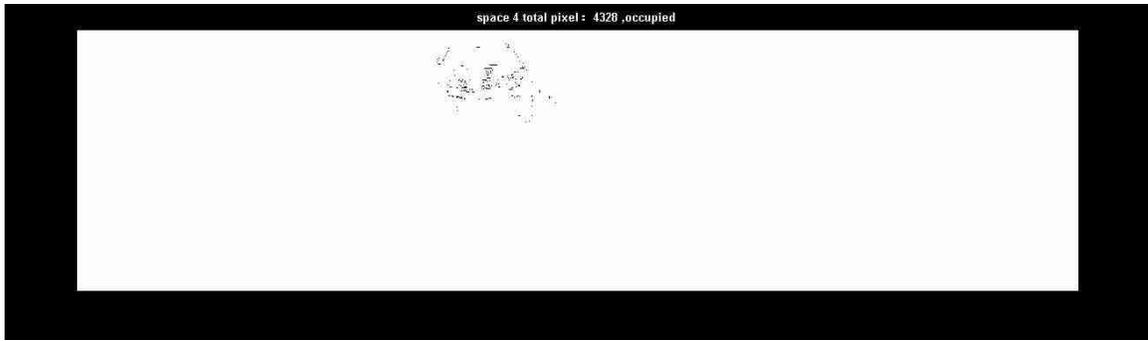


Figure 5.70: Edge pixel counting for space 4 of the input image from a wide angle camera, containing 4328 edge pixels, detected as an occupied space

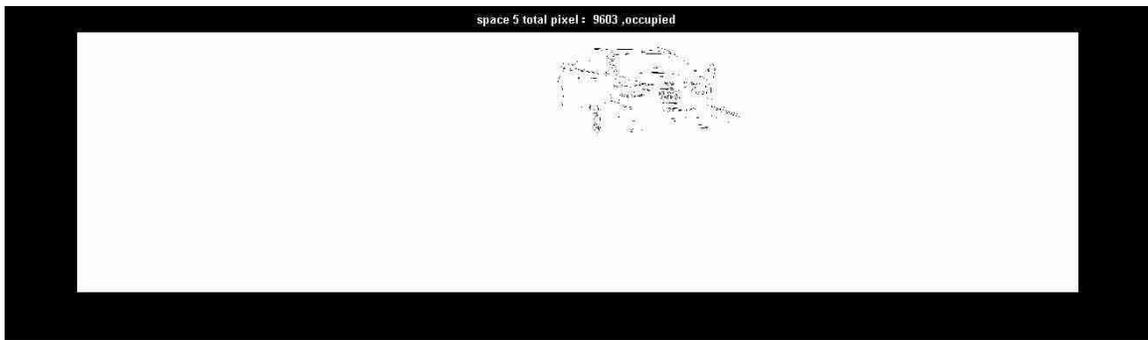


Figure 5.71: Edge pixel counting for space 5 of the input image from a wide angle camera, containing 9603 edge pixels, detected as an occupied space

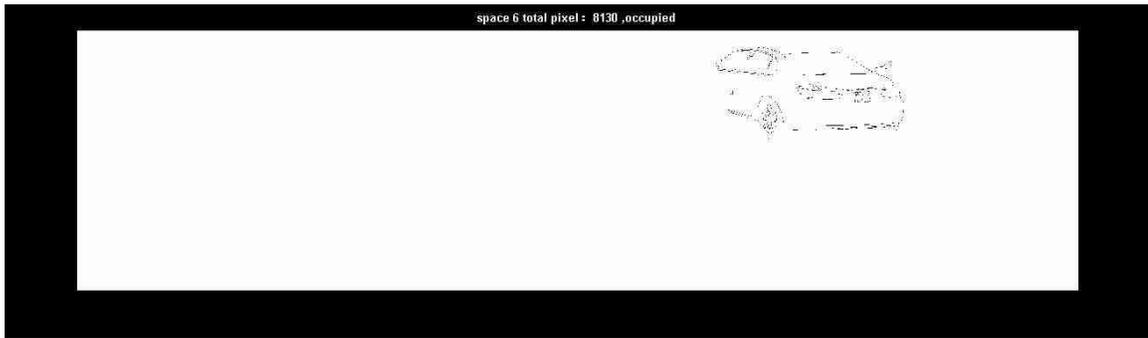


Figure 5.72: Edge pixel counting for space 6 of the input image from a wide angle camera, containing 8130 edge pixels, detected as an occupied space

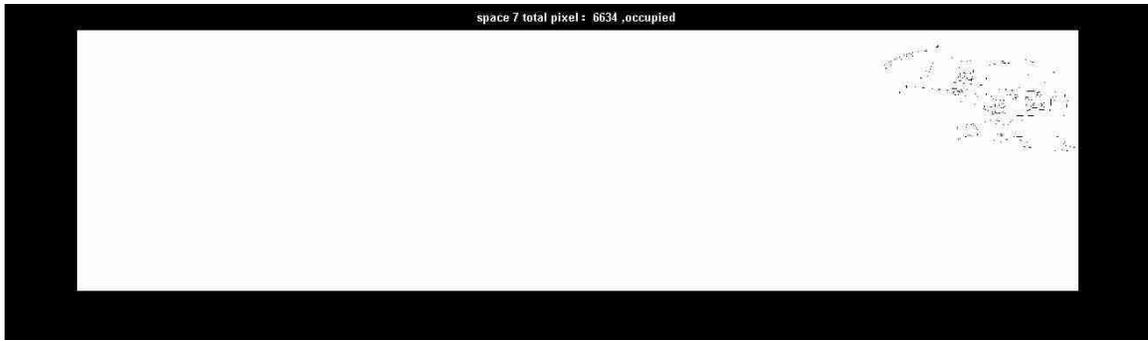


Figure 5.73: Edge pixel counting for space 7 of the input image from a wide angle camera, containing 6634 edge pixels, detected as an occupied space

For the foreground/background segmentation technique, we used the histogram shown in Figure 5.74 to identify the background pixels. Figure 5.75 shows the foreground information generated from image subtraction of both directions and the integrated value.

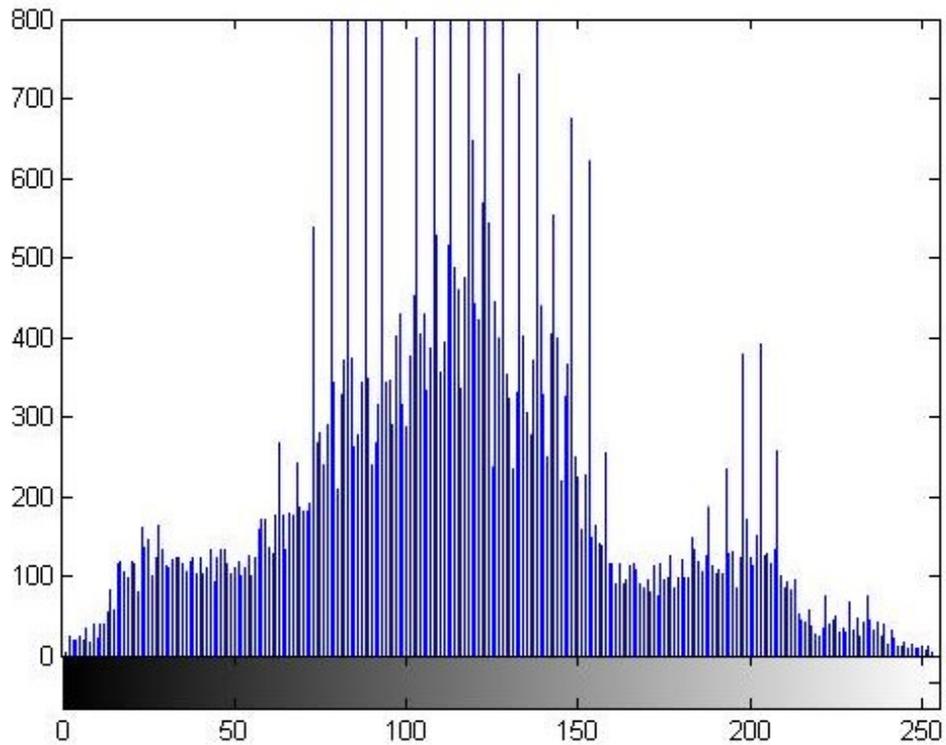


Figure 5.74: Histogram of the input image from a wide angle camera



(a) The dynamic background pixel value subtracts the gray-scale image



(b) The gray-scale subtracts the dynamic background pixel value image



(c) The integrated foreground image of (a) and (b)

Figure 5.75: The foreground generation of the input image from a wide angle camera

Stem figure of doubled FMPV is shown in Figure 5.76

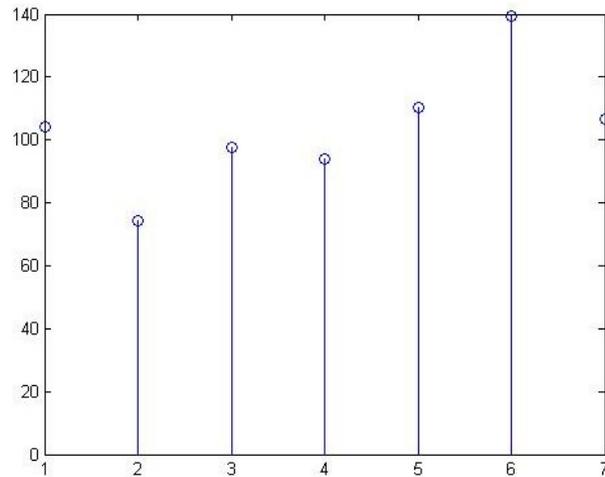


Figure 5.76: Stem figure of doubled FMPV of the input image from a wide angle camera

Table 5.7 shows the detection result of all the seven parking spaces.

Table 5.7: The seven spaces parking lot detection results of Figure 5.65

Space No	1	2	3	4	5	6	7
# edge pixels	2817	193	3961	4328	9603	8130	6634
Decision 1	O	V	O	O	O	O	O
#objects	178	9	263	315	516	545	553
Decision 2	O	V	O	O	O	O	O
FMPV	52	42	49	47	55	70	53
Decision 3	O	V	O	O	O	O	O

where V stands for vacant, O stands for occupied. We can see that when a parking image is from a wide angle camera, the camera's view is much larger than the view of a mobile camera. Also, the wide angle camera used in this section has a high resolution, which is 5184 by 3456 (pixels). High resolution increases the difference between vacant spaces and occupied spaces when considering the feature of the number of edge pixels. Therefore, when the image-based vacant parking space detection system is extended to a database mainly consist of high resolution images from the wide angle camera, edge pixel counting method will be considered as the main category.

5.3.2 Processing of an Outdoor Parking Image

In this section, we deal with an image from an outdoor parking.

We test the three methods on an outdoor parking image with three vacant spaces as shown in Figure 5.77, the edge detection displayed Figure 5.78.



Figure 5.77: Input image (outdoor) with one occupied spots and three vacant spaces

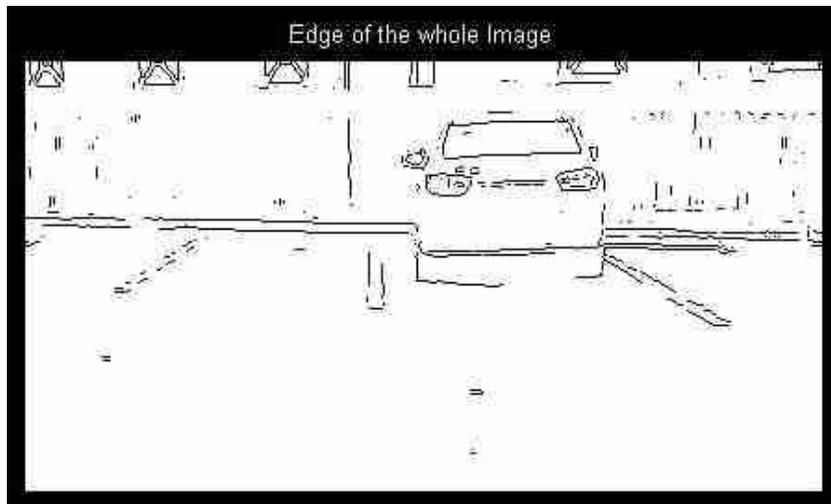


Figure 5.78: Edge detection of an outdoor parking with one occupied spots and three vacant spaces

For the foreground/background segmentation technique, we used the histogram as shown in Figure 5.79 to identify the background pixels. Figure 5.80 shows the foreground information generated from image subtraction of both directions and the integrated value.

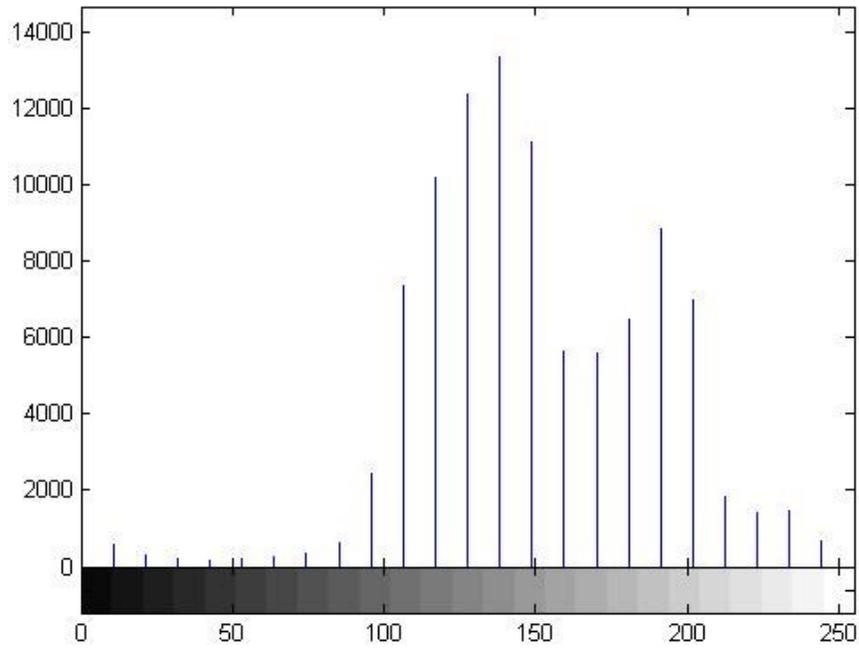


Figure 5.79: Histogram using 25 bins of an outdoor parking with one occupied spots and three vacant spaces



Figure 5.80: The foreground information of an outdoor parking with one occupied spots and three vacant spaces

The stem figure of doubled FMPV is shown in Figure 5.81

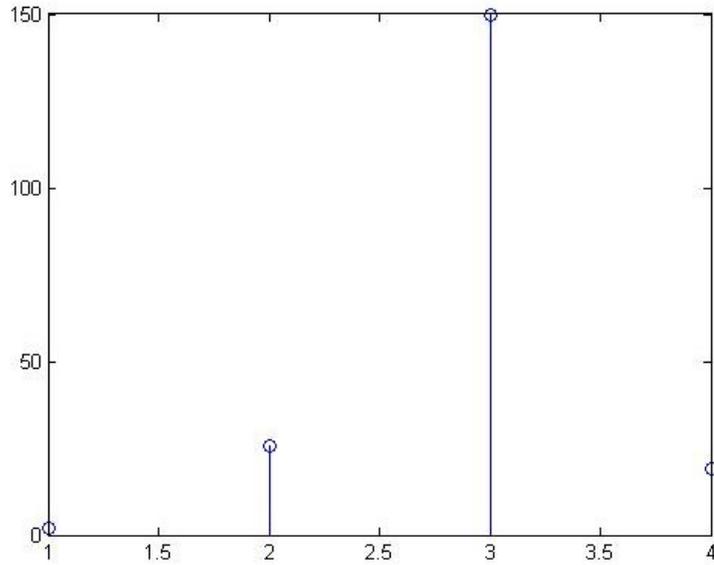


Figure 5.81: Stem figure of double FMPV of an outdoor parking with one occupied spots and three vacant spaces

The results of the three methods are shown in Table 5.8

Table 5.8: Edge pixel counting, object counting, and F/B method of Figure 5.77

	Space 1	Space 2	Space 3	Space 4
No. of edge pixels	39	9	973	0
Decision 1	vacant	vacant	occupied	vacant
No. of objects	3	2	65	0
Decision 2	vacant	vacant	occupied	vacant
FMPV	2	26	149	19
Decision 3	vacant	vacant	occupied	vacant

When dealing with an outdoor parking image, the interference objects, such as trees, windows, doors and telegraph poles may influence the edge pixel counting and object counting. But when static background pixel value is applied, such as using the average pixel value of the road, FMPV achieves an accuracy of more than 90% for 20 parking spaces.

5.3.3 Processing of the Parking Image Containing a Covered Car

In this section, we deal with a special case parking image. It contains a car totally beneath a car cover. The covered car has less region numbers than normal cars. If there are wrinkles at the car cover, like the special case described in this section, the contour feature will be influenced. The small contours such as the license plate and tail lights will all be hidden beneath the car cover and the contour feature becomes a large object. Therefore, the number of edge pixels will be reduced to a small number close to or even less than the number of edge pixels from a vacant parking space. Moreover, if there are no wrinkles at the car cover, even the object number will be reduced to a very small value, meaning that the detection using boundary-based texture segmentation will also provide an error. The input image is shown as Figure 5.82.



Figure 5.82: Input image containing a special case (a five-spaces parking from building 64)

Using the same methods we get the edge information in Figure 5.83 and foreground information in Figure 5.84

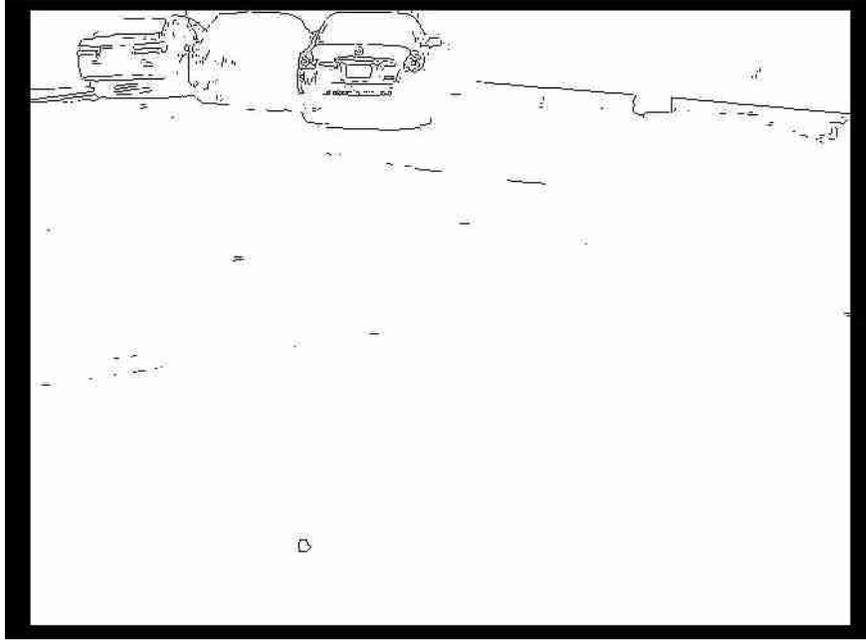


Figure 5.83: Edge information of a parking containing a special case (a five-spaces parking from building 64)



Figure 5.84: Foreground information of a parking containing a special case (a five-spaces parking from building 64)

And the summary of the testing image is show in Table 5.9

Table 5.9: Information Summary of Figure 5.82

Image Information	Image Date	2013-01-19				
	Place	Indoor				
	Final Decision	Occupied	Occupied	Occupied	Vacant	Vacant
	Spaces	1	2	3	4	5
Original Image	x	2592				
	y	1936				
	Total Pixels	5018112				
Edge Based	X Region	80	80	80	80	80
	Y Region	90	90	120	130	130
	Region Total	7200	7200	9600	10400	10400
	Edge Pixels	721(O)	341(VE)	1081(O)	156(V)	198(V)
	% of whole	0.234	0.111	0.351	0.050	0.064
	% of Region	10.0(O)	4.73(O)	11.2(O)	1.50(V)	1.9(OE)
	Objects	50(O)	45(O)	80(O)	8(V)	15(V)
Foreground/Background Method	BG value	191.25				
	FMPV	61(O)	91(O)	83(O)	60(OE)	45(V)

The above table shows the detection result, where V stands for vacant, O stands for occupied, VE means it is occupied but detected as vacant and OE means it is vacant but detect as occupied. If we use only one feature, it is very clear that the detection use the number of edge pixels in the table above will lead to an error. However, using all the features together can avoid such errors.

5.4 Summary

In conclusion, our image-based parking detection method using hybrid features in this project is able to deal with the parking region under different conditions. It is a flexible, efficient and robust system. The hybrid features could be different combinations of different classes. If we want better accuracy, more features will be taken into consideration and the processing time will increase. In other words, given a certain requirement, the system is able to response in a very short time by using minimum number of features. With regard to the whole B2 floor of building 59, KFUPM, there are 63 parking spaces. If we want to avoid the interference of the pillars, 21 normal cameras are needed. Each camera covers 3 parking spaces in average. If it does not matter whether the pillars located at the center of the image or not, 9 wide angle cameras are enough to cover the whole B2 floor.

CHAPTER 6

Conclusions and Future Work

This thesis presents an image-based vacant parking detection system using a Bayesian classifier together with a set of robust hybrid features. The features selected for our application cover three main categories: Edge information, object information and foreground information. An overall recognition accuracy of more than 90% was achieved across different scenarios. We show that when the features are used independently, they may achieve good performance under some conditions but fail under others. In order to increase recognition rate, we combine the three types of features into a single feature vector then use a Bayesian framework to achieve an accuracy of more than 98% (even 100% if there are no special cases).

The proposed framework using hybrid features is shown to overcome some of the disadvantages of single features. The edge pixel counting technique, for example, is very simple to implement but may lead to wrong decisions when the acquired image exhibits a high noise or too much dust covering the parking region. The second technique which is based on counting the number of closed objects is robust however finding closed contours in an image is not a simple task and may need to be preceded with some pre-processing such as morphological dilation.

In addition to the high recognition accuracy achieved, the proposed vacant parking detection system is also cost-effective and scalable. Wide angle cameras, for example,

can be used to cover up to 8-10 parking spots reducing the total number of cameras required to cover large areas.

In further enhance the performance of the proposed system; we will be working on a number of directions:

- A communication module can be added to the system so that the status of the parking lot is displayed at the entrance of the parking area or even more importantly inform the driver on where the vacant spots are located within the parking area.
- Develop an optimal set of features that could accommodate different operating conditions including outdoors, indoors, different lighting conditions, etc.
- Develop a system for the analysis of images covering a large number of parking spots using superresolution techniques from video sequences.
- Combine sensor based and image based approaches into robust detection systems.
- Implement the proposed system on a real time FPGA platform.

References

- [1] W. Lixia and J. Dalin, "A Method of Parking Space Detection Based on Image Segmentation and LBP," in *2012 Fourth International Conference on Multimedia Information Networking and Security (MINES)*, Nov., pp. 229–232.
- [2] S. Banerjee, P. Choudekar, and M. K. Muju, "Real Time Car Parking System Using Image Processing," in *2011 3rd International Conference on Electronics Computer Technology (ICECT)*, 2011, vol. 2, pp. 99–103.
- [3] D. B. L. Bong, K. C. Ting, and K. C. Lai, "Integrated Approach in the Design of Car Park Occupancy Information System (COINS)," *Int. J. Comput. Sci.*, vol. 35, no. 1, 2008.
- [4] S.-F. Lin, Y.-Y. Chen, and S.-C. Liu, "A Vision-Based Parking Lot Management System," in *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06*, 2006, vol. 4, pp. 2897–2902.
- [5] C.-C. Huang and S.-J. Wang, "A Hierarchical Bayesian Generation Framework for Vacant Parking Space Detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1770–1785, Dec. 2010.
- [6] L.-C. Chen, J.-W. Hsieh, W.-R. Lai, C.-X. Wu, and S.-Y. Chen, "Vision-Based Vehicle Surveillance and Parking Lot Management Using Multiple Cameras," in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Oct., pp. 631–634.
- [7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River, N.J.: Prentice Hall, 2002.
- [8] H. Ichihashi, A. Notsu, K. Honda, T. Katada, and M. Fujiyoshi, "Vacant Parking Space Detector for Outdoor Parking Lot by Using Surveillance Camera and FCM Classifier," in *IEEE International Conference on Fuzzy Systems, 2009. FUZZ-IEEE 2009*, 2009, pp. 127–134.
- [9] Z. Bin, J. Dalin, W. Fang, and W. Tingting, "A Design of Parking Space Detector Based on Video Image," in *9th International Conference on Electronic Measurement Instruments, 2009. ICEMI '09*, 2009, pp. 2–253–2–256.
- [10] M. Chen, C. Hu, and T. Chang, "The Research on Optimal Parking Space Choice Model in Parking Lots," in *2011 3rd International Conference on Computer Research and Development (ICCRD)*, 2011, vol. 2, pp. 93–97.
- [11] S. Yu, B. Li, Q. Zhang, and M. Q.-H. Meng, "An Optimal Parking Space Search Model Based on Fuzzy Multiple Attribute Decision Making," in *2012 10th World Congress on Intelligent Control and Automation (WCICA)*, 2012, pp. 4350–4355.
- [12] K. Yamada and M. Mizuno, "A Vehicle Parking Detection Method Using Image Segmentation," *Electron. Commun. Jpn. Part III Fundam. Electron. Sci.*, vol. 84, no. 10, pp. 25–34, Oct. 2001.
- [13] N. True, "Vacant Parking Space Detection in Static Images."
- [14] T. Fabian, "An Algorithm for Parking Lot Occupation Detection," in *Computer Information Systems and Industrial Management Applications, 2008. CISIM '08. 7th*, June, pp. 165–170.

- [15] R. Yusnita, F. Norbaya, and N. Basharuddin, "Intelligent Parking Space Detection System Based on Image Processing," *Int. J. Innov. Manag. Technol.*, vol. 3, no. 3, pp. 232–235, Jun. 2012.
- [16] L. G. Shapiro and G. C. Stockman, *Computer vision*. Prentice Hall, 2001.
- [17] P. Brodatz, *Textures: a photographic album for artists and designers*. Dover Publications, 1966.
- [18] A. Materka, M. Strzelecki, T. Analysis, M. A. Review, A. Materka, and M. Strzelecki, "Texture analysis methods - a review," Institute of Electronics, Technical University of Lodz, 1998.
- [19] E. S. Gadelmawla, "A Vision System for Surface Roughness Characterization Using the Gray Level Co-occurrence Matrix," *NDT E Int.*, vol. 37, no. 7, pp. 577–588, Dec. 2004.
- [20] S. Y. Lu and K. S. Fu, "A Syntactic Approach to Texture Analysis," *Comput. Graph. Image Process.*, vol. 7, no. 3, pp. 303–330, 1978.
- [21] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *J. Neurophysiol.*, vol. 58, no. 6, pp. 1233–1258, Dec. 1987.
- [22] J. G. Daugman, "Complete discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 7, pp. 1169–1179, 1988.
- [23] A. K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation using Gabor Filters," in *IEEE International Conference on Systems, Man and Cybernetics, 1990. Conference Proceedings*, Nov, pp. 14–19.
- [24] S. Qian and D. Chen, "Discrete Gabor Transform," *IEEE Trans. Signal Process.*, vol. 41, no. 7, pp. 2429–2438, 1993.
- [25] F. Riaz, A. Hassan, S. Rehman, and U. Qamar, "Texture Classification Using Rotation- and Scale-Invariant Gabor Texture Features," *IEEE Signal Process. Lett.*, vol. 20, no. 6, pp. 607–610, Jun. 2013.
- [26] Z. Li and C. Liu, "Gray Level Difference-based Transition Region Extraction and Thresholding," *Comput. Electr. Eng.*, vol. 35, no. 5, pp. 696–704, Sep. 2009.
- [27] E. J. Hannan, *Multiple Time Series*. New York [etc.: John Wiley and Sons, 1970.
- [28] O. Marques, "Morphological Image Processing," in *Practical Image and Video Processing Using MATLAB®*, John Wiley & Sons, Inc., 2011, pp. 299–334.
- [29] Y. Xu, J. B. Weaver, J. Healy, D.M., and J. Lu, "Wavelet Transform Domain Filters: a Spatially Selective Noise Filtration Technique," *IEEE Trans. Image Process.*, vol. 3, no. 6, pp. 747–758, 1994.
- [30] M. Stéphane, Ed., "A Wavelet Tour of Signal Processing (Third Edition)," in *A Wavelet Tour of Signal Processing (Third Edition)*, Boston: Academic Press, 2009, p. iv.
- [31] S. Arivazhagan and L. Ganesan, "Texture Classification Using Wavelet Transform," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1513–1521, 2003.
- [32] J. Mao and A. K. Jain, "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models," *Pattern Recognit.*, vol. 25, no. 2, pp. 173–188, 1992.
- [33] K. I. Laws, "Textured Image Segmentation," Ph.D. Dissertation, University of Southern California, 1980.

- [34] D. J. Heeger and J. R. Bergen, "Pyramid-based Texture Analysis/Synthesis," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1995, pp. 229–238.
- [35] W. Yuqing, "Image Quality Assessment Based on Gradient Complex Matrix," in *2012 International Conference on Systems and Informatics (ICSAI)*, 2012, pp. 1932–1935.
- [36] T. Chang and C.-C. J. Kuo, "Texture Analysis and Classification with Tree-structured Wavelet Transform," *IEEE Trans. Image Process.*, vol. 2, no. 4, pp. 429–441, 1993.
- [37] G. Samorodnitsky, "Long Range Dependence," *Found Trends Stoch Sys*, vol. 1, no. 3, pp. 163–257, 2007.
- [38] A. C. H. Yu and L. Lovstakken, "Eigen-based Clutter Filter Design for Ultrasound Color Flow Imaging: a Review," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 57, no. 5, pp. 1096–1111, 2010.
- [39] J.-L. Starck, E. J. Candes, and D. L. Donoho, "The Curvelet Transform for Image Denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.
- [40] C. Høiland, "The Radon Transform," Aalborg University, Nov. 2007.
- [41] F. P. Miranda, L. E. N. Fonseca, and J. R. Carr, "Semivariogram Textural Classification of JERS-1 (Fuyo-1) SAR Data Obtained Over a Flooded Area of the Amazon Rainforest," *Int. J. Remote Sens.*, vol. 19, no. 3, pp. 549–556, 1998.
- [42] N. Ahuja and A. Rosenfeld, "Mosaic Models for Textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 1, pp. 1–11, Jan. 1981.
- [43] J. D. Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80.*, 1980, vol. 5, pp. 291–294.
- [44] L. Wang and D.-C. He, "Texture Classification Using Texture Spectrum," *Pattern Recognit.*, vol. 23, no. 8, pp. 905–910, 1990.
- [45] F. Zhou, J.-F. Feng, and Q.-Y. Shi, "Texture Feature Based on Local Fourier Transform," in *2001 International Conference on Image Processing, 2001. Proceedings*, 2001, vol. 2, pp. 610–613 vol.2.
- [46] A. Al-Janobi, "Performance Evaluation of Cross-diagonal Texture Matrix Method of Texture Analysis," *Pattern Recognit.*, vol. 34, no. 1, pp. 171–180, 2001.
- [47] T. Randen and J. H. Husoy, "Filtering for Texture Classification: a Comparative Study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 291–310, Apr. 1999.
- [48] M. J. Nassiri, A. Vafaei, and A. Monadjemi, "Texture Feature Extraction using Slant-Hadamard Transform," presented at the World Academy of Science, Engineering and Technology, 2008, pp. 1055–1059.
- [49] H. Derin and W. S. Cole, "Segmentation of Textured Images Using Gibbs Random Fields," *Comput. Vis. Graph. Image Process.*, vol. 35, no. 1, pp. 72–98, Jul. 1986.
- [50] M. Mirmehdi, X. Xie, and J. S. Suri, *Handbook of texture analysis*. Imperial College Press, 2008.
- [51] R. M. Haralick, "Statistical and Structural Approaches to Texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May.
- [52] M. Amadasun and R. King, "Textural Features Corresponding To Textural Properties," *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 5, pp. 1264–1274, 1989.

- [53] H. Tamura, S. Mori, and T. Yamawaki, "Textural Features Corresponding to Visual Perception," *IEEE Trans. Syst. Man Cybern.*, vol. 8, no. 6, pp. 460–473, 1978.
- [54] C. chi-H. Chen, L. L. F. Pau, P. S. P. Wang, and P. S. P. Wang, *Handbook of Pattern Recognition and Computer Vision*. World Scientific Publishing Company, Incorporated, 1999.
- [55] M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [56] Z. Huang and J. Leng, "Analysis of Hu's Moment Invariants on Image Scaling and Rotation," in *2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, April, vol. 7, pp. V7–476–V7–480.
- [57] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
- [58] E. Miyamoto and T. Merryman, *Fast Calculation of Haralick Texture Features*. .
- [59] R. C. Gonzalez, S. L. Eddins, and R. E. Woods, *Digital Image Processing Using Matlab*. Prentice-Hall, 2004.
- [60] J. C. Russ, *The image processing handbook*, 4th ed. Boca Raton, Fla.: CRC Press, 2002.
- [61] U. Qidwai and C. H. Chen, *Digital image processing*. Chapman & Hall/CRC, 2010.
- [62] M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," *J. Electron. Imaging*, vol. 13, no. 1, pp. 146–168, 2004.
- [63] "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.
- [64] L. Devroye, *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.
- [65] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss," *Mach Learn*, vol. 29, no. 2–3, pp. 103–130, Nov. 1997.
- [66] "Weka---Machine Learning Software in Java," *SourceForge*. [Online]. Available: <http://sourceforge.net/projects/weka/>. [Accessed: 29-Mar-2014].

Vitae

Name : Junzhao Liu

Nationality : Chinese

Date of Birth : 1/22/1989

Email : g201101950@kfupm.edu.sa; liujunzhao1989@gmail.com

Present Address : KFUPM, Building 801, Room 211

Permanent Address : Building 20, Gate 3, Room 202,
Wuhan Conservatory of Music, Wuhan, China, 430060

Academic Background:

Bachelor of Engineering, Electronics and Information Engineering, Huazhong University of Science and Technology (HUST), Wuhan, China

Master of Science, Electrical Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia

Publications:

1. **J. Liu**, M. Mohandes, and M. Deriche, "A Multi-Classifer Image Based Vacant Parking Detection System," in 2013 20th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2013, pp. 933–936.

2. M. Mohandes, **J. Liu**, and M. Deriche, “A Survey of Image-based Arabic Sign Language Recognition,” in 2014 11th International Multi-Conference on Systems, Signals Devices (SSD), 2014, pp. 1–4.

3. M. Mohandes, M. Deriche, and **J. Liu**, “Image-based and Sensor-based Approaches to Arabic Sign Language Recognition”, Accepted by IEEE Transactions on Human-Machine Systems, 2014.