國立交通大學

資訊科學與工程研究所

博士論文

基於串聯式剔除機制來減少視訊中時空搜尋空間的即時車牌辨識

Real-time License Plate Recognition based on Cascaded Rejection Mechanisms to Reduce Spatio-temporal Search Space in Video Sequences

研究生：王舜正

指導教授：李錫堅　教授

中華民國九十七年六月

基於串聯式剔除機制來減少視訊中時空搜尋空間的即時車牌辨識

Real-time License Plate Recognition based on Cascaded Rejection Mechanisms to Reduce Spatio-temporal Search Space in Video Sequences

研究生：王舜正 Student：Shen-Zheng Wang

指導教授：李錫堅 教授 Advisor：Prof. Hsi-Jian Lee

國立交通大學
資訊科學與工程研究所
博士論文

A Dissertation

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

基於串聯式剔除機制來減少視訊中時空搜尋空間的即時車牌辨識

學生： 王舜正　　　　　　　　　　　　　指導教授： 李錫堅 教授

國立交通大學資訊科學與工程研究所

摘 要

在監控應用中，減少搜索空間(SSR)是發展高效率演算法的一項重要關鍵。 在接受視訊的車牌辨識系統研究中，我們整合了空間上和時間上的減少搜尋空間技術。 然而由於越來越多特徵需量測，計算量可能會顯著地增加。 當考量到大多數輸入樣式都是非目標樣本(不是要認出的物件)的這個事實，似乎盡快剔除大量的非目標樣本是相當有效益的。 因此我們提出了一個基於串聯式剔除架構來減少視訊中時空搜尋空間，同時確保系統效能的即時車牌辨識。 甚至為了在複雜的環境下正確地擷取出車牌，我們首先提出兩個物件表示法:簡潔車牌區域與重複區域。 簡潔車牌區域定義為車牌字元上下界內所包含的區域，此區域可以在第一個步驟中擷取出來，以避免額外的字元上下界偵測程序。 我們提出的方法就是由減少空間上的搜尋空間開始，其中包含了三個模組：單次掃描的簡潔車牌區域偵測、雙階層的車牌字元切割和適應性的機器學習。 經由可便利計算的特徵，如垂直梯度值和擴充 Haar-like 特徵值，這些模組擷取出簡潔車牌或字元的候選區域並驗證之。 再者，我們提出要剔除在視訊中重複出現的相同外觀區域，這些區域通常包含有停止的車輛或是固定的背景，且應可剔除以避免被重複分類。 為了效能考量，重複樣式只會在車牌候選區域內偵測，在此稱之為時空關係上的減少搜尋空間。 且該重複樣式比對是基於區塊為基礎的機制來設計，透過計算正切距離來克服位置、大小、旋轉或是亮度的變異。 在我們的實驗中，經由時空關係上的減少搜尋空間，可減少 87.9% 的搜尋空間；該車牌辨識系統在 Intel P-IV 3-GHz 的個人電腦上，每秒可以處理 38 張 640x480 解析度的影像。

Real-time License Plate Recognition based on Cascaded Rejection Mechanisms to Reduce Spatio-temporal Search Space in Video Sequences

Student: Shen-Zheng Wang Advisor: Prof. Hsi-Jian Lee

Submitted to Institute of Computer Science and Engineering
College of Computer Science
Chiao Tung University

ABSTRACT

In surveillance applications, search space reduction (SSR) is an essential element to efficient algorithms. In this study, spatial and temporal SSRs are integrated for license plate recognition (LPR) in video sequences. However, as more features are measured, the computational load may increase significantly. When regard to the fact that most input patterns are negatives, it is apparently efficient to reject a majority of negatives as soon as possible. Therefore, we propose a realtime LPR based on a cascaded rejection framework to reduce spatiotemporal search space rapidly, while ensuring that the performance is high. To extract plates accurately even in complicated situations, two representations, compact plate regions and repeated regions, are first presented. Compact plate regions, which bound the top and bottom of plate characters, could be extracted in the first stage to avoid the use of additional removal procedures. Our method started from spatial SSR by algorithms of one-pass compact plate extraction, bi-level plate character segmentation, and adaptive machine learning. Region candidates of compact plates or plate characters are extracted and verified by these algorithms performed on effectively calculated features, such as vertical gradients and extended Haar-like features. Moreover, we proposed to exclude repeated patterns with the similar appearances in the same location of consecutive frames, which usually include stopped vehicles or regular backgrounds and could be excluded from repeated classification. For efficiency, repeated patterns were detected only on the plate candidates, named spatiotemporal SSR, based on a block-based mechanism by estimating the tangent distance, which is invariant to the variations in positions, sizes, rotations, or brightness. In our experiments, the search space could be reduced up to 87.9% by the spatiotemporal SSR; the LPR system can recognize plates over 38 frames per second with a resolution of 640 × 480 pixels on a 3-GHz Intel P-IV PC.

Acknowledgements

Writing this dissertation is the most difficult, yet the most gratifying, work that I have ever done. The accomplishment of this work, however, could not have been possible without many people’ s constant support and help. I would like to express my sincere thanks to all the people who have played significant roles during my Ph.D. studies.

First, I am deeply grateful to my advisor Prof. Hsi-Jian Lee for the past years, for his great guidance, infinite patience, and strong faith in me. I am greatly indebted to him for many invaluable contributions he made to the development of this study. On the personal side, I truly appreciate the full confidence he had in my ability and the continuing care and encouragement he gave me at difficult times. He has always known how to help me to go through all the hurdles in this long journey and motivate me to do my best work. It is indeed a great blessing that I have had Prof. Hsi-Jian Lee as my mentor.

Next, I would like to express my special gratitude to the two committee members, Prof. Wen-Hsiang Tsai and Prof. Jen-Hui Chuang. They have been generous with their time giving me detailed comments and eminently helpful suggestions on this study.

Many thanks are also due to my research participants. It has been a wonderful experience working closely on, for, and with them, and I truly appreciate their unreserved sharing and trust in me. Finally, I owe a great deal to my parents and my gril friend, Trista, for always encouraging me to achieve my goals and comforting me during my emotional storms even though we live apart. With deep love and gratitude, I dedicate this dissertation to them.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Motivations

Information automatically collected for smart visual surveillance applications, such as portal controlling, traffic monitoring, or vehicle detecting, has gained increasing importance in Intelligent Transportation Systems (ITS). With rapid development of vehicles and visual analysis technologies, automatic recognition of vehicles becomes more and more practical in many applications during the past two decades. In vehicles, license plates represent the unique identifications with regular patterns, such as a sequence of characters. The design of robust license plate recognition (LPR) systems becomes an important issue for recording, analyzing, and reporting surveillance targets. The inputs of LPR systems could be divided into two different types: still images and frame sequences. In the first type, a still image from a camera can be designed explicitly to process complex scenes such as multiple plates that exist in different locations of an image. In the second type, temporal information could be used to identify plates more robustly. Figure 1 shows examples of the two image types in surveillance applications. Figure 1(a) shows a still image captured by a camera controlled by the police, while Fig. 1(b) shows several frames from a capturing device installed in a car.

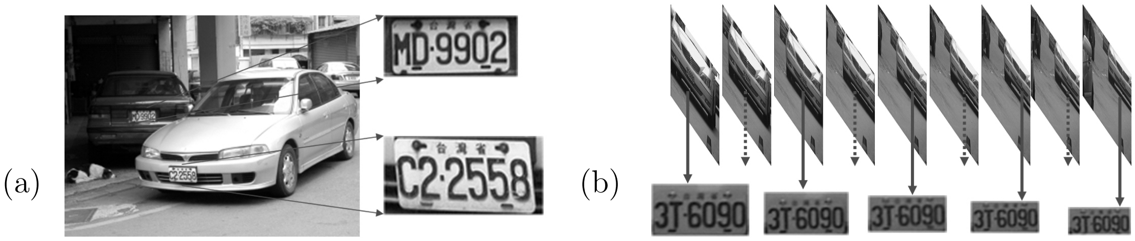


Figure 1: Two image categories in surveillance applications: (a) a still image from a camera, and (b) nine successive frames in a video stream.

1.2 The Considered Problems

To develop a LPR system for surveillance, three requirements must be satisfied. The first is the high accuracy performance; for instance, the accuracy rate measured as the weighted sum of the true positive rate (TPR) and the false positive rate (FPR) should be above 95%. The rate should be measured from a real surveillance environment with variations in plate types and environments. The second requirement is to process all frames from an input stream in real time because successive frames may contain temporal plate information useful in surveillance applications. Moreover, plates may not be recognized in certain frames due to unexpected events; for example, the plates may be covered by other passing vehicles or appear significantly blurred when they are out of focus. From the successively recognized results, certain strategies such as voting could be used to increase the accuracy rate. Currently, a common input stream may arrive at a standard frame rate of approximately 30 fps (frames per second) and a resolution of 640×480 pixels. Thus, a practical plate recognition system should function at over 30 fps, for example, the real time system in this study, with a high dimensional input (640×480 pixels). The third requirement is the adaptability. In real-world environments, there generally exist unfamiliar plate types or background. When new plate types or backgrounds arise, the plate recognition system should have the ability to recognize them correctly after learning the variation in plates and non-plates automatically.

In plate recognition applications, since input plates or environments generally have different variations, the techniques available are probably not very accurate, robust, or practical. For instance, plate sizes and locations may change significantly in different inputs because the target or the capturing device moves continuously; meanwhile, illumination would also be different. Moreover, few of the publishes consider motorcycles. The detection of license plates in the environment with motorcycles would be more difficult because the plates often connect directly with complex backgrounds. However, the detection of license plates on motorcycles is desired because motorcycles are important transportation tools in many countries. Figure 2 shows several license plates captured from different environments; some of the plates belong to motorcycles with white or green plates. Figure 3 shows another examples of variations that were tested in this study. The headlights of different vehicles at night are shown in Fig. 3(a), and examples of environmental lighting in different weather conditions are shown in Fig. 3(b). Plate-like background patterns on a tiled floor are shown in Fig. 3(c).

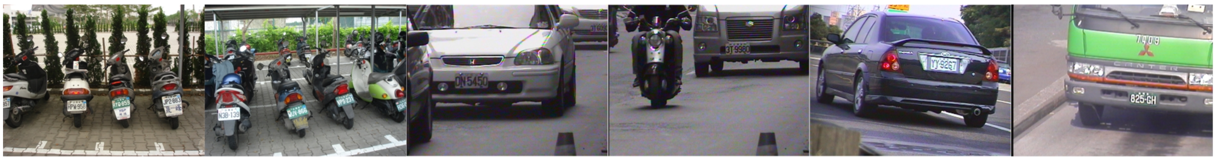


Figure 2: Examples of variations in license plates and their environments.

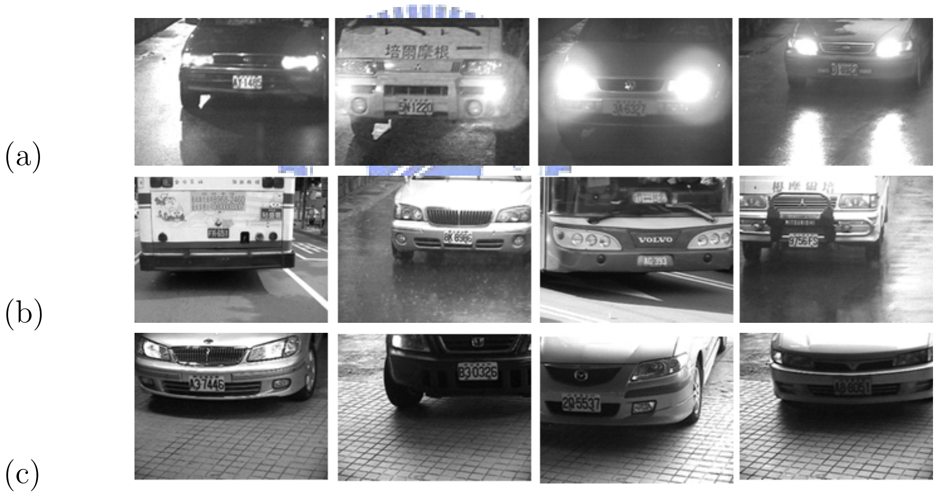


Figure 3: Examples of variations in plate types and environments: (a) different vehicle headlights, (b) different types of environment lighting, and (c) plate-like background.

Statistical mechanisms provide more robust and accurate representations. However, an important problem of the statistical type of features or techniques is that the computational load is high when a statistical mechanism is applied to all input candidates (more than one hundred thousand units). Instead of explicitly segmenting the characters in detected plates, Amit *et al.* [1] use a coarse-to-fine approach for both the detection and recognition of characters on license plates. Although they achieve high recognition rates, the statistical technique requires 3.5 seconds to process an input image, and it does not satisfy the requirement of surveillance applications. Another problem of the proposed algorithms, as detailed in Chapter 2, is that there is a tradeoff between multiscale ability and the computational effect. Most of them compute a multiscale image pyramid, which is complicated and time-consuming to detect targets of a particular size.

It is a challenging task to develop a LPR system that satisfies all the three requirements. This is because it is difficult to adjust the existing approaches to achieve the objective of a real-time system with a high TPR while maintaining a low FPR in various plate types or environments. The system may satisfy the minimum real-time criterion (for example, 15 fps) only for one module such as plate detection [2]. However, it is difficult to meet the real-time requirement in the entire LPR system. All variations under consideration for single images and video sequences are summarized as follows.

1.2.1 Difficulties for Sigle Images

- Plate variations
  - Location
    - Plates may exist in different locations of an input image.
  - Quantity
    - An input image may contain many or no plates.
  - Size
    - Plates with different sizes may exist in an image or different images.
  - Colors of plate characters and backgrounds
    - Plates may have various characters and background colors due to different plate types (taxis, private cars, etc.) or capturing devices.
  - Others
    - In addition to characters, a plate may contain adornments such as frames and screws.
- Environment variations
  - Illumination

    - ∗ Different types of illumination may occur in input images, mainly due to environmental lighting and vehicle headlights.
  - Plate-like background patterns
    - ∗ A background may contain patterns similar to plates, such as numbers stamped on a vehicle, bumpers with vertical patterns, and textured floors.

1.2.2 Difficulties for Video Sequences

In video-based surveillance applications, the input is usually fed from a video, which consists of a sequence of frames with surveillance targets and backgrounds. Utilizing temporal features over the sequence would speed up license plate recognition by avoiding processing unnecessary areas. This necessarily involves the use of motion models which describe and label the expected structure of the input sequence.

Motion segmentation has four conventional approaches: (1) background subtraction, (2) temporal difference, (3) optical flow, and (4) predictive model. Background subtraction is a simple and popular method for motion segmentation, especially under those situations with a relatively static background. Haritaoglu *et al.* [3], for example, used the background subtraction method for segmenting potential foreground objects in the real-time surveillance system W4. Temporal difference makes use of pixel differences between two or more consecutive frames to extract moving regions and is adaptive to dynamic environments. Lipton *et al.* [4], for example, obtained the absolute difference between the current and the previous frame and used a threshold function to determine changes. Optical flow could be used to detect moving objects even in the presence of camera motion. Schunck [5], for example, used characteristics of flow vectors of moving objects to detect moving regions in an image sequence. The comparisons of the three approaches are summarized in Hu *et al.* [6]. In predictive models, a set of features, such as points, edges, or contours, are tracked over a sequence of frames and some prediction mechanisms are used to minimize the potential candidates of these features. For example, Zayed *et al.* [7]

applied Kalman filters while Jia *et al.* [8] utilized mean-shift filters to track recognized vehicles.

The motion segmentation approaches may easily be deteriorated by the problems of changes in illumination, shadow, or colors between the surveillance targets and backgrounds. To deal with the problems, these approaches would become computationally expensive. Direct enhancement and modification on above approaches would increase the computational load.

1.3 Goal

As described above, currently, some important problems are still remained to be resolved. The following problems are focused in this dissertation:

1. Considerations for the development of real-time license plate recognition system
2. Machine learning for adaptive license plate recognition system
3. Search-space reduction(SSR) for improving the computational speed in single images
4. Search-space reduction(SSR) for improving the computational speed in video sequences

Certainly, it is a great challenge to overcome these problems. However, it is well known that the license plates include general characteristics ranging from low-level gradient features to high-level contextual meanings as follows:

1. The color of a plate character is always different from that of the background.
2. Plate characters are arranged in a sequence known as plate character lines.
3. A plate is mainly composed of plate characters.
4. Plate characters usually satisfy the plate specifications; for instance, the English alphabet could not be used in certain parts of the plate or characters have specific width-to-height ratios, depending on the location of the vehicle.

Thus, our major goal is to find out effective approaches to extract the license plates in single images and video sequences.

1.4 Summary of Achievements

To account for the lack of real-time consideration in the overall system, we propose a cascade framework based on rejecting mechanisms to develop a real-time statistical plate recognition system, which could deal with various problems effectively. We summarize some valuable results in our works as follows:

1. A cascade framework for a real-time plate recogniton system: We first propose two representations: spatial compact plate regions and temporal repeated regions. Compact plate regions, which bound the top and bottom of plate characters, could be extracted in the first stage to avoid the use of additional removal procedures. Repeated regions have similar appearances in the same location of consecutive frames. The repeated regions could be detected and excluded from repeated classification to save computation load. Moreover, under the assumption that most of the input candidates are negative, a cascade framework is presented based on rejection mechanisms to reduce spatial and temporal search-space. The computational speed of the entire system can be improved because a majority of the negative candidates may be rejected by the initial rejecters using simpler decision rules with a low computational load.

2. Rejection mechanisms for single images: Cascaded rejection mechanisms have been developed to process single images rapidly at high accuracy rates. The mechanisms are designed to meet the requirements of performance, computational speed, and adaptation for vehicle surveillance applications such as stolen car detection systems. One-pass algorithms are proposed to extract candidates of compact plate regions and segment plate characters precisely and compactly.

3. Rejection mechanisms for video sequences: In this study, we proposed to exclude repeated patterns with the similar appearances in the same location of consecutive

frames, which usually include stopped vehicles or regular backgrounds. For efficiency, repeated patterns were detected only on the candidates of compact plate regions, named spatiotemporal SSR, based on a block-based mechanism by estimating the tangent distance, which is invariant to the variations in positions, sizes, rotations, or brightness. Moreover, a bi-level one-pass plate extraction (BOPE) algorithm developed to extract plates accurately even in complicated situations.

1.5 Organization of this Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 is a review of related research. In Chapter 3, we discribe approaches to detect license plates under two special situations: the license plates with different appearances and the frame inputs with high resolution. In Chapter 4, we will introduce our proposed cascaded license plate framework in detail. Chapter 5 describes how to reject non-plates or non-characters for single images. Furthermore, we developed a real-time statistical license plate recognition system by some one-pass rejection mechanisms. Although variations in plates and environments are existed, only candidates of license plates are extracted and outputted by our system. In Chapter 6, we present the spatiotemporal search-space reduction for video sequences to effectively save the computational load. Chapter 7 offers some conclusions and directions of future work.

Chapter 2 Related Research

In the last decade, many researchers [9–41,41–76] have focused heavily on plate recognition. We will briefly review a number of works related to our research on automatic license plate extraction in single images and video sequences in this chapter.

The first step in the recognition process is obtaining a frame of the vehicle, usually by use of a CCD camera. In the related publishes, most publishes analyze license plates from grayscale frames. Some publishes use color frames in RGB [16,26,56,69–73,76], YCrCb [22,38], rgb [58], HSV [68], HLS [69], or HSI [58,66,67] spaces; few publishes the IR image (cite...) Then, some pre-processing algorithms, such as noise reduction and histogram equalization, would be performed to enhance images.

In general, a general framework for an license plate recognition system would consist of three modules: (1) license plate detection(LPD), (2) plate character segmentation(LPS), and (3) optical character recognition(OCR). In LPD, many methods ranging from simple techniques to sophisticated mechanisms have been developed to detect license plates based on the features of plate characteristics or statistical representations. In LPS and OCR, the available methods generally use approaches that are similar to those used in license plate detection except that features are extracted from characters. Moreover, some publishes [10,33,54,60–65] would normalize the skew of the license plates before segmenting plate characters. Instead of skew normalization, Naito *et al.* [9] recognized inclined plate characters in the OCR module. For better representation, a Markov random field [74,75] could be used to perform super-resolution of license plates [30] in videos.

The natural control parameters for extracting license plates are the feature extraction and classification. An ideal feature extractor would distinctly represent plates and non-plates that make the classification process trivial; conversely, a strong classification process

Table 2.1: List of research focus

	LPD	LPS	OCR
[8,11,21,24,27,32,37,38,42,46,47,52,56,67,68,72,73,76–92]	Yes		
[10,16,19,23,62,63,93]	Yes	Yes	
[9,12–15,17,18,25,26,31,33,34,40,43,51,53,58,64,66,69,70,94–111]	Yes	Yes	Yes
[35,112,113]		Yes	
[39,41,41,45,54,114–117]		Yes	Yes
[28,29,50,61,65,118,119]			Yes

would not require a sophisticated feature extractor. There is a large number of possible feature types and associated classification measurements which emphasize different license plate properties like pixel intensities, color, teture, edges, etc. Currently, most researchers prefer a hybrid detection algorithm, where multiple features are involved in order to make the algorithm more robust. The algorithms proposed in this study is also hybrid algorithms. Different feature types and classification methods would be summarized in the following.

2.1 List of Research Focus

Due to the different goals being emphasized in each publish, the comparison of research focus is shown in Table 2.1. The value, Yes, in the filed means that the publishes listed in the second column mentioned that they did contributions in the corresponding module.

2.2 Features Types

Gradient features:

Gradient features are usually the most important since the features are insensitive to scale, rotation, size, or colors. After thresholding the gradient values, edges would be analyzed in many publishes [12,15,19–21,25,33,36,42,51,53,55,58,62–64,67,79,80,83,91,98, 103,104,114,120]. Moreover, vertical edges [15,20,25,33,36,55,64,79,80,83,91,120] would be the most popular gardient feature to represent the license plate areas. By limiting the colors of license plates, Xu *et al.* extract color edges from an RGB image; Chang *et al.* [58] extract color edges from an *rgb* image; Yang *et al.* [68] extract color edges from an HSV image.

Statistical features:

Statistical features for LPD or OCR are usually composed of the covariance matrix [84], the density [8,16,25–27,36,44,48,76,79,81,82,85,87,91,120], density variance [85,87,91,120], Haar-like features [85,87–89,91,111,120], or Gabor features [23,117]. The region density may be measured from the edges [8,36,76,82] or the gradients [91,120] of the license plate region. Wu *et al.* [86] use the frequency of zero crossing on the map of edges.

Shape-based features:

Shape-based features are usually composed of the skeleton [117], the size [10,20,36,81,94, 111], the width or the height [10,27,31,36,53,54,79,86], the aspect ratio [8,20,25,27,36,53, 60,70,76,79,81,82,86,104,109,111], the rectangularity [8,76,82], or the orientation [25,91,94] of the license plate or plate characters. Although these features may not be scale-invariant or rotation-invariant, they are insensitive to many environment changes. The plate orientation calculated using least second moments would be adopted in some publishes [109]. Moreover, the symmetric property [22,38] of license plate regions is measured to eliminate the false positives.

2.3 Extracting Methods

Morphological methods:

Morphological operators, such as local or global thresholding, Sobel [19,24,78,83,95], thinning [113], smoothing [10,88,89,93,106,110,121], opening or closing [10,36,62,67,79,80,88, 89,107,110], or differencing [10,88,89], are widely used in license plate extraction or character segmentation because the operators does not require complex and heavy mathematical calculations.

Transformation methods:

Hough transformation is a method for detecting the borders of the license plates [24,35,53, 55,104]. Martin and Borges [31] use bottom-hat transformation to enhances the characters. Hou *et al.* [110] measure the differences of top-hat and bottom-hat transformations to extract the license plate characters. Wu *et al.* [86] use bottom-hat transformation to enhance the texture in the input image. Hsieh *et al.* [81] and Guo *et al.* [122] perform wavelet decomposition in each block of the input image and generate four subbands (smoothing, horizontal, vertical, and diagonal).

Projection:

License plates or characters could be extracted by analysing the horizontal and vertical projections [12,34,35,42,51,69,80,93,95,99,104,107]. Because characters of license plates are usually arranged in horizontal lines, some publishes [19,20,31,33,54,60,62,64,81,94, 115,123] only use vertical projection histograms to segment plate characters.

Mean Shift:

The mean shift algorithm [124] is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. Some publishes [8,76,82] would use the mean shift filter to extract candidates of license plates. Kim *et al.* [71] adopted the continuously adaptive mean shift algorithm (CAMShift)[xxx] to extract regions of license plates in the result after plate color measurement.

Sliding window:

Each sub-window [63,78,85,91,104,106,109,111,122] in the input image would be extracted and identifed by heuristic rules or classification methods. Anaqnostopoulos *et al.* [109] proposed a sliding concentric windows(SCWs) to detect regions of license plates.

Vector quantization(VQ):

Based on vector quantization, Zunino and Rovetta [77] encode each row of the input image for locating license plates. [23]

Hidden Markov Chain(HMC):

Franc and Hlavac [112] use the Hidden Markov Chain (HMC) model to describe a relation between the image and its corresponding segmentation.

2.4 Classification Methods

Template matching:

By thresholding the minimum distance calculated from the pre-defined database, template matching algorithms [8,9,13,15,18,20,49,60,64,69,76,82,94,96,99,107,121,123] verified an input pattern as a plate or a character.

K nearest neighbors (KNN):

Cano and Perez-Cortes [11,78] classify every pixels of the input image based on KNN method. Guo *et al.* use a k-means cluster to identify license plate blocks.

Cascade classifier:

For object detection, Viola and Jones [125] propose an cascade classifier trained by AdaBoost [126]. A cascade classifier can be taken as a degenerate decision tree. Some publishes [85,87–89,91,111,120] adopt the cascade classifier to identify license plates.

Neural network(NN):

Various neural network architectures [12,16,28,39,41,41,44,48,55,56,58,70,84,90,95–97,102, 105,108,109,122] are proposed and implemented for plate identification or character recognition. Artificial neural network(ANN) trained by a backpropagation algorithm is used for plate character recognition in [58,97,105,106,121]. Guo *et al.* [122], Yuan *et al.* [90], and Chacon and Zimmerman [21] classify license plates based on pulse coupled neuron model(PCNN) [127], a kind of artifical neural network model. Anaqnostopoulos *et al.* [109] trained a two-layer probabilistic neural network(PNN) to identify plate characters. Hu *et al.* [115] use a PNN to identify low-dimension test samples segmentated from actual license plate images.

Hidden Markov Model(HMM):

For character recognition, Llorens *et al.* [103] and Duan *et al.* [104] use a HMM model with the observations of the ratio of foreground pixels in a window.

Support vector machine(SVM):

Support vector machine with radial basis functions (RBFs) is usually used for plate classification [122] or plate character recognition [111,116]. Direct pixel values of an input region are scaled and the size of the region is normalized. Otherwise, Kim *et al.* [71] adopted a support vector machine (SVM) with a polynomial kernel as the color texture classifier.

Genetic algorithms(GA):

Yoshimori *et al.* [22,37,38] and Yohimori *et al.* [49] change the threshold values based on genetic algorithms to extract license plates. Karungaru *et al.* [121] use a genetic algorithm to select the three parameters, position, size, and orientation of the input characters. Xiong *et al.* [46] apply GA to seach the possible license plate area in the whole image.

Chapter 3 Detection of License Plates under Two Special Situations

3.1 License Plates with Different Appearances

This section proposes an approach to developing an automatic license plate detection system with different appearances. The car images are taken from various positions in outdoors. Because of the variations of angles from the camera to the car, the license plates will have various locations and rotation angles in an image. In the license plate detection phase, since the colors of characters and of the license plate background are generally different, the magnitude of the gradients is used to detect candidate license plate regions. The license plates are usually located on the bumper. In the car images, there are several horizontal lines. If we use the horizontal gradients, it will be difficult to separate the regions of license plates from the bumper. Thus, the magnitude of the vertical gradients is used to detect the candidate license plate regions. These candidate regions are then evaluated based on three geometrical features: the ratio of width and height, the size and the orientation. The last feature is defined by the major axis. The various rotated character images of a specific character can be normalized to the same orientation based on the major axis of the character image. Two different appearances of the license plates are shown in Fig. 4 Experimental results show that the license plates detection method can correctly extract all license plates from 102 car images taken in outdoors.

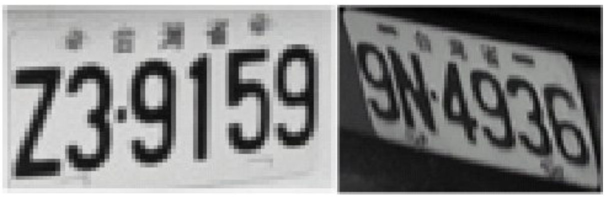


Figure 4: Two different appearances of license plates.

The remaining parts of this section are organized as follows. Section 3.1.1 studies the motivation and modules of the orientation normalization and inverse rotation transformation. Section 3.1.2 presents the procedure of license plates detection. Section 3.1.3 shows experimental results and Section 3.1.4 includes some concluding remarks.

3.1.1 Orientation Normalization

This section aims to detect the license plates of the car image with various locations and to recognize the rotation-free characters in the license plates. It is useful to derive the major axis which shows the orientation of the image to detect and recognize license plates. In the license plate detection phase, the major axis is measured in possible license plate region to evaluate the possibility to be a license plate region. Figure 5 shows the major axis on each image of the character R in five different rotation angles, where the dash lines represent the major axis of the character image.

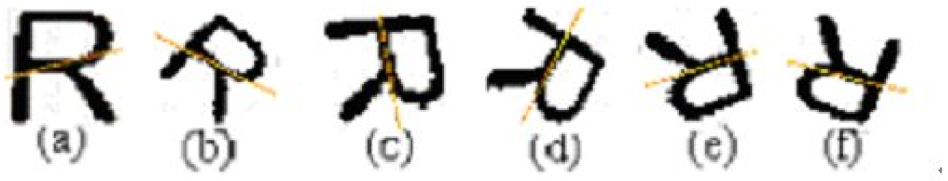


Figure 5: The images of character R in different orientations. The dash lines represent the major axis.

When the rotation angles of a specific character image are between 90^o and 270^o such as Figs. 5(d)-(f), the normalized character images is inverse. However, the situation is regardless because the license plate images would not have rotation angles between 90^o and 270^o.

In general, the new position for the pixel in the original image after rotating an angle is defined below [128]

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.1}$$

The rotation transformation will result in some holes after the rotation transformation is applied. In Fig. 6(b) the destination pixel 2 is mapped from two source pixels, while the destination pixel 1 is mapped from none of the source pixel. These undefined destination pixels produce holes in the image. To solve the problem, for each pixel of the rotated image, the relative origin pixel is checked to see if it is the black pixel of the character image. If it is, the pixel of the rotated image is marked as character pixel; otherwise it is marked as non-character one.

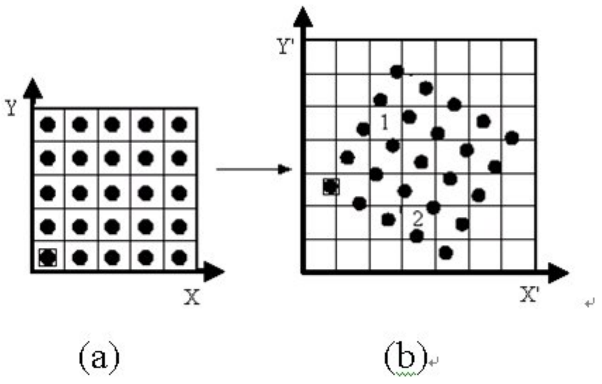


Figure 6: The rotation result of an image block. (a) The original image; (b) The rotated image with a -30-degree transformation.

The processes of orientation detection will be discussed in the following.

A. Orientation Detection

In the binary image, we first define that the mass is the black pixels whose gray level is 1. The moment of mass of the binary image is the distribution of the mass throughout the binary image. Horn [129] mentioned that the first moment of mass which is defined as mass times distance could be used to derive the center location of the mass and the second moment of the mass could be measured the distribution of mass relative to axes through the center of the mass. And the orientation, of the mass is derived from the least second moment of the mass. Then, the major axis of the mass can be achieved from the orientation and the center. The steps to derive the orientation from the binary image are described in the following.

The first moment of mass in the binary image is defined as

$$C = (x_c, y_c) = \left(\int\int xg(x,y)dxdy, \int\int yg(x,y)dxdy \right) \tag{3.2}$$

where $g(x, y)$ is the black point (x, y) which gray level is 1 in the binary image.

The second moment of mass in the binary image is equal to mass times square of the distance from the black point in the binary image to a line as shown below:

$$S = \int\int r^2 g(x,y)dxdy \tag{3.3}$$

where r is the perpendicular distance from the black point (x, y) to a line L. In Fig. 7, for a particular line in the binary image, two parameters are defined: the distance from the origin to the closest point on the line, and the angle between the x-axis and the line, which is measured counterclockwise. The equation of the line is presented as follows.

$$x\sin\theta - y\cos\theta + t = 0 \tag{3.4}$$

Note that the line intersects the x-axis at $\frac{-t}{\sin\theta}$ and the y-axis a $\frac{+t}{\cos\theta}$. The closest point on the line to the origin is located at $-t\sin\theta, +t\cos\theta$. Suppose that the point $(x_0, y_0$ is

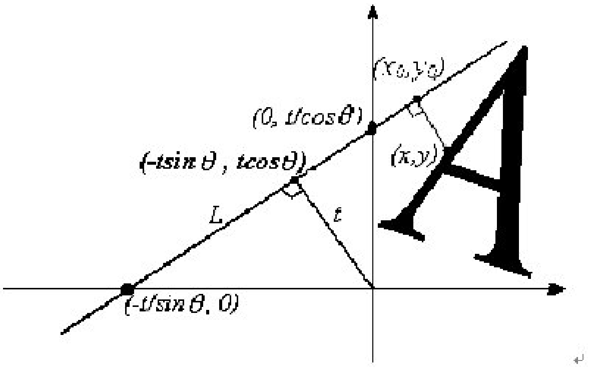


Figure 7: The coordinate diagram.

located on the line The equations for the point (x_0, y_0) on the line L are as the following:

$$\begin{aligned} x_0 &= -t\ sin\theta + r\cos\theta\ and \\ y_0 &= +t\cos\theta + r\sin\theta. \end{aligned} \tag{3.5}$$

Given an arbitrary black point (x, y) in the binary image, the shortest distance between (x, y) and the line L is defined as

$$\begin{aligned} r^2 &= (x - x_0)^2 + (y - y_0)^2 \\ &= t^2 + 2t\,(x\sin\theta - y\cos\theta) - 2r\,(x\cos\theta + y\sin\theta) + r^2 + (x^2 + y^2)\,. \end{aligned} \tag{3.6}$$

Totally differentiating with respect to we obtain

$$r = x\cos\theta + y\sin\theta. \tag{3.7}$$

Substituting the equation 3.7 back into the equation 3.6 lead to

$$r = x\sin\theta - y\cos\theta + t. \tag{3.8}$$

By substituting the equation 3.8 back into the equation 3.3, the second moment of mass can be derived as

$$S = \int\int \left(x\sin\theta - y\cos\theta + t\right)^2 g(x, y)dxdy. \tag{3.9}$$

Because the second moment of mass crosses the center of the binary image, we can substitute the equation 3.2 into the equation 3.9 and totally differentiating with respect to t, we obtain

$$(x_c\sin\theta - y_c\cos\theta + t) = 0, \tag{3.10}$$

where (x_c, y_c) is the center of the binary image. Without losing the generality, we can change the coordinate to $x' = x - x_c$ and $y' = y - y_c$ the equation 3.10 can be rewritten as follows:

$$x\sin\theta - y\cos\theta + t = x'\sin\theta - y'\cos\theta, \tag{3.11}$$

and we can substitute the equation 3.11 back into the equation 3.9, then the new equation is obtained.

$$S = \int\int \Big((x')^2\sin^2\theta + 2\left(x'y'\right)\sin\theta\cos\theta + y'^2\left(\cos 2\theta - 1\right)\Big) g(x, y)dx'dy' \tag{3.12}$$

Total differentiating S with respect to θ and we can obtain

$$\tan 2\theta = \frac{\int\int 2(x'y')dx'dy'}{\int\int x'^2 dx'dy' - \int\int y'^2 dx'dy'} \tag{3.13}$$

Finally, we obtain the orientation θ,

$$\theta = \frac{1}{2}\arctan\left(\frac{\int\int 2(x'y')dx'dy'}{\int\int x'^2 dx'dy' - \int\int y'^2 dx'dy'}\right) \tag{3.14}$$

The major axis in the binary image is defined as the line which slope is θ and the major axis crosses the center.

3.1.2 License Plates Detection

The first step of the license plate detection system is to detect the license plate regions of the input car images. Due to the similar colors of the license plate background and that of the car body, it is difficult to detect the boundary of the license plate from the input car images in outdoors. Because the color of characters is different from that of the license plate background, the gradients of the original image are adopted to detect candidates of license plate regions. Figure 8 shows the processing flow of license plates detection.

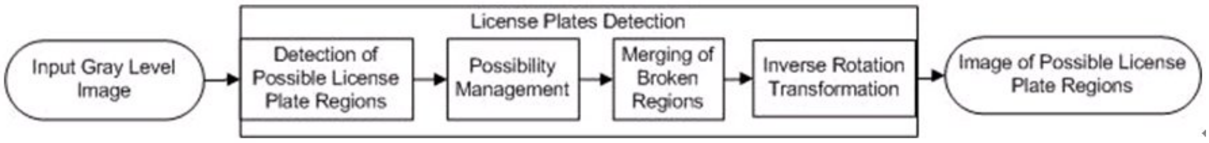

Input Gray Level Image
License Plates Detection
Detection of Possible License Plate Regions
Possibility Management
Merging of Broken Regions
Inverse Rotation Transformation
Image of Possible License Plate Regions

Figure 8: The system diagram.

This section presents the details of license plates detection. The procedure consists of four major steps: (1) detection of possible license plate regions, (2) possibility measurement, (3) merging of broken regions, (4) inverse rotation transformation. The last step, inverse rotation transformation, has already been described in the previous section. The details of the remaining steps are explained as follows.

A. Detection of Possible License Plate Regions

At the first step of the license plate detection phase, the possible license plate regions are detected from the vertical gradients of the input car images. The vertical gradients are derived by multiplying with a mask value for each pixel and its neighboring pixels. In the vertical gradients image, the license plate region is the area with large local variance. The local variances of the vertical gradients image are measured with a local window mask. In this section, in order to cover the characters in the license plate of the input car images, the size of the local window mask is set as 11 × 7. The smaller the window size is, the more possible the license plate regions are separated, while the larger the window size is, the over detected license plate regions occur. Figure 9 shows the possible license plate regions with three different window sizes: 7 × 3, 11 × 7 and 15 × 11.

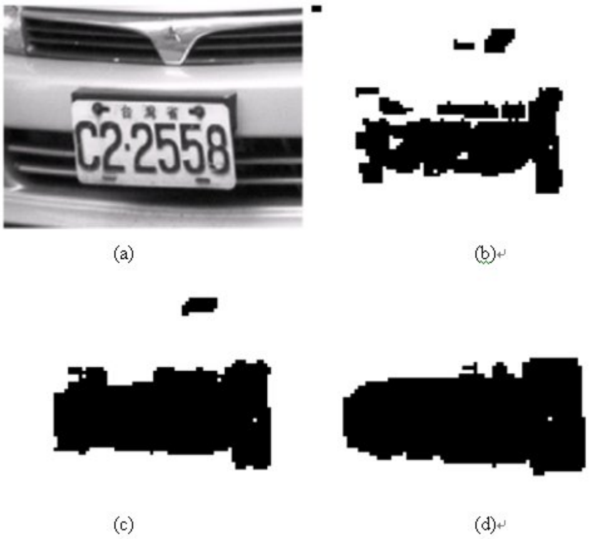


Figure 9: (a) The license plate image; (b)-(d) The possible license plate regions with the sizes of window mask, 7 × 3, 11 × 7 and 15 × 11.

The pixel is defined as 1 for possible license plate regions. When we threshold the local variance image, the image of possible license plate regions is obtained. Figure 10(a) shows the image of a car with a license plate, where the colors of the license plate background and that of the car body are similar. Figure 10(b) displays the vertical gradient image of Fig. 10(a). Figure 10(c) and Fig. 10(d) demonstrate the local variance image of Fig. 10(b) and the possible license plate regions, respectively.

There may be some noise in the images of possible license plates such as holes and single dots. An opening operation of morphological analysis, in which the dilation operation is performed after an erosion operation, is applied in order to reduce the undesired effect of noise and to separate the regions that were slightly connected.

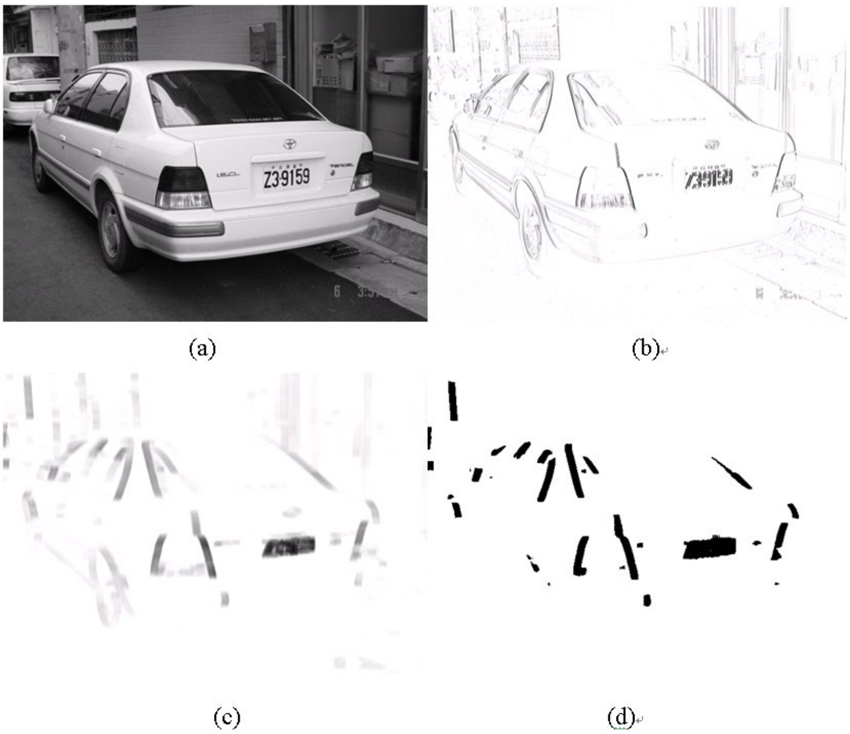


Figure 10: (a) The car image with a license plate; (b) The vertical gradients of Fig. 10(a); (c) The local variance of Fig. 10(b); (d) The possible license plate regions.

B. Possibility Measurement

To detect the most possible license plate regions from the candidate plate regions, the geometrical properties of the license plate are introduced to measure the possibility value. The following defines the geometrical features:

- Area: If the candidate region is large, it is more likely being a license plate. A higher possibility value represents a more possible license plate region. The possibility of the area is defined as $\frac{N_s}{\sum N_s}$, where N_s is the number of boundary rectangle of the possible license plate region, s.

- Orientation: As described before, the orientation of each possible license plate region can be measured. A license plate usually appears as a horizontal rectangle. The smaller the orientation of the possible license plate region is, the higher the possibility value is. The possibility of the orientation is given by $\frac{90-\theta}{90}$, where θ_s is the orientation of the possible license plate region, s.

- Density: The ratio between the black regions and the area of the bounding rectangle is defined as the density of the license plate region. The license plate is always a rectangle. A higher density value means that the region is more likely to be a rectangle and to be viewed as a license plate region. The possibility of the density is defined as B_s/N_s, where B_s is the number of the possible license plate region, s.

For each possible license plate region, s, the possibility value $p(s)$, is defined as the weighted sum of the above three features, as shown below.

$$p(s) = \omega_1 \frac{N_s}{\sum N_s} + \omega_2 \frac{90 - \theta_s}{90} + \omega_1 \frac{B_s}{N_s} \tag{3.15}$$

where ω_i is the weighting coefficient. We need to select proper ω_i that can keep a high detection rate. These values are determined according to experimental results. In this study, $\omega_1 = 0.2,\ \omega_2 = 0.3,\ and\, \omega_3 = 0.5$ are adopted.

C. Merging of Broken Regions

After the detection of all candidate license plate regions, a license plate is probably separated into several adjacent regions. In Fig. 11(a), since the distance between the characters F and 4 in the license plate is larger than the threshold of the window mask defined above, two separated candidate license plate regions are generated. These separated regions have to be merged to extract the accurate license plate region.

Assume that $s1$ and $s2$ are two possible license plate regions and s is the merged region of $s1$ and $s2$. Regions $s1$ and $s2$ are merged when the following two rules are satisfied.

- The distance between $s1$ and $s2$ is smaller than a threshold value.

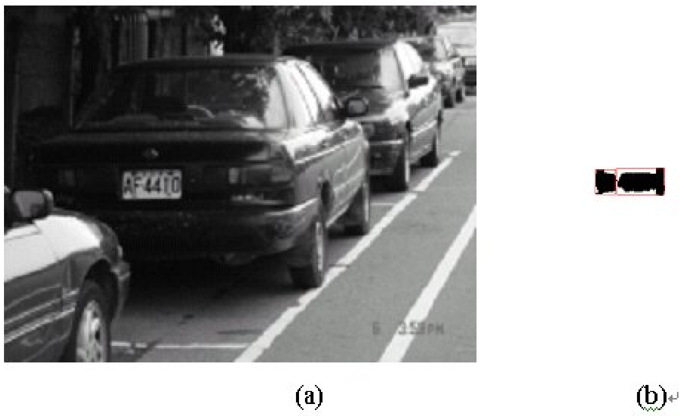


Figure 11: (a) The car image with a license plate; (b) The separated possible license plate regions.

- The possibility value of the merged region s is larger than both of $s1$ and $s2$.

The merging operation is repeatedly performed until no regions could be merged. Then, the region with the largest possibility value is viewed as the license plate region.

3.1.3 Experimental Results

The system proposed in this chapter has been applied to 102 images with 104 license plates, involving vehicles at different pan/tilt angles. We implemented the proposed system on a Pentium II 300MHz PC with C++ language under Windows environment and used Nikon 5700 digital camera as an input device. For the license plate detection method, Fig. 12 shows the original car images, the possible license plate regions and the result image of license plate detection. There are 108 totally license plate images extracted from the test images.

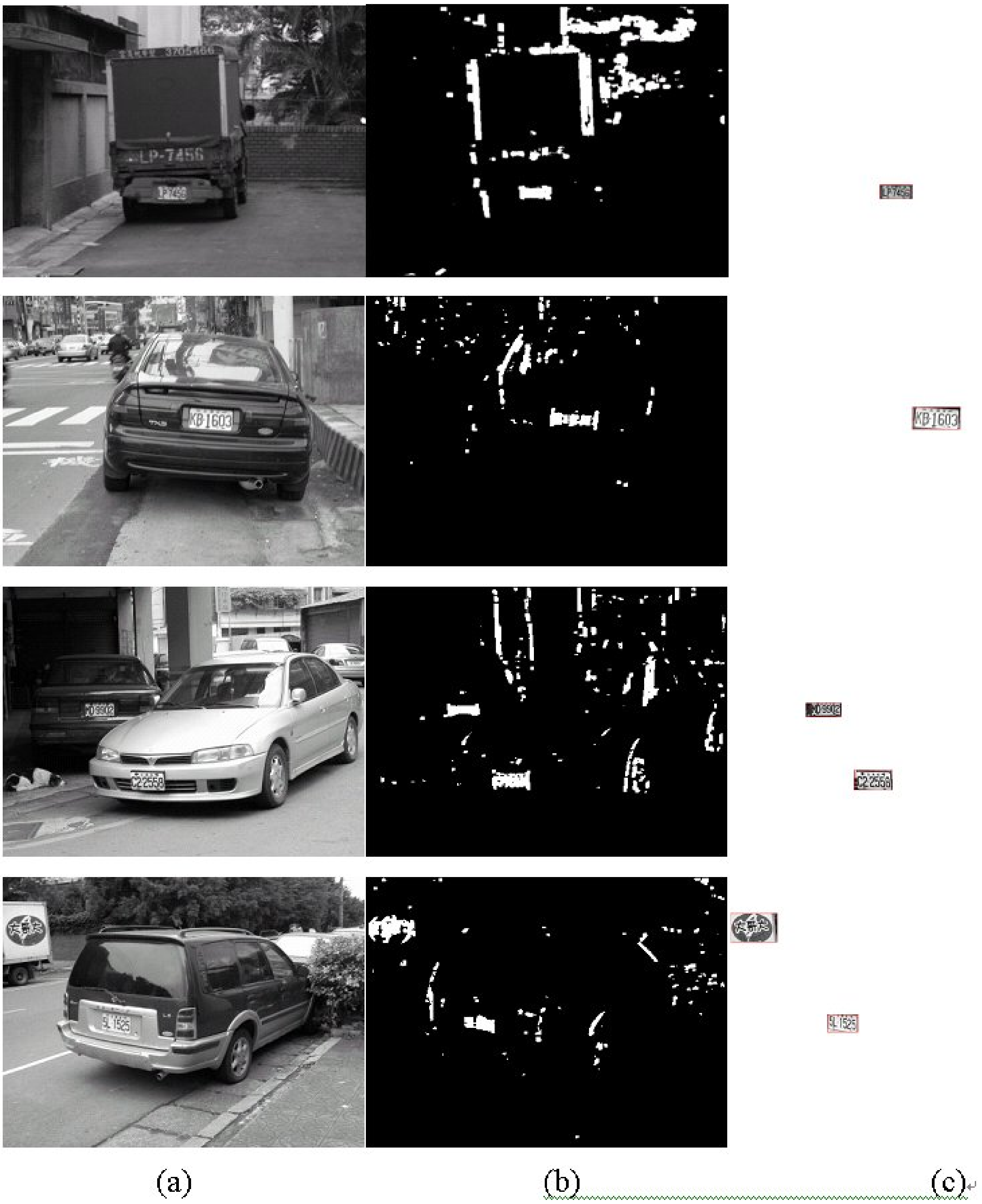


Figure 12: (a) The car images with license plates; (b) The possible license plate regions; (c) The license plate detection results.

3.1.4 Conclusion

In this chapter, we have proposed an automatic license plate detection system with different appearances. In conventional license plate detection methods, it is difficult to determine the license plate with large pan and tilt angles. The proposed methods use major axis information which is non-sensitive to rotation variance to detect the license plate. The major axis is determined by the orientation which is the second moment of the mass and center which is the first moment of the mass in the binary image. Then, the input images can be taken from large pan and tilt angles relative to the car in outdoors. Experiments carried out on some samples of outdoors car images show the feasibility of using the proposed methods to detect the license plates. The system proposed can be applied in general security systems and car violation prevention systems.

3.2 License Plates in High Resolution Frames

To reduce the search space in high resolution inputs, this study presents two fast rejection mechanisms to extract license plates of motorcycles and vehicles on highways. First, a block-based rejection mechanism was proposed to eliminate regions of non-vehicles utilizing temporal information over the input sequence. In the mechanism, three types of blocks, low-contrast, stationary, and false blocks, would be removed to reduce the search space. Second, a spatial projection-based rejection mechanism based on orthogonal gradient projections was proposed to extract candidates of license plates. To reduce the candidates, horizontal lines were rejected in the horizontal projections of the vertical gradients, and vertical lines were rejected in the vertical projections of the horizontal gradients. The remaining candidate plate regions were further segmented and verified by the character segmentation and statistical recognition modules. In our experiments, we tested 180 pairs of images, where an image contains 2560×1920 pixels. Our rejection mechanisms successfully extracted 98% license plates and improved the computational speed of the system to 0.075 seconds per image on a personal computer with Pentium 4 2GHz CPU because 88% of pixels were excluding from processing by the plate extraction, plate segmentation, and character recognition procedures.

Although much work has been done on the recognition of license plates, most studies focused on the processing of images with only one vehicle and few studies reported on that of motorcycles. Moreover, many of these algorithms are robust to small-scale variations (such as local variations of colors or sizes); they are not intended for matching images with large differences in sizes or amounts as shown in Fig. 13. In many environments, an input image may contain several motorcycles and vehicles. The situation becomes more difficult when the target plates are occluded by others as shown in Fig. 14. In another difficult case, the gray levels of plates are similar to those of backgrounds as shown in Fig. 15. This study aims to develop methods to extract license plates of motorcycles and vehicles under these situations rapidly and effectively.

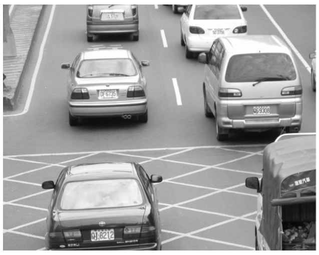


Figure 13: License plates with different sizes in an frame.

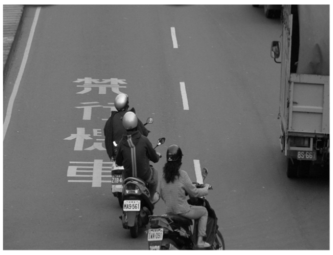


Figure 14: License plates with partial occlusion in an frame.

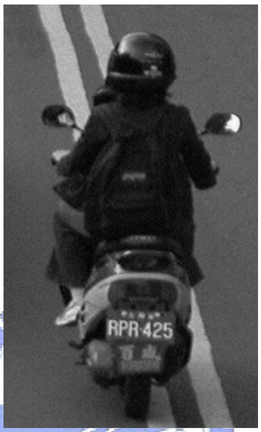


Figure 15: A license plate with unobvious borders. The plate and background have similar gray values.

Although license plate extraction in a video stream has drawn much attention, stable segmentation results would only be obtained from heavy computational approaches with some predefined criteria. The primary purpose of this study is to present two rejection mechanisms to extract license plates in high-resolution image inputs. By combining temporal rejection and spatial rejection mechanisms, we can rapidly extract license plates of motorcycles and vehicles on highways.

The remainder of this section is organized as follows. Section 3.2.1 presents the reject-based plate recognition system. Section 3.2.2 shows the mechanism of block-based rejection to reduce the search space in the high resolution inputs, and Section 3.2.3 describes the mechanism of projection-based rejection to extract license plates rapidly. In

Section 3.2.4, some experimental results are presented. Section 3.2.5 concludes this section and gives suggestions for the future works.

3.2.1 Overview of the Reject-based License Plate Extraction System

To design a plate extraction system, it is apparently efficient to reduce the search space as soon as possible. In this chapter, we design two rejection mechanisms that can dramatically reduce the search space, while ensuring the high performance of the system. The system architecture mainly composed of temporal and spatial rejection mechanisms is depicted in Fig. 16(a). To avoid processing unnecessary areas in high resolution inputs, the block-based rejection, as shown in Fig. 16(b), utilizes temporal features in consecutive frames. Three types of blocks, low-contrast, stationary, and false blocks, would be removed in this module. The three removal procedures are detailed in Section 3.2.2.

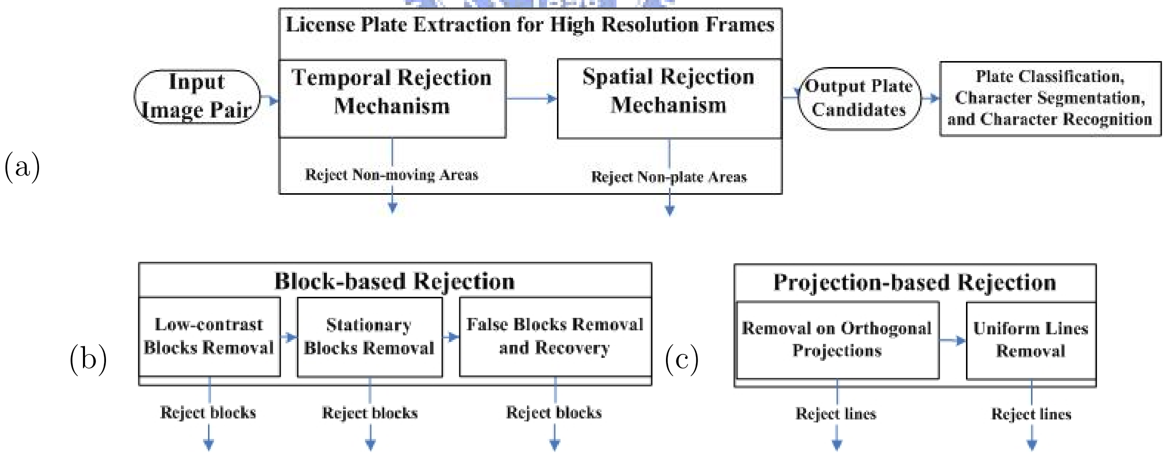
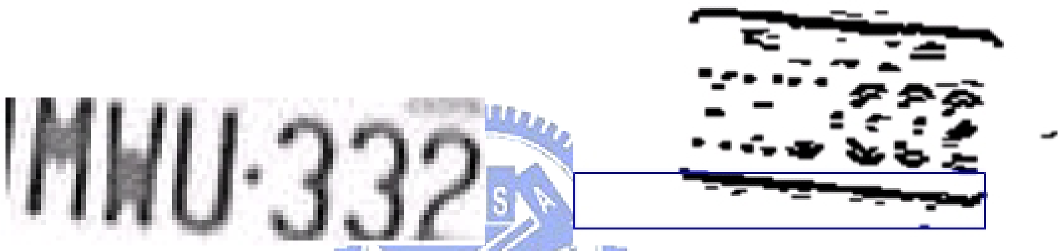

(a)
License Plate Extraction for High Resolution Frames
Input Image Pair
Temporal Rejection Mechanism
Spatial Rejection Mechanism
Output Plate Candidates
Plate Classification, Character Segmentation, and Character Recognition
Reject Non-moving Areas
Reject Non-plate Areas
(b)
Block-based Rejection
Low-contrast Blocks Removal
Stationary Blocks Removal
False Blocks Removal and Recovery
Reject blocks
Reject blocks
Reject blocks
(c)
Projection-based Rejection
Removal on Orthogonal Projections
Uniform Lines Removal
Reject lines
Reject lines

Figure 16: Schematic diagram of the rejection mechanisms for license plate extraction.

Moreover, instead of detecting license plates directly, we propose the projection-based rejection to eliminate non-characters lines. With respect to the computational load, a reject-based method is proposed to rapidly remove uniform lines on vertical and horizontal projections, named orthogonal projections, simultaneously. The projection-based rejection algorithm is shown in Fig. 16(c) and detailed in Section 3.2.3. By only detecting regions of license plate characters, the system can avoid additional procedures for removing the adornments before plate character segmentation. The last block in the system flowchart consists of three modules: plate classification, character segmentation, and character recognition, which are summarized in the following.

After we extract candidates of license plates, characters in the regions may be skew and recognized erroneously. Skew correction is important to recognize correctly the characters. In this study, the skew angle of a license plate is determined from the plate bottom border, as shown in Fig. 17.



(a) Source image (b) Edge image

Figure 17: Skew correction. The bottom border can be obtained from the original image.

After skew correction, the plate region could be classified by support vector machine (SVM), Neural Network, or cascade classifiers. Then, the classified plate region could be divided into individual character regions after binarization. In our collected license plates, the character widths are four times of the dash "-" width. Since we have six characters in our samples, we can divide the plate region into 25 $(6 \times 4 + 1)$ units and then cut the characters in the computed location. For example, the partial segmentation result is depicted in Fig. 18. For characters recognition, a classification mechanism based on support vector machine (SVM) is adopt to recognize the segmented characters of license plates. Moreover, the characters in license plates follow regulation rules. In the current study, the following four rules are used.

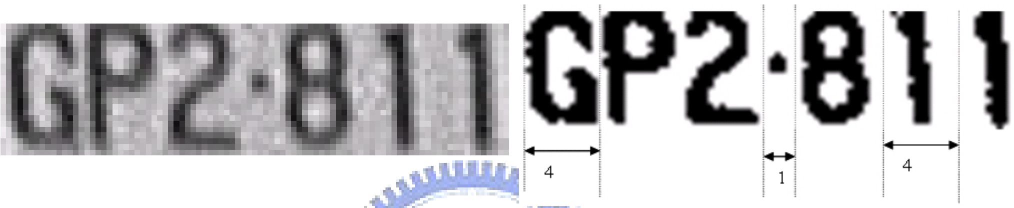
1. The license plates consist of six characters.

2. The first two characters are capital letters form A to Z.

3. The third character is a letter or a number.

4. The late three characters are numbers from 0 to 9.

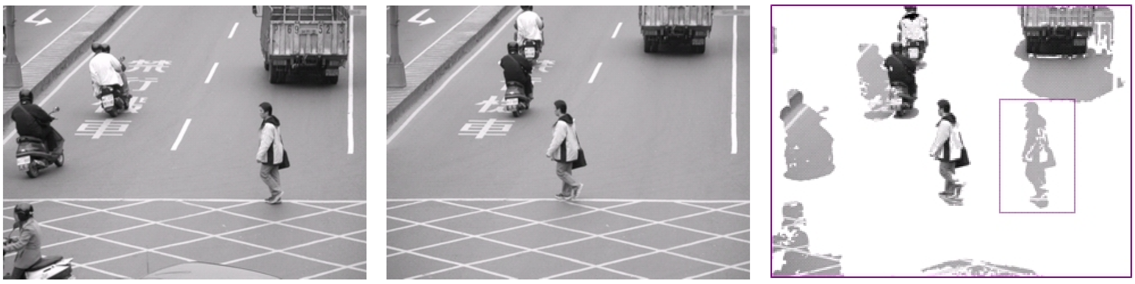
The recognition output of a character is a list sorted by the difference between the input and the candidate. If the first character in the list does not satisfy one of the rules, the following character candidate is selected for testing.



(a) (b)

Figure 18: Character Segmentation: (a) a source image and (b) the segmentation result part.

3.2.2 Block-based Rejection in High Resolution Frames



(a) The reference frame f_{t-1} (b) The current frame f_t (c) The difference frame $Diff_t$

Figure 19: A difference result of two consequent images. If the corresponding pixels in the two images have different grey-levels, the original gray value in the current frame is stored in the difference frame.

The size of an input image in our system may be very large and the image could be composed of various variants of plates and backgrounds. We would spend much time to correctly identify so many variants of the targets in the large searching space. However, only the regions of license plates result in the final output. In other words, not all pixels in the input image have to be evaluated during the recognition procedures. On highways, the plates belong to either motorcycles or vehicles, both of which would move continuously and be presented in different locations in the sequential frames. Note that motionless vehicles may be neglected because we could detect them when they start moving. To remove most motionless regions robustly, rapidly, and effectively, a block-based rejection mechanism is introduced to extract candidates of vehicle regions from the difference of two consecutive frames.

By comparing the reference frame f_{t-1} and the current frame f_t, where t stands for the time, we can generate a difference image in which the gray level in the position that the two frames have different gray levels is set as that of the current frame. Formally, the difference frame, $Diff_t$, at time t is defined as follows:

$$Diff_t\left(x,y\right) = \begin{cases} f_t\left(x,y\right) & \text{if } f_t\left(x,y\right) \neq f_{t-1}\left(x,y\right) \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

From the example in Fig. 19, we can find the candidates of vehicle objects, which are composed of real vehicle objects and background regions. Since the background regions usually exist in the regions where objects left, the appearances of these regions may contain similar intensities. In this study, all these region candidates would be categorized as three types: low-contrast, stationary, and moving blocks. Only the moving regions are the interested ones, in which license plates would exist.

In the beginning, each image is tessellated into $M \times N$ blocks. In our experiments, the size of an image is 2560 ×1920 pixels and the size of each tessellated block is 40 × 20, which is the minimum size of license plates in our experimental samples. To discard non-moving regions, the block-based rejection module is further divided into three stages: low-contrast blocks removal, stationary blocks removal, and false blocks removal and recovery.

Low-contrast blocks removal

In this stage, all blocks are classified as high-contrast and low-contrast ones according to their intensity variances. Because the colors of backgrounds and plate characters have significant differences, the blocks containing license plates will have high contrast. The low-contrast blocks could be removed by analyzing the intensity histogram. Instead of all pixels, for efficiency, we use the pixels located in the two diagonal lines in a block. Two examples after this stage are shown in Figs. 20(b) and (d).

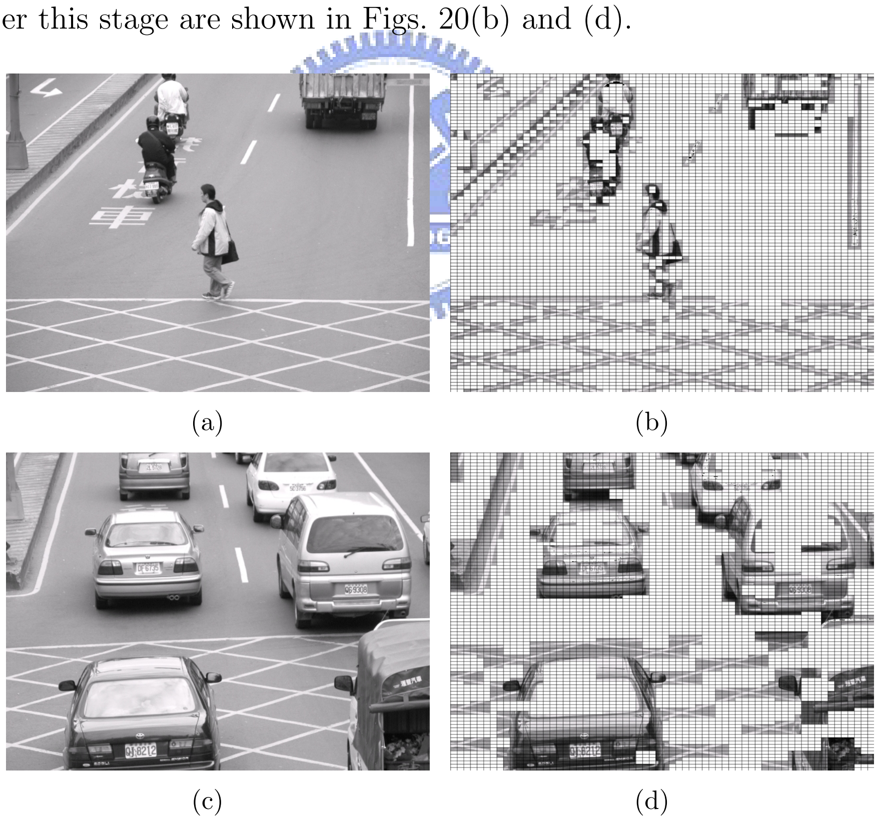

(a) (b) (c) (d)

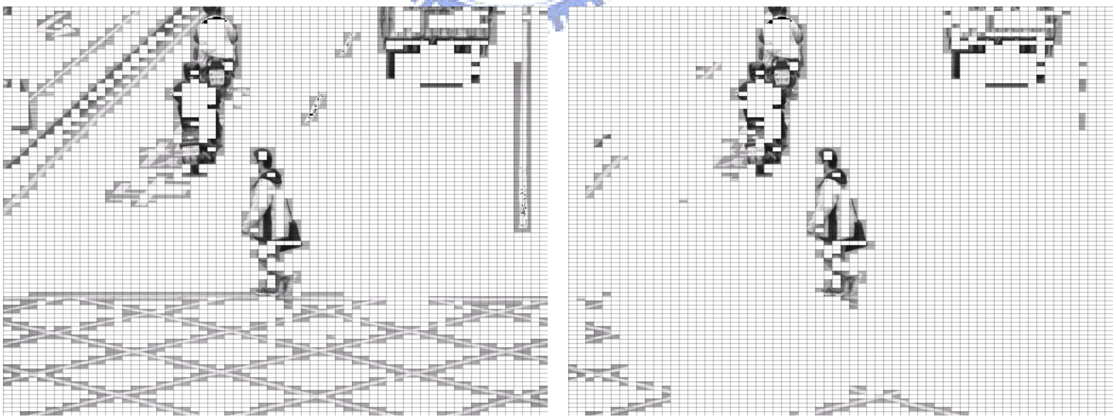
Figure 20: The white blocks have low contrast and will be removed from the current frames. (a) Current frame, F_{t1}; (b) the result of the frame F_{t1}; (c) current frame F_{t2}; (d) the result of the frame F_{t2}.

Stationary blocks removal

After we remove low-contrast blocks, the remaining blocks will be classified as stationary or moving ones. We compare the block in the current frame with the corresponding block in the reference frame. If a block in the current frame is a stationary one, the appearances of the two blocks will be similar; for a moving block, the appearances of the two blocks would be dissimilar. Stationary blocks would be discarded according to a similarity measure defined as follows:

$$sim = \frac{\left[\frac{(S_t+S_{t-1})}{2} + \left(\frac{(M_t-M_{t-1})}{2}\right)^2\right]^2}{S_t \times S_{t-1}}, \tag{3.17}$$

where M and S denote the mean and the variance for the block in the current and the reference frames, respectively. If the similarity sim is less than a threshold, then the block is categorized as a stationary one. One example of Fig. 19 after the stationary blocks are removed is shown in Fig. 21.

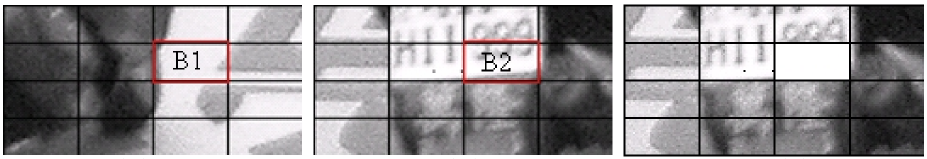


(a) The image after low-contrast blocks removal (b) The image after stationary blocks removal

Figure 21: The results of removing low-contrast and stationary blocks in the current frame of Fig. 19.

False blocks removal and recovery

The remaining blocks after above two stages may have two types of misclassification. First, some blocks in license plates would be misclassified into stationary or low-contrast ones if two different objects with similar illuminations locate on the same position in the current and reference frames. An example is shown in Fig. 22, where block B2 in the current frame would be misclassified into stationary one because the similarity *sim* between blocks B1 and B2 is small than the pre-specified threshold.

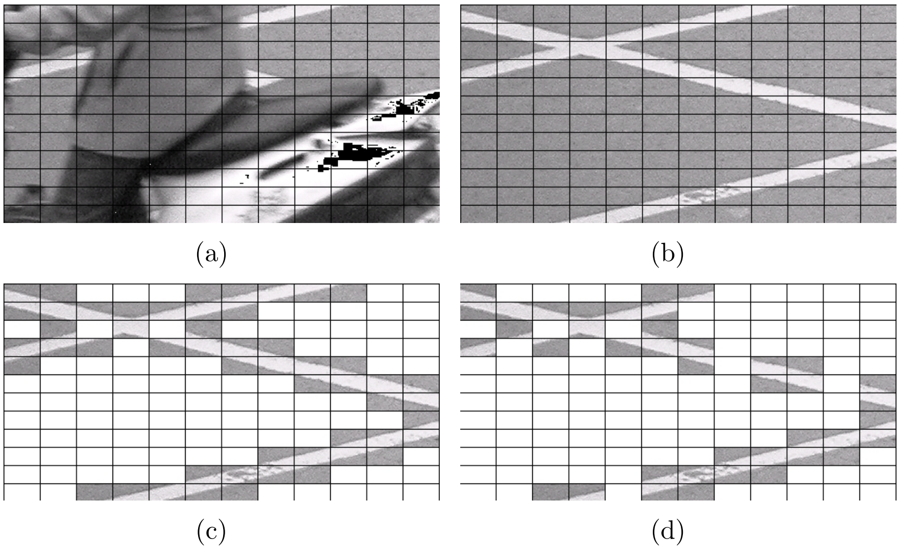


(a) (b) (c)

Figure 22: The example of the misclassified stationary block, block (2, 3).

Second, if a moving object appears on the reference frame but does not appear on the current one, the blocks in the current frame corresponding to the moving object may be misclassified as moving ones. An example is shown in Fig. 23. The non-white blocks in Fig. 8(d) are misclassified into moving ones because a moving object appears in the reference frame.

In order to correct these two kinds of misclassified blocks, we first reduce each moving block as a black pixel and other blocks as white pixels. The closing operator is then performed in the reduced image with a 3 × 3 structure element. To remove misclassified moving blocks, the opening operator is then performed in the reduced image by a 4 × 4 structure element. If a pixel in the reduced image is black, the corresponding block in the current frame is labeled as a moving block instead of a stationary one. Fig. 24 shows the result of the current frame in Fig. 19 being processed after this stage.



(a) (b)

(c) (d)

Figure 23: Misclassified moving blocks. (a) Blocks in the reference frame; (b) blocks in the current frame; (c) image blocks after removing low-contrast ones; (d) image blocks after removing stationary ones.

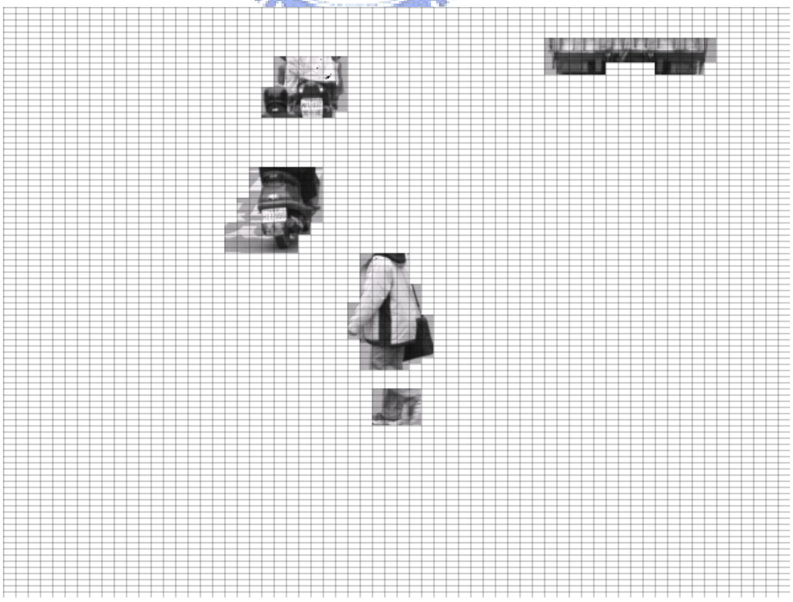


Figure 24: The non-white blocks represent the moving vehicle blocks after our block-based rejection module.

3.2.3 Projection-based Rejection on Orthogonal Projections

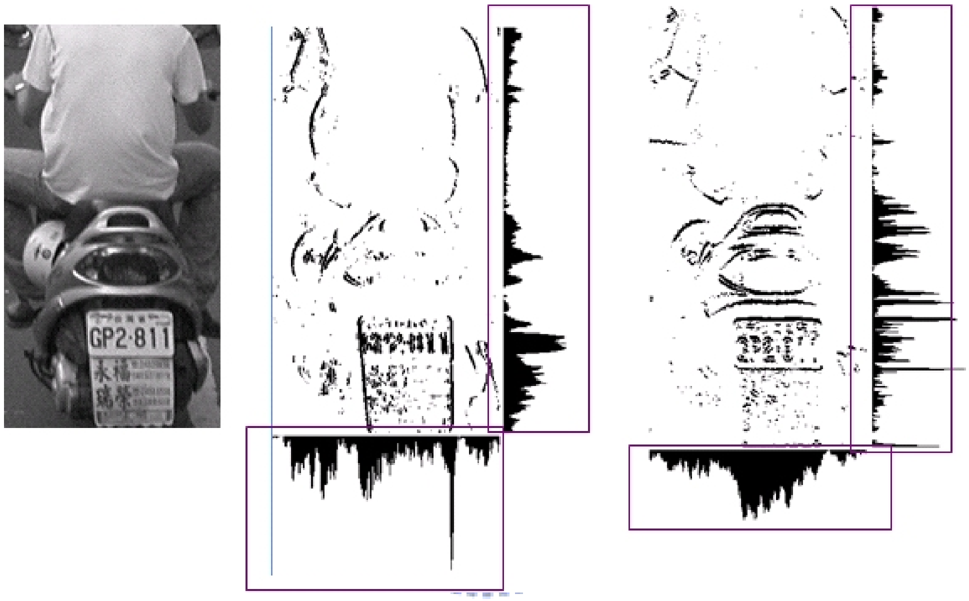
License plate characters generally have distinctive colors to backgrounds. The high gradient values between the characters and backgrounds could be used as key features to detect the desired license plate characters. However, some non-license plate areas would have similar properties and be extracted from the images captured on highways. For example, the bodies of vehicles or humans both have high gradients. To extract license plates on highways, four difficulties to be solved in this chapter are summarized in the following:

1. The vehicles on a highway may have different sizes of license plates due to different distances from the camera to the objects. The license plates closed to the camera are bigger than those of far away.
2. There are various types of objects on a highway at the same time.
3. Some objects in an image may be occluded by others.
4. The intensity values of backgrounds vary greatly because of sudden illumination changes.

Instead of extracting a whole license plate, we propose to extract the characters on the license plate only. We will describe a projection-based rejection method below to quickly eliminate non-character areas on orthogonal projections, which will be introduced in the following Sections.

Orthogonal projections removal

Since the color of plate characters are always different from that of plate backgrounds, there would be high gradient counts when we project gradients values horizontally or vertically. The locations of these high gradient counts could be detected as peaks in the projections. However, because of high background variations, several peaks may exist and the two peaks bounding the license plates have to be verified by using more information. To discriminate license plates from backgrounds effectively, we propose to project the vertical and the horizontal gradients separately, which are measured independently by the Sobel operators. For example, Figs.25(b) and (c) show the vertical and horizontal gradients of the source image in Fig. 25(a), respectively. In the two figures, the histograms of the vertical and horizontal projections are also shown.

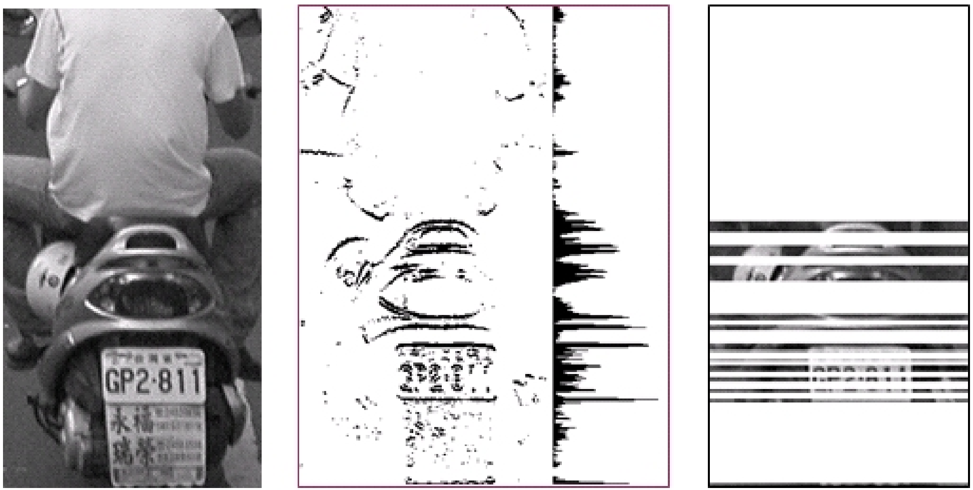


(a) (b) (c)

Figure 25: Results of gradients measurements and the vertical and horizontal projections: (a) the original image, (b) vertical gradients image, and (c) horizontal gradients image.

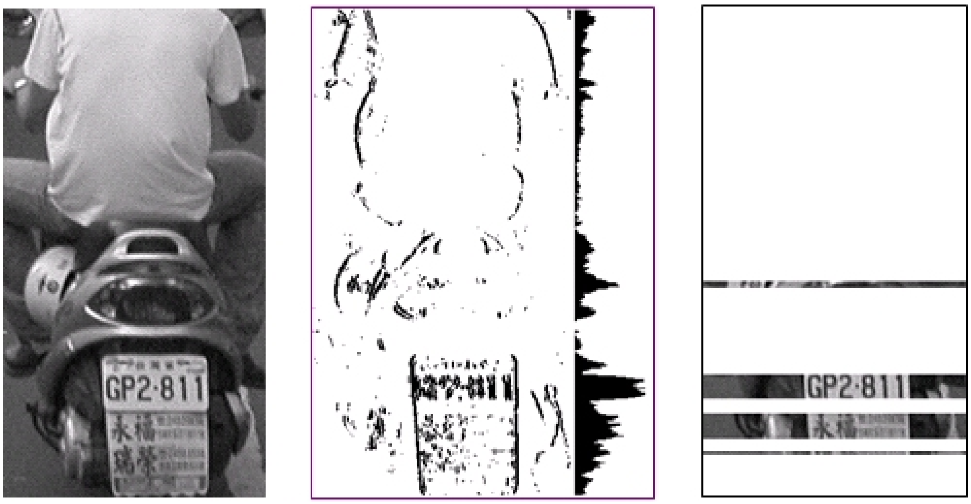
A pixel with a high gradient value is termed as a strong gradient pixel, otherwise a weak gradient one. Because a license plate usually contains many strong gradient pixels, the projection count of the scanning lines across license plates will be high. The horizontal and vertical scanning lines with weak gradient pixels are removed because they are possibly not the plate regions.

As mentioned before, there are two horizontal projection histograms generated from both horizontal and vertical gradients. For the histogram generated from the horizontal gradients, the horizontal scanning lines across characters may not have high count. Some scanning lines across characters will be removed. The regions of characters in license plates are possibly broken as shown in Fig. 26. However, this problem does not happen in that of vertical gradients, as shown in Fig. 27 because the horizontal projections of vertical gradients have consistently high counts. Thus, when we remove the horizontal scan lines with less vertical gradients count, we can keep the license plate more completely.



(a) (b) (c)

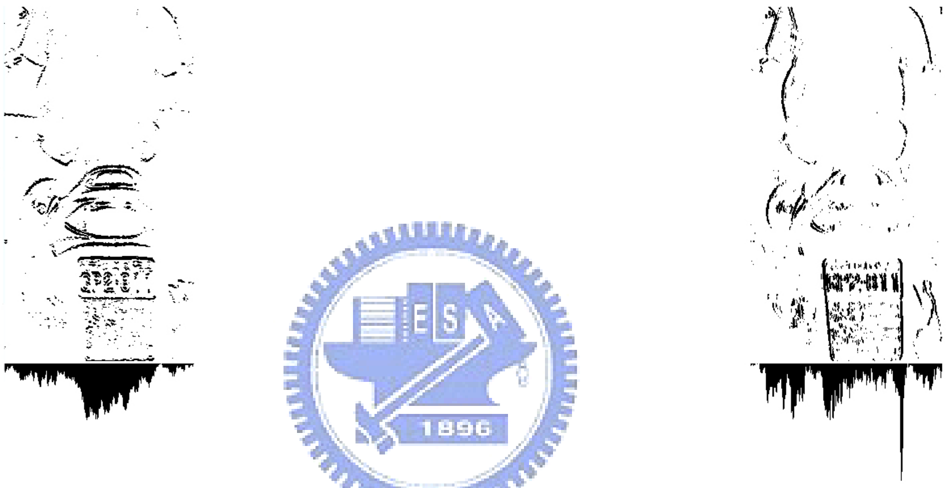
Figure 26: A result of removing horizontal lines from horizontal projections. The characters in license plates become broken. (a) The source image, (b) the horizontal gradient image, and (c) the result image.



(a) (b) (c)

Figure 27: A result of removing horizontal lines from vertical projections. The characters in license plates remain completely. (a) The source image, (b) the vertical gradient image, and (c) the result image.

To reduce the high background gradients, horizontal lines would be only rejected in the horizontal projections of the vertical gradients. The same concept can be used to remove vertical scanning lines. Characters have a low gradient count in the vertical projections of vertical gradients. Removing the scan lines with lower gradients count will break the regions of the license plates. The problem does not happen in the projections of horizontal gradients, which is accordingly used to remove vertical scanning lines as shown in Fig. 28. For the regions of moving objects in Fig. 29(a), two complete license plates are preserved as shown in Fig. 29(b) after horizontal scanning lines are removed.



(a)Horizontal gradients (b)Vertical gradients

Figure 28: A comparison of vertical projections of the horizontal and the vertical gradients. Characters have low counts in the projections of vertical gradients.

Uniform lines removal

In this stage, we will remove the scan lines with weak gradient pixels in the remaining blocks by the orthogonal projection-based method again. Fig. 30 shows that most non-plate regions would be removed, while the regions of three license plates are kept. However, the lowest license plate “IWA-091” is broken into two blocks, which is caused by the vertical projection of horizontal gradients.

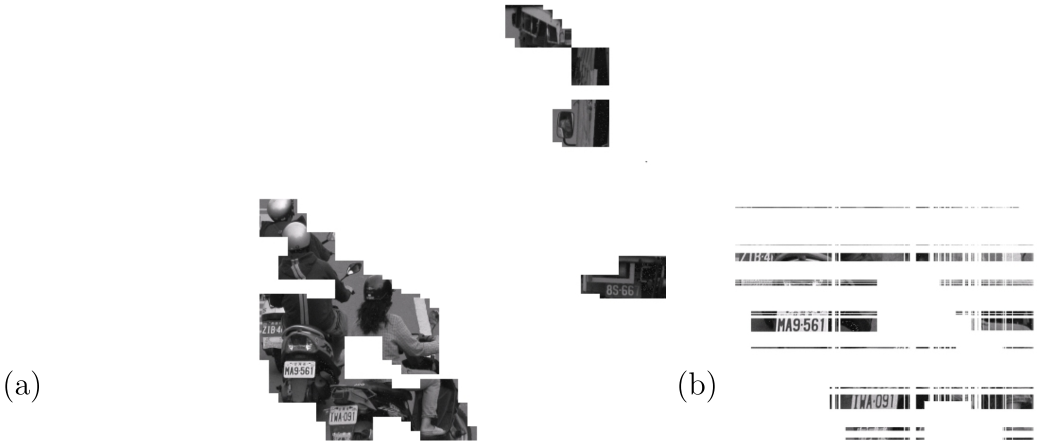


Figure 29: The result of removing non-license plate regions for the image in Fig. 14. (a) Result of the block-based rejection; (b) Result after the horizontal scan lines of non-license plate regions are removed.

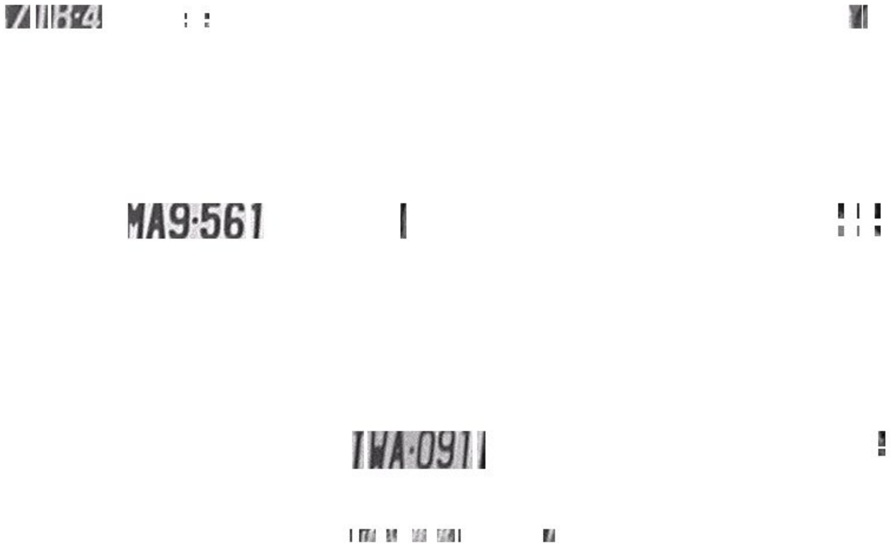


Figure 30: A result of removing scan lines in blocks. Note that the license plate "IWA-091" is broken into two blocks

Extraction of license plate candidates

The height of detected blocks is approximately equal to the height of characters in the license plate. Small blocks on the same row are merged if they are close enough and satisfy the constraint on the aspect ratio of a license plate. We can thus detect license plates of different sizes.

Not all blocks detected so far belong to license plates. We will use geometric properties of those regions to select license plate ones. The properties used are listed in the following:

1. The size of a license plate region lies in a limited range.
2. The aspect ratio is in a predefined range.
3. The scanning line in a license plate region cannot have a very long white or black run.
4. The numbers of black and white runs in the middle scanning line are kept in a limited range.

For the example shown in Fig. 31, region R1 does not satisfy property 3; region R2 does not satisfy property 1; region R3 does not satisfy property 2. To identify plate regions more reliably, more constraints could be added. The regions satisfied all the properties will be identified as the candidates of license plates.

3.2.4 Experimental Results and Analysis

Our experimental images are captured every 1.5 second by a fixed camera mounted upon a bridge. The system was implemented on a personal computer with Pentium 4 2GHz

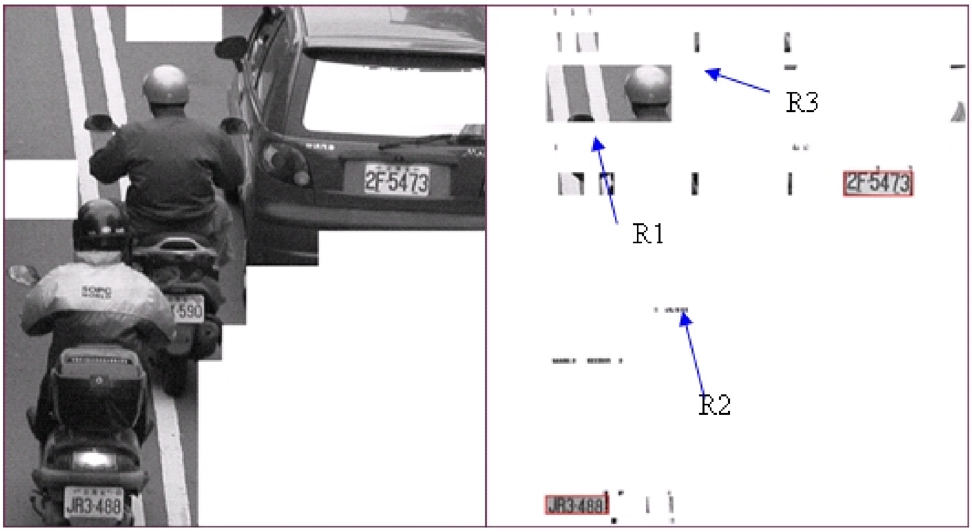


Figure 31: Candidate regions selection. Several regions are excluded since they do not satisfy the geometrical constraints.

CPU with 512 MB RAM. Each input image pair includes two images, a current frame and a reference frame. The training data contains 30 image pairs. The testing data are composed of 180 images pairs with 351 whole license plates, excluding the occluded ones. In the following, we show the experiments of the testing data according to our three modules.

For interesting region detection, the average pixels of the regions we detected are $592,667$ per image, while an input image contains 2560×1920 pixels. Thus, we can remove 88 percent of pixels. Figure 32 shows an example that five license plates of the motorcycles are detected successfully. Figure 33 shows an erroneous example where the license plate of the up-middle car is missed because the blocks near the vehicle's license plate are removed by the opening operator.

A license plate is considered as detected successfully if the characters in the license plate are classified into the same interesting region as shown in Fig. 34. Even though the vehicle is divided into two interesting regions, the license plate is viewed as detected successfully because the whole license plate is included in one interesting region. In our experiments, 344 plates are detected successfully; thus the accuracy rate is 98.0%.

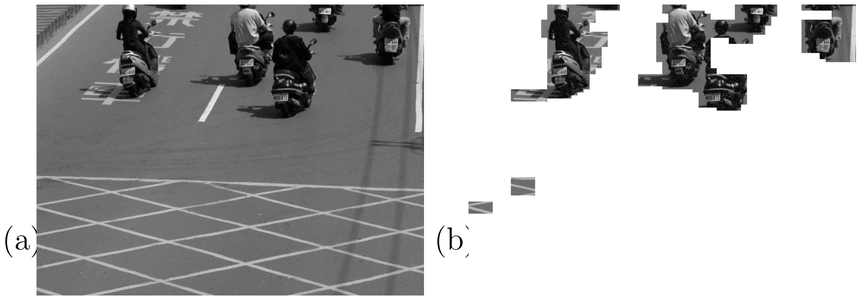


Figure 32: Five license plates detected successfully. (a) The source image; (b) The interesting regions detected.

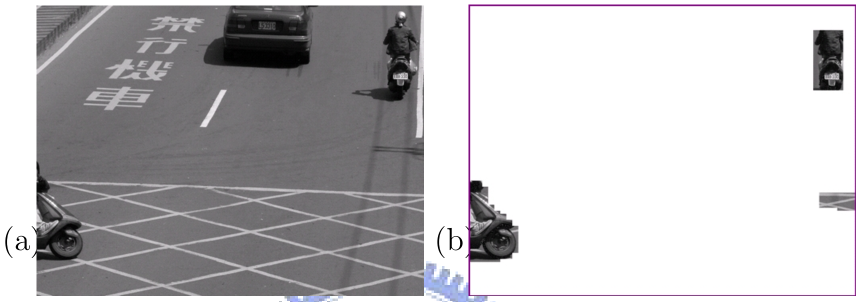


Figure 33: An erroneous example of interesting region detection. (a) Source image; (b) The result of interesting regions detection.

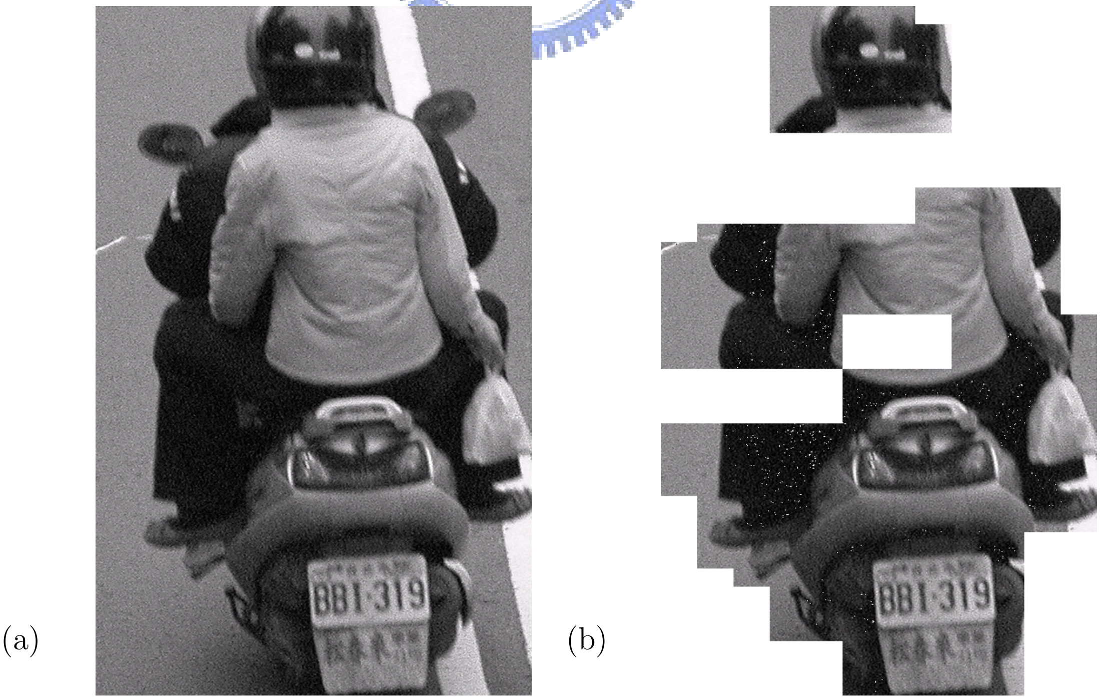


Figure 34: A result of license plate detection.(a) The test image; (b) Two regions of a moving object are detected. The plate region is detected successfully.

For license plate extraction, a license plate is extracted successfully if the characters in license plates are all extracted. There are 344 license plates in the experimental data. Among them, 325 plates are extracted successfully. The accuracy rate is 94.4%. An interesting region without license plates is removed successfully if no inside area is detected as a license plate. For the 1222 interesting regions without license plates, 1128 objects are removed successfully while 94 are fail to be removed. The accuracy rate is thus 92.3%. Moreover, applying block-based and projection-based rejections sequentially, the processing speed is 0.075 seconds per image. If more pixels could be rejected, the processing speed would be faster.

Figure 35 shows an erroneous case that the system fails to detect the license plate region because the image contains other strong-gradients points except the license plate. The area of the license plate is only a small part comparing with that of vegetables.

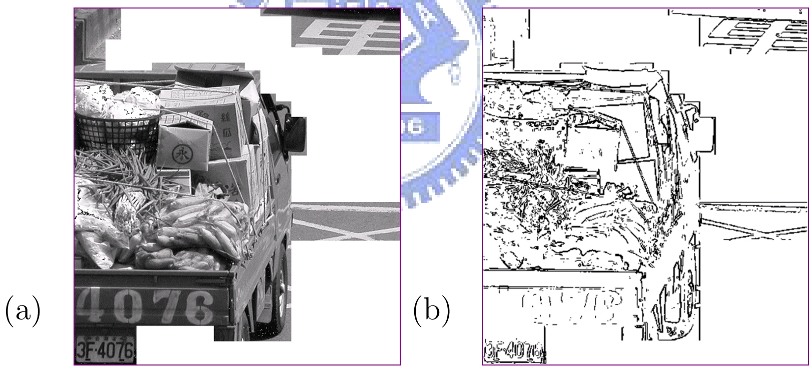


Figure 35: An erroneous example for license plate extraction. (a) Source image; (b) Image after gradients measurement.

3.2.5 Conclusion and Future Work

In this section, we have proposed two fast rejection mechanisms to extract license plates of motorcycles and vehicles on highways. The first temporal rejection mechanism, block based rejection, could reduce 88 percent searching space in the processed frames. The second spatial rejection mechanism, orthogonal projection-based rejection, provides an effective and rapid solution to extract different sizes and various amounts of license plates. By cascading the two rejection mechanisms, we could extract license plates rapidly in high resolution frames.

To improve the performance in each phase, we propose the following two suggestions for future study:

1. For block-based rejection, it may be sensitive to its structure elements. Dynamically adjusting the size of a structure element would improve the results. For example, if the blocks contain strong-gradients pixels, the structure element can be bigger than those with weak-gradients pixels.

2. For orthogonal projection-based rejection, it may be restricted by the illumination effect. A shadow or non-uniform light may cause an error in the stage. The Sobel operator cannot handle the illumination problem. Using other local operators, the mistakes may be reduced.

Chapter 4 Fundamentals of Cascade License Plate Recognition Framework

To achieve a high accuracy rate with variations in plate types or environments, spatial features and temporal information could be used simultaneously to detect and recognize plates. However, as more features are measured, the computational load may increase significantly. A tradeoff then exists in the plate recognition system between discrimination and computation. We must design a system that satisfies the constraints in various environments simultaneously. However, even while detecting a single plate type, it is still difficult to detect all plates without false positives.

In this study, we propose a cascade framework for license plate recognition composed of two representations of license plates and three strategies, which are detailed in the following.

4.1 Representations of License Plates

Given an image, appropriate representations will speed up extraction of license plates. In this study, we proposed two representations according to spatial and temporal characteristics, respectively.

4.1.1 Spatial Compact Plate Regions

The plate region contains plate characters and various adornments such as frames, screws, and subtitles. This type of region is termed the normal plate region. The presence of adornments increases the difficulties in plate character segmentation because the characters and adornments may contact each other in the captured input. It may be difficult to correctly recognize certain adornments such as screws and emblems because they may be very small and blurred; moreover, they do not provide distinguishing information for plate recognition. Therefore, these adornments should be discarded before plate character segmentation and recognition. However, the use of additional procedures for removing the adornment increases the computation time of the system. In this study, we propose to detect compact plate regions, i.e., the regions that bound the top and bottom of plate characters, in the initial stage to avoid the use of additional removal procedures. Figure 36 shows the patterns of normal plate regions and compact plate regions.

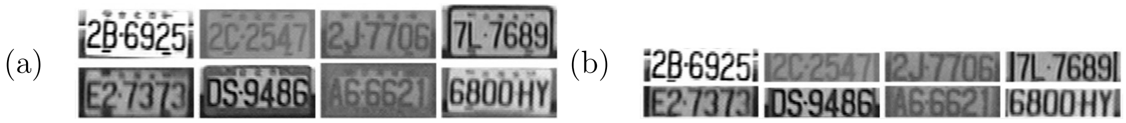


Figure 36: Patterns of (a) normal plate regions and (b) compact plate regions.

4.1.2 Temporal Repeated Regions

In a motion detection system, moving or foreground objects are usually detected first and then classified by using their characteristics. Instead of detecting moving objects directly in the cascade framework, we propose to reject repeated patterns with similar appearances in the same location of consecutive frames. In vehicle surveillance, repeated patterns usually stand for static objects, such as stopped vehicles or regular background patterns. To save computational load, these repeated patterns should be excluded from repeated classification.

4.2 Design Strategies

This study investigates the problems of plate detection in video sequences from the following directions.

4.2.1 Cascade Framework with Rejecters

The natural control parameters for balancing discrimination and computation are the complexities of feature extraction and classification. The conceptual boundary between feature extraction and classification is rather arbitrary. An ideal feature extractor would distinctly represent positive and negative inputs that make the classification process trivial; conversely, a strong classification process would not require a sophisticated feature extractor. However, it would be computationally demanding or impossible to distinguish between features that are invariant to irrelevant transformations of the input. When emphasis is placed on the computational load, it is apparently efficient to reject a majority of negatives immediately with a single relatively inexpensive process. By the concatenation of rejecting processes with increasing complexity, the overall computational load of the system can be reduced because not all input candidates are processed by the more heavy-computational processes.

Based on the above reasoning, we consider two coarse-to-fine computational designs that can dramatically reduce the computation, while ensuring that the performance of the system is high. First, a major portion of the searching space may be quickly discarded by features with increasing complexity: low computational features such as gradients followed by heavy computational features such as statistical region representations. Second, increasingly more feature dimensions are adopted to correctly discriminate between positives and negatives. The two designs allow quick rejection of incorrect input candidates, while more computation time is used for promising candidates of plates or character regions. To produce the two designs, a cascade framework is built on successively more complex processes, termed cascaded rejecters. In comparison with a process with a high accuracy rate, a rejecter only requires a slightly higher accuracy than chance and has a minimal computational load. Figure 37 shows the differences of the ROC curves between a rejecter, a strong classificer, and a random classificer. The key constraints of such a rejecter are a high TPR and a relatively low FPR, for instance, less than 50%. In other words, all, or almost all, positive targets must be preserved after they are processed by the rejecter.

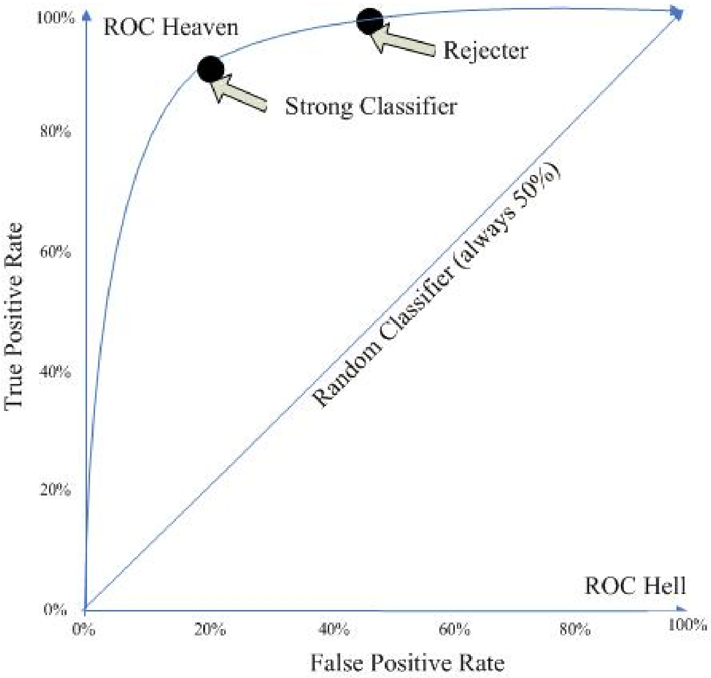


Figure 37: Differences of the ROC curves between a rejecter, a strong classificer, and a random classificer.

Based on rejecting negatives in the shortest possible time using simpler decision rules, the average computational speed of the system would be faster than that obtained by adopting the more complex processes for all input candidates. Section 4.3 summarizes the advances of the cascade framework based on a "rejection chain," which is composed of gradually complicated rejecting procedures to reject negatives (said, non-character, non-plate, and repeated regions) as quickly as possible.

4.2.2 Spatiotemporal Search-space Reduction

Since we can extract plate candidates by a very efficient algorithm BOPE using spatial information, the repeated candidates can be discarded using temporal information of consecutive frames. In consecutive frames, candidates extracted by BOPE would not have the same dimensions or not exist in the same locations due to variations in plates or environments. It is time consuming to compare the candidates in the current frame to all of the candidates in the preceding frame. In this study, we proposed to exclude repeated patterns with the similar appearances in the same location of consecutive frames, which usually include stopped vehicles or regular backgrounds.

4.2.3 Computational Reduction in Algorithmic Design

In addition to the plate characteristics, the cascade framework, and the search-space reduction, the system could not function at a high speed without appropriate algorithms. Duplicate memory access and computation would occur in different processes or even in the same process. In this study, we propose one-pass algorithms, as mentioned in Sections 5.2 and 6.2, that scans the entire input only once but can extract all candidates of compact plate regions. The values of the features could be calculated at constant time using acceleration tables, which will also be generated in a single scanning process, as mentioned in Sections 5.2.3 and 5.3.3. Moreover, Section 6.3 presented a block-based procedure to verify repeated patterns by the detected reapeated blocks, while the repeated blocks would not be calculated every frames.

4.3 Overview of the Framework

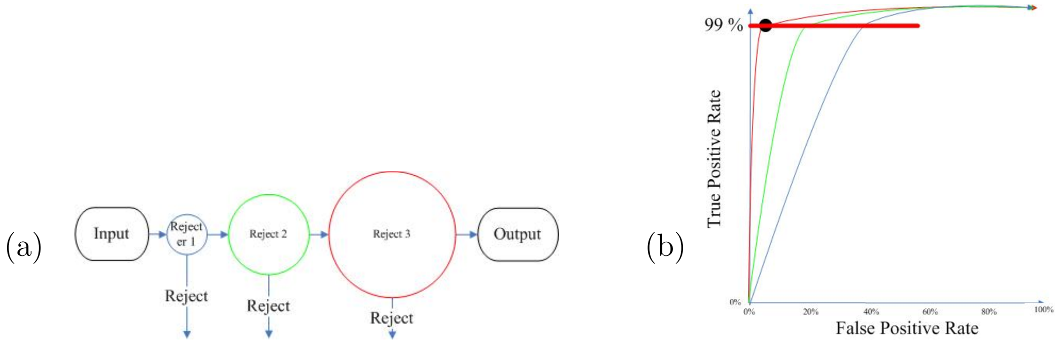


Figure 38: (a) Schematic diagram of the cascade framework and (b) ROC curves for the three rejecters.

Based on the strategies, the framework is composed of cascaded rejecters to reject negative from the output of the preceeding rejecter, as presented in Fig. 38(a) while the ROC curves of the three rejecters in Fig. 38(a) are shown in Fig. 38(b). For the cascade framework, the performance and computation speed requirements for surveillance applications are analyzed below.

4.3.1 Performance

To explain the performance requirement, we make the following three assumptions: (1) the total number of rejecters is 20, (2) the TPR of each rejecter is 99.9%, and (3) the FPR of each rejecter is 40%. Then, the final TPR and FPR of the entire cascade plate recognition system are $98\%(0.999^{20})$ and $10^{-6}\%(0.4^{20})$, respectively. The final accuracy rate is 99%; it is measured by the following formula:

$$AccuracyRate = \lambda \times TPR + (1 - \lambda) \times (1 - FPR), \tag{4.1}$$

where TPR is the rate of positives that are correctly recognized; FPR, the rate of negatives that are incorrectly recognized; and λ, the weighting coefficient, for example, 0.5.

4.3.2 Computational Speed

Under the assumption that most of the input candidates are negative, the computational speed of the entire system can be improved because a majority of the negative candidates may be rejected by the initial rejecters using simpler decision rules with a low computational load. The strategy is referred to as *computational risk reduction.* Consider the following experimental case as an example: In a 640×480 input image, the total number of plate region candidates for detecting a minimum recognizable plate size, 60×15, is $270,746$. In our experiments, approximately 20 candidates of the plate regions remain after the application of the spatial search-space reduction. In addition to computational risk reduction, duplicate memory access can be avoided by one-pass algorithms, which will be described in sections 5.2 and 6.2, respectively.

Chapter 5 Rejection Mechanisms for Single Images

This chapter describes a plate recognition system that can process images rapidly at high accuracy rates. This system is designed to meet the requirements of performance, computational speed, and adaptation for vehicle surveillance applications such as stolen car detection systems. These requirements are satisfied by adopting a cascade framework, utilizing plate characteristics, and developing fast one-pass algorithms. Our system is composed of three main cascading modules for plate detection, character segmentation, and post-processing. Each module is further decomposed into several cascading procedures, which are composed of successively more complex rejecters. The first module rapidly rejects a majority of non-plate regions by using low computational gradient features and a one-pass scanning algorithm followed by heavy computational statistical rejecters. The second module rejects a majority of non-character regions in a similar manner. A peak-valley analysis algorithm is proposed to rapidly detect all promising candidates of character regions. The third module eliminates the plate characters that do not satisfy the plate specifications. In our experiments, the system can recognize plates over 38 frames per second with a resolution of 640 × 480 pixels on a 3 GHz Intel Pentium 4 personal computer.

5.1 Introduction

Based on the three strategies, the rejection mechanisms for single images, as presented in Fig. 39(a), are composed of three cascaded modules for plate detection, character segmentation and recognition, and post-processing.

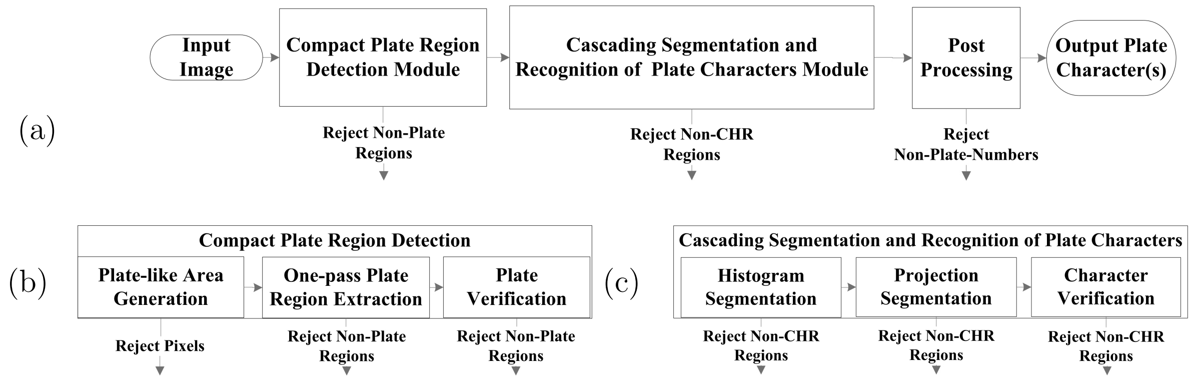


Figure 39: Schematic diagram of the rejection mechanisms for single images.

The first module shown in Fig. 39(b) aims to reduce the searching space of the plates by three cascading procedures. The first procedure eliminates most of the non-character regions by the low computational features, gradients, according to plate characteristic 1. The second procedure extracts the promising plate regions by a one-pass extraction algorithm according to plate characteristic 2. The third procedure rapidly rejects a majority of the non-plate regions by using complex rejecters with increasing complexity and higher dimensions of statistical features according to plate characteristic 3. Moreover, these rejecters could be automatically learned by a boosting algorithm from the characteristics between the plate and non-plate regions. In the second module, shown in Fig. 39(c), three procedures are designed for the aforementioned scenario in order to reduce the searching space of the characters. To extract character candidates rapidly, a peak-valley analysis algorithm is proposed to detect all promising segments from intensity histograms and projection profiles in the first and second procedures, respectively. Non-character candidates would be rejected quickly by the rejecters, which are automatically trained with character candidates detected from the preceding procedures. Then, the remaining character candidates will be recognized by an optical character recognizer (OCR). In the last module, the recognized plate characters that do not satisfy the plate specifications are discarded. The plate specifications used for the experimental data are defined as follows.

- PLATE = T1 | T2 | T3 | T3
- T1 = (M2)(N4) | (N4)(M2)
- T2 = (M2)(N3)
- T3 = (M2)(N2)
- T4 = (M3)(N3)
- M2 = Alphabet Alphabet | Alphabet Number | Number Alphabet
- M3 = M2 Alphabet | M2 Number
- N2 = Number Number
- N3 = N2 Number
- N4 = N3 Number

In general, the rejection mechanisms for single images are a cascade of rejecters, each of which focuses on the rejection of non-characters, non-plate character lines, or non-plates.

5.2 Compact Plate Region Detection

5.2.1 Generation of Plate Region Candidates

Irrespective of the plate type, according to plate characteristic 1, the color of a plate character is always different from that of the background. Therefore, plate regions may be distinguished from most non-plate regions by using gradient features. However, horizontal gradients are not distinctive features because plates may usually contact some other parts, such as bumpers, which also have strong horizontal gradients. To reduce the searching space effectively, only vertical gradients are used in this step. The vertical gradients in Fig. 40(c) represent the plate regions that appear clearer than others. After the vertical gradient operation, pixels whose gradient values are higher than a threshold are used as the pixels of plate candidates. The threshold is obtained automatically from Otsu's method [130] and the thresholding map is termed G_p.

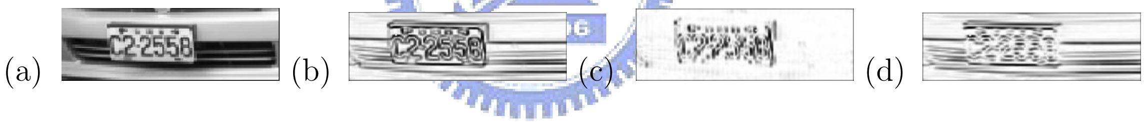


Figure 40: Different gradient results of an input image: (a) Original image, (b) Sobel gradients, (c) vertical gradients, and (d) horizontal gradients.

5.2.2 Extraction of Compact Plate Regions

Non-plate Run Suppression

To extract the compact plate regions with precision, four types of non-plate runs should be suppressed. Here, a "plate run" is defined as the successive pixels of plate candidates in G_p, while a "non-plate run" is defined as successive pixels of non-plate candidates. Let W_p and H_p denote the width and height of a compact plate region, respectively. We define suppressed non-plate runs as follows:

1. Vertical plate runs higher than H_p
2. Horizontal plate runs longer than W_p
3. Plate runs located between two non-plate runs longer than the maximum width of plate characters
4. Small isolated plate regions

Runs of types 1 and 2 are mainly caused by plate frames or bumpers, while those of types 3 and 4 are generated by screws, noises, and other parts connected with plate characters.

Compact Plate Region Extraction

After the non-plate runs are suppressed, the remaining plate runs may not comprise a set and they may be scattered significantly, thereby making it difficult to compose compact plate regions. A horizontal smearing technique is then used to group two plate runs whose distance is less than the criterion λ_s into compact plate regions. The smearing criterion could be determined from the plate characteristic 2. Since the distance between two plate runs should not exceed the maximum width of the plate characters, λ_s is defined as W_p/N, where N indicates the number of plate characters. In our experiments, the range of W_p is set to four times H_p according to the plate specifications, while H_p is selected from 15 pixels (minimal height of the recognizable plate size) to the quarter-width of the input image. Finally, the connected components whose size exceeds a threshold defined by type 4 suppression are extracted as candidates of compact plate regions. Moreover, the map, G_p, should be restored in the first step of every iteration because it would be modified by the non-plate run suppression. The extraction steps are summarized in Alg. 1.

```
Data: $G_p$
Result: candidates of compact plate regions
foreach $(w_p, h_p)$ in $W_p$ and $H_p$ do
    $\lambda_s \leftarrow (w_p/N)$ ;
    1. restore the map, $G_p$;
    2. remove vertical plate runs higher than $h_p$ (type 1 suppression);
    3. remove horizontal plate runs longer than $w_p$ (type 2 suppression);
    4. remove horizontal plate runs which are shorter than $\lambda_s$ and located between
    two horizontal non-plate runs longer than $w_p$ (type 3 suppression);
    5. smear $G_p$ with the parameter, $\lambda_s$;
    6. remove regions smaller than $(h_p \times w_p)/4$ (type 4 suppression);
    7. extract connected components;
end
```

Algorithm 1: Multi-pass algorithm for extracting compact plate regions.

```
Data: the $row_y$ of $G_p$
Result: plate and non-plate runs
Initialize an array Acc[width of $G_p$];
foreach column(x) in the $row_y$ do
    if $row_y(x) > 0$ then
        Acc[x] = Acc[x] + 1;
    else
        Acc[x] = 0;
    end
    if $row_y(x) > 0$ and $Acc[x] \leq H_p$ then
        update the plate run;
    else
        update the non-plate run (Type 2 suppression) ;
    end
end
```

Algorithm 2: Algorithm for run determination, FindAllRuns.

One-pass Extraction of Compact Plate Regions

For each size of compact plate regions, the procedure of compact plate region extraction scans the input image seven times when the extraction algorithm is executed. In this study, we propose a one-pass algorithm for the extraction of compact plate regions, as illustrated in Fig. 41. During one-pass scanning, type 1 suppression is performed in the step of run determination described in Alg. 2 and type 4 suppression is performed in connected component extraction. Smearing and type 2 and 3 suppressions are performed based on the run information without modifying the image. The one-pass extraction algorithm is described in detail in Alg. 3, where $dist(Run_i, Run_j)$ implies the distance from the end of Run_i to the start of Run_j. Figure 42 shows an example in which a candidate of the compact plate region is extracted explicitly by the application of the algorithm.

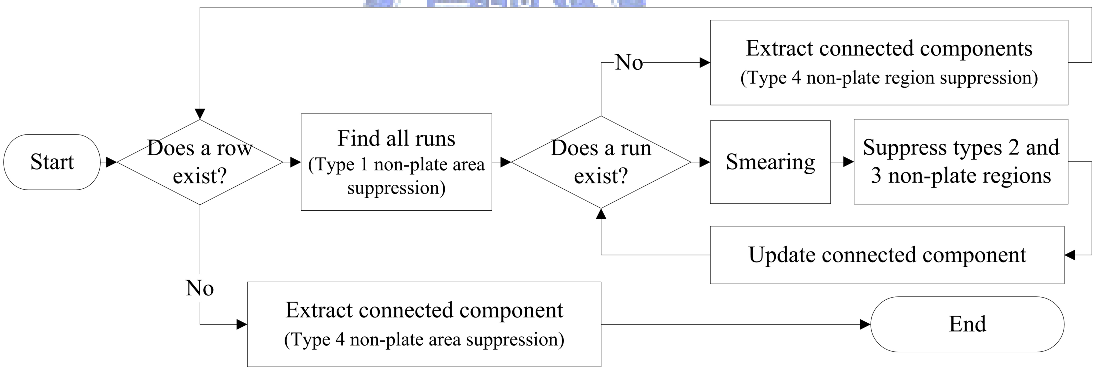

Start
Does a row exist?
Find all runs
(Type 1 non-plate area suppression)
Does a run exist?
No
Extract connected components
(Type 4 non-plate region suppression)
Smearing
Suppress types 2 and 3 non-plate regions
Update connected component
No
Extract connected component
(Type 4 non-plate area suppression)
End

Figure 41: Schematic diagram of the one-pass algorithm for extracting compact plate regions.

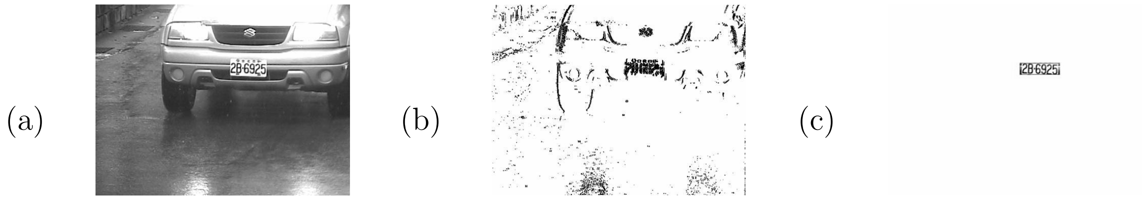


Figure 42: An example in which the one-pass plate extraction algorithm is applied: (a) Original image, (b) G_p of plate candidates, and (c) detected compact plate region.

```
Data: $G_p$
Result: candidates of compact plate regions
foreach row in the image do
    FindAllRuns() ;
    foreach $Run_i$ do
        j = i - 1;
        k = i + 1;
        if $dist(Run_i, Run_j) \leq \lambda_s$ then group $Run_i$ and $Run_j$ // Smearing;
        if $length(Run_i) > W_p$ then continue to process next run // Type 1
        suppression;
        if $length(Run_i) < \lambda_s$ and $dist(Run_j, Run_i) > W_p$ and
        $dist(Run_i, Run_k) > W_p$ then
            continue to process next run // Type 3 suppression;
        end
        // Type 4 suppression;
        update corresponding connected components information when $Run_i$ is
        connected with runs of the previous row
    end
    extract connected components whose sizes are larger than $(H_p \times W_p)/4$ and are
    not updated at previous procedures;
end
```

Algorithm 3: One-pass scanning algorithm for extracting compact plate regions.

5.2.3 Plate Verification

After the plate regions are extracted based on low-level gradient features, background patterns such as bumpers or texture floors may not be rejected. The properties of a compact plate region are different from those of a non-plate region. A significant property is that a sequence of character patterns will exist in compact plate regions but not in most of the non-plate regions. However, it is difficult to determine all heuristic rules for all variations in different types of plate and non-plate regions. To reject non-plates quickly and robustly, we adopt a cascade classifier [125] and a learning mechanism, AdaBoost [126], to determine the minimum dominant features from our feature pool (Sec. 5.2.3) and the statistical weak learners, $h_i(x)$ (Sec. 5.2.3), using the samples of compact plate regions and non-plate ones. As shown in Fig. 43, the final rejecter is composed of all selected weak learners, $h_i(x)$, combined with the weight, w_i, and the threshold, θ. As shown in Fig. 44, plate verification is performed by cascading rejecters with increasing complexity to reject non-plate regions in the shortest possible time. The verification procedure is described in detail in the following subsections.

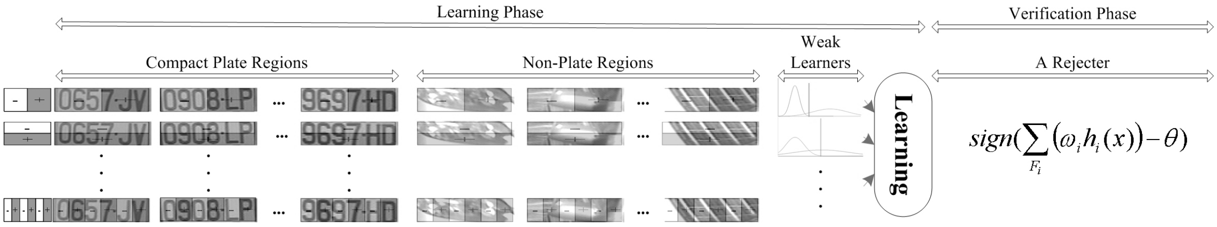


Figure 43: Schematic diagram of the learning and verification steps.

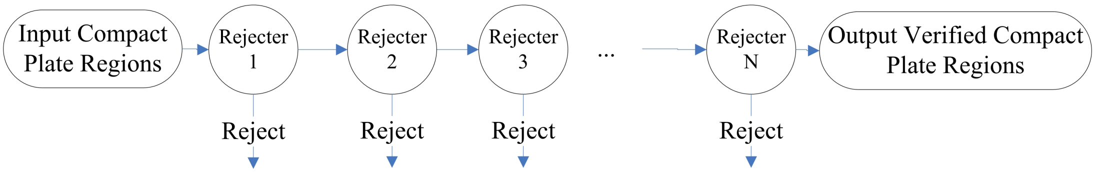


Figure 44: Schematic diagram of the plate verification procedure.

Feature Extraction

To verify the plates correctly, instead of using raw pixel values, features are extracted to minimize the within-class variability while maximizing the between-class variability. In response to significant characteristics originating from variations in the plate characters, our feature pool is based on the Haar-like features in Oren [131] and the fast computation schemes proposed by Viola [125]. To represent all discriminating features, we extract over-complete Haar-like features in all sizes. More specifically, the values of twenty feature prototypes shown in Fig. 45, which include four edge features, eight line features, two center-surround features, and six plate character-line features, are extracted to represent compact plate regions and non-plate ones. Their prototypes are classified into two types: upright and skewed. As discussed later, our features can be computed at any position and scale in the same constant time; only one scanning procedure is required.

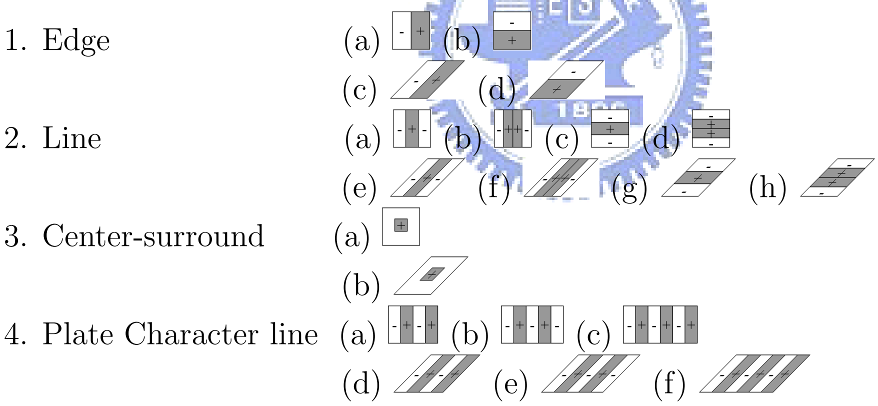


Figure 45: Feature prototypes of upright and skewed Haar-like features used in our system.

(i) Upright Haar-like features: Haar-like features, which represent regional brightness differences, are reminiscent of Haar basis functions. As shown in the left part of Fig. 43, Haar-like features are computed by subtracting the weighted sum of the pixels covered by the dark rectangle from that covered by the white rectangle. For a normalized input region of 32×8, the total number of upright Haar-like features extracted in this stage is $175,256$. In general, the total number of upright feature prototypes with a size of $w \times h$ in an image of size $W \times H$ could be defined as

$$N = XY\left(W + 1 - w\frac{X+1}{2}\right)\left(H + 1 - h\frac{Y+1}{2}\right), \tag{5.1}$$

where $X = \left\lfloor \frac{W}{w} \right\rfloor$ and $Y = \left\lfloor \frac{H}{h} \right\rfloor$ denote the maximum scaling factors in the horizontal and vertical directions, respectively. This formula is explained below.

When sliding along the row, the feature size could be $(w \times 1)$, $(2w \times 1)$, $\cdots$, *or* $(Xw \times 1)$. The feature size can be measured from

$$\begin{aligned} &(W - w + 1) + (W - 2w + 1) + \cdots + (W - Xw + 1) \\ &= X\left(W + 1 - w \times \tfrac{X+1}{2}\right). \end{aligned}$$

When sliding along the column, the feature size could be $(1 \times h)$, $(1 \times 2h)$, $\cdots$, *or* $(1 \times Yh)$. The feature size can be measured from

$$\begin{aligned} &(H - h + 1) + (H - 2h + 1) + \cdots + (H - Yh + 1) \\ &= Y\left(H + 1 - h \times \tfrac{Y+1}{2}\right). \end{aligned}$$

(ii) Skewed Haar-like Features: To represent skewed plate characters, we extend the basic Haar-like features by an efficient set of horizontal skewed features. The number of skewed feature types with a size of $w \times h$ could be observed as an upright feature type with a size of $(w + h) \times h$.

(iii)Fast Computation Schemes: One problem of these features is that the computational effort increases when a window sweeps the entire input region at various scales. To reduce duplicate summation operations, each used feature can be computed by the acceleration tables, summed area table (SAT) [125] [132], or skewed summed area table (SSAT), as shown in Fig. 46. In the SAT, the value at location (x, y) contains the sum of the pixels above and to the left of (x, y):

$$SAT(x, y) = \sum_{x' \leq x; y' \leq y} i(x', y'), \tag{5.2}$$

where $i(x, y)$ is the value of the input image.

In the SSAT, the value at location (x, y) of SSAT contains the sum of the pixels of the indefinite quadrangle that begins at the rightmost corner at (x, y) and extends to the boundaries of the image:

$$SSAT(x, y) = \sum_{x' \leq x; y' \leq y; x' \leq x-(y-y')} i(x', y'), \tag{5.3}$$

where $i(x, y)$ is the value of the input image. In this study, we propose to calculate the values of SSAT in one-pass by the following formula:

$$SSAT_{(x,y)} = SSAT_{(x-1,y)} + SSAT_{(x-1,y-1)} + i(x, y) - SSAT_{(x-2,y-1)}$$

with

$$SSAT_{(-1,y)} = SSAT_{(-2,y)} = SSAT_{(x,-1)} = 0$$

Then, each value of the summation in a quadrilateral region can be measured by two additions and one subtraction independent of the position or scale. For example, the gray region denoted by D in Fig. 46 can be measured by

$$D = 4 - (2 + 3) + 1, \tag{5.4}$$

where $1 \cdots 4$ are the values in the SAT or SSAT.

Thereafter, all values of the Haar-like features can be measured at constant time from the acceleration tables. In the system, these acceleration tables are also created in the one-pass extraction algorithm of the compact plate regions without scanning the image again in subsequent procedures.

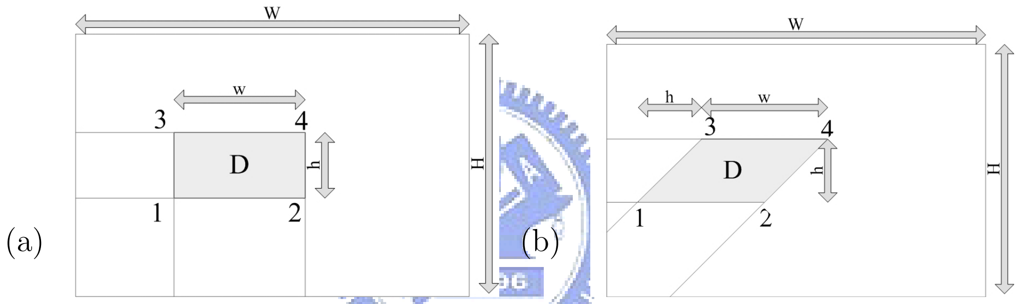


Figure 46: Acceleration tables: (a) SAT and (b) SSAT.

Feature Dimension Reduction

More than one hundred thousand quadrilateral features are associated with each region of plate candidates. Although each feature could be computed efficiently, the hypothesis space was still very large and therefore it was difficult to satisfy the real-time requirement of the system. In this study, we attempt to determine a dominant subset of the exhaustive Haar-like features that reject the non-plates very effectively. For this purpose, we use boosting techniques that could help to improve the accuracy of any given learning algorithm. There are many variations in basic boosting. AdaBoost (adaptive boosting) [126], which is the most popular boosting technique, allows the designer to continue adding weak learners until some desired low training error has been achieved.

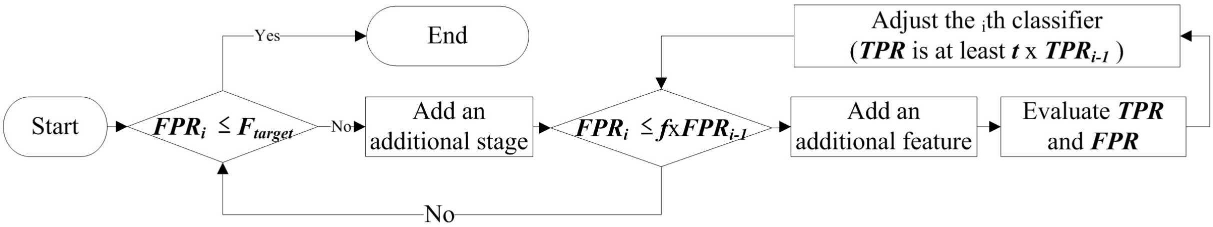


Figure 47: Schematic diagram of the learning phase: f denotes the maximum acceptable FPR per stage; t, the minimum acceptable TPR per stage; F_{target}, the overall false positive rate; FPR_i, the false positive rate at stage i; and TPR_i, the true positive rate of stages i.

AdaBoost: The main concept of AdaBoost is to assign a weight to each sample of the learning set. In the beginning, all weights are equal; however, in every iteration, the weak learner returns a hypothesis, and the weights of all samples classified incorrectly by that hypothesis are increased. The weak learner is then forced to focus on the difficult samples of the learning set. The final hypothesis is a combination of the hypotheses of all iterations, namely, a weighted majority vote, where hypotheses with a lower classification error have higher weights. The AdaBoost algorithm is summarized in Alg. 4.

Learning Procedure: In the learning phase, the feature dimension in a rejecter and the number of rejecters are automatically determined by the AdaBoost learning algorithm and the criteria satisfying the required TPR and FPR. As shown in the loop procedure in the right side of Fig. 47, the learning procedure constructs cascading rejecters by searching the set of possible features and returns the feature with the lowest classification error iteratively. For each feature, the weak learner determines the optimal classification function that has the minimum number of misclassified samples. The learner is termed weak because we do not expect to classify the training data effectively (i.e, it may only classify the training data correctly 51% of the time). Therefore, a weak classifier, $h_j(x)$, consists of a feature, $f_j(x)$, in which $h_j(x)$ is defined as a threshold-type function with a boolean-valued output that indicates whether x is positive or negative. Here, x is a 32×8 image of the normalized regions of the plate candidates. The formal expression of a threshold-type function is $h_j(x) = sign[f_j(x) - b]$, where b is the threshold. Learning samples are then reweighted in order to emphasize the false negatives and positives by previous learned classifiers. The final rejecter takes the weighted combination of weak classifiers followed by a threshold. In each iteration of boosting: (i) evaluate each quadrilateral feature (a weak classifier) of each input training sample, (ii) select the best threshold for each weak classifier, (iii) select the best weak classifier (feature) and the threshold, (iv) re-weight the samples, and (v) determine the number of rejecters when the overall FPR is satisfied, as shown in the loop procedure in the left part of Fig. 47.

```
Data: Given samples $(x_1, y_1), \cdots, (x_n, y_n)$, where $y_i = 0, 1$ for negative and
      positive samples, respectively.
Result: The final classifier $H(x)$
Initialize weights $w_{1,i} = 1/2m, 1/2l$ for $y_i = 0, 1$ respectively, where $m$ and $l$ are the
number of negatives and positives respectively.
for $t = 1, \ldots, T$ do
    1. Train one weak classifier $h_j$ for each feature $j$ using $w_t$, with error
    $\epsilon_j = \sum_i w_i \, |h_j(x_i) - y_i|$;
    2. Choose $h_t(x) = argmin(\epsilon_k)$, where $k \in features$ (i.e., the hypothesis with
    the lowest error $\epsilon_j$). Let $\epsilon_t = \epsilon_k$;
    3. Update the weights: $w_{t+1,i} = w_{t,i}\beta^{1-e_i}$, where $e_i = 0$ if sample $x_i$ is classified
    correctly, $e_j = 1$ otherwise, and $\beta_t = \epsilon_t/(1 - \epsilon_t)$;
    4. Normalize $w_{t+1,i} \leftarrow \frac{w_{t+1,i}}{\sum_{j=1}^{n} w_{t+1,j}}$ so that $w_{t+1}$ is a distribution.
end
Form the final classifier as
```

$$H(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} w_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} w_t \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

Algorithm 4: Algorithm of AdaBoost

5.2.4 Multiscale Implementation

Scale selection is a fundamental problem in computer vision and a key bottleneck for object detection algorithms. To increase the speed of detecting variant plate sizes, we scale our detector in the parameter space (W_p and H_p) without spending computational effort on scaling the input image, as in the case of other pyramid searching techniques. Moreover, without normalizing the input plate candidates, the quadrilateral features used for plate verification are measured in proportion to the extracted plate regions because these features are rather invariant to the variations in the plate size and shape. As shown in Fig. 48, the left image is detected with $W_p = 60$ pixels and $H_p = 15$ pixels, while the right image is detected with $W_p = 120$ pixels and $H_p = 30$ pixels. The two images in the middle show the results of one Haar-like feature measurements of the compact plate regions detected from the two input images.

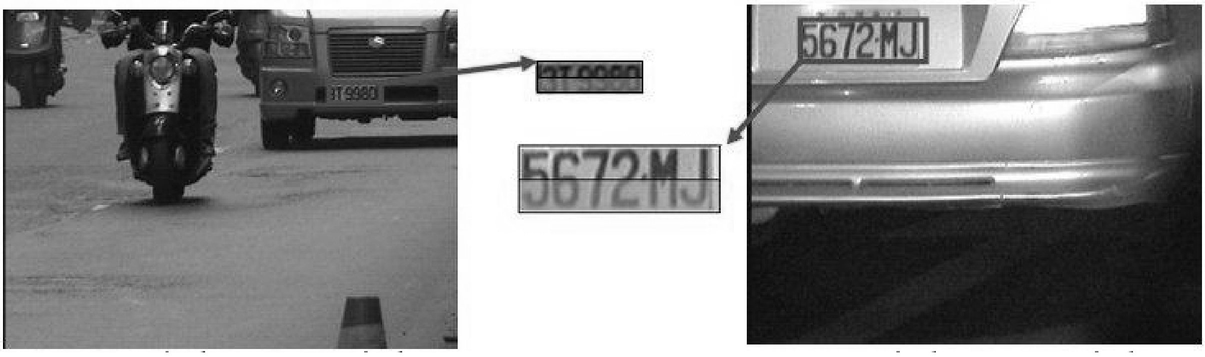


Figure 48: Schematic diagram of the extraction of one Haar-like feature (1×2) from two compact plate regions.

5.3 Cascading Segmentation and Recognition of Plate Characters

Three procedures are developed in a cascade to segment and recognize the plate characters. First, low-computational histogram features are adopted in the first two procedures to extract promising plate character candidates. Moreover, to segment characters effectively with different variations, such as illumination changing on a plate or the variant numbers of plate characters, a new segmentation approach is proposed based on peak-valley analysis in conjunction with heuristics that determine the potential segments. In the third procedure, heavy-computational statistical Haar-like features are selected to reject non-character regions robustly. The remaining character regions are recognized by an OCR.

5.3.1 Plate Character Segmentation

A compact plate region may contain two types of plate characters: distinct and indistinct. A distinct character is a plate character that can be separated from the plate region on the basis of thresholding from gray values, while an indistinct character is connected with other ones. Four compact plate regions are shown in Fig. 49(a). The distinct characters extracted as connected components after a certain values of thresholding are exhibited in Fig. 49(b), and the indistinct characters are presented in Fig. 49(c). In most cases, the plate characters would be distinct, which will be extracted first to avoid the use of additional splitting procedures. In indistinct character regions, the region candidates of single characters are determined by projection analysis. For measuring the projection effectively, we use an additional acceleration table, PSAT, to avoid the recomputation of the summation operations during the projection. The value at location (x, y) in the

PSAT is the sum of the pixels above (x, y). Each value in the projection histogram could be calculated by one subtraction operation.

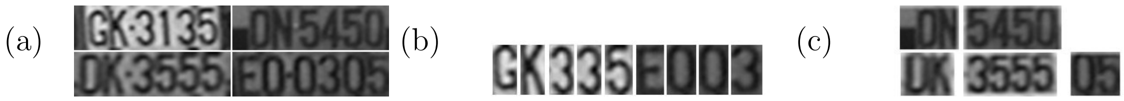


Figure 49: Patterns of (a) compact plate regions, (b) distinct characters, and (c) indistinct characters.

Searching for natural segmentation points

It is generally difficult to determine optimum thresholds during histogram segmentation or projection segmentation because the difference between the color of the characters and the background is usually not evident. For example, a compact plate region with indistinct colors between the characters and the background is shown in Fig. 50(a), whose histogram is displayed in Fig. 50(b), where the white line represents the threshold value detected by Otsu's method. The result obtained after thresholding by Otsu's method is presented in Fig. 50(c), in which only one distinct character, "7," could be detected after connected-component extraction. Other characters such as "5" or "8" would be extracted by other thresholds.

With the assumption that each character is composed of pixels with similar colors, the potential thresholds to separate distinct characters from others would occur in a valley of the histogram for the plates with light-coloured characters or in a peak for the other plates with dark-coloured characters. After thresholding, the connected components are selected as candidates of distinct characters when their regions satisfy plate characteristic 4, for example, the width to height ratio of 0.5. To segment the indistinct characters in the remaining regions, we propose to use the projection profile to detect all promising segment points, which would exist in the local peaks or valleys of the projection profile according to the change from the plate characters to the background.

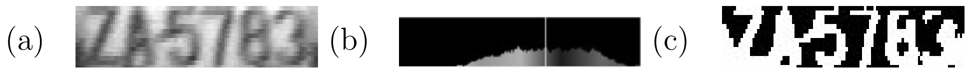


Figure 50: Inevident difference between the colors of the characters and the background: (a) compact plate region, (b) histogram, and (c) binary representation after thresholding by Otsu's method.

Certain local peaks (white lines) and valleys (gray lines) of the histogram are shown in Fig. 51(b), and three results obtained after thresholding with different peak values are shown in Fig. 51(c)-(e). The distinct character, "7," would be extracted from the connected components of Fig. 51(c) and the others, "5" and "8," could be extracted from Fig. 51(d). All the extracted distinct characters "5," "7," and "8," are displayed in Fig. 51(f). The character representation after thresholding with a lower threshold would be more fragmented than that shown in Fig. 51(e); this shows that other plate characters marked as indistinct characters shown in Fig. 51(g) could not be extracted effectively at this moment. The lower part of Fig. 51(g) shows the projection profile of the upper part and the corresponding peaks and valleys. Using the peak or valley values for segmentation, we could separate all single characters from the indistinct characters, as shown in Fig. 51(h).

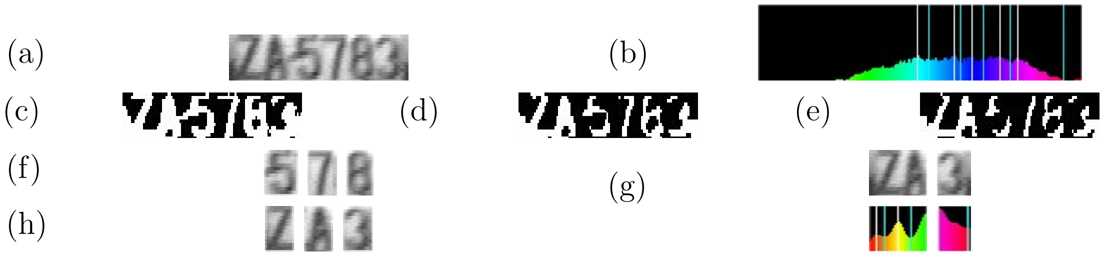


Figure 51: One segmentation result: (a) Original compact plate region, (b) histogram of image (a) where peaks and valleys are indicated by white and gray lines, respectively; (c)-(e) three thresholding results with three different peak values-from low to high; (f) indistinct characters detected by histogram segmentation; (g) distinct characters detected by histogram segmentation; (h) projection profile of (f); (i) single characters detected by projection segmentation.

Searching for special segmentation points

Some natural segmentation points are highly concentrated or dispersed according to the degree of noise or different combinations of characters. The high concentration of segmentation point appears usually in the middle of thin connected characters such as “H,” “N,” or “U.” To reduce the number of character candidates, we select only the regions that satisfy plate characteristic 4, for example, regions with a width-to-height ratio of 0.5. The segmentation points are dispersed when the character regions are not segmented exactly because these points would not exist in the boundaries of characters such as “T” or “L.” For example, a valley point would be detected at the middle of the “T” projection or at the left of the “L” projection. To improve the performance of projection segmentation, a segmentation point also locate in the middle of two detected segmentation points whose distance is greater than λ_s.

5.3.2 Peak-Valley Analysis

In a histogram or projection profile, a peak is a point with the value of the maximum height between two local minima. In order to consider a point as a peak, the value must be the local maximum and at least λ greater than the local minimum. Similarly, a valley point is defined as a point with the minimum value that is less than λ between two local maxima.

To detect all peaks and valleys, two decision modes, which are termed peak decision mode and valley decision mode, are designed to determine a peak and a valley, respectively. Moreover, two cascading procedures are proposed to detect all promising peaks and valleys in one-pass scanning. The first procedure is termed mode initialization as shown in Fig. 52, where $f(i)$ denotes the value of the profile at position i. The mode of peak finding or valley finding is determined according to the initial progressive increase or decrease in profile values. The second procedure is termed peak-valley decision as shown in Fig. 53. A point is selected as a peak when its value is higher than that of its neighbor and the previous valley by λ in the peak finding mode, while a point is selected as a valley when its value is less than that of its neighbor and the previous peak by λ in the valley finding mode. The peak and valley finding modes are performed alternately in order to determine all peaks and valleys.

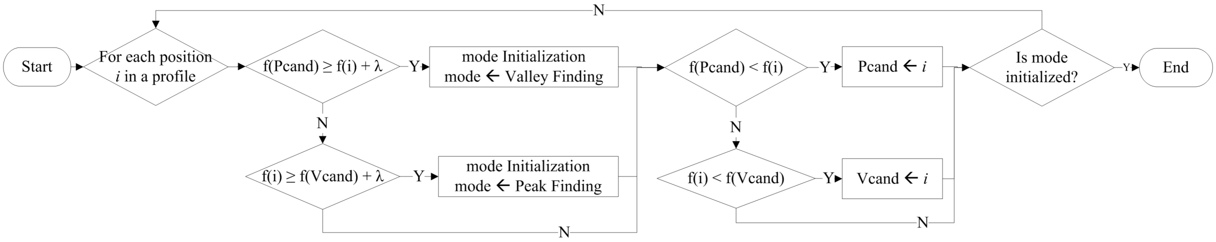


Figure 52: Schematic diagram of the mode initialization step.

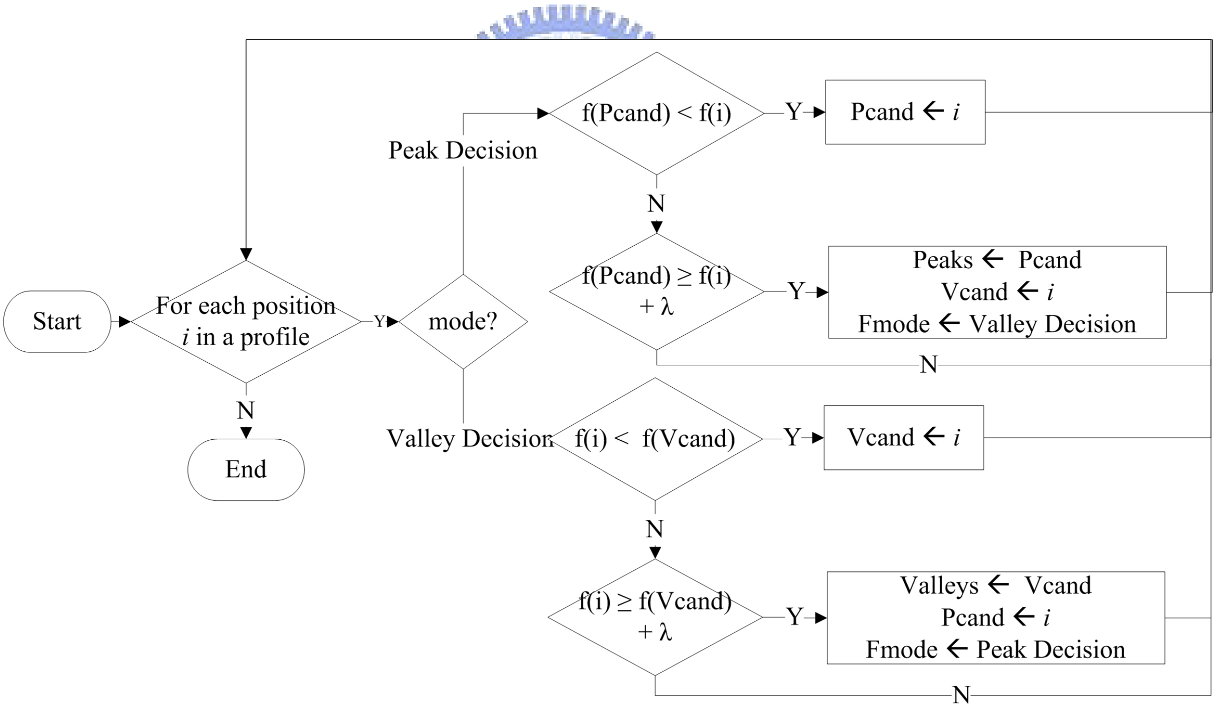


Figure 53: Schematic diagram of the peak-valley decision step.

5.3.3 Character Verification

The variations in characters and non-characters extracted from the preceding segmentation procedures are automatically learned using the same scenario of plate verification as that described in section 5.2.3. The classification phase of character verification is depicted in Fig. 54. During the feature extraction step, the SAT and SSAT are reused to calculate the Haar-like features for character rejecters. As compared with the plate verification, one more procedure of the OCR is adopted to recognize each character explicitly. We have investigated the development of an OCR for plate characters with different appearances [133]. In the OCR, the feature extraction step is first adopted to transform the character images into feature vectors.

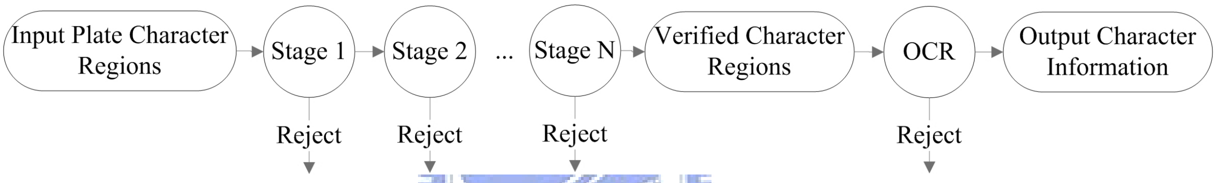

Input Plate Character Regions
Stage 1
Stage 2
...
Stage N
Verified Character Regions
OCR
Output Character Information
Reject
Reject
Reject
Reject

Figure 54: Schematic diagram of the character verification step.

OCR feature Extraction

Three robust character features are extracted to represent the characters: (1) contour-crossing counts (CCs), (2) directional counts (DCs), and (3) peripheral background area (PBA).

For the CC feature, each subimage is segmented non-uniformly into 8 strips in both the horizontal and vertical directions. These strips have the same number of character pixels. In order to increase the speed of the extraction, four scan lines are selected to extract the features in each strip. One value of the contour-crossing count, CC_k, is the number of strokes intersecting each scan line k, as indicated by the dashed line in Fig. 55(a). The CC feature dimension of an input sample is $16(= 8 \times 2)$. Each CC feature value f_i in strip i is defined as follows.

$$f_i = \sum_{k=1}^{4} CC_{ik} \tag{5.6}$$

The DC feature represents the number of contour points in four main directions. As shown in Fig. 55(b), we divide the directions into four groups: G1, G2, G3, and G4, which correspond to the horizontal, vertical, diagonal, and inverse diagonal strokes, respectively. The patterns are only measured from the pixels located at the boundaries of the strokes. The angle of a stroke, ranging from 0^o to 180^o, is determined by the Sobel operator. The DC feature dimension of an input sample is estimated as $64(4 \times 4 \times 4)$ by splitting the input character image into 4×4 sub-blocks.

The PBA feature is the length of line segments that begin from the boundary of the image to the character contour. The input character image is first segmented into 8 strips in both the horizontal and vertical directions by the method described in the above processing technique. In each strip, two feature values are measured from the lengths between the strip boundary and the nearest character contour. Each PBA faeture value is then divided by the length of the strip for normalization. The PBA feature dimension of an input sample is $32(= 8 \times 2 \times 2)$. The final feature dimension of each character is $112(16 + 64 + 32)$.

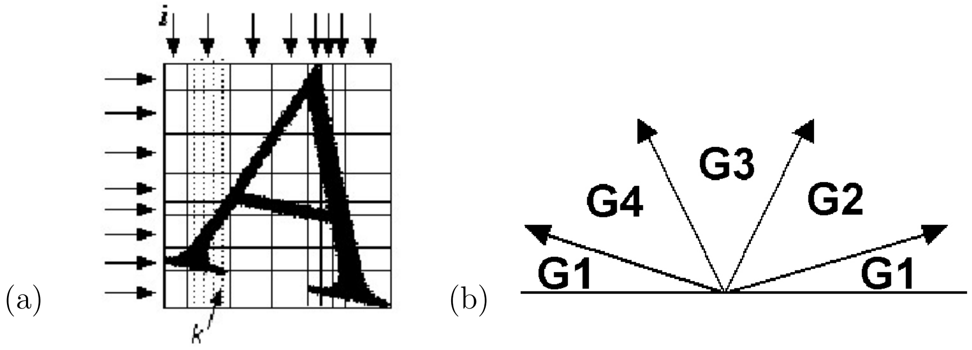


Figure 55: Feature prototypes of OCR: (a) CC and (b) DC.

OCR Classification

The classification step is realized by a support vector machine (SVM) [134] to classify each plate character. The SVM method searches for a linear separating hyperplane as a linear combination of support vectors (in addition to a constant). The distance from all the support vectors to the hyperplane is 1. The SVM implementation used in this study is LibSVM [135]. In order to determine the best setting for the SVM, we test different settings for the parameters. The radial basis function (RBF) is the final used kernel given to LibSVM for the OCR. The highest OCR accuracy (with only one character error on the training set) is obtained by setting gamma (related to the RBF radius) to 0.0078125.

Non-plate Character Suppression

After the extraction and recognition of the plate characters, several characters would be suppressed when they exist in the subregions of the real characters such as "L" in the region of "C," "7" in the region of "Z," or "F" in the region of "E." For the first example, if "L" is another recognition candidate in the region of the recognized character "C," it should be suppressed according to the geometrical similarity between the lower half of the plate character "C" and the plate character "L." In this study, these heuristic rules are collected and generated manually in the training phase from the recognition candidates and the ground truth data. In the region of character "Z", for example, if over 30% sub-regions, are recognized as "7," the recognition result should be supressed.

5.4 Experimental Results and Discussion

This section describes the results of our cascade plate recognition system, including the detection and the final recognition results. The discussion includes details on a comparison with other publishing systems as well as the results of a large real-surveillance testing set. The system proposed in this study was implemented in C language and tested on a standard Intel Pentium 4, 3 GHz personal computer. To recognize the plates more effectively, we reduced the height of the test images to half or directly captured the half-height images from the capturing devices without performing the deinterlace procedure. This indicates that the width-to-height ratio of the plate character is set to 1 in the experiments.

5.4.1 Plate Recognition System Performance

The plate training set consisted of $8,520$ manually selected plates scaled to a base resolution of 32×8 pixels. The plates were extracted from images captured in actual visual surveillance environments. The training and test images showed different plate types, sizes, locations, or illuminations. The character training set consisted of $4,786$ manually selected characters scaled to a base resolution of 16×16 pixels. The characters were extracted from the plate training set.

The test images are arranged into three sets according to the captured image size to demonstrate the performance, computational speed, and adaptation. Each test set is mainly composed of three kinds of capturing environments, which are a portal, roadside, and pavement background. These datasets are used to show different types of lighting or environment on a wet day, cloudy day, and sunny day. The datasets also shows tolerance of our plate recognition system to camera's pan and tilt motion.

In these images, the plate candidates extracted by compact plate region extraction are shown with blue rectangles, while the final recognized plates are indicated by red rectangles. Figure 56 shows the experimental results obtained with different headlights on a wet day, while Figs.57 and 58 show the results obtained with different types of environmental lighting on cloudy and sunny days, respectively. The results of images captured in the environment of the pavement background with texture patterns are presented in Fig. 59. Figure 60 demonstrates the ability to recognize the frames from a capturing device installed in a car. Although the capturing device placement (pan/tilt) was changed (approximately 20^o), as shown in Fig. 60, our plate recognition system still performed well without the application of any skew correction procedure.

The system performances are shown in Tables 5.1 and 5.2. In Table 5.1, plate heights are shown in the third column, while the testing numbers of images with and without plates are presented in the fourth column. The numbers of images with plates recognized are given in the fifth column. In dataset 3, for example, $11,598$ images (say, true positives) are recognized correctly in the $11,896$ plate images and 516 images (say, false positives) are recognized erroneously in the $52,000$ non-plate images. The TPR, defined as the ratio of true positives to the number of input images with plates, is shown in the sixth column; the FPR, defined as the ratio of false positives to the number of input images without plates, is shown in the seventh column. In dataset 3, for instance, the FPR is 0.01%, standing for one false alarm in ten thousand input images. The errors occur due to the existence of similar patterns such as the symbol “|” near the borders of the plate, or ambiguous characters such as “8” and “B” or “0” and “D.” Finally, the system accuracy rates shown in the last column are all greater than 96%, which are the weighted sum of the TPR and the FPR, as defined in section 4.3.1.

Table 5.1: Recognition results of our plate recognition system (I)

	Dim.	Max./Min.	Plate/Non-plate	True Positives/
		$Plate_{Height}$	Image #	False Positives
1	320×240	40/15	$3,756/19,000$	$3,575/369$
2	640×240	60/15	$4,153/22,000$	$4,028/357$
3	640×480	120/30	$11,896/52,000$	$11,598/516$

Table 5.2: Recognition results of our plate recognition system (II)

	Dim.	TPR (%)	FPR(%)	Accuracy (%)
1	320×240	95.176	0.019	96.638
2	640×240	96.990	0.016	97.695
3	640×480	97.495	0.01	98.251

5.4.2 Plate Detection Performance

The number of features used in the stages of the plate verifier using only upright Haar-like features and both upright and skewed Haar-like features are demonstrated in Fig. 61. The detection performance is shown in Table 5.3. The detection accuracy rates are greater than 99%. As shown in Figs. 56(c)-60(c), the algorithm for one-pass compact plate region extraction could detect the plate character regions without restrictions on the sizes, locations, and colors of the plate or the environmental lighting.

5.4.3 Comparison of Computational Speed

A comparison of the experimental computational speed and performance comparisons with those of two other studies is shown in Table 5.4. The term, "LPD," indicates that the system only dealt with plate detection, "LPS" presents that the system only dealt with plate character segmentation and character recognition, and "All" denotes the system including plate detection, plate character segmentation, and character recognition. Moreover, our system also outperforms commercial systems with respect to computational speeds. The VECON-VIS system [136] processes each image in less than 0.5 seconds while the SeeCar library [137] could response at 25msec per image but requiring at least 80 pixels

Table 5.3: Detection results of our plate detection module

	Dim.	Max./Min.	Plate/Non-plate	True Positives/	TPR (%)	FPR(%)
		$Plate_{Height}$	Candidate #	False Positives		
1	320×240	40/15	3,756/ 341,240	3,718/ 87,268	98.997	0.256
2	640×240	60/15	4,153/ 418,438	4,145/ 89,998	99.807	0.215
3	640×480	120/30	11,896/1,137,348	11,879/223,636	99.857	0.197

height per typical plate. When we applied our system to the dataset, thirty-two local vehicle images used in the SeeCar library [137], all plates in the images are recognized correctly. Other systems, such as Zamir's LPR engine [138] and CARMENR Parking License Plate Recognition Engine [139], would perform less than 20 frames per second. Above commercial systems all proclaim that the accuracy rates are over 95%. However, the details of settings and testing environments are unavailable for detail comparisons.

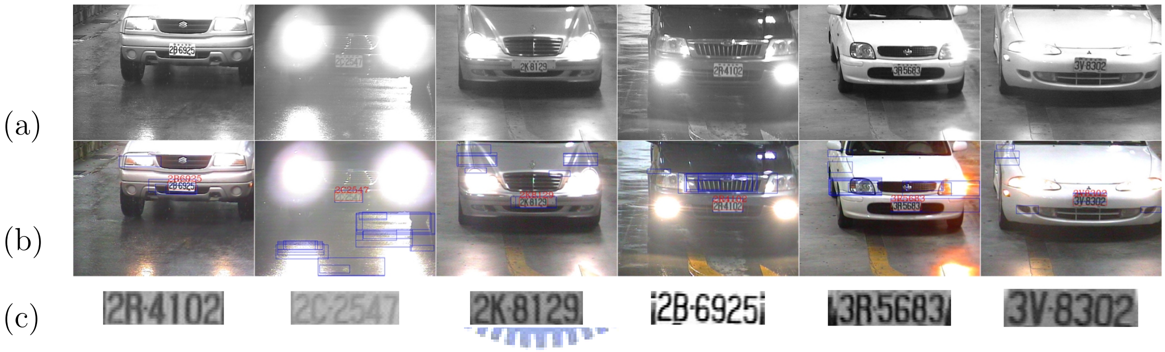


Figure 56: Examples of a variety of plates on a wet day: (a) input images, (b) recognized results shown in color images, and (c) enlarged recognized plates.

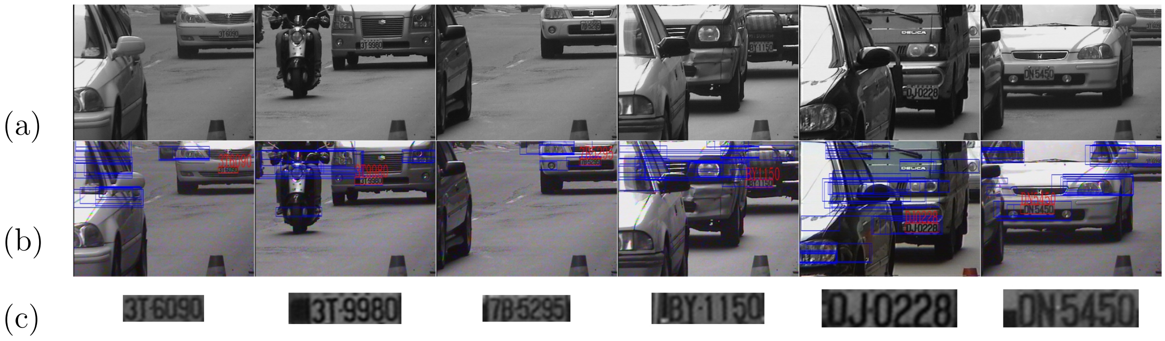


Figure 57: Examples of a variety of plates on a cloudy day: (a) input images, (b) recognized results shown in color images, and (c) enlarged recognized plates.

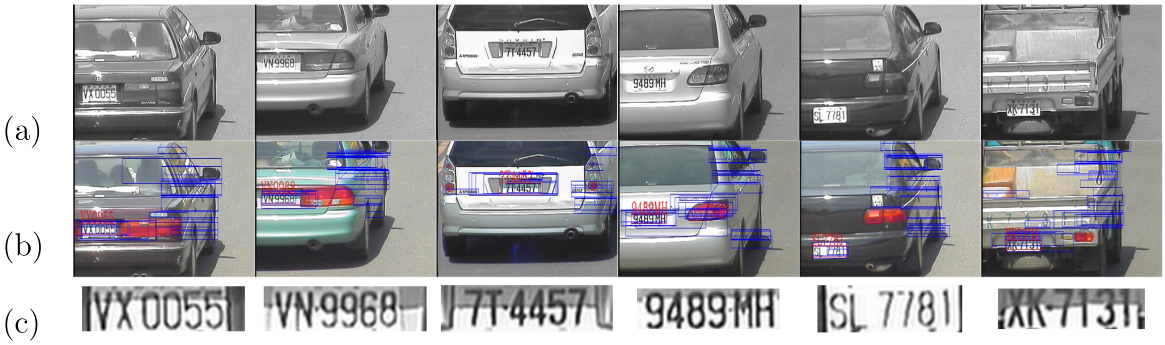


Figure 58: Examples of a variety of plates on a sunny day: (a) input images, (b) recognized results shown in color images, and (c) enlarged recognized plates.

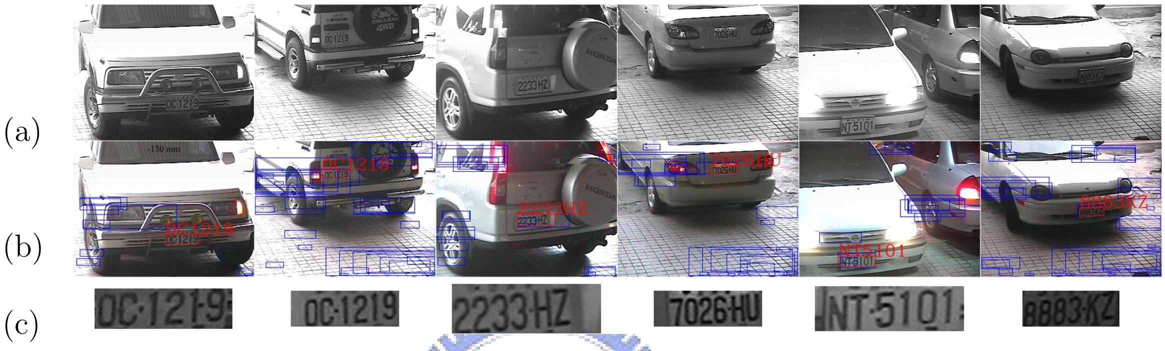


Figure 59: Examples of a variety of plates with the pavement background of texture patterns: (a) input images, (b) recognized results shown in color images, and (c) enlarged recognized plates.

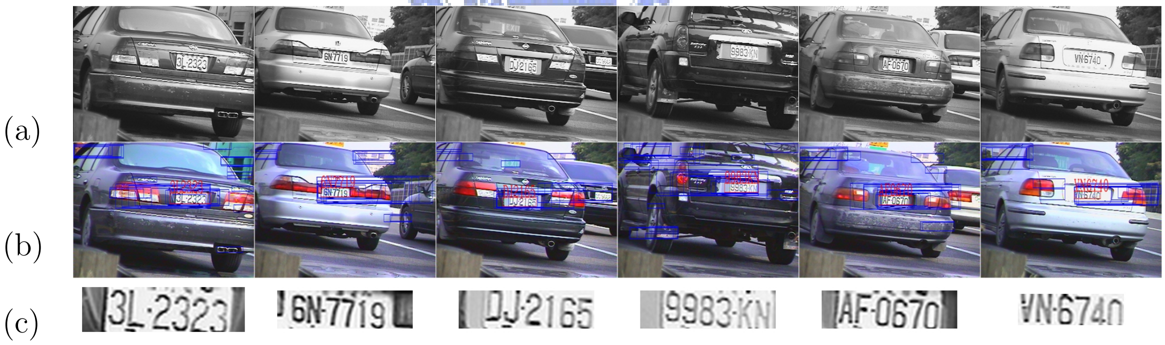


Figure 60: Examples of a variety of plates captured from a device installed in a car: (a) input images, (b) recognized results shown in color images, and (c) enlarged recognized plates.

Table 5.4: Comparison results of the computational speed and the accuracy rate for plate recognition

	Speed (fps)	Dim.	#	Accuracy(%)	Env.
Chang, *et al.*, ITS'04	˜2.5(LPD)	640 × 480	1061	˜93.7	P4 1.6 GHz
Amit, *et al.*, PAMI'04	˜0.3(LPS)	None	520	˜99	P3 1 GHz
Our system	˜40(All)	320 × 240	22,756	96.638	P4 3.0 GHz
Our system	˜36(All)	640 × 240	26,153	97.695	P4 3.0 GHz
Our system	˜38(All)	640 × 480	63,896	98.251	P4 3.0 GHz

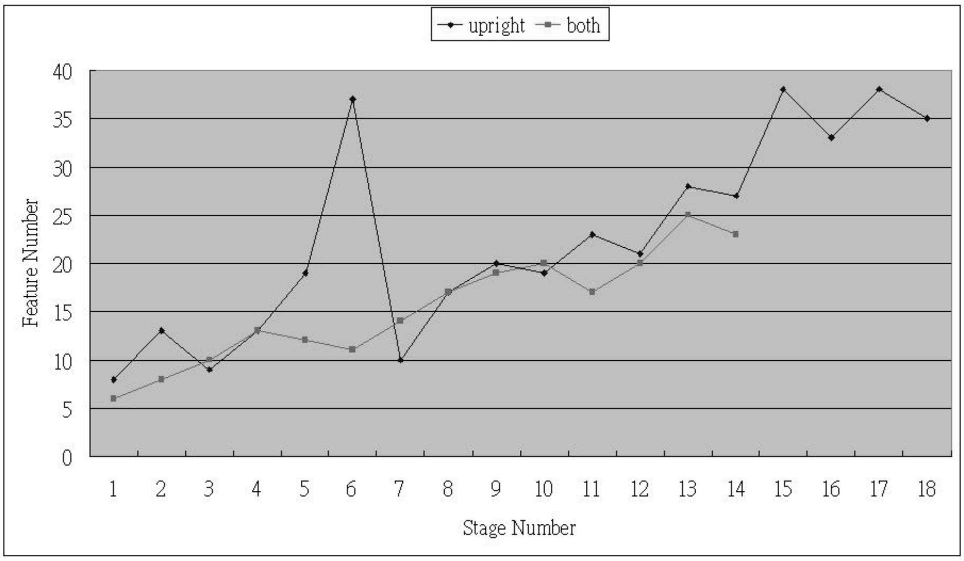


Figure 61: Comparison of feature numbers between classifiers using only upright and both upright and skewed Haar-like features.

Chapter 6 Rejection Mechanisms for Video Sequences

In surveillance applications, search space reduction (SSR) is an essential element to efficient algorithms. In this study, spatial and temporal SSRs are integrated for license plate detection in video sequences; the plates could be extracted robustly and extremely fast. Our method started from spatial SSR by a bi-level one-pass plate extraction (BOPE) algorithm developed to extract plates accurately even in complicated situations. The search space was reduced by applying information from both spatial characteristics and temporal representation in consecutive frames. We proposed to exclude repeated patterns with the similar appearances in the same location of consecutive frames, which usually include stopped vehicles or regular backgrounds. For efficiency, repeated patterns were detected only on the results of BOPE, named spatiotemporal SSR, based on a block-based mechanism by estimating the tangent distance, which is invariant to the variations in positions, sizes, rotations, or brightness. To reduce the computational load, the repeated patterns and their measured invariant features could be retained for next estimation. In our experiments, the search space could be reduced up to 87.9% by the spatiotemporal SSR; all plates with the heights ranging from 24 to 40 are extracted correctly. The implementation of the system on a *3G Hz PC* run on average 76 frames per second. The high performance suggests that the real-time goal for surveillance applications could be accomplished.

6.1 Introduction

This study develops a bi-level one-pass plate extraction (BOPE) to detect plate candidates robustly and accurately even in complicated situations, where vehicles and motorcycles may appear simultaneously in parking spaces. In these environments, plates might be close or connected with backgrounds in the captured frame. Moreover, to save computational load, we propose to reject repeated patterns with similar appearances in the same location of consecutive frames instead of detecting moving objects directly. The patterns repeated in consecutive frames would be identified utilizing the similarity measurement between the extracted plate candidates of the previous frame and those of the current frame. In LPR system, usually stand for unnecessary ones, such as stopped vehicles or regular background patterns. By the procedure of repeated region detection, redundant computations could be avoided. However, it is time consuming to compare the candidates in the current frame to all of the candidates in the preceding frame. In Section 6.3, a block-based procedure is presented to verify repeated patterns.

To detect plate candidates rapidly and effectively, the mechanism of our spatiotemporal SSR is depicted in Fig. 62. Three major modules of the mechanism are summarized as follows:

- Plate candidates generation: This module generates a plate candidate map, G_p, by thresholding the vertical gradients of the original input.
- Bi-level one-pass plate extraction: This module extracts candidates of compact plate regions, $Cand_p$, by the BOPE algorithm (Section 6.2).
- Repeated region detection: This module detects repeated regions by the similarity measurement (Section 6.3) between $Cand_p$ at the current frame and ones at the preceding frame.

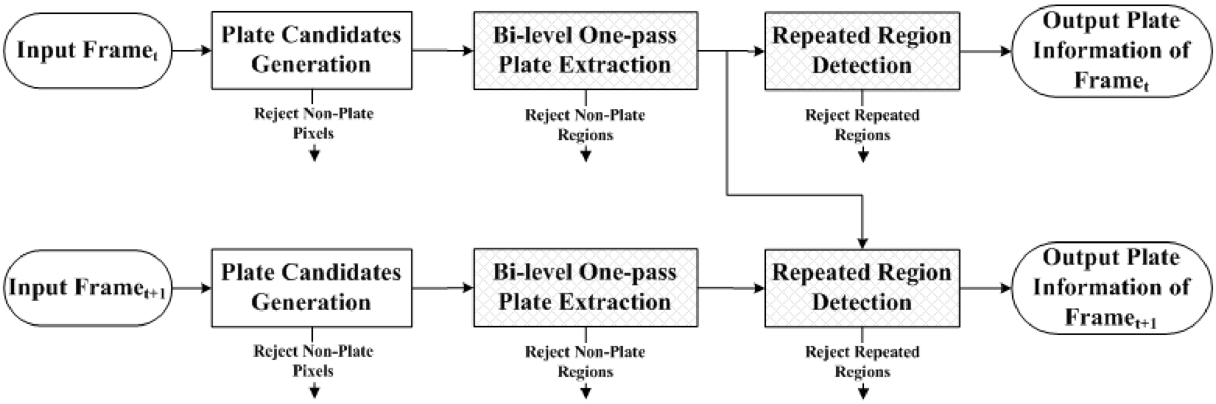

Input Frame$_t$
Plate Candidates Generation
Reject Non-Plate Pixels
Bi-level One-pass Plate Extraction
Reject Non-Plate Regions
Repeated Region Detection
Reject Repeated Regions
Output Plate Information of Frame$_t$
Input Frame$_{t+1}$
Plate Candidates Generation
Reject Non-Plate Pixels
Bi-level One-pass Plate Extraction
Reject Non-Plate Regions
Repeated Region Detection
Reject Repeated Regions
Output Plate Information of Frame$_{t+1}$

Figure 62: Overview of the rejection mechanisms for video sequences.

6.2 Bi-level One-pass Plate Extraction

A one-pass plate extraction (OPE) was designed to extract all plate candidates from the connected components after the removal of four types of plate runs and horizontal smearing. The steps of OPE are summarized in the following:

1. For an image input, calculate vertical gradients, G_p.
2. For each row in the G_p, find all plate runs; and remove the runs of type 1.
3. For each plate run, smear horizontally with other plate runs and remove the runs of types 2 and 3.
4. After processing all plate runs in each row, update information of connected components.
5. After processing al rows, extract connected compoents and remove the type 4 runs.

In more complex surveillance environments, the plate candidates might be close or connected with complex backgrounds, in which there are high gradient values. To further explain the problem, a frame of a parking lot is demonstrated in Fig. 63(a), while the corresponding plate candidate map, G_p, and the result after we remove the four types of plate runs are displayed in Figs. 63(b) and (c), respectively. Because of the trees in the background, OPE would extract many candidates with large regions as shown in Fig. 63(d). The large candidates would meaninglessly consume recognition time. For example, the largest candidate, region R1, is a non-plate one, which is still demanded to be rejected in the cascade plate recognition system. Moreover, we would have inclined plates connected with other patterns; it is difficult to extract compact plates accurately only by suppressing the four types of plate runs. Some of these plate candidates may have wide-margins and are not compact plate regions. For example, the characters in region R2 are connected with other patterns, where the rightmost character is connected with a label. Region R3 shows another example of a plate with wide-margin, where the plate is connected with the backgrounds. The extracted regions R1, R2, and R3 are enlarged in Figs. 63(f)-(h), respectively.

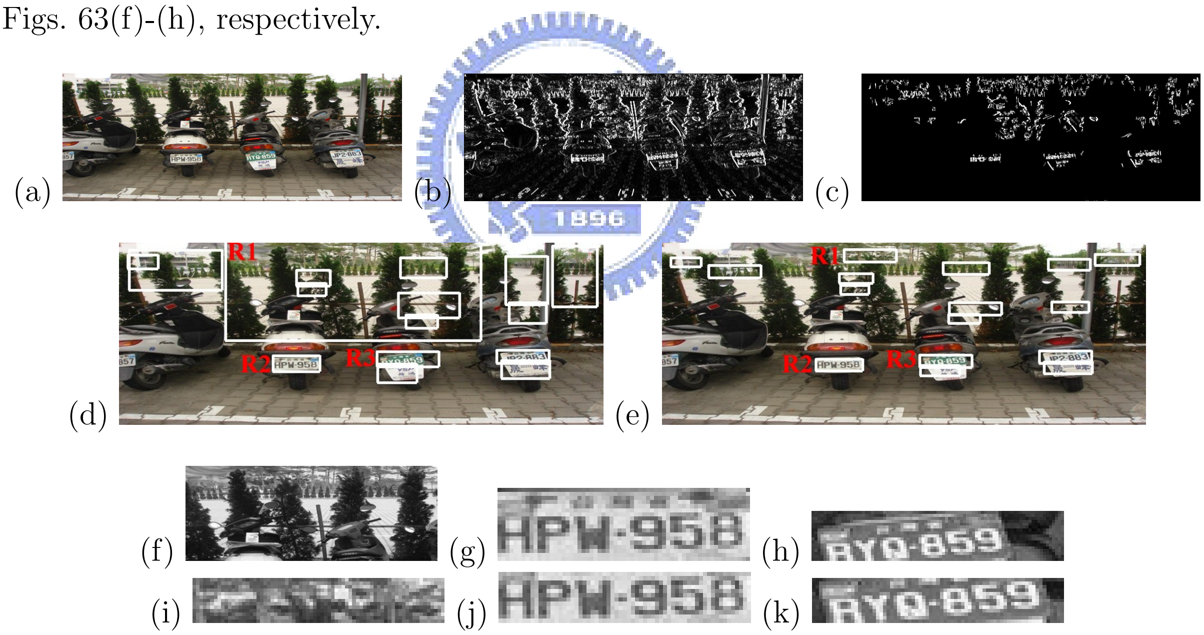


Figure 63: Results of the OPE and the BOPE algorithms in a complex environment of a parking lot: (a) the original frame with multiple motorcycle plates, (b) the corresponding plate candidate map, (c) the result after removing the four types of plate runs from (b), (d) the result after applying the OPE procedure in (b), (e) the result after applying the BOPE procedure in (b), (f)-(h) the extracted regions R1-R3 of (d), and (i)-(k) the extracted regions R1-R3 of (e).

The problems after we apply the OPE are summarized as follows:

- **Large plate candidates.** This type of candidates is caused by the complex environments.
- **Wide-margin plate candidates.** This type of candidates is caused by the complex environments or adornments closed to the plate characters.
- **Small plate candidates.** This type of candidates sometimes occurs when special characters, such as “1”, exist successively with a large gap.

In order to solve the problems, some procedures, such as character boundary detection, could be applied to reject the large background patterns or to remove the wide-margins, but they would be time-consuming. To extract all compact plate regions accurately without increasing the algorithm’s time complexity, we propose a bi-level one-pass plate extraction algorithm (BOPE) as shown in Algos. 5, 6, and 7.

In the first level as depicted in Algo. 5, plate candidates are extracted as connected components after we apply the smearing method and exclude the three types of plate runs 2, 3, and 4. These steps are similar to the OPE and shown in the loop “foreach” of Algo. 5. The sub-procedure of runs finding as shown in Algo. 6 is used to determine all horizontal plate runs while excluding the vertical plate runs 1 higher than H_p. In the algorithm, the numbers of pixels, which are grouped in plate candidates, are accumulated in a summed area table [132], CS, to speed up the second level procedure. Each value at location (x, y) of CS contains the sum of the pixels above and to the left of (x, y):

$$CS_{(x,y)} = \sum_{x' \leq x; y' \leq y} pr_{(x',y')} = CS_{(x-1,y)} + CS_{(x,y-1)} - CS_{(x-1,y-1)} + pr_{(x,y)} \tag{6.1}$$

with

$$CS_{(-1,y)} \quad = \quad CS_{(x,-1)} = 0,$$

where $pr(x,y)$ equals to one if the point (x,y) belongs to the plate run, or zero if it is not.

In the second level as depicted in Algo. 7, large or wide-margin plate candidates are adjusted to compact plate ones by comparing the probabilities of compactness for each sub-region inside the candidates. The probability could be estimated from the number of pixels in a grouped plate run, since each point in the plate run is identified as a part of the plate region. Then, for each sub-region, the probability of compactness P_c is defined as

$$P_c = \sum_x \sum_y pr(x,y), \tag{6.2}$$

where (x,y) is the point in the sub-region. The sub-region with the highest P_c is selected as the new compact plate candidate based on the assumption that the sum in the plate region would be larger than those of neighboring regions. Moreover, to avoid redundant calculations, P_c is measured in constant time from CS and re-defined as

$$P_c = CS_{(r,t)} - CS_{(l,t)} - CS_{(r,b)} + CS_{(l,b)}, \tag{6.3}$$

where l, t, r, b denote the left, top, right, and bottom boundaries of the sub-region, respectively.

As shown in Fig. 63(e), the three plates extracted by our BOPE are tighter than those extracted by the OPE. The extracted regions R1, R2, and R3 by the BOPE are shown in Figs. 63(i)-(k), where the regions R2 and R3 are compact plate regions.

```
Data: $G_p$
Result: candidates of compact plate regions
foreach row in the image do
    FindAllRuns() ;
    foreach $Run_i$ do
        if $\|Run_i, Run_j\| \leq \lambda_s$ then  group $Run_i$ and $Run_j$  // Smearing;
        if $\|Run_i\| > W_p$ then  continue to process next run  ;
        if $\|Run_i\| < \lambda_s$ and $\|Run_{i-1}, Run_i\| > W_p$ and $\|Run_i, Run_{i+1}\| > W_p$ then
            continue to process next run  ;
        end
        update information of corresponding connected components when $Run_i$ is
        connected with runs of the previous row  ;
    end
    for non-updated connected components do
        PlateReestimation() ;
        extract connected components whose sizes are larger than $(H_p \times W_p)/4$;
    end
end
```

Algorithm 5: BOPE algorithm for extracting compact plate regions.

```
Data: the $row_y$ of $G_p$
Result: plate and non-plate runs
Initialize an array Acc[width of $G_p$];
foreach column(x) in the $row_y$ do
    if $row_y(x) > 0$ then
        Acc[x] = Acc[x] + 1;
    else
        Acc[x] = 0;
    end
    if $row_y(x) > 0$ and $Acc[x] \leq H_p$ then
        update the plate run;
    else
        update the non-plate run;
    end
    $CS_{(x,y)} = CS_{(x-1,y)} + CS_{(x,y-1)} - CS_{(x-1,y-1)} + pr_{(x,y)}$ (Eq. 6.1)  ;
end
```

Algorithm 6: Algorithm for run determination, FindAllRuns.

```
Data: connected components and CS
Result: plate candidates
foreach connected component, $CC_i$ do
    if the size of $CC_i$ is larger than $(H_p \times W_p)$ then
        determine the compact plate candidate, which size is $(H_p \times W_p)$ with the largest $P_c$ (Eq. 6.3);
        reject the plate candidate when the gradient sum is lower than a threshold
    end
    else
        extend the left and right boundaries
    end
end
```

Algorithm 7: Algorithm for plate re-estimation, PlateReestimation.

6.3 Repeated Region Detection

After we extract plate candidates from BOPE, the candidates in the current frame may be classified as the same results with those in the preceding frame if they have similar appearances. The similarity can be defined as a distance measurement such as the Euclidean distance, assuming that the source and target patterns are represented as vectors. However, there are always variations between plate candidates even when they are the same targets because of variations of sensing or motion of surveillance targets. Measuring image distances is a complex task because these different image variabilities have to be considered.

One way to measure the similarity is to extract features invariant to certain sensing variations. To avoid the computational load of feature extraction, we propose an alternative way using "deformable" prototypes. During the matching process, each target pattern is deformed so as to best fit the incoming pattern. Mathematically, when the repeated patterns are subject to spatial transformations (rotations, scaling, changes of positions, etc.) or variable lighting, they span manifolds in the high-dimensional Euclidean space of pixel intensities; here, a manifold is an abstract mathematical space that locally resembles Euclidean space, but globally represents more complicated structure. With the examples of Fig. 64, the regions labeled as "R1" or "R2" in the right of Fig. 64(b) would be shifted or scaled so as to best match the previous appearances shown in the left. Hence, the applicable measure of distances is the one between the manifolds resulting from all possible transformations of the input pattern, rather than the Euclidean distance between the patterns.

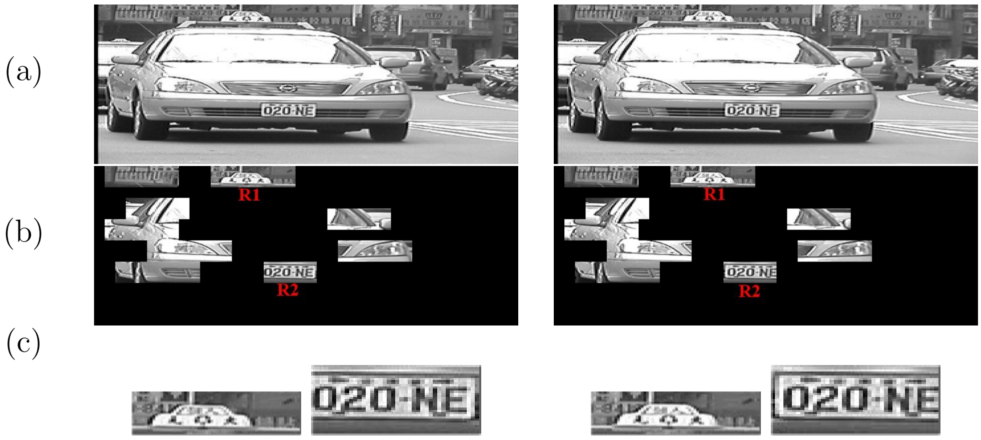


Figure 64: Examples of repeated regions in the same locations of two successive frames: (a) Left: the frame at time $t-1$, Right: the frame at time t, (b) the candidate regions in the two frames, (c) and the enlarged regions.

This idea has been formalized in the machine learning literature through the "tangent distance" between the manifold tangents at the location of the input patterns [140]. The tangent distance has been shown to be especially effective in pattern recognition applications [141] and is quite easy to understand and implement. Through the tangent distance, the various transformations of an input pattern could be represented by a parameterized multi-dimensional manifold, where each parameter accounts for one transformation. This idea can be extended to any desired transformation, such as brightness changes.

In the following, we first give an overview of the invariant distance measure, tangent distance (TD) in Section 6.3.1. The compensation for several transformations of the repeated patterns is discussed in Section 6.3.2. We show its integration into the license plate detection system in Section 6.3.3. Finally, an improvement mechanism for reducing computational load is presented in Section 6.3.4.

6.3.1 Overview of Tangent Distance

The tangent distance between two manifolds is defined as the Euclidean distance between their closest points. Let the function $s\left(P,\alpha\right)$ transform an input pattern P according to the parameter α. The function s is required to be differentiable with respect to α and P, and $s\left(P,0\right)$ is equal to P. The parameter α would be an m-dimensional parameters vector; each parameter stands for one control factor of the transformation, such as shifting or rotation. Then, the set of all transformed patterns is a manifold of at most m dimensions after orthogonalization [142]. For example, if the allowable transformations of input patterns are vertical shift and rotation, the space will be a two-dimensional manifold. Given two patterns P and E, the distance between the associated manifolds is defined as

$$D_{MD}\left(P,E\right)=min_{\alpha}\left\{\left\|s\left(P,\alpha\right)-s\left(E,\alpha\right)\right\|\right\} \tag{6.4}$$

where $\|x\|$ stands for Euclidean norm; the norm of x is the positive square root of the scalar product of x with itself. For another example, the pixel values of a 32 by 8 plate image can be represented as a 256-dimensional vector. Assuming that the set of allowable transformations is continuous, the set of all the transformed patterns is a surface in the 256-dimensional pixel space. Formally, when a pattern P is transformed (e.g. rotated) according to a transformation $s\left(P,\alpha\right)$, which is dependent on the parameter α (e.g. the angle of the rotation), the set of all the transformed patterns is

$$S_P=\left\{x|\exists\alpha\rightarrow x=s\left(P,\alpha\right)\right\}. \tag{6.5}$$

In general, these manifolds will not be linear and the global minimum-distance matching would be very expensive and unreliable. In practice, the minimum distance can be considered as the distance between the manifolds' tangent spaces. By approximating the surface of possible transforms of an input pattern by its tangent plane, the matching procedure is reduced to finding the shortest distance between two planes, that is, the tangent distance. The line, T_p, in Fig. 65 depicts the linear approximation of the curve, S_p, given by the Taylor expansion of s around $\alpha = 0$:

$$s\left(P,\alpha\right) = s\left(P,0\right) + \alpha \times \frac{\partial s\left(P,\alpha\right)}{\partial \alpha} + O\left(\alpha^2\right) \approx P + \alpha T_P, \tag{6.6}$$

where $\frac{\partial s(P,\alpha)}{\partial \alpha}$ is termed as the tangent vector, T_P.

The lines T_P and T_E would not intersect in three-dimensional space. The tangent distance between P and E is the shortest distance between them;

$$D_{TD}\left(P,E\right) = min_{\alpha}\left\{\|x-y\|\right\}, \tag{6.7}$$

where $x \in T_P$ and $y \in T_E$.

Figure 65 schematically represents the difference between the Euclidean distance, the full invariant distance (the minimum distance between manifolds), and the tangent distance in 3-dimensional space. In the figure, both the input patterns are deformable, but it is also possible to deform only one pattern (one-sided distance) for simplicity.

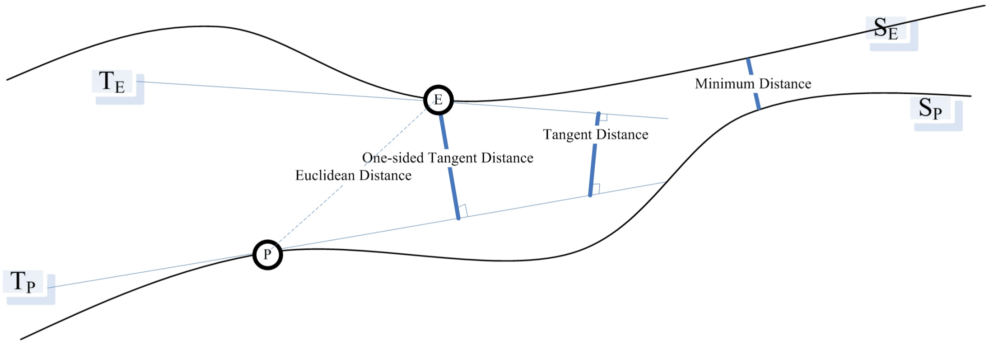


Figure 65: Schematic diagram of the Euclidean distance and the tangent distance between P and E. The curves S_p and S_e represent the sets of patterns after certain transformations to P and E. The lines T_P and T_E represent the tangent spaces passing through P and E, respectively.

6.3.2 Tangent Vector

During the matching process by the tangent distance, the appearance of each input pattern is deformed so as to best fit that of the preceding pattern. This approach has two problems to be solved. First, allowed deformations must be known a priori. Fortunately, this is the case for our plate detection system and we will summarize the allowed deformations in detail later. Second, the search for the best-matching deformation is often enormously expensive. To simplify the search, the one-sided tangent distance is selected, which is defined as

$$D_{TD}\left(P,E\right) = min_{\alpha}\left\{\|P - E - \alpha T_{E}\|\right\}. \tag{6.8}$$

In this study, the representation of repeated patterns should be invariant with respect to position, size changes, slight rotations, or changes in brightness. The dimension of the manifold for each repeated pattern is five corresponding to the five transformations, which are vertical shift, horizontal shift, scaling, rotation, and brightness. The transform in Eq. 6.6 can be rewritten as:

$$s\left(P,\alpha\right) = P + \left(\begin{array}{c} T_{P1}, T_{P2}, T_{P3}, T_{P4}, T_{P5} \end{array} \right) \times \left(\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha 5 \end{array} \right), \tag{6.9}$$

where α_n denotes the scaling parameter of the nth tangent vector, T_{Pn}.

The tangent vectors used in this study are explained as follows:

Horizontal shift

The transformation is useful when vehicles or the sensor moves slightly and horizontally.

$$s\left(P,\alpha_1\right):\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x+\alpha_1 \\ y \end{pmatrix} \tag{6.10}$$

The tangent vector is defined by:

$$T_{P1} = \frac{\partial s\left(P,0\right)}{\partial x} \tag{6.11}$$

Vertical shift

The transformation is useful when vehicles or the sensor moves slightly and vertically.

$$s\left(P,\alpha_2\right):\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y+\alpha_2 \end{pmatrix} \tag{6.12}$$

The tangent vector is defined by:

$$T_{P2} = \frac{\partial s\left(P,0\right)}{\partial y} \tag{6.13}$$

Rotation

This transformation is useful when vehicles make a turn suddenly. In the entry of a building, for example, the vehicle may turn to left or right 90 degree quickly, which causes the captured patterns rotated.

$$s\left(P,\alpha_3\right):\begin{pmatrix} x \\ \\ y \end{pmatrix}\mapsto\begin{pmatrix} x\times\cos\alpha_3-y\times\sin\alpha_3 \\ \\ x\times\sin\alpha_3+y\times\cos\alpha_3 \end{pmatrix} \tag{6.14}$$

The tangent vector is defined by:

$$T_{P4}=y\times\frac{\partial s\left(P,0\right)}{\partial x}-x\times\frac{\partial s\left(P,0\right)}{\partial y} \tag{6.15}$$

Scaling

This transformation is useful when the sensor is installed in the front of a surveillance vehicle. The size of plate regions would become larger and larger when the vehicle approaches the camera.

$$s\left(P,\alpha_4\right):\begin{pmatrix} x \\ \\ y \end{pmatrix}\mapsto\begin{pmatrix} x+\alpha_4\times x \\ \\ y+\alpha_4\times y \end{pmatrix} \tag{6.16}$$

The tangent vector is defined by:

$$T_{P4}=x\times\frac{\partial s\left(P,0\right)}{\partial x}+y\times\frac{\partial s\left(P,0\right)}{\partial y} \tag{6.17}$$

Brightness

This transformation is useful when we monitor vehicles in outdoor environments. The brightness of plate region candidates would be variant in consecutive frames because of lighting changes.

$$s\left(P, \alpha_5\right) : \begin{pmatrix} x \\ \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ \\ y \end{pmatrix} + \alpha_5 \tag{6.18}$$

The tangent vector is defined by:

$$T_{P5} = 1 \tag{6.19}$$

6.3.3 Repeated Block Matching

After BOPE at time t, a plate candidate PC_t could be detected and the candidate would be overlapped with some plate candidates at time $t-1$. The overlapped region is named a candidate region CR as shown in Fig. 66(a). A candidate region would be classified as a repeated region or a non-repeated one in consecutive frames, when we measure the difference between the candidate region at the current frame and the same region at the preceding frame. For example, if patterns P and E represent one of the candidate regions in the current and previous frames, respectively, patten P could be classified as a repeated one if the tangent distance in Eq. 6.8 is lower than a threshold.

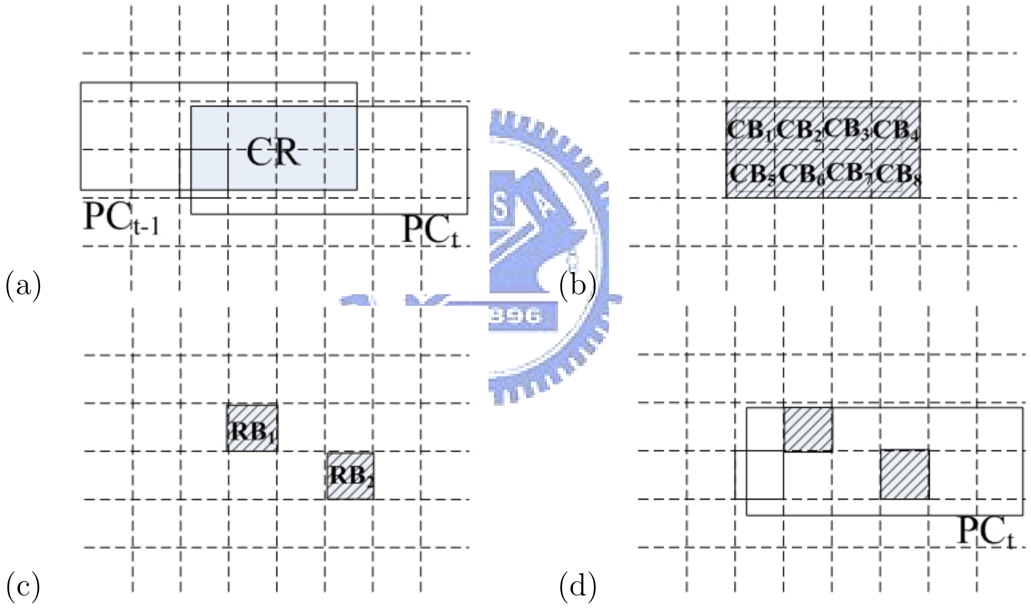


Figure 66: Schematic diagram of the repeated block matching mechanism: (a) the candidate region CR extracted from the overlap between the plate candidates in the current frame PC_t and the preceding frame PC_{t-1} (b) the selected candidate blocks, (c) the matched repeated blocks, and (d) the selected regions to detect repeated regions.

However, the large input dimension of candidate regions may result in matching errors because the terms after the second order in the Taylor expansion are truncated, as shown in Eq. 6.6. In order to reduce the errors and to judge the candidate region easily by a threshold, each image is tessellated into blocks with 16 × 16 dimensions as shown in the dash rectangles of Fig. 66(b). Instead of matching the candidate regions, the tangent distance is measured on the candidate blocks CB, which are selected from the blocks overlapped with the candidate region, as shown in the dark blocks of Fig. 66(b). Then, repeated blocks RB in the current frame, as depicted in Fig. 66(c), could be identified by matching against the same block in the preceding frame. Finally, the repeated candidate is determined when the overlapping ratio of the repeated blocks and the candidate is lower than a threshold. For example, the plate candidate PC_t in Fig. 66(d) could be classified as a non-repeated region because of the low overlapping ratio. The detailed steps are summarized as follows.

1. We obtain two consecutive frames from time $t-1$ to t as displayed in Fig. 67(a).
2. Extract plate candidates by BOPE as shown in Fig. 67(b).
3. Extract candidate regions that are overlapped plate candidates in frames $t-1$ and t as shown in Fig. 67(c).
4. Select candidate blocks that are overlapped with the extracted candidate regions as shown in Fig. 67(d).
5. For each candidate block, compute the tangent vectors according to Eqs.6.10-6.19 in Sections 6.3.2 and 6.3.4.
6. For each candidate block, calculate the one-sided tangent distance defined in Eq. 6.8.
7. Select the repeated blocks whose distance values are lower than a threshold, as shown in Fig. 67(e) with blue rectangles.
8. Select repeated regions when they are overlapped with any repeated block and the overlapped size is greater than a pre-defined percentage, say, 60%, of the region size.

In the example of Fig. 67, among 124 blocks detected as candidate blocks, 113 blocks are matched as repeated ones. All plate candidates extracted by the BOPE algorithm would be rejected because they are all repeated regions. Note that the license plate has already been extracted from one of the previous frames.

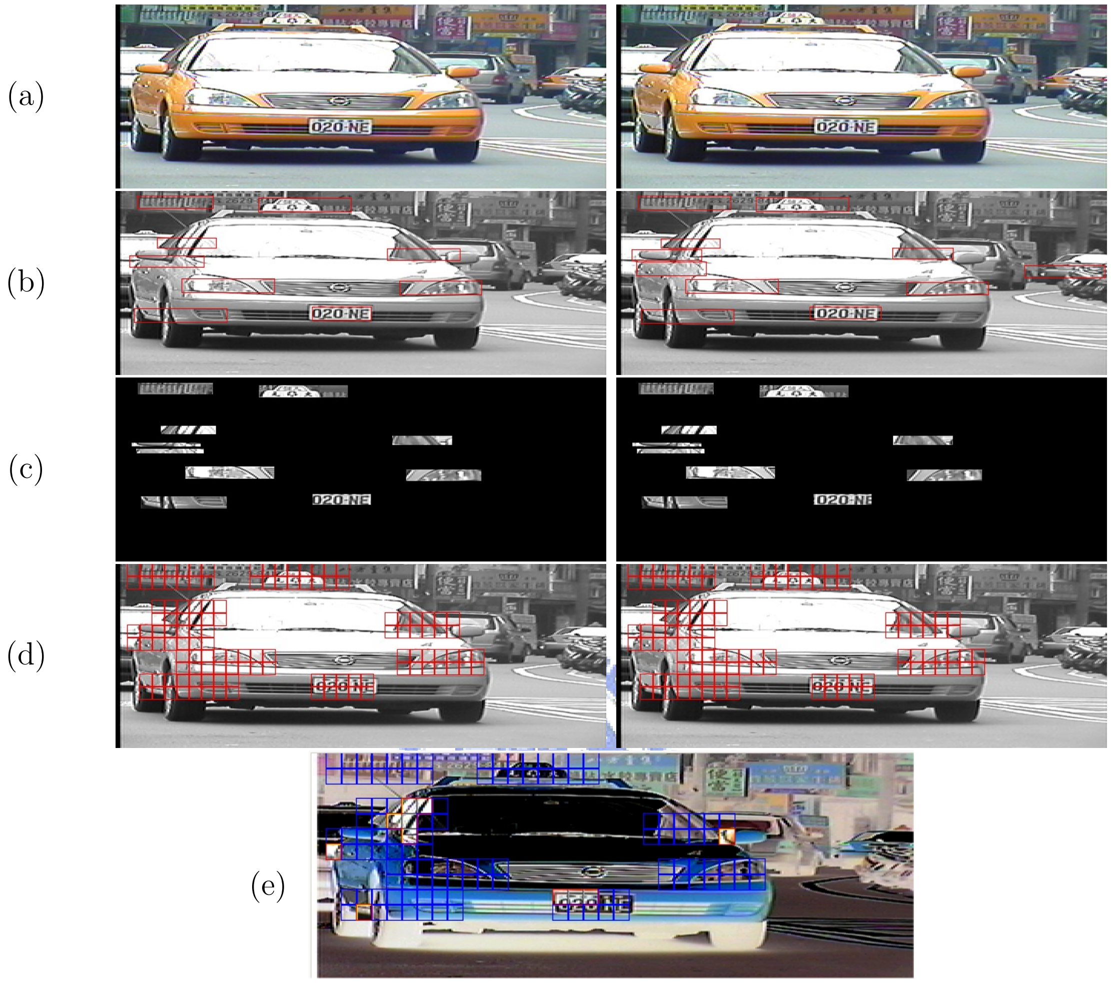


Figure 67: An example of the repeated block matching mechanism: (a) Left: the frame at time $t-1$, Right: the frame at time t, (b) the plate candidates extracted by BOPE algorithm, (c) the candidate regions, (d) the candidate blocks, and (e) the repeated blocks of the frame at time t.

6.3.4 Improvement of Computation Speed

To detect repeated regions, the tangent vectors should be calculated for each input candidate block. Undoubtedly, the calculation cost involved in repeated region detection will increase when there are more candidate blocks. To reduce the computational load, we match the similarity between the two input patterns, P and E, by the one-sided tangent distance from P to E and vice versa. Since we match the similarity by the one-sided tangent distance, 50 percent of computational load would be reduced; that is, only one tangent vector of the candidate block is calculated for every two frames, as shown in Fig. 68(a).

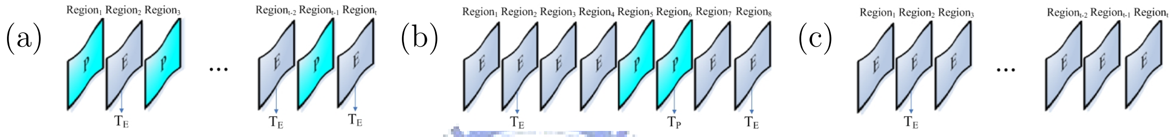

(a)
Region$_1$ Region$_2$ Region$_3$
P
E
P
...
Region$_{t-2}$ Region$_{t-1}$ Region$_t$
E
P
E
T$_E$
T$_E$
T$_E$
(b)
Region$_1$ Region$_2$ Region$_3$ Region$_4$ Region$_5$ Region$_6$ Region$_7$ Region$_8$
E
E
E
E
P
P
E
E
T$_E$
T$_P$
T$_E$
(c)
Region$_1$ Region$_2$ Region$_3$
E
E
E
...
Region$_{t-2}$ Region$_{t-1}$ Region$_t$
E
E
E
T$_E$

Figure 68: Schematic diagram of the acceleration mechanism for the calculations of tangent vectors. (a) For the worst case, tangent vectors are re-calculated at every two frames since the input region varies continuously. (b) For the generic case, tangent vectors are re-calculated at the next time after identifying the input region is identified as a non-repeated region. (c) For the best case, tangent vectors are calculated only once because the same patterns occur continuously.

However, for a sequence of images with repeated blocks, we still have to compute the same tangent vectors n times if the same blocks exist for $2n$ frames. Since a repeated block means that the block in the current frame is closed to the one in the preceding frame in tangent distances, we could keep the block and its tangent vectors for matching the blocks in the following frames. In other words, the tangent vectors of a candidate block have only to be re-calculated for a non-repeated one. For example, in Fig. 68(b), the tangent vectors are only calculated at time instances 2, 6, and 8. The mechanism provides a significant reduction in computations since repeated patterns exist in surveillance applications very commonly. In the best case illustrated in Fig. 68(c), the tangent vectors would only be calculated once no matter how long the sequence is.

6.4 Experimental Results and Discussion

This section presents the performance of the BOPE and repeated region detection to extract license plates in different sequences. The system proposed in this study was implemented in C language and tested on a standard Intel Pentium 4, 3 GHz personal computer. To evaluate the system performance, the ground truth for each frame is obtained by segmenting the images with manual classification of license plates and background regions. Five parameters of the test sequences collected outdoors are listed in Table 6.1. In the table, the dimensions are given in column "Dim", while the ranges of plate heights by pixels in column "P_H". Moreover, the numbers of license plates, the number of frames without license plates, and total number of frames are listed in columns "P_N", "BI_N", and "TI_N", respectively.

Table 6.1: The sequences used in the benchmark

Dataset	Highway	Campus	Entrance Gate
Type	outdoor	outdoor	outdoor
Dim	640 × 480	640 × 480	640 × 240
P_H	30–41	36-50	24–40
P_N	620	648	1400
BI_N	6720	0	7510
TI_N	7540	360	8910

6.4.1 Performance of Spatial SSR

The comparison between OPE and BOPE has been set up with four indexes: numbers of detected plates (TPC_N), numbers of non-plate candidates (NPC_N), average pixels of plate candidates per frame (AP_f), and plate region fitness (PRF). For the first index, the number of plate regions is counted when two criteria are confirmed: (1) The plate region is overlapped with some plate candidates extracted by BOPE. (2) The differences

Table 6.2: The comparisons between the OPE and the BOPE algorithms

Dataset	Highway			Campus			Entrance Gate		
	OPE	BOPE		OPE	BOPE		OPE	BOPE	
TPC_N	620	620	-	596	648	**8.02%**	1,400	1,400	-
NPC_N	90,480	89,328	**1.27%**	2,892	2,858	**1.18%**	31,952	30,355	**5.00%**
AP_f	0.537	0.226	**57.91%**	0.895	0.526	**41.23%**	0.723	0.282	**61.00%**
PRF_t	0.110	0.027	**75.45%**	0.146	0.053	**63.70%**	0.024	0.003	**87.50%**
PRF_b	0.034	0.021	**38.23%**	0.089	0.041	**53.93%**	0.010	0.003	**70.00%**
PRF_l	0.066	0.050	**24.24%**	0.168	0.053	**68.45%**	0.163	0.047	**71.17%**
PRF_r	0.069	0.056	**18.84%**	0.135	0.060	**55.55%**	0.132	0.048	**63.64%**

of top or bottom boundaries between the plate region and the detected candidate are less than a threshold (said, four pixels in our experiments) for avoiding over-detection or under-detection. The second index NPC_N is counted from the detected plate candidates not overlapping with any true plate regions. The third index AP_f is presented to measure how many pixels remained for the following procedures. This index is defined as the ratio of the total number of pixels in the detected plate candidates to the numbers of pixels in the input frame. The last index PRF is used to evaluate the compactness from the differences of boundaries between the plate region and the detected candidate. For clearly expressing the compactness for the four boundaries, the PRF is further separated into four sub-indexes: the differences of the top PRF_t, the bottom PRF_b, the left PRF_l, and the right PRF_r boundaries.

The experimental results after OPE and BOPE are presented in Table 6.2; the improvements of the BOPE for each measurement are also demonstrated. All 648 plates in the dataset Campus were extracted by BOPE. AP_f is improved up to 61% because large plate candidates would be reduced to the restricted sizes in BOPE. Several large plate candidates were extracted by OPE because of the noisy background environments in the outdoor area on a rainy day or in a parking space.

6.4.2 Performance of Spatiotemporal SSR

Table 6.3: The comparisons between adopting repeated region prediction and none

Dataset	Highway			Campus			Entrance Gate		
	OPE	RRP		OPE	RRP		OPE	RRP	
TPC_N	620	620	-	596	215	**63.93%**	1,400	1,230	**12.14%**
NPC_N	90,480	45,182	**50.06%**	2,892	1,239	**57.16%**	31,952	21,780	**31.84%**
AP_f	0.537	0.121	**77.47%**	0.895	0.226	**74.75%**	0.723	0.121	**83.26%**

After repeated region detection, performance is demonstrated by three indexes: numbers of detected plates (TPC_N), numbers of non-plate candidates (NPC_N), and average numbers of pixels of plate candidates per frame (AP_f). The experimental results with and without repeated region detection are shown in Table 6.3; the rates of spatiotemporal search-space reduction (SSR) after the RRP for each measurement were also demonstrated.

When the great majority of candidate regions are extracted, TPC_N or NPC_N could be reduced significantly after repeated patterns are rejected. Comparing to OPE, for the dataset Highway, our spatiotemporal mechanism improves NPC_N and AP_f up to 50.06% and 77.47%, respectively, but zero to TPC_N because the vehicles were constantly moving. In other words, 87.9% of the search-space could be rejected. For other datasets, Campus and Entrance Gate, both TPC_N and NPC_N are saved; meanwhile, AP_f would also be improved up to 83.26% because most of the candidates are rejected.

Table 6.4: The comparisons adopting different parameters of the plate height H_p used in OPE and BOPE

Dataset	Entrance Gate								
H_p	10 ∼ 30 pixels			30 ∼ 60 pixels			24 ∼ 40 pixels		
	OPE	RRP		OPE	RRP		OPE	RRP	
TPC_N	1,392	1,123	-	1,400	1,230	-	1,400	1,230	-
NPC_N	57,690	23,332	**59.56%**	35,393	21,322	**39.76%**	31,952	21,780	**31.84%**
AP_f	0.733	0.057	**92.24%**	1.308	0.330	**59.48%**	0.723	0.121	**83.26%**
PRF_t	0.030	0.005	**83.33%**	0.024	0.007	**70.83%**	0.024	0.003	**87.50%**
PRF_b	0.005	0.003	**40.00%**	0.006	0.003	**50.00%**	0.010	0.003	**70.00%**
PRF_l	0.202	0.039	**80.69%**	0.170	0.071	**58.24%**	0.163	0.047	**71.17%**
PRF_r	0.197	0.042	**78.68%**	0.176	0.053	**69.89%**	0.132	0.048	**63.64%**

Different ranges of the parameter, plate height H_p, in OPE and BOPE were experimented as shown in Table 6.4 to demonstrate the improvements in search space reduction of our spatiotemporal SSR mechanism to detect variant license plates. Another parameter of the plate width W_p in OPE and BOPE was fixed to eight times of H_p. Several phenomena were observed from the experiments. First, we could extract all compact license plate regions when the range of parameters is enlarged to cover variations in plate heights. Second, we could extract more plate candidates when the inputs have complex patterns such as the texture of the vehicles or the backgrounds. Third, there were different groups of plate candidates when we adopted OPE and BOPE with varied parameters; several plate candidates covered the same regions. In other words, some regions should be processed repeatedly even if they were not plate regions; the dimension of the search space increased when OPE extracted large covered regions or a great quantity of plate candidates. For the range of plate height from 30 to 60, AP_f is up to 1.308, which indicates that the dimension of the search space for the range is higher than dimensions for the others. When there are more repeated candidates with covered regions, we could reduce the search space more effectively and significantly using the same repeated blocks. Above results indicate that the spatiotemporal SSR mechanism would perform well when we provide a precise range of H_p. Also, the mechanism could detect smaller license plates when we provide a larger range of H_p.

Chapter 7 Conclusion and Future Work

7.1 Achievements

In this dissertation, we have presented a cascaded framework for a plate recognition system that minimizes the computation time while achieving a high frame rate and a high accuracy rate. The framework is used to develop a plate recognition system that operates only with information presented in a single gray-level image and at a rate of over 30 fps. Without scaling the input image, the proposed system performs well even if the plates have different sizes. Moreover, this study has combined spatial plate detection and temporal repeated region detection into a spatiotemporal license plate detection system. By eliminating non-interesting regions (said, non-plate, non-character, and repeated ones), the proposed mechanisms significantly reduces the search space required for further plate classification, character segmentation, and character recognition.

This study combines low-level image processing and high-level statistical analyses, which are very generic and may have broader applications in computer vision. The first contribution of this study is to propose new techniques for directly extracting compact plate regions in various plates and environments by using the OPE or BOPE. By eliminating the need to compute a multiscale image pyramid, the BOPE algorithm significantly reduces the initial image processing effort required for plate detection. Moreover, the BOPE algorithm can reduce large spatial search-space and tightly bound the top and bottom of license plates for compact plate regions.

The second contribution is the repeated region representation (said, backgrounds or classified patterns) and the detection technique with invariant matching measures. The tangent distance is introduced to measure the similarity with respect to certain sensing variations of positions, sizes, slight rotations, or brightness. By representing each frame as several block candidates in the tangent space, the matching problem is reduced to finding the distance between subspaces under arbitrary and parameterized transformations. Without any explicit background modeling or tracking of the plates, the proposed mechanism could be expected to be more robust in cases where background modeling and tracking are difficult. In the cases of many repeated patterns, the algorithm of repeated region detection performs well to reject repeated ones, including the backgrounds and license plates. On the other hand, in the case of few repeated patterns, the mechanism may increase the computational load when we monitor constantly changed scenes, such as the sequence captured from a high speed moving camera. Fortunately, the situation seldom appears in the surveillance applications because we do not need to compare many candidate regions for moving vehicles.

The third contribution of this study is a simple and efficient segmentation technique developed from computationally efficient features using peak-valley selection for the determination of dominant thresholds. This segmentation method is computationally efficient since only a small number of candidates must be evaluated by the computation-required procedure, OCR.

The fourth contribution of this study is a rejection-based cascade framework for constructing a cascade of rejecters, which dramatically reduces the computation time while maintaining very high detection accuracy. The initial rejecters of the cascade are designed to reject a majority of the input regions in order to focus the successive processing on promising regions. The advantage of this framework is that it allows tradeoffs between the processing time and the system performance. This framework is clearly effective for the plate recognition and we are confident that it will also be effective in other domains such as sign or text detection.

Finally, this dissertation presents the co-design of plate detection and character segmentation using the same summed area tables to avoid recalculating the procedures. Skewed Haar-like features are also proposed to represent skewed plate images in a better manner. We tested the mechanisms with a large number of frames captured in different environments from real applications.

7.2 Future Works

A cascade framework has been presented and a real-time statistical license plate recognition system is developed based on spatial and temporal rejection mechanisms. Alternative sources of information, such as color filter or motion detection could also be integrated into our system to achieve higher frame rates or deal with special cases.

The computational resource demand of the learning procedure is currently the main drawback, taking several days the processing of about one thousand positives on a Pentium IV PC, at 3.0GHz in the conditions of the experiments reported.

Bibliography

[1] Y. Amit, D. Geman, and X. Fan, “A Coarse-to-Fine Strategy for Multiclass Shape Detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 12, pp. 1606–1621, 2004.

[2] B. Enyedi and *et al.*, “Strategies for Fast License Plate Number Localization,” in *IEEE Intl. Symposium on Electronics in Marine*, pp. 579–584, Zadar, Croatia, June. 2004.

[3] I. Haritaoglu, D. Harwood, and L. Davis, “W4: real-time surveillance of people and their activities,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[4] A. Lipton, H. Fujiyoshi, and R. Patil, “Moving Target Classification and Tracking from Real-time Video,” in *IEEE workshop applications of Computer Vision*, vol. 2, pp. 8–14, Monterey, CA, USA, Nov. 1998.

[5] B. G. Schunck, “The Image Flow Constraint Equation,” *Intl. Journal on Graphics, Vision and Image Processing*, vol. 35, pp. 20–46, 1986.

[6] W. Hu and *et al.*, “A Survey on Visual Surveillance of Object Motion and Behaviors,” *IEEE Trans. on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 34, pp. 334–352, Aug 2004.

[7] M. Zayed, J. Boonaert, and M. Bayart, “License Plate Tracking for Car Following with a Single Camera,” in *IEEE Intl. Conf. on Intelligent Transportation Systems*, pp. 719–724, Osaka, Japan, May 2004.

[8] W. Jia, H. Zhang, X. He, and M. Piccardi, “Mean Shift for Accurate License Plate Localization,” in *IEEE Intl. Conf. on Intelligent Transportation Systems*, Vienna, Austria, Sept. 2005.

[9] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, “Robust license-plate recognition method for passing vehicles under outside environment,” *IEEE Trans. on Vehicular Technology*, pp. 2309–2319, Nov. 2000.

[10] J.-W. Hsieh, S.-H. Yu, and Y.-S. Chen, “Morphology-based license plate detection in images of differently illuminated and oriented cars,” *Jour. of Electronic Imaging*, vol. 11, no. 4, pp. 507–516, 2002.

[11] J. Cano and J.-C. Perez-Cortes, “Vehicle License Plate Segmentation in Natural Images,” in *Iberian Conf. on Pattern Recognition and Image Analysis*, pp. –, Mallorca, Spain, June 2003.

[12] A. Taleb-Ahmed, D. Hamad, and G. Tilmant, “Vehicle License Plate Recognition in Marketing Application,” in *IEEE Intelligent Vehicles Symposium*, pp. 90–94, Ohio, USA, June 2003.

[13] L. Xiaobo, L. Xiaojing, and H. Wei, “Vehicle License Plate Character Recognition,” in *Intl. Conf. on Neural Networks and Signal Processing*, vol. 2, pp. 1066–1069, Nanjing, China, Dec. 2003.

[14] V. Koval, V. Turchenko, V. Kochan, A. Sachenko, and G. Markowsky, “Smart License Plate Recognition System Based on Image Processing Using Neural Network,” in *IEEE Intl. Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 123–127, Lviv Ukraine, Sep. 2003.

[15] M. Sarfraz, M. J. Ahmed, and S. A. Ghazi, “Saudi Arabian License Plate Recognition System,” in *Intl. Conf. on Geometric Modeling and Graphics*, pp. 36–41, Los Alamitos, CA, USA, July 2003.

[16] J. B. Kim, S. W. Jang, and C. K. Kim, “Recognition of Car License Plate by Using Dynamical Thresholding Method and Enhanced Neural Networks,” in *Intl. Conf. on Computer Analysis of Images and Patterns*, pp. 309–319, Groningen, Netherlands, Aug. 2003.

[17] H. Lee, D. Kim, D. Kim, and S. Y. Bang, "Real-time Automatic Vehicle Management System using Vehicle Tracking and Car Plate Number Identification," in *IEEE Intl. Conf. on Multimedia and Expo*, vol. 2, pp. II–353–6, Maryland, USA, July 2003.

[18] M.-A. Ko and Y.-M. Kim, "License Plate Surveillance System using Weighted Template Matching," in *Applied Imagery Pattern Recognition Workshop*, pp. 269–274, Washington, US, Oct. 2003.

[19] M. Cinsdikici and T. Tunah, "License Plate Segmentation for Intelligent Transportation Systems," in *Intl. Symposium on Computer and Information Sciences*, Antalya, Turkey, Nov. 2003.

[20] M. J. Ahmed, M. Sarfraz, A. Zidouri, and W. G. Al-Khatib, "License Plate Recognition System," in *Intl. Conf. on Electronics, Circuits and Systems*, vol. 2, pp. 898–901, Sharjah, United Arab Emirates, Dec. 2003.

[21] M. M. I. Chacon and S. A. Zimmerman, "License Plate Location Based on a Dynamic PCNN Scheme," in *IEEE Intl. Joint Conf. on Neural Networks*, vol. 2, pp. 1195–1200, Portland, Oregon, July 2003.

[22] S. Yoshimori, Y. Mitsukura, M. Fukumi, N. Akamatsu, and R. Khosal, "License Plate Detection System in Rainy Days," in *IEEE Intl. Symposium on Computational Intelligence in robotics and Automation*, pp. 972–976, Kobe, Japan, July 2003.

[23] F. Kahraman, B. Kurt, and M. Gökmen, "License Plate Character Segmentation Based on the Gabor Transform and Vector Quantization," in *Intl. Symposium on Computer and Information Sciences*, Antalya, Turkey, Nov. 2003.

[24] Y. Yanamura, M. Goto, D. Nishiyama, M. Soga, H. Nakatani, and H. Saji, "Extraction and Tracking of the License Plate Using Hough Transform and Voted Block Matching," in *IEEE Intelligent Vehicles Symposium*, pp. 243–246, Columbus, OH, USA, June 2003.

[25] S.-Z. Wang and H.-J. Lee, "Detection and Recognition of License Plate Characters with Different Appearances," in *IEEE Intl. Conf. on Intelligent Transportation Systems*, vol. 2, pp. 979–984, Shanghai, China, Oct. 2003.

[26] J. Xu, S. Li, and Z. Chen, "Color Analysis for Chinese Car Plate Recognition," in *IEEE Intl. Conf. on Robotics*, vol. 2, pp. 1312–1316, Taipei, Taiwan, Oct. 2003.

[27] Z. B. Huang and Y. F. Guo, "Classifier Fusion-based Vehicle License Plate Detection Algorithm," in *Intl. Conf. on Machine Learning and Cybernetics*, vol. 5, pp. 2984–2989, Xian, China, Nov. 2003.

[28] L. Jin and J. Qin, "Car Plate Number Characters Recognition using Gabor Orientation Features and Neural Networks," in *Intl. Conf. on Neural Networks and Signal Processing*, vol. 2, pp. 1628–1631, Nanjing, China, Dec. 2003.

[29] P. Han, W. Han, D. F. Wang, and Y. J. Zhai, "Car License Plate Feature Extraction and Recognition based on Multi-stage Classifier," in *Intl. Conf. on Machine Learning and Cybernetics*, vol. 1, pp. 128–132, Xian, China, Nov. 2003.

[30] F. Cortijo, S. Villena, R. Molina, and A. Katsaggelos, "Bayesian Super-Resolution of Text Image Sequences from Low Resolution Observations," in *IEEE Intl. Symposium on Signal Processing and Its Applications*, vol. 1, pp. 421–424, Paris, France, July 2003.

[31] F. Martin and D. Borges, "Automatic Car Plate Recognition using a Partial Segmentation Algorithm," in *The IASTED Conf. on Signal Processing, Pattern Recognition, and Applications*, Rhodes, Greece, June 2003.

[32] G. Gao, J. Chen, and J. Jiang, "An Adaptive Approach to Vehicle License Plate Localization," in *IEEE Intl. Conf. on Industrial Electronics Society*, vol. 2, pp. 1786–1791, Roanoke, VA, Nov. 2003.

[33] S. Vladimir, D. Dimo, B. Stefan, V. Veselin, and G. Georgi, "Adaptive License Plate Image Extraction," in *ACM Intl. Conf. on Computer Systems and Technologies*, pp. 1–7, Sofia, Bulgaria, June 2003.

[34] C. A. Rahman, W. Badawy, and A. Radmanesh, "A Real Time Vehicle's License Plate Recognition System," in *IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance*, pp. 163–166, Miami, USA, July 2003.

[35] Y. Zhang and C. Zhang, "A New Algorithm for Character Segmentation of License Plate," in *IEEE Intelligent Vehicles Symposium*, pp. 106–109, Columbus, OH, USA, June 2003.

[36] H. Bai, J. Zhu, and C. Liu, "A Fast License Plate Extraction Method on Complex Background," in *IEEE Intl. Conf. on Intelligent Transportation Systems*, vol. 2, pp. 985–987, Shanghai, China, Oct. 2003.

[37] S. Yoshimori, Y. Mitsukura, M. Fukumi, and N. Akamatsu, "A Design of the Object Detection System using the RGA," in *IEEE Intl. Joint Conf. on Neural Networks*, vol. 2, pp. 1227–1231, Portland, Oregon, July 2003.

[38] S. Yoshimori, Y. Mitsukura, M. Fukumi, and N. Akamatsu, "License Plate Detection Using Hereditary Threshold Determine Method," *Lecture Notes in Computer Science*, vol. 2773, pp. 585–593, 2003.

[39] T. Nukano, M. Fukumi, and M. Khalid, "Vehicle License Plate Character Recognition by Neural Networks," in *IEEE Intl. Symposium on Intelligent Signal Processing and Communication Systems*, pp. 771–775, Seoul, Korea, Nov. 2004.

[40] S. Demir, H. H. Eryilmaz, and A. Ercil, "Extracting License Plates from Images with 3-tier Filter Banks," in *IEEE Intl. Conf. on Signal Processing and Communications Applications*, pp. 347–350, Kusadasi, Turkey, April 2004.

[41] G. Ji, H. Yi, B. Qin, and H. Xu, "The Recognition of the Vehicle License Plate Based on Modular Networks," in *IEEE Intl. Symposium on Neural Networks*, Dalian, China, Aug. 2004.

[42] B. Enyedi, L. Konyha, C. Szombathy, and K. Fazekas, "Strategies for Fast License Plate Number Localization," in *IEEE Intl. Symposium on Electronics in Marine*, pp. 579–584, Zadar, Croatia, June. 2004.

[43] T. H. Wang, F. C. Ni, K. T. Li, and Y. P. Chen, "Robust License Plate Recognition based on Dynamic Projection Warping," in *IEEE Intl. Conf. on Networking, Sensing and Control*, vol. 2, pp. 784–788, Taipei, Taiwan, Mar. 2004.

[44] V. S. L. Nathan, J. Ramkumar, and P. S. Kamakshi, "New Approaches for License Plate Recognition System," in *Intl. Conf. on Intelligent Sensing and Information Processing*, Madras, India, Jan. 2004.

[45] V. Shapiro and G. Gluhchev, "Multinational License Plate Recognition System: Segmentation and Classification," in *IEEE Intl. Conf. on Pattern Recognition*, pp. 352–355, Cambridge, UK, Aug. 2004.

[46] J. Xiong, S. D. Du, and D. T. Gao, "Locating Car License Plate Under Various Illumination Conditions using Genetic Algorithm," in *Intl. Conf. on Signal Processing*, Beijing, China, Aug. 2004.

[47] R. C. P. Marques and F. N. S. Medeiros, "License Vehicle Plates Localization Using Maximum Correlation," in *Joint IAPR international workshops on Structural, Syntactic, and Statistical Pattern Recognition*, Lisbon, Portugal, Aug. 2004.

[48] V. Seetharaman, A. Sathyakhala, N. Vidhya, and P. Sunder, "License Plate Recognition System using Hybrid Neural Networks," in *IEEE Annual Meeting of Fuzzy Information, NAFIPS*, pp. 363–366, Alberta, Canada, June 2004.

[49] S. Yohimori, Y. Mitsukura, M. Fukumi, N. Akamatsu, and N. Pedrycz, "License Plate Detection System by Using Threshold Function and Improved Template Matching Method," in *IEEE Annual Meeting of Fuzzy Information, NAFIPS*, pp. 357–362, Alberta, Canada, June 2004.

[50] X. Fan and D. Geman, "Hierarchical Object Indexing and Sequential Learning," in *IEEE Intl. Conf. on Pattern Recognition*, vol. 3, pp. 65–68, Singapore, Oct. 2004.

[51] H. J. Lee, S. Y. Chen, and S.-Z. Wang, "Extraction and Recognition of License Plates of Motorcycles and Vehicles on Highways," in *IEEE Intl. Conf. on Pattern Recognition*, vol. 4, pp. 356–359, Singapore, Oct. 2004.

[52] X. Fan, "Contextual Disambiguation for Multi-class Object Detection," in *IEEE Intl. Conf. on Image Processing*, Singapore, Oct. 2004.

[53] D. D. Tran, A. D. Duong, and H. D. L. Tran, "Combining Hough Transform and Contour Algorithm for Detecting Vehicles' License-Plates," in *Intl. Symposium on Intelligent Multimedia, Video and Speech Processing*, HongKong, China, Oct. 2004.

[54] S. Zhang, M. Zhang, and X. Ye, "Car Plate Character Extraction under Complicated Environment," in *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, vol. 5, pp. 4722–4726, Sydney, Austria, Oct. 2004.

[55] Y. Cheng, J. M. Lu, and T. Yahagi, "Car License Plate Recognition Based on the Combination of Principal Components Analysis and Radial Basis Function Netowrks," in *Intl. Conf. on Signal Processing*, Beijing, China, Aug. 2004.

[56] J. F. Xu, S. F. Li, and M. S. W, "Car License Plate Extraction using Color and Edge Information," in *Intl. Conf. on Machine Learning and Cybernetics*, pp. 26–29, Shanghai, China, Aug. 2004.

[57] Y. H. Shu, M. C. Sun, and W. S. Feng, "Blind Signal Extraction Algorithm for the License Plate Matching of Vehicle Positioning System," in *IEEE Intl. Workshop on Electronic Design, Test and Applications*, p. 440, Taipei, Taiwan, July 2004.

[58] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic License Plate Recognition," *IEEE Trans. on Intelligent Transportation Systems*, vol. 5, pp. 42–53, Mar. 2004.

[59] W. Jia, X. He, and M. Piccardi, "Automatic License Plate Recognition: a Review," in *Intl. Conf. on Imaging Science, Systems and Technology*, pp. 43–49, Las Vegas, Nevada, USA, June 2004.

[60] Y. P. Huang, S. Y. Lai, and W. P. Chuang, "A Template-Based Model for License Plate Recognition," in *IEEE Intl. Conf. on Networking, Sensing and Control*, vol. 2, pp. 737–742, Taipei, Taiwan, Mar. 2004.

[61] P. Xiang, Y. Xiuzi, and Z. Sanyuan, "A Hybrid Method for Robust Car Plate Character Recognition," in *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, vol. 5, pp. 4733–4737, Sydney, Austria, July 2004.

[62] J. Kong, X. Liu, Y. Lu, X. Zhou, and Q. Zhao, "Robust License Plate Segmentation Method Based On Texture Features and Radon Transform," in *Australian Conf. on Artificial Intelligence*, vol. 3809, pp. 510–519, Sydney, Australia, Dec. 2005.

[63] J. Kong, X. Liu, Y. Lu, and X. Zhou, "A Novel Approach For Car License Plates Localization Based On Textural Feature Analysis," in *IEEE Intl. Symposium on Signal Processing and Information Technology*, pp. 275–279, Athens, Greece, Dec. 2005.

[64] V. Shapiro, G. Gluhchev, and D. Dimov, "Towards a Multinational Car License Plate Recognition System," *Jour. of Machine Vision and Applications*, vol. 17, no. 3, pp. 173–183, 2006.

[65] X. Pan, X. Ye, and S. Zhang, "A Hybrid Method for Robust Car Plate Character Recognition," *Engineering Applications of Artificial Intelligence*, vol. 18, pp. 963–972, 2005.

[66] Y. A. Syed and M. Sarfraz, "Color Edge Enhancement Based Fuzzy Segmentation of License Plates," in *Intl. Conf. on Information Visualisation*, pp. 227–232, London, UK, July 2005.

[67] Y. Lu, L. Yu, J. Kong, and C. Tang, "A Novel License Plate Location Method Based on Neural Network and Saturation Information," in *Australian Joint Conf. on Artificial Intelligence*, pp. 1037–1040, Sydney, Australia, Dec. 2005.

[68] Y.-Q. Yang, J. Bai, R.-L. Tian, and N. Liu, "Research of Vehicle License Plate Location Algorithm Based on Color Features and Plate Processions," in *Intl. Conf. on Machine Learning and Cybernetics*, vol. 3930, pp. 1077–1085, Guangzhou, China, Aug. 2005.

[69] X. Shi, W. Zhao, and Y. Shen, "Automatic License Plate Recognition System Based on Color Image Processing," in *Intl. Conf. on Computational Science and its Applications*, Singapore, May 2005.

[70] Y. Q. Yang, J. Bai, R. L. Tian, and N. Liu, "A Vehicle License Plate Recognition System Based on Fixed Color Collocation," in *Intl. Conf. on Machine Learning and Cybernetics*, pp. 18–21, Guangzhou, China, Aug. 2005.

[71] K. Kim, K. Jung, and H. Kim, "Fast color texture-based object detection in images: Application to license plate localization," *Lecture Notes in Computer Science*, vol. 177, pp. 297–320, 2005.

[72] W. Jia, H. Zhang, X. He, and Q. Wu, "Refined Gaussian Weighted Histogram Intersection and Its Application in Number Plate Categorization," in *Intl. Conf. on Computer Graphics, Imaging and Visualization*, pp. 249–255, Sydney, Australia, July 2006.

[73] W. Jia, H. Zhang, X. He, and Q. Wu, "Gaussian Weighted Histogram Intersection for License Plate Classification," in *IEEE Intl. Conf. on Pattern Recognition*, pp. 574–577, Hong Kong, Aug. 2006.

[74] K. V. Suresh and A. N. Rajagopalan, "A Discontinuity Adaptive Method for Super-Resolution of License Plates," in *Indian Conf. on Computer Vision, Graphics, and Image Processing*, pp. 25–34, Madurai, India, Dec. 2006.

[75] K. V. Suresh, G. M. Kumar, and A. N. Rajagopalan, "Superresolution of License Plates in Real Traffic Videos," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, pp. 321–331, April 2007.

[76] W. Jia, H. Zhang, and X. He, "Region-based license plate detection," *Jour. of Network and Computer Applications*, vol. 30, no. 4, pp. 1324–1333, 2007.

[77] R. Zunino and S. Rovetta, "Vector quantization for license plate location and image coding," *IEEE Trans. on Industrial Electronics*, pp. 159–167, 2000.

[78] J. Cano and J. C. Perez-Cortes, "Vehicle License Plate Segmentation in Natural Images," *Lecture Notes in Computer Science*, vol. 2652, pp. 142–149, 2003.

[79] H. Bai and C. Liu, "A hybrid License Plate Extraction Method Based On Edge Statistics and Morphology," in *IEEE Intl. Conf. on Pattern Recognition*, vol. 2, pp. 831–834, Cambridge, UK, Aug. 2004.

[80] Z. Ma and F. Yang, "Vehicle License Plate Detection based on Projection and Mathematical Morphology.," in *The IASTED Conf. on Circuits, Signals, and Systems*, pp. 234–238, Marina del Rey, CA, USA, Oct. 2005.

[81] C. T. Hsieh, Y. S. Juan, and K. M. Hung, "Multiple License Plate Detection for Complex Background," in *ACM Intl. Conf. on Advanced Information Networking and Application*, vol. 2, pp. 389–392, Tamkang Uni., Taiwan, Mar. 2005.

[82] W. Jia, H. Zhang, and X. He, "Mean Shift for Number Plate Detection," in *IEEE Intl. Conf. on Information Technology and Applications*, Sydney, Austria, July 2005.

[83] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognition Letters*, vol. 26, pp. 2431–2438, Nov. 2005.

[84] F. Porikli and T. Kocak, "Robust License Plate Detection Using Covariance Descriptor in a Neural Network Framework," in *IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance*, pp. 107–107, Sydney, Australia, Nov. 2006.

[85] H. Zhang, W. Jia, X. He, and Q. Wu, "Real-Time License Plate Detection Under Various Conditions," in *Intl. Conf. on Ubiquitous Intelligence and Computing*, vol. 4159, pp. 192–199, Wuhan, China, Sep. 2006.

[86] H.-H. P. Wu, H.-H. Chen, R.-J. Wu, and D.-F. Shen, "License Plate Detection for Low-resolution Video," in *IEEE Intl. Conf. on Pattern Recognition*, vol. 1, pp. 824–827, Hong Kong, Aug. 2006.

[87] H. Zhang, W. Jia, X. He, and Q. Wu, "Learning-Based License Plate Detection Using Global and Local Features," in *IEEE Intl. Conf. on Pattern Recognition*, pp. 1102–1105, Hong Kong, Aug. 2006.

[88] C.-C. Chen, J.-W. Hsieh, J.-C. Wu, and Y.-S. Chen, "Detecting License Plates from Videos Using Morphological Operations and Adaboosting Algorithm," in *Intl. Computer Symposium*, Taipei, Taiwan, Dec. 2006.

[89] C.-C. Chen, J.-W. Hsieh, J.-C. Wu, and Y.-S. Chen, "Detecting License Plates from Videos Using Morphological Operations and Adaboosting Algorithm," in *Conf. on Computer Vision, Graphics, and Image Processing*, Taoyuan, Taiwan, Aug. 2006.

[90] X. Yuan, L. Wang, and M. Zhu, "Car Plate Localization Using Modified PCNN in Complicated Environment," in *Intl. Conf. on Intelligent Computing*, pp. 1116–1124, Kunming, China, Aug. 2006.

[91] H. Zhang, W. Jia, X. He, and Q. Wu, "A Fast Algorithm for License Plate Detection in Various Conditions," in *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, Taipei, Taiwan, Oct. 2006.

[92] M. Ebrahimi, R. Monsefi, S. Ildarabadi, M. R. Akbarzadeh, and M. Habini, "License Plate Location Based on Multi Agent Systems," in *Intl. Conf. on Intelligent Engineering Systems*, pp. 155–159, Budpaest, Hungary, June 2007.

[93] B. R. Lee, K. Park, H. Kang, H. Kim, and C. Kim, "Adaptive Local Binarization Method for Recognition of Vehicle License Plates," pp. 646–655, Auckland, New Zealand, Dec. 2004.

[94] P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Trans. on Vehicular Technology*, vol. 44, no. 4, pp. 790–799, 1996.

[95] S. Draghici, "A neural network based artificial vision system for license plate recognition," *Intl. Jour. of Neural Systems*, vol. 8, pp. 113–126, Feb. 1997.

[96] F. Aghdasi and H. Ndungo, "Automatic License Plate Recognition System," in *IEEE AFRICON Conf. in Africa*, vol. 1, pp. 45–50, Taipei, Taiwan, Sept. 2004.

[97] I. Paliy, V. Koval, A. Sachenko, and G. Markowsky, "Approach to Recognition of License Plate Numbers Using Neural Networks," in *IEEE Intl. Joint Conf. on Neural Networks*, pp. 2965–2970, Budapest, Hungary, July 2004.

[98] Y.-P. Huang and T. Tsai, "A Practical License Plate Recognition System on PDAs," in *Intl. Computer Symposium*, vol. 26, pp. 731–736, Taipei, Taiwan, Dec. 2004.

[99] Q. Tao, X. He, D. Luo, and W. Wu, "A New Car Plate Recognition Method Based on Fuzzy Entropy," in *Fifth World Congress on Intelligent Control and Automation*, Hangzhou, China, June 2004.

[100] C. Anagnostopoulos1, I. Anagnostopoulos2, and *et al.*, "Using Sliding concentric Windows for License Plate Segmentation and Processing," in *IEEE Intl. Workshop on Signal Processing Systems Design and Implementation*, pp. 337–342, Athens, Greece, Nov. 2005.

[101] J. Matas and K. Zimmermann, "Unconstrained Licence Plate and Text Localization and Recognition," in *IEEE Intl. Conf. on Intelligent Transportation Systems*, pp. 225–230, Vienna, Austria, Sept. 2005.

[102] Y. Takeuchi, H. Fukumoto, Y. Mitsukura, and M.Khalid, "Neural Network Based Threshold Determination for Malaysia License Plate Character Recognition," in *Intl. Conf. on Mechtornics Echnology*, vol. 1, p. 15, Kuala Lumpur, Malaysia, Dec. 2005.

[103] D. Llorens, A. Marzal, V. Palazon, and J. Vilar, "Car License Plates Extraction and Recognition Based on Connected Components Analysis and HMM Decoding," in *Iberian Conf. on Pattern Recognition and Image Analysis*, Estoril, Portugal, June 2005.

[104] T. D. Duan, T. L. H. Du, T. V. Phuoc, and N. V. Hoang, "Building an Automatic Vehicle License-Plate Recognition System," in *IEEE Intl. Conf. on Computer Sciences*, pp. 21–24, Can Tho, Vietnam, 2005.

[105] L. G. C. Hamey and C. Priest, "Automatic Number Plate Recognition for Australian Conditions," in *Intl. Conf. on Digital Image Computing: Techniques and Applications*, p. 14, Cairns, Australia, Dec. 2005.

[106] C. Oz and F. Ercal, "A Practical License Plate Recognition System for Real-Time Environments," in *Intl. work-conference on Artificial Neural Networks*, Barcelona, Spain, June 2005.

[107] C. Wu, L. C. On, C. H. Weng, T. S. Kuan, and K. Ng, "A Macao License Plate Recognition System," in *Intl. Conf. on Machine Learning and Cybernetics*, Guangzhou, China, Aug. 2005.

[108] A. Broumandnia and M. Fathi, "Application of pattern recognition for farsi license plate recognition," *Intl. Journal on Graphics, Vision and Image Processing*, vol. V2, pp. 25–31, Jan. 2005.

[109] C. Anaqnostopoulos, I. Anaqnostopoulos, V. Loumos, and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, pp. 377–392, Sep. 2006.

[110] P. G. Hou, J. Zhao, and M. Liu, "A License Plate Locating Method Based on Tophat-bothat Changing and Line Scanning," in *Intl. Symposium on Instrumentation Science and Technology*, pp. 431–436, Harbin, China, Aug. 2006.

[111] C. Arth, F. Limberger, and H. Bischof, "Real-Time License Plate Recognition on an Embedded DSP-Platform," in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, Minnesota, June 2007.

[112] V. Franc and V. Hlavac, "License Plate Character Segmentation Using Hidden Markov Chains," in *DAGM Symposium for Pattern Recognition*, p. 385, Vienna, Austria, Aug. 2005.

[113] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," *Pattern Recognition*, vol. 38, pp. 1961–1975, Jan. 2005.

[114] Y. Amit, D. Geman, and X. Fan, "A Coarse-to-Fine Strategy for Multiclass Shape Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 12, pp. 1606–1621, 2004.

[115] Y. Hu, F. Zhu, and X. Zhang, "A Novel Approach for License Plate Recognition Using Subspace Projection and Probabilistic Neural Network," in *IEEE Intl. Symposium on Neural Networks*, vol. 3497, pp. 216–221, Chongqing, China, Oct. 2005.

[116] L. Zheng and X. He, "Number Plate Recognition Based on Support Vector Machines," in *IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance*, pp. 13–13, Sydney, Australia, Nov. 2006.

[117] C. Mancas-Thillou and B. Gosselin, "Character Segmentation-by-Recognition Using Log-Gabor Filters," in *IEEE Intl. Conf. on Pattern Recognition*, vol. 2, pp. 901–904, Hong Kong, Aug. 2006.

[118] L. Torres-Mendez, J. Ruiz-Suarez, L. Sucar, and G. Gomez, "Translation, rotation, and scale-invariant object recognition," *IEEE Trans. on Systems, Man and Cybernetics*, pp. 125–130, Feb. 2000.

[119] M. D. Gupta, S. Rajaram, N. Petrovic, and T. S. Huang, "Restoration and Recognition in a Loop," in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 638–644, San Diego, CA, USA, June 2005.

[120] Q. Wu, H. Zhang, and et al., "Car Plate Detection Using Cascaded Tree-Style Learner Based on Hybrid Object Features," in *Intl. Workshop on Intelligent Computing in Pattern Analysis/Synthesis*, Taipei, Taiwan, Oct. 2006.

[121] S. Karungaru, M. Fukumi, and N. Akamatsu, "Malaysian License Plate Recognition Using Artificial Neural Networks and Evolutionally Computation," in *Intl. Federation for Systems Research*, vol. 1, pp. 1–5, Kobe, Japan, Nov. 2005.

[122] M. Guo, L. Wang, and X. Yuan, "Car Plate Localization Using Pulse Coupled Neural Network in Complicated Environment," in *PRICAI 2006: Trends in Artificial Intelligence, 9th Pacific Rim Intl. Conf. on Artificial Intelligence*, vol. 4099, pp. 1206–1210, Guilin, China, Aug. 2006.

[123] R. Juntanasub and N. Sureerattanan, "Car License Plate Recognition through Hausdorff Distance Technique," in *IEEE Intl. Conf. on Tools with Artificial Intelligence*, p. 5, Hong Kong, China, Nov. 2005.

[124] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[125] P. Viola and M. Jones, "Robust Real-time Object Detection," *Intl. Jour. of Comp. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[126] Y. Freund and R. Schapire, "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting," *Jour. of Comp. and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[127] J. L. Johnson, "Pulse-Coupled Neural Nets: Translation,. Rotation,Scale, Distortion, and Intensity Signal. Invariances for Images," *Applied Optics*, vol. 33, no. 26, pp. 6239V–6253, 1994.

[128] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Prentice Hall, second ed., 2002. ISBN: 0201180758.

[129] B. K. P. Horn, *Robot Vision.* MIT Press, first ed., 1986. ISBN: 0262081598.

[130] N. Otsu, "A Threshold Selection Method from Gray-Scale Histogram," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 8, pp. 62–66, 1978.

[131] M. Oren and *et al.*, "Pedestrian Detection using Wavelet Templates," in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pp. 193–199, 1997.

[132] F. C. Crow, "Summed-Area Tables for Texture Mapping," in *ACM Intl. Conf. on Comp. Graphics and Interactive Techniques*, vol. 18, pp. 207–212, 1984.

[133] S.-Z. Wang and H.-J. Lee, "Detection and Recognition of License Plate Characters with Different Appearances," in *IEEE Intl. Conf. on Intelligent Transportation Systems*, vol. 2, pp. 979–984, Shanghai, China, Oct. 2003.

[134] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," vol. 2, no. 2, pp. 121–167, 1998.

[135] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[136] Asia Vision Corp., "ALPR VECON-VIS."

[137] Hi-Tecn Solutions Ltd., "SeeCar Library."

[138] Zamir Recognition Systems Corp., "Zamir's LPR engine."

[139] Adaptive Recognition Hungary Ltd., "Parking LPR Engine."

[140] P. Simard, Y. L. Cun, and J. Denker, "Efficient Pattern Recognition Using a New Transformation Distance," in *Advances in Neural Info. Proc. System*, vol. 5, pp. 50–58, San Francisco, CA, USA, 1993.

[141] D. Keysers, W. Macherey, and et al., "Adaptation in Statistical Pattern Recognition Using Tangent Vectors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 269–274, 2004.

[142] G. Arfken, *Mathematical Methods for Physicist.* Academic Press, fifth ed., 2000. ISBN: 0120598256.