# 國立中央大學

## 資訊工程學系
## 博士論文

# 特徵顏色表示方法及其在物體偵測上之應用
# Eigen Color Representation and Its Applications to Object Detection

研 究 生：蔡洛緯

指導教授：范國清 博士

謝君偉 博士

中華民國 九十八 年 三 月

# 國立中央大學圖書館
# 碩博士論文電子檔授權書

(95 年 7 月最新修正版)

本授權書所授權之論文全文電子檔(不包含紙本、詳備註 1 說明)，爲本人於國立中央大學，撰寫之碩/博士學位論文。(以下請擇一勾選)

(✓)同意 (立即開放)

(　)同意 (一年後開放)，原因是：________________

(　)同意 (二年後開放)，原因是：________________

(　)不同意，原因是：________________

以非專屬、無償授權國立中央大學圖書館與國家圖書館，基於推動「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、微縮、光碟及其它各種方法將上列論文收錄、重製、公開陳列、與發行，或再授權他人以各種方法重製與利用，並得將數位化之上列論文與論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名: 蔡洛緯　　學號： 945402001

論文名稱: 特徵顏色表示方法及其在物體偵測上之應用

指導教授姓名: 范國清

系所 ： 資訊工程 所 ☑博士班 □碩士班

日期：民國 98 年 1 月 9 日

備註：

1. 本授權書之授權範圍僅限電子檔，紙本論文部分依著作權法第 15 條第 3 款之規定，採推定原則即預設同意圖書館得公開上架閱覽，如您有申請專利或投稿等考量，不同意紙本上架陳列，須另行加填聲明書，詳細說明與紙本聲明書請至 http://blog.lib.ncu.edu.tw/plog/ 碩博士論文專區查閱下載。
2. 本授權書請填寫並**親筆**簽名後，裝訂於各紙本論文封面後之次頁（全文電子檔內之授權書簽名，可用電腦打字代替）。
3. 請加印一份單張之授權書，填寫並親筆簽名後，於辦理離校時交圖書館（以統一代轉寄給國家圖書館）。
4. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

# 國立中央大學博士班研究生

# 論文指導教授推薦書

**資訊工程** 學系/研究所 **蔡洛緯** 研究生所

提之論文

**特徵顏色表示方法及其在物體偵測上之應用**

係由本人指導撰述，同意提付審查。

指導教授 范國清 謝君偉 (簽章)

97 年 12 月 29 日

94.11.24

# 國立中央大學博士班研究生

# 論文口試委員審定書

資訊工程 學系/研究所 蔡洛緯 研究生

所提之論文

**特徵顏色表示方法及其在物體偵測上之應用**

經本委員會審議，認定符合博士資格標準。

學位考試委員會召集人 蔡文祥

委　　　員 蔡文祥　賈叢林

曾定章　林淑芳　謝君偉

王俊傑　范國清

謝朝和

中華民國 98 年 1 月 9 日

94.11.16

# 摘要

在電腦視覺領域中，物體偵測是相當基礎且重要的問題。同時可應用在很多方面，例如：視訊監控，導航，影像檢索…等。主要目的是找出物體在影像中的正確位置不論場景如何地變化.

本論文提出一套新穎的系統架構應用於彩色影像中。首先，我們發展出一種稱做特徵值顏色的方法。此方法是透過對某特定物體類別做一統計上的分析所推導得到的結果.在這個新的特徵色彩空間上，前景物像素點可以容易地與背景物的像素點作區分，即使是在一些具有光線變化的場景。至於在候選區塊的確認步驟，我們利用數種重要的物體外觀特徵包含角點、邊緣資訊與小波轉換之係數，來建構一串連且多重維度之物體分類器。依據此串連架構，可以對輸入影像中可能的前景物像素點作有效之確認。由於先前已利用色彩資訊濾除大量無關的背景像素點，故此掃瞄步驟將可快速的執行並找出前景物。

與一般傳統外觀類型的偵測方式相比,我們所提出的特徵色彩空間可以事先過濾大量無關的背景像素點.因此可以有效的快速定位出物體的位置。即使是靜態影像，我們仍舊可以成功的從非固定式的照相機偵測出前景物。我們分別利用車輛與交通號誌的偵測來驗證所提出方法的可行性。實驗結果證明結合特徵色彩資訊與局部外觀資訊之偵測方式是強而有效的。

# Abstract

Object detection is a fundamental and important problem in computer vision and can be applied to various applications like video surveillance, navigation, content-based image retrieval and so on. Its goal is to find the exact location of an object no matter how the environmental conditions change.

This thesis presents a novel framework for detecting objects in color images. First of all, a novel eigen color representation derived from a statistical analysis of object instances is presented. In this new eigen-color space, different object pixels can be easily identified from background, even though they are lighted under varying illuminations. At the hypothesis verification stage, each detected pixel corresponds to an object hypothesis. Several important appearance features including corners, edge maps and coefficients of wavelet transforms were used for constructing a cascade multi-channel classifier. With the cascade structure, an effective scanning process can be performed to verify all possible candidates. Because the color feature eliminates most background pixels in advance, the scanning process can be performed extremely quickly to locate each desired object.

Compared with the traditional appearance-based methods, our proposed eigen-color space can filter out most of impossible candidates in advance and thus each desired object can be very efficiently located from the background. Even thought still images are handled, each object still can be efficiently detected from a non-stationary camera. Two important applications are demonstrated in this thesis; that is, vehicle detection and road sign detection. Experimental results demonstrate that the integration of eigen color feature and local appearance features can form a powerful and superior tool in object detection.

# 誌謝

回首研究所求學的過程，經歷了數千個日子，在漫長的修業過程中，隨著本論文的完成，即將告一階段。不過這並非是結束，反倒是另一個新開始。這段旅程，該從我踏入恩師范國清博士的辦公室開始。還記得當時的范老師仔細的述說著我們的研究領域的應用方向，還聽的一知半解的我，只覺得新奇有趣，毫不猶豫的加入了圖形識別實驗室。在往後的日子裡， 雖然范老師總是事務繁忙，但對於學生的照顧以及關心卻絲毫不減。只要有事去找老師，總可以得到許多方面的想法與解答。范老師為人親切和藹，嚴肅中帶有些許的幽默，對於研究大方向的提點總是恰到好處。范老師的照顧與關心，內心總是無限感激。此外，還要感謝我另一個指導教授，元智大學的謝君偉博士。謝博士對於我在研究上，提供了很多實做方面的寶貴經驗，對於實驗的設計與數據的分析也有著獨到的見解。我想我是個幸運的人，在兩位老師身上各自學習到不同方面的研究精神與方法。也要感謝口試委員們所提供的寶貴意見，不論是論文的撰寫要領，編排的流程順序，以及內容思路的仔細確認，再再都讓我見識到真正嚴謹的研究態度。

此外還要感謝實驗室內有許多一同陪伴走過研究生活的伙伴，與我同窗的幾個學長執彰，映濃，啟宏…除了在學問上的互相討論，私底下也有許多開心的吃喝玩樂活動。加上許多搞笑活潑的學弟妹們：王強，小穎，柚子…有時候無俚頭的對話，在好多個只能寫程式K書的煩悶夜晚，讓人感覺到做研究的路其實並不孤獨。感謝助理小鳳，幫我處理很多文書雜務；感謝PR鐵三角的蛋蛋，阿良與飯粒，陪我運動打籃球；感謝我家可愛的寶貝，在我獲得榮譽時與我分享開心喜悅；在我失意灰心時給予我精神上的鼓舞。此外，也要感謝一直支持我的家人，讓我沒有後顧之憂的一直在求學的路上前進。最後，僅以此論文，獻給所有關心和支持鼓勵我的人。

# Index

# List of Figure

# List of Table

# Chapter 1. Introduction

Automatic object detection in unstructured environments where illumination varies dynamically has been a great challenge in pattern recognition and computer vision research. It is also the first step in all monitoring systems that the objects is detected and examined. It determines most important quality of accuracy in the system. After detecting and examining the moving objects accurately, tracking and analysis of the object's behavior are relatively simpler.

## 1.1 Motivation

To detect objects from their background, the most commonly used feature is motion. When the camera is static, this feature can be easily extracted using the techniques like image differencing or background subtraction [4], [5]. The former technique usually will extract a moving object with many holes and the latter one will fail in object detection when illuminations change large. More importantly, the two techniques will fail to deal with static objects since motion features no longer exist. When static objects are handled, the appearance-based method [23]-[27], [29] will be better adopted. For example, Sung and Poggio [24] proposed an example-based method to train a detector from lots of training samples for face detection. Papageorgiou and Poggio [25] adopted the similar example-based learning technique to detect people in complex scenes without using any motion information. In [27] Agarwal and Roth proposed a part-based representation for body part extraction. Their framework selected distinguishable parts of objects and learned the discriminative classifier over the parts and then detected each desired object from images. The above method can detect objects only from a fixed view. To tackle the multiple-view problem, Schneiderman and Kanade [29] proposed a wavelet-based method for training a detector to detect faces or cars no matter what their viewpoints are. For the above example-based method, a lot of positive and negative training examples should be collected at the training stage. Then, at the detection stage, a window with a fixed size is used to scan all pixels in the input image to locate all possible candidates. To tackle the size variation of an object, the input image will be scaled to different resolutions and then a very time-consuming scanning is performed from left to right and from up to down to locate each desired object. To improve the detection efficiency, Viola and Jones [22] presented a cascade structure to train a classifier for detecting object in real time. This scheme takes advantages of integral image and Adaboost algorithm to filter out all impossible candidates extremely efficiently. Their major contribution is the idea of cascade structure which can avoid lots of feature verification for detecting objects in real time. This technique also lacks capabilities for detecting objects from multiple viewpoints. Li *et al*. [23] extended this boosting idea called “Float Boost” for real-time multi-view face detection. Furthermore, Zhu *et al*. [28] used corners and edge densities of a vehicle to define two special templates called EAT and CAT for vehicle detection. However, it may fail when highly textured images are handled.

As mentioned before, objects will include larger appearance variations like their colors, sizes, and shapes changing according to their different viewing positions, lighting conditions, and cluttered background. All the variations will increase many difficulties and challenges in selecting a general feature for describing an object. The general feature can be global or local for well describing an object under different conditions. For the global feature like skin color, it can be used for filtering out impossible candidates in advance. For example, we can use neural networks to learn a skin color classifier for skin color detection and then detect each face candidate from still images at the coarse stage. At the fine stage, the local feature can be then used for verifying candidates more accurately. For example, features like edge fragments, corners, or wavelet features can be used for object description and verifying an object more accurately. When an object (like face) owns its specially color, its global feature can be easily defined and extracted for object detection. However, for objects like vehicles or road signs, since their colors are not fixed, it is very difficult to define a global feature for object representation and thus narrowing down the search areas of possible candidates. If a general color transform can be found for object representation (even objects lighted under different conditions), the color will become a very useful cue to filter out impossible candidates more efficiently.

This thesis presents a novel color-based algorithm for detecting static objects directly from images by first locating possible candidates using their colors and then combining different appearance features together to form a cascade classifier for candidate verification. The contribution of this thesis is to present a statistic method for deriving an eigen-color space that makes object colors more sufficiently concentrated on a compact area. The model is leaned by observing how the object colors change in static images under different lighting conditions and cluttered backgrounds. In this space, a classifier can be then designed for searching possible candidates from images. Since this classifier can filter out most of background candidates, only few candidates should be further checked. Due to the filtering effect and discriminative abilities of the proposed method, desired objects can be very effectively detected from static images. We focus on the problem of detecting specific objects which commonly appear in applications like surveillance systems, driver assistance systems, and image retrieval. In these applications, two common targets repeatedly appear in the analyzed scenes; that is, vehicles and traffic signs. In Section 1.2, a brief review of these two targets will be given.

# 1.2 Review of Related Works

## *1.2.1 Previous Methods for Vehicle Detection*

One of the major issues in object diction is vehicle detection [1]-[11]. It has many related applications such as self-guided vehicles, driver assistance system, intelligent parking system, or measurement of traffic parameters like vehicle count, speed, and flow. One of most common approaches to vehicle detection is using vision-based techniques to analyze vehicles from images or videos. However, due to the variations of vehicle colors, sizes, orientations, shapes, and poses, developing a robust and effective system of vision-based vehicle detection is very challenging. To address the above problems, different approaches using different features and learning algorithms for locating vehicles have been investigated. For example, many techniques [2]-[6] used background subtraction to extract motion features for detecting moving vehicles from video sequences. However, this kind of motion feature is no longer usable and found in still images. For dealing with static images, Wu *et al.* [7] used wavelet transform to extract texture features for locating possible vehicle candidates from roads. Then, each vehicle candidate is verified using a PCA (principal component analysis) classifier. In addition, Z. Sun *et al*. [8] used Gabor filters to extract different textures and then verified each candidate of vehicles using a SVM (support vector machines) classifier. In addition to textures, “symmetry” is another important feature used for vehicle detection. In [9], Broggi *et al*. described a detection system to search for areas with a high vertical symmetry for locating vehicles. However, this cue is prone to false detections such as symmetrical doors or other objects. Furthermore, in [10], Bertozzi *et al.* used corner features to build four templates of vehicles for vehicle detection and verification. In [11], Tzomakas and Seelen found that the area shadow underneath a vehicle is a good cue to detect vehicles. In [12], Ratan *et al*. developed a scheme to detect vehicles’ wheel features as cues to find possible vehicle positions and then used a method called Diverse Density to verify each vehicle candidate. In addition, Bensrhari [13] and Aizawa [14] used stereo-vision methods and 3D vehicle models to detect vehicles and obstacles. The major drawback of the above methods to search vehicle using local features is the need of a fully time-consuming search to scan all pixels of the whole image. For the color feature, although color is an important perceptual descriptor to describe objects, there were seldom color-based works addressed for vehicle detection since vehicles have very large variations in their colors. In [15], Rojas and Crisman used a color transform to project all road pixels on a color plane such that vehicles can be identified from road backgrounds. Similarly, in [16], Guo *et al*. used several color balls to model road colors in L*a*b* color space and then vehicle pixels can be identified if they are classified no-road regions. However, since these color models are not compact and general in modeling vehicle colors, many false detections were produced and leaded to the degradation of accuracy of vehicle detection.

### *1.2.2 Previous Methods for Road Sign Detection*

Road sign detection is an important and essential task in an intelligent driver support system. The texts embedded in a road sign usually carry much useful information like limited speed, guided direction, and current traffic situations for helping the drivers drive safely and comfortably. However, it is difficult to detect road signs directly from videos due to different environmental condition changes. For example, a road sign will have different appearance changes including its lightings, colors, or shadows under different days, seasons, and weathers. In addition, for the camera mounted in front of a moving car, the perspective effects will make a road sign have different sizes, shapes, contrasts, and motion blurs. In some cases, it would be occluded with other objects like trees. To tackle the above problems, there have been many works [32]-[39] proposed for automatic road sign detection and recognition via a vision-based technique. Since a road sign usually has a high-contrast color to its background and a regular shape, we can divide these approaches into two categories, i.e., color-based and shape-based. For the color-based approach [32], Bénallal and Meunier found that the difference between *R* and *G*, and the difference between *R* and *B* channels can form two stable features for road sign detection in day time. In [33], Escalera *et al*. used a color threshoding technique to separate road sign regions in the (*R*, *G*, *B*) channels from the background. In addition to the RGB space, other color spaces like YIQ and HSV are also good for road sign detection. For example, in [36], Kehtarnavaz and Ahmad used a discriminant analysis on the YIQ color space for detecting desired road signs from the background. As to the HSV color space, in [37], [53], Vitabile *et al.* presented a sub-space dynamic thresholding technique to find a road sign color like red so that all possible road sign candidates can be detected. They further improved their system using FPGA language for hardware implementation. Moreover, Fleyeh [52] used an improved HLS (Hue, Lightness, Saturation) color space for detecting color road signs from road scenes. Since a road sign has different colors (like red, blue, or green) for demonstrating its functionalities like warning or direction messages, different detectors should be designed for tackling its color variations.

In addition to color, shape is another important feature for detecting road signs. In [44], Barnes and Zelinsky adopted the radial symmetry feature to detect possible road signs and then to verify them using a correlation technique. In [45], Piccioli *et al*. proposed a template matching scheme to search all possible road signs from images. In addition, Wu *et al.* [43] used the corner feature and a vertical plane criterion to cluster image data to different categories so that each road sign candidate can be found. Moreover, Blancard [48] used an edge linking technique and the contour feature to locate road sign candidates and then verified them according to their perimeters and curvature features. The shape feature also can be learned from a set of training samples. In [46], Haritaoglu and Haritaoglu used texture features and support vector machines to detect road signs in a scene. Garcia-Garrido *et al*. [55] extended the Hough transform to find any curves in an image for detecting circular and stop signs. Usually, different shapes of a road sign represent different warning or directional messages. If only the shape feature is used, different shape detectors should be designed for detecting different road signs and will make the detection process become very time-consumed. Therefore, there are some hybrid methods proposed for road sign detection. For example, Bahlmann *et al*. [47] used a color representation, integral features, and the AdaBoost algorithm for training a stronger classifier to detect road signs from videos in real time. Furthermore, Fang *et al*. [34]used fuzzy neural networks and a gradient feature to locate and track road signs and then incorporated a geometry model of road signs for road sign verificaiton. In [45], Piccioli *et al*. used a clustering technique to extract "red" regions as road sign candidates and then verified them using a set of shape features. In [38], Kastinaki *et al*. used local color, texture features, and a conditional maximum entropy model to detect road signs. Then, all candidates are verified by matching them against a set of predefined road sign templates. Moreover, Escalera *et al.* [54] integrated three features, i.e., chromatic image, gradient and distance energy, to more accurately detect road signs from road scenes and then recognize them using the genetic algorithm and simulated annealing. For a good shape-based approach, it should have good abilities to overcome the shape variations and occlusions of a road sign when it is captured using a moving camera.

# 1.3 Overview of Approach

## *1.3.1 Vehicle Detection system*

The flowchart of the vehicle detection system is shown in Fig. 1. At the beginning, we propose a color transformation to project all the colors of input pixels on a color space such that vehicle pixels can be easily identified from backgrounds. Here, Bayesian classifier and radial basis function (RBF) network are used for this identification. Then, each detected vehicle pixel will correspond to a possible vehicle. Since vehicles have different sizes and orientations, different vehicle hypotheses are generated from each detected vehicle. For verifying each hypothesis, we use three kinds of vehicle features to filter out impossible vehicle candidates. The features include edges, coefficients of wavelet transform, and corners. Using proper weights obtained from a set of training samples, these features can be then combined together to form an optimal vehicle classifier. Then, desired vehicles can be very robustly and accurately verified and detected from static images.

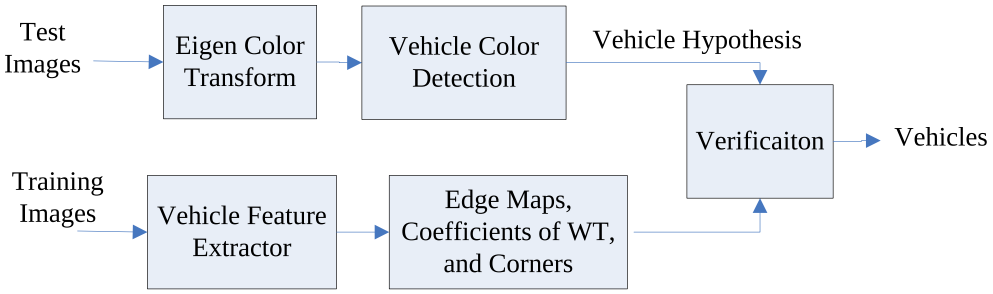


Fig. 1: Flowchart of the proposed vehicle detector.

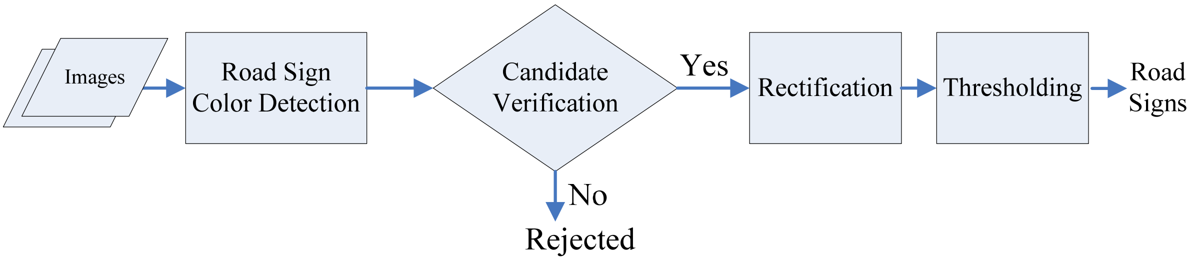


Fig. 2: Flowchart of the proposed road sign detection system.

### *1.3.2 Road Sign Detection system*

Fig. 2 illustrates the flowchart of the proposed system to detect road signs. The system assumes that the camera is mounted in the front of the car for capturing different video sequences. The camera optical axis direction is not required being perpendicular to the image plate of road signs. Four major modules are included in the proposed system; that is, road sign candidate extraction, verification, rectification, and thresholding, respectively. Details of each component are described below:

(a) Road sign candidate extraction: In order to quickly locate the road signs, a novel eigen color model is proposed to detect road sign candidates from their backgrounds. The model is learned from thousands of road sign images. Then, using this model and RBF (radial basis function) networks, only one detector is needed for extracting different road sign candidates even though their colors are different.

(b) Verification: Once all potential road signs have been selected, a verification procedure is then proposed for candidate verification. The process will build a set of road sign templates for reducing the perspective effects. Then, each candidate is converted into a distance map so that impossible road sign candidates can be filtered out.

(c) Rectification：After verification, a rectification process is then applied for more accurately recognizing a skewed road sign.

(d) Thresholding: Once a road sign is extracted, a thresholding scheme is then adopted for binarizing it so that different texts embedded in it can be extracted.

## 1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows. In chapter 2, we describe the proposed eigen color method and its corresponding eigen color model to vehicle and road sign. The design of the color detector using different learning engines is also given. In chapter 3, we will introduce several appearance features used in our object recognition module. We also present the object verification module with cascade structure in this section. In chapter 4, we provide experimental results in object detection under different environments and the accuracy of the color pixels classification. Comparison results with other sophisticated methods are also discussed. Finally, we give the conclusions and list the future works in chapter 5.

# Chapter 2. Eigen Color Detector

Color information is a useful feature as pre-processing in object detection. For example, face and fire detection use the skin and fire color as detection cues. In general, skin and fire candidate regions are easy to extract and model in image by its color distribution at RGB color space. However, for the specific objects class (like vehicles), it is difficult and complex to model the color distribution due to the large variation. This section will introduce a new color transformation for mapping all pixels in images from ($R$, $G$, $B$) color space to a new domain. By extracting the useful information to form a new feature space, feature energy of the object pixels are compacted and easy to separate from the background pixels. Then, a specific "object color" can be found and located for effective object detection.

## 2.1 Karhunen-Loe`ve Transform

Dimensionality reduction is a useful skill in pattern recognition. Too many features lead to more computation load and create confusion such as to decreasing the classifier performance. By selecting a subset of features from the original data, the high dimensional data still have its main distinguishing characteristic. That is to reduce the dimensionality of a high dimensional data without significant loss in accuracy. In practice, high-dimensional data are often loose without tight clusters. Human beings can not realize the shape, density of the data in high dimension. Projecting data into lower-dimension space makes data clusters easy to observe by human eyes. By projecting data onto an appropriate lower-dimensional space (feature space), data clusters would have a local structure that makes the close neighborhood meaningful.

Karhunen-Loe`ve transform (K-L transform) is a well-know and widely used technique for statistical analysis. It has various names, such as principal component analysis, the eigenvector transform or the Hotelling transform. This method usually was adopted in feature extraction, data compression, image processing…etc. K-L transform tries to describe the data as good as possible in a lower dimensional space. Assume the data set contains *N* samples and $x_i$ be an *n*-dimensional vector. The algorithm maps the *n*-dimensional patterns onto an *m*-dimensional space, where *m*<*n*. We have to compute a transformation matrix **H** which is constructed by the eigenvector of the covariance matrix. The computation of the transformation matrix is as the following algorithm:

**Step 1** : Let $m^*$ denote the mean and $\overline{C}$ denote the covariance matrix respectively.

$$m^* = \frac{1}{N}\sum_{k=1}^{N} x_k \ ,$$

$$\overline{C} = \frac{1}{N}\sum_{i=1}^{N}(x_i - m^*)(x_i - m^*)^T .$$

**Step 2** : Compute the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_d$ and construct the associate

eigenvector $e_1, e_2, ..., e_d$ of $\overline{C}$ . Sort them as $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$ .

**Step 3** : Form the matrix **H** = $[e_{1,} e_2, ..., e_d]^T$.

After transformation, the covariance matrix of the feature becomes a diagonal matrix. This matrix projects the input data into a subspace whose axes are in the direction of the largest variation as follow:

**Step 4** : $y_i = Hx_i$ for $i = 1, ..., N$

## 2.2 Color Feature Extraction

In this section, we present the detail procedure about the eigen color extraction. Assume that there are *N* training images collected from various natural scenes. We choose the RGB color space and the *i*th pixel *x* in the selected image can be represented as:

$$x_i = [x_{i,r} \ x_{i,g} \ x_{i,b}].$$

Through a statistic analysis, we first compute the covariance matrix $\Sigma^*$ of the color distributions of *R*, *G*, and *B* from these *N* images. Using the Karhunen-Loe've (KL) transform, the eigenvalues and the corresponding eigenvectors of $\Sigma^*$ can be further obtained and represented as $\lambda_i$ and $e_i$ respectively, where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ for $i$ = 1, 2, and 3. Then, we selected the eigenvectors to form three new color features $C_i$ by a linear combination of the each RGB components, which can be defined as:

$$C_i = e_i^r R + e_i^g G + e_i^b B \text{ for } i\text{=1, 2, and 3,} \tag{1}$$

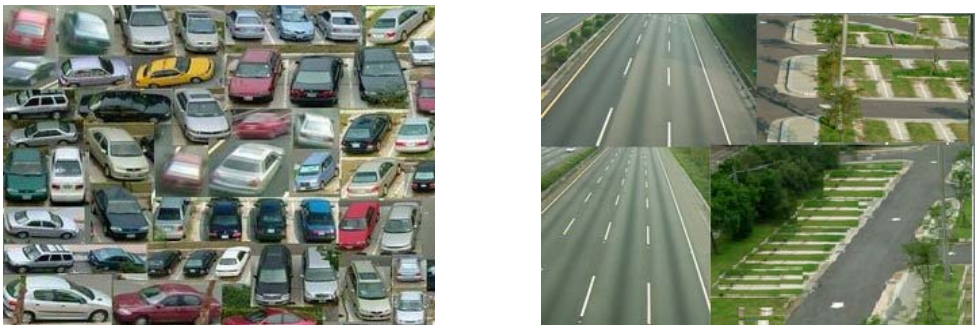
where $e_i = (e_i^r, e_i^g, e_i^b)$. In [17], Ohta *et al*. used the above principal component analysis for region segmentation and indicated the color feature $C_1$ with the largest eigenvalue is the one used for color-to-gray transform, i.e.,

$$C_1 = \frac{1}{3}R + \frac{1}{3}G + \frac{1}{3}B \ . \tag{2}$$

Other two color features $C_2$ and $C_3$ are orthogonal to $C_1$ and have the following forms:

$$C_2 = \frac{\mathrm{R} - B}{2} \quad \text{and} \quad C_3 = \frac{2G - R - B}{4} \ . \tag{3}$$

All the color features can be obtained by projecting the pixels' gray values of red, green, and blue components onto a color space which is expanded with the three vectors (1/3, 1/3, 1/3), (1/2, 0, -1/2), and (-1/4, 1/2, -1/4). In [18], Healey used the similar idea for image segmentation and pointed out that the colors of homogeneous dielectric surfaces (like roads or clouds) will move close along the axis directed by Eq.(2), i.e., (1/3, 1/3, 1/3) . In [15], Rojas and Crisman also found that the colors of roads will concentrate around a small cylinder along the axis directed by Eq. (2). In other words, if we project all the road colors to a plane which is perpendicular to the axis pointed by $C_1$, all the road colors will concentrate around a small circle [15]. The above principal component analysis (PCA) gives us an inspiration to analyze object colors. We use an example-based learning approach to derive a color model of an object class. From a training set of examples collected in different lighting environments, the derived color features can cope with small illumination variances.



(a) (b)

Fig. 3: Parts of vehicle training samples. (a) Vehicle images. (b) Non-vehicle training images.

## 2.3 Eigen Color Model

### *2.3.1 Vehicle Color Model*

At the beginning, the vehicle training images are collected from different scenes including roads, parking lots, building, and natural scenes. Fig. 3 shows parts of our training samples. (a) is the set of vehicle images and (b) the examples of no-vehicle images. Based on the training samples, using the *KL* transform, we found that the eigenvector with the largest eigenvalue of this data set is (1/3, 1/3, 1/3) (the same as in Eq. (2)). In addition, the color plane $(u,v)$ perpendicular to the axis (1/3, 1/3, 1/3) expanded by other two eigenvectors is:

$$u_p = \frac{2Z_p - G_p - B_p}{Z_p} \text{ and } v_p = Max\{\frac{B_p - G_p}{Z_p}, \frac{R_p - B_p}{Z_p}\}, \tag{4}$$

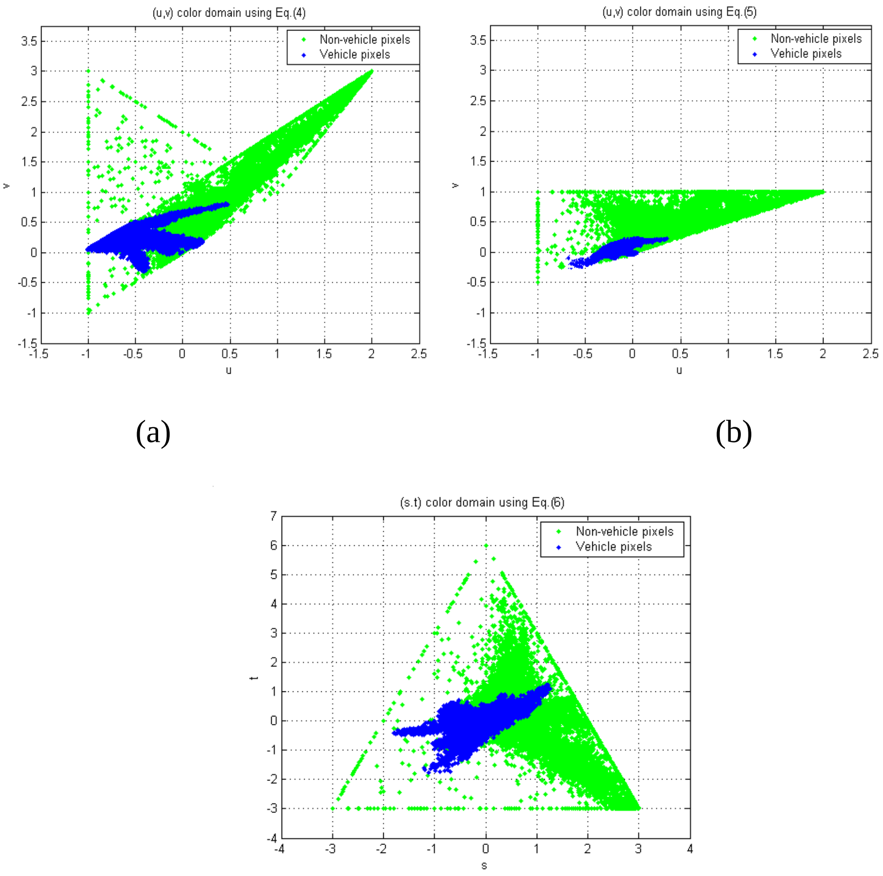
where ( $R_p$ , $G_p$ , $B_p$ ) is the color of a pixel $p$ and $Z_p = (R_p + G_p + B_p)/3$ used for normalization. In practice, the noise existing in the training samples will disturb the accuracy of the orthogonal basis (described in Eq.(4)) to find correct vehicle colors. Thus, many false alarms in detecting vehicle colors will be found if Eq.(4) is directly used. Actually, in color image processing, for the *R*, *G*, and *B* channels, if they are used separately, each of them will be more easily affected by noise than their composition, i.e., (*R*+*G*+*B*)/3. Therefore, we can replace the components $B_p$ and $R_p$ in Eq.(4) with the gray component $Z_p$ to reduce these false alarms. According to this idea, the following is a new color transform created for vehicle color detection:

$$u_p = \frac{2Z_p - G_p - B_p}{Z_p} \text{ and } v_p = Max\{\frac{Z_p - G_p}{Z_p}, \frac{Z_p - B_p}{Z_p}\}. \tag{5}$$

The color transformation described in Eq.(5) will concentrate all vehicle pixels on a smaller area than Eq.(4). There are also other color planes perpendicular to the axis (1/3, 1/3, 1/3). For example, if the training images are collected only from road images, another color plane (*s, t*) perpendicular to the axis (1/3, 1/3, 1/3) can be found, i.e.,

$$s_p = \frac{R_p \text{ - } B_p}{Z_p} \text{ and } t_p = \frac{\text{-}R_p\text{+}2G_p \text{ - } B_p}{Z_p}. \tag{6}$$

However, the color space $(u,v)$ has better discrimination abilities to extract vehicle pixels from the background than the color space *(s, t)*. We can plot all the colors of vehicle pixels of training images on the $(u, v)$ and $(s, t)$ planes using Eq.(5) and Eq.(6), respectively. Fig. 4 shows the results of color transformation using Eqs.(4)-(6), where (a), (b), and (c) are the results using Eq.(4), (5), and Eq.(6), respectively. The variances of pixel distributions of (a)-(c) are listed in Table 1.



(a)

(b)

(c)

Fig. 4: Plots of the results of color transformations of vehicle pixels. (a) Result of color transformation using Eq.(4). (b) Result of color transformation using Eq.(5). (c) Result of color transformation using Eq.(6).

We adopt the Fisher criterion [19] to evaluate the separation abilities of each transform in the classification of vehicle pixels. The criterion uses the ratio of the “between-class” variance to the “within-class” variance to measure how well a transform $T$ can separate a space into two classes $C_1^*$ and $C_2^*$. The “between-class” variance is the distance between the means of two classes (denoted by $m_1$ and $m_2$, respectively). The “within-class” variance is the sum of their variances, i.e. $s_1$ and $s_2$. Then, the Fisher criterion is defined by

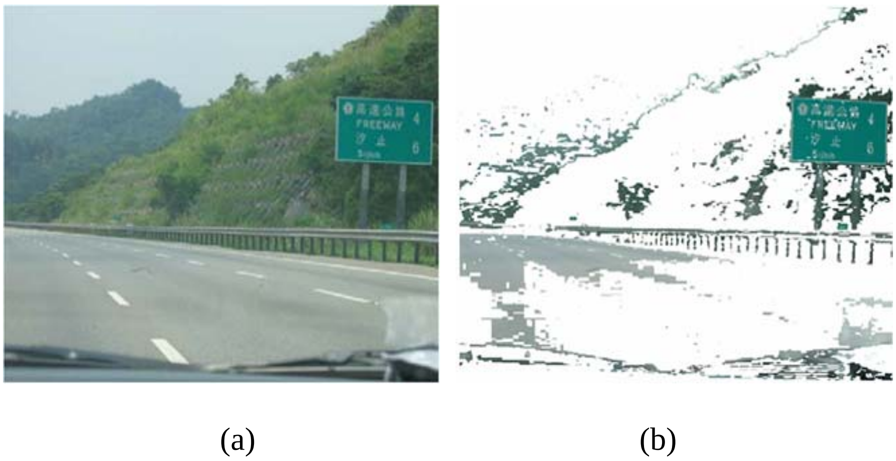
$$J = \frac{\text{between-class distance}}{\text{within-class distance}} = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2}.$$

The larger the ratio $J$ is, the more separation ability the transform $T$ has. Table 1 lists the values of $J$ when Eqs. (4), (5), and (6) are used. Clearly, the color space $(u, v)$ defined in Eq. (5) has better discrimination abilities to separate the vehicle pixels from background pixels than other spaces defined in Eq. (4) and Eq.(6), respectively. The ability of these methods to detect vehicle pixels will be further compared in the experimental section.

Table 1: Separation ability analysis among different color transformations.

| Color Transform | Clusters centers | | Cluster‘s variances | | Distance Between Centers | *J(T)* |
|---|---|---|---|---|---|---|
| | Vehicle | Non-vehicle | Vehicle | Non-vehicle | | |
| Eq. 4 | (-0.05, 0.13) | (0.27, 0.58) | 0.37 | 0.73 | 0.55 | 0.4516 |
| Eq. 5 | (-0.05, 0.05) | (0.27, 0.3) | 0.29 | 0.48 | 0.41 | 0.5345 |
| Eq. 6 | (-0.03, 0.22) | (0.53 , 0) | 0.65 | 0.99 | 0.6 | 0.2567 |

Given an input image, we first use Eq. (5) to project all color pixels on the (*u*, *v*) space. Then, the problem of vehicle detection becomes a 2-class separation problem which tries to find a best decision boundary from the (*u, v*) space such that all vehicle pixels can be well separated from the non-vehicle class. Section 2.4 will presents two kinds of classifiers for classifying vehicle pixels in the (*u, v*) space. The first is a Bayesian classifier and the other one is trained by the radial basis function network.



(a) (b)

Fig. 5: Green color detection in the HIS color space. (a) Original image including a road sign and various green background objects. (b) Result of green color detection in the HIS color space.

### 2.3.2 Road Sign Color Model

Since a road sign has a specific color, we can design a detector to search this color for finding different road sign candidates. For example, in Fig. 5(a), the road sign has a specific “green” color. Then, we can create a green color detector to detect all green objects. However, in real scenes, many irrelevant green objects like tree, mountain, or grass will also be detected. Like Fig. 5(b), after simple green color classification, many non-road-sign objects were also detected. Hence, precise color modeling method is necessary for road sign detection.

This thesis assumes that all the road signs are made of kinds of metal or plastic material with smoother and flatter surfaces. Due to the smooth and flat surface of a road sign, its reflectance property will be very different to (or higher than) other objects (like trees, mountains, buildings) in the analyzed road scene. Then, our idea is to design a novel color transform model for detecting the pixels with higher reflectance from their backgrounds. After the transformation, these pixels will form a connected region. Through a connected component analysis, different road sign candidates can be then extracted from videos for further verification and recognition.

In our experiments, there were 280 road sign samples collected for deriving these eigenvectors. After calculations, the three eigenvectors are given as follows: (0.3396, 0.3392, 0.3212), (0.4896, 0.0923, -0.4181), and $(0.2898, -0.4823, 0.2279)$. The color feature $C_1$ with the largest eigenvalue is the one used for color-to-gray transform and is approximated as follows:

$$C_1 = \frac{1}{3}R + \frac{1}{3}G + \frac{1}{3}B. \tag{7}$$

In addition, the color plane $(u, v)$ perpendicular to the axis (1/3, 1/3, 1/3) expanded by the other two eigenvectors is defined as follows:

$$u = \frac{2(R-B)}{\Omega} \text{ and } v = \frac{R+B-2*G}{\Omega}, \tag{8}$$

where $\Omega$ is a normalized factor.

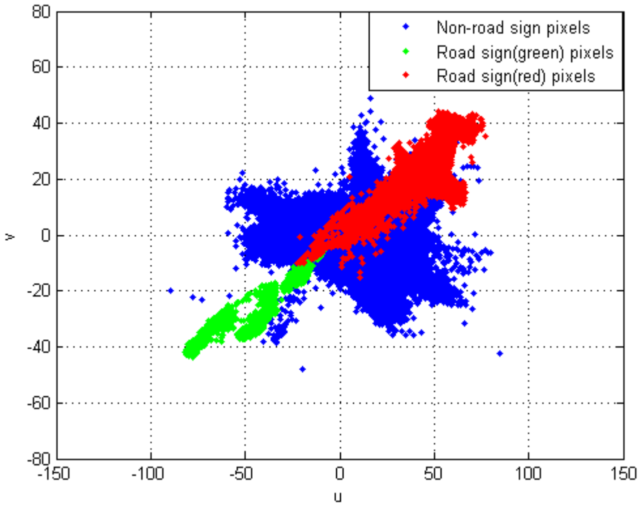


Fig. 6: Color projection using Eq.(8). Blue color means the background pixels. The green region is obtained from guide signs. The red region is obtained from regulatory signs.

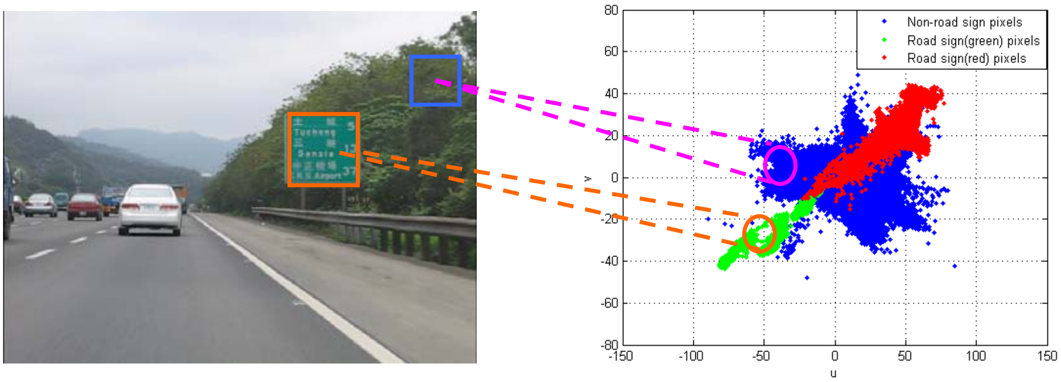


Fig. 7: Result of color projection. (a) Original image. (b) Result of color projection using Eq.(8).

Fig. 6 shows the projection result of road sign pixels and non-road sign pixels using Eq.(8). Here, the green and red regions denote the projection results of pixels in green road signs and red ones, respectively. We also re-project the results of two green regions which are trees and green road signs, respectively as shown in Fig. 7. Although these two regions are "green", they can be easily separated on the (u, v) space if a proper classifier is designed for finding the best separation boundary.

## 2.4 Training Color Detector

In order to effectively locating object from background, we propose a new color model to directly model object colors as described in the previous sections. Then, similar to skin color detection, we can use this color model to transform all pixels on a 2-D color space. On this color space, all pixels will concentrate on a much distinctive area. By modeling the characteristics of this area, the learning engines we used are Bayesian classifier and radial basis function (RBF) network.

### 2.4.1 Bayesian Classifier

In statistical pattern recognition, we focus on how to developing decision or classification strategies which form classifiers. The design of classifier attempts to integrate all available information such as measurement of a priori probabilities of data. Then, the classifier minimizes the total expected loss and using Bayes' formula as the optimum measure of performance. The class-conditional density function of probability of a pattern ***x***, when ***x*** belongs to class $w_i$ , can be given as follow:

$$p(x/w_i), i=1,2,...,M$$

All the class-conditional densities are completely know a prior, the decision boundary between pattern classes can be established using the optimal Bayes decision rules. By way of introduction, consider a vector ***x*** with Gaussian distribution, the probability density function of ***x*** is:

$$P(x)=\frac{1}{\sqrt{2\pi}*\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right]$$

$m$ and $\sigma$ are mean and standard deviation respectively. We can get the decision boundary function is:

$$d_i(x)=\ln p(w_i)-\frac{1}{2}\ln\left|C_i\right|-\frac{1}{2}\left[\left(x-m_i\right)^T C_i^{-1}\left(x-m_i\right)\right], i=1,2,...,M$$

**C** is covariance matrices. The Bayes classifier assigns a pattern ***x*** to class $w_i$ if

$$p(x/w_i)p(w_i)>p(x/w_j)p(w_j), j=1,2,....M, j\neq i$$

The detail derivate procedure can be found in [58].

As mentioned in Section 2.3, after transformation, we design a Bayesian classifier for accurately identifying object pixels from backgrounds with colors. We assume that the RGB color components in the (*u*, *v*) domain forms a multivariate Gaussian distribution. Assume that $m_v$ and $m_n$ are the mean colors of the vehicle and non-vehicle pixels calculated from our collected training images in the (*u*, *v*) color domain, respectively. In addition, $\sum_v$ and $\sum_n$ are their corresponding covariance matrices in the same color domain, respectively. Then, given a pixel *x*, we define its probability belonging to a object pixel as a normal distribution:

$$p(x|object) = \frac{1}{2\pi\sqrt{|\Sigma_v|}}\exp(\text{-}d_v(x)), \tag{9}$$

where $d_v(x) = \frac{1}{2}(x - m_v)\Sigma_v^{-1}(x - m_v)^t$. Similarly, the probability of *x* belonging to a non-object pixel is defined as follows:

$$p(x|non\text{-}object) = \frac{1}{2\pi\sqrt{|\Sigma_n|}}\exp(\text{-}d_n(x)), \tag{10}$$

where $d_n(x) = \frac{1}{2}(x - m_n)\Sigma_n^{-1}(x - m_n)^t$. According to the Bayesian classifier, we can assign a pixel *x* to the class "object" if

$$p(\text{object}|x) > p(non\text{-}object|x). \tag{11}$$

Eq.(11) can be further rewritten as follows:

$$p(x \mid object)P(object) > p(x \mid non\text{-}object)P(non\text{-}object), \tag{12}$$

where $P(object)$ and $P(\text{non-objec})$ are the priori class probabilities of object pixels and non-object ones, respectively. Plugging Eqs. (9) and (10) into Eq.(12) and taking its log form, we have the following classification rule:

$$\text{Assign a pixel } x \text{ to class "object" if } d_n(x)\text{-}d_v(x) > \lambda, \tag{13}$$

where $\lambda = \log[\sqrt{\frac{|\Sigma_{\text{v}}|}{|\Sigma_{\text{n}}|}}\frac{P(non\text{-}object)}{P(object)}]$.

### 2.4.2 Radial Basis Function Network

In addition to the Bayesian classifier, we also use the radial basis function (RBF) network to classify object pixels. As shown in Fig. 8, the RBF network we used includes an input layer, one hidden layer, and an output layer. Each hidden neuron is associated with a kernel function by the form:

$$\varphi_j(x) = \exp(-\frac{\| x - m_j \|^2}{2\sigma_j^{\ 2}}),$$

where $x$ is an $n$-dimensional input feature vector, $m_j$ and $\sigma_j$ represent the center and the width of the $j$th hidden neuron. Each output neuron is approximated using a linear combination of kernel functions, i.e.,

$$\psi_i(x) = \sum_{j=1}^{R} w_{ij}\varphi_j(x), \text{ for } i=1, \ldots, C,$$

where $w_{ij}$ is the connection weight between the $j$th hidden neuron and $i$th output layer neuron, and $C$ the number of outputs. When classifying, the output of the radial basis function is limited to the interval (0, 1) by a sigmoid function:

$$F_i(x) = \frac{1}{1 + \exp(\text{-}\psi_i(x))}.$$

The parameters $w_{ij}$ of the RBF networks are computed by the gradient descent method such that the cost function is minimized:

$$E = \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{C}(y_i(x_k) - F_i(x_k))^2,$$

where $N$ is the number of inputs and $y_i(x_k)$ denotes the $i$th output associated with the input sample $x_k$ from the training set. Then, if a pixel belongs to the object class, it will be labeled to 1; otherwise, 0. When training, all pixels in (*R*, *G*, *B*) domain are transformed to (*u*, *v*) domain using Eq.(5) or Eq.(8) according to different object classes (i.e. vehicles and road signs).

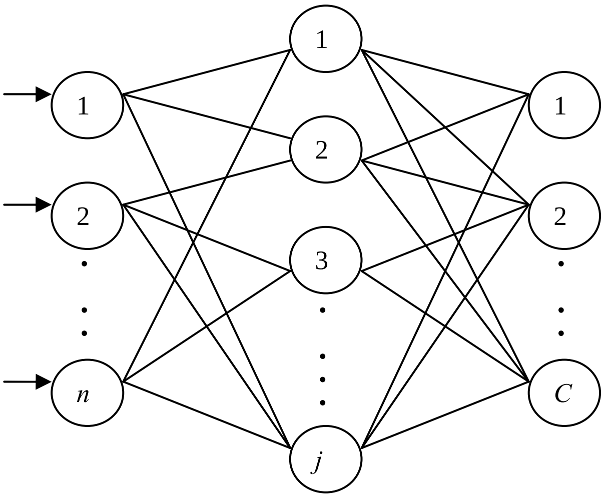


Fig. 8: Structure of the RBF network

Compared with other classification algorithms like Adaboositng [26] or SVM (support vector machines)[23], the RBF network is simpler and has limited performances in data classification. However, our focus is to prove the superiorities of our proposed color transform to detect vehicle pixels even if only a simple classifier is used. The comparisons between the RBF network and the Bayesian classifier for detecting object pixels will be performed in the experimental section.

# Chapter 3. Object Verification

In the previous section, the eigen color model and two classifiers were presented for extracting object pixels from static images. Each detected pixel will correspond to a possible candidate. Then, this section will present a verification scheme for verifying each candidate more accurately. In some cases, due to noise, some candidate pixels will be lost and cannot be recovered from the color classification process. The problem can be easily handled using a morphological dilation operation [51] for generating more candidate regions. In this technique, if a pixel passes the classification stage, we generate not only a hypothesis but also its neighborhoods as possible candidates. Then, even though some pixels are lost, their corresponding real pixels still can be found using the extending technique.

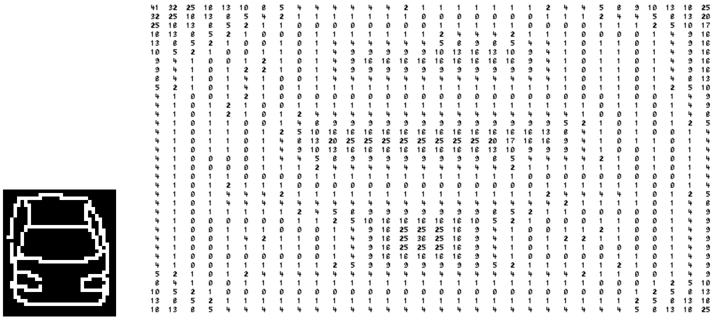
## 3.1 Object Hypothesis

Given a vehicle pixel X, the verification process will first generate different hypothesizes with different sizes for tackle the size variations of vehicle appearances. Here, a hypothesis $H_s^I(X)$ is a sub-image extracted from the input image *I* with the size $w_s \times h_s$ at the center X. The minimum vehicle size used in this thesis is assumed to be $36 \times 36$. To generate the set of hypotheses more efficiently, we can gradually reduce *I* into a series of smaller images with the scale factor 0.9 so that a pyramid structure is constructed. Each layer in this structure is a smoothed image $I_s$ having the size $0.9^s w_I \times 0.9^s h_I$, where $w_I$ and $h_I$ are the width and height of *I*. In the pyramid structure, a vehicle hypothesis $H_s^I(X)$ having the size $w_s \times h_s$

in $I$ will become a $36\times36$ sub-pattern in $I_s$. Thus, in what follows, the hypothesis $H_s^I(X)$ is verified all based on the $36\times36$ sub-pattern in $I_s$. More details about the pyramid structure will be discussed in Section 3.2.5.

In order to verify the correctness of $H_s^I(X)$, we build a set of classes $C_{\theta_i}$ of vehicle templates for estimating its maximum vehicle response at different orientations. Here $C_{\theta_i}$ is a collection of different vehicle templates whose orientations are all at the same angle $\theta_\mathrm{i}$.

All the vehicle templates in $C_{\theta_i}$ have the same size $36\times36$. The maximum vehicle response is defined as the maximum similarity between $H_s^I(X)$ and all vehicle templates. In this thesis, two features including vehicle contour and wavelet coefficients are used to measure this similarity. In addition to these two features, we also use the corner feature to enhance the accuracy of vehicle detection. In what follows, details of each feature are introduced.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 41 | 32 | 25 | 18 | 13 | 10 | 8 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 4 | 5 | 8 | 9 | 10 | 13 | 18 | 25 |
| 32 | 25 | 18 | 13 | 8 | 5 | 4 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 4 | 4 | 5 | 8 | 13 | 20 |
| 25 | 18 | 13 | 8 | 5 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 5 | 10 | 17 |
| 18 | 13 | 8 | 5 | 2 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 4 | 4 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 4 | 9 | 16 |
| 13 | 8 | 5 | 2 | 1 | 0 | 0 | 1 | 0 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 8 | 9 | 8 | 5 | 4 | 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 9 | 16 |
| 10 | 5 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 4 | 9 | 9 | 9 | 9 | 9 | 10 | 13 | 16 | 13 | 10 | 9 | 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 9 | 16 |
| 9 | 4 | 1 | 0 | 0 | 1 | 2 | 1 | 0 | 1 | 4 | 9 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 9 | 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 9 | 16 |
| 9 | 4 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 4 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 9 | 16 |
| 8 | 4 | 1 | 0 | 1 | 4 | 1 | 0 | 0 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 8 | 13 |
| 5 | 2 | 1 | 0 | 1 | 4 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 | 5 | 10 |
| 4 | 1 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 1 | 2 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 1 | 2 | 1 | 0 | 1 | 2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 0 | 0 | 1 | 0 | 1 | 4 | 8 |
| 4 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 4 | 8 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 5 | 2 | 1 | 0 | 1 | 0 | 1 | 2 | 5 |
| 4 | 1 | 0 | 1 | 1 | 0 | 1 | 2 | 5 | 10 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 13 | 8 | 4 | 1 | 0 | 1 | 0 | 0 | 1 | 4 |
| 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 8 | 13 | 20 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 20 | 17 | 16 | 16 | 9 | 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 |
| 4 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 9 | 10 | 13 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 13 | 10 | 9 | 9 | 9 | 4 | 1 | 0 | 0 | 1 | 0 | 1 | 4 |
| 4 | 1 | 0 | 0 | 0 | 0 | 1 | 4 | 4 | 5 | 8 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 8 | 5 | 4 | 4 | 4 | 4 | 2 | 1 | 0 | 1 | 0 | 1 | 4 |
| 4 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 4 |
| 4 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 |
| 4 | 1 | 0 | 1 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 4 |
| 4 | 1 | 0 | 1 | 4 | 4 | 4 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 4 | 4 | 4 | 1 | 0 | 1 | 2 | 5 |
| 4 | 1 | 0 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | 1 | 1 | 1 | 1 | 0 | 1 | 4 | 8 |
| 4 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 5 | 8 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 8 | 5 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 5 | 10 | 16 | 16 | 16 | 16 | 16 | 10 | 5 | 2 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 4 | 9 | 16 | 25 | 25 | 25 | 16 | 9 | 4 | 1 | 0 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 0 | 1 | 4 | 2 | 1 | 1 | 0 | 1 | 4 | 9 | 16 | 25 | 36 | 25 | 16 | 9 | 4 | 1 | 0 | 1 | 2 | 2 | 1 | 1 | 0 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 4 | 9 | 16 | 25 | 25 | 25 | 16 | 9 | 4 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 9 | 16 | 16 | 16 | 16 | 16 | 9 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 4 | 9 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 5 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 5 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 1 | 4 | 9 |
| 5 | 2 | 1 | 0 | 1 | 2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | 1 | 1 | 0 | 0 | 1 | 4 | 9 |
| 8 | 4 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 5 | 10 |
| 10 | 5 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 5 | 8 | 13 |
| 13 | 8 | 5 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 5 | 8 | 13 | 18 |
| 18 | 13 | 8 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 8 | 13 | 18 | 25 |

(a) (b)

Fig. 9: Result of distance transform. (a) Original Image. (b) Distance transform of (a).

## 3.2 Vehicle Features

### *3.2.1 Contour Feature*

Contour is a good feature to describe vehicle's shapes and usually represented by chain coding. However, the technique of chain coding is easily affected by noise. Therefore, different from chain coding, we use a distance transform to convert an object contour to a distance map. Based on this map, different vehicle hypothesis can be well discriminated.

First, a 3×3 mask is used to detect all boundary points from a vehicle *V*. When this mask is used and moved at a non-zero pixel *p*, if one pixel in this mask is zero, then *p* is a boundary pixel. Assume that $B_V$ is a set of boundary pixels extracted from *V*. Then, the distance transform of a pixel *p* in *V* is defined as

$$DT_V(p) = \min_{q \in B_V} d(p,q), \tag{14}$$

where $d(p,q)$ is the Euclidian distance between *p* and *q*. In order to enhance the strength of distance changes, Eq.(14) is further modified as follows

$$\overline{DT}_V(p) = \min_{q \in B_V} d(p,q) \times \exp(\kappa d(p,q)), \tag{15}$$

where $\kappa = 0.1$. Fig. 9(b) shows the result of the distance transform of Fig. 9(a). Thus, according to Eq.(15), a set $F_C(V)$ of contour features can be extracted from *V*. If we scan all pixels of $V$ in a row major order, $F_C(V)$ can be then represented as a vector, i.e.,

$$F_C(V) = [\overline{DT}_V(p_0),....,\overline{DT}_V(p_i),....], \tag{16}$$

where all $p_i$ belong to $V$ and *i* is the scanning index.

### 3.2.2 Wavelet Coefficients

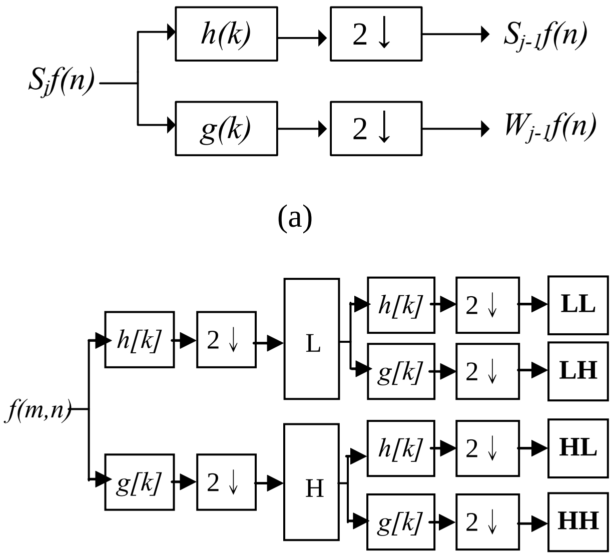
Wavelet transform is a very useful tool to represent images at different resolutions. It has been successfully applied in many applications like compression, watermarking, texture analysis, communications, and so on. The wavelet transform uses two kinds of filers to decompose a signal into different resolutions, i.e., the low-pass filter $h(k)$ and the high-pass one $g(k)$. Then, given a discrete signal *f(n)* (assumed at the fine resolution *j*=0 and represented as $S_0 f(n)$), with the low-pass filter $h(k)$, the approximation of *f(n)* at lower resolution *j*-1 can be calculated as follows:

$$S_{j-1}f(n)=\sum_{k=-\infty}^{\infty}S_j f(k)h(k-2n)\,. \tag{17}$$

In addition, information lost between $S_j f(n)$ and $S_{j-1}f(n)$ can be obtained using the high-pass filter $g(k)$ as follows

$$W_{j-1}f(n)=\sum_{k=-\infty}^{\infty}S_j f(k)g(k-2n)\,. \tag{18}$$

From the view of signal processing, $S_{j-1}f(n)$ and $W_{j-1}f(n)$ are, respectively, the components of low frequency and high frequency of $S_j f(n)$. The above procedure, which is also known as the sub-band coding, can be repeatedly performed. Fig. 10(a) shows the diagram of 1D wavelet transform. The 1D wavelet transform can be easily extended to two dimensions. The simplest way to generate 2D wavelet transform is to apply two 1D transforms to the rows and columns of a 2D signal *f(m, n)*, respectively. Fig. 10(b) shows the block diagram of 2D wavelet transform. Given *f(m, n)*, convolving its rows with $h(k)$ and $g(k)$, we get two sub-images whose horizontal resolutions are reduced by a factor 2. Both sub-images are then filtered columnwise and downsampled to yield four quarter-size output subimages. The filters $h(k)$ and $g(k)$ we use are the D4 family of Daubeches's basis [21], i.e., $\{h(0), h(1), h(2), h(3)\}=\{\frac{1+\sqrt{3}}{4\sqrt{2}},\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{3-\sqrt{3}}{4\sqrt{2}},\frac{1-\sqrt{3}}{4\sqrt{2}}\}$ and $\{g(0), g(1), g(2), g(3)\} = \{h(3), -h(2), h(1), -h(0)\}$.



(a)

(b)

Fig. 10: Block diagram of discrete wavelet transform. (a) 1D Wavelet transform. (b) 2D Wavelet transform.

A three-scale wavelet transform is used to process all vehicle images. Since all templates have the same size, the scale factor of the wavelet transform to all the processed vehicle patterns is the same. Then, each wavelet coefficient is quantized to three levels, i.e., 1, 0, -1, if its value is larger than 0, equal to 0, and less than 0, respectively. After that, all the quantized coefficients are recorded for further recognition. When recording, each wavelet coefficient is further classified into different bands, i.e., *LL*, *LH*, *HL*, and *HH*. According to this classification, a pixel $p$ is labeled as 1, 2, 2, and 4 if it locates in the *LL*, *LH*, *HL*, and *HH* bands, respectively. Let $l(p)$ denote the labeling value of $p$. Then, given a vehicle $V$, from its wavelet coefficients, we can extract a set $F_W(V)$ of WT features. If we scan $V$ in a row-major order, $F_W(V)$ is further represented as a vector, i.e.,

$$F_W(V) = [l(p_0)Coeff_V^W(p_0),....,l(p_i)Coeff_V^W(p_i),....], \tag{19}$$

where all $p_i$ belong to $V$ and $i$ is the scanning index. Usually, the *HH* band contains more edge information than other bands. Therefore, in Eq.(19), a larger label is used to weight the *HH* band. As to the *LH* and *HL* bands, since they contain more edges than the *LL* band, they have larger weights than the *LL* band (but less than the *HH* band).

### *3.2.3 Integration of Wavelet Feature and Edge Map*

In Sections *3.2.1* and *3.2.2*, two features have been illustrated to describe the visual characteristics of a vehicle template. We are now able to integrate these two features together for computing the similairty between a hypothesis $H_s^I(X)$ and a vehicle template *V*. Given *V*, based on Eqs. (16) and (19), we can extract its two feature vectors $F_C(V)$ and $F_W(V)$ from its contour and wavelet transform, respectively. For convience, we combine these two features together to form a new feature vector $F(V)$, i.e., $F(V) = [F_C(V), F_W(V)]$. For a vehicle class $C_{\theta_i}$, if there are $N_{\theta_i}$ templates in $C_{\theta_i}$, we can calculate its mean $\mu_{\theta_i}$ and variance $\Sigma_{\theta_i}$ of $F(V)$ from all samples *V* in $C_{\theta_i}$. Then, given a vehicle hypothesis *H*, the similary between *H* and $C_{\theta_i}$ can be measured by this equation:

$$S(H, C_{\theta_i}) = \exp(-(F_H - \mu_{\theta_i})\Sigma_{\theta_i}^{-1}(F_H - \mu_{\theta_i})^t), \tag{20}$$

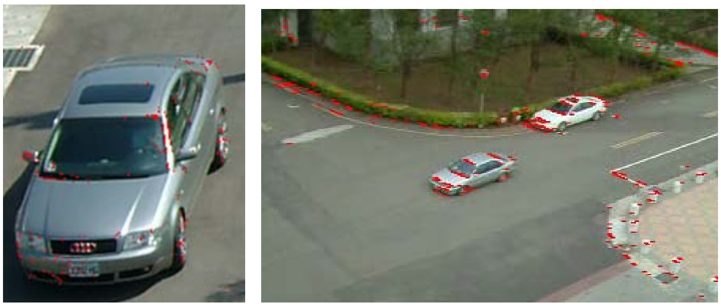
where $t$ means the transpose of a vector and $F_H$ is the feature vector of the vehicle hypothesis *H*. Therefore, given a position *X*, its vehicle reponse can be defined as follows

$$R(X) = \max_{s,\theta_i} S(H_s^I(X), C_{\theta_i}). \tag{21}$$

When calculaing Eq.(21), the parameter $\theta_i$ can be further eliminated if the direction of the hypothesis $H_s^I(X)$ is known in advance. In [51], a good moment-based method is provided for estimating the orientation of the longest axis of a region. If $\theta_s^I(X)$ is the orientaiton of $H_s^I(X)$, Eq.(21) can be further rewriten by

$$R(X) = \max_s S(H_s^I(X), C_{\theta_s^I(X)}) . \tag{22}$$

Based on Eq.(22), the vehicle response at the position $X$ can be easily estimated.



(a) (b)

Fig. 11: Results of corner deteciton. (a) and (b): vehicles containning many corners.

### 3.2.4 Corner Feature

Vehicles usually contain many corners even though they have different visual changes like orientations, sizes, colors, or types. Like Fig. 11, vehicles in (a) and (b) contain different numbers of corners. Therefore, corners can form a good feature for vehicle verificaiton. The Harris corner detector [20] is used to extract various corners for the task of vehicle verification. Assume that $I_x$ and $I_y$ are the first derivatives of an image $I$ in the $x$ and $y$ directions, respectively. Then, the detector operates on the matrix:

$$M(x,y) = \sum_{(x_i,y_i)\in Ne(x,y)} \begin{bmatrix} I_x^2(x_i,y_i) & I_x(x_i,y_i)I_y(x_i,y_i) \\ I_x(x_i,y_i)I_y(x_i,y_i) & I_y^2(x_i,y_i) \end{bmatrix}. \tag{23}$$

where $Ne(x, y)$ is a local neighborhood centered around ($x$, $y$). If the two eignevalues of $M$ are large, then a small motion in any direction will cause significant changes of intensity at the point ($x$, $y$). This indicates that the point is a corner. Accoding to this obsevation, the corner response function $CR$ is given by:

$$CR = \det M - \kappa\ (trace\ M)^2 , \tag{24}$$

where $\kappa$ is a parameter set to 0.04 (see Harris [20]). The local maxima of $CR$ (larger than a threshold) indicates the corner's positions.

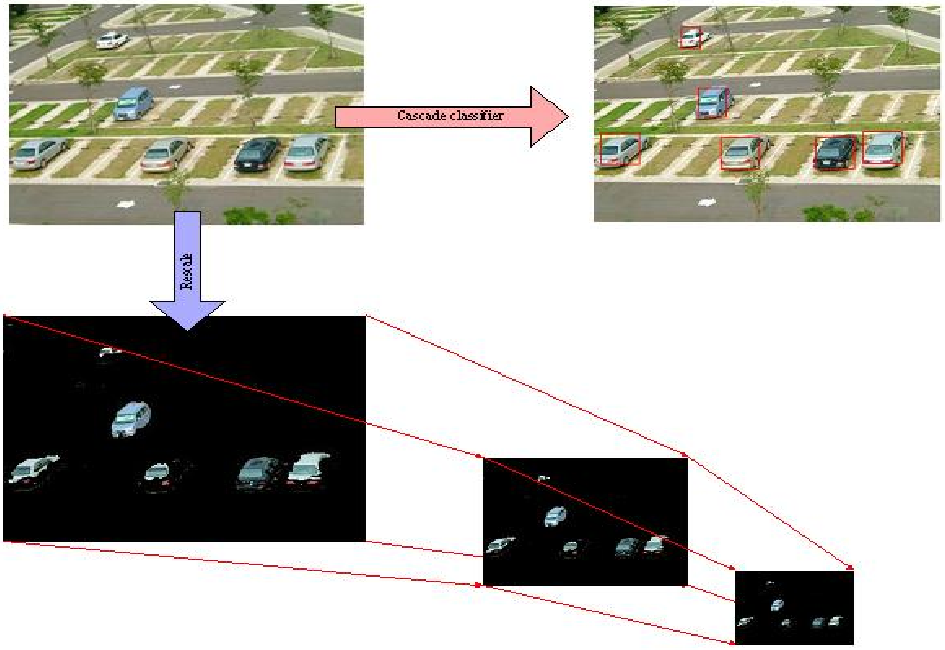


Fig. 12: Image pyramid structure for locating vehicles. At each step, the image is rescaled with 0.9 ratio until a pre-defined resolution is achieved.

### *3.2.5 Verification Procedure*

In real implementation, we borrow a well-known pyramid technique from face detection [22]-[23] to speed up the calculation of Eq.(22). Like Fig. 12, this technique constructs a pyramid structure of an image by gradually reducing its size. Assume that $w_I$ and $h_I$ are the width and height of a test image *I*. Each layer in this structure is a smoothed image $I_s$ having the size $0.9^s w_I \times 0.9^s h_I$ obtained by sub-sampling *I*. In this structure, a vehicle hypothesis $H_s^I(X)$ having the size $w_s \times h_s$ (or $0.9^{-s}36 \times 0.9^{-s}36$ ) in $I$ will become a $36 \times 36$ sub-pattern in $I_s$. Then, given a vehicle pixel *X* in *I*, if the maximum value of $S(H_s^I(X), C_{\theta_s^I(X)})$ is found in the layer $I_s$, its real vehicle size will be $w_s \times h_s$.

In order to quickly find desired vehicles from the above pyramid structure, we follow the idea of Viola and Jones [22] to construct a simple cascade (or hierarchical) classifier. The classifier uses a set of weak classifiers to gradually filter out impossible candidates. In this structure, each weak classifier uses a lower threshold to detect vehicles such that a higher detection rate can be maintained but also with a high false alarm rate. However, the false alarm rate will gradually decrease if more features are accordingly used. With the structure, all desired vehicles can be located very efficiently and accurately. As shown in Fig. 13, the color feature is first used for eliminating almost impossible vehicle candidates. Then, the corner feature is used to filter out additional negatives. Finally, the subsequent classifier finds desired vehicles using the features of edge maps and wavelet coefficients.

In this cascade structure, two thresholds are used to remove spurious negatives and to declare whether a vehicle is detected at the position *X*. Let $\lambda_C$ be the average number of corners appearing in all the training vehicle samples. If *X* contains a real vehicle, the number of corners around *X* should be larger than $0.5\lambda_{\mathrm{C}}$. In addition to $\lambda_C$, we use another threshold $\lambda_R$ to eliminate impossible vehicle candidates according to their vehicle responses. Let $\lambda_R$ be the average value of $R(X)$ for all the centers *X* of the training vehicle samples. For a vehicle pixel *X*, if its response $R(X)$ is larger than $0.8\lambda_R$, it is a vehicle candidate. The parameters $\lambda_R$ and $\Sigma_{\theta_i}^{-1}$ (the weight used in Eq.(20)) can also be learned using the AdaBoost algorithm [26] for increasing the accuracy of vehicle verification. However, experimental results prove that the above verification scheme performs well enough in detecting all desired vehicles. Finally, there would be many vehicle candidates which are overlapped together due to noise or shadows. If candidates are inside other stronger candidates, they will be eliminated.

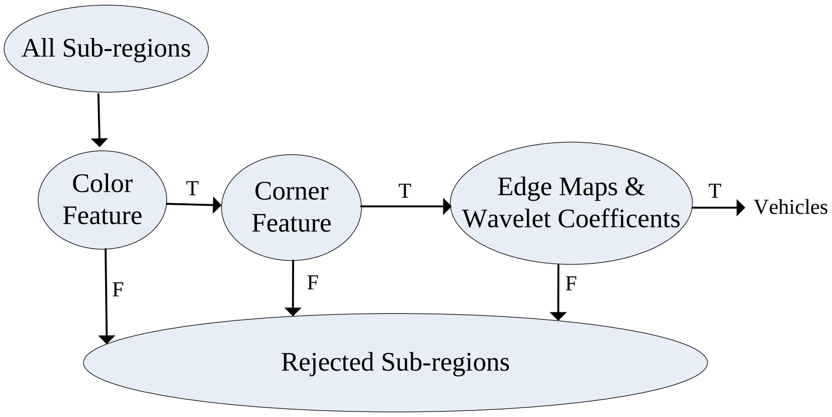


Fig. 13: Cascade structure used for vehicle detection.

# 3.3 Road Sign Features

## *3.3.1 Geometrical Properties*

Given an image, after the color classification and a connected component analysis [51], different road sign candidates can be then extracted. This section will use their geometrical properties to filter out impossible candidates. Since a road sign has different shapes according to its different functionalities, we divide the road signs into three categories, i.e., circle, rectangle, and triangle, respectively. Then, a coarse-to-fine scheme is proposed to gradually remove impossible candidates. At the coarse stage, three criteria are first used to roughly filter out impossible candidates. The first criterion requires the dimension of road sign being large enough, that is:

$$w_R > 10 \text{ and } h_R > 10,$$

where $w_R$ and $h_R$ are the width and length of a road sign $R$, respectively. The second criterion requires the ratio between $w_R$ and $h_R$ satisfying

$$\min(\frac{h_R}{w_R+1}, \frac{w_R}{h_R+1}) > 0.6,$$

since the ratio is close to 1. Let $E_R$ and $Area_R$ denote the number of edge pixels and the area of $R$, respectively. The third criterion requires the road sign $R$ having enough edge pixels; that is, if

$$E_R / Area_R < 0.02,$$

$R$ is filtered out.

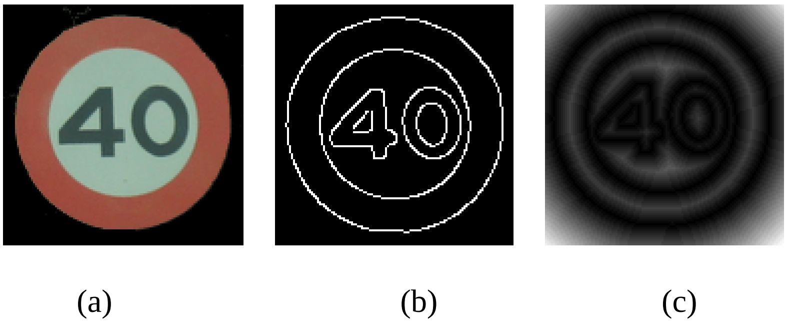


Fig. 14: Result of distance transform. (a) Original Image. (b) Edge map. (c) Distance transform of (b).

### *3.3.2 Modified Distance Transform with Weighting*

At the fine stage, each candidate is verified using its shape. Assume that $B_R$ is a set of boundary pixels extracted from $R$. Then, the distance transform of a pixel $p$ in $R$ is defined as

$$DT_R(p) = \min_{q \in B_R} d(p,q), \tag{25}$$

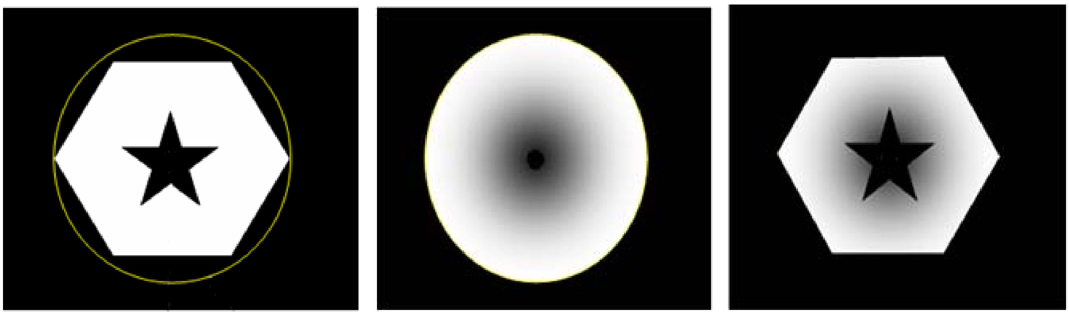
where $d(p,q)$ is the Euclidian distance between $p$ and $q$. In order to enhance the strength of distance changes, Eq.(25) is further modified as follows

$$\overline{DT}_R(p) = \min_{q \in B_R} d(p,q) \times \exp(\kappa d(p,q)), \tag{26}$$

where $\kappa = 0.1$. Fig. 14 shows the result of the distance transform. (a) is an image *R* of road sign and (b) is its edge map. Fig. 14(c) shows the result of its distance transform. Thus, according to Eq.(26), a set $F_R$ of contour features can be extracted from *R*. If we scan all pixels of *R* in a row major order, $F_R$ can be then represented as a vector, i.e.,

$$F_R = [\overline{DT}_R(p_0), \ldots., \overline{DT}_R(p_i), \ldots.], \tag{27}$$

where all $p_i$ belong to *R* and *i* is the scanning index. In addition to the outer contour, a road sign usually contains many text patterns. To verify a road sign candidate more accurately, its outer shape plays a more important role than its inner text patterns in road sign classification. Thus, a new weight $w_i$ which increases according to the distance between the pixel $p_i$ and the original *O* is included. Assume that *O* is the central of *R* and $r_i$ is the distance between $p_i$ and *O*, and the circumcircle of *R* has the radius *z*. Then, the weight $w_i$ is defined by:



(a) (b) (c)

Fig. 15: Weighting result of an image. (a) Original Image. (b) Weighting function. (c) Result of (a) after weighting.

$$w_i = \begin{cases} \exp(-|r_i - z|^2), \text{ if } r_i \le z; \\ 0, \qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{28}$$

With the help of $w_i$, Eq.(27) can be rewritten as follows:

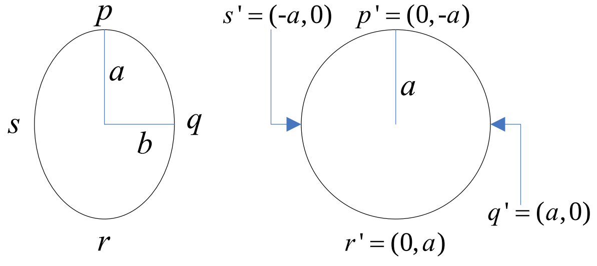
$$\bar{F}_R = [w_0 \overline{DT}_R(p_0),...., w_i \overline{DT}_R(p_i),....]. \tag{29}$$

Fig. 15 shows the result of distance transform with a weighting function. (a) is the original image *R* and the yellow circle shows the circumcircle of *R*. (b) is the weighting function defined in Eq.(28) and (c) is the result after weighting. There are only three types of road signs, i.e., circle, triangle, and rectangle needed for further verification. For each type $R_i$, a set of training samples is collected for capturing its shape characteristics. If there are $N_i$ templates in $R_i$, we can calculate its mean $\mu_i$ and variance $\Sigma_i$ of $\bar{F}_R$ from all the samples in $R_i$. Then, given a road sign candidate *H*, the similarity between *H* and $R_i$ can be measured by this equation:

$$S(H,R_i) = \exp(-(\bar{F}_H - u_i)\Sigma_i^{-1}(\bar{F}_H - u_i)^t), \tag{30}$$

where *t* means the transpose of a vector and $\bar{F}_H$ is the feature vector of the candidate *H*. Based on Eq.(30), we can well categorize *H* into different types with the equation:

$$type(H) = \arg\min_{R_i} S(H,R_i). \tag{31}$$



(a) (b)

Fig. 16: Rectification of a circle road sign. (a) Input road sign. (b) Rectification result of (a).

# 3.4 Road Sign Rectification

After verification, in order to handle skewed road signs, a rectification procedure should be further applied to it for recognizing all its embedded texts more accurately.

## 3.4.1 Circular Road Sign

Assume that *R* is the detected road sign. First of all, the Canny edge operator [49] is utilized for getting all its edge pixels. Then, the chain coding technique with 8 neighbors is adopted for extracting its outer boundary. According to the boundary feature, the shape type of *R* can be recognized using Eq.(31).

If *R* is recognized as a circular type, four control points are selected for rectification. Like Fig. 16(a), *p*, *q*, *r*, and *s* are the most top, right, bottom, and left points of *R*, respectively. Considering them as control points, we can get the longest axis of *R*. Assume that its length is 2*a*. Then, a projective transformation *M* can be found for rectifying *R* into a normal shape *R'* (see Fig. 16(b)). The relationship between *R* and *R'* can be defined as follows

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1} \text{ and } y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1}, \quad (32)$$

where (*x*, *y*) is the coordinate of a pixel in *R*, $(x', y')$ the coordinate of its corresponding point in *R'*, and $(m_0, m_1, ..., m_7)$ the parameters of the projective model *M*. Let *p'*, *q'*, *r'*, and *s'* be the four corresponding points of *p*, *q*, *r*, and *s* in *R'*, respectively, and have the following coordinates:

$$p' = (0, \text{-}a),\ q' = (a, 0),\ r' = (0, a), \text{ and } s' = (\text{-}a, 0).$$

Given the above four pairs of correspondence, *M* can be solved by a linear method [50].

Once *M* is obtained, all the points in *R* can be transformed into *R'* using Eq.(32).

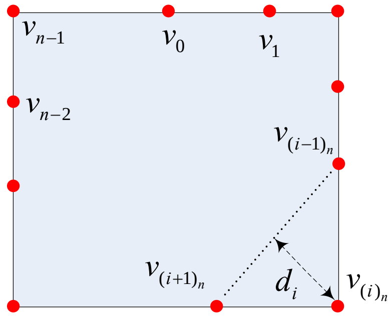


Fig. 17: Technique for iteratively pruning a point $v_\mathrm{i}$ whose $d_i$ is the minimum.

**Iterative Minimum Distance Pruning Algorithm**:

Step 1: Find a vertex $v_i$ from $C_R$ such that $d_i = \min_{v_k \in C_R} d_k$ ;

Step 2: Eliminate $v_i$ from $C_R$;

Step 3: If $C_R$ includes more than *m* points, go to Step 1.

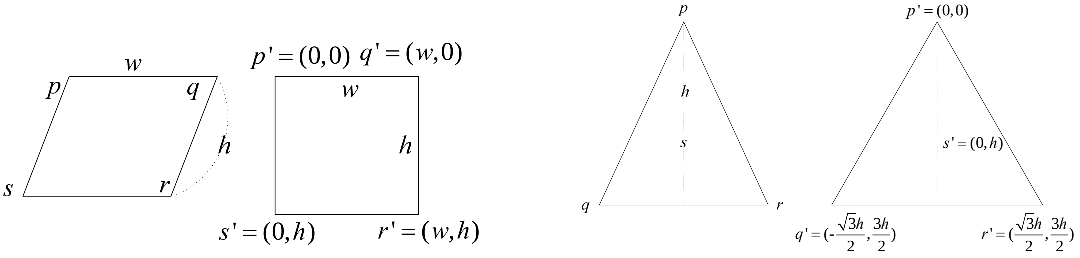
### *3.4.2 Rectangular and Triangular Road Signs*

If *R* is recognized as a rectangular or triangular road sign, another method will be used for the rectification task. First of all, we use a corner detection method [20] to detect all high curvature points along the boundary of *R*. Let $C_R$ be the set of high curvature points in *R*. If there are *n* points in $C_R$, we will use a curve pruning technique to reduce the number of points in $C_R$ to *m* points. If *R* is a rectangle, *m* will be four. If *R* is a triangle, *m* will be three. The pruning technique is described as follows. Assume that $C_R = \{v_0, v_1, ..., v_{n\text{-}1}\}$ and $L_{v_i}$ represents the straight line formed by the neighbor vertices $v_{(\mathrm{i}\text{-}1)_\mathrm{n}}$ and $v_{(\mathrm{i}+1)_\mathrm{n}}$ of $v_\mathrm{i}$. Here, $(i)_n$ means *i* mod *n*. If $L_{v_i}$ has the form $y = m_i x + c_i$, the distance $d_i$ between $v_\mathrm{i}$

and $L_{v_i}$ can be then calculated by:

$$d_i = \frac{| y_{v_i} - m_i x_{v_i} - c_i |}{\sqrt{1 + m_i}},$$

where the coordinates of $v_{\mathrm{i}}$ are $(x_{v_i}, y_{v_i})$. Fig. 17 shows details of the simplification technique. This technique is to iteratively prune a point $v_{\mathrm{i}}$ whose distance $d_i$ is the minimum and summarized as follows.


w
p
q
h
s
r
$p' = (0,0)$
$q' = (w,0)$
w
h
$s' = (0,h)$
$r' = (w,h)$
p
h
s
q
r
$p' = (0,0)$
$s' = (0,h)$
$q' = (-\frac{\sqrt{3}h}{2}, \frac{3h}{2})$
$r' = (\frac{\sqrt{3}h}{2}, \frac{3h}{2})$

(a) (b)

Fig. 18: Rectification of road signs. (a) Rectangle road sign. (b) Triangle road sign.

After pruning, *R* will have its corresponding number of control points. Like Fig. 18(a), if a rectangle is detected, four control points are selected and denoted as *p*, *q*, *r*, and *s*, respectively. Let $w_R$ be the distance between *p* and *q*, $h_R$ the distance between *q* and *r*. Then, the rectified rectangle *R'* has four points *p'*, *q'*, *r'*, and *s'* with the following coordinates:

$$p' = (0,0),\ q' = (w,0),\ r' = (w,h), \text{ and } s' = (0,h).$$

Based on the four matching pairs: (*p*, *p'*), …, and (*s*, *s'*), like Eq.(32), we can build an affine model *M* to transform *R* into *R*'. Similarly, *M* can be solved by a linear method [50].

If *R* is a triangle, only three control points are selected. Like Fig. 18(b), the three control points are denoted as *p*, *q*, and *r*, respectively. In addition to them, another new control point *s* is generated and selected as the gravity point of *R*. Let $h$ be the distance between *p* and *s*. Then, the rectified rectangle *R'* has four points *p'*, *q'*, *r'*, and *s'* with the following coordinates:

$$p' = (0,0)\,,\;\; q' = (-\frac{\sqrt{3}h}{2}, \frac{3h}{2})\,,\;\; r' = (\frac{\sqrt{3}h}{2}, \frac{3h}{2})\,, \text{ and } \; s' = (0,h)\,.$$

Based on the four matching pairs: (*p*, *p'*), …, and (*s*, *s'*), like Eq.(32), we can find another affine model *M* for transforming *R* into *R*'. Once *M* is found, even though a skewed road sign is handled, it still can be rectified into a regular shape.

## 3.5 Binarization

Once a road sign *R* is extracted, to recognize the texts in *R*, we use a moment-based thresholding approach [51] to binarize *R*. Let $\sigma_g$ denote the global variance of *R*. In addition, $\sigma_f$ and $\sigma_b$ are the variances of foreground and background objects parts. The optimal threshold *t* for binarizing an image can be found by minimizing the within-group variance as follows:

$$t* = \arg\min_t (q_f^t \sigma_f^2(t) + q_b^t \sigma_b^2(t))\,, \tag{33}$$

where $q_b^t = \sum_{i=0}^{t} P(i)$, $q_f^t = \sum_{i=t+1}^{G} P(i) = 1 - q_b^t$, *G* the maximum gray value of *R*, and $P(i)$ the occurrence probability of intensity $i$ in *R*. The global variance $\sigma_g^2$ is defined as:

$$\sigma_g^2(t) = \sum_{i=0}^{G} |\, i - \mu_g \,|^2 \; P(i)\,, \tag{34}$$

where $\mu_g = \sum_{i=0}^{G} iP(i)$ . Let $\sigma_w^2(t) = q_f\sigma_f^2(t) + q_b\sigma_b^2(t)$ . From Eq.(34), after some calculations, we have

$$\sigma_g^2(t) = \sigma_w^2(t) + q_f(t)q_b(t)[u_f^t - u_b^t]^2,$$

where $u_f^t$ and $u_b^t$ are the mean values of foreground and background objects in *R* when the threshold *t* is used. Let $\sigma^2(t) = q_b^t q_f^t [\mu_b^t - \mu_f^t]^2$. Then, we have

$$\sigma_g^2(t) = \sigma_w^2(t) + \sigma^2(t). \tag{35}$$

Since $\sigma_g^2$ is constant, the problem to find a threshold *t* that minimizes $\sigma_w$ becomes finding a threshold *t* for maximizing $\sigma^2$. To more efficiently calculate $\sigma^2(t)$, its three terms can be updated using the following three recursive forms:

$$q_b^{t+1} = q_b^t + P(t+1),\ \ u_b^{t+1} = \frac{q_b^t u_b^t + (t+1)P(t+1)}{q_b^{t+1}},\ \text{and}\ \ u_f^{t+1} = \frac{u_g - q_b^{t+1} u_b^{t+1}}{1 - q_b^{t+1}}.$$

Since the number of possible values of *t* is small, the optimal *t* can be easily found by trying all the possible values of *t* which maximizes $\sigma^2$. Once *t* is obtained, the texts in *R* can be easily extracted.

# Chapter 4. Experimental Results

## 4.1 Vehicle Detection Performance

In order to analyze the performance of our proposed method, various static images captured under different weather conditions and lighting conditions were used. For the training vehicles, they were collected from different sources including roads, parking lots, cluttered backgrounds, and so on. The dimension of training vehicles is clipped to 36×36. To tackle the variations of vehicle orientation, eight classes of vehicles with different orientations were collected. We quantize the vehicle classes according to the angle between the line from camera center and the vehicle center. (i.e., camera tilt angles $0^\circ$, $15^\circ$ and vehicle orientations $0^\circ$, $30^\circ$ $90^\circ$ and $120^\circ$ respectively. ) In addition, for measuring the accuracy of our proposed method to detect vehicles directly from still images, a database including 354 images acquired under different lighting and weather conditions was used.

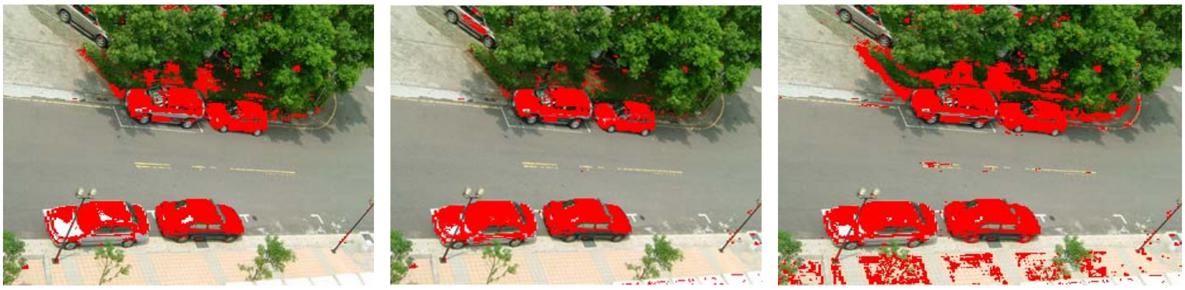
### 4.1.1 Results of Vehicle Pixels Classification

To evaluate and measure the performances of our proposed method to detect vehicle colors, the precision and false-alarm rates are defined. Precision is the ratio of the number of correctly detected vehicle pixels to the number of exactly existing vehicle pixels. False alarm rate is the ratio of the number of background pixels but misclassified as vehicles to the number of all background pixels. These two measures are defined as:

$$Precision = C_{vehicle} \ / N_{vehicle} \quad \text{and} \quad Rate\ of\ False\text{-}Alarm = F_{vehicle} / N_{back\text{-}pixels},$$

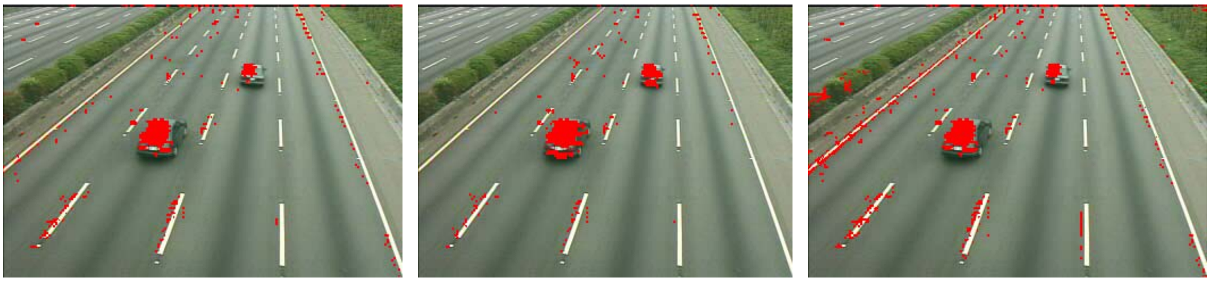
where $N_{vehicle}$ is the total number of vehicle pixels, $C_{vehicle}$ the number of correctly detected vehicle pixels, $N_{back\text{-}pixels}$ the number of all background pixels, and $F_{vehicle}$ the number of background pixels but misclassified as vehicle ones. When calculating these two measures, the ground truth of vehicle pixels was manually obtained. In what follows, several experiments under different conditions were demonstrated for analyzing the robustness and effectiveness of our proposed method.

The first experiment was conducted to compare the results of vehicle color classification when different color transforms are used. Fig. 19 shows the comparison results of vehicle color classification when Eq.(4), Eq.(5) and Eq.(6) are used, respectively. (a), (b), and (c) are the results using Eq.(4), Eq.(5), and Eq.(6), respectively. The precision rates of Eq.(4), Eq.(5), and Eq.(6) are 85.3%, 87.2%, and 86.3%, respectively. In addition, their corresponding false-alarm rates of vehicle pixel detection are 4.1%, 2.4%, and 8.9%, respectively. The lower false-alarm rate implied that most of background pixels were filtered out and did not need to be further verified. Thus, many redundant searches can be avoided in advance and the verification process can be significantly speeded up to find desired vehicles. Although Eq.(6) has a better precision rate than Eq.(4), its false-alarm rate is too higher than Eq.(4). Eq.(5) performs the best among the three methods. Fig. 20 shows another comparison among them, where the results of Eq.(4), Eq.(5), and Eq.(6) are shown in (a), (b), and (c), respectively. Their precision rates are 75.3%, 81.6%, and 78.9%, respectively. In addition, their corresponding false-alarm rates of vehicle pixel detection are 3.6%, 2.1%, and 4.5%, respectively. Eq.(5) still performs the best among these methods in the classification of vehicle pixels.



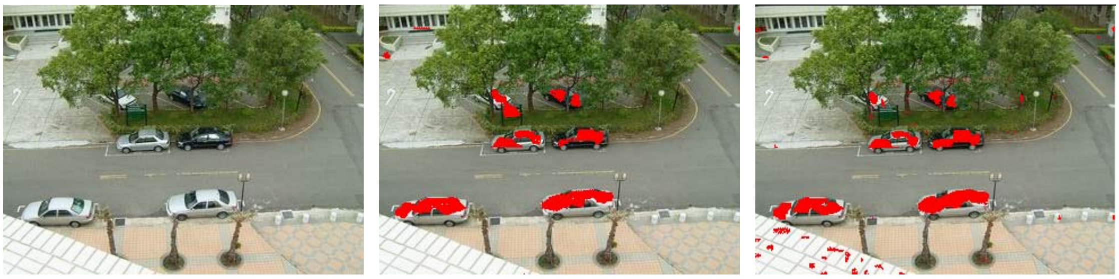
(a) (b) (c)

Fig. 19: Results of vehicle color classification when different color transforms are used. (a) Result of color classification using Eq.(4). (b) Result of color classification using Eq.(5). (c) Result of color classification using Eq. (6).



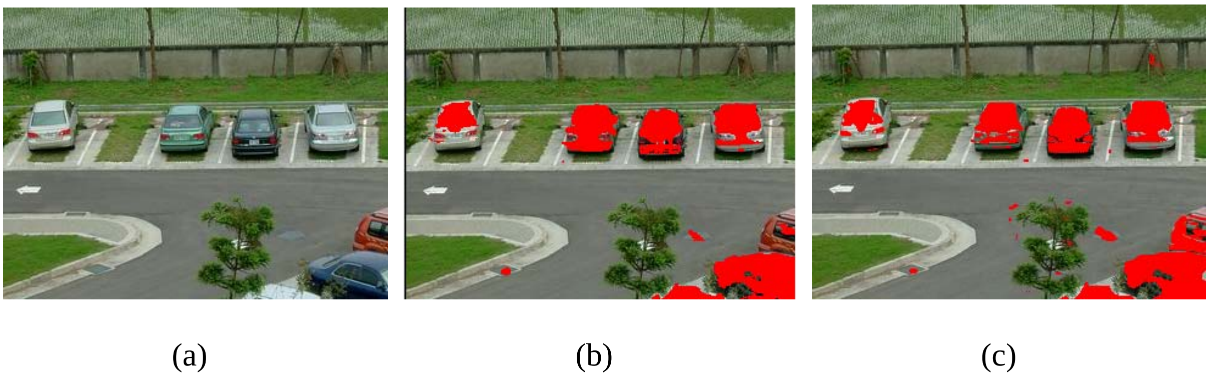
(a) (b) (c)

Fig. 20: Results of vehicle color classification when different color transforms are used. (a) Result of vehicle color detection using Eq.(4). (b) Result of vehicle color detection using Eq.(5). (c) Result of vehicle color detection using Eq. (6).



(a) (b) (c)

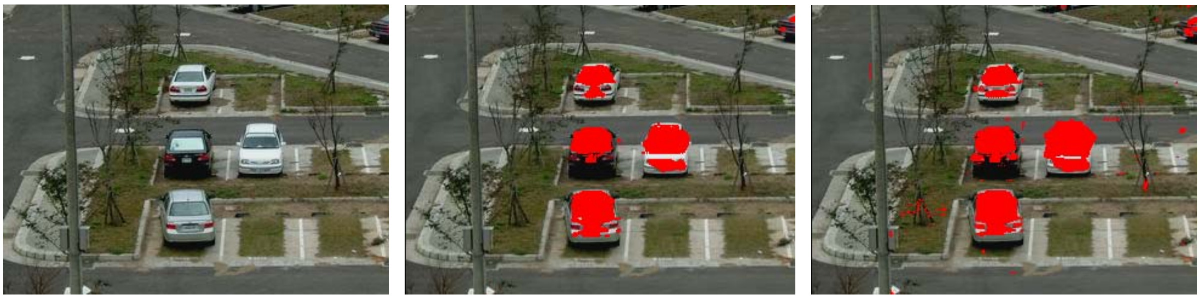
Fig. 21: Results of vehicle color detection. (a) Original image. (b) Result of vehicle color detection using the Bayesian classifier. (c) Result of vehicle color detection using the RBF network.



(a) (b) (c)

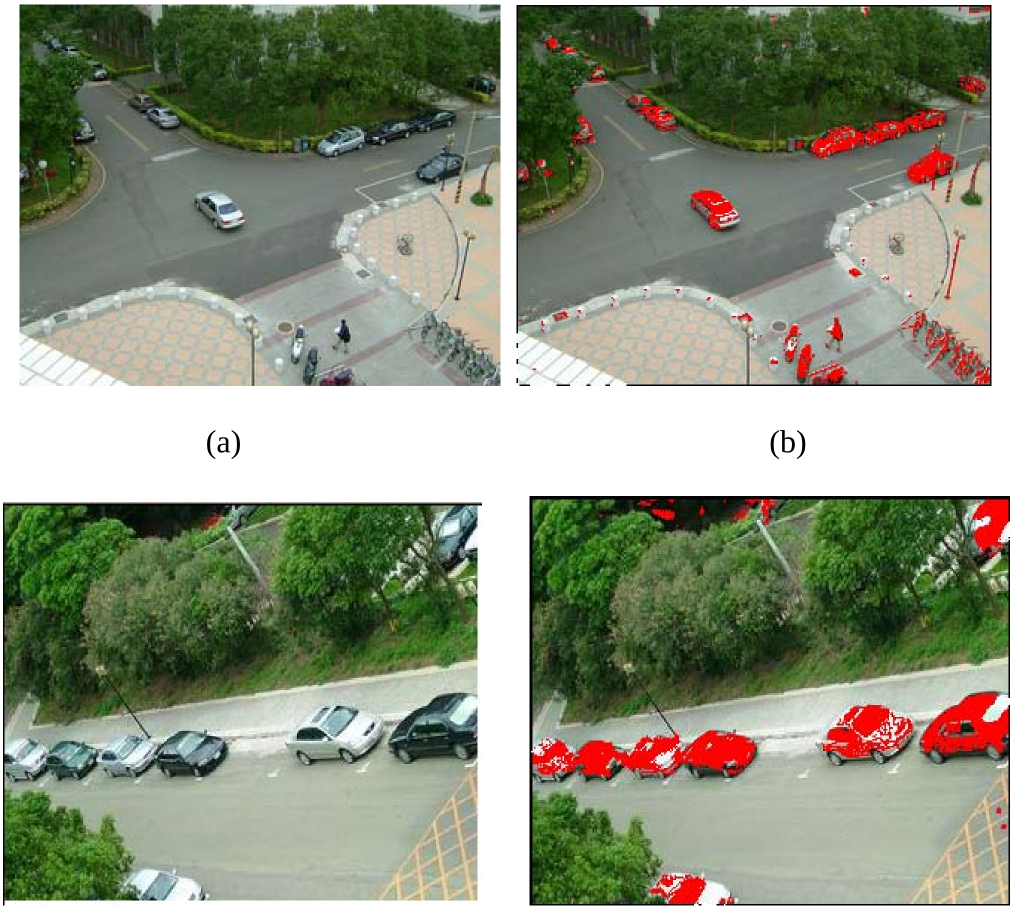
Fig. 22: Results of vehicle color detection. (a) Original image. (b) Result of vehicle color detection using the Bayesian classifier. (c) Result of vehicle color detection using the RBF network.

The second experiment was conducted to compare the performances of vehicle color detection using the Bayesian classifier and the RBF network, respectively. Fig. 21 shows the results of vehicle color detection using Eqs.(13) and the RBF network, respectively. (b) is the detection result of (a) using the Bayesian classifier and (c) the one using the RBF network. In (b), the precision rate and false-alarm rate of vehicle pixel detection are 87.7% and 2.8%, respectively. In (c), its corresponding precision rate and false-alarm rate are 86.1% and 5.9%, respectively. Clearly, the false-alarm rate of the RBF network is higher than the Bayesian classifier. Therefore, the Bayesian classifier performs better than the RBF network. Fig. 22 shows another set of performance comparisons between the two classifiers. (b) and (c) are the results of vehicle color detection using the Bayesian classifier and the RBF network, respectively. Their precision rates are 84.1% and 84.4%, respectively. In addition, their false-alarm rates are 3.1% and 5.2%, respectively. The Bayesian classifier still had a lower false-alarm rate and performed better than the RBF network.



(a) (b) (c)

Fig. 23: Results of vehicle color detection under a cloudy day. (a) Original image. (b) Result of vehicle color detection using the Bayesian classifier. (c) Result of vehicle color detection using the RBF network.

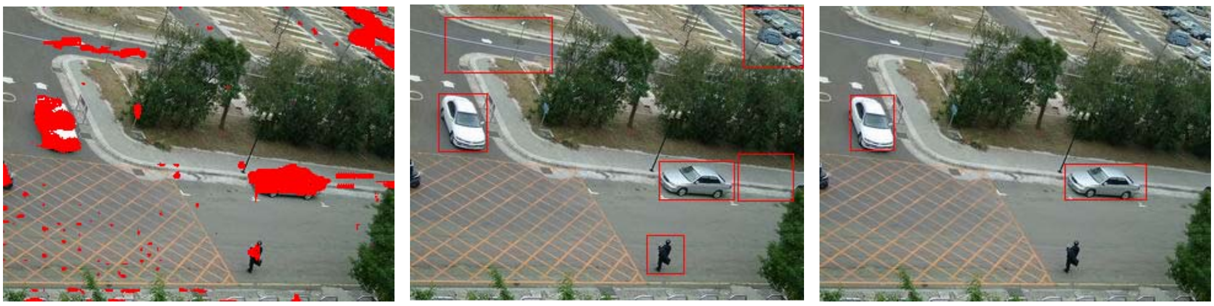


(a) (b)

(c) (d)

Fig. 24: More examples of vehicle color detection. (a),(c) Original image. (b),(d) Detection result of vehicle color.

In addition to sunny images, a cloudy image was also used for making a fair comparison between their performances under different weather conditions. Fig. 23 shows the detection results of vehicle color obtained from a cloudy image. The precision rates of (b) and (c) are 83.1% and 83.5%, respectively. As to the false-alarm rates of (b) and (c), they are 1.6% and 5.7%, respectively. The Bayesian classifier still performed better than the RBF network since the former had a lower false-alarm rate than the latter one. Although other complicated neural networks like SVM (Support Vector Machines) can be used for reducing the false-alarm rate, the Bayesian classifier performs well enough to detect vehicle pixels.



(a) (b) (c)

Fig. 25: Results of vehicle detection when the verification process is used. (a) Result of color classification. (b) Result of vehicle detection without verification. (c) Result of vehicle detection after verification.

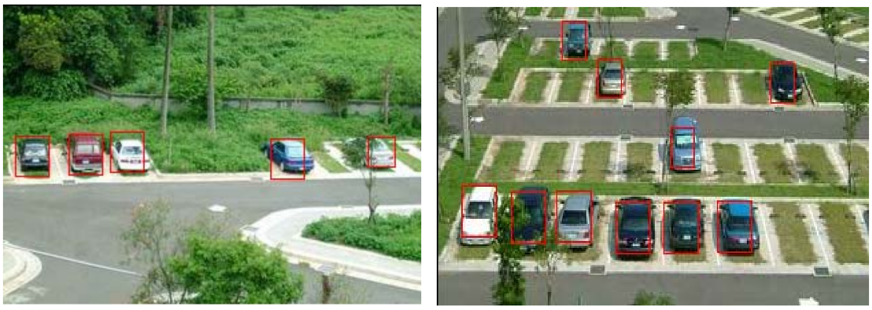
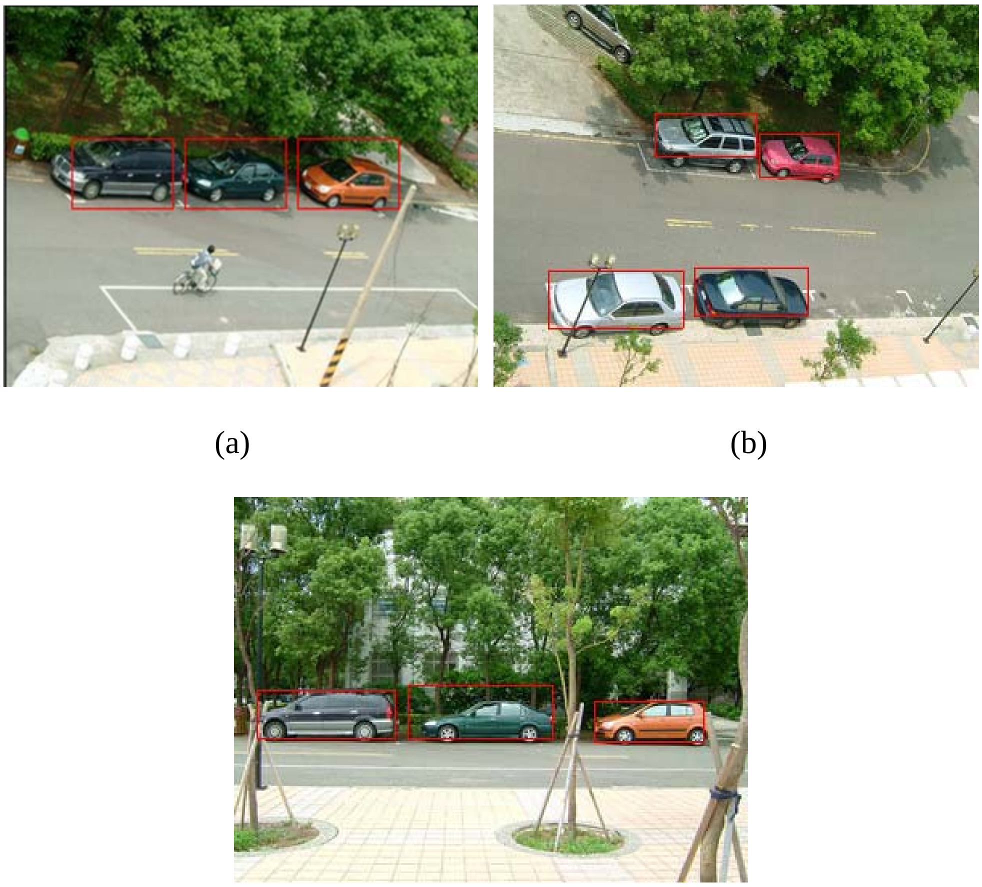


Fig. 26: Result of vehicle detection in a parking lot. Although these vehicles had different colors, all of them were correctly detected.

For more accurately analyzing the performances of our proposed vehicle color detector, two more experiments were performed. Fig. 24 demonstrates more results of vehicle color detection when the Bayesian classifier was used. In Fig. 24(b), the precision rate and false-alarm rate were 86.1% and 6.3%, respectively. Although the false-alarm rate is high, none of vehicle candidates was missed. Fig. 24(d) shows another result of vehicle color detection. The precision rate and false-alarm rate of vehicle pixel detection are 89.9% and 2.1%, respectively. All of possible vehicle candidates were correctly extracted even though vehicles were parked behind the trees.

### *4.1.2 Vehicle Detection Results*

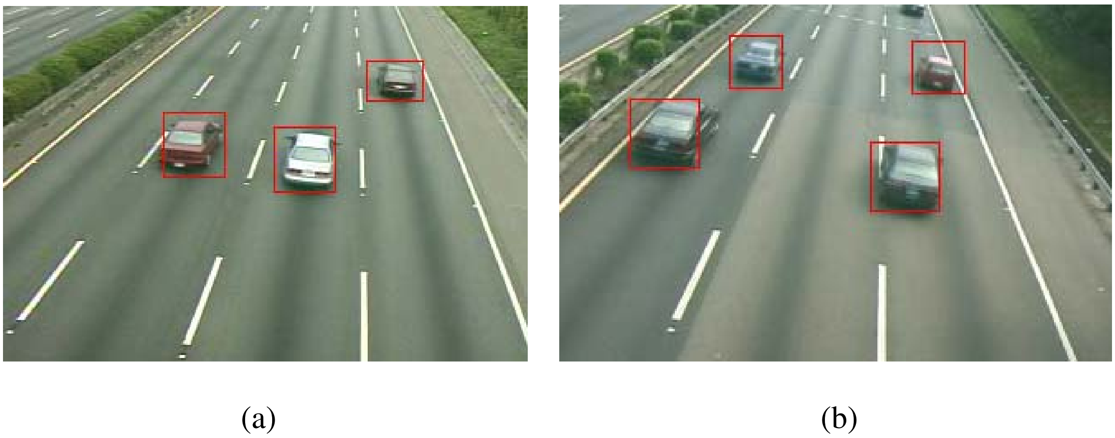
Another set of experiments was performed to examine the abilities of our vehicle detection method to detect vehicles directly from static images. Actually, after color classifying, all the detected vehicle pixels will form different regions. In Fig. 25, (a) is the result of color classification. If a region has enough pixels, we can consider it as a vehicle candidate. Then, we had the detection result shown in (b). However, there were many false alarms in (b). After using our proposed verification method, a more accurate result was obtained in (c). Clearly, the proposed verification scheme makes quite improvements in the accuracy of vehicle detection. Fig. 26 shows another result of vehicle detection obtained from a parking lot. Although these vehicles had different colors, all of them were still correctly detected and located. Although some vehicles occluded by a tree, they still were correctly detected. Fig. 27 shows three cases of vehicle detection when vehicles in roads have other orientations. Even though vehicles have various orientations, they still were well detected.



(a) (b)

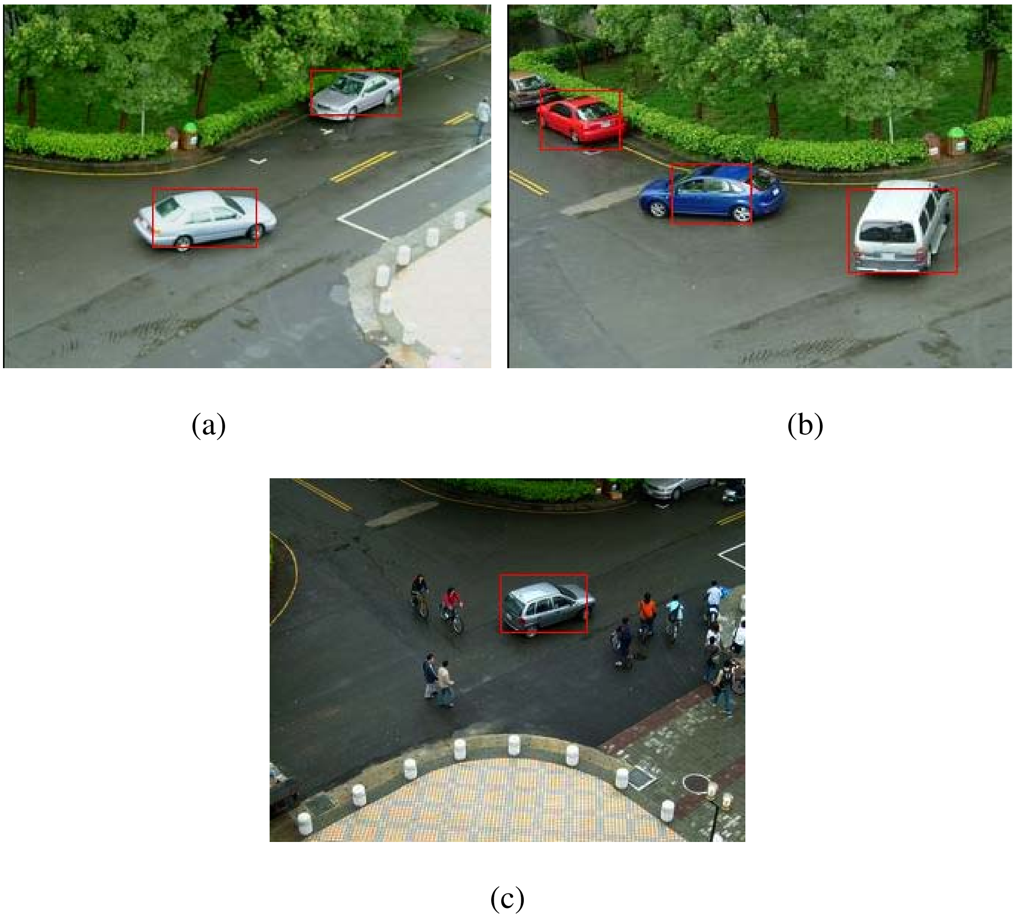
(c)

Fig. 27: Results of vehicle detection from roads. Although these vehicles had different orientations and colors, all of them were correctly detected.



(a) (b)

Fig. 28: Results of detecting vehicles from highways. Although these vehicles were with different colors, all of them were correctly detected.



(a)

(b)

(c)

Fig. 29: Results of vehicle detection under a rainy day.

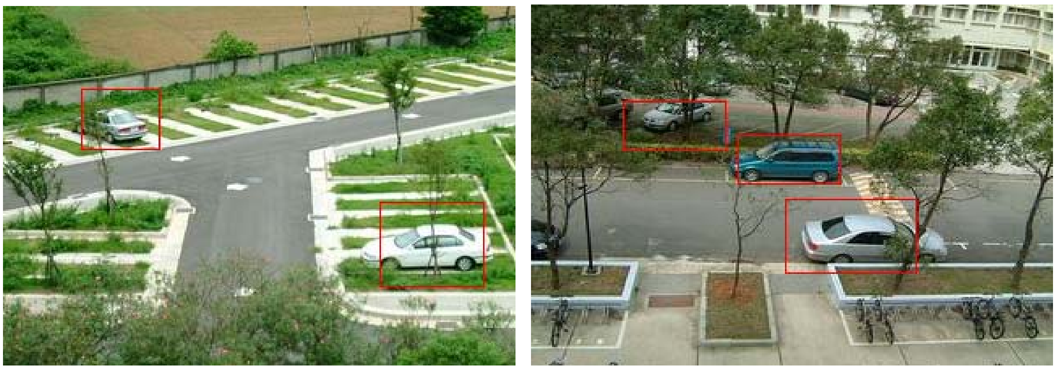


Fig. 30: Results of vehicle detection when occlusions happened.

In addition to parking lots and roads, the proposed method also works well to detect vehicles from highways. Fig. 28 shows two results of vehicle detection when highway images were used. Fig. 29 shows another case of vehicle detection when images were captured under a rainy day. According to the results in Fig. 28 and Fig. 29, even though vehicles were captured under different lighting or weather conditions, the proposed method still performed very well to detect all kinds of vehicles. Table 2 lists the quantitative performance analysis of our proposed method to detect vehicles under different weather conditions and scenes. Since the sunny day has better lighting conditions, our proposed method works better in such a day than other weather conditions. The average precision rate of vehicle detection using the proposed algorithm is 94.9%. In addition, the average false-alarm rate and missing rate are 8.23% and 5%, respectively. The above three measures are defined as:

$$Precision = C_{no.vehicle} / N_{no.vehicle} \quad ,$$

$$False\text{-}Alarm = F_{no.vehicle} / (C_{no.vehicle} + F_{no.vehicle}) \quad ,$$

$$Missing = 1 - Precision.$$

where $N_{no.vehicle}$ is the total number of vehicles, $C_{no.vehicle}$ the number of correctly detected vehicles, and $F_{no.vehicle}$ the number of background objects but misclassified as vehicles.

Table 2: Evaluation on vehicle detection when different weather conditions and scenes were handled.

| Weather and Scene | | No. of Vehicles | Correct detection | False detection | No. of miss | Precision Rate | False Alarm Rate: | Missing Rate |
|---|---|---|---|---|---|---|---|---|
| Sunny | High way 1 | 175 | 170 | 6 | 5 | 97.14% | 3.42% | 2.85% |
| | High way 2 | 153 | 147 | 7 | 6 | 96.07% | 4.57% | 3.92% |
| | Road 1 | 126 | 121 | 10 | 5 | 96.03% | 7.93% | 3.96% |
| | Road 2 | 77 | 75 | 9 | 2 | 97.4% | 11.68% | 2.59% |
| Cloudy | Road 3 | 146 | 136 | 22 | 10 | 93.15% | 15.06% | 6.84% |
| | Parking lot 1 | 92 | 88 | 8 | 4 | 95.65% | 8.69% | 4.34% |
| | Parking lot 2 | 106 | 98 | 13 | 8 | 92.45% | 12.26% | 7.54% |
| | High Way 3 | 117 | 107 | 16 | 10 | 91.45% | 13.67% | 9.4% |
| Rainy | Parking lot 3 | 109 | 104 | 7 | 5 | 95.41% | 6.42% | 4.58% |
| | Road 4 | 96 | 90 | 4 | 6 | 93.75% | 4.16% | 6.25% |
| Average | | 1197 | 1136 | 102 | 61 | 94.90% | 8.23% | 5% |

For making fair comparisons, three methods proposed by Agarwal and Roth[27], Zhu *et al*. [28], and Schneiderman and Kanade [29] were implemented, respectively. For other methods using motion information [4], [30] or range data [13], [14], [31] to detect vehicles, we did not compare them in this thesis. Table 3 lists the accuracy comparisons of vehicle detection among these three methods [27], [28], and [29]. In [27], Agarwal and Roth proposed a part-based representation scheme to represent vehicles using a vocabulary of 400 vehicle parts. Since the combination of vehicle parts is huge, their detector will generate many false alarms when a complicated background (including many edges and corners) is handled. In [28], Zhu *et al*. used SVM to classify vehicles based on the features of Gabor moments, edge area templates (EAT), and corner area templates (CAT). These two methods [27], [28] failed to detect vehicles with various orientations. As to the Schneiderman and Kanade's detector [29], they used lots of labeled training images to build the appearance models of the detected objects based on the wavelet feature. Eight detectors were designed for detecting vehicles having different orientation changes. Compare with these three methods, our proposed color classification scheme can eliminate over 80% non-vehicle pixels so that our approach has the best efficiency in vehicle detection. In addition, since our color classification scheme can pre-filter out many false alarms, our proposed method has higher tolerance to complicated backgrounds. It also makes our approach have the highest detection accuracy. All the quantitative analyses of these methods in vehicle detection are tabulated in Table 3. Table 4 lists all the detailed functional comparisons. The two methods proposed in [27]-[28] have limited abilities to handle vehicles having different orientations. In [29], since they used eight detectors to find all vehicles having different orientations, their time complexity is the highest. The speed of each method was measured on a Pentium CPU 2.4G with 512M memory. Since our scheme can filter out most of false candidates using the color feature, it has the best efficiency than other approaches. All the above results have proved that the proposed method is a robust, accurate, and powerful tool for vehicle detection.

Table 3: Performance comparisons among different methods.

| View, Scale, and Orientation<br>Method | Detection Accuracy | | |
|---|---|---|---|
| | ● Side view<br>● Fixed scale<br>● Fixed orientation | ● Front and rear view<br>● Multi-scale<br>● Fix orientation | ● Multi- views<br>● Multi- scales<br>● Multi- orientations |
| Our method | 95.7% | 95.1% | 94.9% |
| Agarwal and Roth [27] | 77.9% | X | X |
| Zhu et al. [28] | 81.3% | X | X |
| Schneiderman and Kanade [29] | 92.5% | 92.1% | 91.3% |

Table 4: Functional comparisons among different methods.

| Functionality<br>Methods | Multiple views, and orientation | Efficiency and Speed | Features | Number of hypotheses. |
|---|---|---|---|---|
| Our method | YES | High, 0.15 ~ 0.5sec | Color, Corners, WT, and Edges. | Pixels that pass the color classification |
| Agarwal and Roth [27] | NO | Low, 10 sec | Forstner interest operator | All the pixels and sub-windows. |
| Zhu *et al*. [28] | NO | Low, 5 sec. | Gabor moments. | Based on EAT and CAT templates |
| Schneiderman and Kanade [29] | YES | Low, 5 min | WT | Search the whole image exhaustively. |

Table 5: Frame rate analyses of our system among different video dimensions and functions.

| Functions<br>Dimension | Road Sign Detection | Road Sign Detection<br>+ Text Detection | Road Sign Detection<br>+Text Detection + Rectification |
|---|---|---|---|
| 320×240 | 20 *fps* | 16 *fps* | 8 *fps* |
| 640×480 | 5 *fps* | 4 *fps* | 2 *fps* |

## 4.2 Road Sign Detection Performance

To examine the performances of our proposed method, several video sequences collected from different highways and roads were used. The sequences were captured under different lighting and weather conditions (like sunny, cloudy, and rainy). The camera was mounted in the front of the car and its optical axis is not required being perpendicular to the road sign. A database including more than four thousand images was collected and constructed for examining the robustness of the proposed system. Our system was implemented and tested on a general PC with the Intel Pentium CPU 2.0G. The language for implementing our system is Microsoft visual C++6.0. Table 5 shows the frame rate analyses of our system when different video dimensions are handled and functions are added. When a 320×240 video frame is handled, the detection rate is 20 *fps*. When more functions are added or higher frame dimension are handled, the frame rate becomes worse.

The precision and false-alarm rates are defined for evaluating and measuring the performances of our proposed method to detect road signs. Precision is the ratio of the number of correctly detected road sign pixels to the number of exactly existing road sign pixels. False alarm rate is the ratio of the number of background pixels but misclassified as road sign to the number of all background pixels. These two measures are defined as:

$$Precision = C_{sign} \; / N_{sign} \quad \text{and} \quad Rate\ of\ False\text{-}Alarm = F_{sign} / N_{back\text{-}pixels},$$

where $N_{sign}$ is the total number of road sign pixels, $C_{sign}$ the number of correctly detected road sign pixels, $N_{back\text{-}pixels}$ the number of all background pixels, and $F_{sign}$ the number of background pixels but misclassified as road sign ones. When calculating these two measures, the ground truth of road sign pixels was manually obtained.

### 4.2.1 Road Sign Color Segmentation

To theoretically analyze which color coordinate system provides the best traffic sign color segmentation result, we use the "Fisher criterion" [19] to evaluate their separation abilities. The criterion uses the ratio of the "between-class" variance to the "within-class" variance to measure how well a transform $T$ can separate a space into two classes $C_1$ and $C_2$. The "between-class" variance is the distance between their means (denoted by $m_1$ and $m_2$, respectively). The "within-class" variance is the sum of their variances, i.e. $s_1$ and $s_2$. Then, the Fisher criterion is defined by

$$J(T) = \frac{\text{between-class distance}}{\text{within-class distance}} = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2}.$$

For a color domain, the larger value of $J$ is, the better separation ability it has. Table 6 lists the values of $J$ when different color spaces including RGB, YIQ, HSV, L*a*b, Luv, and our method were compared. Clearly, our method has the best $J$ among all the color domains.

Table 6: Separation ability analysis among different color spaces.

| Separation Ability | | RGB | YIQ | HSV | L*a*b | Luv | Proposed method |
|---|---|---|---|---|---|---|---|
| *J(T)* | Red | 0.81 | 1.41 | 0.01 | 4.11 | 4.12 | 4.62 |
| | Green | 0.49 | 0.70 | 0.20 | 2.13 | 1.75 | 3.52 |

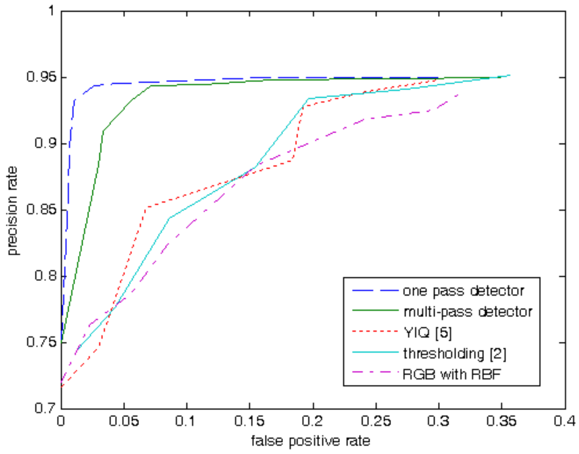
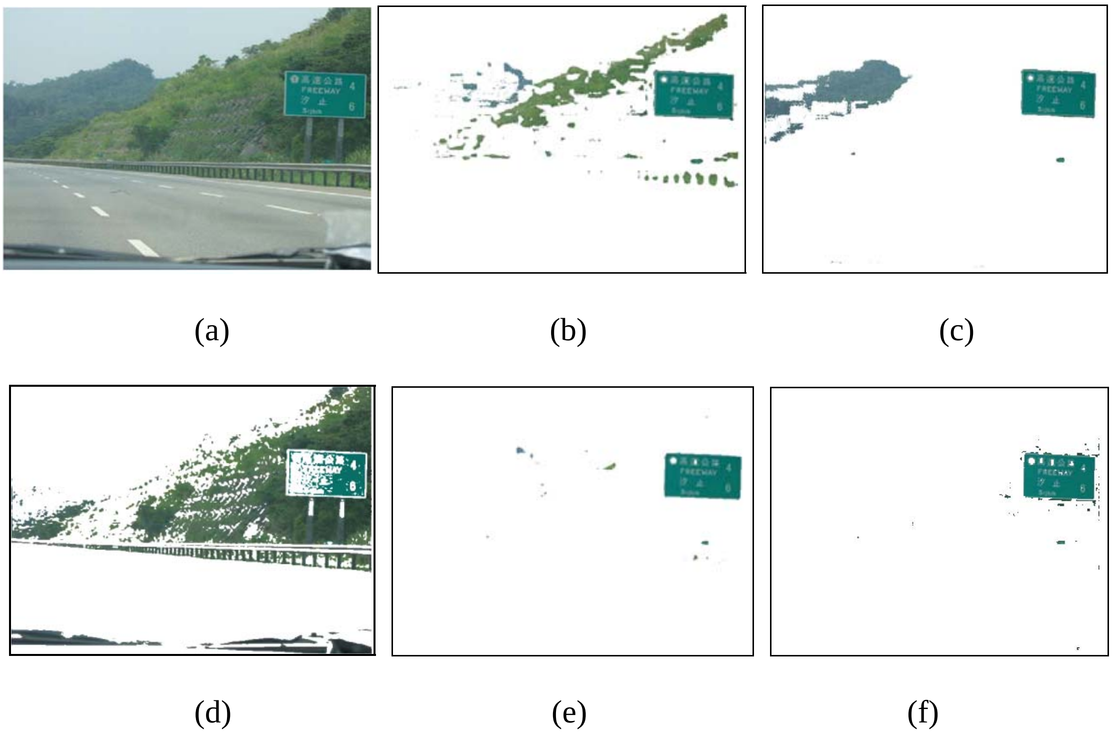


Fig. 31: ROC curve analysis among different methods.

In addition to color space, we also compared our method to other approached including the color thresholding technique [33] and the YIQ color module [36]. In our method, an RBF neural network was used for altering the effect of our proposed color transformation. For fair comparisons, the RBF neural network was also used to train a classifier on the RGB space for road sign detection. As to our detector, two different schemes were adopted for detecting road signs having different colors. For the first one, each road sign color has its corresponding detector which was trained at different time. Then, an “OR” operation was used to combine the detection results together. Since it needs multiple passes to locate a road sign, we name it a multiple-pass color detector in this thesis. For the second one, only one detector is trained for all types of road sign even though their colors are different. Since only one scanning pass is needed for road sign detection, it is named as one-pass detector in this thesis. Fig. 31 shows the performance analyses among different methods using ROC curves [56], [57] . Clearly, the one-pass detector performs the best among the above methods.

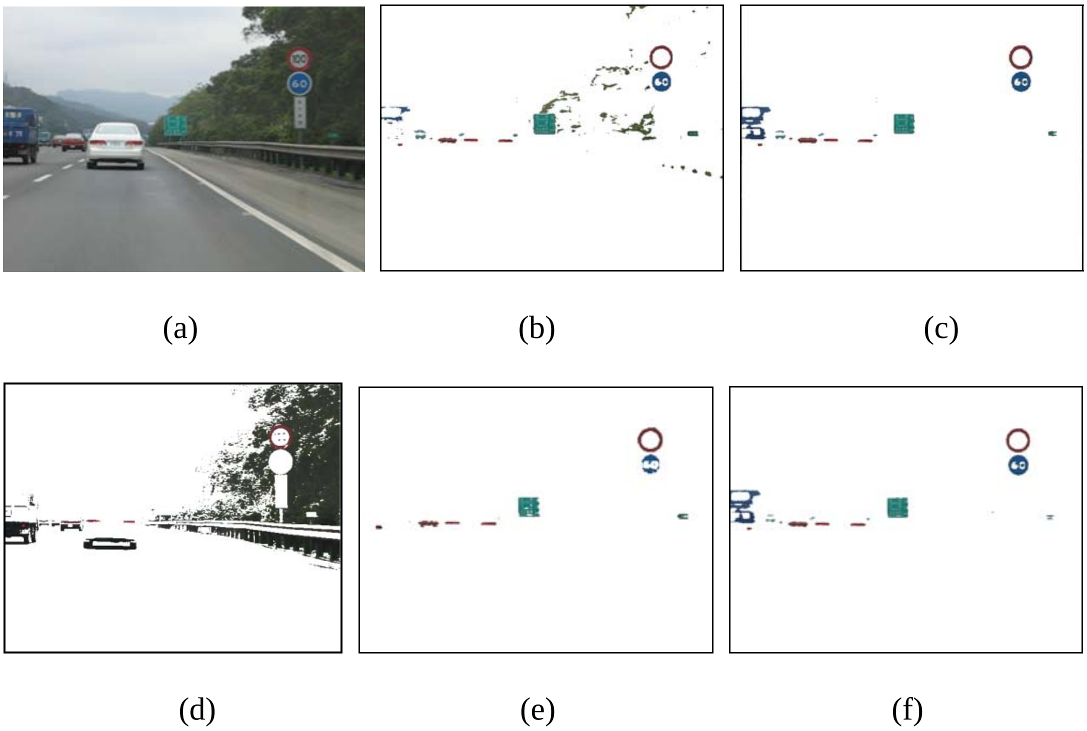


(a) (b) (c)

(d) (e) (f)

Fig. 32: Result of color classification. (a) Original image. (b) Classification result of the thresholding technique. (c) Result of the YIQ technique. (d) Result of the RBF classification on the RGB color space. (e) Result of the one-pass detector. (f) Result of the multiple-pass detector.

Fig. 32 shows the results of road sign color detection among these techniques. (a) is the original image. (b) and (c) are the results using the thresholding and YIQ techniques, respectively. The precision and false alarm rates of (b) and (c) are 95.1%, 94.8% and 19.7%, 18.4%, respectively. It is noticed that the mountain has a similar color to the road sign and thus there were many false road sign region detected in both (b) and (c). (d) is the result of the RBF classification on the RGB color space. Higher false alarm rate was obtained from (d). (e) is the result of the one-pass detector. The precision and false alarm rates of (e) are 93.2% and 2.62%, respectively. (f) is the detection result using the multiple-pass detector. The precision and false alarm rates of (f) are 94.4% and 2.97%, respectively. The precisions of both our schemes are similar to the threshold technique and the YIQ method. However, the false alarm rates of our schemes are much lower than the above three methods. Usually,


(a) (b) (c)

(d) (e) (f)

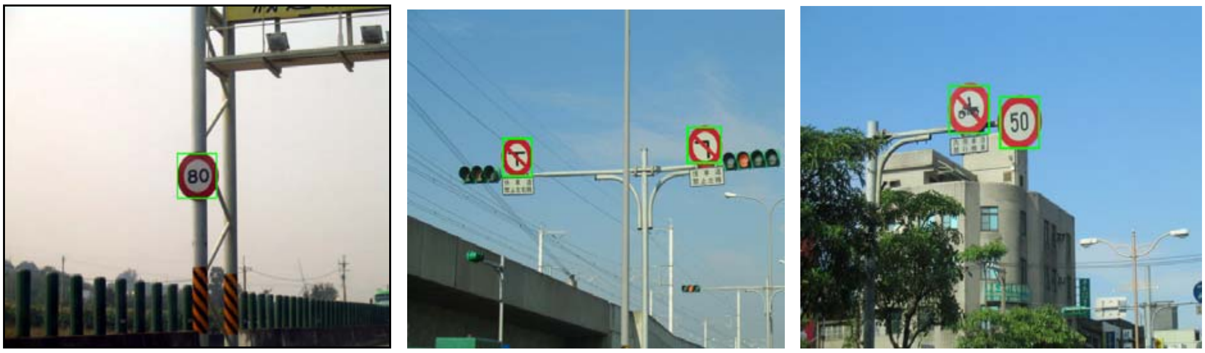
Fig. 33: Comparisons of color classification among different methods. (a) Original image. (b) Classification result using the thresholding technique. (c) Result of the YIQ technique. (d) Result of the RBF classification on the RGB color space. (e) Result of the one-pass detector. (f) Result of the multiple-pass detector.

a lower false alarm rate means less computation time for candidate verification. About the multi-pass detector, since we did not know which color a road sign had, two color detectors were implemented for detecting green and red road signs, respectively. Then, the two results were combined together using an “OR” operation. There is no significant performance difference between (e) and (f). However, the one-pass detector needs less scanning pass to detect desired road signs. Thus, it is more efficient than the multiple-pass detector. Compared with other methods, our proposed methods still had better performances.

Fig. 33 shows the case when multiple road signs appeared together in the same frame. (b) and (c) are the results of the thresholding and YIQ schemes, respectively. (d) is the result of the RBF classification on the RGB space for detecting road sign pixels. (e) and (f) are the results using our one-pass and multiple-pass schemes, respectively. Clearly, our proposed two schemes have lower false alarm rates and better accuracies than other schemes. Although the one-pass detector had less accuracy than the multiple-pass one, its false alarm rate is much lower than the multiple-pass one. Since the one-pass detector has better efficiency than the multiple-pass one, we adopt the one-pass detector for road sign color classification in all the following experiments.

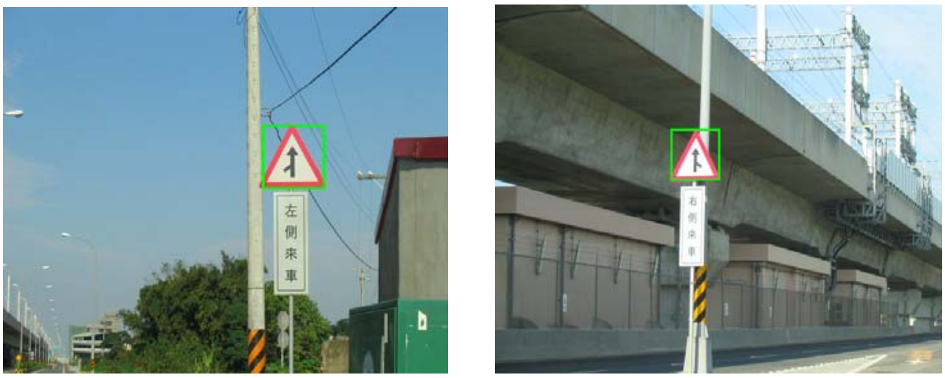
### *4.2.2 Road Sign Detection, Rectification and Text Extraction*

After detecting road sign pixels, the verification process is then applied for verifying each road sign candidate. Fig. 34 shows a series of detection results when circular road signs were handled. All these circular road signs were correctly detected. Fig. 35 shows the detection results when two triangular roads were handled. Fig. 36 shows the cases when multiple rectangular road signs appeared together in the analyzed scene. In (b) and (c), even though the road signs had similar colors to the sky, our method still worked very well to find them. According to the above results, clearly no matter what type a road sign is, our proposed method works very well to detect it.



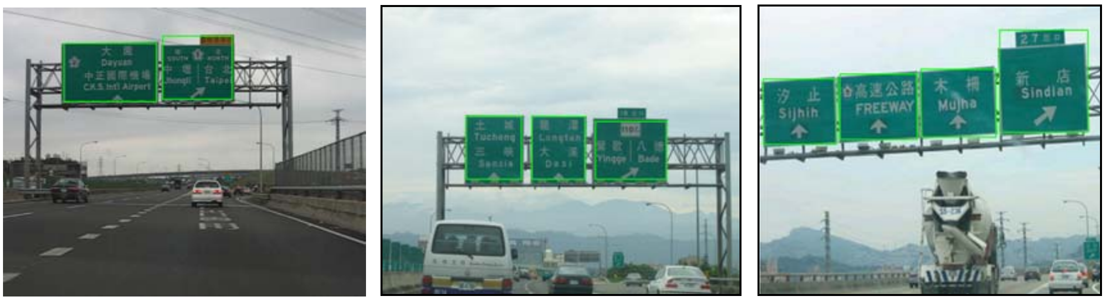
(a) (b) (c)

Fig. 34: Detection results of circular road signs.



(a) (b)

Fig. 35: Detection results of two triangular road signs.



(a) (b) (c)

Fig. 36: Detection results when multiple rectangular road signs appeared in the same scene.

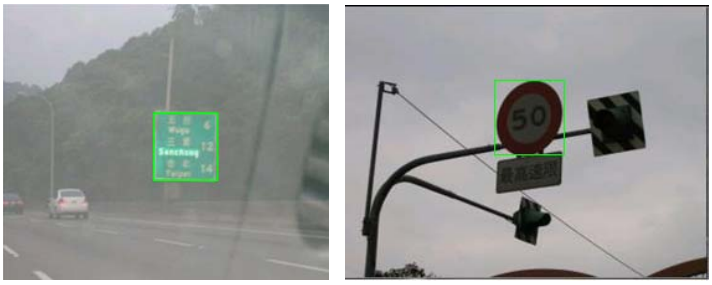
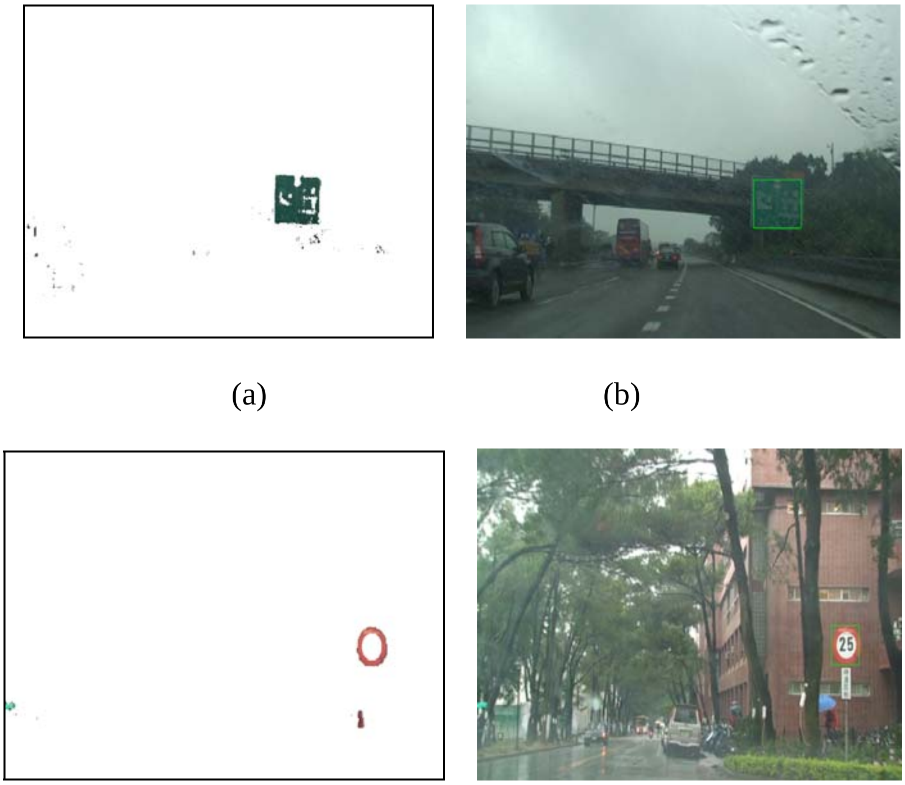


Fig. 37: Results of road sign detection when low contrast video frames were handled.

Fig. 37 is another case when a low-contrast frame was handled. Even though the frame contrast was low, each road sign was still successfully detected using our proposed method. Fig. 38 shows the detection results when rainy days were handled. (a) and (c) are the detect results of road signs. (b) and (d) are the results of color classification using our proposed method. Our method is robust to detect road signs when different weather conditions are handled. Fig. 39 shows the case when a skewed road sign was handled. (a), (b), and (c) are the detection results using our proposed scheme. (d), (e), and (f) are the rectification results of (a), (b), and (c), respectively. Clearly, no matter how skewed the road signs are and what color they have, the proposed rectification scheme can handle all of them correctly.

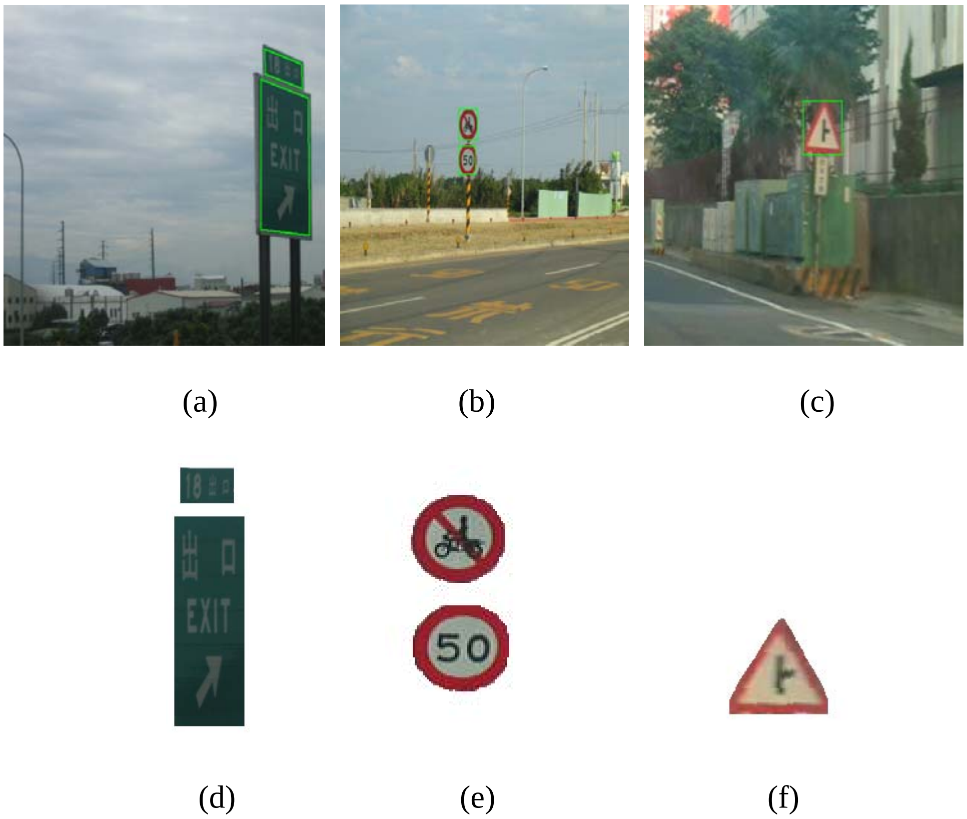


(a) (b)

(c) (d)

Fig. 38: Results of road sign detection when rainy days were handled. (a) and (c): results of color classification using our proposed method. (b) and (d): results of road sign detection.

The next set of experiments was used to demonstrate the performances of our method to detect road signs under different weather conditions in a sequential video. Fig. 40 shows a series of detection results when consecutive video frames under a cloudy day were handled. In (a) and (b), a smaller and darker road sign was detected. Then, its size gradually became larger. (c) shows the detection result of a larger road sign. Experiment results shows that the proposed method is reliable and suitable for real time issue. All its variations were still successfully detected.

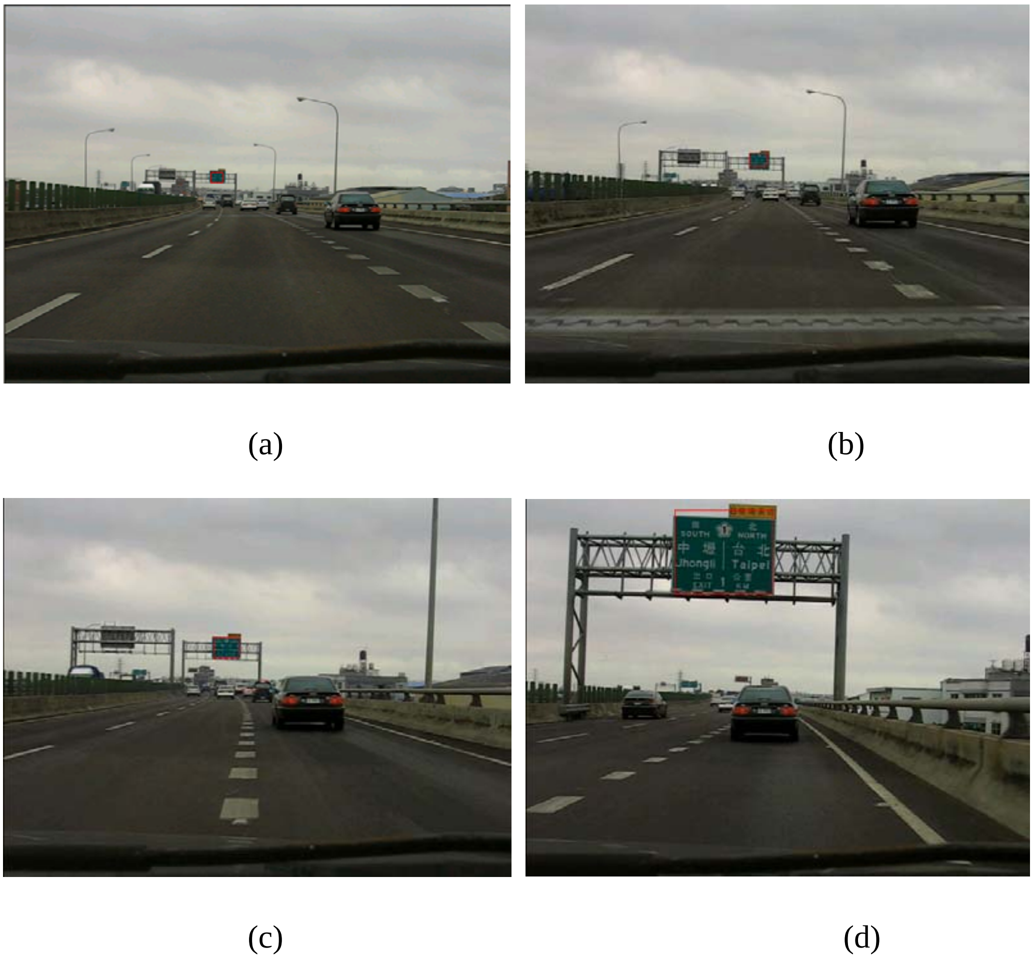


(a) (b) (c)

(d) (e) (f)

Fig. 39: Detection result when a skewed road sign was handled. (a), (b), and (c): results of road sign detection. (d), (e), and (f): Results of rectification.

After road sign detection, the embedded texts on a road sign can be then extracted though a bi-leveling technique (see Section 3.5) and a connected component analysis. Fig. 41 shows the results of text line extraction from road signs under different lighting conditions. (a) is the result of a normal road sign. (b) and (c) are the text extraction results obtained from blurred images. (d) is the text result extracted from a dark road sign. Each desired text line was correctly detected using our proposed method.

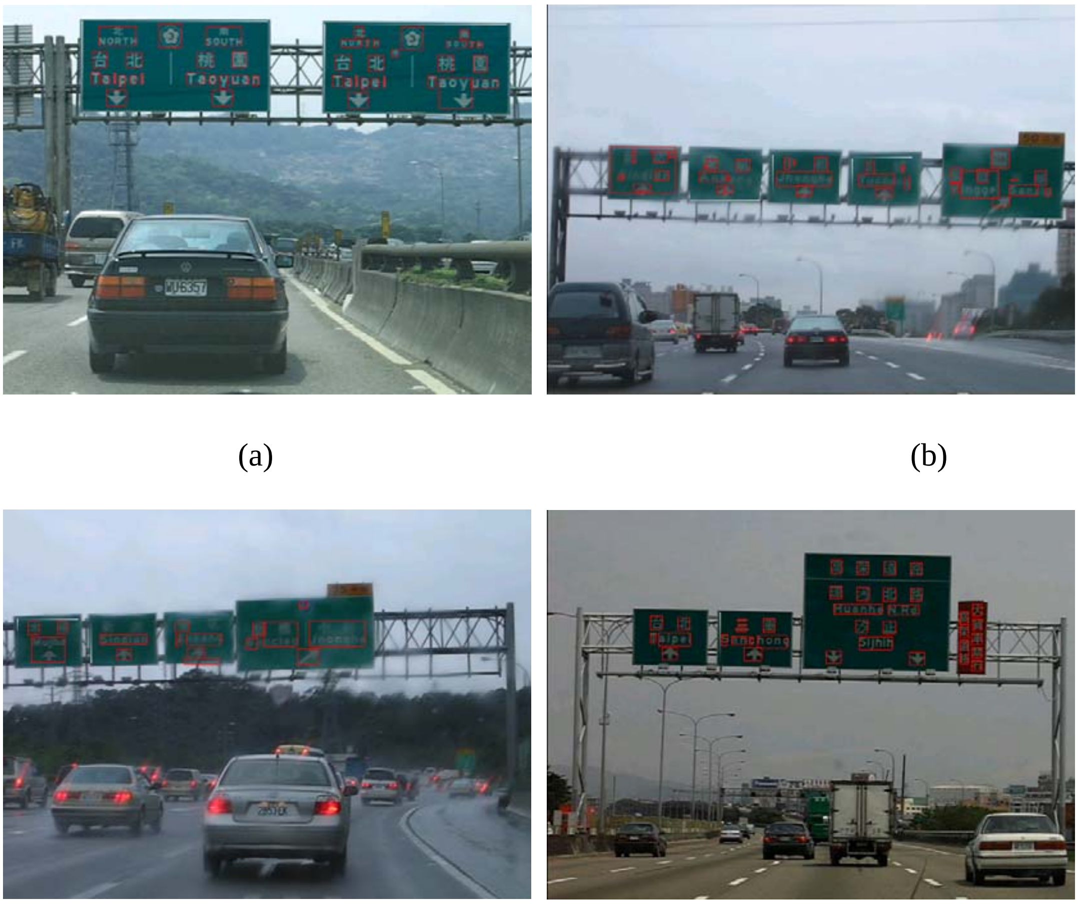


(a) (b)

(c) (d)

Fig. 40: Road sign detection in a video sequence under a cloudy day.

The database we used included 502 roads signs which come from 20 individual videos. The number of correctly detected road signs is 480. In addition, the number of falsely-detected road signs is 27 and the number of missed road signs is 22. After calculation, the accuracy of our proposed method is 95.6%. The false alarm rate and missed rate are 5.32% and 4.38%, respectively. All the above results have proved that the proposed method is a robust, accurate, and powerful tool in road sign detection.



(a) (b)

(c) (d)

Fig. 41: Results of text line detection. (a) Normal road signs. (b) and (c): Blurred road signs. (d) Dark road signs.

# Chapter 5. Conclusions and Future Works

## 5.1 Conclusions

This thesis has presented a novel color-based object detection method for detecting objects directly from static images. First of all, an eigen color transform is proposed for making object colors more sufficiently concentrated on a compact area. Then, different object pixels will form different clusters if they are projected on this eigen color space. Two classifiers were trained in this thesis for detecting vehicles and road signs, respectively. With these classifiers, lots of redundant no-object pixels can be filtered out in advance and only few candidates should be further verified. At the verification stage, three visual features including edge maps, coefficients of wavelet transform, and corners were integrated to form a multi-channel classifier. Then, each desired object can be very accurately detected from static image. With a coarse-to-fine verification scheme plus an image pyramid, desired objects can be accurately and effectively detected and verified even though they are with various scale and orientation changes. Due to the filtering effect of the proposed color transformation, the searching domain of object candidates can be significantly narrowed down. Thus, the proposed framework can detect each object efficiently even without using any motion feature. Two targets, i.e., vehicles and road signs were demonstrated for proving the flexibilities of our proposed transform in object detection. Different weather conditions and road scenes were testing in this thesis for proving that our method is robust and effective. Several quantitative comparisons among our method and some well-known approaches were also performed. All the experiment results show that the proposed method is superior to other approaches for detecting objects in terms of accuracy, robustness, and stability.

## 5.2 Future Works

Recently, detecting the important objects in traffic images is the active research of Intelligence Transportation System (ITS). Forward-looking image processing for driver assistance has attracted more studies to date. Generally, the problem focuses on two main topics: the detection of roads and the detection of obstacles, vehicles or traffic signs. In this thesis, we have present reliable vehicle and traffic sign detection results. Our work can be extended to combine the lane extraction module yielding the promising vision-based driver systems. Moreover, we are trying to extend our eigen color method for traffic scene analysis in real-time. It is useful to analyze the whole traffic images and segments into some meaningful regions. The lower part of the road scene image contains more important objects than upper part. Therefore, the background objects such as sky, mountains and trees in a road scene can be ignored in the upper parts. Our future work will explore the framework to integrate the other ITS applications like lane and road area detection for improving the driver's safety.

Another improvement is to utilize the component-based classifier in our cascade structure. For example, in the cases of vehicles, the major components should be wheels, the headlight and taillight. We trained a set of view-turned, global classifier to achieve view invariance in our system. In fact, it is difficult to collect complete training set, which cover the all possible variations of objects especially for partially occlusion. Building the classifier based the components is a solution to decrease the number of required training examples. Furthermore, we can also decrease the computation in global approach which needs to verify the region of interest with each view-turned classifier.

# References

[1] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 5, pp. 694-711, May 2006.

[2] V. Kastinaki, M. Zervakis, and K. Kalaitozakis, "A survey of video processing techniques for traffic applications," *Image, Vision, and Computing*, vol. 21, no. 4, pp.359-381, April 2003.

[3] R. Cucchiara, P. Mello, and M. Piccardi, "Image analysis and rule-based reasoning for a traffic monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp.37-47, March 2002.

[4] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, vol. 2, pp.119-130, June 2000.

[5] I. Haritaoglu, D. Harwood, and L. Davis, "W4: who? when? where? what? a real time system for detecting and tracking people," *International Conference on Face and Gesture Recognition*, pp.222-227, 1998.

[6] G. L. Foresti, V. Murino, and C. Regazzoni, "Vehicle recognition and tracking from road image sequences," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 1, pp.301-318, Jan. 1999.

[7] J. Wu, X. Zhang, and J. Zhou, "Vehicle detection in static road images with PCA-and-wavelet-based classifier," *IEEE Intelligent Transportation Systems Conference,* Oakland, C.A., USA, pp.740-744, , Aug. 25-29, 2001.

[8] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines," *IEEE International Conference on Digital Signal Processin*g**,** Santorini, Greece, pp.1019-1022, July 2002.

[9] A. Broggi, P. Cerri, and P. C. Antonello, "Multi-resolution vehicle detection using artificial vision," *IEEE Intelligent Vehicles Symposium*, pp. 310- 314, June 2004.

[10] M. Bertozzi, A. Broggi, and S. Castelluccio, "A real-time oriented system for vehicle detection," *Journal of Systems Architecture*, pp. 317-325, 1997.

[11] C. Tzomakas and W. Seelen, "Vehicle detection in traffic scenes using shadow," Tech. Rep. 98-06, Institut fur neuroinformatik, Ruhtuniversitat, Bochum, Germany, 1998.

[12] A. Lakshmi Ratan, W.E.L. Grimson, and W.M. Wells, "Object detection and localization

by dynamic template warping," *International Journal of Computer Vision*, vol. 36, no. 2, pp.131-148, 2000.

[13] A. Bensrhair, A. Bertozzi, A. Broggi, A. Fascioli, S. Mousset, and G. Toulminet, "Stereo vision-based feature extraction for vehicle detection," *IEEE Intelligent Vehicles Symposium*, vol. 2, pp. 465-470, June 2002.

[14] T. Aizawa, A. Tanaka, H. Higashikage, Y. Asokawa, M. Kimachi, S. Ogata, "Road surface estimation against vehicles' existence for stereo-based vehicle detection," *IEEE 5th International Conference on Intelligent Transportation Systems,* pp. 43-48*,* Sep. 2002.

[15] J. C. Rojas and J. D. Crisman, "Vehicle detection in color images," *IEEE Conference on Intelligent Transportation System*, pp.403-408, Nov. 9-11, 1997.

[16] D. Guo, T. Fraichard, M. Xie, and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," *IEEE Intelligent Vehicles Symposium*, pp. 249-254, Oct. 2000.

[17] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, pp. 222-241, 1980.

[18] G. Healey, "Segmenting images using normalized color," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 1 , pp. 64-73, 1992.

[19] R. O. Duda, P. E. Hart, and D. G. Stork. "Pattern classification", *John Wiley & Sons, New York,* 2001.

[20] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," *Proceedings of the Fourth Alley Vision Conference*, Manchester, pp. 147-151, 1988.

[21] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.

[22] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision,* vol. 57, no. 2, pp. 137-154, May 2004.

[23] Stan Z. Li, L. Zhu, ZQ Zhang, A. Blake, HJ Zhang, and H. Shum, "Statistical Learning of Multi-View Face Detection," *Proceedings of the 7th European Conference on Computer Vision,* vol. 2353, pp.67-81, 2002.

[24] K.K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39-51, 1998.

[25] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15-33, 2000.

[26] R. E. Shapire and Y. Singer, "Improving boosting algorithms using confidence-rated

predictions," *Machine Learning*, vol. 37, no. 3, pp. 297-336, Dec. 1999.

[27] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," *Proceedings of the European Conference on Computer Vision*, vol. 4, pp. 113-130, May 2002.

[28] Z. Zhu, H. Lu, J. Hu, and K. Uchimura, "Car detection based on multi-Cues integration," *17th International Conference on Pattern Recognition,* vol. 2, pp. 699-702, Aug. 2004.

[29] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *International Journal of Computer Vision*, vol. 45, no. 3, pp.151-177, Feb. 2004.

[30] Y. Park, "Shape-resolving local thresholding for object detection," *Pattern Recognition Letters*, vol.22, no. 8, pp. 883-890, June 2001.

[31] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo vision-based vehicle detection," *IEEE Intelligent vehicle symposium*, pp. 39-44, Oct. 2000.

[32] M. Bénallal and J. Meunier, "Real-time color segmentation of road signs," *Proceedings of IEEE Conference on Electrical and Computer Engineering*, vol. 3, pp.1823–1826, May 2003.

[33] A. de la Escalera, L. E. Moreno, M. A. Salichs, and José María Armingol, "Road traffic sign detection and classification," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp.848- 859, Dec.1997.

[34] C. Y. Fang, S. W. Chen, and C. S. Fuh, "Road-sign detection and tracking," *IEEE Transactions on Vehicular Technology*, vol. 52, no.5, pp.1329-1341, Sep. 2003.

[35] A. D. L. Escalera, J. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image and Vision Computing*, vol. 21, pp. 247-258, 2003.

[36] N. Kehtarnavaz and A, Ahmad, "Traffic sign recognition in noisy outdoor scenes," *Proceedings of Intelligent Vehicles '95 Symposium*, pp.460-465, Sep. 1995.

[37] S. Vitabile, G. Pollaccia, G. Pilato, and E. Sorbello, "Road signs recognition using a dynamic pixel aggregation technique in the HSV color space," *Proceedings of IEEE International Conference on Image Analysis and Processing*, pp.572 – 577. Sep. 2001.

[38] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes," *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp.87–99, 2004.

[39] J. Miura, T. Kanda, and Y. Shirai, "An active vision system for real-time traffic sign recognition," *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, pp. 52-57, Dearborn, MI, Oct. 2000.

[40] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the

characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, 1990.

[41] G. Healey, "Segmenting images using normalized color," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no.1, pp. 64-73, 1992.

[42] Luo-Wei Tsai, Jun-Wei Hsieh, and Kao-Chin Fan, "Vehicle detection using normalized color and edge map," *IEEE Transactions on Image Processing,* vol. 16, no. 3, pp.850-864, Mar. 2007.

[43] W. Wu, X. Chen, and J. Yang, "Detection of text on road signs from video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 378- 390, Dec. 2005.

[44] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," *Proceedings IEEE Intelligent Vehicles Symposium*, Italy, pp. 566- 571, June 2004.

[45] G Piccioli, E De Micheli, P Parodi, and M Campani, "Robust method for road sign detection and recognition," *Image and Vision Computing*, vol.14, no. 4, pp.209-223, 1996.

[46] E. D. Haritaoglu and I. Haritaoglu, "Real time image enhancement and segmentation for sign/text detection," *Proceedings of the 2003 IEEE Int. Conf. Image Processing*, Barcelona, Spain, pp. 993–996, 2003.

[47] C. Bahimann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information," *Proceedings of IEEE Intelligent Vehicles Symposium*, pp.255-260, June 2005.

[48] M. D. S. Blancard, "Road sign recognition: a study of vision-based decision making for road environment recognition," in I. Masaki (ed.), *Vision-based Vehicle Guidance*, Springer-Verlag, Berlin, pp. 162- 172, 1992.

[49] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, Nov. 1986.

[50] J.-W. Hsieh, "Fast stitching algorithm for moving object detection and mosaic construction," *Image Vision and Computing Journal*, vol. 22, no. 4, pp. 291-306, April 2004.

[51] M. Sonka, V. Hlavac, and R. Boyle, "Image Processing, Analysis and Machine vision," *Brooks/Cole Publishing Company,* 1999.

[52] H. Fleyeh, "Color detection and segmentation for road and traffic signs," *IEEE Conference on Cybernetics and Intelligent Systems*, vol. 2, pp.809-814, 2004.

[53] S. Vitabile, A. Gentile, S. M. Siniscalchi, and F. Sorbello, "Efficient rapid prototyping of

image and video processing algorithms," *Euromicro Symposium on Digital System Design*, pp.452-458, 2004.

[54] A. de la Escalera, J. M. Armingol, J. M. Pastor, and F. J. Rodriguez, "Visual sign information extraction and identification by deformable models for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, pp.57-68, 2004.

[55] M. A. Garcia-Garrido, M. A.Sotelo, and E. Martin-Gorostiza, "Fast traffic sign detection and recognition under changing lighting conditions," *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp.811-816, 2006.

[56] M. CE, "Basic principles of ROC analysis," *Seminars in Nuclear Medicine*, vol.8, pp.283-298, 1978.

[57] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, vol. 39, pp. 561-577, 1993.

[58] JULIUS T. Tou, and RAFAEL C. GONALEZ, Pattern recognition principles, *Addison-Wesley Publishing Company,* 1974.