Detection of White Arrows in Blue Directional Signs Segmented from Blue Sky Scenes Using Blue-Yellow Opponent Color Filters

by

Mohammed Hayyan ALSIBAI

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Engineering

in

Computer Science

in the

Graduate School of Systems and Information Engineering
of the
University of Tsukuba, JAPAN

Thesis advisor:
Prof. Yuzo HIRAI

(March 2012)

Detection of White Arrows in Blue Directional Signs Segmented from Blue Sky Scenes Using Blue-Yellow Opponent Color Filters

Graduate School of Systems and Information Engineering

University of Tsukuba

March 2012

Mohammed Hayyan ALSIBAI

To those who sacrifice their lives
for the sake of ALLAH and SYRIA
I dedicate this work.

. . .

15/2/2012

Abstract

Traffic accidents are a serious worldwide problem. Therefore, safety driving systems and driver assistant systems are becoming more popular research field to reduce accidents. Traffic sign recognition system is a major part of these researches.

Traffic sign recognition is not a new field of research. It has been studied for more than three decades. In general, traffic signs can be grouped into several types such as danger warning signs, priority signs, prohibitory or restrictive signs and mandatory signs.

This research is dedicated to a real-time recognition system of blue traffic signs indicating directions using an in-vehicle video camera. These signs are important because they guide drivers to follow the mandatory directions and to choose a correct lane on the road. Missing this information may lead to accidents.

The recognition system presented in this thesis consists of three main processing stages. The first stage is to label the blue objects in each frame and segment them by cropping into rectangular areas circumscribing them. Next stage is to verify if the segmented blue object is a sign candidate or not. The third stage is classification. If the blue object is verified as a sign candidate, inner white objects, which are arrow candidates, are segmented and classified to one of the target patterns or rejected. Besides these three main stages the recognition system may be enhanced by applying a tracking algorithm which will facilitate detection of the same signs in consecutive video frames.

There are difficult problems in every processing step such as determination of sign positions, variations in ambient light, signs' color fading, the vehicle speed, and real-time constraint. Besides these challenges, a blue traffic sign recognition system has two special difficulties. The first is sign-sky discrimination in the labeling stage. The existence of the sky as a blue background of the blue objects makes popular color segmentation algorithms insufficient. The second difficulty is fragmentary segmentation of signs due to motion blur or some lighting conditions. One important aspect of the work presented in this thesis is to propose solutions for these two special problems.

For sign-sky discrimination, two solutions have been presented. The first solution is to use a multi-threshold labeling technique according to the brightness of the image. The second solution is to use blue-yellow type opponent color filters trying to discriminate between color components of sky and signs. For fragmentary segmentation of signs, two solutions have been proposed also. The first solution is to apply a structured dilation

algorithm to aggregate fragmentations. In the second solution, aggregation is performed by searching for fragmented parts in the neighboring areas. This is carried out in two steps to achieve real-timeness. The first step predicts the searching direction to speed up the aggregation. If the first step fails, the second step is applied. Larger searching area is considered this time and the shape information of the sign and arrows is taken into account to discard noisy parts.

Two arrow classification methods have been considered in this thesis. The first one is to use a decision tree according to geometrical features. However, when the aggregation algorithm was changed from dilation method to more robust searching method, variations in segmented shapes of signs became large and the hand-coding of the decision tree became more difficult. To absorb those variations, a subspace method using principal component analysis (PCA) with histogram of oriented gradients (HOG) features is employed. More than one subspace for each arrow class is used to cover complex distributions of arrow patterns.

The work presented in this thesis provides an in-depth analysis of the main three processing stages by comparing the performance of the proposed methods and how they improve the performance of the system. The performance of final version of the system is as follows: Precision is 75%, recall is 93% and F-measure is 83%. Average processing time is 190 ms and the maximum is 250 ms.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Traffic accidents are a serious global problem which leads to severe injuries. According to World Health Organization (WHO) [1], a road traffic injury is defined as a fatal or non-fatal injury incurred as a result of a collision on a public road involving at least one moving vehicle. Children, pedestrians, cyclists and the elderly are among the most vulnerable of road users.

The *World report on road traffic injury prevention* [2] was jointly launched by World Health Organization and the World Bank in 2004. A key purpose of this report, as mentioned in its foreword, was to communicate knowledge and thinking about prevention of road injury to a wider audience involved in managing road safety.

On October 26, 2005, the United Nations General Assembly called for the third Sunday in November to be considered as the annual day of remembrance for road traffic victims as a way to draw attention to road traffic accidents [3].

In 2009, WHO published the *Global status report on road safety* [4]. The statistics in the report showed that road traffic injuries kill nearly 1.3 million people annually, and cause between 20 and 50 million sustain non-fatal injuries. In Japan, for example, police data reported 6,639 fatal and 1,034,445 non-fatal road traffic injuries in 2007. According to the same report, road traffic injuries are one of the top three causes of death for people aged between 5 to 44 years. Beyond the sufferings they cause, traffic accidents have a socioeconomic effects. They may drive a family into poverty as crash survivors and their families struggle to cope with the long-term consequences of the event. They have to face medical costs, funeral costs, vehicle repair or losing the householder. Moreover, traffic accidents have economic impacts. The global economic cost of motor vehicle collision was

estimated at 518 billion US-dollar per year. Costs include public property damages, vehicle repair, cost of towing services, costs associated with police investigation, legal activities and insurance administration. The report states that in most regions of the world this epidemic of road traffic accidents is still increasing. Therefore, addressing road safety in a comprehensive manner necessitates the involvement of multiple sectors.

According to the *World report on road traffic injury prevention* [2], since the last major WHO world report on road safety issued in 1962 [5], and because of the increasing international interest, there has been a major change in the perception, understanding and practice of road injury prevention – a shift of paradigms – among traffic safety professionals around the world. This new understanding include the following points:

- Road crash injury is largely preventable and predictable; it is a human-made problem amenable to rational analysis and countermeasure.
- Road safety is a multi sectoral issue and a public health issue. Therefor, all sectors including health, need to be fully engaged in responsibility, activity and advocacy for road crash injury prevention.
- Common driving errors and common pedestrian behavior should not lead to death and serious injury. The traffic system should help users to cope with increasingly demanding conditions.
- The vulnerability of the human body should be a limiting design parameter for the traffic system and speed management is central.
- Road crash injury is a social equity issue. Equal protection to all road users should be aimed for since non-motor vehicle users bear a disproportionate share of road injury and risk.
- Technology transfer from high-income to low-income countries needs to fit local conditions and should address research-based local needs.
- Local knowledge needs to inform the implementation of local solutions.

The report outlines the efforts needed to respond to this problem. They might be summarized as following:

- A scientific approach to the problem.

- The provision, careful analysis and interpretation of data.
- The setting-up of targets and plans.
- The creation of national and regional research capacity.
- Institutional cooperation across sectors.

As part of the scientific approach, Intelligent Transportation Systems (ITS) are becoming more popular research field. The concept of ITS is to link road infrastructure and vehicles with information and communication technologies [6]. Governments, industrial bodies, research societies and academics are more involved in ITS currently. For example, Intelligent Transportation Society of America (ITSA) [7] was established in 1991 as a not-for-profit organization to foster the use of advanced technologies in surface transportation systems. ITSA members include private corporations, public agencies, academic institutions and research centers involved in the research, development and design of ITS technologies that enhance safety, increase mobility and sustain the environment. Similar local and regional organizations were established all over the world to achieve similar tasks. The list includes ERTICO (ITS Europe), ITS Australia, ITS-Japan, AITS India and ITS Malaysia. The full list and links to each organization are available on the website of ITS Japan [8].

The Intelligent Transportation Society of America, ERTICO-ITS Europe, and ITS Asia-Pacific organize annual ITS World Congress that is an international conference and exhibition of ITS products and services.

Other scientific and research societies organize many conferences and symposium related to ITS. For example, IEEE Intelligent Transportation Systems Society (ITSS) sponsors a number of conferences and events like: IEEE Intelligent Vehicular Symposium, IEEE International Conference on Vehicular Electronics and Safety and International IEEE Conference on Intelligent Transportation Systems. Besides these activities, a number of journals, magazines and other professional publications are issued.

Driver assistance systems, which are an important section of ITS, were studied by many researchers. These systems are designed to increase driver's awareness and therefore his or her safety. Traffic sign recognition system is a major part of these researches. The aim of such system is to assist drivers by providing information on traffic signs. This thesis is dedicated to a real-time recognition system of blue traffic signs indicating directions using an in-vehicle video camera.

1.1 Research aim

Among the main causes for car accidents are reckless, negligent driving and alcohol. Another important reason is inattention of drivers. Many people talk, listen to radio or music, use mobiles, eat, or do other activities while driving. This might be fatal because it distracts drivers from the traffic. The main objective of the research presented in this thesis is the development of a real-time recognition system of blue traffic signs indicating directions.

In general, traffic signs can be grouped into several types such as danger warning signs, priority signs, prohibitory or restrictive signs and mandatory signs. They guide the driver to follow the traffic rules to insure his or her safety. Figure 1.1 shows examples of the target signs in this thesis. These signs are important to help drivers to follow the mandatory directions and to choose a correct lane on the road. The missing of such information might lead to accidents.

Figure 1.1: Examples of blue traffic signs designating directions used in Japan.

Figure 1.2: Using an in-vehicle camera for the traffic sign recognition system.

Actually, this work is a part of a general intelligent transportation system for a real-time recognition of road traffic signs [9]. The system uses an in-vehicle video camera as shown in Figure 1.2 to develop an integrated system which may be used mainly in:

- A warning system to assist drivers not to miss the traffic signs unconsciously.

Other applications of this system may include :

- Car navigation systems: Usually car navigation systems rely on pre-stored data which need to be updated from time to time. Real-time traffic sign recognition system may be used in a system for finding the best way and/or best lane according to the current scene without relying on old data.
- Robotic systems: Robot drivers can use this system.

1.2 Motivation and challenges

The recognition system presented in this thesis consists of three main processing stages. The first stage is to label the blue objects in each frame and segment them by cropping into rectangular areas circumscribing them. Next stage is to verify if the segmented blue object is a sign candidate or not. If the blue object is verified as a sign candidate, inner white objects, which are arrow candidates, are segmented. The third stage is classification. Segmented white objects are classified to one of the target patterns or rejected. Besides these three main stages, a tracking algorithm may be applied to ease the detection of the same signs in consecutive video frames.

There are difficult problems in every processing step which need to be solved. They are:

- Determination of the position of the traffic sign in the scene: Sign positions are not fixed. They should be placed in obvious positions for the drivers especially if there are obstacles in the scene. Moreover, because of the car movement, the sign positions are changing relatively with camera position.
- Variations in ambient light and illumination during the day through morning, noon, evening, and night.
- Variations in ambient light and illumination according to the sun position corresponding to the position of the sign and the camera.
- Weather conditions such as rain, snow, cloudy and shiny day.
- Color fading of signs.
- The vehicle speed.

- Real-time constraint.

Besides the above mentioned challenges, blue traffic sign recognition system has two special difficulties:

1. The problem of sign-sky discrimination in the labeling stage. The existence of the sky as a blue background of the blue objects makes popular color segmentation algorithms insufficient.

2. The fragmentary segmentation of signs due to motion blur and/or lighting conditions.

In this thesis, solutions to these problems are proposed.

1.3 Solutions and original contributions

With the motivation of solving the problems mentioned in Section 1.2, the following solutions are proposed in this thesis:

1. For sign-sky discrimination, two solutions are proposed and discussed:

   - Multi-threshold labeling technique according to the brightness of the image [10].
   - Using blue-yellow opponent color filters and seeded growing algorithm for labeling. Real-time constraint was taken into account by confining the area of the computation to the blue area determined by HSV values [11].

2. For fragmentary segmentation of the signs, two solutions are proposed and discussed also:

   - Aggregation of fragmented parts using structured dilation algorithm [10].
   - Using two-step algorithm. In step 1, the neighboring areas are searched to find fragmented parts. Real-time processing is achieved by predicting the scanning direction [11]. In step 2, shape information of arrows (inner parts) and signs (outer shape) are used to decide if segmented objects are fragmented parts of one object [12].

3. Since the car is moving, the size of the sign changes according to the distance between the sign and the camera. To avoid the need of size normalization of the images,

geometrical features such as the height (H) and width (W) of the segmented images are used [10][11].

In the following parts of the thesis, the merits of using the above mentioned methods are described by comparing their processing speed and performance to other methods.

1.4 System developing environment and tools

The working environment and the experiment conditions are summarized as follows:

- Road scenes are taken by a video camera (Panasonic AG-DVX100A) mounted on a tripod firmly fixed at the front seat of a car as shown in Figure 1.3.

Figure 1.3: The video camera is mounted on a tripod firmly fixed at the front seat of a car.

- Videos are recorded and converted to '.avi' files to be processed. Also they are directly processed in real-time by a notebook PC (Dell Inspiron 1520, 2GB memory, Core2Duo processor).
- Image size is 720 x 480 pixels.
- Frame processing is done by a laboratory-developed software called Image Filter Suite which is explained in more details in Appendix A. Image Filter Suite consists of different projects developed using Microsoft Visual Studio C++, which produce DLLs as filters to be applied to each frame. Each filter is used in two execution programs:

CameraVideoEffector.exe: for video processing.

ImageFilter.exe: for single video frame image processing. Frame images are captured from the recorded videos.

The user interface of the two programs are shown in Figures 1.4 and 1.5.

1.5 Structure of the thesis

In literature of traffic sign recognition, the processing steps are usually divided into two main stages which are detection and classification [13][14][15]. Some researchers add a third stage which is tracking [16][17].

In this study, the detection stage is divided into labeling and verification stages to explain the related problems and the proposed solutions in detail.

The rest of the thesis consists of the following chapters:

- Chapter 2 covers related works that are similar to the system presented in this research. Other systems with different techniques, methods and algorithms are discussed.
- Chapter 3 presents the system in general with brief explanation of the main processing stages. The algorithms used in each stage is explained by emphasizing the relationship among them. The main stages are then explained one by one in detail in the following chapters.
- Chapter 4 discusses the first main stage which is labeling and segmenting of blue objects. Two solutions for the problem of sign-sky discrimination are discussed in detail and compared to other algorithms. Experiments which were carried out to study the effects of illumination and to decide thresholds are also described in this chapter.
- Chapter 5 discusses the second main stage which is verification of sign candidates. The rules used to decide if the segmented blue object is a sign candidate or not are explained. The problem of fragmentary sign segmentation and two solutions are also discussed.

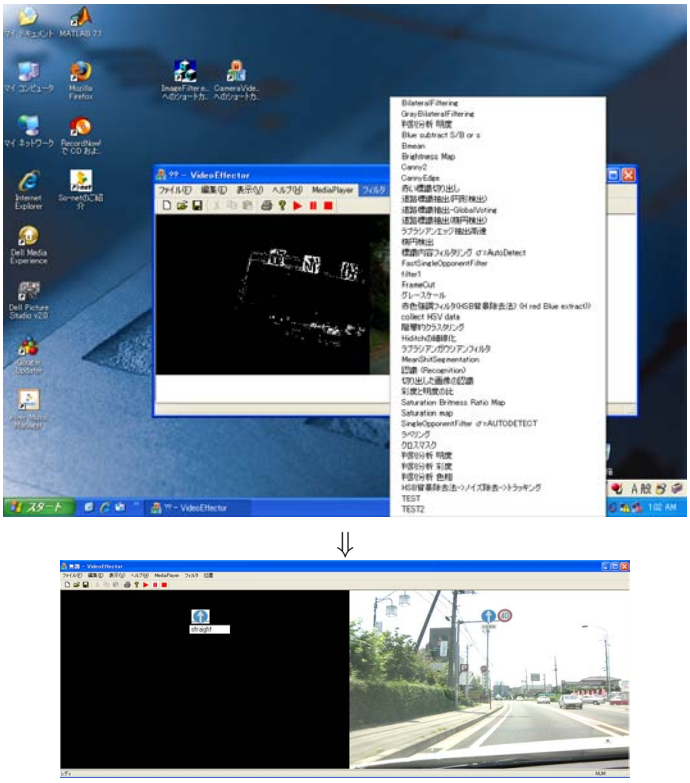


Figure 1.4: VideoEffector program and filters loaded from the DLL files.

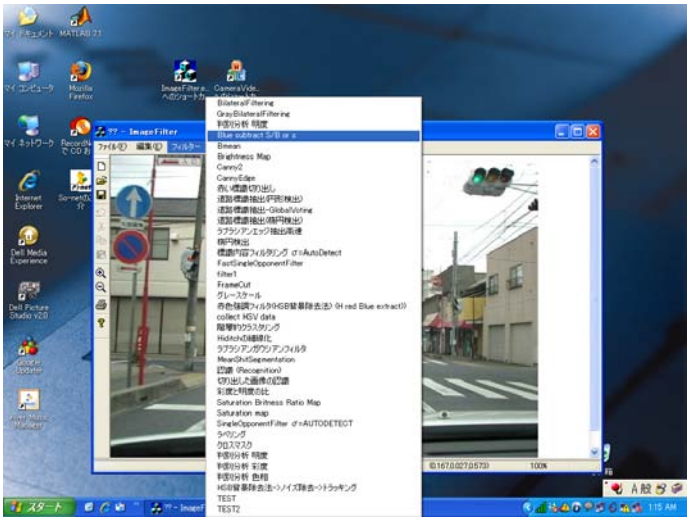


Figure 1.5: ImageFilter program and filters loaded from the DLL files.

- Chapter 6 discusses the classification stage. Two classifiers were used and compared. One is a decision tree classifier which uses geometrical features. The other one uses subspace method based on principal component analysis and histogram of oriented gradients are used as constituting features.
- Chapter 7 introduces a simple tracking algorithm.
- Chapter 8 presents results of the complete recognition process. Evaluation of the performance of the different algorithms used in this work is also discussed.
- Finally, results are summarized and further research plans are discussed in chapter 9.

Chapter 2

Related works

As mentioned in chapter 1, intelligent transportation systems are popular research field. Driver assistant systems are important aspect of ITS. One of the main objectives of these systems is to increase drivers awareness for their safety.

Some of these systems are available as commercial technologies in market while others are still under development. For example GPS navigation systems are not very expensive nowadays and many people use them. They contain many applications like guiding the drivers to a user defined destination, calculating arriving time and distance to the destination, giving information about nearer facilities such as fuel stations or restaurants. Such kinds of applications are depending on a pre-stored database. Some other technologies such as parking assistant systems [18], rear cameras or devices for blind spot elimination [19][20] are also becoming common technologies. Laser sensors are used in many similar applications and researches [21][22].

Many researches are based on the idea of using digital images to detect potential mistakes or dangers and notify about them. Examples include systems to detect damaged tires [23], systems to detect driver's fatigue by tracking his or her eyes [24][25] and systems for pedestrian detection [26].

In this scope, many researchers have been interested in designing a recognition system for traffic signs. The first work related to the road sign recognition was published in Japan in 1984 [27]. Later on, more researchers all over the world have become interested in this subject [16, 28–36].

The commercial traffic sign recognition systems are still not very popular. According to the article retrieved by the title "Traffic sign recognition" in *Wikipedia, the free*

encyclopedia [37], there are a number of vehicles using traffic sign recognition system. The list, which was last updated in August 2011, includes :

- Audi A8.
- BMW 7-Series.
- BMW 5 Series Gran Turismo.
- BMW 5-Series.
- 2011 Ford Focus (European).
- Mercedes-Benz E-Class.
- Mercedes-Benz S-Class.
- Opel/Vauxhall Insignia.
- Saab 9-5.
- Volkswagen Phaeton.

Most of the studies have focused on mandatory signs such as red annular signs [16, 28, 29, 33, 38]. In our laboratory, red annular signs shown in Figure 2.1 were processed with recognition rate over 99% [9].

Figure 2.1: Examples of mandatory red annular signs.

Some researchers, like Roberto Marmo and Luca Lombard, focused on special kind of traffic signs such as highway signs [32]. They dealt with only two patterns of arrows.

To avoid determination of the position of the traffic sign in each frame, two cameras may be used as in [39]. One camera is fixed and the other is moving automatically to keep traffic sign in the middle. But the mechanical movement consumes time. Moreover the car movement leads to vibration in the camera and changing the size of the targeted sign. These factors make it difficult to apply this method in real-time.

Most of the well-known recognition algorithms have been applied and tested to detect signs. For example, Haar wavelet, Bayesian and AdaBoost patches algorithms were proposed in [16]. Normalized cross correlation is one of the commonly used techniques in template matching as in [39][40]. Genetic algorithms and neural networks have been used in [41]. Radial basis functions neural network [42] and support vector machine were used for the shape classification and pattern recognition of segmented traffic signs from image sequences [43]. Principal component analysis (PCA) and SIFT features were also proposed [44].

As far as we know, there is no deep study which had focused on the blue traffic sign recognition in real-time. There is one similar study by Gao X.W. et al. [31]. They attained a blue sign recognition rate of 100% for still images with 5, 10 and 20% additive Gaussian noises, and 93% for 50% additive noise. In this research, they did not deal with the problem of motion blur which cannot be avoided in video images taken from moving vehicles.

In the following chapters, more detailed description of our algorithms and other methods used in each processing step will be presented.

Chapter 3

Outline of the recognition process

In the research described here, recognition of blue traffic signs is done according to the process diagrammatically shown in Figure 3.1. The overall processing scheme is similar to the one proposed by my colleagues on red signs [9].

The process consists of the following components:

- The system acquires a frame of image directly from the video camera connected to the computer or from the previously recorded videos.
- The RGB image is converted to another one in HSV color coordinate system. The original RGB image is also saved for further processing. The labeling algorithm is applied on the image in HSV coordinates to label pixels belong to blue objects which are sign candidates and to ignore other non-sign pixels.
- In the next step, the labeled pixels are traced to find the edges of blue objects. Blue objects are then segmented by cropping a rectangular area circumscribing the detected edges from the original RGB image.
- The segmented object is then tested by a set of criteria to decide if it is a sign candidate or not.
    1. If the blue object is verified as sign candidate, the system continues to the next steps which are arrow segmentation and classification.
    2. If the verification fails, the possibility that the sign is segmented as fragmented parts is considered. Parts are aggregated by a connecting algorithm. If the parts

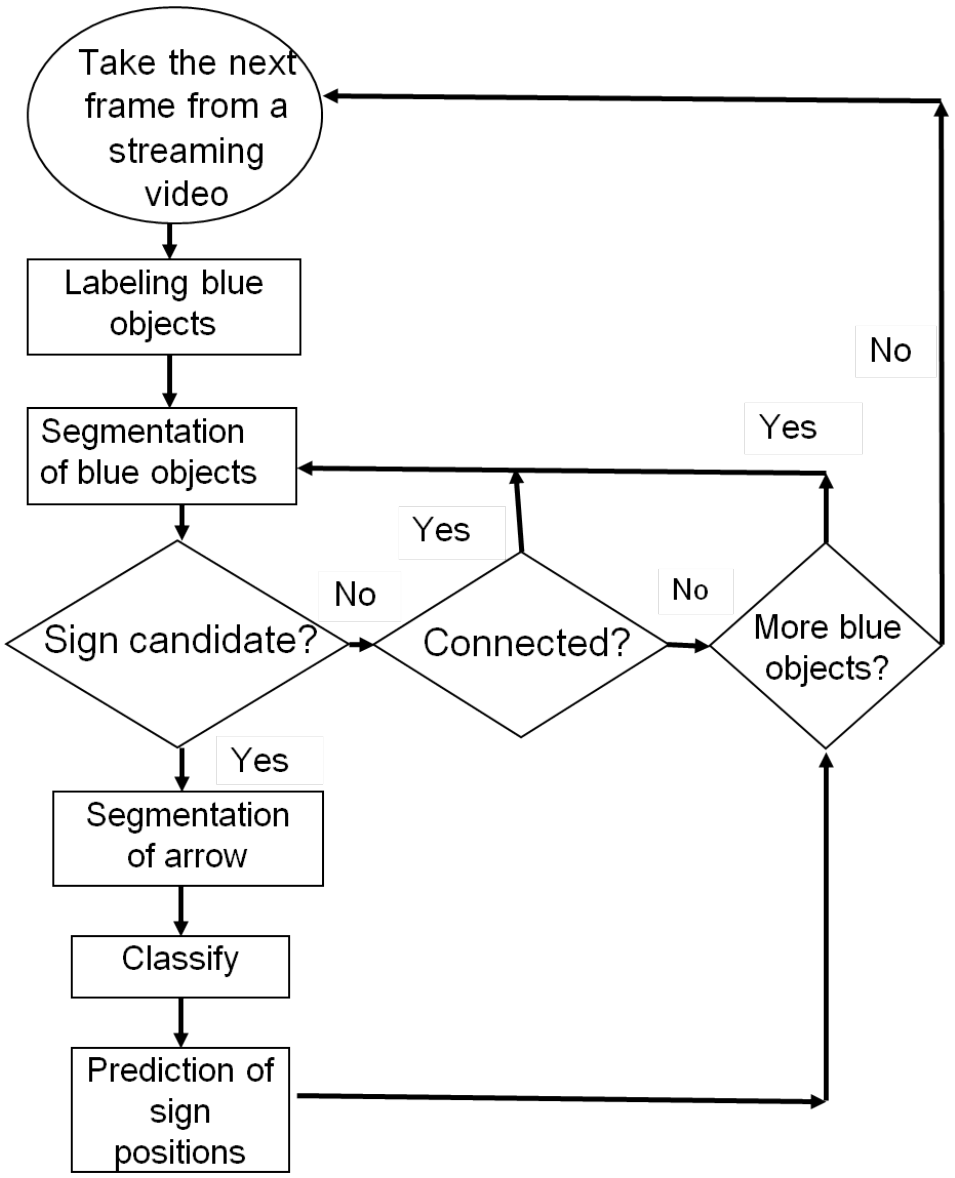


Figure 3.1: Recognition flow diagram of blue direction signs.

are connected correctly, the system segment them as one integrated sign and it is tested once again to decide if it is a sign candidate or not.

- After verifying that the segmented object is truly a sign candidate, the system detects the white object within the borders of the blue object as an arrow candidate. It is then classified to one of the target arrow patterns.
- The above mentioned steps are repeated for each labeled blue object in the frame acquired in the first step. When there is no more blue objects, the system acquires new frame and starts over from the first step.
- A simple tracking algorithm is applied to facilitate the detection of the same signs which will appear in consecutive video frames.

As mentioned in the first chapter, there are difficult problems to be solved in every processing stage. The difficulties and their solutions are explained in detail in the following chapters. They are processed into three main stages, namely, labeling, verification, and classification.

Chapter 4

Labeling and segmentation of blue objects

4.1 Introduction

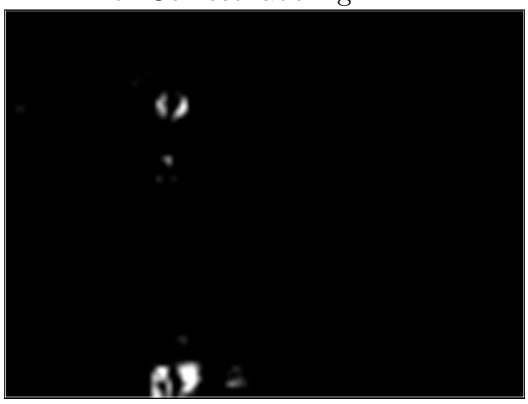
Pixel labeling is the first and one of the most important steps in the proposed system. The main objective of this stage is to distinguish between road traffic signs and any other objects or background in the scene environment. This is done by transforming frame images to binary images with 1 for pixels belonging to blue sign candidates and 0 for other non-sign pixels. Wrong labeling will affect the probability of correct recognition.

Examples of correct and incorrect labeling are shown in Figure 4.1. Figure 4.1-a is an original image, 4.1-b is an example of correct labeling and 4.1-c shows two examples of incorrect labeling.

One of the problems related to this stage is the variation in ambient light and its influence on the appearance of the road sign colors. Illumination of a sign changes during the day and according to the weather conditions such as clouds, fog and rain. Shadows caused by trees, buildings or other objects are another factor of variation in lighting. Such factors affect the saturation of the traffic sign color. Another important cause of the change in road sign color is the position of the light source which is the sun in day time. For example when the sun faces the camera and appears behind the road sign, the colors of the sign become darker. Finally, color fade by aging is also a factor that affects the correct labeling and segmentation of signs.

a: Original image.

b: Correct labeling.



c: Two examples of incorrect labeling.

Figure 4.1: Examples of correct and incorrect labeling.

For blue signs, there are two main blue classes in road scenes, namely, blue traffic signs and the sky in the background. The difficulty appears when we try to segment a blue sign or blue objects in general from the blue sky, when the sky is in the background of the sign.

4.2 Color segmentation methods

Many solutions were proposed to solve the problem of variation in ambient light. There is no perfect technique to solve this problem so far. Some researchers prefer to work on gray scale images to avoid this problem [45]. In this case the sign edges, corners or contours within the image should be detected to find signs in the scene according to their shape. For example, in [46] different gradient methods in gray scale have been used. In [47], an interesting comparative analysis among different methods is presented, but because the detection is implemented on the whole image using algorithms like Hough transform, the processing time is considered long which is inconvenient in real-time studies.

The easiest and most popular way is to find threshold experimentally that discriminate the pixels belong to signs from other background pixels. One benefit of this method is the speed of processing which is more suitable for real-time applications. Some pre-processing may also enhance this technique. For example in [48] gamma correction was used to reduce the effect of the color of the light source.

The authors in [49] studied the changes in the perception of road sign color during the time of the day. Their study showed that simple pairwise comparisons of the RGB components (i.e. Δ_{RG}, Δ_{GB}, and Δ_{RB}) are sufficient to segment road signs in real-time. In [50], authors used RGB color normalization and enhancement, polygon detection and color distance transform.

Researchers prefer to use color spaces which are less affected by illumination changes. For example XYZ, which is one of the first mathematically defined color spaces created by the International Commission on Illumination, is used in [51]. l*a*b color space, which is a color-opponent space, was used in [52]. Dimension l stands for lightness and a, b for the color-opponent dimensions.

HSV and HSI are famous color coordinate systems. HSV stands for hue, saturation, and value, and is also often called HSB where B stands for brightness. HSI stands for hue, saturation, and intensity which is the same as brightness. In [53] for example, the saturation

and hue were used for digital static images in HSI color coordinates. In [47], gray scale and HSV were used and compared.

Road sign recognition during night time is covered by few studies such as [16]. Night recognition is difficult because of the high noise in the acquired images which usually have low brightness.

4.2.1 Summary

Referring to the previous researches and other related efforts in this field, there are variety of methods and algorithms used for color segmentation. Some algorithms exclude color information and are implemented on the whole image in gray scale. They include shape detection algorithms like Hough transform and template matching or similar algorithms. The most common strategy is based on the use of color spaces which are less sensitive to changes in illumination. Some researchers prefer to use both color and shape detection algorithms to obtain more robust results.

Most of the algorithms are applied on red circular signs, and few studies have been dedicated to blue signs. The problem of sign-sky discrimination is not investigated as far as we know. Moreover, most of the researches have been applied on still images not videos.

In the research presented in this thesis, real-time constraint is an important issue. Therefore, methods including preprocessing or global transform on the whole image are not much preferable. Using HSV coordinates was found to be the best choice for two reasons:

1. Hue values do not change a lot when the ambient light changes or sign color is fading. Therefore this coordinate system was chosen for red signs [9]. To facilitate the integration of blue and red sign subsystems, HSV coordinate were also adopted in blue traffic sign recognition.

2. Transforming RGB images to HSV is not a time consuming process which makes it applicable in real-time systems. Details of HSV conversion are found in Appendix B.

In the next section, the effect of illumination on color parameters is discussed.

4.3 The effect of illumination on color parameters

Using HSV coordinate system, single threshold was enough for labeling the red signs [9]. The pixel is labeled as red if its $0.9 > Hue > 0.08$, $Saturation > 0.15$, and $Value > 0.1$.

Figure 4.2 shows the hue values for more than 25,000 pixels sampled from a variety of blue signs in different illumination and weather conditions. According to this figure, hue values for blue is in the range $0.5 > \mathrm{H} > 0.74$.

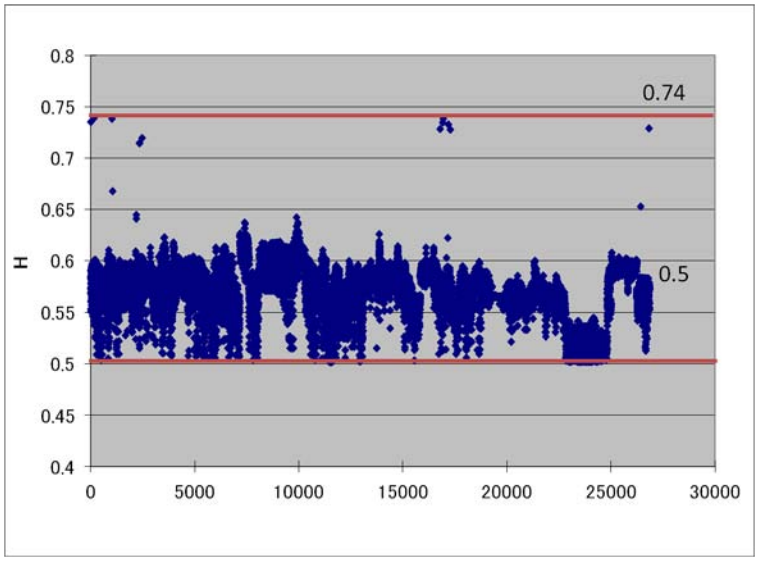


Figure 4.2: Hue values of pixels sampled from blue signs in different illumination and weather conditions.

As stated previously, hue values do not change a lot when the ambient light changes or sign color fades. However, saturation and value (brightness) do change. To study the effect of these factors, saturation map, value map, and saturation/value ratio map for frames containing signs and the sky were drawn using levels of 0.1 steps as shown in Figure 4.3.

The three maps were drawn in the blue space only $0.5 < H < 0.74$ and other colors were set to black. The maps were drawn in different illuminations and weather situations:

- In the morning as shown in Figure 4.4.
- At the evening as shown in Figure 4.5.
- While sun is facing the camera as shown in Figure 4.6.

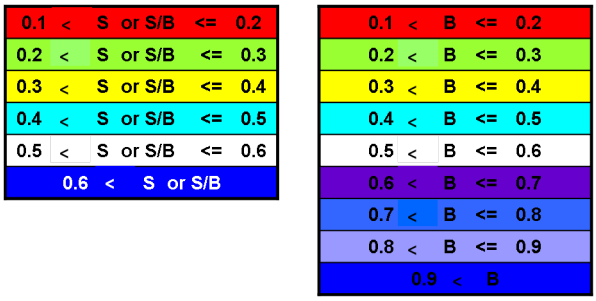


Figure 4.3: Color legends for saturation, value, and their ratio maps.

- While sun rays are reflected with high brightness on the objects in the scene as shown in Figure 4.7.
- In cloudy weather as shown in Figure 4.8.

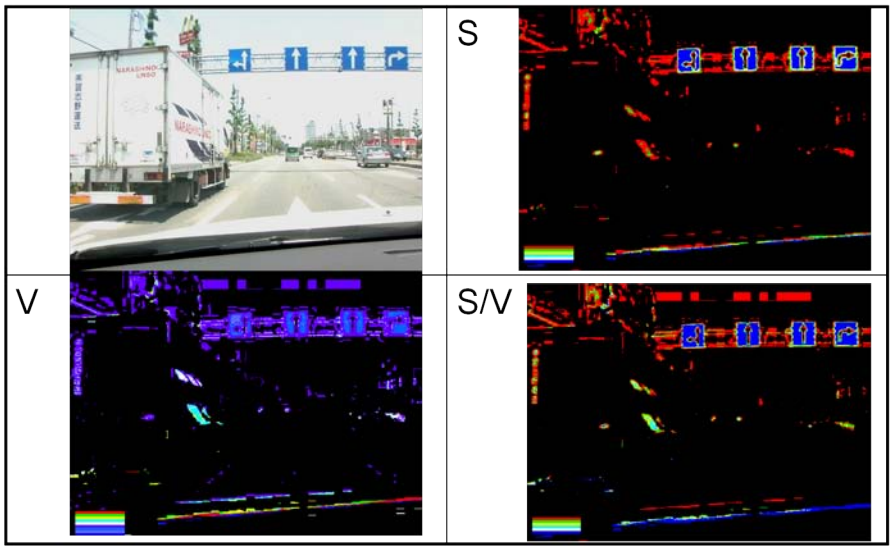


Figure 4.4: Scene taken in the morning: Maps of saturation, value, and their ratio.

From these maps, we can observe that in the morning, the saturations of sign and sky can separate them well, the brightness or value have some overlapping, but the ratio between saturation and brightness is good for discrimination. In the evening the values of ratio between saturation and brightness for the sign and the sky are close, but the saturation provides better separation results. In cloudy day the sky is not blue, the sign saturation

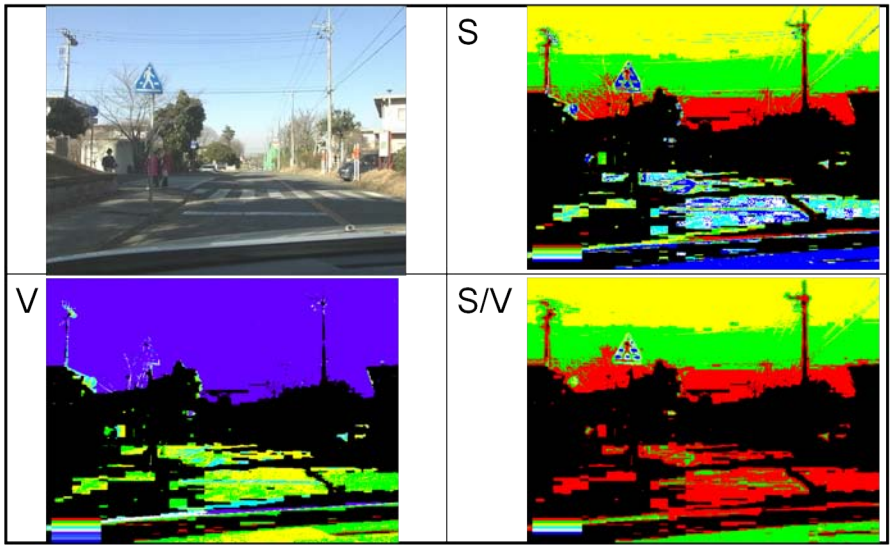


Figure 4.5: Scene taken at the evening: Maps of saturation, value, and their ratio.

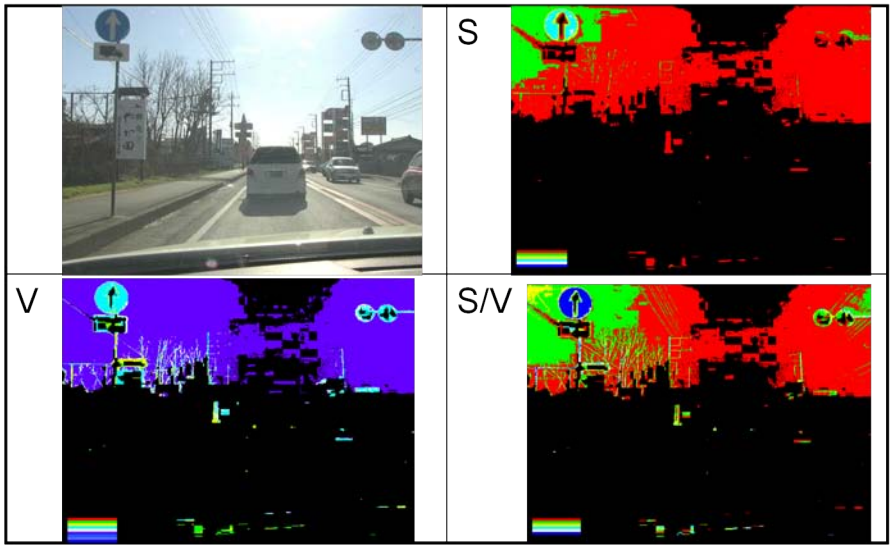


Figure 4.6: Scene taken while sun is facing the camera: Maps of saturation, value, and their ratio.

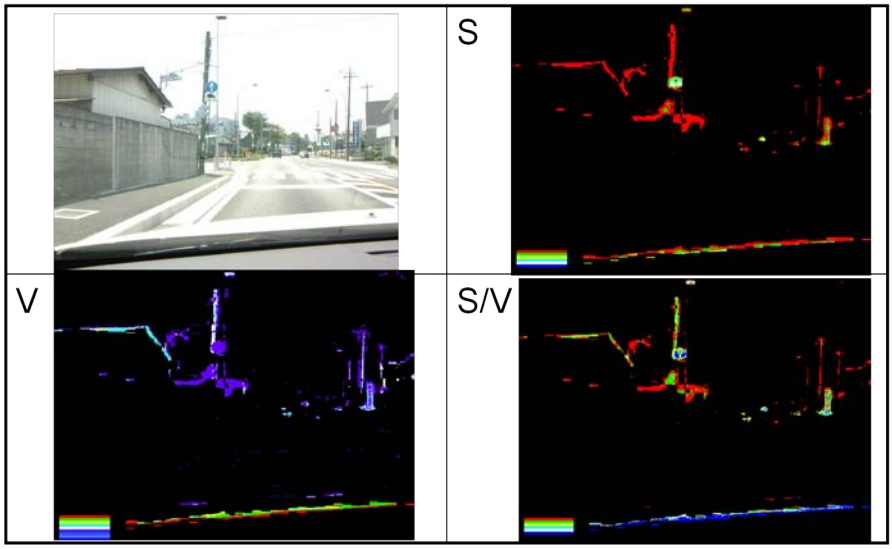


Figure 4.7: Sun rays reflected with high brightness on the objects in the scene: Maps of saturation, value, and their ratio.

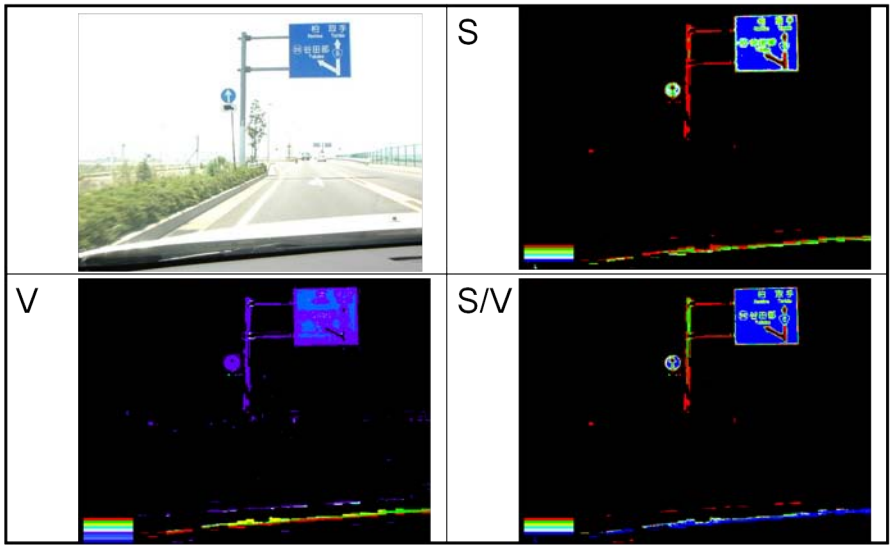


Figure 4.8: Scene taken in cloudy weather: Maps of saturation, value, and their ratio.

is low, but value is high. Then we need lower level threshold for discrimination. In the situation where sun is facing the camera the saturation is low, but the ratio gives good result for signs when low level threshold is used. For far signs the value is high. In the sunshine the saturation and value are high. The ratio also gives good results.

As a conclusion, it is difficult to choose a single threshold to segment blue signs. The main problem is that the distributions of HSV coordinates of the two classes are overlapped each other in many cases.

4.4 Sign – sky overlapping

Figure 4.9 shows an example of a scatter diagram of hues, saturations and values of a sign and the sky in HSV coordinates. Their distributions overlap with one another. As a result of this overlapping, it is difficult to choose a single threshold in HSV coordinates to label sign pixels correctly.

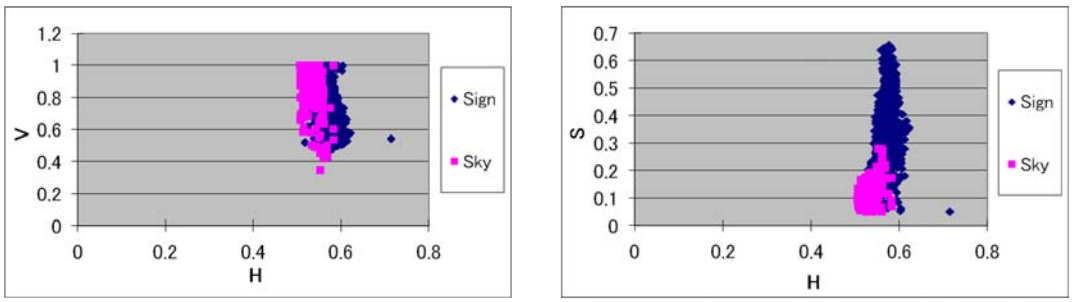


Figure 4.9: Scatter diagram of hues, saturations and values of a sign and the sky in HSV coordinates.

4.5 Sign – sky discrimination using multi-threshold

To enhance sign segmentation, a multi-threshold system is proposed. The idea is to extract blue signs from blue sky background using HSV parameters according to a threshold which adaptably changes with the brightness of the whole image.

Brightness in HSV coordinates is defined as $V = max\{R, G, B\}$. When the scene is well illuminated like the cases of morning or in sunny places, the number of pixels with high value of $max\{R, G, B\}$ will be big. This number will decrease when the sun is facing

the camera as pixels with high brightness represents the sun. It will decrease also in cloudy day as the sky will have high brightness but not other parts of the scene. Of course it will decrease more and more at the evening or at night.

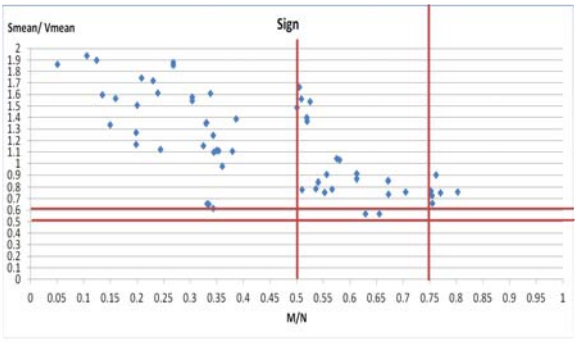
Let M be the number of pixels with $max\{R, G, B\} > 180$, which is a high brightness threshold, and N is the number of pixels in the whole image. The value M/N represents the normalized number of pixels with high brightness. To find different thresholds according to illumination changes, the relation between S and S/V with M/N was studied for 60 signs and 60 sample regions of the sky. These data were taken from images in different illumination and weather situations. Figure 4.10 shows the distribution mean values of S and S/V for sky and signs according to the values of M/N. Maximum variance of these means is about 0.05. Figure 4.10-a and b show that when the number of pixels with high brightness are more than 3/4 of the number of the pixels of the whole image, saturation/-value ratio for signs and sky regions is discriminated at $S/V = 0.6$. When $1/2 <$ M/N $<$ 3/4, saturation/value ratio for signs and sky regions is discriminated at $S/V = 0.5$. On the other hand, as shown in Figure 4.10-c and d, when the number of pixels with high brightness is less than half of the whole pixel number, only the saturation of the sign is sufficient to discriminate between the sky and the signs at the threshold $S = 0.4$.

From the above discussion, the multi-threshold solution can be described in the following algorithm. It improved the correct blue sign segmentation rate to 85% from 77% using single threshold.

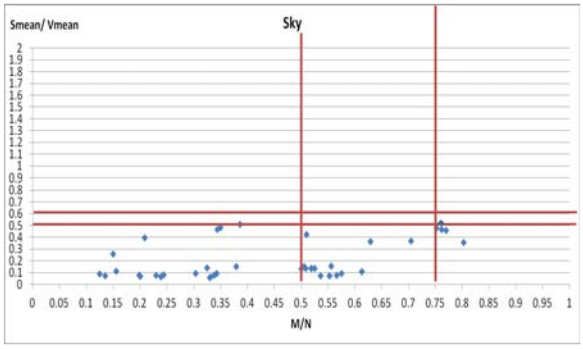
Algorithm 4.1: A multi-threshold algorithm to extract blue signs from blue sky background.

```
if M > 3N/4, thresholds are:
S/V > 0.6 & 0.5 > H > 0.74 (high brightness, morning)

if M > N/2, thresholds are:
S/V > 0.5 & 0.5 > H > 0.74 (sun in face, cloudy)

else thresholds are S > 0.4 & 0.5 > H > 0.6 (evening)
```

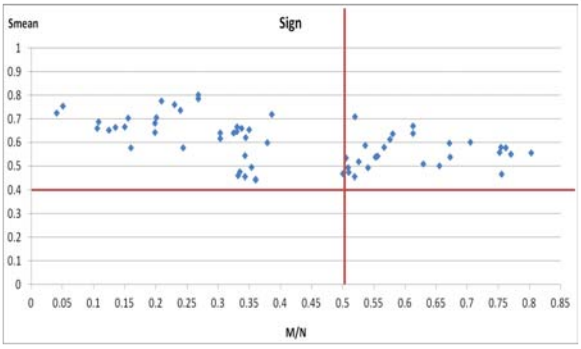
Where: M is the number of pixels with $max\{R, G, B\}$ (high brightness) in RGB coordinates, N is the number of pixels in the whole image.
*Thresholds are determined experimentally.



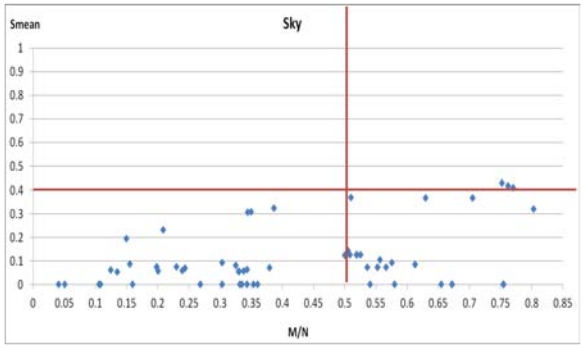
a: S/V mean values for 60 signs.



b: S/V mean values for 60 sample regions of the sky.



c: S mean values for 60 signs.



d: S mean values for 60 sample regions of the sky.

Figure 4.10: The distribution mean values of S/V and S for sky and signs according to the values of M/N.

One drawback of this method is its dependence on the camera and the experimental conditions. Although 85% correct segmentation was attained, it is not enough.

4.6 Sign – sky discrimination using opponent color filters

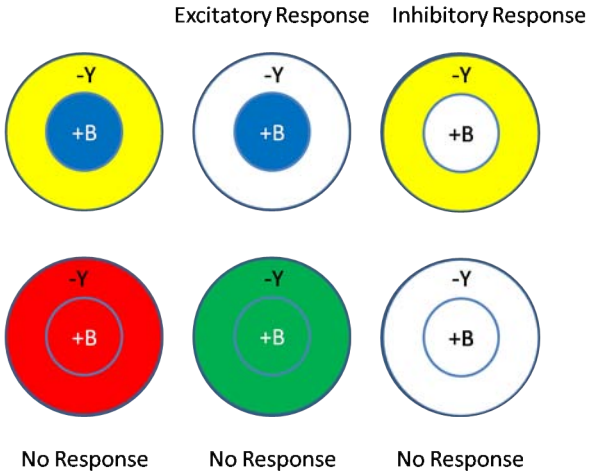
Since the results of multi-threshold system were not satisfactory, a new method, which tries to discriminate between color components of sky and signs by employing opponent color filters, is proposed [11].

In our retinas, there are three color sensitive cones (RGB) [9][54][55]. The output from the retina is coded according to color opponency: red-green (R-G), blue-yellow (B-Y) and black-white. A B-Y opponent color cell has a receptive field composed of excitatory and inhibitory concentric regions as shown in Figure 4.11-a. The central region receives excitatory inputs from B-cones and the annular region receives inhibitory inputs from R and G-cones, which produces yellow sensation. Therefore, it responds to a small blue light presented at the center of an excitatory region, but the response is suppressed by the simultaneous presentation of a yellow light to the surrounding inhibitory region. It does not respond to a large green lights because they contain blue and yellow components which induce excitation and inhibition simultaneously and their effects are canceled out. Red light is not effective because it does not contain any blue or yellow components.

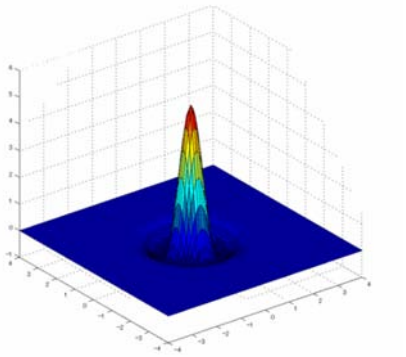
For black and white signals, since they have both blue and yellow components, the positive and negative outputs cancel each other and zero output is produced. The filter is modeled by a two-dimensional Laplacian Gaussian function and the output is calculated by convolution as illustrated in Figure 4.11-b. Since Gaussian filter is an optimal smoothing filter in the sense of maximum likelihood, noisy blue input is transformed to a smooth positive output, and noisy yellow signal is transformed to a smooth negative output. The two-dimensional Laplacian-Gaussian model is built using Equations (4.1), (4.2), (4.3) and (4.4).

$$\text{Gaussian filter: } G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.1}$$

$$\text{Laplacian Gaussian filter: } F(x,y) = -\nabla^2 G(x,y) = \frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.2}$$



a: The receptive field structure of a blue-yellow opponent color cell and its responses to color stimuli.



b: Filters are modeled by a 2D Laplacian Gaussian function.

Figure 4.11: The receptive field structure and the model of a blue-yellow opponent color filters.

$$F^{+}(x,y) = \begin{cases} F(x,y) \text{ if } F(x,y) > 0, \\ 0 \qquad\quad \text{otherwise} \end{cases}$$
$$F^{-}(x,y) = \begin{cases} 0 \quad \text{if} \quad F(x,y) > 0, \\ F(x,y) \quad \text{otherwise} \end{cases} \tag{4.3}$$

$$\begin{aligned} O(x,y) = & \int\int F^{+}(\xi,\eta)B(\xi - x, \eta - y)d\xi d\eta \\ & + \int\int F^{-}(\xi,\eta)(R(\xi - x, \eta - y) + G(\xi - x, \eta - y))d\xi d\eta \end{aligned} \tag{4.4}$$

where, the RGB components at an input location (x, y) are $R(x, y)$, $G(x, y)$ and $B(x, y)$ respectively. The parameter σ , which defines the spatial spread of filter, was adjusted to 0.5 pixels.

Because the system is intended to run in real-time, we need to reduce the computation time for the convolution. Since the signs will appear only in the upper half of the frame, the B-Y opponent color filters are applied to the upper half of the image. Moreover we confine the area of the computation to the blue area determined by HSV values which are in the range $0.5 < \mathrm{H} < 0.74$. The filters are applied on the original image in RGB coordinates. The overlap in HSV coordinates can be segregated by B-Y opponent color filters because it produces larger response to blue signs than to blue sky. The result of applying the B-Y opponent color filters are shown in Figure 4.12.

It is clear from the result that by applying the B-Y opponent color filters, blue signs produce large responses in comparison with blue background of the sky. This is because the filters are applied on the original RGB image. The sign is designed to reflect blue light and absorb other colors. The sky is the source of light at the day time. Although it looks blue to our eyes or via a camera image, actually it has R+G=Y components more than the sign. In other words, the blue component of the sign is more than that of the sky.

From areas with high response, a sign candidate is detected by a seeded region growing algorithm [56][57], which is based on local comparison of pixel properties without reference to a more global viewpoint. Regions (or pixels) are merged if they are homogeneous. One important step is to find the seed or the starting pixel to grow it.

a

b

c

Figure 4.12: The result of applying blue-yellow type opponent color filters on an image containing blue sign and the sky. a: Original image. b: After applying the filters on the whole image. c: After applying the filters only on the blue area in the upper half.

4.6.1 The algorithm

Some results obtained by applying the opponent color filters and the seeded region growing algorithm are shown in Figure 4.14. The algorithm can be formulated as follows:

1. Apply the B-Y opponent color filters on the blue area of the image and calculate the gradient of the filter response by Sobel filter.

2. Remove high-level edge pixels (gradient values > 250) because they present boundaries between blue and non-blue areas. Examples of the result of this step are shown in Figure 4.13.

3. Scan the B-Y opponent color filter output image to find a not-yet visited pixel with the maximum response (intensity) > 200.

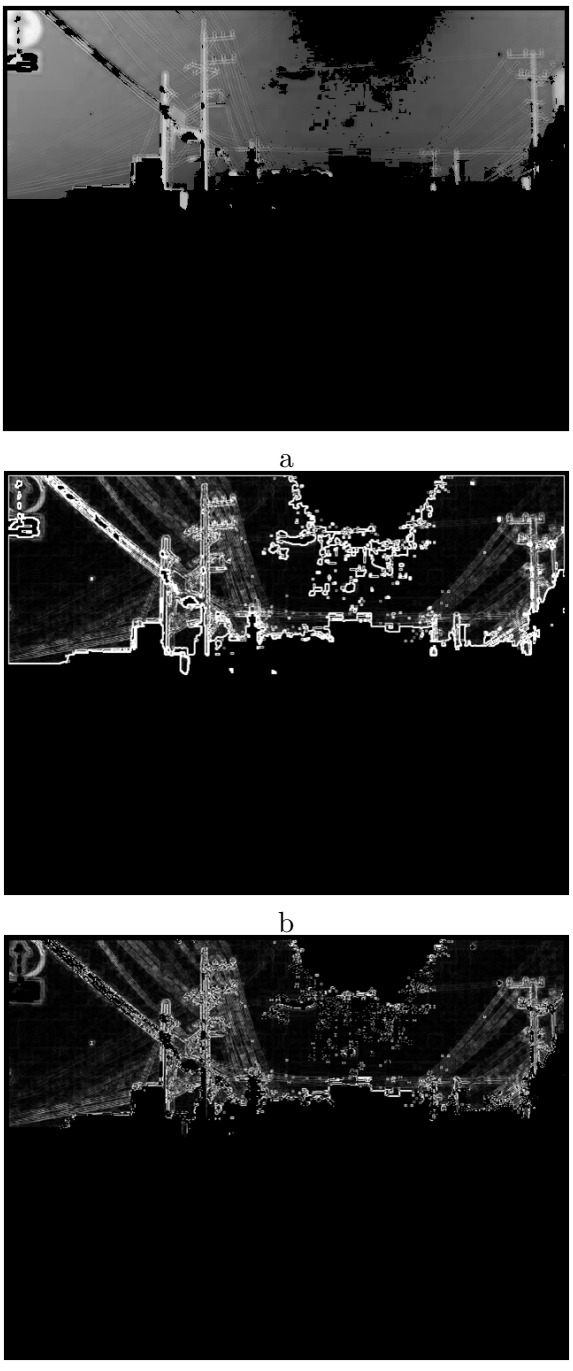
4. If the pixel and its neighbor are of high response and make a cluster of pixels, consider it as growing seed and proceed to step 5. If the pixel is an isolated one, delete the pixel and go back to step 3.

5. Calculate the mean value μ and standard deviation σ of a 3 x 3 neighborhood centered by the current pixel and include the neighbor pixel into the region if it satisfies the following conditions [56]:

   - If the gradient of the pixel (sobel response) is less than a preselected threshold (the pixel does not belong to edges) AND the difference in opponent color filter response of the 2 pixels is less than or equal to the preselected threshold.
   - If the gradient of the pixel is more than or equal to the preselected threshold (the pixel belongs to an inner edge or a noisy edge) AND the opponent color filter response of the pixel is not more than or equal to one standard deviation away from the region mean.

6. Move to the newly included pixel and repeat steps (5) to (6) until all pixels have been considered to be grown or the pixel cannot be grown anymore.

7. If not all the blue area is scanned, go back to step 3.



a

b

c

Figure 4.13: The result of applying Sobel filter on an image after applying the blue-yellow opponent color filters. a: Original opponent color filter result. b: Sobel filter output without removing high level edge pixels. c: The same image after removing high level edge pixels.

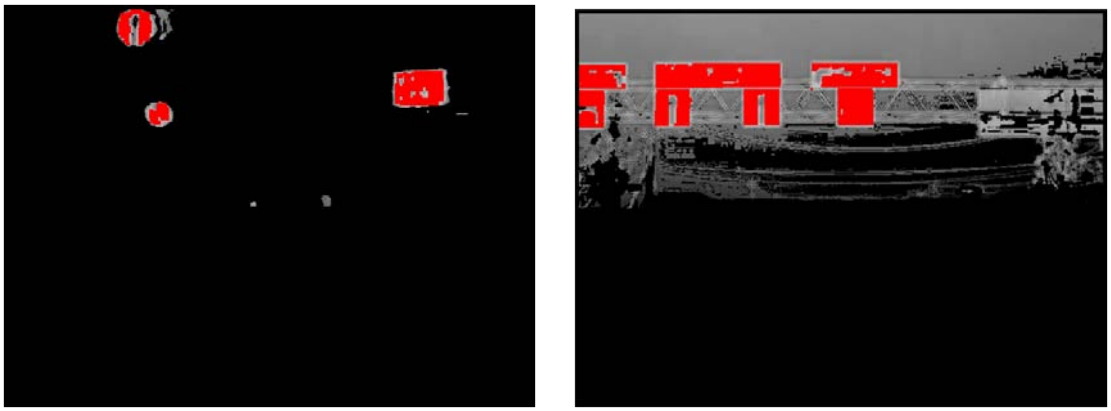


Figure 4.14: Results obtained by applying blue-yellow type opponent color filters and seeded region growing algorithm.

4.7 Segmentation of labeled objects

After changing each frame to a binary image, a noise reduction algorithm is applied. Next step is to trace the edges of each labeled object and segment it. For tracing edges, neighborhood tracking algorithm is used. First, the binary image is scanned until the first labeled pixel on the scanning line is found. Finding one of the object's edge pixels, we trace the neighborhood pixel values to find pixels constituting the edge components in the binary image. After saving all the edge's coordinates, the object is segmented. Segmentation is performed by finding a rectangular area circumscribing all the detected edges of the labeled object and cropping this area from the original image as shown in Figure 4.15. The segmented object is then tested by a set of criteria to decide if it is a sign candidate or not.

4.8 Summary and discussion

In this chapter, the first stage of the blue traffic sign recognition process was explained. Labeling is achieved by using HSV coordinates which are found to be best because hue values do not change a lot according to the illumination or fading factors. Moreover, transforming RGB images to HSV is more appropriate for real-time systems.

Two solutions for the problem of sign-sky discrimination were proposed. The first one is using a multi-threshold system and opponent color filters with seeded growing

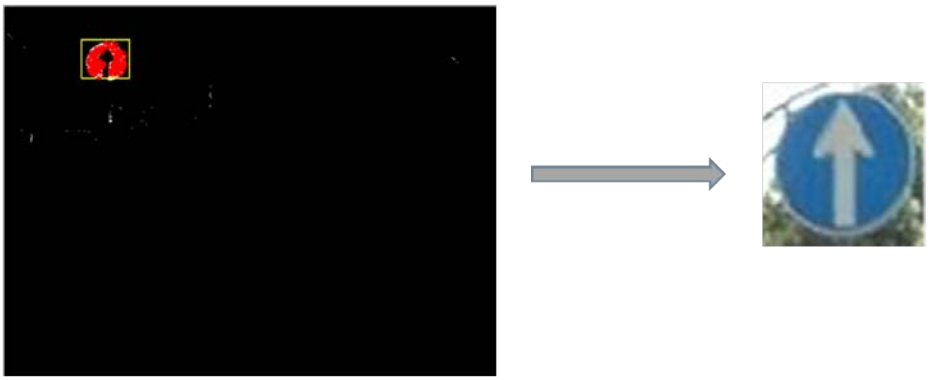


Figure 4.15: Segmentation of a blue object by cropping a rectangular area circumscribing the detected edges.

algorithm. The experiments showed that applying the multi-threshold method manged to improve the segmentation rate to 85% from 77% using single threshold. Using the opponent color filters and seeded growing algorithm improved the segmentation rate to 91%.

Chapter 5

Sign candidate verification

5.1 Introduction

After segmenting blue objects, it is necessary to obtain road sign candidates. The candidates then undergo the next classification stage. The characterization of road signs can be carried out using three different features: Shape, color and the inner symbol.

5.2 Candidate verification methods

Most researchers segment signs according to their shapes, such as red circular objects for red signs [28][29] or rectangular area for highway signs [32]. For red signs mainly shape and color of the road signs are the most relevant features. Some researchers ignore the color information and focus on shape detection only in gray scale images. Shape detection has several drawbacks. For example it necessitates a robust edge detection and/or matching algorithm to detect the relevant shapes. This is particularly difficult when the road sign appears relatively small in the image. Moreover, even if a shape of interest is identified, it can be confused with several other shapes such as commercial signs and building windows. One informative paper is [47] in which four algorithms were compared for a complete recognition system. It presents an interesting comparative analysis between the different methods for recovering circular contours. The detection process is performed on the whole image using algorithms like Hough transform. Therefore, the processing time for these algorithms is between 6 to 21 seconds which is unsuitable for real-time applications.

In blue traffic signs, the same arrow patterns are used in rectangular and circular signs to indicate the same meaning. Moreover, the rectangular blue objects are not dedicated for blue signs as shown in Figure 5.1.

Applying opponent color filters and seeded growing algorithm (Section 4.6) improves the labeling as discussed in the previous chapter. However, the algorithm still has some drawbacks. One of them is shown in Figure 5.2. Some rectangular objects give high response to the B-Y opponent color filters even though they are not blue.

From the above discussion, we conclude that shapes of signs cannot provide unique information for the determination of directional sign candidates. Therefore, shape information can be neglected to reduce the processing time. On the other hand, in some cases skipping the shape information completely may lead to miss important information, especially when the sign is fragmented as it will be explained in Section 5.4.

Figure 5.1: Example of blue rectangular objects which are not signs.

Sign candidate verification is affected by the problem of fragmentary segmentation of signs due to motion blur and lighting conditions. Only one method was discussed in related works. This method is the morphology processing like dilation or closing morphology algorithms [58][59]. The researchers have applied the morphology process over the whole frame which is time consuming. Related topic to this process might be the occlusions of road signs due to occluding objects. To avoid this problem deformable models have been applied in [60]. The system proposed in this study uses a large number of different features to detect the road sign. Therefore, the problem was considered as if some of the features are lost

a b

Figure 5.2: Example of an object which gives high response after applying opponent color filters although it is not a blue object. a: The original image. b: The output of opponent color filters.

during the detection process. In [61], the authors proposed an algorithm to detect the outer ellipses of red signs by combining the left and right fragments of the ellipse. The algorithm search for ellipse fragments in a way similar to genetic algorithms. Average detection time was 3.73 seconds. Again, such algorithms are not suitable for real-time applications.

5.3 Criteria for the verification

Without shape information, the system relies on the following criteria to check if the blue object is a sign candidate or not [10]:

1. The number of edge pixels must be adequate for signs. Object is taken in consideration if the number of edge pixels is between 50 and 800. This means to ignore very big or very small objects.

2. After tracing the edges of a blue object, we scan the segmented area to check if there is a white object adjacent to its borders. This process is done by the following steps:

   - Apply a circular mask as shown in Figure 5.3-a to b to segment the inner parts of the signs. This mask is applied on circular and rectangular signs because we are concerned with the inner part not the borders.

- After adjusting the position of the circular mask, colors other than blue or white are removed.
- Change the candidate sign area from RGB to gray scale using the following equation:

$$wb = 0.3R + 0.59G + 0.11B \tag{5.1}$$

- Obtain a gray level histogram as shown in Figure 5.3-c.
- Calculate the best gray level threshold, which separates white area from blue background, by maximizing the ratio between the between-class variance and the within-class variance using Equation (5.2) [62][63].

$$f(t) = \frac{Between - Class\,variance}{Within - Class\,variance} = \frac{P(C_1)(\eta_1 - \bar{\eta})^2 + P(C_2)(\eta_2 - \bar{\eta})^2}{P(C_1)(\sigma_1)^2 + P(C_2)(\sigma_2)^2} \tag{5.2}$$

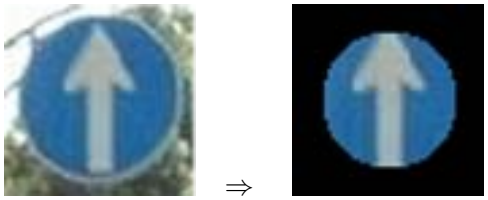
  where$P(C_i), i = 1, 2$, is the class probability, η_i is the class mean, $\bar{\eta}$ is the overall mean and σ_i^2 is the within-class variance.
- If the above threshold T was found, we use it to change the image to a binary image: 1 for white objects and 0 for other pixels.
- If there is one or more white objects inside the borders of the blue object, next step is to trace the edges of each white object and by applying the neighborhood tracking algorithm similar to the one used for blue objects which is described in section 4.7. If one or more white objects are found inside the borders of the blue object the following third criterion is checked.

3. The white object must be near the center of the segmented blue area.

   If there is a white object near the middle of the blue segmented object, the blue object is considered as a sign candidate. This criterion is tested by checking the distance between the detected inner white object's center and the sign's center. Figure 5.4 shows how to check this criterion. The red lines intersect at the center of the sign, the green lines intersect at the centers of the rectangles circumscribing the inner white objects and the yellow arrows show the distance between the detected inner white object's center and the sign's center.

Sometimes even after applying the circular mask, noisy objects cannot be removed completely. Figure 5.4-a shows examples of noisy white objects. In the upper right corner, there is an example of an object resulted from the effect of the high brightness of the sky. In the upper left corner appears a white noisy dot. These objects are supposed to be ignored because the distance between their centers and the center of the sign is relatively big according to an experimentally determined threshold. However, in Figure 5.4-b the white object (the arrow) in the green inner rectangle is the first object to be segmented because the distance between its center and the center of the sign is small. In Figure 5.4-c, an example of the white object segmented from the image in Figure 5.4-b is shown. Figure 5.4-b is obtained without applying the circular mask, just to show the effect of the mask, but in real cases the mask is applied to all segmented blue objects.

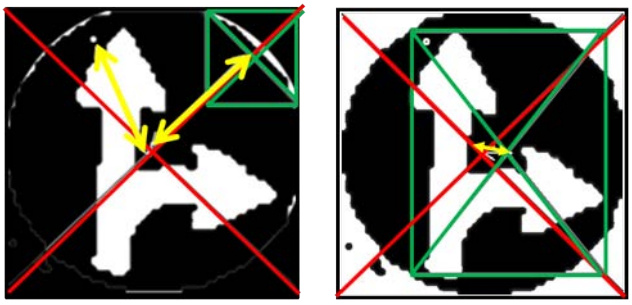
If more than one object exists near the sign center, they are segmented and checked from the nearest white object one by one whether it is an arrow pattern or not. If one of them is found to be a target arrow pattern by the classifier, the checking process terminates at that time and the sign is recognized.



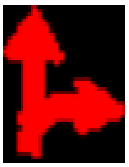
a: Original segmented image. b: Masked area.

c: Gray level histogram.

Figure 5.3: Gray level histogram of the blue candidate sign after applying a circular mask.



a: White objects which are not in the middle. b: Middle white object.



c: Segmenting the middle white object.

Figure 5.4: Checking if the white object is near the center of the blue object.

5.4 Problem of sign fragmentation

One main problem to be solved is the fragmentary segmentation of sign parts. Since we are dealing with moving images, due to motion blur or some lighting conditions, the white area can spread into the blue area and the blue background of the sign can be disconnected after applying the labeling stage as shown in Figure 5.5. If the blue object fails to satisfy the criteria in section 5.3, the possibility of fragmentation of sign is considered. Some solutions to aggregate the broken backgrounds are discussed in the following sections.

5.5 Structured dilation to aggregate fragmented signs

The first solution to aggregate the fragmented parts of the blue objects is by using a dilation morphology process according to the structuring elements shown in Figure 5.6 [62][64].

When a blue object does not match the criteria as a sign candidate, the structuring elements are applied in turn on the edges of the segmented blue parts. An example of the result of dilation process is shown in Figure 5.7. The idea of using this dilation process

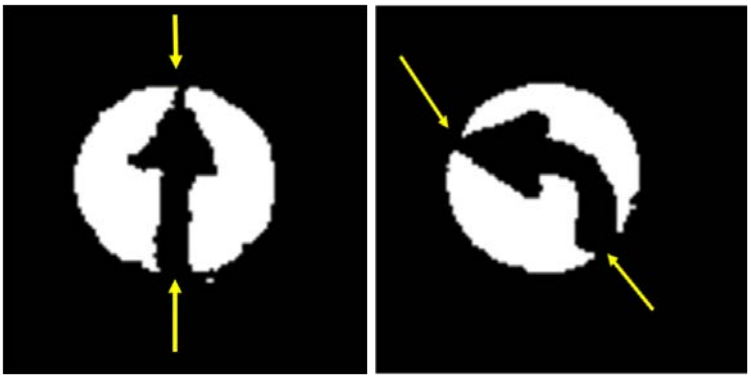


Figure 5.5: Examples of signs fragmented into two parts.

instead of applying a simple dilation over the whole image from the beginning [34] is to reduce the processing time. By using the structuring elements, we are able to fill a gap which size is up to 3 pixels. If the dilation process is successful, two parts may be connected and complete sign image can be segmented.

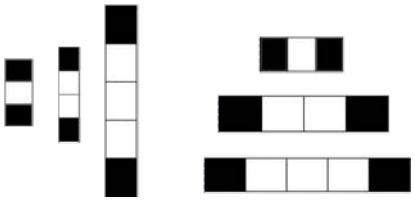


Figure 5.6: Structure elements for dilation.

5.6 Two-step algorithm to aggregate fragmented signs

The dilation algorithm is fast but it is very limited regarding the distance between the fragmented parts. To overcome this problem a new algorithm is proposed. Aggregation is performed by expanding one of the segmented rectangular areas to include another neighboring area. This expansion is carried out in two steps.

Figure 5.8 shows the flow chart of these two steps. If we have a fragmented sign, aggregation is tried by step 1. If step 1 is sufficient for aggregation, the system continues to segment the white objects near the middle of the aggregated sign and they are fed to the classifier. However, if step 1 fails, step 2 is tried.

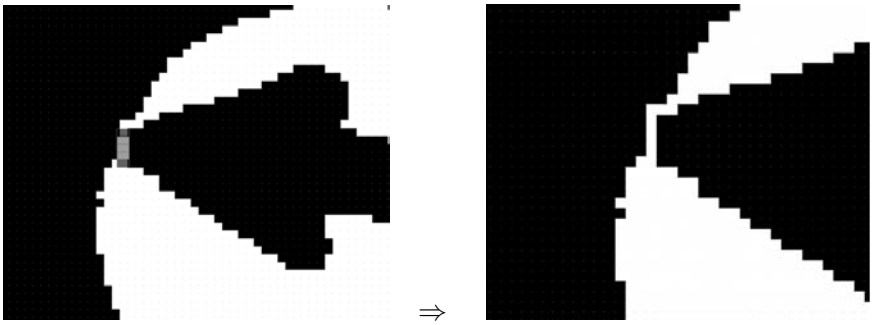


Figure 5.7: Applying the structure elements.

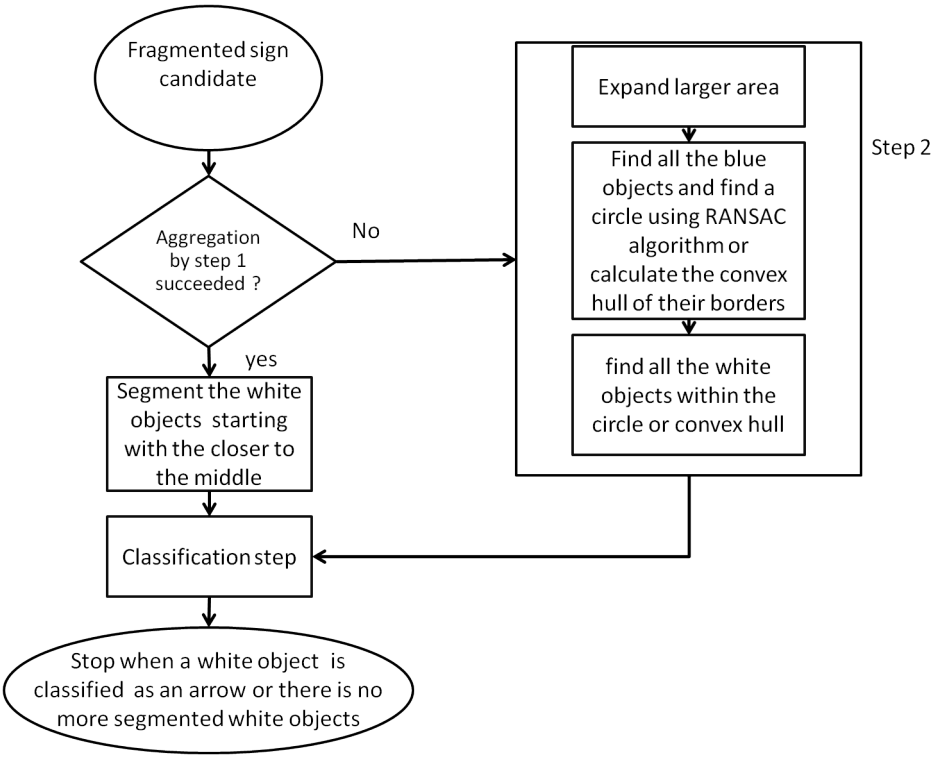


Figure 5.8: The two-step aggregating algorithm to unite fragmented parts.

Since the system is supposed to be used in real-time, processing time is an important constraint. Processing time of step 1 is between 30-60 ms, while the complete process of step 2 will take up to 250 ms. Experiments showed that step 1 is enough in most of the cases. If real-timeness is critically necessary, step 2 can be discarded at the expense of small reduction in recognition rate.

5.6.1 Step 1

If the segmented blue object fails to be verified as a sign candidate, the system scans for blue objects which lie close to the one considered as fragmented part. Trying to minimize the processing time of aggregation, the algorithm guesses the expanding direction according to the ratio between the height (H) and width (W) of the rectangular area surrounding the segmented blue object. If $H \approx W$, the expanding direction is determined by the vector drawn from the center of the blue object to the center of the nearest white object. Examples are shown by the small green arrows in Figures 5.9-1 to 8. The centers of the blue parts are determined by the intersection of the red lines and the centers of the white objects are determined by the intersection of the yellow lines.

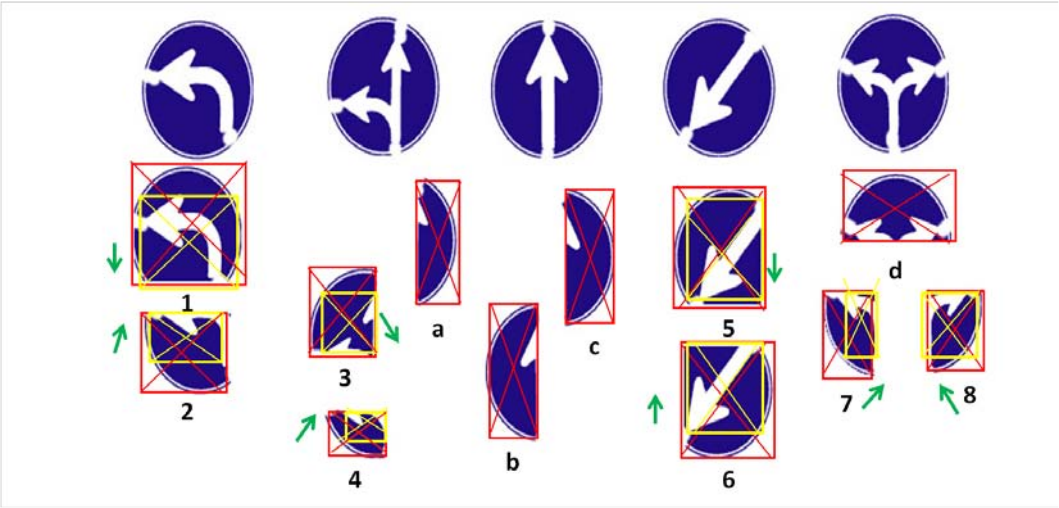


Figure 5.9: Examples of expanding directions which are indicated by the small green arrow directions.

Suppose ϵ is a small tolerance number. Expanding direction is determined as follows:

1. If $H/W > 1 + \epsilon$, as in Figure 5.9-a, b and c, then the segmented area is expanded to the right. If no blue object is found in the new expanded area, the expansion is directed to the left.

2. If $H/W < 1 - \epsilon$, as in Figure 5.9-d, then the segmented area is expanded downward. If no blue object is found in the expanded area, expansion is directed upward.

3. If $H \approx W$ and (a) the blue object has a white object in peripheral area, or (b) the white object was rejected as an arrow candidate, it is expected that the sign is fragmented like the example marked by a red circle in Figure 5.10. Therefore, the search for another fragmented part of the sign is performed in the direction of the vector drawn from the center of the blue object to the center of the white object as shown by the small green arrows in Figure 5.9-1 to 8.

If another blue object is found in the search area, then its edges are traced and convex hull [65][66] of the edges of the two objects is calculated to integrate the two objects as one object. The result is shown Figure 5.11. Convex hull is explained in Appendix D.

The next step is to detect white objects to see if the aggregated blue object forms a sign candidate according to the algorithm and criteria described in Section 5.3. After aggregation by step 1, if the system fails to detect the arrows, several reasons can be considered:

1. The two parts of the sign are far from each others as shown in Figure 5.12-a.

2. The borders of the signs are connected with noisy part of the scene as shown in Figure 5.12-b and c. The sign in this case may be refused by criterion of the adequate number of edge pixels.

3. Different objects are segmented as one object, which leads to incorrect adjustment of the circular mask parameters as shown in Figure 5.13.

These reasons will affect the correct segmentation of white objects (arrows) and consequently the overall recognition rate. If step 1 fails, step 2 is tried.

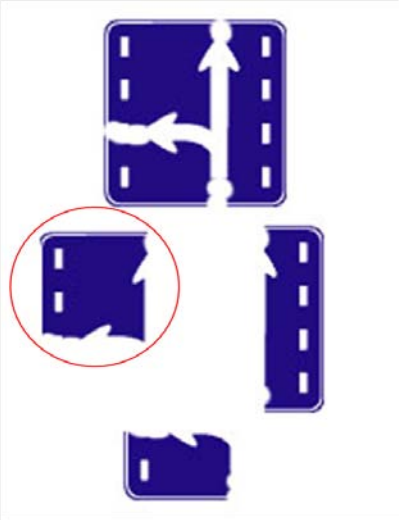


Figure 5.10: Example of a sign segmented into three parts.

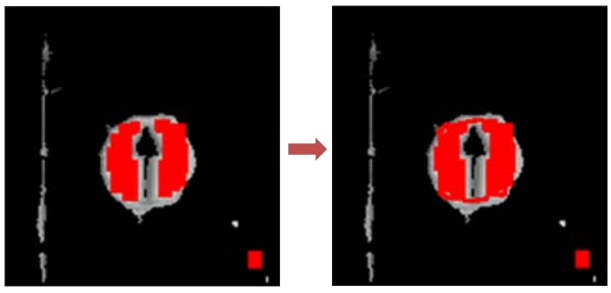


Figure 5.11: Applying convex hull to connect two segmented parts.

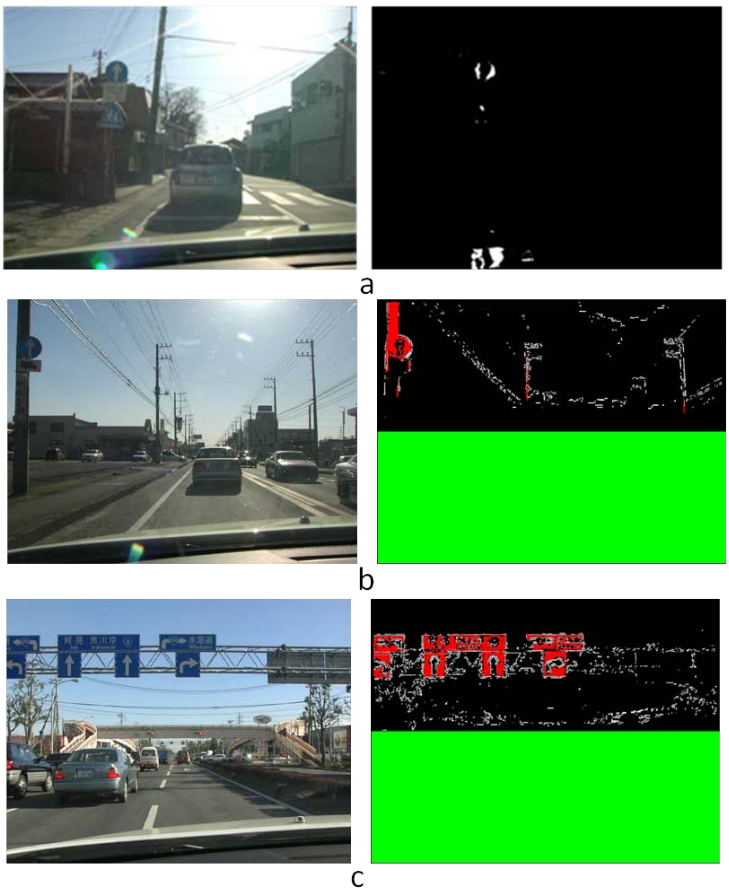


Figure 5.12: Examples of cases in which the system fails to detect the arrows after the aggregation by step 1. Original images are on the left and the outputs of the labeling stage are on the right. a: The two parts of the sign are far from each others. b and c: The borders of the signs are connected with noisy part of the scene.

5.6.2 Step 2

The key point of step 2 is to use the arrow shape information to help in deciding if the blue objects are belonging to a fragmented sign or not. Step 2 consists of the following processing steps:

1. The rectangular area is expanded to H = 4H, W = 4W centered at the center of the original rectangle to produce a new bigger region of interest (ROI).

2. The new ROI is scanned and the borders of all labeled blue objects are traced.

3. RANSAC (Random Sample Consensus) [67] is used to check if there is a circular sign in the new ROI. If a circular blue object is detected, we segment it and check if there is a white object inside it. For the case shown in Figure 5.13, true sign border can be detected by this step. If no circular blue object is detected, The convex hull that includes all the edges in the expanded area is calculated.

   White objects can be detected as described in Section 5.3 but without applying the circular mask in this case. Colors other than blue or white are removed and the candidate area is transformed from RGB to a gray scale image. The same threshold obtained in the process of criterion (2) of Section 5.3 is used to make binary images.

4. All the detected white objects are fed to the classification stage. If one of them is found to be a target arrow pattern, the process terminates and the sign is classified.

5.7 Summary and discussion

In this chapter, the second part of the detection process is discussed. Verification of blue objects as sign candidates or not is performed according to the following criteria: If the number of the blue object edges is adequate, and there is a white object adjacent to its borders and close to its center, then the blue object is considered as sign candidate. Since shape information cannot provide unique information for the determination of directional sign candidates, it can be neglected at the verification stage. When the sign is fragmented, an aggregating algorithm is needed and therefore, two methods were proposed. First method is to use structuring dilation. The second method consists of two steps. If step 1 is successful and the aggregation is completed, the system continues to white object segmentation and

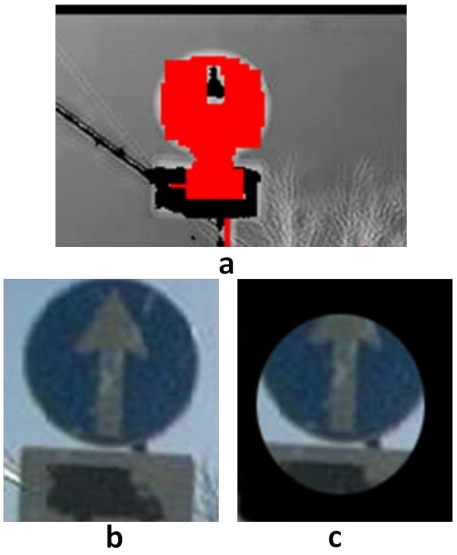


Figure 5.13: Example of different objects segmented as one object which leads to incorrect adjustment of the circular mask parameters. a: The output of labeling stage. b: The result of segmentation. d: Applying the circular mask.

classification. When step 1 fails, step 2 which exploits the shape information of arrows and signs is used.

The results of applying labeling, segmentation and verification algorithms are compared in tables 5.1, 5.2, 5.3 and 5.4.

Table 5.1 shows the results of segmentation using multi-threshold labeling algorithm and the structuring dilation for aggregation of fragmented signs. From 94 signs 71 were segmented correctly, all the 64 non-sign objects were segmented correctly and the total segmentation rate is 85%. Average processing time is about 40 ms.

Table 5.2 shows the results of segmentation using multi-threshold labeling algorithm and only step 1 for aggregation of fragmented signs. From 94 signs, 80 were segmented correctly, all the 64 non-sign objects were segmented correctly and the total segmentation rate is 91%. Average processing time is about 40 ms.

Table 5.3 shows the results of segmentation using opponent color filters and seeded growing algorithm for labeling and only step 1 for aggregation of fragmented signs. All blue objects which might be considered as sign candidates were labeled correctly. Only 4 straight direction signs were not segmented correctly because they were fragmented and the

Table 5.1: Results of segmentation using multi-threshold labeling algorithm and the structuring dilation for aggregation of fragmented signs.

Sign type	The total number of signs	The number of correctly segmented signs
Straight	65	52
Left	7	3
Right	4	3
Straight or left	9	6
Straight or right	9	7
Not signs	64	64
Total	158	135 (85%)

Table 5.2: Results of segmentation using multi-threshold labeling algorithm and only step 1 for aggregation of fragmented signs.

Sign type	The total number of signs	The number of correctly segmented signs
Straight	65	58
Left	7	5
Right	4	4
Straight or left	9	6
Straight or right	9	7
Not signs	64	64
Total	158	144 (91%)

aggregation algorithm failed in unifying them. All the 64 non-sign objects were segmented correctly and the total segmentation rate is 97%. Average processing time is about 60 ms.

Table 5.4 shows the results of segmentation using opponent color filters and seeded growing algorithm for labeling and the two-step algorithm for aggregation of fragmented signs. From 94 signs, 93 were segmented correctly, all the 64 non-sign objects were segmented correctly and the total segmentation rate is 99%. Average processing time is about 90 ms and maximum processing time is 250 ms.

Table 5.3: Results of segmentation using opponent color filters and seeded growing algorithm for labeling and only step 1 for aggregation of fragmented signs.

Sign type	The total number of signs	The number of correctly segmented signs
Straight	65	61
Left	7	7
Right	4	4
Straight or left	9	9
Straight or right	9	9
Not signs	64	64
Total	158	154 (97%)

Table 5.4: Results of segmentation using opponent color filters and seeded growing algorithm for labeling and the two-step algorithm for aggregation of fragmented signs.

Sign type	The total number of signs	The number of correctly segmented signs
Straight	65	64
Left	7	7
Right	4	4
Straight or left	9	9
Straight or right	9	9
Not signs	64	64
Total	158	157 (99%)

Chapter 6

Classifying the arrow patterns

6.1 Introduction

An output example of applying the labeling and the verification stages is shown in Figure 6.1.

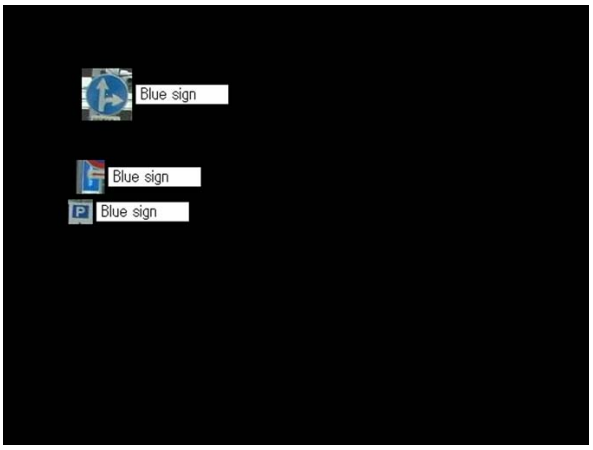


Figure 6.1: Result of blue object detection.

The target signs in this research consist of 13 patterns of blue signs. These 13 signs are reduced to 8 arrow patterns as shown in Figure 6.2. The purpose of this stage is to classify the white objects by matching these 8 arrow patterns.

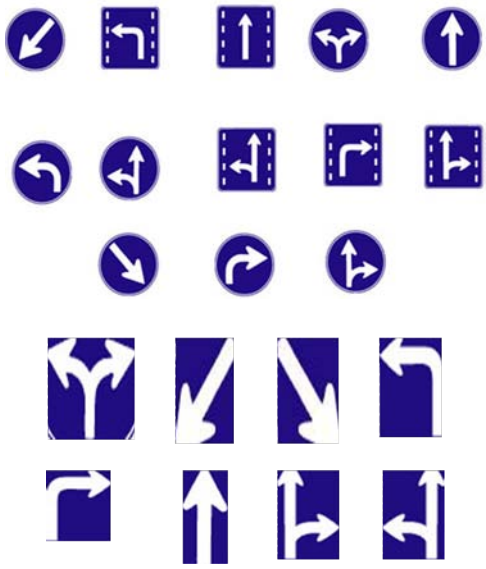


Figure 6.2: The sign and arrow patterns.

6.2 Classification methods

Most of the well-known recognition algorithms have been applied during the last three decades. For example, Haar wavelet, Bayesian and AdaBoost patches algorithms were proposed in [16]. Normalized cross correlation is one of the commonly used techniques in template matching as in [39][68][40]. In [47], the authors have presented a recognition system which makes use of the normalized cross correlation value to obtain the final positive recognition. The candidates are normalized to 50 x 50 pixels and tested over two possible templates. One drawback of this technique is the time consuming.

Radial basis functions neural network are common robust recognition algorithm [42]. Usually in neural network applications, false sign candidates are supposed to be rejected during a verification phase like in [69][70]. Since the verification step in our system does not reject all the false sign candidates, neural network classifiers are not suitable for our system.

Support vector machine (SVM) were used for the shape classification and pattern recognition of segmented traffic signs from image sequences [43]. In [71][72], different Laplace kernel classifiers for different classes of road signs have been used. The input to the classifier is a feature vector constructed from different moments (spatial, central,

normalized) and compactness. The main goal has been to obtain a smooth function which determines the effectiveness of the classifier.

Principal component analysis and SIFT features were also proposed in [44]. But the calculation of SIFT features is time consuming.

In this thesis, two classifiers are discussed. The first classifier is a decision tree with geometrical features, which is fast and sufficient when the number of false arrow candidates is small. Using the two-step algorithm for fragmented sign aggregation, many false arrow candidates must be classified as non-arrow. Therefore, we needed to use the second and more robust classifier which uses principal component analysis subspace method with histogram of oriented gradients features.

6.3 Decision tree using geometrical features

There are many ways for arrow classification [73][30]. Real-time constraint requires the use of fast methods. Because arrow template matching is difficult to tolerate geometrical variations, geometrical features which are invariant to variations are used for a decision tree as shown in Figure 6.3. The classification process consists of the following steps:

1. Classify the segmented arrow according to the height (H) to width (W) ratio of the rectangular area surrounding the segmented arrow. Arrows can be divided into three types:

   - $H/W > 1$: Straight arrow candidate.
   - $H/W < 1$: Left and right connected arrow candidate.
   - $H/W \approx 1$: Other arrows.

   These rules are applied at the root of the tree as shown in Figure 6.3.

2. Find geometrical features from the segmented white object, such as positions of arrow top, arrow bottom, most-left and most-right arrow edges. These features are found by scanning the edges of the white objects.

3. Match the white object to one of the arrow patterns using the features extracted in the previous step. Matching is preformed by applying geometrical-feature rules like what is shown in Table 6.1 through the decision tree shown in Figure 6.3, and decide if

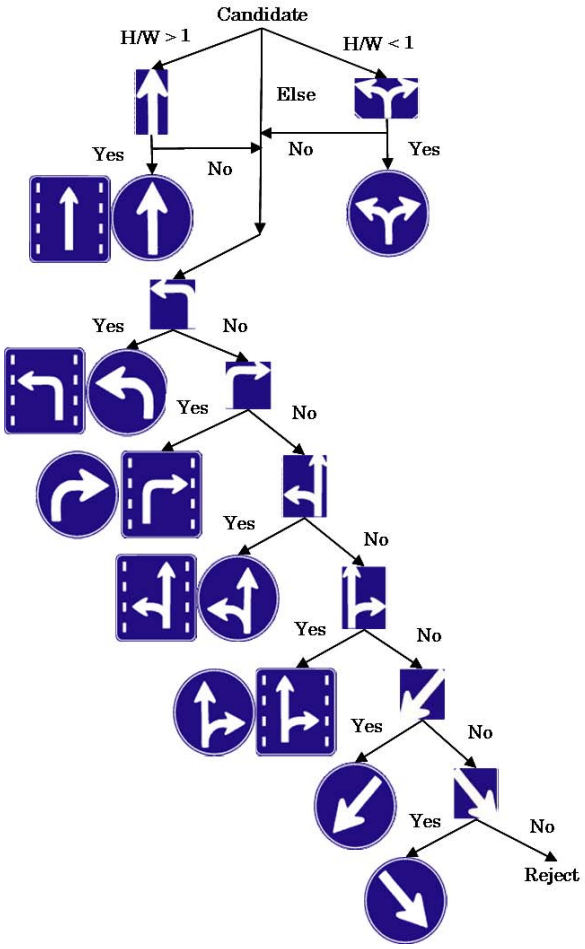
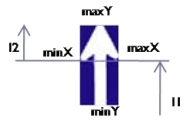
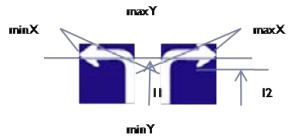
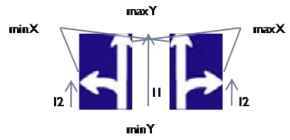
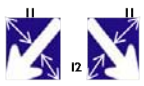


Figure 6.3: A decision tree.

the candidate object is one of the target arrow patterns. If the candidate white object matches one of the targets in the tree, the system shows the classification result in real-time using the program *CameraVideoEffector.exe* described in Section 1.4. An example of the system output is shown in Figure 6.4. If the candidate does not match any target, the blue candidate object is rejected.

Discrimination rules were designed by referring to the design manual describing all traffic signs in Japan [74]. As we know, the car is moving and consequently the size of the sign is changing. Therefore, the geometrical-feature rules should not contain constants. To solve this problem, the rules are designed by using ratios of the height and width of the segmented image, which do not change as the distance between the sign and the camera changes.

Table 6.1: Examples of geometrical rules for classification.

	1- the four points do not conicide 2- $l(minX, minY) < image\,width + \epsilon$[1] 3- $1_1 > 1_2$
	1- $l(maxX, maxYR) < \epsilon, l(minX, maxYL) < \epsilon$ 2- $maxY$ is one of $maxYR$ or $maxYL$ 3- $l(maxYL, maxYR) > \frac{image\,width}{4}$
	1- $0.8 < \frac{l2}{l1} < 1$ 2- for Right arrow: $l(minX, minY) < \sqrt{(\frac{image\,width}{3})^2 + (\frac{image\,height}{3})^2}$ $l(maxX, maxY) < \sqrt{(\frac{image\,width}{3})^2 + (\frac{image\,height}{3})^2}$ * for the left arrow we replace minX and maxX
	1- $\frac{l1}{l2} < 0.5$ 2- for Right Arrow: $l(maxX, minY) < \sqrt{\frac{image\,height}{5}}$ * for the left arrow we replace minX and maxX
	$0.8 < \frac{l1}{l2} < 1.2$

[1] ϵ is a small value for tolerance, $l(x_1, x_2)$ means the cartesian distance between x_1, x_2

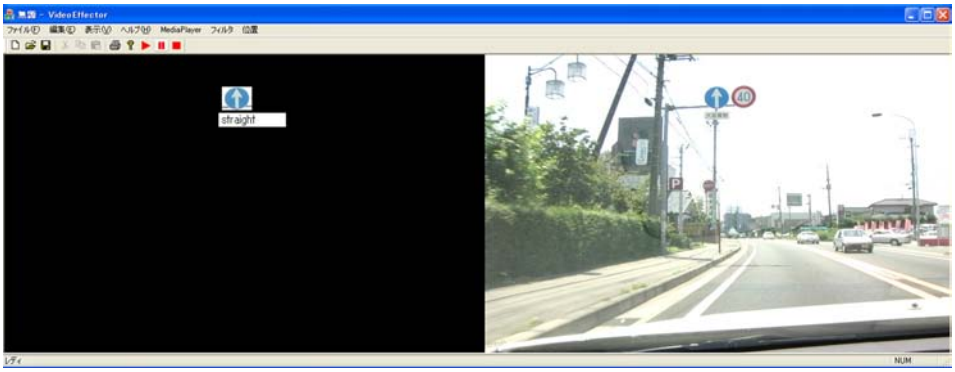


Figure 6.4: System output.

6.4 PCA-HOG classifier

6.4.1 Introduction

Principal components analysis is described as an unsupervised feature extraction technique. It makes no assumptions about the existence or groupings within the data. One important merit of using PCA subspace method is that it does not need a training data for negative class. Negative class can be rejected by an experimentally determined threshold which can be set to less than the minimum projection length of positive data.

6.4.2 Classification using PCA and HOG features

Segmented white objects are normalized to the size of 16 x 16 pixels as shown in Figure 6.5. The normalized region is transformed to 4 x 4 grids of HOG descriptors [75][76]. HOG features are calculated by the following equations:

$$\forall x, y: \ g_x(x, y) = I(x + 1, y) - I(x - 1, y) \tag{6.1}$$

$$\forall x, y: \ g_y(x, y) = I(x, y + 1) - I(x, y - 1) \tag{6.2}$$

where $g_x(x, y)$ and $g_y(x, y)$ denotes the x and y components of the image gradient, respectively.

The magnitude $m(x, y)$ and orientation $\theta(x, y)$ of the image gradient are calculated by:

$$m(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2} \tag{6.3}$$

$$\theta(x, y) = \tan^{-1}(g_y(x, y)/g_x(x, y)) \tag{6.4}$$

Unsigned orientation of the image gradient suggested by Dalal et al. [75] is used:

$$\check{\theta}(x, y) = \begin{cases} \theta(x, y) + \pi \text{ if } \theta(x, y) < 0 \\ \theta(x, y) \qquad\qquad \text{otherwise} \end{cases} \tag{6.5}$$

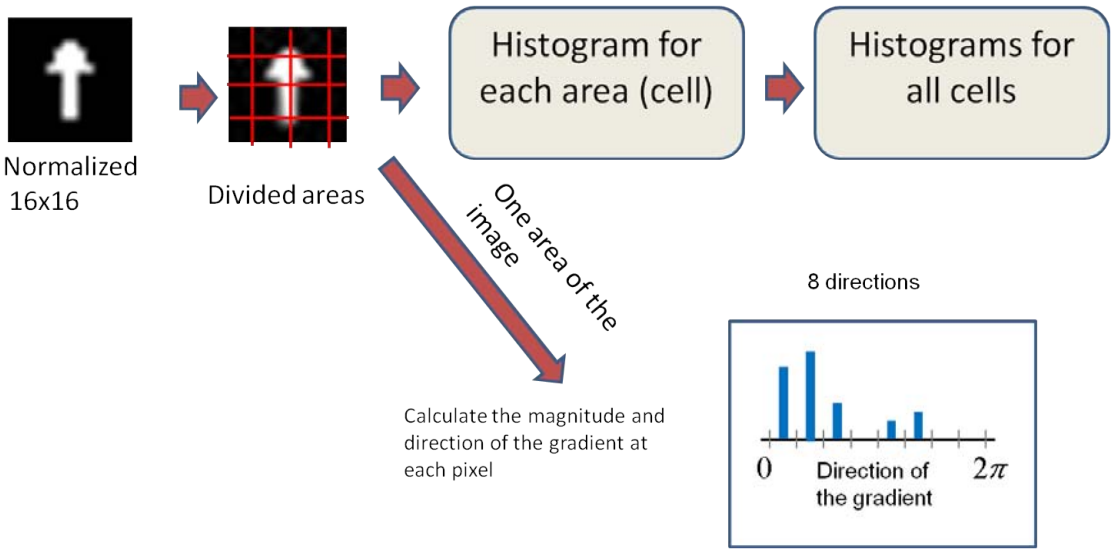


Figure 6.5: The HOG features.

Principal components analysis [77][78][79] is used to derive new variables in decreasing order of importance which are uncorrelated. PCA produces an orthogonal coordinate system in which the axes are ordered in terms of the amount of variance in the original data for which the corresponding principal components account. If the first few principal components account for most of the variation, then these may be used to describe the data, thus leading to a reduced-dimension representation. After extracting the HOG features each arrow image is represented as a vector of dimension 16 * 8 = 128. Training

data of each arrow pattern are divided to subgroups with similar size and color features, then multi-subspace for each pattern is created as shown in Figure 6.6. The unit vector u which gives the maximum value to u^TCu is the eigenvector of the covariance matrix C with the largest eigenvalue. The second, third and subsequent principal component axes are the other eigenvectors sorted by eigenvalue. Each subspace is created from a training set by calculating $N-1$ eigenvectors and eigenvalues [80] where N is the number of data in the smallest training subgroup. The subgroups are described in detail in Section 8.1.

6.4.3 Subspace classifier

The subspace classifier as many other classifiers has two phases: The first is design phase which is the phase of computing the classifier parameters from know classifications. The second is the classification phase, which is the phase of classifying the input vector to the relevant class. The decision rule is the optimality in terms of minimal average error of classification. The central decision rule is based on projection.

Suppose we have k classes $w_1, ..., w_k$. Each class is represented by a subspace: $L_1, ..., L_k$ with $p_i = dim(L_i), i = 1, ..., k$. Each subspace is spanned by p_i orthonormal vector $u_{ij}, j = 1, ..., p_i$. The projection matrix and the basic classification rule are [79]:

$$P_i = \sum_{k=1}^{p_i} u_{ik} u_{ik}^T \in R^{n \times n} \tag{6.6}$$

$$\text{if } \; x^T P_i x > x^T P_j x \text{ for all } j \neq i, \text{then classify } x \text{ in class } w_i \tag{6.7}$$

Or we can use the quadratic form $x^T P_i x = x^T P_i P_i x = \|P_i x\|^2$

The decision rule classify x to the class whose subspace gives the longest projection i.e. the longest norm.

The rejection threshold is determined experimentally. If the maximal projection length on the subspaces is smaller than the threshold, then the input is rejected.

6.5 Summary and discussion

Two classification methods are presented in this chapter. The first classification method is to use a decision tree according to geometrical features which is fast and sufficient enough as far as the number of false arrow candidates are small. The two-step aggregation

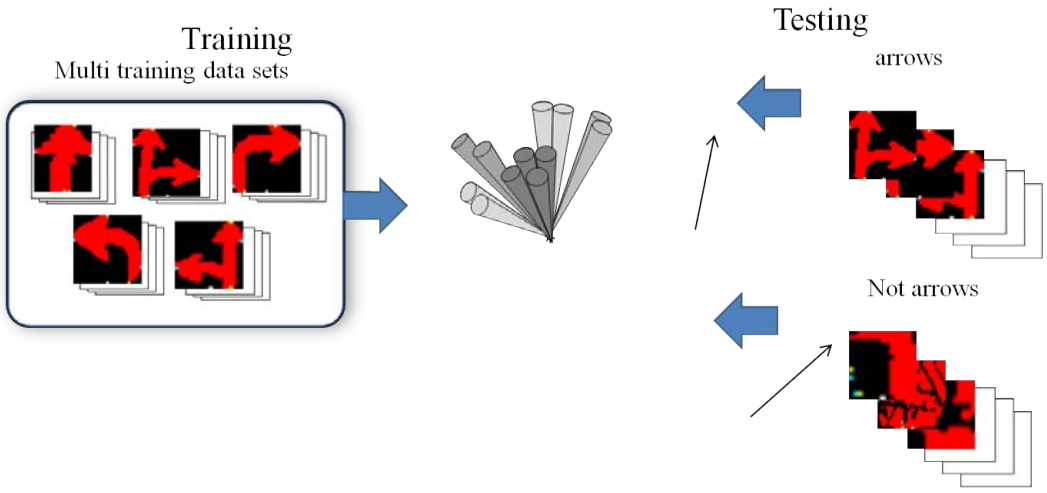


Figure 6.6: The subspace classifier.

method, which was experimentally proved more robust than the dilation method, produced subtle variations in segmented shapes of signs and made it difficult to hand-code the decision tree. Therefore, principal component analysis subspace method with histogram of oriented gradients features is employed. More than one subspace for each arrow class is used to cover complex distributions of arrow patterns.

The test data to be discussed in Chapter 8 consist of 94 rectangular and circular signs and 64 non-sign blue objects with white parts inside them. Applying the decision tree classifier on the 93 correctly segmented true signs, 86 were classified correctly. Applying the PCA classifier, 87 were classified correctly.

Modifying the aggregation algorithm increased the number of false arrow andidates to be rejected by the classifier from 64 non-arrow white objects to 365. Applying the decision tree classifier on the 365 segmented white objects, 302 were classified (i.e. rejected) correctly. While after applying the PCA classifier, 337 were rejected correctly.

Chapter 7

Sign tracking

The final processing stage is to incorporate the sign tracking algorithm [9] to facilitate the detection of the same signs which will appear in a consecutive number of video frames. Since car moves, the positions and sizes of traffic signs also change. In order to identify successive sign images as the same sign, it must be tracked throughout successive frames. To predict the position in the next frame we used a simple linear tracking rule (7.1).

$$p(t+1) = p(t) + (p(t) - p(t-1)) \tag{7.1}$$

where $p(t)$ is a vector coordinate of the center of a sign at time (t). Initial prediction, where the $(p(t) - p(t-1))$ term is not available, is determined according to the motion direction given by the initial position as illustrated in Figure 7.1 [9]. By this function we can facilitate detection of the same sign at the predicted position and frame by frame recognition can be integrated by majority votes. This stage is very simple and need to be improved in the future works. One idea is to incorporate kalman or particle filters.

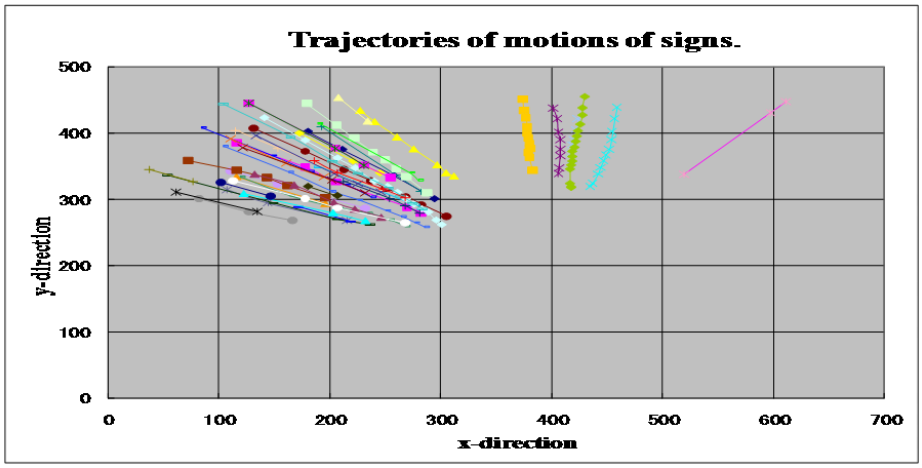


Figure 7.1: Trajectories of motions of signs.

Chapter 8

System performance

8.1 Experimental data description

Two data sets were used, one for designing (the training set) and the other for testing the system (the test set). Both data sets are captured from videos recorded in various places in Tsukuba city in Japan. Videos were recorded in different day times and weather conditions to get various illumination effects. Minimum diameter, or width in the rectangular case, of signs used for training and testing is $D = 33\ pixels$ which is the system limitation.

The training set consists of 90 blue rectangular and circular signs captured from 3 recorded videos. From six to eight images of each sign with different sizes are taken while the car is moving toward the sign. This gives us a total of 600 training segmented arrows as follows:

- 60 signs * 6 sizes = 360 straight signs.
- 6 signs * 8 sizes = 48 left signs.
- 6 signs * 8 sizes = 48 right signs.
- 8 signs * 8 sizes = 64 straight or left signs.
- 10 signs * 8 sizes = 80 straight or right signs.

Straight signs are divided into 4 subgroups according to two factors:

1. The size of the sign (big-small) which is related to the distance of the sign from the car.

2. The brightness of the sign according to the illumination factors. This includes: Good brightness conditions (i.e. in the morning or in high brightness with direct light on the scene) and bad brightness conditions (i.e. in cloudy weather, in the evening or when the sun is facing the camera).

The rest of the signs is divided into 2 subgroups only according the size of the sign because of data limitation. Each subspace consists of each subgroup.

Test set consists of different 149 images captured from other 4 videos recorded at different day times and location of the city. This set includes 158 blue objects which are supposed to be detected as sign candidates by the system. These blue objects contain 94 true rectangular and circular signs and 64 non-sign blue objects with white parts inside them. The signs are of different sizes reflecting the distances from the camera. They were taken in different illuminations and weather conditions (30 in the morning, 11 in the evening, 13 at the sun in face, 46 in high brightness with direct light on the scene and 49 in cloudy weather). The system has to reject the 64 non-sign objects and classify the 94 true signs according to the classification process.

8.2 Results of experiments

Tables 8.1, 8.2 and 8.3 show the improvements in results according to the different algorithms explained in the previous chapters. The first column in each table represents the true categories of inputs and each entry in each row shows the number of classifications.

To evaluate the system, precision and recall, which are two widely used evaluation measures, are calculated. Precision is considered as a measure of exactness or fidelity, while recall is a measure of completeness.

Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives incorrectly labeled as belonging to the class). Score of 1.0 for a class C means that every item labeled as belonging to class C does indeed belong to class C, but says nothing about the number of items from class C that were not labeled correctly.

Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items not labeled as belonging to the positive class but should have been). Score of 1.0 means that every item from class C was labeled as belonging to class C, but says nothing about how many other items were incorrectly labeled as belonging to class C.

In the context of classification tasks, the terms true positives, true negatives, false positives and false negatives are used to compare the given classification of an item (the class label assigned to the item by a classifier) with the desired correct classification (the class the item actually belongs to). This is illustrated by the table in Figure 8.1.

		correct result / classification	
		E1	E2
obtained result / classification	E1	**tp** (true positive) Classified correctly	**fp** (false positive) not signs but Misclassified as signs
	E2	**fn** (false negative) Missed (Rejected or misclassified) signs	**tn** (true negative) Correctly rejected

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

Figure 8.1: Precision and recall.

Table 8.1 shows the result of applying the preliminary version of the system using the multi-threshold labeling algorithm described in Section 4.5 and the aggregating fragmented parts using structured dilation algorithm described in Section 5.5. Classification was performed by the decision tree classifier described in Section 6.3. From 158 blue objects 135 (85%) objects were detected and segmented correctly while 23 were undetected. From those 135 detected objects 128 (81%) objects were classified correctly either as one of the sign patterns or rejected if they are not signs. Precision is 97% and recall is 71%. Average processing time is 110 ms. Regarding the negative data, false alarm or false positive rate is $2/64 = 3.1\%$.

Table 8.1: Results of recognition by applying multi-threshold labeling algorithm, the structured dilation algorithm for aggregation and the decision tree classifier.

Input signs and their number in test data set	Classified to					
	Straight	Left	Right	Straight or left	Straight or right	Not signs
Straight(65)	50					15
Left(7)		2				5
Right (4)			3			1
Straight or left(9)				5		4
Straight or right (9)	1				6	2
Not signs(64)					2	62

Table 8.2 shows the performance on the same 149 images using the opponent color filters and seeded growing algorithm described in Section 4.6 and the preliminary aggregating algorithm using only step 1 described in Section 5.6.1. Classification was also performed by the decision tree classifier described in Section 6.3.

From 158 blue objects 154 (97%) objects were detected and segmented correctly. From those 154 detected objects 127 (80%) objects were classified correctly either as one of the sign patterns or rejected if they are not signs.

It is recognized that although the detection rate increased to 97%, but the recognition rate stayed similar to the results shown in Table 8.1. This is because the system fails to detect the arrows if the two parts of the sign are far from each others or if borders of the signs are connected with noisy parts. Similar cases were discussed in Section 5.6.1 and shown in Figure 5.12 and Figure 5.13. Another reason is that the results in Table 8.2 were calculated by an old version of the step 1. In this version, only one white object which is the closest to the sign center was segmented.

Precision is 94% and recall is 71%. Average processing time is 160 ms. Regarding the negative data, false alarm or false positive rate is $4/64 = 6.2\%$.

Table 8.3 shows the performance of the system of final version on the 149 images using the opponent color filters and seeded growing algorithm described in Section 4.6,

Table 8.2: Results of recognition by applying opponent color filters and seeded growing algorithm for labeling, the aggregating algorithm using only step 1 for segmentation and the decision tree classifier.

Input signs and their number in test data set	Classified to					
	Straight	Left	Right	Straight or left	Straight or right	Not signs
Straight(65)	48					17
Left(7)		4				3
Right (4)			3			1
Straight or left(9)				6		3
Straight or right (9)					6	3
Not signs(64)				2	2	60

the two-step aggregating algorithm described in section 5.6 and the PCA-HOG classifier described in section 6.4.

From 158 blue objects 157 (99%) objects were detected and segmented correctly. Many false arrow candidates were also segmented. The 158 blue objects consist of 94 true rectangular and circular signs and 64 non-sign blue objects with white parts inside them. Because of the segmentation of all white objects surrounded by blue areas, the number of segmented white objects increases. From 94 true arrows 93 were segmented correctly and 87 of them were classified correctly. From 365 non-arrow white objects 337 were rejected correctly. Therefore, we can say that 87+337 white objects were classified correctly either as one of the arrow patterns or rejected if they are not arrows. The classification rate in this case is (87+337)/(94+365) = 93%. Precision is 75% and recall is 93%. Average processing time is 160 ms using step 1 and 190 ms for both steps. The complete process of step 2 will take up to 250 ms. Regarding the negative data, false alarm or false positive rate is 28/365 = 7.6%.

The 190ms/frame average processing time is equivalent to 5.2 fps. In the worst case when the processing time is maximum, the frame rate will be reduced to 4 fps. A car moving in a speed of 60 km/h will pass around 16 m per 1 second. The 4 fps means that

a sign in the range of 16 m will have 4 chances to be recognized. In spite of many difficult problems, the overall performance can be considered satisfactory.

Table 8.3: Results of recognition by applying the opponent color filters and seeded growing algorithm for labeling, the two-step aggregating algorithm for segmentation and the PCA-HOG classifier.

Input signs and their number in test data set	Classified to					
	Straight	Left	Right	Straight or left	Straight or right	Not signs
Straight(65)	61					4
Left(7)		6				1
Right (4)			4			0
Straight or left(9)				8		1
Straight or right (9)					8	1
Not arrow(365)	5	6	8	4	5	337

Chapter 9

Conclusion and future works

This research is a part of large project which aims to build a real-time traffic sign recognition system. In this part, we focused on the recognition of the blue traffic signs designating directions. In addition to the constraint of real-timeness, the system was designed to solve other challenges such as the position of the traffic sign, illumination variations during the day and according to the sun position, weather conditions and signs' color fading. Dealing with blue signs, difficult problems such as discrimination between the blue signs and the sky and sign fragmentation were necessary to be solved.

The preliminary verification process did not use sign shape because there are circular and rectangular blue signs with similar arrows. However, the shape information was used later on for the aggregation of fragmentarily segmented signs.

To achieve the recognition in this research, a set of algorithms and techniques were implemented. These algorithms and techniques are related to image processing, pattern recognition and color segmentation. They include conversion among RGB, HSV and gray-scale coordinate systems, multi-threshold pixel labeling, opponent color filters, seeded region growing, histogram discriminant analysis, structuring dilation morphology, decision tree classification based on geometrical-features, neighborhood edge tracking, region of interest expansion, random sample consensus, convex hull, principal component analysis subspace method and histogram of oriented gradient features.

In the final version of the system, the performance of precision was 75%, recall was 93% and F-measure was 83%.

For future works, we have plans to revise and improve the current algorithms. Moreover, a plan to incorporate an advanced tracking algorithm to track the same signs in

a sequence of video frames to improve the performance is considered. Furthermore, we aim to extend the work to handle other blue and green traffic signs.

Optimizing the system parameters effective even at night is also an important part of our future works.

Acknowledgments

I would like to thank my advisor, Prof. Yuzo Hirai, for accepting me as one of his students, for giving me the opportunity to do this research and for his guidance and continuous support since I have arrived Japan.

I would like to thank Professors Hiroyuki Kudo, Ko Sakai, Keisuke Kameyama and Kazuhiro Fukui in the examining committee for their valuable comments and suggestions.

My thanks are also to the family of Visual Information Processing Laboratory, to all my colleagues and friends with whom I cooperated and to everyone who gave me technical, academic, or moral support.

Also, I would like to express my deep gratitude to my parents and my brothers for their love and support every time. I know how much they miss me and how much they wish the best for me.

My love is to my wife Nour for adding the light (Nour means light in Arabic) to my life.

Finally, I would like to thank the Ministry of Education – Japan (Monbukagakusho), for the financial scholarship support that gave me the opportunity to study and stay in the country I always dreamed to visit (Japan).

Appendix A

Image filter suite

Image Filter Suite is a software package developed in our laboratory. It consists of multi projects under Microsoft Visual Studio C++, which produce Dynamic-link libraries (DLL) as filters to be applied on frames. The DLLs are released to a common folder to be used by one of the two other released programs under Image Filter Suite as described in section 1.4. the two other released programs are:

CameraVideoEffector.exe: for video processing.

ImageFilter.exe: for single video frame image processing. Frame images are captured from the recorded videos.

Each filter project is a DLL type project under Microsoft Visual Studio C++. The project consist of two main files which may be linked to other files.

The main two files are:

- FilterName.def:

The file: FilterName.def.

```
LIBRARY FilterTemplete
EXPORTS
    DoFilter    @1
    GetFilterPluginInfo    @2
```

- FilterName.cpp: Which contains the body of the two base functions in each DLL filter

The first function is to export the name to appear in the filters list.

Function 1: Exporting the name to appear in the filters list.

```
int DLLEXPORT WINAPI GetFilterPluginInfo(LPSTR buf,int buflen){
        lstrcpy( buf, ''filter name'');
        return 1;
}
```

The second function contains the executional body of the DLL filter.

Function 2: The executional body of the DLL filter.

```
//      Filter base function
int DLLEXPORT WINAPI DoFilter( BITMAPINFO *bi, BYTE *imagedata,
   PROGRESSCALLBACK lpPrgressCallback,long lData)
{
        CDIB dib;
        dib.CreateFromDirectMemoryData( bi, imagedata );
//      Filter code goes here.


        return 1;
}
```

Appendix B

HSV coordinate system

The HSV color model is a mathematical representation of color, in a way more similar to our color perception. It breaks the color into 3 components as shown in Figure B.1 [37]:

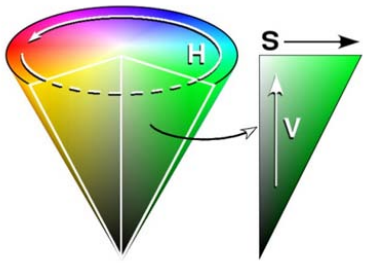


Figure B.1: HSV coordinate system.

Hue (H): the color type such as red, blue, or yellow. Ranges from 0-360 or normalized to 0-100% in some applications.

Saturation (S): the vibrancy of the color. Ranges from 0-100%, also sometimes called the *purity* by analogy to the colorimetric quantities excitation purity and colorimetric purity. The lower the saturation of a color, the more *grayness* is present and the more faded the color will appear.

Value (V): the brightness of the color: Ranges from 0-100%.

This model comes in handy where the RGB model can not help, for instance, classifying similar colors, classifying colors by levels of darkness, sorting colors, etc.

The relation between RGB and HSV

The relation between the two models is as following

Conversion from RGB to HSV

$$MAX = max\{R, G, B\}, MIN = min\{R, G, B\} \tag{B.1}$$

$$H = \begin{cases} (0 + \frac{G-B}{MAX-MIN}) \times 60 & if\ R = MAX \\ (2 + \frac{B-R}{MAX-MIN}) \times 60 & if\ G = MAX \\ (4 + \frac{R-G}{MAX-MIN}) \times 60 & if\ B = MAX \end{cases} \tag{B.2}$$

$$S = \frac{MAX - MIN}{MAX} \tag{B.3}$$

$$V = MAX \tag{B.4}$$

Conversion from HSV to RGB

$$h_i = \left\lfloor \frac{H}{60} \right\rfloor \mod 6 \tag{B.5}$$

$$f = \frac{H}{60} - \left\lfloor \frac{H}{60} \right\rfloor \tag{B.6}$$

$$p = V \times (1 - S) \tag{B.7}$$

$$q = V \times (1 - f \times S) \tag{B.8}$$

$$t = V \times (1 - (1 - f) \times S) \tag{B.9}$$

$$(R, G, B) = \begin{cases} (V, t, p) & if\ h_i = 0 \\ (q, V, p) & if\ h_i = 1 \\ (p, V, t) & if\ h_i = 2 \\ (p, q, V) & if\ h_i = 3 \\ (t, p, V) & if\ h_i = 4 \\ (V, p, q) & if\ h_i = 5 \end{cases} \tag{B.10}$$

Appendix C

RANSAC

* The following information is summarized from Wikipedia, the free encyclopedia and [67].

RANSAC is an abbreviation for "RANdom SAmple Consensus". It is a method to estimate parameters of a mathematical model iteratively from a set of observed data. The idea is depending on a basic assumption that the data consist of observations whose distribution can be explained by some set of model parameters. These observations are called "inliers". The data also contain observations that do not fit the model which is called "outliers". The outliers exist because of noise or wrong measurements. Given a small set of inliers, RANSAC assumes that there exists a procedure which can estimate the parameters of a model that optimally fits the data.

Consider the problem of fitting a 2D line to a set of observations. Figure C.1 shows a data set contains both inliers, and outliers. Using a simple least squares method may produce a bad fit because it is fitted to all points, including the outliers. RANSAC, on the other hand, can produce a better model which is computed from the inliers, assuming that the probability of choosing only inliers in the selection of data is high enough. To keep the level of probability reasonably high, a number of algorithm parameters must be chosen carefully.

The input variables of the algorithm are the data, the model which the data will be fitted to it, the minimum number of data required to fit the model, the number of iterations, a threshold to determine when an observation fits a model, and the number of close data values required to assert that a model fits well to data.

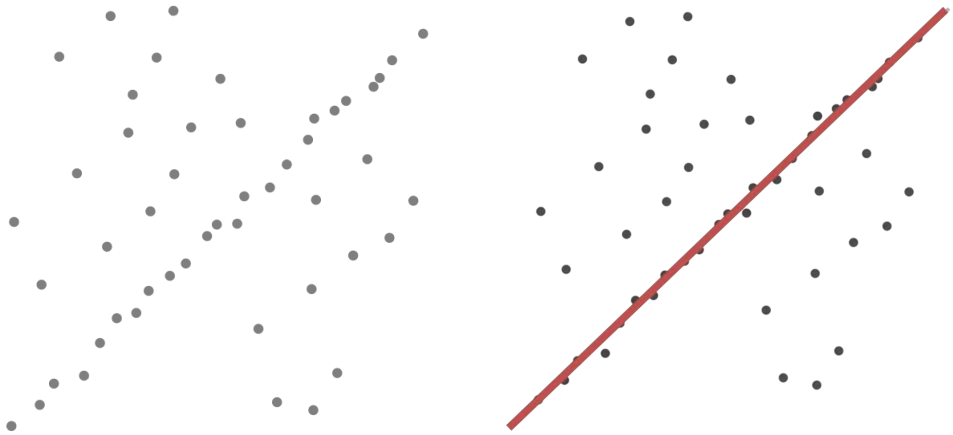


Figure C.1: a: A data set with many outliers for which a line has to be fitted. b: Fitted line with RANSAC, outliers have no influence on the result.

The outputs of RANSAC are the parameters of the model which best fit the data or nil if no good model is found, data points from which this model has been estimated and the best error of this model relative to the data.

RANSAC achieves its goal by iteratively selecting a random subset of the original data. These data are hypothetical inliers and this hypothesis is then tested as follows:

1. A model is fitted to the inliers by reconstructing all free parameters of the model from the inliers.

2. Other data are tested against the fitted model and, if a point fits well to the estimated model, also considered as a hypothetical inlier.

3. The estimated model is reasonably good if sufficiently many points have been classified as hypothetical inliers.

4. The model is re-estimated from all hypothetical inliers, because it has only been estimated from the initial set of hypothetical inliers.

5. Finally, the model is evaluated by estimating the error of the inliers relative to the model.

This procedure is repeated a fixed number of times, each time producing either a model which is rejected because too few points are classified as inliers or a refined model together with a corresponding error measure. In the latter case, we keep the refined model if its error is lower than the last saved model.

Appendix D

Convex hull

* The following information is summarized from [65], [66], [81] and wikipedia.

What is convex, convex hull

A subset S of a plane is called convex if and only if for any pair of points $p, q \in S$ the line segment $\overline{pq}$ is completely contained in S. The Convex hull CH(S) of a set S is the smallest convex set that contains S. The convex hull may be easily visualized by imagining an elastic band stretched and left to surround a given object.

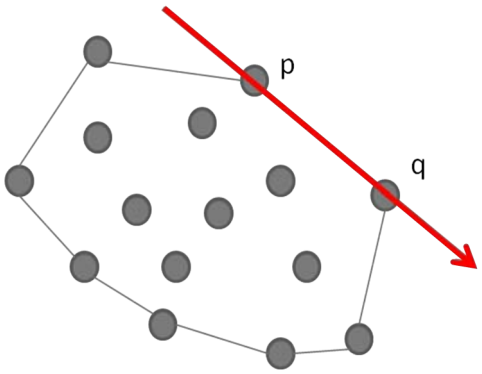


Figure D.1: Convex hull.

Algorithms

The convex hull of a set of n points in the plane can be computed in $O(n \log n)$ using the following algorithm:

Suppose a set P of points in plane. We need to find a list containing the vertices of CH(P) in clockwise order by applying the following steps [81]:

1. Sort the points by x-coordinate to get the sequence $p_1, ..., p_n$. If two points have the same x, value we sort them according to y.

2. Insert the points p_1, p_2 in a list L_{upper} with p_1 first.

3. For i = 3 to n do
   Append P_i to L_{upper}
   While (L_{upper} contains more than 2 points AND the last 3 points in L_{upper} do not make right turn): Do delete the middle of the last 3 points from L_{upper}.

4. Insert P_n, P_{n-1} in the list L_{lower} with P_n first.

5. For i = n-2 down to 1 do
   Append P_i to L_{lower}
   While (L_{lower} contains more than 2 points AND the last 3 points in L_{lower} do not make right turn): Do delete the middle of the last 3 points from L_{lower}.

6. Remove the first and the last points from L_{lower} to avoid point duplication if the upper and lower hull meet.

7. Append L_{lower} to L_{upper} in one list L which is the result convex hull.

Quick hull

A faster algorithm is called QUICK HULL [65][82] because of its similarity to the QUICK SORT. It is a divide-and-conquer algorithm which is based on the idea that most of the points in the given set can be discarded as interior points and we concentrate on the remaining points.

Let S be a set of n points. The algorithm divides S into two subsets which are determined by the line passing through the rightmost lowest point r and the leftmost highest point l. The complete convex hull is composed of two hulls which are: The upper hull and the lower hull. The algorithm works as following

1. Start with two extreme points (x, y).

2. Find a third extreme point z right of line(xy). Point z which is furthest away from line (zy) must be on the hull. Therefore, we can discard all points on or in the triangle (xyz) except for the vertices of the triangle.

3. Repeat the same procedure is recursively on the set of points right of line (xz) and the set of points right of line (zy).

List of publications related to this thesis

Peer-reviewed journals

- M. H. Alsibai and Y. Hirai, "Real-Time Recognition of Blue Traffic Signs Designating Directions," *Int. J. ITS Research*, vol. 8, no. 2, pp. 96–105, March 2010.

Peer-reviewed conference proceedings

- M. H. Alsibai and Y. Hirai, "Real-Time Recognition of Blue Traffic Signs from Blue Sky Background," in *Proc. 17th ITS World Congress*, 10 pages, October 2010.
- M. H. Alsibai and Y. Hirai, "Recognition of Blue Traffic Signs Enhanced By Aggregating Fragmentarily Segmented Sign Images," in *Proc. 13th IASTED Int. Conf. on Signal and Image Processing*, pp. 27–34, December 2011.

Bibliography

[1] WHO, "World health organization – health topics: Injuries, traffic," last visit: 10 2011 (http://www.who.int/topics/injuries_traffic/en/).

[2] WHO, "World report on road traffic injury prevention," tech. rep., World Health Organization, Geneva, 2004.

[3] WDOR.org, "World day of remembrance for road traffic victims," last visit: 10 2011 (http://www.worlddayofremembrance.org/).

[4] WHO, "Global status report on road safety, time for action," tech. rep., World Health Organization, 2009.

[5] WHO, "Road traffic accidents: epidemiology, control and prevention," tech. rep., World Health Organization, Geneva, 1962.

[6] ITSWC10-Korea, last visit: 5 2011 (http://www.itsworldcongress.kr/).

[7] ITS-America, "Intelligent Transportation Society of America," last visit: 12 2011 (http://www.itsa.org/).

[8] ITS-Japan, last visit: December 2011 (http://www.its-jp.org/english/links_e/).

[9] K. Baba and Y. Hirai, "Real-time recognition of traffic signs using opponent color filters," in *Proc. of 14th ITS World Congress (ITSWC2007)*, 12 pages, 2007.

[10] M. H. Alsibai and Y. Hirai, "Real-time recognition of blue traffic signs designating directions," *Int. J. ITS Research*, vol. 8, pp. 96–105, March 2010.

[11] M. H. Alsibai and Y. Hirai, "Real-time recognition of blue traffic signs from blue sky background," in *Proc. of 17th ITS World Congress (ITSWC2010)*, (Busan, Korea), 10 pages, 2010.

[12] M. H. Alsibai and Y. Hirai, "Recognition of blue traffic signs enhanced by aggregating fragmentarily segmented sign images," in *Proc. of 13th IASTED Int. Conf. on Signal and Image Processing*, (Dallas, USA), pp. 27–34, December 2011.

[13] S. H. Hsu and C. L. Huang., "Road sign detection and recognition using matching pursuit method," *Image and Vision Computing*, vol. 19, pp. 119–129, 2001.

[14] A. de la Escalera, J. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image and Vision Computing*, vol. 21, pp. 247–258, 2003.

[15] D. M. Gavrilar, "Traffic sign recognition revisited," in *Proc. of 21st DAGM Symposium fur Musterekennung*, pp. 86–93, 1999.

[16] C.Bahlmann, Y.Zhu, V. Ramesh, M.Pellkofer, and T.Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information," in *Proc. of Intelligent Vehicles Symposium, 2005*, pp. 255–260, IEEE, 2005.

[17] G. Siogkas and E. Dermatas, "Detection, tracking and classification of road signs in adverse conditions.," in *Proc. of Electrotechnical Conference, MELECON IEEE Mediterranean*, pp. 537–540, 2006.

[18] K. Fintzel, R. Bendahan, C. Vestri, S. Bougnoux, and T. Kakinami, "3d parking assistant system," in *Proc. of Intelligent Vehicles Symposium*, pp. 881–886, IEEE, 2004.

[19] R. A. Hicks and R. K. Perline, "Blind-spot problem for motor vehicles.," *Applied optics*, vol. 44, pp. 3893–3897, July 2005.

[20] M. W. Park, K. H. Jang, and S. K. Jung, "Panoramic vision system for an intelligent vehicle using a laser sensor and cameras," in *Proc. of 17th ITS World Congress*, (Busan South korea), 2010.

[21] P. Lindner and G. Wanielik, "Towards 3D Automotive LIDAR Processing - How to deal with Vehicle Dynamics," in *Proc. 17th ITS World Congress*, 10 pages, 2010.

[22] N. Kaempchen and K. C. J. Dietmayer, "FUSION OF LASERSCANNER AND VIDEO FOR ADVANCED DRIVER ASSISTANCE SYSTEMS," in *Proc. of 11th World Congress on Intelligent Transportation Systems*, (Nagoya, Japan), 8 pages, 2004.

[23] Y. Wiseman, "Take a picture of your tire!," in *Proc. of Conference on Vehicular Electronics and Safety (ICVES-2010)*, (Qingdao, ShanDong, China), pp. 151–156, IEEE, 2010.

[24] M. S. Devi and P. R. Bajaj, "Driver fatigue detection based on eye tracking," *Emerging Trends in Engineering and Technology, International Conference on*, pp. 649–652, 2008.

[25] Nissan, "Nissan technology magazine," last visit: 10 2011 (http://www.nissan-global.com/EN/TECHNOLOGY/MAGAZINE/).

[26] H. Kasai and K. Onoguchi, "Pedestrian Detection Using Local Pedestrian Classifiers," in *Proc. 17th ITS World Congress*, 12 pages, 2010.

[27] "Survey by faculty of transportation sciences, technical university in prague, czech," last visit: 10 1999 (http://euler.fd.cvut.cz/research/rs2/index.html).

[28] L. Priese, V. Rehrmann, R. Schian, and R. Lakmann, "Traffic sign recognition based on color image evaluation," in *Proc. of Intelligent Vehicles Symposium*, pp. 95–100, IEEE, 1993.

[29] L. Priese, J. Klieber, R. Lakmann, V. Rehrmann, and R. Schian, "New results on traffic sign recognition," in *Proc. of Intelligent Vehicles Symposium*, pp. 249–254, IEEE, 1994.

[30] S. Vacek, C. Schimmel, and R. Dillmann, "Road-marking analysis for autonomous vehicle guidance," in *Proc. of European Conference on Mobile Robots*, ECMR, 6 pages, 2007. online proceedings.

[31] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, "Recognition of traffic signs based on their colour and shape features extracted using human vision models," *Journal of Visual Communication and Image Representation - ELSEVIER*, vol. 17, pp. 675–685, 2006.

[32] R. Marmo and L. Lombardi, "Road bridge sign detection and classification," in *Proc. of Intelligent Transportation Systems Conference*, pp. 823–826, IEEE, 2006.

[33] D. G. Shaposhnikov, L. N. Podladchikova, A. V. Golovan, N. A. Shevtsova, K. Hong, and X. Gao, "Road sign recognition by single positioning of space-variant sensor window," in *Proc. of* 15^{th} *International Conference on Vision Interface*, 2002.

[34] M. Shneier, "Road sign detection and recognition." Submitted to the IEEE Computer Society International Conference on Computer Vision and Pattern Recognition, June 2005.

[35] V. Rehrmann, R. Lakmann, and L. Priese, "A parallel system for real-time traffic sign recognition," *On line*, 1995.

[36] R. Ach, N. Luth, and A. Techmer, "Real-time detection of traffic signs on a multi-core processor," in *Proc. of IEEE Intelligent Vehicles Symposium*, pp. 307–312, 2008.

[37] Wikipedia, "Wikipedia, the free encyclopedia," last visit: 10 2011 (http://en.wikipedia.org).

[38] D. Deguchi, M. Shirasuna, K. Doman, I. Ide, and H. Murase, "Intelligent traffic sign detector: Adaptive learning based on online gathering of training samples," in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, (Baden-Baden, Germany), pp. 72–77, June 2011.

[39] J. Miura, T.Kanda, and Y.Shirai, "An active vision system for real-time traffic sign recognition," in *Proc. of Intelligent Transportation Systems, 2000*, pp. 52–57, IEEE, 2000. ISBN: 0-7803-5971-2.

[40] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," in *Proc. of IEEE Intelligent Vehicles Symposium*, pp. 566–571, 2004.

[41] Y. Aoyagi and T. Akasura, "A study on traffic sign recognition in scene image using genetic algorithms and neuronal networks," in *Proc. of the 22nd Int. Conf. On Industrial Electronics, Control and Instrumentation*, vol. 3, (Taipei, Taiwan), pp. 1838–1843, 1996.

[42] W. J. Kuo and C. C. Lin., "Two-stage road sign detection and recognition," in *Proc. of International Conference on Multimedia and Expo*, pp. 1427–1430, IEEE, 2007.

[43] C. G. Kiran, L. V. Prabhu, V. A. Rahiman, and K. Rajeev., "Traffic sign detection and pattern recognition using support vector machine," in *Proc. of 7th International Conference on Advances in Pattern Recognition*, pp. 87–90, 2009.

[44] A. Ihara, H. Fujiyoshi, M. Takagi, H. Kumon, and Y. Tamatsu, "Improved matching accuracy in traffic sign recognition by using different feature subspaces," in *Proc. of 11th IAPR Conference on Machine VIision Applications.*, pp. 130–133, 2009.

[45] A. Soetedjo and K. Yamada, "Traffic sign classification using ring partitioned method," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E*, vol. 88-A, no. 9, pp. 2419–2426, 2005.

[46] M. de Saint Blancard, "Road sign recognition: a study of vision-based decision making for road environment recognition," *Vision-based Vehicle Guidance, Springer-Verlag*, pp. 162–172, 1992.

[47] G. Piccioli, E. D. Michelib, P. Parodi, and M. Campani, "Robust method for road sign detection and recognition," *Image and Vision Computing*, vol. 14, pp. 209–223, 1996.

[48] A. Broggi, P. Cerri, P. Medici, P. P. Porta, and G. Ghisio, "Real time road signs recognition," in *Proc. of IEEE Intelligent Vehicles Symposium*, pp. 981–986, 2007.

[49] M. Benallal and J. Meunier, "Real-time color segmentation of road signs," in *Proc. of Canadian Conference on Electrical and Computer Engineering CCECE*, vol. 3, pp. 1823–1826, 2003.

[50] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognition*, vol. 43, pp. 416–430, Jan. 2010.

[51] T. Zin and H. Hama, "Robust road sign recognition using standard deviation.," in *Proc. of The 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 429–434, 2004.

[52] G. Siogkas and E. Dermatas., "Detection, tracking and classification of road signs in adverse conditions," in *Proc. of Electrotechnical Conference, MELECON 2006. IEEE Mediterranean*, pp. 537–540, 2006.

[53] G. Mo and Y. Aoki, "Recognition of traffic signs in color images.," in *Proc. of TENCON 2004 IEEE Region 10 Conference*, vol. B, pp. 100–103, 2004.

[54] L. Spillmann and J. S. Werner, *Visual Perception: The Neurophysiological Foundation.* San Diego, California: Academic Press, 1990.

[55] P. Gouras, "Webvision, the organization of the retina and visual system, color vision.," last visit: 10 2011 (http://webvision.med.utah.edu/book/part-vii-color-vision/color-vision/).

[56] N. A. Mat-Isa, M. Y. Mashor, and N. H. Othman, "Seeded region growing features extraction algorithm; its potential use in improving screening for cervical cancer," *International Journal of the Computer, the Internet and Management*, vol. 13(1), pp. 61–70, 2005.

[57] R. D. Boyle and R. C. Thomas, *Computer Vision: A First Course*. Blackwell Scientific Publications, 1988.

[58] D. Kang, N. Griswold, and N. Kehtarnavaz, "An invariant traffic sign recognition system based on sequential color processing and geometrical transformation.," in *Proc. of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 88–93, 1994.

[59] S. Zhu, L. Liu, and X. Lu, "Color-geometric model for traffic sign recognition," in *Proc. of IMACS Multiconference on Computational Engineering in Systems Applications*, pp. 2028–2032, 2006.

[60] A. de la Escalera, J. Armingol, J. Pastor, and F. Rodriguez, "Visual sign information extraction and identification by deformable models for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems.*, vol. 5, no. 2, pp. 57–68, 2004.

[61] A. Soetedjo and K. Yamada, "Fast and robust traffic sign detection.," in *Proc. of International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1341–1346, IEEE, Oct. 2005.

[62] J. Parker, *Algorithms for Image Processing and Computer Vision*. John Wiley and Sons Inc, 1997.

[63] A. Z. Arifin and A. Asano, "Image thresholding by histogram segmentation using discriminant analysis," in *Proc. of Indonesia-Japan Joint Scientific Symposium: IJJSS'04*, pp. 169–174, 2004.

[64] E. DAVIES, *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann, 3^{rd} ed., 2004.

[65] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[66] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, pp. 469–483, Dec. 1996.

[67] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[68] M. Betke and N. C. Makris, "Recognition, resolution, and complexity of objects subject to affine transformations," *Int. J. Comput. Vision*, vol. 44, no. 1, pp. 5–40, 2001.

[69] D. M. Gavrila and V. Philomin, "Real-time object detection for 'smart' vehicles," in *Proc. of The Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 87–93, 1999.

[70] U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Puetzold, and C. Wohler, "Autonomous driving goes downtown," *Intelligent Systems and their Applications*, vol. 13, pp. 40–48, Nov/Dec 1998.

[71] P. Paclík and J. Novovičová, "Road sign classification without color information," in *Proc. of the 6th Conference of Advanced School of Imaging and Computing*, 2000.

[72] P. Paclík, J. Novovičová, P. Pudil, and P. Somol, "Road sign classification using laplace kernel classifier," *Pattern Recognition Letters*, vol. 21, pp. 1165–1173, December 2000.

[73] L. Wendling and S. Loria, "Recognition of arrows in line drawings based on the aggregation of geometric criteria using the choquet integral," in *Proc. of 7th International Conference on Document Analysis and Recognition*, pp. 299–303, IEEE, 2003.

[74] In Japanese, *Traffic Signs Handbook*. Japan Contractors Association of Traffic Signs and Lane Makings (JCASM), 2004.

[75] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of International Conference on Computer Vision & Pattern Recognition CVPR2005*, vol. 2, (San Diego), pp. 886–893, June 2005.

[76] W. L. Lu and J. J. Little, "Simultaneous tracking and action recognition using the pca-hog descriptor," in *Proc. of The Third Canadian Conference on Computer and Robot Vision (CRV-06)*, (Quebec, Canada), June 2006.

[77] A. R.Webb, *Statistical Pattern Recognition*. John Wiley and Sons Inc, 3^{rd} ed., 2002.

[78] D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice Hall, 1st ed., 1982.

[79] E. Oja, *Subspace Methods of Pattern Recognition*. John Wiley & Sons, 1984.

[80] H. Moon and P. Phillips, "Computational and Performance aspects of PCA-based Face Recognition Algorithms," *Perception*, vol. 30, pp. 303–321, 2001.

[81] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer, 2^{nd} ed., 2008.

[82] R. B. Muhammad, "Computational geometry lecture note on-line." Kent State University Lecture, last visit: 10 2011.