

COLOR AND SHAPE BASED TRAFFIC SIGN DETECTION

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE ULAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

NOVEMBER 2008

Approval of the thesis

**COLOR AND SHAPE BASED TRAFFIC SIGN DETECTION**

submitted by **Emre Ulay** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen \_\_\_\_\_  
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmek \_\_\_\_\_  
Head of Department, **Electrical and Electronics Engineering Dept., METU**

Prof. Dr. Gözde Bozdağı Akar \_\_\_\_\_  
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Assoc. Prof. Dr. Mehmet Mete Bulut \_\_\_\_\_  
Co-Supervisor, **Electrical and Electronics Engineering Dept., METU**

**Examining Committee Members:**

Assoc. Prof. Dr. Aydın Alatan \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gözde Bozdağı Akar \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Mehmet Mete Bulut \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Çağatay Candan \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Emre Turgay, MSc. \_\_\_\_\_  
MGEO, ASELSAN

**Date:** 27.11.2008

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name : Emre Ulay

Signature :

## **ABSTRACT**

### **COLOR AND SHAPE BASED TRAFFIC SIGN DETECTION**

Ulay, Emre

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Gözde Bozdağı Akar

Co-Supervisor: Assoc. Prof. Dr. Mehmet Mete Bulut

November 2008, 93 pages

In this thesis, detection of traffic signs is studied. Since, both color and shape properties of traffic signs are distinctive; these two properties have been employed for detection.

Detection using color properties is studied in two different color domains in order to examine and compare the advantages and the disadvantages of these domains for detection purposes.

In addition to their color information, shape information is also employed for detection purpose. Edge information (obtained by using the Sobel Operator) of the images/frames is considered as search domain to find triangular, rectangular, octagonal and circular traffic signs.

In order to improve the performance of detection process a joint implementation of shape and color based algorithms is utilized. Two different methods have been used

in order to combine these two features. Both of the algorithms help reducing the number of pixels to check whether they belong to a sign or not. This, of course, reduces the processing time of detection process.

Each utilized algorithm is tested and compared with the others by using both static images from different sources and video streams. Images having adverse properties are used in order to state algorithms response for some specific conditions such as bad illumination and shadow. After implementation, results show that joint implementation of the color and shape based detection algorithms produces more accurate results. Moreover, joint implementation reduces the processing time of the detection process when compared to application of algorithms individually since it diminishes the search domain.

**Keywords:** Traffic Sign Detection, Object Detection, Driver Support System, Color Segmentation, Shape Based Detection, Color Based Detection

## ÖZ

### RENK VE ŞEKİL TEMELLİ TRAFİK İŞARETİ TESPİTİ

Ulay, Emre

Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Gözde Bozdağı Akar

Ortak Tez Yöneticisi: Doç. Dr. Mehmet Mete Bulut

Kasım 2008, 93 sayfa

Bu tezde, trafik işaretlerinin tespiti üzerine çalışılmıştır. Trafik işaretlerinin hem renkleri hem de şekilleri kolaylıkla ayırt edilebildiği için bu iki özellik tanıma işlemi için kullanılmıştır.

Birbirlerine karşı avantaj ve dezavantajlarının belirlenmesi amacıyla renk özelliklerinin kullanılarak işaret tespiti amacıyla iki farklı renk alanı kullanılmıştır.

Renk özelliklerine ek olarak, trafik işaretlerinin şekil özellikleri de işaret tespiti amaçlı kullanılmıştır. Sobel kenar yakalayıcısıyla elde edilen kenar özellikleri üçgen, dörtgen, sekizgen ve dairesel trafik işaretlerinin tespiti amacıyla kullanılmıştır.

Tespit işlemi performansının artırılması amacıyla, renk ve şekil temelli trafik işareti tespit metotları beraberce kullanılmıştır. Birbirinden farklı iki farklı yolla bu iki özelliğin birleştirilmesi sağlanmıştır. Bu iki yol da arama yapılacak olan bilgi alanını daraltmıştır. Böylece tespit işlemi işlem zamanının kısalması sağlanmıştır.

İşaret tespiti amacıyla faydalanılan yöntemler farklı kaynaklardan elde edilen resimler ve videolar kullanılarak test edilmiş ve birbirleriyle karşılaştırılmıştır. Uygulanan yöntemlerin farklı koşullar altındaki tepkilerinin bulunması amacıyla kötü aydınlatılmış, gölgeli gibi elverişsiz özelliklere sahip resimlerden yararlanılmıştır. Uygulama sonuçlarından, şekil ve renk temelli metotların beraberce kullanılmasının daha kesin sonuçlar yarattığı gözlenmiştir. Buna ek olarak, bu metotların beraber uygulanması aramak yapmak için kullanılan bilgi alanının daralması nedeniyle işlem zamanını kısaltmıştır.

**Anahtar Kelimeler:** Trafik İşareti Tespiti, Nesne Tespiti, Sürüş Destek Sistemi, Renk Bölümlenme, Şekil Temelli Tespit, Renk Temelli Tespit

## **ACKNOWLEDGMENTS**

I wish to express my deepest thanksgiving to my supervisor, Prof. Dr. Gzde BOZDAĐI AKAR for her boundless help, excellent supervision and leading guidance from beginning to end of thesis work.

I also express my sincere gratitude to my co-supervisor, Assoc. Prof. Dr. Mehmet Mete BULUT, for his initiative ideas and guidance that helped to construct this work.

I would also like to thank my family for their precious support and endless patience since the beginning of primary school till to the end of MSc. study.

Also thanks to my wife for making my life more bearable.



# TABLE OF CONTENTS

ABSTRACT .....	iv
ÖZ .....	vi
ACKNOWLEDGMENTS .....	viii
TABLE OF CONTENTS .....	ix
LIST OF FIGURES .....	xi
LIST OF TABLES .....	xiii
LIST OF ABBREVIATIONS .....	xvi
CHAPTERS	
1. INTRODUCTION	
1.1. Scope of Thesis .....	3
1.2. Thesis Outline .....	3
2. SURVEY ON TRAFFIC SIGN DETECTION	
2.1. Introduction .....	5
2.2. Detection of Traffic Signs .....	6
2.2.1. Detection Using Color Feature .....	7
2.2.2. Detection Using Shape Feature .....	11
3. DETECTION OF TRAFFIC SIGNS	
3.1. Color Based Detection .....	17
3.1.1. Color Based Detection Algorithm I .....	17
3.1.1.1. Candidate Detection .....	18
3.1.1.2. Color Segmentation .....	19
3.1.1.3. Locating Sign Candidates .....	19
3.1.1.4. Labeling Connected Regions .....	23
3.1.1.5. Calculating Candidate Properties & Applying Constraints .....	24
3.1.2. Color Based Detection Algorithm II .....	25
3.1.2.1. Candidate Detection .....	25
3.1.2.2. Labeling Candidates & Checking Validity .....	29
3.2. Shape Based Detection .....	30
3.2.1. Detection of Circular Traffic Signs .....	30
3.2.1.1. Edge Detection .....	30
3.2.1.2. Candidate Detection .....	32
3.2.1.3. Choosing the Parameters .....	36
3.2.1.3.1. Set of Radii .....	36
3.2.1.3.2. Gaussian Kernel $A_n$ .....	37
3.2.1.3.3. Radial Strictness Parameter .....	37
3.2.1.3.4. Scaling Factor .....	38
3.2.2. Detection of Triangular, Rectangular and Hexagonal Traffic Signs .....	38
3.2.2.1. Edge Detection .....	39
3.2.2.2. Candidate Detection .....	39
3.2.3. Detection Circular Traffic Signs by Using Hough Transform .....	44

3.2.3.1.	Edge Detection.....	44
3.2.3.2.	Circular Hough Transform.....	44
3.3.	Joint Treatment of Color and Shape Features for Detection.....	48
3.3.1.	Combination of Color and Shape Detection by And Rule.....	48
3.3.2.	Combination of Color and Shape Detection by Product Rule.....	51
4.	EXPERIMENTS AND RESULTS	
4.1.	Introduction.....	54
4.2.	Test Results on Separate Algorithms.....	54
4.2.1.1.	Test Results on Static Images in Set-1.....	56
4.2.1.2.	Test Results on Static Images in Set-2.....	60
4.2.1.3.	Test Results on Static Images in Set-3.....	64
4.2.1.3.1.	Test Results on Illumination Group.....	65
4.2.1.3.2.	Test Results on Occlusion Group.....	68
4.2.1.3.3.	Test Results on Translation Group.....	70
4.2.1.3.4.	Test Results on Rotation Group.....	72
4.2.1.3.5.	Test Results on Shadows Group.....	74
4.2.1.3.6.	Test Results on Deformed Group.....	76
4.3.	Test Results on Joint Application of Color and Shape Based Algorithms.....	79
4.3.1.	Test Results on Static Images.....	79
4.3.2.	Test Results on Video Streams.....	82
5.	CONCLUSIONS AND FUTURE WORK	
5.1.	Conclusions.....	86
5.2.	Future Work.....	87
	REFERENCES.....	89

# LIST OF FIGURES

## FIGURES

Figure 2.1	Detection of Traffic Signs using color feature .....	7
Figure 2.2	Hue, Saturation look up tables.....	8
Figure 2.3	Detection of Traffic Signs using shape feature.....	12
Figure 3.1	Detection process.....	18
Figure 3.2	Original picture having red and blue traffic signs. ....	20
Figure 3.3	Blue traffic sign candidates of picture in Figure 3.2 .....	21
Figure 3.4	Red traffic sign candidates of picture in Figure 3.2.....	21
Figure 3.5	3x3 median filtered blue candidates of picture in Figure 3.2 .....	22
Figure 3.6	3x3 median filtered red candidates of picture in Figure 3.2 .....	23
Figure 3.7	$hd_{red}$ & $hd_{blue}$ functions .....	26
Figure 3.8	Function for detection of color saturation .....	27
Figure 3.9	Original picture having red and blue traffic signs. ....	28
Figure 3.10	Blue traffic sign candidates of picture in Figure 3.2 .....	28
Figure 3.11	Red traffic sign candidates of picture in Figure 3.2.....	29
Figure 3.12	Original image containing one circular traffic sign.....	31
Figure 3.13	Magnitude of gradients in x and y directions for image in Figure 3.10	32
Figure 3.14	Locations of positively and negatively affected pixels ( $n = 2$ ).....	33
Figure 3.15	$O_n$ function for radius equals to 18 pixel. ....	34
Figure 3.16	Block diagram showing the main steps of the process.....	36
Figure 3.17	(i) Pixels on darker background. (ii) Gradient from adjacent pixels. (iii) Number of gradients pointing at each pixel $O_n$ , (iv) square of the number of gradients pointing at each pixel $O_n$ .....	38
Figure 3.18	Voting lines associated with the gradient element $g(p)$ for different shapes.....	40
Figure 3.19	Size of a voting line for single gradient element.....	40
Figure 3.20	Three edge points $p_i$ on a triangle. (i) The angles of the unit gradient vectors. (ii) The resulting vectors obtained by multiplying the gradient angles by $n = 3$ .....	42
Figure 3.21	N-angle gradient projected from a point $p$ .....	42
Figure 3.22	The contribution of edge pixels in parameter space .....	46
Figure 3.23	Original image for which Accumulator Matrix is calculated .....	46
Figure 3.24	Accumulator Matrix for Figure 3.23 .....	47
Figure 3.25	The combination of color and shape detection by And rule process.....	49
Figure 3.26	Intermediate and final outputs of joining color and shape based detection process.....	50
Figure 3.27	The combination of color and shape detection by Product rule process	52

Figure 3.28 Intermediate and final outputs of joining color and shape based detection process.....	53
Figure 4.1 Original image having one red one blue traffic sign .....	61
Figure 4.2 Blue candidates. Blue sign is discarded because it is not isolated from the non sign region.....	61
Figure 4.3 Search domains for each shape based algorithms.....	81

## LIST OF TABLES

### TABLES

Table 2.1 Features of traffic signs.....	5
Table 2.2 Color enhancement values (normalized to 255) .....	9
Table 2.3 Threshold values used by Kuo, Lin in color segmentation stage.....	11
Table 4.1 Detection results with algorithm Color I in RGB Domain .....	58
Table 4.2 Detection results with algorithm Color I in HSV Domain .....	58
Table 4.3 Detection results with algorithm Color II .....	58
Table 4.4 Detection results with shape algorithms .....	59
Table 4.5 Detection results with shape algorithms (constraints optimized) .....	59
Table 4.6 Detection results with algorithm Color I in RGB Domain .....	62
Table 4.7 Detection results with algorithm Color I in RGB Domain (constraints optimized) .....	62
Table 4.8 Detection results with algorithm Color I in HSV Domain .....	62
Table 4.9 Detection results with algorithm Color I in HSV Domain (constraints optimized) .....	63
Table 4.10 Detection results with algorithm Color II .....	63
Table 4.11 Detection results with algorithm Color II (constraints optimized) .....	63
Table 4.12 Detection results with shape algorithms .....	64
Table 4.13 Detection results with shape algorithms (constraints optimized) .....	64
Table 4.14 Detection results with algorithm Color I in RGB Domain .....	66
Table 4.15 Detection results with algorithm Color I in RGB Domain (constraints optimized) .....	66
Table 4.16 Detection results with algorithm Color I in HSV Domain.....	66
Table 4.17 Detection results with algorithm Color I in HSV Domain (constraints optimized) .....	66
Table 4.18 Detection results with algorithm Color II .....	66
Table 4.19 Detection results with algorithm Color II (constraints optimized) .....	67
Table 4.20 Detection results with shape algorithms .....	67
Table 4.21 Detection results with shape algorithms (constraints optimized) .....	67
Table 4.22 Detection results with algorithm Color I in RGB Domain .....	68
Table 4.23 Detection results with algorithm Color I in RGB Domain (constraints optimized) .....	68
Table 4.24 Detection results with algorithm Color I in HSV Domain.....	68
Table 4.25 Detection results with algorithm Color I in HSV (constraints optimized) .....	69
Table 4.26 Detection results with algorithm Color II .....	69
Table 4.27 Detection results with algorithm Color II (constraints optimized) .....	69
Table 4.28 Detection results with shape algorithms .....	70
Table 4.29 Detection results with shape algorithms (constraints optimized) .....	70

Table 4.30	Detection results with algorithm Color I in RGB Domain .....	70
Table 4.31	Detection results with algorithm Color I in RGB Domain (constraints optimized) .....	70
Table 4.32	Detection results with algorithm Color I in HSV Domain.....	71
Table 4.33	Detection results with algorithm Color I in HSV Domain (constraints optimized) .....	71
Table 4.34	Detection results with algorithm Color II .....	71
Table 4.35	Detection results with algorithm Color II (constraints optimized) .....	71
Table 4.36	Detection results with shape algorithms .....	72
Table 4.37	Detection results with shape algorithms (constraints optimized) .....	72
Table 4.38	Detection results with algorithm Color I in RGB Domain .....	72
Table 4.39	Detection results with algorithm Color I in RGB Domain (constraints optimized) .....	72
Table 4.40	Detection results with algorithm Color I in HSV Domain.....	73
Table 4.41	Detection results with algorithm Color I in HSV Domain (constraints optimized) .....	73
Table 4.42	Detection results with algorithm Color II .....	73
Table 4.43	Detection results with algorithm Color II (constraints optimized) .....	73
Table 4.44	Detection results with shape algorithms .....	74
Table 4.45	Detection results with shape algorithms (constraints optimized) .....	74
Table 4.46	Detection results with algorithm Color I in RGB Domain .....	74
Table 4.47	Detection results with algorithm Color I in RGB Domain (constraints optimized) .....	75
Table 4.48	Detection results with algorithm Color I in HSV Domain.....	75
Table 4.49	Detection results with algorithm Color I in HSV Domain (constraints optimized) .....	75
Table 4.50	Detection results with algorithm Color II .....	75
Table 4.51	Detection results with algorithm Color II (constraints optimized) .....	75
Table 4.52	Detection results with shape algorithms .....	76
Table 4.53	Detection results with shape algorithms (constraints optimized) .....	76
Table 4.54	Detection results with algorithm Color I in RGB Domain .....	77
Table 4.55	Detection results with algorithm Color I in RGB Domain (constraints optimized) .....	77
Table 4.56	Detection results with algorithm Color I in HSV Domain.....	77
Table 4.57	Detection results with algorithm Color I in HSV Domain (constraints optimized) .....	77
Table 4.58	Detection results with algorithm Color II .....	77
Table 4.59	Detection results with algorithm Color II (constraints optimized) .....	78
Table 4.60	Detection results with shape algorithms .....	78
Table 4.61	Detection results with shape algorithms (constraints optimized) .....	78
Table 4.62	Detection results for combination of color and shape based algorithms by And rule.....	80
Table 4.63	Detection results for combination of color and shape based algorithms by Product rule .....	80
Table 4.64	Results for combination of algorithms by And rule for Video-1 .....	82
Table 4.65	Results for combination of algorithms by And rule for Video-2.....	83
Table 4.66	Results for combination of algorithms by And rule for Video-3.....	83
Table 4.67	Results for combination of algorithms by And rule for Video-4.....	84

Table 4.68 Results for combination of algorithms by Product rule for Video-1 .....	85
Table 4.69 Results for combination of algorithms by Product rule for Video-2 .....	85
Table 4.70 Results for combination of algorithms by Product rule for Video-3 .....	85
Table 4.71 Results for combination of algorithms by Product rule for Video-4 .....	85

## **LIST OF ABBREVIATIONS**

- CHT : Circular Hough Transform  
HLS : Hue Luminance Saturation  
HSI : Hue Saturation Intensity  
HSV : Hue Saturation Value  
IP : Image Processing  
IHLS : Improved Hue Luminance Saturation  
RGB : Red Green Blue  
ROI : Region of Interest



# CHAPTER 1

## INTRODUCTION

Today, one of the most important concerns of the automobile manufacturers is to improve safety. Advances in automobile industry allow manufacturer to produce cars which carry more electronic equipment. As electronics becomes more powerful and reliable some tasks are shifted from human responsibility to computers'. That is, embedded systems on cars. In addition to active safety systems adapted to cars, manufacturers produce systems which avoid accidents to occur.

In the last a few decades, like automobile industry computer vision has also improved fast as well. As a result of this improvement some systems developed on obstacle detection and road detection as an aid to driver. However, traffic sign detection is a wide research area which still needs to be worked on [9].

Driving is completely based on visual information processing [7]. In this process, traffic signs play a key role on determining the road conditions and situations. They provide the driver all the information necessary for a safe drive. Signs prevent the driver doing inappropriate actions such as driving above speed limit, going in wrong direction, passing through a passing forbidden zone etc [8]. Further, traffic signs are helpful for direction finding and guidance.

Shapes, colors and material of signs are defined by the government and a sign has always same shape, color and physical dimensions throughout the whole country. Although traffic signs are apparent and have several obvious characteristics, some conditions may prevent driver perceiving them. For instance, at night or in bad lighting conditions drivers are less likely to notice the traffic signs. Some distracting events on road may result in a skip of signs. Moreover, sometimes only the driver her/himself is not able to notice the signs due to lack of concentration. In this

manner, a driver aiding systems will definitely improve safety in such conditions and prevent some accidents to occur.

According to one research, 60 percent of crashes at intersections and 30 percent of head-on collisions could be avoided if the driver had an additional half-second to react and about 75 percent of vehicular crashes are caused by inattentive drivers [6]. Therefore, small contributions from some kind of assistance system may prevent considerable amount of accident to occur.

In addition to assistance it provides to careless drivers, such an aiding system is one of the most important parts of autonomous vehicles. Though, researches on automatic traffic sign detection and recognition system started recently, it has been one of the most rapidly developing subject in autonomous vehicle projects [14].

At first glance, although the traffic sign detection system has some beneficial characteristics, some inevitable effects make this task quite complex and challenging. The main problems arising can be listed as follows:

- Illumination of signs varies continuously during day. Especially in bad lighting conditions, it is harder to gather color and contour information of traffic signs [10], [7].
- Some weather conditions such as rain, snow or fog reduce the ease of getting some sign characteristics.
- Traffic signs may partially be occluded by other objects. This may render a sign very hard, even impossible to detect [10], [7].
- Although their shapes and colors are deterministic, size and aspect ratio of the traffic sign vary enormously. This makes things more complicated.
- Installation of signs and surface material of signs may physically be damaged or changed.
- Camera mounted on the car will always suffer from vibrations [10].
- Computational complexity of the applied algorithm should let the system run in real time. A late detection would completely be useless [10].

## **1.1. Scope of Thesis**

In this thesis, detection of traffic signs in static images and in video streams is studied. Static images and video streams -used in comparisons of detection of algorithms- are mostly found from the internet. Static images are found from the internet from different sources. Traffic signs in each set of test image have some adverse effects, which enables providing more realistic results.

In the thesis, mainly two different features of traffic signs are to be used for detection. The first approach explores the traffic signs by using their color properties. Second, traffic signs are detected by using their shape properties.

The major difference in using color based algorithms occurs in the determination of the color domain. Therefore, three different color segmentation algorithms in two different color domains are implemented and compared. When it comes to detection by shape features, triangular, rectangular and hexagonal signs are explored within the frame separately. Signs are searched within an image containing the edge information obtained by using Sobel Operator.

As a final step, algorithm used for shape and color based detection are combined to improve detection process by means of detection rate and processing time.

Algorithms applied are developed in both MATLAB and C++ (via OpenCV libraries) environments. Because of its better performance (better processing times) C++ environment is used for tests.

## **1.2. Thesis Outline**

In Chapter 2, a brief history of traffic sign detection process is given. In this chapter detection process is considered as having two different approaches. First of these two is to use color properties; the other one is to use shape properties of signs. Major techniques in both approaches are mentioned. Moreover, several studies which use combination of these two features are cited.

In Chapter 3, application of algorithms using color features and shape features of traffic signs will be explained in a detailed manner. In addition to that, approaches used to combine color and shape based detection algorithms are going to be explored.

In Chapter 4, implemented algorithms are tested by using both static images and video streams obtained from different sources. Results obtained from these algorithms are discussed. Not only the detection algorithms but also the environments in which results are obtained are also compared.

In Chapter 5, conclusion is drawn and possible future studies are discussed.

## CHAPTER 2

### SURVEY ON TRAFFIC SIGN DETECTION

#### 2.1. Introduction

Improving safety is a key goal in road vehicle development [4]. Driver support systems that help drivers for changing road conditions can potentially improve safety [5]. However, in road environment there are too many objects that can be accepted as a traffic sign by a computer. Therefore, to build a driver support system which has ability to classify only the true traffic signs and react against them, detection of true objects is the key.

Traffic signs are designed to attract drivers' attention easily. Therefore their physical properties are so important. Their shapes and colors are distinctive, and different from other signs which we may encounter in nature. Table 2.1 shows the pairs of type-color and type-shape of traffic signs respectively.

**Table 2.1** Features of traffic signs

<b>Sign Type</b>	<b>Possible (Border) Colors</b>	<b>Sign Shape</b>
<b>Restricting &amp; Warning</b>	Red, Blue, Black	Triangle, Rectangle, Octagon, Circle
<b>Information</b>	Blue, Red	Rectangle
<b>Highway Information</b>	Green	Rectangle

A driver support system is a system which supports the driver in controlling the car - especially in critical conditions- while keeping the driver in the loop [2]. Within this

area of the driver support, there are mainly three tasks [9]. First of these three is road detection and following, which has been studied for a long time. Consequently, there are many commercially available solutions about this topic. Second part of the driver support system is obstacle detection and avoidance. Though, researches on this area also yield some products such as Active Cruise Control Systems, this is still an open area to work on [9]. Finally, last one is traffic sign detection and recognition, which is the latest to start among others [9].

There are many algorithms developed for detection of traffic signs so far. Below is a summary of these studies and researches.

## **2.2. Detection of Traffic Signs**

In a driver support system, module responsible from detecting traffic signs, investigates the frame supplied from a video capturing device -possibly a video camera- in order to find regions which possibly contains traffic sign. Detection module feeds these regions over a frame to another module which is responsible for recognition.

While investigating the possible traffic sign locations in a frame several properties of signs can be used. Studies on traffic sign detection are concentrated on mainly two approaches. First and the most widely used one is segmentation via color features of the signs. In this group of work traffic signs are investigated in the gathered frame by using several different color domains.

Second one is using shape feature for detection. In this approach, generally, segmentation is achieved by using border detection in a monochromatic image. Then, some shape detection algorithms are applied to this gradient image.

In addition to these major approaches to sign detection, there are also some studies which use a common two step strategy. First, a pre-segmentation is done by using color properties. Color segmentation is done in any of the color domains. Subsequently, final decision of traffic sign detection is achieved shape based features, applied only to the pre-segmented regions.

### 2.2.1. Detection Using Color Feature

The use of color features for traffic sign detection generally consists of stages which are similar for all methods. That is, first of all, segmentation is applied for each desired color in a previously defined color space such as HSV, RGB, YUV, etc. Then, a binary image is formed which consists of regions satisfying the segmentation rules (considered as “1”) and regions not satisfying them (considered as “0”). The next step is applying some image processing algorithms (IP) in order to improve the results obtained by segmentation. For instance, some filtering application may result in a reduction in noise or some dilation algorithms may prevent from some incorrect results due to a small number of pixels. Finally, group of pixels which are labeled as “1” are used to localize a traffic sign. This process can be summarized in Figure 2.1



**Figure 2.1** Detection of Traffic Signs using color feature

Damavandi and Mohammadi [24] propose that although the HSI color space usually gives better results due to its robustness in brightness variation, the YCbCr color space leads to the best practical result since its luminance component Y is separated. Moreover, extraction of red chrominance is less time consuming.

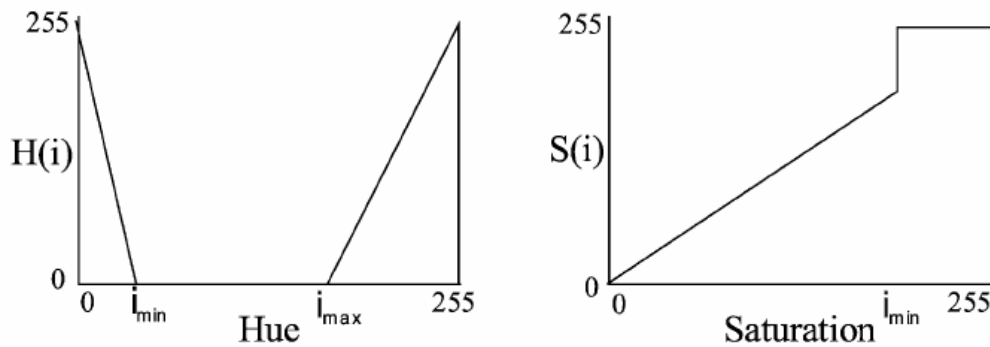
Then, they apply an averaging filter and a thinning algorithm to remove speckles and to leave the skeleton and reduce the number of data for the remaining stages respectively.

Since HSI color domain has different information for each component, Escalera et al. [12], [13] use this color domain for segmentation. Only Hue and Saturation components are taken into account for detection. Intensity component is discarded.

For each component taken into consideration, a look up table is formed. The main idea in forming look up tables for both Hue and Saturation component is that; if one of the components for a sign has a deceiving value, the result would be improved by the other one. The look up tables for Hue and Saturation components are determined according to formulas given in (2.1) and (2.2) respectively. The two look up tables constructed can be seen in Figure 2.2.

$$H(i) = \begin{cases} 255 \frac{i_{\min} - i}{i_{\min}} & 0 \leq i \leq i_{\min} \\ 0 & i_{\min} \leq i \leq i_{\max} \\ 255 \frac{i - i_{\max}}{i_{\max}} & i_{\max} \leq i \leq 255 \end{cases} \quad (2.1)$$

$$S(i) = \begin{cases} i & 0 \leq i \leq i_{\min} \\ 255 & i_{\min} \leq i \leq 255 \end{cases} \quad (2.2)$$



**Figure 2.2** Hue, Saturation look up tables.

Here the  $i$  values stands for former values for Hue and Saturation.  $H(i)$ ,  $S(i)$  are the new values for these quantities. Once the two intermediate images ( $H(i)$ ,  $S(i)$ ) has been formed they are multiplied and normalized to 255.

In their later studies, Escalera et al. [12] use a similar color enhancement algorithm. Values (normalized to 255) for these studies can be seen in Table 2.2.



**Table 2.2** Color enhancement values (normalized to 255)

	<b>Hue Min.</b>	<b>Hue Max.</b>	<b>Saturation Min.</b>
<b>Red Signs</b>	11-224	0(255)	23
<b>Blue Signs</b>	128-143	137	84

Ghica, Yu and Yuan [25] state that to evaluate the closeness between a pixel and a reference color, the distance between the pixel color and the reference color is calculated. Distance calculation is done RGB color space. Since forming basis vectors for RGB color space are Red, Green and Blue any pixel color  $c$  can be represented by:

$$c = c1 * Red + c2 * Green + c3 * Blue \quad (2.3)$$

where  $0 \leq c1, c2, c3 \leq 1$

For a given reference color (expressed by using same basis functions)  $r = (r1, r2, r3)$ , the distance between arbitrary color  $c$  and  $r$  is:

$$c = \sqrt{(r1 - c1)^2 + (r2 - c2)^2 + (r3 - c3)^2} \quad (2.4)$$

Then a threshold function  $F$  is given by:

$$\tilde{O}_n(p) = \begin{cases} (0,0,0) & \text{if } \|r - c\| \leq t \\ c & \text{if } \|r - c\| > t \end{cases} \quad (2.5)$$

where  $t$  is a suitable threshold.

Finally, they apply a morphological filter in order to eliminate noise and fill some gaps formed after due to thresholding. The morphological filter has a set of rules and implemented by a modified Hopfield Network [25].

Zhu and Liu [29] present a new approach to detection of traffic sign by using their color property. In this approach, in order to simplify the complexity caused by the colors; color standardization is employed. That is, color information is reduced to five types of standard colors namely red, yellow, blue, white and black. 16,777,216 kinds of color (from 24-bit bitmap) are mapped into only five different colors. They state that, this process significantly simplifies complexity of signs' color information.

Color classification model consists of four Support Vector Machine classifiers. Neural Network formed is composed of three input neurons for R, G, and B components respectively. Similarly, five output neurons are used for five standard colors. Consequently, network formed accomplishes the mapping of 16,777,216 colors into a single space with five binary vectors.

Kantawong [14] states that since RGB color is very sensitive to lighting changes, segmentation in this color domain would produce incorrect results. However, HSI color domain has its own difficulties. The problem with HSI is that; formulas used to obtain values are non-linear. Hence, computational cost for changing from RGB to HSI is too high. Therefore, instead of these color domains, Kantawong uses a modified version of RGB. This method is computationally simple when compared to HSI as well. In this process, color thresholding is used as contrast improvement technique.

Color threshold process used by Kantawong [14] has the following expressions:

$$p(x, y) = 255 - \frac{\sum(f_R, f_G, f_B)}{3} \quad (2.6)$$

where  $f_R, f_G, f_B$  are red green and blue values of the selected pixel.

$$p(x, y)_{avg} = \frac{\left(10 * p(x, y) + \sum_1^8 g(x, y)\right)}{9} \quad (2.7)$$

where  $g(x, y)$  is the neighboring pixels of the selected pixel.

$$p(x, y) = \begin{cases} \text{black} & \text{if } p(x, y)_{avg} \geq 3 * P(x, y)_{avg} \\ \text{white} & \text{if } p(x, y)_{avg} < 3 * P(x, y)_{avg} \end{cases} \quad (2.8)$$

where  $P(x, y)_{avg}$  is the average color value of all pixels in the image.

Kuo, Lin [15] use the HSI color space since it is immune to lighting changes. Each image element is classified according to its hue saturation and intensity. Table 2.3 shows the threshold values used in color segmentation stage.

**Table 2.3** Threshold values used by Kuo, Lin in color segmentation stage.

	<b>Red</b>	<b>Blue</b>
<b>Hue</b>	$H \geq 0$ and $H \leq 0.111\Pi$ $H \geq 1.8\Pi$ and $H \leq 2\Pi$	$1.066\Pi < H \leq 1.555\Pi$
<b>Saturation</b>	$0.1 < S \leq 1$	$0.28 < S \leq 1$
<b>Intensity</b>	$0.12 < I \leq 0.8$	$0.22 < I \leq 0.5$

Although HSV is one of the most widely used color spaces used in color segmentation, some ranges for Saturation and Value quantities makes this space totally useless. That is, Vitabile et al. [17] defined three different areas in HSV color space. These three regions are:

Achromatic Area :  $S \leq 0.25$  or  $V \leq 0.2$  or  $V \geq 0.9$

Unstable Chromatic Area :  $0.25 \leq S \leq 0.5$  and  $0.2 \leq V \leq 0.9$

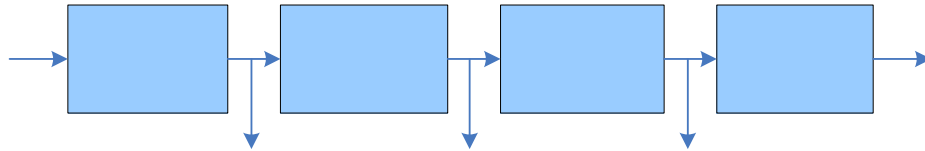
Chromatic Area :  $S \geq 0.5$  and  $0.2 \leq V \leq 0.9$

### 2.2.2. Detection Using Shape Feature

Shape is an important attribute of road signs. Therefore, this attribute can be used for sign detection. Using shape features does not require color information and can be

achieved in gray scale images. Shape detection is more robust to changes in illumination conditions as it detects shapes based on edges or boundary, and will efficiently reduce the search for a road sign regions from the whole image to a small number of pixels [26]. However cluttered scene, imperfect shape of signs, occlusion of other objects may cause the task to be quite challenging. General detection process using the shape properties of signs may be summarized as in Figure 2.3.

Reliability of shape detection mostly depends on the boundary detection or matching algorithm. Better boundary finding algorithms result in better shape detection. This becomes difficult when the road sign appears relatively small in the image, a situation that often occurs in low resolution cameras [26].



**Figure 2.3** Detection of Traffic Signs using shape feature

Garcia-Garrido, Sotelo and Martin-Gorostiza [20] use Canny [21] edge detection algorithm to gather the gradient image. In order to make detection more reliable they have chosen to adapt two Canny Thresholds in a dynamic way, depending on the histogram distribution of the image. Therefore, the histogram of the image has been divided into eight part and two threshold levels have been assigned to each of them. This approach enables them to use this edge detection algorithm in changing visibility conditions.

They use generalized Hough Transform [30] to find any curves that can form a road sign. Hough Transform for straight lines is employed for triangular signs. A straight line is expressed as,

$$x \cdot \cos(\theta) + y \cdot \sin(\theta) = \rho \quad (2.9)$$

where the parameter space  $p = (\rho, \theta)$  should be quantized.

Similarly, circular signs are detected by using Circular Hough Transform, in which a circle (with center coordinates  $(X, \Psi)$  and with radius  $\rho$ ) is expressed as,

$$(x - X)^2 + (y - \Psi)^2 - \rho^2 = 0 \quad (2.10)$$

where the parameter space,  $p = (X, \Psi, \rho)$  should be quantized.

The Hough Transform reaches a prohibitive computational cost, if all edges in the image are analyzed. Therefore, in order to reduce the computational cost Garcia-Garrido, Sotelo and Martin-Gorostiza [20] use a ‘‘chain code’’ [22] to find the acceptable edges.

Liu, Ikenaga and Goto [23] use an accurate edge detector based on differential geometry for their road sign detection method. In edge detection stage, first color image gradient metric matrix is computed as:

$$M_C = J_C^T J_C = \begin{bmatrix} M_{XX} & M_{XY} \\ M_{XY} & M_{YY} \end{bmatrix} \quad (2.11)$$

where  $J_C$  is the Jacobian Matrix of the image:

$$J_C = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \\ \frac{\partial b}{\partial x} & \frac{\partial b}{\partial y} \end{bmatrix} = \begin{bmatrix} r_X & r_Y \\ g_X & g_Y \\ b_X & b_Y \end{bmatrix} \quad (2.12)$$

$$M_{XX} = (r_X)^2 + (g_X)^2 + (b_X)^2 \quad (2.13)$$

$$M_{YY} = (r_Y)^2 + (g_Y)^2 + (b_Y)^2 \quad (2.14)$$

$$M_{YY} = r_X \times r_Y + g_X \times g_Y + b_X \times b_Y \quad (2.15)$$

The larger eigenvalue of matrix  $M_c$  is calculated using:

$$V = \left( \left( M_{XX} + M_{YY} + \sqrt{(M_{XX} + M_{YY})^2 - 4 \cdot (M_{XX} \times M_{YY} - M_{XY}^2)} \right) / 2 \right) \quad (2.16)$$

And its corresponding eigenvector is  $\{M_{XY}, V - M_{XX}\}$ .

The gradient direction angle  $\theta$  is defined by eigenvector:

$$\theta = \arctan \left( \frac{V - M_{XX}}{M_{XY}} \right) \quad (2.17)$$

Liu, Ikenaga and Goto compute the gradient magnitude and direction of each pixel by computing corresponding eigenvalue  $V(i, j)$  and eigenvector  $\theta(i, j)$  respectively.

Then, they separate finding geometric patterns process into two parts:

- Circular and Elliptical Traffic Signs
- Rectangular and Triangular Traffic Signs

For circular and elliptical object detection, since circles may appear as ellipses from different point of view, algorithm searches for ellipses. To fix five parameters (center coordinates  $(x_c, y_c)$ , semi-major axis length  $a$ , semi-minor axis length  $b$  and orientation  $\vartheta$ ) of an ellipse algorithm first selects two edge pixel as two vertices of the major axis of a potential ellipse and calculate four parameters  $((x_c, y_c), a, \vartheta)$  of ellipse. The one more edge pixel is used to accumulate last parameter  $b$  in a one dimensional array  $A_b$ .

For given arbitrary ellipse:

$$b = \sqrt{\frac{a^2 (y \cos \vartheta + x \sin \vartheta - y_c)^2}{a^2 - (x \cos \vartheta - y \sin \vartheta - x_c)^2}} \quad (2.18)$$

By finding peaks in the array  $A_b$  true ellipses are detected.

However, for triangular and rectangular object, since both rectangles and triangles consist of straight line segments, probability of a line segment being side of one of them depends on that of its neighbors [23]. Hence they build an MRF model [27] to label and to identify whether these segments are sides of a triangle (rectangle) or not.

Vitabile et al. [16], [17] use a shape classification by means of a similarity coefficient analysis. This method assumes that both sample and segmented image have the same dimensions. For each sample sign, segmented region is rotated from -5 to +5 degrees with a step of 1 degree. The similarity coefficient is calculated for angle value to find the greatest similarity factor. The similarity function between two images is calculated according to Tanimoto distance measure [19].

Wu et al. [28] use a procedure very similar to method proposed by Loy and Barnes [5]. In their algorithm, every edge element votes for a potential center of a traffic sign. They state that, algorithm developed increases speed and reduces the execution time 2~5 times theoretically when compared to similar algorithms.

Yves et al. [18] introduced a template matching encoding that allows different templates to be combined according to the embedded color information. With this encoding, a template is constructed for an object, and a correlation computation can be defined, which serves as a measure for computing matches between the templates. The method is fast and can be easily modified to include new classes of signs.

Sequential appliance of color shape detection has been proposed by Escalera et al. [9], [12]. In their study, first, hue and saturation components of the image are analyzed and the regions satisfying some color restrictions are detected. Then, a search for traffic signs is done in these regions by using a genetic algorithm. However, main disadvantage in sequential application of color and shape based detections is that, regions falsely rejected by color segmentation can not be recovered in further processing.

J. Torresen et al. [37] use the following steps in order to detect Norwegian speed signs,

- Image Filtering: This part extracts the three colors red, white and black. Filtering is achieved in RGB color domain in order to emphasize the red circle on the speed signs. Broad definition of colors is employed since various intensities and saturations can occur in real life images [37].
- Template Matching: This part of the algorithm is used to locate the sign in an image by searching the red circle surrounding the numbers.

J. Miura et al. [38] made use two different cameras for detection purpose. One camera –the wide camera– in the system is for detection of candidates of traffic signs. Candidates are obtained by using color, intensity and shape information. For each candidate, second camera –the telephoto camera– is used to capture the candidate in a larger size. Edge information in the algorithm is employed for the candidates gathered by color and intensity analysis. That is to say, color and shape detection are applied consecutively.

Fang et al. [7] propose a joint treatment of color and shape properties. The authors compute a feature map of the entire image frame, based on color and gradient information simultaneously, while incorporating a geometry model of signs. This approach prevents formation of regions falsely rejected by color segmentation.



## CHAPTER 3

### DETECTION OF TRAFFIC SIGNS

A road sign is distinct in its color, shape, and pattern. It is designed to offer its basic meaning through the combination of these three properties. Therefore, most of the solutions rely heavily on these features of a road sign [26]. Algorithms for detection of traffic signs can accumulate into two groups:

- Color Based Algorithms
- Shape Based Algorithms

After the detection algorithms using color and shape were deployed separately, two different joint detection methods using the previously developed algorithms (color, shape) in principal are developed. The usage of both color and shape detection algorithms concurrently is to combine the advantages of both methods.

#### 3.1. Color Based Detection

In color based detection, two different algorithms are utilized. The first of them uses the RGB color domain for segmentation, while the second one uses HSV domain. However, both of the algorithms use the sequence given in Figure 2.1 for segmentation purpose. Algorithms applied for color based detection find the red and blue traffic signs.

##### 3.1.1. Color Based Detection Algorithm I

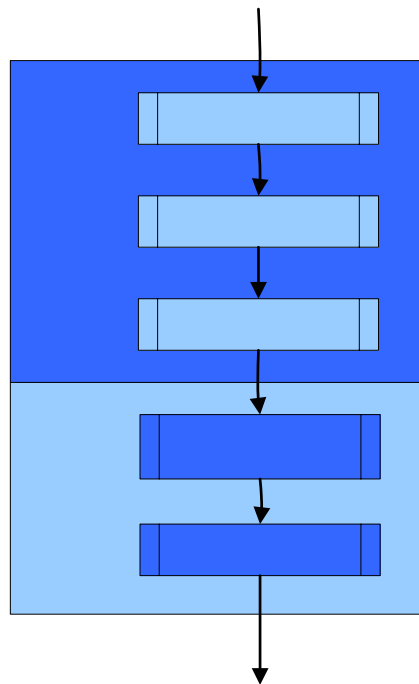
In the proposed algorithm, traffic signs are classified into two groups. First group consists of traffic signs with red borders. These signs are used as a warning or a restricting sign. Second group is composed of information signs. Since all signs in the first group have red border, and second group of signs are blue; a color based algorithm can be applied to detect them [1].

The proposed algorithm uses the color and shape properties of the traffic signs in order to locate the candidate regions [1]. Detection algorithm mainly consists of two parts:

- Candidate detection
- Filtering candidates

### 3.1.1.1. Candidate Detection

In the candidate detection part of the algorithm, color properties of the signs are used. At the second step, several constraints on the shape properties are used to eliminate some of the candidates which can not be a sign. The whole detection process can be summarized as in Figure 3.1.



**Figure 3.1** Detection process

### 3.1.1.2. Color Segmentation

Although the HSV color space is invariant to changing lighting conditions since it has its own computational complexity, color segmentation is done in RGB color space [1]. Output of this color segmentation process provides a binary image which shows either the location is a part of a candidate of the first or the second group of signs (possible sign) or not [1].

### 3.1.1.3. Locating Sign Candidates

In order to detect traffic signs in red or having a red border around, two different criteria are used. Decision of applying one of them is a matter of illumination. First criterion shows good results in good lighting conditions. However, in bad lighting conditions some of the signs can be missed or misdetection may occur. First criterion states, the pixel belongs to a red candidate region if,

$$(I_{i,j}^R \geq 50) \& (I_{i,j}^R - I_{i,j}^G \geq 15) \& (I_{i,j}^R - I_{i,j}^B \geq 15) \quad (3.1)$$

where  $I_{i,j}^{R,G,B}$  is the components of the pixel of image with coordinates  $(i, j)$  [1].

Although the second criterion provides fair result in badly illuminated regions, it is too sensitive to lighting. That is, any small changes in lighting may result in inappropriate changes in candidate regions. However, this method provides more robust sign detection on dark images.

$$k = \frac{255}{\max(I_{i,j}^R, I_{i,j}^G, I_{i,j}^B)} \quad (3.2)$$

$$\tilde{I}_{i,j}^R = I_{i,j}^R \cdot k \quad (3.3)$$

$$\tilde{I}_{i,j}^G = I_{i,j}^G \cdot k \quad (3.4)$$

$$\tilde{I}_{i,j}^B = I_{i,j}^B \cdot k \quad (3.5)$$

Pixel belongs to a red sign if

$$\left(\tilde{I}_{i,j}^R - \tilde{I}_{i,j}^G \geq 10\right) \& \left(\tilde{I}_{i,j}^R - \tilde{I}_{i,j}^B \geq 10\right) \quad (3.6)$$

In order to detect information signs (second group of signs, blue or green ones) the constraint applied is given as:

The pixel belongs to an information sign if

$$M_{i,j} - I_{i,j}^R \geq 0.4 \cdot I_{i,j}^R \quad (3.7)$$

where

$$M_{i,j} = \max(I_{i,j}^G, I_{i,j}^B) \quad (3.8)$$

Below Figure 3.2 shows the original picture from which blue and red sign candidates are extracted. Blue and red sign candidates (pixels satisfying the formula (3.7) and (3.1)) are seen in Figure 3.3 and Figure 3.4 below respectively as well.



**Figure 3.2** Original picture having red and blue traffic signs.



**Figure 3.3** Blue traffic sign candidates of picture in Figure 3.2



**Figure 3.4** Red traffic sign candidates of picture in Figure 3.2

Although the application of equations (3.1) and (3.7) to the image provides satisfactory results for detection of traffic signs in well illuminated pictures, it also produces noisy pixels as well. These noisy pixels are considered as sign candidates and therefore, each of them is labeled independently as explained in 3.1.1.4.

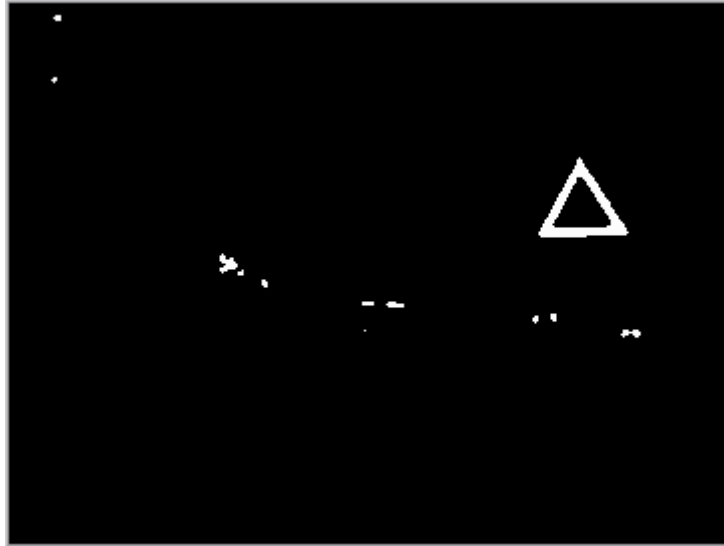
Each labeled traffic sign candidate is checked whether it satisfies the constraints in 3.1.1.5. Hence the number of candidates increases the computation time reasonably. The greater the number of candidates, the more time algorithm needs.

Since these noisy pixels happen to be a kind of salt and pepper noise, a 3x3 median filter is applied to binary outputs (in Figure 3.3 and Figure 3.4) in order to get rid of these noisy pixels. Application of 3x3 median filter to binary outputs greatly eliminates these unwanted pixels. Figure 3.5 and Figure 3.6 shows the filtered blue and red traffic sign candidates respectively.

After the application of the median filter, in order to improve the binary image, dilation with a 3x3 rectangular structuring elements is applied once.



**Figure 3.5** 3x3 median filtered blue candidates of picture in Figure 3.2



**Figure 3.6** 3x3 median filtered red candidates of picture in Figure 3.2

The HSV color domain is not chosen in [1] due to its own computational complexity. Nevertheless, similar segmentation algorithm in HSV color space is also implemented to make a gain-loss comparison. Since HSV domain enables evaluating color and intensity quantities separately, it is more reliable in changing lighting conditions. However, changing domain from RGB to HSV increases computation time. Criteria used in segmentation in HSV domain is:

Pixel belongs to a red sign if

$$\left( (I_{Hue} > 150) \parallel (I_{Hue} < 10) \right) \& (I_{Sat} > 120) \quad (3.9)$$

Pixel belongs to a blue sign if

$$(100 < I_{Hue} < 130) \& (I_{Sat} > 120) \quad (3.10)$$

while hue value lies within  $[0, 180]$  and saturation value lies in  $[0, 255]$ .

#### **3.1.1.4. Labeling Connected Regions**

When the traffic signs candidates are located, the next step is labeling the connected candidate regions.

While generating the groups 8-neighbors are used. The candidate regions are grouped as one if they are in 8-neighbors; else a new group is formed.

After the all candidate regions are grouped, the coordinates and dimensions of a box bounding each region are calculated. These values are to be used in order to check some shape constraints for each connected region. In this part of the process area of the candidate region (number of pixels corresponding to each region) is also calculated as well.

### **3.1.1.5. Calculating Candidate Properties & Applying Constraints**

In some cases the detection algorithm returns too many candidate regions. Therefore, next step is to mark all connected regions as a candidate and check for some physical constraints. These constraints ensure possible sign candidates to satisfy some size and aspect ratio values. Candidates having too small or too large size values will be discarded. Similarly aspect ratios out of boundary will cause the candidate sign to be eliminated. The constraints employed for each sign candidate for first group signs are;

- Candidate height and width should be in pre-defined range [1].
- Area of sign candidate has to be in pre-defined range [1].
- The rate of height and width should be in the interval  $[0.5, 2]$  [1].

The exact figures for the constraints are found through an exhaustive search and specified for each test set in Chapter 4 with the test results.

This part of the algorithm greatly reduces the number of the candidates which can not be a sign. After the application of the constraints, any group of pixels satisfying all of the constraints above, are considered as a true traffic sign. The bounding box (coordinates, height and width) corresponding to candidate is returned as a result.



### 3.1.2. Color Based Detection Algorithm II

The main goal of this algorithm is to detect danger, information, obligation and prohibition signs by using their color property. Danger and prohibition signs are characterized by a red border; obligation and information signs appear to be blue [2].

#### 3.1.2.1. Candidate Detection

In the detection process, a fuzzy detection algorithm rather than a linear one of relevant hue and saturation component is employed. Characteristic functions of the hue and saturation components are obtained in an exponential manner over the whole hue and saturation range separately [2].

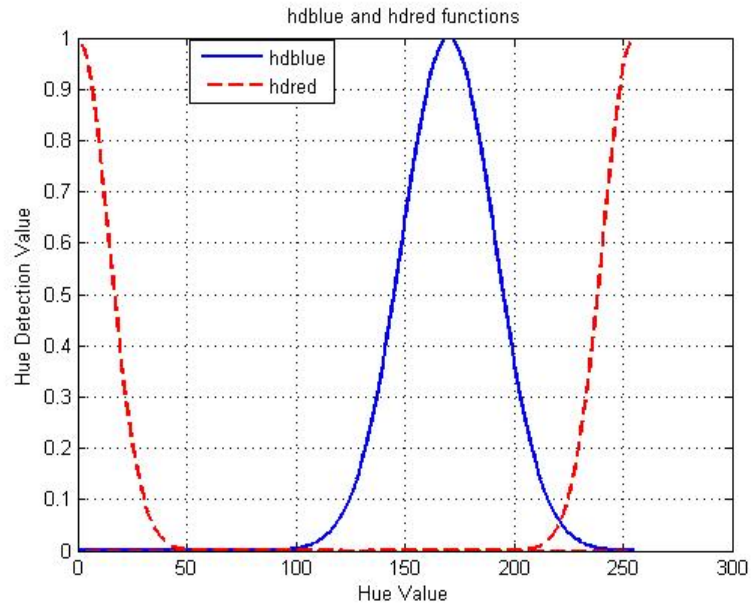
Unlike RGB color domain, since HSV color space enables evaluating color and intensity values separately, each image frame is converted to HSV domain. The color characteristics of the sign are analyzed by using the hue component. However, color information of an object becomes quite unreliable and so impractical if the saturation value it has is too low. Hence, in this algorithm not only the hue component but also the saturation value of the object is going to be used for detection purpose.

For each pixel in the image, hue based detection values ( $hd_{blue}$ ,  $hd_{red}$ ) for blue and red colors are calculated according to equations (3.11) and (3.12) respectively [2].

$$hd_{blue} = e^{\frac{-(x-170)^2}{30^2}} \quad (3.11)$$

$$hd_{red} = e^{\frac{-x^2}{20^2}} + e^{\frac{-(x-255)^2}{20^2}} \quad (3.12)$$

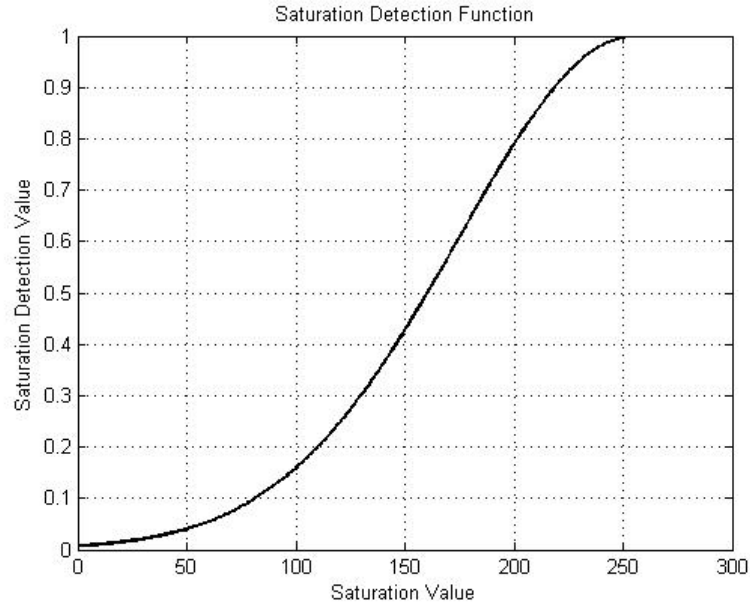
In the proposed algorithm the hue value has 8-bit depth so is considered to be in the range of [0-255]. Therefore, hue value for the blue pixels lays around 170. Values for red pixels are close to 0 and 255. By using the equation above,  $hd_{red}$  gives a value close to one on red regions. Similarly the function  $hd_{blue}$  tends to one over the blue regions of the image [2].  $hd_{red}$  and  $hd_{blue}$  function can be seen in Figure 3.7.



**Figure 3.7**  $hd_{red}$  &  $hd_{blue}$  functions

Similar to hue component, saturation component has 8-bit depth and has values between 0 and 255 as well. Saturation detection value is calculated according to (3.13) [2]. Function for detection of color saturated regions is seen in Figure 3.8.

$$sd = e^{\frac{-(x-255)^2}{115^2}} \quad (3.13)$$



**Figure 3.8** Function for detection of color saturation

In order to obtain an initial detection value for the red and blue regions over the whole frame, output of the  $hd_{red}$  and  $hd_{blue}$  functions are multiplied by the  $sd$  function respectively. This product is simply called the  $hs$  function. Multiplication results in very low values (both for red and blue components) for non-sign regions. Regions having very low values are discarded by setting their values to zero. This filtration enables reducing the number of candidates in a considerable amount.

Although the product  $hs$  gives a fair estimate, remaining non-zero parts are re-scaled using the equation (3.14) to enhance the importance of higher values and reduce that of lower ones.

$$hsn = \begin{cases} 0 & hs < 0.3 \\ \left( \frac{hs - 0.3}{0.7 - 0.3} \right)^2 & 0.3 \leq hs \leq 0.7 \\ 1 & hs \geq 0.7 \end{cases} \quad (3.14)$$

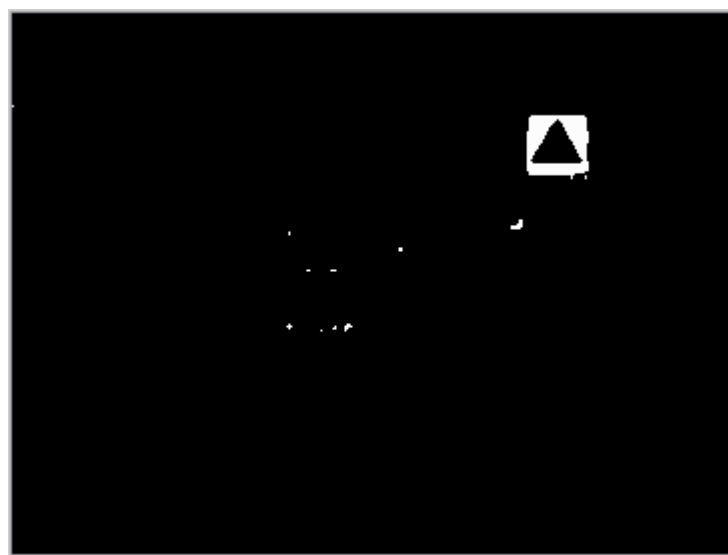
Below Figure 3.9 shows the original picture from which blue and red sign candidates are extracted.

Since same reasons in 3.1.1.3 about noisy pixels also apply for this algorithm, same 3x3 median filter is employed to eliminate these noisy pixels. Filtered blue and red sign candidates are seen in Figure 3.10 and in Figure 3.11 below respectively as well.

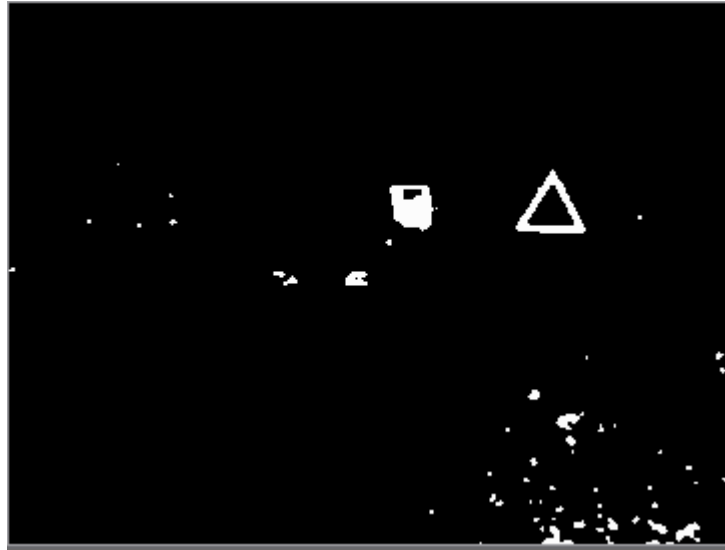
Similar to the previous algorithm, after the application of the median filter, in order to improve the binary image dilation with a 3x3 rectangular structuring elements is applied once.



**Figure 3.9** Original picture having red and blue traffic signs.



**Figure 3.10** Blue traffic sign candidates of picture in Figure 3.2



**Figure 3.11** Red traffic sign candidates of picture in Figure 3.2

The *hsn* is a temporary image which is used to detect the threshold value for the function *hs*. That is, the threshold value is determined by applying Otsu's algorithm [11] to temporary image *hsn*. Then, threshold found is used on the *hs* image forming a binary image, containing candidate sign regions [2]. Each connected components in this image are tested to decide whether they satisfy certain size and aspect ratio conditions.

### **3.1.2.2. Labeling Candidates & Checking Validity**

In some cases the detection algorithm returns too many candidate regions. Therefore, next step is to mark all connected regions as a candidate and check for some physical constraints. These constraints ensure possible sign candidates to satisfy some size and aspect ratio values. Candidates having too small or too large size values will be discarded. Similarly aspect ratios out of boundary will cause the candidate sign to be eliminated. The constraints employed for each sign candidate for first group signs are;

- Candidate height and width should be in pre-defined range [1].
- Area of sign candidate has to be in pre-defined range [1].
- The rate of height and width should be in the interval [0.5, 2] [1].

The exact figures for the constraints are found through an exhaustive search and specified for each test set in Chapter 4 with the test results.

This part of the algorithm greatly reduces the number of the candidates which can not be a sign. After the application of the constraints, any group of pixels satisfying all of the constraints above, are considered as a true traffic sign. The bounding box (coordinates, height and width) corresponding to candidate is returned as a result.

## **3.2. Shape Based Detection**

### **3.2.1. Detection of Circular Traffic Signs**

The proposed algorithm searches the image frame for circular or near circular objects. The shape of the sign plays a key role in the detection process. Assuming signs are in orthogonal direction to the road, the camera is in the direction of motion and the surface of signs will approximately be parallel to image plane [3], [5]. Although on rapidly curving road the traffic sign may not appear parallel to the image plane, it will get closer to be, when the car gets closer to sign [6].

The transformation here works on the gradient information of images. Hence, edge detection is accompanied to gathered frames. Then, algorithm searches the gradient information for a circle with an initial estimate of its radius. Algorithm is carried on a set of radii (around initial estimate) to locate sign candidates.

#### **3.2.1.1. Edge Detection**

First of all, the gradient image is calculated by convolving image frame with the 3x3 Sobel Operator. The reason for choosing the Sobel Operator for gradient calculation is its low computational complexity and fair performance in finding edges. The gradients in x and y directions are calculated separately for later purposes by using the 3x3 masks in equations (3.33), (3.34) respectively.

$$Grad_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (3.15)$$

$$Grad_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.16)$$

Insignificant gradient elements whose values are less than a predefined value (namely 120 -approximately 10% of the possible maximum gradient value-) are set to zero. Eliminating gradient elements whose values are small, increases the speed of the algorithm as it decreases the number of elements to check. Original image and the magnitude of the gradients obtained can be seen in Figure 3.12 and in Figure 3.13 respectively.



**Figure 3.12** Original image containing one circular traffic sign

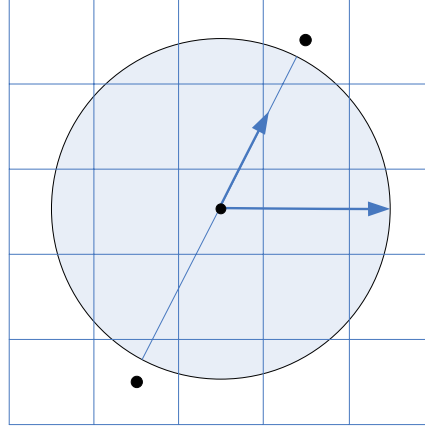


**Figure 3.13** Magnitude of gradients in x and y directions for image in Figure 3.10

### **3.2.1.2. Candidate Detection**

In the algorithm, remaining non zero gradient elements votes for a potential circle center a distance  $r$  (where  $r$  is the radius of the candidate circle) away along the line of the gradient (either having same direction or having opposite direction). For locating the center of the circle, an orientation projection image is calculated for each image frame by examining the gradient image. At each non-zero point of the gradient image corresponding positively affected and negatively affected pixels are determined as shown in Figure 3.14.





**Figure 3.14** Locations of positively and negatively affected pixels ( $n = 2$ ).

The positively-affected pixel is defined as the pixel which is  $n$  pixel away from the point  $p$  in the direction of gradient. Similarly, the negatively-affected pixel is the pixel which is  $n$  pixel away from point  $p$  in the opposite direction with gradient. [4] The coordinates of the positively-affected pixel are given by;

$$p_{+ve}(p) = p + \text{round} \left( \frac{g(p)}{\|g(p)\|} n \right) \quad (3.17)$$

and in a similar manner coordinates of the negatively-affected pixel are;

$$p_{-ve}(p) = p - \text{round} \left( \frac{g(p)}{\|g(p)\|} n \right) \quad (3.18)$$

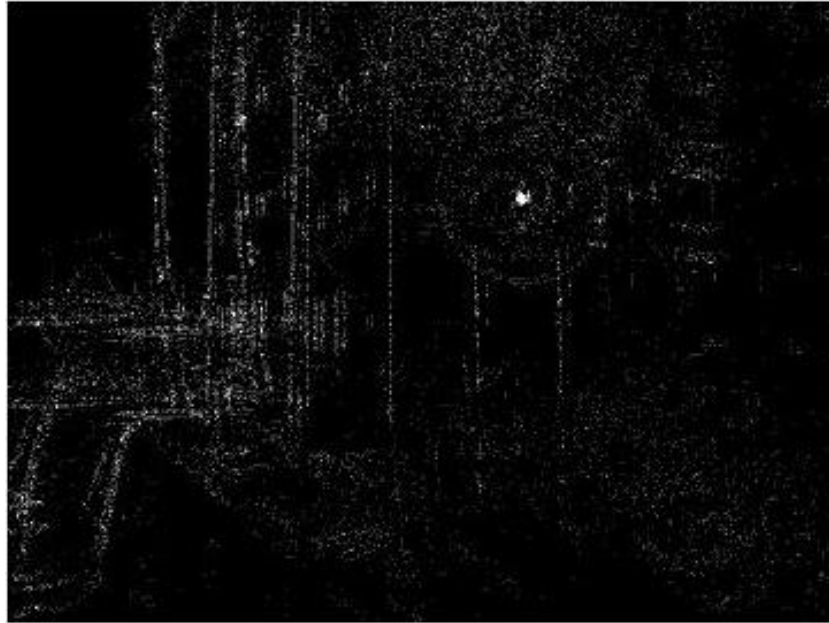
where  $g(p)$  represent the gradient vector at point  $p$ .

The orientation projection image  $O_n$  is initially zero. For the pixels whose gradient differs from zero coordinates of the positively and negatively affected pixels are calculated. Then the orientation projection image  $O_n$  is incremented by one at each positively-affected pixel. In addition to that,  $O_n$  is decremented at each negatively-affected pixel. That is,

$$O_n(p_{+ve}(p)) = O_n(p_{+ve}(p)) + 1 \quad (3.19)$$

$$O_n(p_{-ve}(p)) = O_n(p_{-ve}(p)) - 1 \quad (3.20)$$

$O_n$  for Figure 3.12 can be seen in Figure 3.15.



**Figure 3.15**  $O_n$  function for radius equals to 18 pixel.

The reason for calculating both positively and negatively affected pixel is the lack of information about the brightness of the image. In other words, we do not know whether the sign is lighter or darker than the background. Therefore, the direction of gradient vectors to the sign (not away from it) is not available. However, if such information is known it can be used. This, of course, results in an increase in the speed of the algorithm.

The radial symmetry contribution at radius  $n$  is defined as the convolution

$$S_n = F_n * A_n \quad (3.21)$$

where

$$F_n = \text{sgn}(\tilde{O}_n(p)) \left( \frac{|\tilde{O}_n(p)|}{k_n} \right)^\alpha \quad (3.22)$$

and

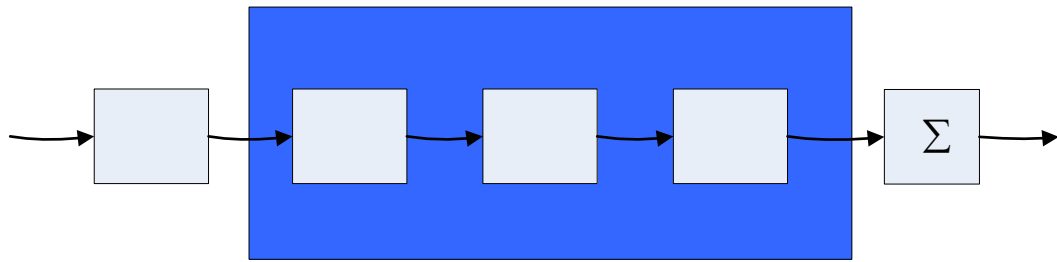
$$\tilde{O}_n(p) = \begin{cases} O_n(p) & \text{if } O_n(p) < k_n \\ k_n & \text{otherwise} \end{cases} \quad (3.23)$$

And  $A_n$  is the two dimensional Gaussian,  $\alpha$  is the radial strictness parameter  $k_n$ , is the scaling factor normalizing  $O_n$  for different radii.

The transform is calculated at one or more radii  $n \in N$ , where  $N$  is the set of radii of the radially symmetric features to be detected. The value of the transform at radius  $n$  indicates the contribution to radial symmetry of the gradients a distance  $n$  away from each point. In case of calculating the transformation for more than one radius value, the final transform is defined as average of the contributions over all the radii included,

$$S = \frac{1}{|N|} \sum_n S_n \quad (3.24)$$

Application of transform over a set of radii rather than a single value provides flexibility on the radius estimation. In other words, if the estimate differs from the actual one by several pixels, this will not abandon the algorithm. The whole transform is summarized with a block diagram in Figure 3.16.



**Figure 3.16** Block diagram showing the main steps of the process.

### 3.2.1.3. Choosing the Parameters

There are several parameters which play a very important role on the value derived by the transformation. Detection rate obtained from the transformation can be maximized by optimizing these parameters. These parameters can be summarized as,

- Set of Radii to calculate transformation
- The Gaussian Kernel  $A_n$
- The radial strictness parameter  $\alpha$
- The normalizing factor  $k_n$

Input  
Image

Determine  
Gradient

#### 3.2.1.3.1. Set of Radii

In order to reduce the processing time of the algorithm, it is applied only to speed signs whose radii are practical for in traffic images. For such a system, (while using in a driver assistance system) detecting circles with a small number of pixels will result in too many candidate regions for classification part of the entire system. Moreover, some candidates which have small number of pixel will have insufficient information for classification. In such situations waiting until the sign gets closer will be proper [4].

Similarly, in normal driving conditions a sign will possibly be detected before it becomes too large. This will enable us to bring an upper limit for the candidate detection.

Therefore, in the circular or near circular traffic sign detection process using continuous set of radii would be improper and unreasonable. In the algorithm the boundary for the set of radii is determined empirically and turns out to be in [16, 18].

### 3.2.1.3.2. Gaussian Kernel $A_n$

The purpose of the Gaussian kernel  $A_n$  is to spread the influence of the positively and negatively-affected pixels as a function of the radius  $n$  [3].

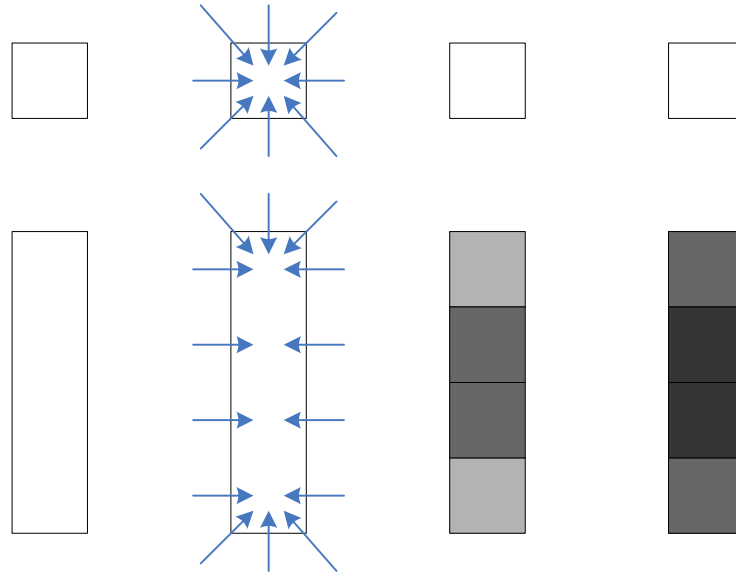
All  $A_n$  are defined as two-dimensional Gaussians whose elements sum to  $n$ . Convolution with  $A_n$  has the result of spreading the effect of each gradient element by an amount proportionate to the standard deviation of the Gaussian and amplifying its magnitude by  $n$ . Amplifying the magnitude is necessary to prevent the effect of gradient elements becoming negligible at large radii as a result of being spread too thinly across the image [3]. In the algorithm, Gaussian kernel is chosen as 2D Gaussian of size  $n \times n$  with,

$$\sigma = 0.25n \quad \text{and} \quad n = 7 \quad (3.25)$$

### 3.2.1.3.3. Radial Strictness Parameter

The radial strictness parameter  $\alpha$  determines how strictly radial, the output value of the transformation. In other words, the higher the radial strictness parameter, the higher attenuation on the non-radially symmetric features over the image. Figure 3.17 shows the effect of the parameter  $\alpha$  on pixel level. Decision of choosing radial strictness parameter becomes very important because higher  $\alpha$  attenuates non-radially symmetric components such as lines. However, a higher  $\alpha$  value causes elimination of some radially symmetric points.

In addition to this effect, since  $\alpha$  appears as an exponent in the formula the lower the value of  $\alpha$ , the lower computation time for the transformation. That is, choosing  $\alpha$  equals to 1 will minimize the computation time of the transformation.



**Figure 3.17** (i) Pixels on darker background. (ii) Gradient from adjacent pixels. (iii) Number of gradients pointing at each pixel  $O_n$ , (iv) square of the number of gradients pointing at each pixel  $O_n$

#### 3.2.1.3.4. Scaling Factor

The number of gradient elements affecting output value of the transformation becomes greater as the radius considered becomes larger [3]. Therefore in order to combine symmetry images with different radii, some kind of normalization has to be applied to bring every image into a similar scale. This normalization is achieved by dividing the orientation projection image  $O_n$  by scaling factor  $k_n$  [3]. The parameter  $k_n$  is determined experimentally and it turned out to be fifteen.

By applying this algorithm, center coordinates of the circles; whose radii are around the predefined value, in the image is obtained as a final output. Since an estimate of the radius is also available, necessary information is complete for detection.

### 3.2.2. Detection of Triangular, Rectangular and Hexagonal Traffic Signs

Detection algorithm to find regular polygons (triangles, octagons, and rectangles) is similar to the method used in circle shaped traffic signs. However, a radius of a

polygon is supposed to be defined. Therefore, a radius of regular polygon is defined as the perpendicular distance from edge to centroid.

### 3.2.2.1. Edge Detection

First of all, the image gathered is converted to gray scale for edge detection. After converting to gray scale, image is convolved with the Sobel Masks (both in x and y direction) given in (11). Resulting image contains the edge information of the input image in both x and y direction.

However, some of the edges whose magnitude is less than a specified value are better to be eliminated. The threshold value for this elimination is chosen to be 10% of the maximum possible value of the gradient elements obtained from experiments. The gradient elements below the threshold are set to zero. The remaining non-zero elements are normalized. By normalizing, unit vector in the direction of gradient is obtained.

$$\tilde{g}(p) = \frac{g(p)}{\|g(p)\|} \quad (3.26)$$

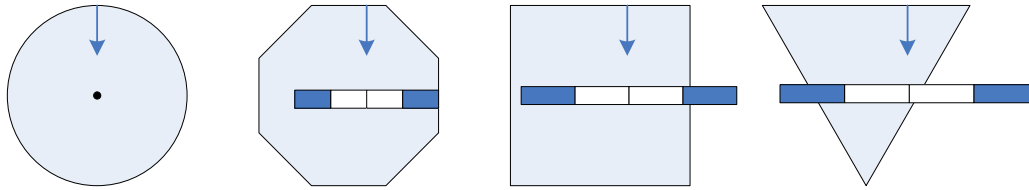
### 3.2.2.2. Candidate Detection

Each remaining non-zero gradient element votes for a potential polygon centre a distance  $r$  away (where  $r$  is the radius of the polygon being targeted) along the line of the gradient vector. The vote is placed at the closest pixel to this point. The points voted for are called affected pixels [5]. Positively and negatively affected pixels are defined by (12) and (13) respectively.

There are positive  $p_{+ve}$  and negative  $p_{-ve}$  affected pixels corresponding to points that the gradient points towards and away from respectively. Again, due to the absence of the brightness information of the sign, both  $p_{+ve}$  and  $p_{-ve}$  have to be calculated.

In contrast to algorithm used to detect circles in the image, a line of votes is cast describing possible shape centroid positions rather than gradient elements voting for

a single point [5]. Figure 3.18 shows different votes cast by a gradient element  $g(p)$  when searching for different shapes at a given radius (only the votes associated with the positively affected pixels are shown).

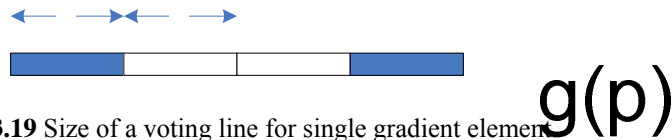


**Figure 3.18** Voting lines associated with the gradient element  $g(p)$  for different shapes.

Bar formed by the potential locations has four equal length parts. In Figure 3.19, parts receiving positive vote (lying in the middle) is shown in white. Dark parts of the bar (lying on the edges) however, are indicating the parts receiving negative vote. The width parameter  $w$  is chosen so that every point on a shape edge will cast a vote for the correct shape centroid, and therefore is given by [5]:

$$w = \text{round}\left(r \tan \frac{\pi}{n}\right) \quad (3.27)$$

where  $r$  is the radius and  $n$  the number of sides of the polygon being targeted. Size of a voting line for single gradient element can be seen in the Figure 3.19.



**Figure 3.19** Size of a voting line for single gradient element.

The line on which the affected pixels lie can be approximated by:

$$L(p, m) = p_{\pm ve}(p) + \text{round}(m\tilde{g}'(p)) \quad (3.28)$$



where  $\tilde{g}'(p)$  is the unit vector perpendicular to the gradient vector. Pixels receiving positive and negative vote are given respectively by relations (3.29) and (3.30).

$$L(p, m) \mid m \in [-w, w] \quad (3.29)$$

$$L(p, m) \mid m \in [-2w, -w-1] \cup [w+1, 2w] \quad (3.30)$$

Negative voting is introduced to reduce the effect of long straight lines. With the help of negative voting, the value of the line of possible centroid locations will decrease as we process to the both ends of the straight line. These voting lines are calculated and accumulated into an intermediate image  $O_r$  for each non-zero gradient element of the image.

Regular polygons are equiangular. In other words, their sides are separated by a regular angular spacing; for an  $n$ -sided polygon this is  $360/n$  degrees. To improve proposed detection algorithm of these shapes a rotationally invariant measure of how well a set of edges fits a particular angular spacing is introduced.

$\gamma$  is defined as angular values of the  $n$ -angle vectors:

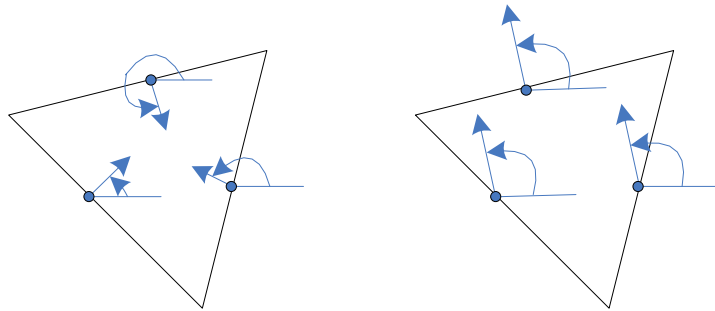
$$\gamma(x, y) = n\theta(x, y) \quad (3.31)$$

where  $\theta = \angle g$  is the gradient angle, and  $n$  is the number of sides of the target polygon.

Let  $v$  be the unit vector field such that  $\angle v(x, y) = \gamma(x, y)$ . For a given set of edge points  $p_i$ , the magnitude of the vector sum  $\sum_i v(p_i)$  indicates how well the set of edges  $g(p_i)$  fits the angular spacing defined by  $n$ .

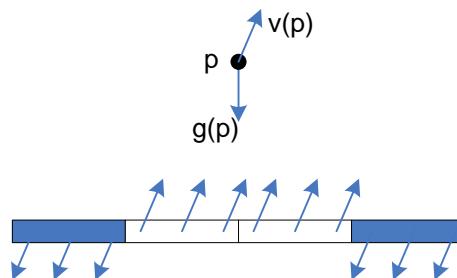
Figure 3.20 shows the formation of the  $n$ -angle gradient vectors for a triangle.  $P_1, P_2$  and  $P_3$  represents arbitrarily chosen three edge points from the sides of an equilateral triangle. The unit gradient vectors and their associated angles are shown in (a). Since the polygon is triangular by multiplying the gradient angles by  $n$  ( $n=3$  for a

triangle) the resulting vectors share the same direction if, and only if, their original orientations were spaced  $360/n$  degrees apart. Magnitude of the vector sum of the all  $n$ -angle gradient vectors become maximum if the polygon has the angular spacing we are looking for.



**Figure 3.20** Three edge points  $p_i$  on a triangle. (i) The angles of the unit gradient vectors. (ii) The resulting vectors obtained by multiplying the gradient angles by  $n = 3$ .

To utilize this result we construct a vector field of projected  $n$ -angle gradients by considering each non-zero element of  $g$  and projecting its associated  $n$ -angle vector onto its voting space as shown in Figure 3.21 (note the sign is reversed when projecting onto negatively voted pixels). Vectors projected onto the same pixel are summed.



**Figure 3.21**  $N$ -angle gradient projected from a point  $p$

The result is a vector field  $B_r$ , whose magnitude indicates how well the gradient elements voting on each point match the target angular spacing. Once the intermediate functions  $O_r$  and  $B_r$  have been calculated, final result for a radius generated by the algorithm is given by the equation:

$$S_r(p) = \frac{O_r(p) \|B_r(p)\|}{(2wr)^2} \quad (3.32)$$

The transformation can be applied to set of radii in order to find polygon of an unknown radius in image. The combined output of the function  $S$  is formed by summing up all the  $S_r$  values. Finally points with sufficient magnitude are described as centroid of a traffic sign candidate.

The entire process is summarized as below:

- Determine the gradient image. Set gradient values below threshold (approximately 10% of the maximum gradient value) to zero. Normalize the ones above threshold to unity.
- Determine n-angle gradient such that:

$$\|v\| = \|g\| \quad \text{and} \quad \angle v = \angle g \quad (3.33)$$

- For each non zero gradient element:
  - Determine vote location.
  - Accumulate the contribution to the vote image  $O_r$ , and the equiangular image  $B_r$ .
- Calculate the output image  $S_r$ .
- Sum  $S_r$  over all radii to determine the final output image  $S$ .

### **3.2.3. Detection Circular Traffic Signs by Using Hough Transform**

The Hough transform is a technique which can be used to isolate features of a particular shape within an image [32]. The classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses [31]. Because, it requires the desired features to be specified in some parametric form. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise [31]. This property makes Hough Transform valuable for traffic sign detection since occlusion is one of the important problems to be solved. The Circular Hough Transform (CHT) here works on the gradient information of images. Therefore, edge detection is applied to gathered frames. Then, algorithm searches the gradient information for a circle with an initial estimate of its radius.

#### **3.2.3.1. Edge Detection**

First of all, the gradient image is calculated by convolving image frame with the 3x3 Sobel Operator. The reason for choosing the Sobel Operator for gradient calculation is its low computational complexity and fair performance in finding edges. The Sobel Masks are given in equation in (3.15) and (3.16).

Insignificant gradient elements whose values are less than a predefined value (approximately 10% of the possible maximum gradient value) are set to zero. Eliminating gradient elements whose values are small increases the speed of the algorithm as it decreases the number of elements to check.

#### **3.2.3.2. Circular Hough Transform**

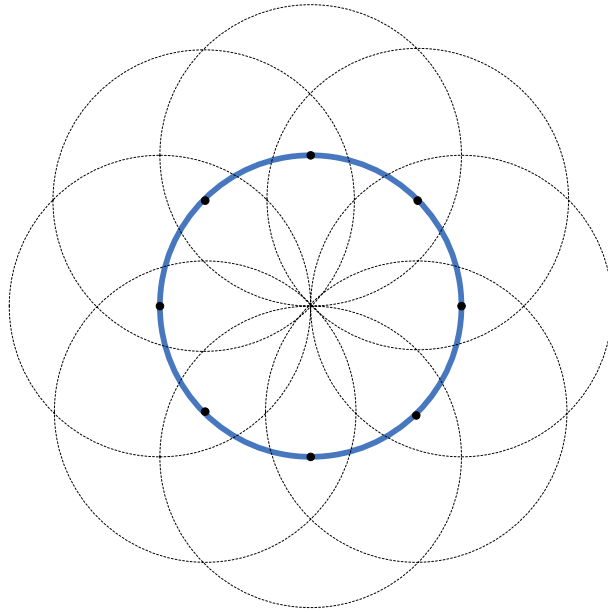
The circular Hough Transform can be described as a transformation of a center point of a circle in  $x$ - $y$  plane to the parameter space. The representation of a circle in  $x$ - $y$  plane is,

$$(x - a)^2 + (y - b)^2 = r^2 \quad (3.34)$$

hence, a circle is represented in x-y plane with 3 parameters which are  $a$ ,  $b$  the center coordinates and the  $r$  the radius. Thus, the parameter space for a circle will belong to  $R^3$  [33]. However in this thesis, in order to find circles in images radius parameter is kept constant. That is, the Circular Hough Transform is utilized for constant radius of traffic signs. Although this assumption causes an increase in false negative sign detection ratio for static images, it is not going to reduce the detection rate for video streams as the radius of the sign equals to a pre-defined value sooner or later.

After finding all edge points in an image, at each edge point a circle is drawn with center in the point with the desired radius [33]. This circle is drawn in the parameter space. At the coordinates which belong to the perimeter of the drawn circle the accumulator matrix which essentially has the same size as the parameter space is incremented by one. In this way every edge point in the input image is swept for circles with the pre-defined radii and the values of the accumulator is incremented [33]. When every edge point and every desired radius is used, the accumulator matrix has the all information about the original image. The accumulator will now contain numbers corresponding to the number of circles passing through the individual coordinates [33], [35]. Thus the maximum value or values correspond to the center of the circles in the image. In order to determine correct number of circle which image has a threshold is applied to values of accumulator. That is, if the maximum value of the accumulator is less than a specified value, then no circle is drawn.

Figure 3.22 shows the contribution of edge pixels to parameter space. The blue circle represents the original circle extracted by edge detection. The black dots indicate a set edge points within the image. Each edge point contributes a circle of radius  $R$  to an output accumulator space indicated by the dashed circles. The output of the accumulator matrix has a peak where these contributed circles overlap at the center of the original circle [35].

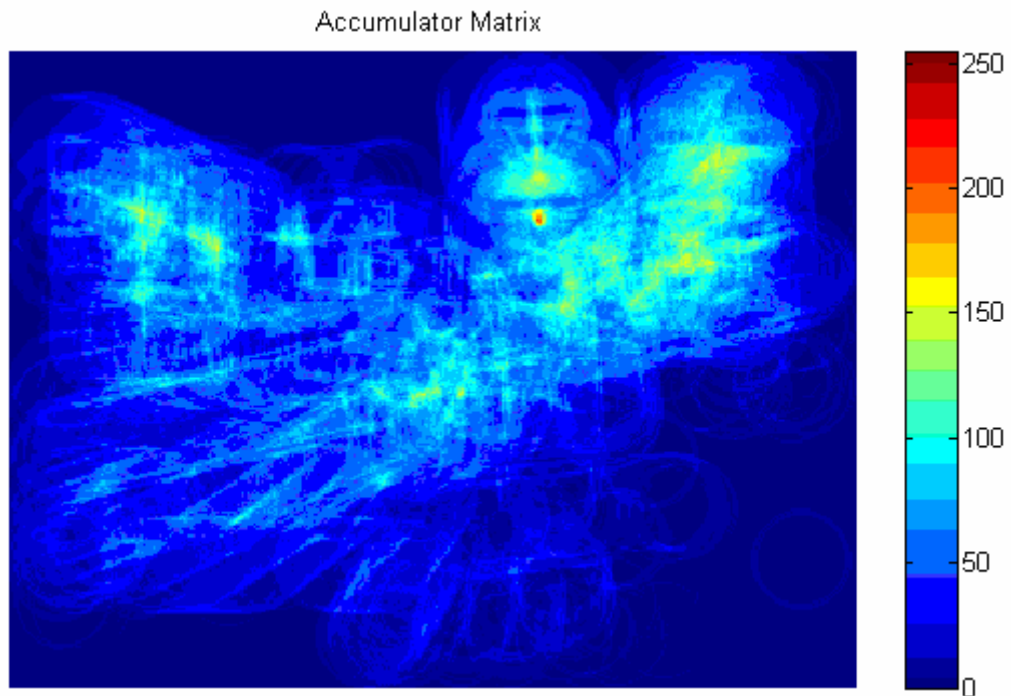


**Figure 3.22** The contribution of edge pixels in parameter space

For the original image in Figure 3.23 accumulator matrix is calculated and can be seen in Figure 3.24. The red dot shows the position of the maximum value of the matrix. Therefore it gives the coordinates of the center of the desired circle.



**Figure 3.23** Original image for which Accumulator Matrix is calculated



**Figure 3.24** Accumulator Matrix for Figure 3.23

Algorithm for CHT can be summarized as [34],

- Determine the gradient image. Set gradient values below threshold (approximately 10% of the maximum gradient value) to zero. Normalize the ones above threshold to unity.
- For each edge point in the gradient image
  - Draw a circle with a pre-defined radius to the parameter space.
  - Increment the value of the accumulator matrix by one for the boundary points of the circle drawn in the parameter space.
- Apply a threshold value to the accumulator.
- Find the local maxima in the accumulator.

### **3.3. Joint Treatment of Color and Shape Features for Detection**

In order to improve the performance of detection process a joint implementation of shape and color based algorithms is utilized. Two different methods have been used in order to combine these two features. Both of the algorithms help reducing the number of pixels to check whether they belong to a sign or not. This, of course, reduces the processing time of detection process.

#### **3.3.1. Combination of Color and Shape Detection by And Rule**

In this approach, first a pre-segmentation is employed with a thresholding operation by using the color based detection algorithm explained in section 3.1.1 in HSV color domain (Equations 3.9 and 3.10). Small variations from segmentation equation are introduced to improve detection rate. Namely, minimum saturation value required for red and blue candidates are reduced from 120 to 90 (over 255). By using these new values of saturation, a single joint image (rather than a one image for each color) for red and blue candidates is gathered. That is, by setting the pixels satisfying the new relations a binary mask for color features is obtained.

The second part the joint application is just composed of edge detection. For edge detection again, the one described in section 3.2.1.1 is used. By comparing the gradient values with a threshold value, the binary image for edges is obtained. However, once more to increase detection rate, the threshold value is reduced from 120 to 100.

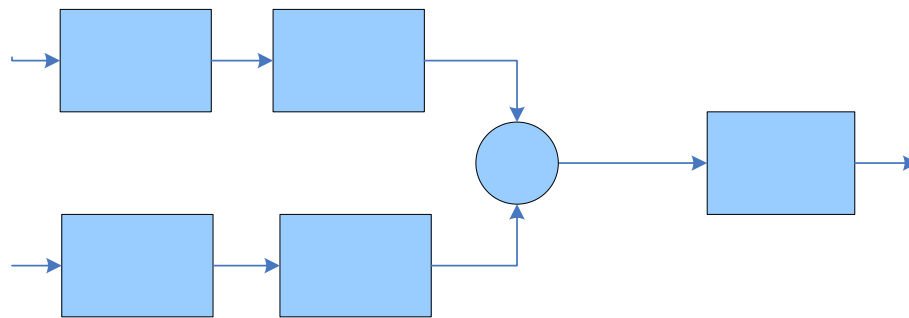
After both of the binary masks are obtained, per-pixel conjunction of them is found in order to join them. Equation below (3.35) explains this relation. The resulting binary mask is to be used as a domain for the algorithms explained in section 3.2.1 and in section 3.2.2. The combination of color and shape detection by And rule process can be summarized as in Figure 3.25.

$$I_{Joint} = I_{Color} \& I_{Edge} \quad (3.35)$$



where  $I_{Color}$  and  $I_{Edge}$  stands for binary images obtained from image segmentation and edge detection respectively.

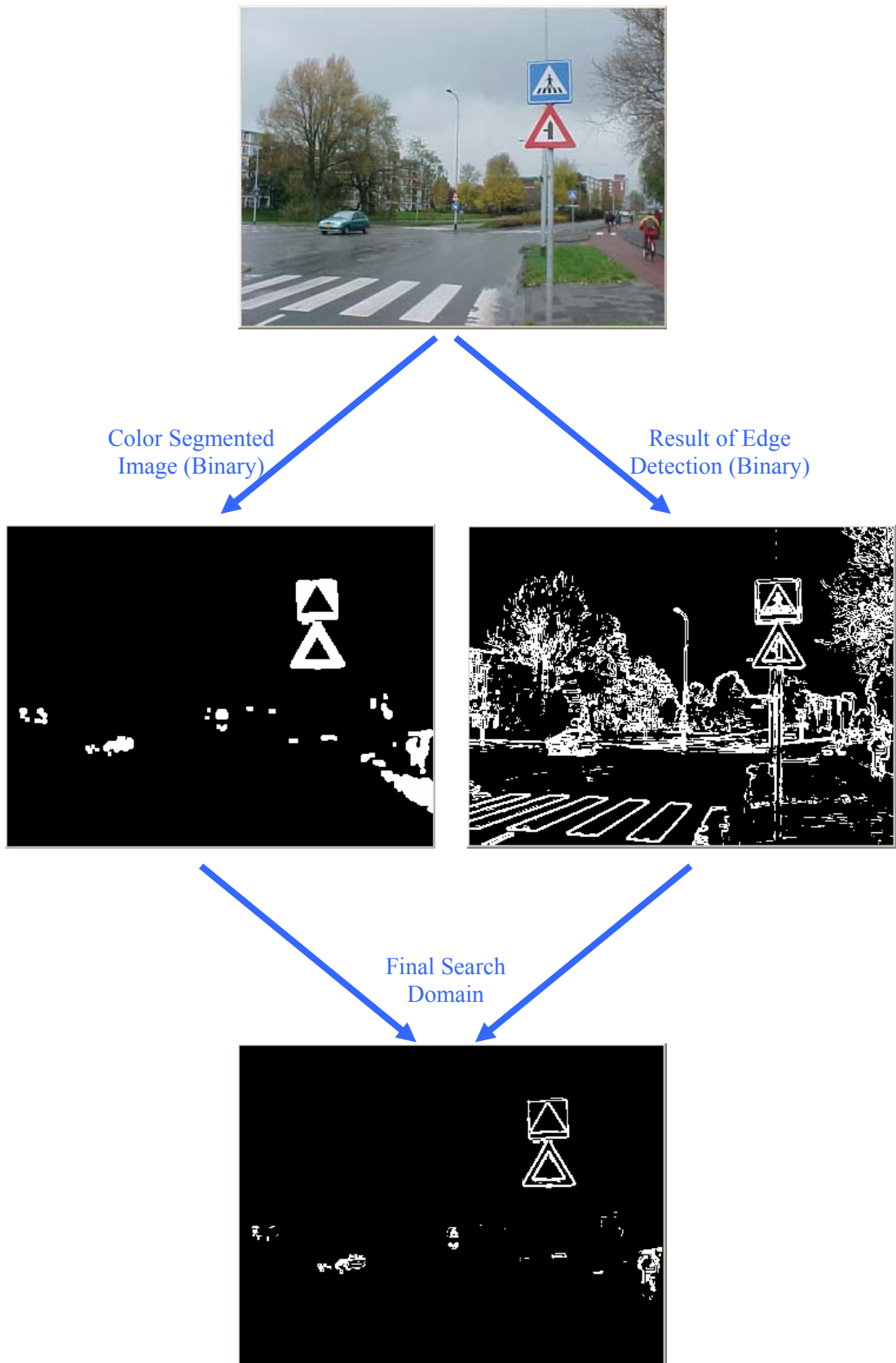
The influence of outdoor illumination, which varies constantly, can not be controlled [7]. Moreover, the paint on signs often deteriorates with age. Since the color segmentation algorithm in HSV color domain provides accurate and robust results, and since it is fast enough, it is chosen as the color segmentation algorithm.



**Figure 3.25** The combination of color and shape detection by And rule process

The most important drawback of this combination of color and shape detection by And rule is as follows. Regions, which have falsely been rejected by the color segmentation, cannot be recovered in the further processing [8]. The same thing applies for edge detection part as well. That is to say, for the detection process to be successful both of the arguments (both color segmentation and edge detection) should identify the traffic sign. Figure 3.26 specifies a sample for joint application.

Although the bounds for detection constraints are widened, since per-element bit-wise conjunction of two binary images is used, the domain in which we are looking for traffic signs has much smaller number of candidates. Figure 3.26 shows such a case. The binary image shown at the bottom of the page indicates the points on which the shape detection algorithm will work. Thus, smaller number of candidates results in shorter processing time for shape detection.



**Figure 3.26** Intermediate and final outputs of joining color and shape based detection process

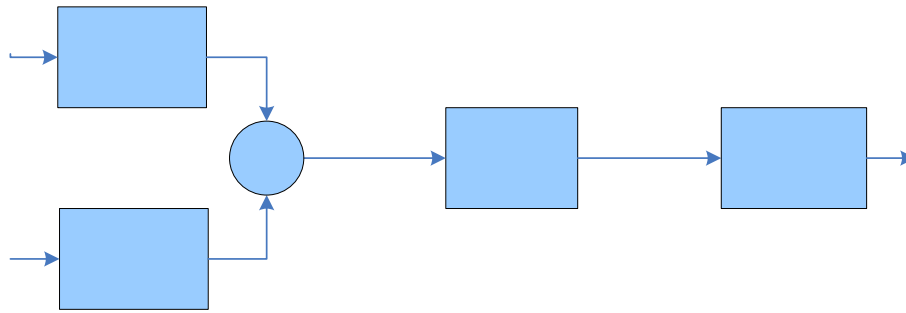
### **3.3.2. Combination of Color and Shape Detection by Product Rule**

Second approach applied to join color and shape based detections differs slightly from the previous one. Although, this method is also composed of one color segmentation and one edge detection phases, combination of their output is somehow different.

In this method, first a pre-segmentation is employed by using the color based detection algorithm explained in section 3.1.2 in HSV Color domain (Equations from 3.11 and 3.14). However unlike previous one, no threshold is applied to the result of the color segmentation algorithm. In other words, output of the color based algorithm is left as it is. Thus, it has the information of “how well” a color is, rather than whether it satisfies certain relation or not. Similar to the other one, single joint image (rather than a one image for each color) for red and blue candidates is gathered.

The second part the joint application consists of edge detection. For edge detection again, the one described in section 3.2.1.1 is used. However no comparison between gradient magnitude and a threshold value is done. Therefore, output of this step is merely the magnitude of the gradient.

Once both of these intermediate images are obtained, per-pixel multiplication of them is found in order to join them. After the multiplication, a proper threshold is applied. The resulting binary mask is to be used as a final search domain for the algorithms explained in section 3.2.1 and in section 3.2.2. The combination of color and shape detection by Product rule process can be summarized as in Figure 3.27.



**Figure 3.27** The combination of color and shape detection by Product rule process

The most important advantage of this method to the previous one is accuracy. Since thresholding is applied to the product image, any sign rejected by one of the methods can be recovered by the other one. That is, for instance, if edge detection (color segmentation) fails to detect a distinctive edge (color), this part of image may still be accepted as a true candidate if its color (edge) is “good enough”. An example of this situation can be seen in Figure 3.28. Though the lower part of the triangular traffic sign does not appear to produce a distinctive edge, it is included in the final search domain due to its dominant color value.

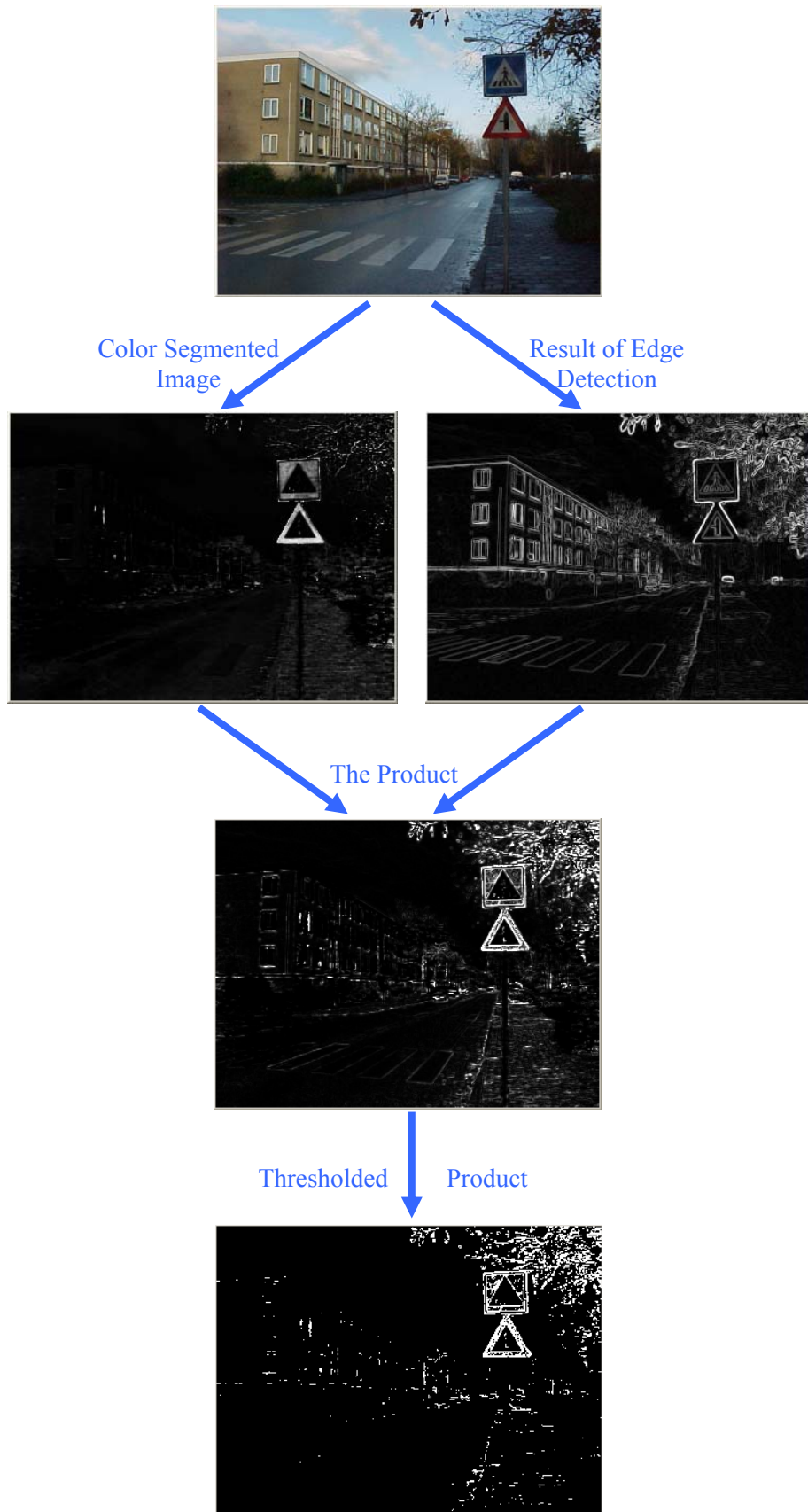
Again, the domain in which we are looking for traffic signs has much smaller number of candidates. The binary image shown at the bottom of the page indicates the points on which the shape detection algorithm will work. Thus, smaller number of candidates results in shorter processing time for shape detection.

Input  
Image

Color  
Segmentation  
in HS  
Domain

Input  
Image

Edge  
Detection  
using S  
Operator



**Figure 3.28** Intermediate and final outputs of joining color and shape based detection process

## CHAPTER 4

### EXPERIMENTS AND RESULTS

#### 4.1. Introduction

Algorithms applied for detection of traffic signs are developed on two different platforms which are:

- MATLAB
- Borland C++ by using OpenCV Libraries

Firstly, all of the algorithms are tested on static images separately. Static images used in tests of detection algorithms are mostly found from the internet. Tests are carried out over 337 images (containing more than 570 signs) where one or more traffic signs can be found in almost each frame.

In addition to test of separate algorithms with static images, joint application of algorithms (explained in section 3.3.1 and in 3.3.2) were tested with video streams as well as with static images. Video streams used in tests are obtained from [36]. Results of the joint application and separate application are compared by means of processing time and number of candidates.

Number of signs, number of detected signs, detection ratio and number of false positives for each test are shown in tables.

#### 4.2. Test Results on Separate Algorithms

There are three sets of static images. First set contains 16 circular, 21 rectangular, 19 triangular traffic signs (36 red signs, 20 blue signs) in 47 images with resolution 360x270. All of the images are well illuminated in Set-1.

Second set contains 89 circular, 33 rectangular, 8 triangular traffic signs (73 red signs, 17 blue signs) in 90 images with resolution 512x384. All of the images are

recorded during day-hours. Nonetheless, illumination of some of the image is greatly degraded by shadows. In addition to that, color of some signs in the Set-2 had changed due to the excessive exposure to sun light or some other weather conditions.

Set-3 consists of six groups of images all of which have different properties. These groups are:

- **Illumination:** In this group of 40, images are either not well illuminated or illumination causes detection process to be more difficult (i.e. sunlight directly coming just behind the traffic sign). This group contains 29 circular, 8 rectangular, 23 triangular and 1 octagonal traffic signs (50 red, 10 blue).
- **Occlusion:** Occlusion group contains 37 images. Traffic signs in this set of images are partially or heavily occluded by certain objects which makes detection process harder. This group contains 38 circular, 15 rectangular, 29 triangular and 2 octagonal traffic signs (62 red, 19 blue).
- **Translation:** For this set traffic signs are either too far or close to camera or they are located near the borders of the image. The signs near borders disable determining a region of interest for the search area. Translation group contains 33 circular, 18 rectangular, 16 triangular and 5 octagonal traffic signs (62 red, 19 blue) in 40 images.
- **Rotation:** In this set of images signs have lost their orientation. Some of them are rotated in image frame. That is, they are still perpendicular to the camera direction but they are dislocated. The remainings are somehow not perpendicular to the camera. This group of images consists of 14 circular, 13 rectangular, 11 triangular and 14 octagonal traffic signs (30 red, 9 blue). There are 30 images in this set.
- **Shadows:** Traffic signs in this group of 30 are mostly in shade. Irregular shadows on signs sometimes make edge detection process harder. This group of images consists of 29 circular, 15 rectangular, 36 triangular and 1 octagonal traffic signs (64 red, 13 blue).
- **Deformed:** Images involve deformed (broken or vandalized) signs. Deformed group has 23 images which have 17 circular, 7 rectangular, 15 triangular and 1 octagonal traffic signs (64 red, 13 blue).

All of the images in Set-3 are 640x427 in resolution and gathered from [36].

Since each set of static images is taken from a different source, each set has its own optical specs such as aspect ratio and zoom. Inevitably this results in a requirement of some small adjustments in some parameters of the detection algorithms. For instance, height, width and area parameters, used in eliminating some improper regions, are adjusted for each set of images. These parameters are given with test results.

In addition to that, in Set-2 and in Set-3 traffic signs may appear much closer or further than the remaining ones for some images. In contrast to video streams; in static images, traffic signs do not come closer to camera which causes a great variation in the dimensions of signs. That is, sizes of signs may vary in a wide (five to fifty pixels in height/width/radius) range. Excessively varying dimensions sometimes makes it impossible to put some constraints (as in Section 3.1.1.5). However, in video streams as sign approaches the camera, sooner or later it will reach a predefined dimension. Hence, a static dimension constraint would be specified without producing considerable false negative rate.

For Set-2 and Set-3 same algorithms are repeated over the sets for each algorithm with different parameters/constraints to obtain more realistic outputs. However, since the CHT is “radius sensitive”, for Set-1 only the CHT is re-tested with optimized constraint values.

First test of each algorithm is completed with constant set of parameters and constraints. Hence, detection rates are comparably lower. These values for each test are provided with the results.

Second test, however, is performed without these constant size constraints. Each traffic sign in images are considered independent from others and constraints.

#### **4.2.1.1. Test Results on Static Images in Set-1**

Test results on Set-1 by using color features show that, detection percentages are above %90. All three algorithms (Color based detection algorithm I, Color based



detection algorithm I in HSV domain and Color based detection algorithm II) provide very similar detection rates on test Set-1. Since all of the images in this set are well illuminated, algorithms using HSV color domain show no explicit superiority to one using RGB color domain.

However, when it comes to number of false positives algorithm in RGB domain finds much more false positive signs than the other two do. Especially when the red color is the concern, number of false positives approaches the half of the number of traffic signs. Algorithm in RGB domain hardly differentiates between the red traffic sign and colors close to red. High (%94.4 for red signs, %90 for blue signs) detection rate in this color domain is a result of “all-pass” characteristics of the algorithm.

Since the sizes of the traffic signs are equal or close to each other in Set-1 a single test with the constant size constraints (height, width, aspect ratio and area) is adequate to determine realistic results for color detection. Therefore no tests are performed by using optimized detection constraints.

Constraints used for the test of Set-1 can be specified as follows:

- Height and the width of the bounding box should be greater than 25 pixels.
- For the blue sign candidates, the rate of height to width of the bonding box should be in the interval [0.5 2]
- For the red sign candidates, the rate of height to width of the bonding box should be in the interval [0.55 1.9]
- For the blue sign candidates, area of the candidate should be greater than 18% and less than 95% of the area of the bounding box.
- For the red sign candidates, area of the candidate should be greater 15% and less than 70% of area of the bounding box.

Constraint values for the test of Set-1 are determined empirically.

Detection results and rates for algorithms using color properties are given from Table 4.1 to Table 4.3.

**Table 4.1** Detection results with algorithm Color I in RGB Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	36	34	94.4	17
<b>Blue</b>	20	18	90.0	5

**Table 4.2** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	36	35	97.2	3
<b>Blue</b>	20	19	95.0	4

**Table 4.3** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	36	35	97.2	1
<b>Blue</b>	20	19	95.0	1

When it comes to detection using shape features; for circular signs, algorithms found all of the signs correctly with no false positives. In addition to that, for rectangular signs detection rate is %95.2 with 2 false positives. For triangular signs however, although the number of false positive is only 1, detection rate is not as high as the rate for circular or rectangular ones. Comparably low detection rates are due to low gradient values caused by the shadows over the signs.

Without the radius value changed for each image, algorithm for detection of circular signs (explained section 3.1.2.1) gave much better results than CHT does. That is to say, since output of the CHT is much more dependent to the radius value, small variations in the radius value results in wrong outputs for detection.

Table 4.5 shows that CHT finds all of the circular traffic signs only when the radius for CHT is just the same as radius of the sign. However, for algorithm in described in Section 3.2.1 radius is not that determinative. This algorithm uses not only the radius value (like CHT) but also the direction value of the gradients to find the circular object. This property makes the algorithm provide less coincidental outputs. Therefore, for this algorithm, it is easier to determine whether a candidate is a true sign or not.

Radius of the shape is chosen as 20 pixels for the rectangular and circular signs. Since radius of a triangular sign is defined as the distance between the centroid and edge (in Section 3.2.2) radius is considered as 13 pixels for them. For edge detection gradient whose value is less than 120 is ignored. These values for the test of Set-1 are determined empirically.

Detection results for algorithms using shape features are in Table 4.4 and Table 4.5.

**Table 4.4** Detection results with shape algorithms

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	19	14	73.7	1
<b>Rectangular</b>	21	20	95.2	2
<b>Circular</b>	16	16	100.0	0
<b>Circular by CHT</b>	16	6	37.5	12

**Table 4.5** Detection results with shape algorithms (constraints optimized)

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	19	14	73.7	1
<b>Rectangular</b>	21	20	95.2	2
<b>Circular</b>	16	16	100.0	0
<b>Circular by CHT</b>	16	16	100.0	12

#### **4.2.1.2. Test Results on Static Images in Set-2**

Similar to results obtained from test of Set-1, detection of signs in RGB color domain produces plenty of (49 for red, 36 for signs) false positives. Especially number of false positives for blue images doubles the number of blue signs. This is because; according to the equations (2.25) and (2.26) the darker parts of the images are accepted as blue candidates.

Unlike results for Set-1, algorithms in HSV domain (Color based detection algorithm I in HSV domain and Color based detection algorithm II) provide significantly better results. One of the main reasons causing low detection rate in RGB domain is again high false positive rate. That is to say, false positive pixels (pixels satisfying equation (2.19), (2.25), (2.26) but not belonging to a traffic sign) are distorting the shape of the true candidates, if they are in the 8-neighbors of the pixels of true candidates.

Blue sign in Figure 4.1 can not be extracted because the blue candidates can not be isolated for the non-sign region. Therefore, bounding box for the sign candidates does not satisfy the constraints applied. This, of course, decreases the detection rate of the algorithm.

Another reason for low detection rate of algorithm in RGB domain is poor illumination. If the illumination of the sign in the image is not good enough, then the differentiation of color gets harder. That is, since the distance in color space among colors is small, small variation in figures affects the color too much.



**Figure 4.1** Original image having one red one blue traffic sign



**Figure 4.2** Blue candidates. Blue sign is discarded because it is not isolated from the non sign region.

Constraints used for the test of Set-2 can be specified as follows:

- Height and the width of the bounding box for blue candidates should be between 35 and 80 pixels.
- Height and the width of the bounding box for red candidates should be between 30 and 80 pixels.
- For the blue and red sign candidates, the rate of height to width of the bonding box should be in the interval  $[0.5 \ 2]$

- For the blue sign candidates, area of the candidate should be greater than 50% and less than 95% of the area of the bounding box.
- For the red sign candidates, area of the candidate should be greater 15% and less than 70% of area of the bounding box.

Radius of the shape is chosen as 19 pixels for the rectangular and circular signs. However, radius for triangular ones is chosen to be 12. For edge detection gradient whose value is less than 100 is ignored. These values for the test of Set-2 are determined empirically.

Detection results and rates for algorithms using color properties are given from Table 4.6 to Table 4.11.

**Table 4.6** Detection results with algorithm Color I in RGB Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	73	42	57.5	49
<b>Blue</b>	17	7	41.1	36

**Table 4.7** Detection results with algorithm Color I in RGB Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	73	54	74.0	49
<b>Blue</b>	17	9	53.0	36

**Table 4.8** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	73	55	75.3	11
<b>Blue</b>	17	12	70.6	5

**Table 4.9** Detection results with algorithm Color I in HSV Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	73	67	91.8	11
<b>Blue</b>	17	14	82.4	5

**Table 4.10** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	73	46	63.0	1
<b>Blue</b>	17	14	82.4	4

**Table 4.11** Detection results with algorithm Color II (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	73	58	79.4	1
<b>Blue</b>	17	14	82.4	4

Algorithm used to find circular objects provides noticeably better results than CHT does. Since this algorithm employs more information from the image, it provides better results.

Results of the rectangular sign detection algorithm happen to be lower than the other shape detection algorithms. Detection algorithms use the description of “radius” as the distance between the centroid and the edges for triangular, rectangular, and octagonal signs. This description works well for triangular and octagonal signs, because these shapes are equilateral. However, rectangular (not square) traffic signs are not equilateral. Therefore the definition of “radius” for a rectangle is ambiguous. That is, radius can be defined either from longer edges or from shorter ones. In each case for the detection, only three edges contribute to result rather the four. This

decreases the performance of the algorithm. This explains why the detection rate tends to be comparably lower.

Detection rates for algorithms using color features in HSV color domain produce similar results when compared to algorithms which use shape features. In other words, during day time when all colors are as obvious as edges, color properties become a solid alternative.

Detection results and rates for algorithms using shape properties are given in Table 4.12 and in Table 4.13.

**Table 4.12** Detection results with shape algorithms

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	8	6	75.0	2
<b>Rectangular</b>	33	14	42.4	1
<b>Circular</b>	89	55	61.8	3
<b>Circular by CHT</b>	89	12	13.5	4

**Table 4.13** Detection results with shape algorithms (constraints optimized)

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	8	7	87.5	2
<b>Rectangular</b>	33	26	78.8	1
<b>Circular</b>	89	79	88.8	5
<b>Circular by CHT</b>	89	61	68.5	4

### **4.2.1.3. Test Results on Static Images in Set-3**

Similar to the results obtained for rectangular sign detection in Set-2, detection rates are lower than that of other shape detection algorithms. This is because, the same reasons apply as in section 4.2.1.2.



For the algorithms using color properties constraints used for the test of Set-3 can be specified as follows:

- Height and the width of the bounding box for blue candidates should be between 25 and 100 pixels.
- Height and the width of the bounding box for red candidates should be between 26 and 100 pixels.
- For the blue sign candidates, the rate of height to width of the bounding box should be in the interval  $[0.5 \ 2]$
- For the red sign candidates, the rate of height to width of the bounding box should be in the interval  $[0.69 \ 1.45]$
- For the blue sign candidates, area of the candidate should be greater than 60% and less than 95% of the area of the bounding box.
- For the red sign candidates, area of the candidate should be greater 20% and less than 85% of area of the bounding box.
- For edge detection gradient whose value is less than 120 is ignored.

Constraint values for the test of Set-3 are determined empirically.

#### **4.2.1.3.1. Test Results on Illumination Group**

Color itself and illumination are two independent parameters in HSV color domain. This domain provides more successful results for signs under bad illumination. However, in RGB color domain, detection rates are lower, and the number of false positives is also higher.

Detection rates for all three algorithms decrease with the improper or low illumination conditions.

Detection results and rates for algorithms using color properties are given from Table 4.14 to Table.

**Table 4.14** Detection results with algorithm Color I in RGB Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	50	23	46.0	16
<b>Blue</b>	10	3	30.0	9

**Table 4.15** Detection results with algorithm Color I in RGB Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	50	29	58.0	16
<b>Blue</b>	10	7	70.0	9

**Table 4.16** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	50	26	52.0	1
<b>Blue</b>	10	3	30.0	0

**Table 4.17** Detection results with algorithm Color I in HSV Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	50	30	60.0	1
<b>Blue</b>	10	8	80.0	0

**Table 4.18** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	50	27	54.0	6
<b>Blue</b>	10	3	30.0	1

**Table 4.19** Detection results with algorithm Color II (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	50	38	76.0	6
<b>Blue</b>	10	8	80.0	1

Although improper illumination (when sun shines just behind the sign or the sign faces the sun) makes color segmentation harder, it has no such adverse effect on the edge information of the image. Therefore, detection rate of algorithms using shape properties does not diminish. Only in case of very low illumination these algorithms may fail. Detection results and rates for algorithms using shape properties are given in Table 4.20 and in Table 4.21.

**Table 4.20** Detection results with shape algorithms

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	23	5	21.7	2
<b>Rectangular</b>	8	4	50.0	1
<b>Circular</b>	29	10	34.5	8
<b>Octagonal</b>	1	0	0.0	0

**Table 4.21** Detection results with shape algorithms (constraints optimized)

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	23	22	95.7	2
<b>Rectangular</b>	8	6	75.0	1
<b>Circular</b>	29	26	89.7	8
<b>Octagonal</b>	1	1	100.0	0

#### 4.2.1.3.2. Test Results on Occlusion Group

As previously explained, it is hard to differentiate a sign from the background by using RGB color domain, hence; it produces too many false positive pixels. However, for images in Occlusion Group this argument unexpectedly comes in handy. In other words, some objects occluding traffic signs are accepted as a part of a sign by the algorithm. That is why; the detection rate in RGB domain is comparably higher than the other ones. Especially for blue signs the situation is more obvious (as in Figure 4.1 and Figure 4.2)

Detection results and rates for algorithms using color properties are given from Table 4.22 to Table 4.27.

**Table 4.22** Detection results with algorithm Color I in RGB Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	62	43	69.4	16
<b>Blue</b>	19	9	47.4	26

**Table 4.23** Detection results with algorithm Color I in RGB Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	62	50	80.6	16
<b>Blue</b>	19	15	78.9	26

**Table 4.24** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	62	38	61.3	6
<b>Blue</b>	19	3	15.8	4

**Table 4.25** Detection results with algorithm Color I in HSV (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	62	46	74.2	6
<b>Blue</b>	19	8	42.1	4

**Table 4.26** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	62	37	59.7	6
<b>Blue</b>	19	6	31.6	4

**Table 4.27** Detection results with algorithm Color II (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	62	45	72.6	6
<b>Blue</b>	19	11	57.9	4

When the sign is not fully occluded by an object, results of the shape detection algorithms are significant. The sign can be found even if some part of the sign does not contribute to the results, by using the same threshold over the final function given by (2.39) and (2.50).

Detection results and rates for algorithms using shape properties are given in Table 4.28 and in Table 4.29.

**Table 4.28** Detection results with shape algorithms

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	28	1	3.6	1
<b>Rectangular</b>	15	0	0.0	0
<b>Circular</b>	38	16	42.1	3
<b>Octagonal</b>	2	1	50.0	0

**Table 4.29** Detection results with shape algorithms (constraints optimized)

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	28	24	85.7	1
<b>Rectangular</b>	15	11	73.3	0
<b>Circular</b>	38	35	92.1	3
<b>Octagonal</b>	2	2	100.0	0

#### 4.2.1.3.3. Test Results on Translation Group

Detection results and rates for algorithms using color properties are given from Table 4.30 to Table 4.35.

**Table 4.30** Detection results with algorithm Color I in RGB Domain

Color	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Red</b>	48	29	60.4	33
<b>Blue</b>	14	8	57.1	12

**Table 4.31** Detection results with algorithm Color I in RGB Domain (constraints optimized)

Color	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Red</b>	48	38	79.2	33
<b>Blue</b>	14	12	85.7	12

**Table 4.32** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	48	33	68.8	4
<b>Blue</b>	14	9	64.3	0

**Table 4.33** Detection results with algorithm Color I in HSV Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	48	43	89.6	4
<b>Blue</b>	14	13	92.9	0

**Table 4.34** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	48	31	64.6	5
<b>Blue</b>	14	10	71.4	1

**Table 4.35** Detection results with algorithm Color II (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	48	41	85.4	5
<b>Blue</b>	14	13	92.9	1

Detection results and rates for algorithms using shape properties are given in Table 4.36 and in Table 4.37.

**Table 4.36** Detection results with shape algorithms

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	16	1	6.3	2
<b>Rectangular</b>	18	4	22.2	1
<b>Circular</b>	33	6	18.2	5
<b>Octagonal</b>	5	2	40.0	0

**Table 4.37** Detection results with shape algorithms (constraints optimized)

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	16	15	93.8	2
<b>Rectangular</b>	18	13	72.2	1
<b>Circular</b>	33	31	94.0	5
<b>Octagonal</b>	5	5	100.0	0

#### 4.2.1.3.4. Test Results on Rotation Group

Detection results and rates for algorithms using color properties are given from Table 4.38 to Table 4.43.

**Table 4.38** Detection results with algorithm Color I in RGB Domain

Color	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Red</b>	30	21	70.0	25
<b>Blue</b>	9	3	33.3	12

**Table 4.39** Detection results with algorithm Color I in RGB Domain (constraints optimized)

Color	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Red</b>	30	26	86.7	25
<b>Blue</b>	9	6	66.7	12



**Table 4.40** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	30	18	60.0	2
<b>Blue</b>	9	2	22.2	1

**Table 4.41** Detection results with algorithm Color I in HSV Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	30	22	73.3	2
<b>Blue</b>	9	5	55.6	1

**Table 4.42** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	30	16	53.3	2
<b>Blue</b>	9	3	33.3	0

**Table 4.43** Detection results with algorithm Color II (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	30	20	66.7	2
<b>Blue</b>	9	6	66.7	0

Since all of the algorithms used for sign detection by using shape features are rotationally invariant, results are not affected by such a deformation.

Detection results and rates for algorithms using shape properties are given in Table 4.44 and in Table 4.45.

**Table 4.44** Detection results with shape algorithms

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	11	2	18.2	1
<b>Rectangular</b>	13	3	23.1	0
<b>Circular</b>	14	4	28.6	5
<b>Octagonal</b>	14	3	21.4	0

**Table 4.45** Detection results with shape algorithms (constraints optimized)

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	11	9	81.9	1
<b>Rectangular</b>	13	11	84.6	0
<b>Circular</b>	14	11	78.6	5
<b>Octagonal</b>	14	13	92.9	0

#### 4.2.1.3.5. Test Results on Shadows Group

Partial shadows on signs do not reduce the detection rates of color algorithms much, because loss of some parts of binary sign candidates (as in Figure 3.5) does not change the shape of the bounding box of candidate. As long as no dark shadow appears over whole or most of the sign, algorithms using color responds well.

Detection results and rates for algorithms using color properties are given from Table 4.46 to Table 4.51.

**Table 4.46** Detection results with algorithm Color I in RGB Domain

Color	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Red</b>	64	39	61.0	23
<b>Blue</b>	13	8	61.5	7

**Table 4.47** Detection results with algorithm Color I in RGB Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	64	50	78.1	23
<b>Blue</b>	13	13	100.0	7

**Table 4.48** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	64	45	70.3	1
<b>Blue</b>	13	8	61.6	3

**Table 4.49** Detection results with algorithm Color I in HSV Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	64	61	95.3	1
<b>Blue</b>	13	10	76.9	3

**Table 4.50** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	64	39	60.9	4
<b>Blue</b>	13	7	53.8	2

**Table 4.51** Detection results with algorithm Color II (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	64	55	85.9	4
<b>Blue</b>	13	13	100.0	2

Since a small number of gradients are ignored (whose value is less than 10% of maximum gradient), traffic signs under shadow can still be detected. Although shadow reduces the gradient value of an edge pixel, it is still very likely to find signs by using shape detection algorithms. Moreover, edges resulting from the shadows on signs do not upset the algorithm since they are random in gradient direction.

Detection results and rates for algorithms using shape properties are given in Table 4.52 and in Table 4.53.

**Table 4.52** Detection results with shape algorithms

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	36	1	2.8	1
<b>Rectangular</b>	15	0	0.0	0
<b>Circular</b>	29	18	62.1	3
<b>Octagonal</b>	1	0	0.0	2

**Table 4.53** Detection results with shape algorithms (constraints optimized)

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	36	31	86.1	1
<b>Rectangular</b>	15	10	66.7	0
<b>Circular</b>	29	28	96.7	3
<b>Octagonal</b>	1	1	100.0	2

#### **4.2.1.3.6. Test Results on Deformed Group**

Small bents, spray paints or small deformations have no diminishing effect on detection rate neither for shape algorithms nor for color algorithms.

Detection results and rates for algorithms using color properties are given from Table 4.54 to Table 4.59.

**Table 4.54** Detection results with algorithm Color I in RGB Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	24	13	54.2	9
<b>Blue</b>	12	6	50.0	19

**Table 4.55** Detection results with algorithm Color I in RGB Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	24	21	87.5	9
<b>Blue</b>	12	11	91.2	19

**Table 4.56** Detection results with algorithm Color I in HSV Domain

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	24	14	58.3	2
<b>Blue</b>	12	5	41.7	6

**Table 4.57** Detection results with algorithm Color I in HSV Domain (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	24	22	91.7	2
<b>Blue</b>	12	10	83.3	6

**Table 4.58** Detection results with algorithm Color II

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	24	15	62.5	2
<b>Blue</b>	12	5	41.7	4

**Table 4.59** Detection results with algorithm Color II (constraints optimized)

<b>Color</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Red</b>	24	21	87.5	2
<b>Blue</b>	12	11	91.2	4

Since shape based detection algorithms use only the gradient information and since they are rotationally invariant, no dyed or rotated traffic sign causes an exception for it. Deformation has no effect on the detection rate, unless the sign is heavily bended or broken.

Detection results and rates for algorithms using shape properties are given in Table 4.60 and in Table 4.61.

**Table 4.60** Detection results with shape algorithms

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	15	7	46.7	0
<b>Rectangular</b>	7	0	0.0	0
<b>Circular</b>	17	4	23.5	5
<b>Octagonal</b>	1	0	0.0	0

**Table 4.61** Detection results with shape algorithms (constraints optimized)

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	15	14	93.3	0
<b>Rectangular</b>	7	5	71.4	0
<b>Circular</b>	17	15	88.2	5
<b>Octagonal</b>	1	1	100.0	0

### **4.3. Test Results on Joint Application of Color and Shape Based Algorithms**

Algorithms combining color and shape based detection are tested with both static images and video streams.

#### **4.3.1. Test Results on Static Images**

As explained in section 3.3, by combining two features of signs, the domain in which we are looking for traffic signs becomes much smaller in terms of the number of candidates. Tests carried on 25 randomly selected images from Set-3 produces the following results. Combination of detection algorithms by And rule (explained in 3.3.1) produces approximately 13 times smaller number of candidates. Depending on this figure, processing time required for shaped based algorithms (triangular, rectangular etc.) is reduced by nearly 10 times which brings a considerable improvement.

Combination of detection algorithms by And rule is also tested with the 40 images in Illumination Group. Test results show that, in spite of the great improvement in number of candidates, detection rates are not affected much. That is, reduction in processing time for shape algorithms does not result in a considerable fall in detection rates. Detection results and rates for algorithms using shape properties are given in Table 4.62. However, as explained before, regions rejected by one of the algorithms can not be recovered by the other one. Therefore, detection rate of triangular signs decreases (when compared to results in Table 4.21) from %95.7 to %87.

Since finding traffic signs in a frame requires both shape and color based algorithms to be successful, number of false positives is expected to decrease. Tests results satisfy the expectations. Total number of false positives reduces from 11 to 2. Especially for circular signs figures are distinctive. This difference mainly results from the candidates having curvature with appropriate radius but in incorrect color.

**Table 4.62** Detection results for combination of color and shape based algorithms by And rule

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	23	20	87.0	0
<b>Rectangular</b>	8	6	75.0	0
<b>Circular</b>	29	26	89.7	2
<b>Octagonal</b>	1	1	100.0	0

When it comes to combination of algorithms by Product rule, this method reduces the number of candidates by 85% when tested with same 25 images. This means, this application produces approximately 7 times smaller search domain. Corresponding time reduction is roughly 5 times. Search domains for each algorithm can be seen in Figure 4.3.

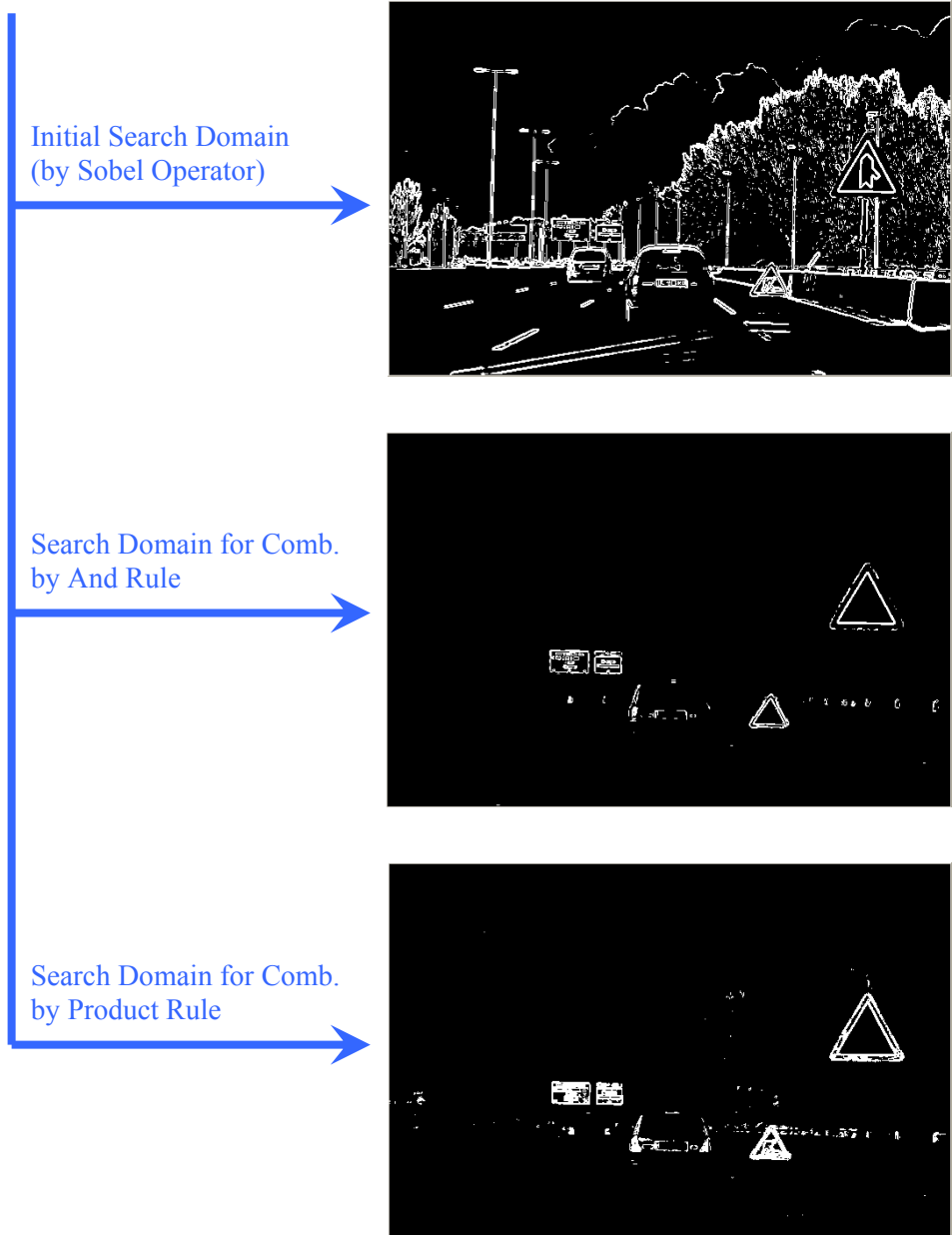
Combination of algorithms by Product rule is also tested with the same 40 images in Illumination Group. Tests resulted in a small increase (from %75 to %87.5) in detection rate of rectangular signs when compared to tests results in Table 4.21.

Increase in detection rate is due to the pixels having low gradient value but having the “proper” color. The combination by Product rule produces more candidate pixels and therefore is more time consuming. However, for static images having one of the features improper (shape or color), it produces better results. Similar to the combination by And rule, combination by Product rule produces much less false positives when compared to detection algorithm using merely shape features.

**Table 4.63** Detection results for combination of color and shape based algorithms by Product rule

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	23	22	95.7	0
<b>Rectangular</b>	8	7	87.5	0
<b>Circular</b>	29	26	89.7	4
<b>Octagonal</b>	1	1	100.0	0





**Figure 4.3** Search domains for each shape based algorithms

### 4.3.2. Test Results on Video Streams

Joint algorithms are also tested with video streams as well. In order to test both of the algorithms, four video streams gathered from [36] are used. Each video had been taken in different time intervals and for different whether conditions. Each video stream is 720x576 in resolution.

First stream (Video-1) which has 9 triangular and 9 circular signs to be detected is recorded in the evening before the sunset. Second one (Video-2) is taken at noon in a sunny day. This video contains 8 triangular and 9 circular signs. The third one (Video-3) having 10 circular and 5 triangular traffic signs is recorded at night on a road which is not illuminated. The last one (Video-4) is gathered during daytime on a rainy day. The last stream has 11 circular and 11 triangular signs.

Combination of detection algorithms by And rule (explained in section 3.3.1) are tested with these videos with the following parameters:

- Radius for triangular and circular signs 9 and 17 respectively
- Threshold value saturation of pixels is 90 (over 255)
- Hue value for blue color  $110 < Hue < 130$  (over 180)
- Hue value for red color  $170 < Hue \text{ and } Hue < 10$  (over 180)
- Threshold value for edge detection 100
- Threshold value for circular and triangular signs 0.2 and 0.025 respectively.

Detection results and rates for combination of algorithms by And rule are given from Table 4.64 to Table 4.67.

**Table 4.64** Results for combination of algorithms by And rule for Video-1

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
Triangular	9	8	88.9	1
Circular	9	9	100.0	0

Test on Video-1 shows that only one triangular sign is missed. Since the sign is not perpendicular to the direction of the car, its shape is a little bit distorted from an equilateral triangle. Therefore, the red border of the triangle seems thinner than it is. This causes loss of some candidate pixels for color detection. In addition to that, distortion in shape causes the detection value for this triangle to stay below threshold which is put to eliminate false positives.

**Table 4.65** Results for combination of algorithms by And rule for Video-2

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	8	8	100.0	0
<b>Circular</b>	9	9	100.0	0

At noon both edges and colors are so distinctive that joining algorithms do not miss any signs. Moreover, no illumination problem (camera facing the sun etc.) occurs at noon.

**Table 4.66** Results for combination of algorithms by And rule for Video-3

<b>Shape</b>	<b>Signs to be Detected</b>	<b>Detected Traffic Signs</b>	<b>% of Detection</b>	<b>False Positives</b>
<b>Triangular</b>	5	4	80.0	0
<b>Circular</b>	10	10	100.0	0

Video-3 was recorded during night, so illumination of signs is a matter of headlights of the cars. Therefore, as long as the headlights illuminate the sign, detection is possible. Hence, the reason for the missed triangle is poor illumination.

**Table 4.67** Results for combination of algorithms by And rule for Video-4

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
Triangular	11	9	81.8	0
Circular	11	11	100.0	0

Since the saturation value for pixels is lowered due to the rain, two of the triangular signs are missed. Too low (less than %25) saturation value for pixels causes the color detection algorithm to ignore these pixels. When one of the algorithms ignores, candidate is ignored. Therefore, low saturation value results in a miss. However, if the threshold value for saturation is reduced, detection algorithm would be affected badly in two different manners. Firstly, number of false positives increases. Since saturation is too low, colors can not be differentiated from each other. Some pixels are accepted by color detection algorithm although they are not blue or red. Secondly, number of noisy pixels increases and so processing time increases.

Combination of detection algorithms by Product rule (explained in section 3.3.2) are tested with these videos with the following parameters:

- Radius for triangular and circular signs 9 and 17 respectively
- Threshold value saturation of pixels is 90 (over 255)
- Hue value for blue color  $110 < Hue < 130$  (over 180)
- Hue value for red color  $170 < Hue \text{ and } Hue < 10$  (over 180)
- Threshold value for edge detection 100
- Threshold value for circular and triangular signs 0.2 and 0.025 respectively.

Furthermore, in order to obtain more accurate red results equation 3.12 is converted to,

$$hd_{red} = e^{\frac{-x^2}{14^2}} + e^{\frac{-(x-255)^2}{14^2}} \quad (4.1)$$

Results for combination by And rule are given from Table 4.68 to Table 4.71.

**Table 4.68** Results for combination of algorithms by Product rule for Video-1

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	8	8	100.0	2
<b>Circular</b>	9	9	100.0	0

Although combination by And rule misses one of the triangles, combination by Product rule detects this triangle. Loss in color domain is compensated by the high gradient value so candidates lost by one algorithm are recovered by the other one.

**Table 4.69** Results for combination of algorithms by Product rule for Video-2

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	8	7	87.5	1
<b>Circular</b>	9	9	100.0	0

**Table 4.70** Results for combination of algorithms by Product rule for Video-3

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	5	4	80.0	0
<b>Circular</b>	10	10	100.0	0

**Table 4.71** Results for combination of algorithms by Product rule for Video-4

Shape	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positives
<b>Triangular</b>	11	9	81.8	1
<b>Circular</b>	11	11	100.0	1

The same reason explained for the combination by And rule applies here as well. Decrease in saturation value results in losses.

## CHAPTER 5

### CONCLUSIONS AND FUTURE WORK

#### 5.1. Conclusions

In this thesis, detection of traffic signs was studied. Two different and main distinctive features -namely shape and color- of the signs are used for detection purpose. The applied algorithms are tested with static images and video streams. In order to provide more accurate and realistic results, images taken in different time intervals and different weather conditions are tested. Four different video streams and 337 images are used for testing.

Algorithms using color properties are implemented in two different color domains which are HSV and RGB. For each color algorithm, a binary mask is formed by setting pixels satisfying certain color conditions. Then a bounding box is drawn for all 8-neighbor connected pixels. Bounding box containing a traffic sign is determined by checking some size and aspect ratio constraints for each bounding box.

Detection rates for algorithm using color features in HSV color domain produces satisfactory results. When illumination of signs is good enough, RGB domain shows no major weakness when compared to HSV for detection. However, algorithm in RGB produces much more false positive, which may eventually affect the detection performance. Although low illumination values of traffic signs make things harder, during day time when all colors are obvious, color properties become a considerable alternative for detection.

When it comes to detection by using shape features; algorithms for circular, octagonal, rectangular and triangular signs are implemented. Sobel edge detector and binary threshold are used to extract edges. Once edges are extracted, both the magnitude and direction information of the edges are utilized. Detection rates for

shape detection algorithms are close to each other. Only difference arises for rectangular (not square) traffic signs. Even though the rectangular sign detection algorithm provides reasonable outputs, rectangular sign detection rate is lower than that of others. Since triangular shapes are infrequent in nature, number of false positives for triangular signs is lower than that for other shapes.

Circular Hough Transform is also implemented. Test result of this method shows that CHT is more dependent to the radius chosen. That is to say, small variations in the radius may result in big variations in the output. This property makes CHT harder to use for such a system.

In order to improve the performance of detection process, a joint implementation of shape and color based algorithms is also utilized. Two different methods have been used in order to combine these two features. First method uses per-pixel conjunction of color and shape detection results to form search domain for shape detection algorithms. However, the second one uses per-pixel multiplication of these results in order to form the search domain. Both of the algorithms help reducing the number of pixels to check whether they belong to a sign or not. This, of course, reduces the processing time of detection process.

Joint implementations of shape and color detection methods are tested with static images and video streams. Test with static images produce much less false positives when compared to separate application of algorithms. Video tests reveal that saturation value for colors and edges of an image play the key roles for detection. In other words, low saturation values and blurry images degrade the detection rates.

## **5.2. Future Work**

In order to improve the performance of the detection process several improvements can be implemented.

First would be defining a region of interest over the image for the detection. That is, instead of the whole image, a meaningful part of it can be searched. Since this reduces the number of pixel to search, it would shorten the computation time.

However, this would not be as easy as it seems. Because, sometimes the traffic signs are spread over the whole image. Especially signs which are placed temporarily (for a road work or etc.) may cause a problem.

Applying some tracking algorithm would be a good choice to improve performance of detection process. A simple tracking algorithm would prevent most of the false positives to occur. Hence, accuracy can be improved. However, since detection of signs is quite time critical, complexity of tracking algorithm should be kept low.

Another future work would be to determine the image statistics for setting pre-defined parameters. For during day time or at night, previously defined parameters would increase the accuracy and effectiveness of the algorithm. For example, gathering information about the brightness of a sign could decrease the computation time as explained in section 3.2.1.2.



## REFERENCES

- [1] V. Andrey, K. H. Jo, "Automatic Detection and Recognition of Traffic Signs using Geometric Structure Analysis" SICE-ICASE International Joint Conference 2006 Oct. 18-21, 2006 in Bexco, Busan Korea
- [2] C. F. Paulo, P. L. Correia, "Automatic Detection and Classification of Traffic Signs" Eight International Workshop on Image Analysis for Multimedia Interactive Services 2007
- [3] G. Loy, A. Zelinsky, "Fast Radial Symmetry for Detecting Points of Interest" IEEE Transactions on Pattern Analysis and Machine Intelligence" Vol. 25, No. 8, August 2003
- [4] N. Barnes, A. Zelinsky, "Real-time radial symmetry for speed sign detection" 2004 IEEE Intelligent Vehicle Symposium University of Parma, Italy. June 14-17 2004
- [5] G. Loy, N. Barnes, "Fast Shape-based Road Sign Detection for a Driver Assistance System" 2004 IEEE Intelligent Vehicle Symposium University of Parma, Italy. June 14-17 2004
- [6] "Road Sign Recognition Survey" [On-line] Available: <http://euler.fd.cvut.cz/research/rs2/files/skoda-rs-survey.html>, last accessed date: 4/11/2008
- [7] C.Y. Fang, S. W. Chen, C. S. Fuh, "Road-Sign Detection and Tracking" Vehicular Technology, IEEE Transactions on Volume 52, Issue 5, pp. 1329-1341, Sept. 2003.

- [8] C. Balmann, Y. Zhu, V. Ramesh, "A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information" Intelligent Vehicle Symposium, 2005 Proceedings, pp. 255-260, 2005
- [9] A. De la Escalera, L. E. Moreno, M. A. Salichs, J. M. Armigol, "Road Traffic Sign Detection and Classification" Industrial Electronics, IEEE Transactions on," Vol. 44, No. 6., pp. 848-859, 1997.
- [10]"Road Sign Recognition Survey" [On-line] Available: <http://euler.fd.cvut.cz/research/rs2/files/skoda-rs-survey.html>, last accessed date: 13/11/2008
- [11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms" IEEE Transactions On Systems, Man, And Cybernetics, Vol. SMC-9, NO. 1, January 1979
- [12] A. de la Escalera, J. Armingol, and M. Mata, "Traffic Sign Recognition and Analysis for Intelligent Vehicles" Image and Vision Comput., Vol. 21, pp. 247-258, 2003.
- [13] A. de la Escalera, J. M. Armingol, M. A. Salichs, "Traffic Sign Detection for Driver Support Systems" International Conference on Field and Service Robotics, 2001.
- [14] S. Kantawong, "Road Traffic Signs Detection and Classification for Blind Man Navigation System" International Conference on Control, Automation and Systems 2007 Oct. 17-20, 2007 in COEX, Seoul, Korea
- [15] W.J. Kuo, C.C. Lin "Two-Stage Road Sign Detection and Recognition" 1-4244-1017-7/07 IEEE, ICME 2007
- [16] S. Vitabile, G. Pollaccia, G. Pilato, F. Sorbello, "Road Signs Recognition Using a Dynamic Pixel Aggregation Technique in the HSV Color Space" Proceedings of the 11th International Conference on Image Analysis and Processing, pp. 572-577, 2001.

- [17] S. Vitabile, S. Gentile, F. Sorbello, "A Neural Network Based Automatic Road Signs Recognizer" Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, Vol. 3, pp. 2315-2320, 2002.
- [18] Y. B. Lauziere, D. Gingras and F. P. Ferrie, "A Model-Based Road Sign Identification System," in Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Vol.1,2001,ppI-1163- 1-1170.
- [19] K. S. Jr, S. L. Tanimoto, "Progressive Refinement of Raster Images" IEEE Transactions on Computers, pp. 871-874, Nov. 1979.
- [20] M. A. Garcia-Garrido, M. A. Sotelo, E. Martín-Gorostiza, "Fast Traffic Sign Detection and Recognition Under Changing Lighting Conditions" Proceedings of the IEEE ITSC 2006. 2006 IEEE Intelligent Transportation Systems Conference Toronto, Canada, September 17-20, 2006.
- [21] J. F. Canny, "A computational Approach to Edge Detection," IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679-698, Nov. 1986.
- [22] H. Freeman, "On the encoding of arbitrary geometric configurations" IRE Transactions on Electronic Computers, vol. 10, 1961.
- [23] Y. Liu, T. Ikenaga, S. Goto, "Geometrical, Physical and Text/Symbol Analysis Based Approach of Traffic Sign Detection System" Intelligent Vehicles Symposium 2006, June 13-15, 2006, Tokyo, Japan.
- [24] Y.B. Damavandi, K. Mohammadi, "Speed Limit Traffic Sign Detection & Recognition" Proceedings of the 2004 IEEE. Conference on Cybernetics an Intelligent Systems Singapore, 1-3 December, 2004.
- [25] D. Ghica, S.W. Lu, X. Yuan "Recognition of Traffic Signs by Artificial Neural Network" 1995 IEEE International Conf. on Neural Networks, Perth, Australia, vol.3, pp.1444-1449, Nov. 1995.

- [26] Y.Y. Nguwi, A.Z. Kouzani, “A Study on Automatic Recognition of Road Signs” 1-4244-0023-6/06 IEEE CIS 2006
- [27] Y. Liu, S. Goto, T. Ikenaga, “An MRF Model Based Algorithm of Triangular Shape Object Detection in Color Images” the International Conference on Intelligent Computing, pp. 423-432, 2005
- [28] G. Wu, W. Liu, X. Xie, Q. Wei, “A Shape Detection Method Based On The Radial Symmetry Nature and Direction-Discriminated Voting” 1-4244-1437-7/07 IEEE. ICIP 2007
- [29] S. Zhu, L. Liu, “Traffic Sign Recognition Based on Color Standardization” Information Acquisition, 2006 IEEE International Conference on Aug. 2006
- [30] P. Hough, “A Method and Means for Recognizing Complex Patterns”. US Patent, 3069654, 1962.
- [31] D. H. Ballard, “Generalizing the Hough Transform to Detect Arbitrary Shapes” 1980.
- [32] R. O. Duda, P. E. Hart “Use Of The Hough Transformation To Detect Lines And Curves In Pictures” Technical Note 36, Artificial Intelligence Center
- [33] S. J. K. Pedersen, “Circular Hough Transform” Aalborg University, Vision, Graphics, and Interactive Systems, November 2007
- [34] B. S. Morse, “Lecture 15: Segmentation (Edge Based, Hough Transform)” Brigham Young University, 1998–2000
- [35] M. Rizon, H. Yazid, P. Saad, A. Y. Md Shakaff, A. R. Saad, M. Sugisaka, S. Yaacob, R. Mamat, M. Karthigayan, “Object Detection using Circular Hough Transform” American Journal of Applied Sciences 2 (12): 1606-1609, 2005. 2005 Science Publications
- [36] Alcalá de Henares (Madrid, Spain) Universidad de Alcalá <http://roadanalysis.uah.es>, last accessed date: 14/11/2008

[37] J. Torresen, J. W. Bakke, L. Sekania, “Efficient recognition of speed limit signs” In Proc. IEEE Conf. on Intelligent Transportation Systems (ITS), Washington, DC, 2004.

[38] J. Miura, T. Kanda, and Y. Shirai, “An active vision system for real-time traffic sign recognition” In Proc. IEEE Conf. on Intelligent Transportation Systems (ITS), pages 52–57, Dearborn, MI, 2000.