# Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition

Hilario Gómez-Moreno, Member, IEEE, Saturnino Maldonado-Bascón, Senior Member, IEEE, Pedro Gil-Jiménez, and Sergio Lafuente-Arroyo

Abstract—This paper presents a quantitative comparison of several segmentation methods (including new ones) that have successfully been used in traffic sign recognition. The methods presented can be classified into color-space thresholding, edge detection, and chromatic/achromatic decomposition. Our support vector machine (SVM) segmentation method and speed enhancement using a lookup table (LUT) have also been tested. The best algorithm will be the one that yields the best global results throughout the whole recognition process, which comprises three stages: 1) segmentation; 2) detection; and 3) recognition. Thus, an evaluation method, which consists of applying the entire recognition system to a set of images with at least one traffic sign, is attempted while changing the segmentation method used. This way, it is possible to observe modifications in performance due to the kind of segmentation used. The results lead us to conclude that the best methods are those that are normalized with respect to illumination, such as RGB or Ohta Normalized, and there is no improvement in the use of Hue Saturation Intensity (HSI)-like spaces. In addition, an LUT with a reduction in the less-significant bits, such as that proposed here, improves speed while maintaining quality. SVMs used in color segmentation give good results, but some improvements are needed when applied to achromatic colors.

*Index Terms*—Detection, recognition, segmentation, support vector machines (SVMs), traffic sign.

#### I. INTRODUCTION

**R** ECENTLY, both automatic traffic sign detection and recognition have been the subject of many studies [1]–[10], although references to them have been appearing since 1990 [11]–[13]. Traffic sign recognition is important for driver-assistant systems, automatic vehicles, and inventory purposes. This paper is aimed at developing an inventory system [14], [15] to obtain a complete catalog of all the traffic signs on a given road and gather information about their state. Detection of red and blue traffic signs was considered for inventory purposes in [16]. However, that system focused solely on the position of the possible signs and not on their recognition.

Manuscript received April 16, 2009; revised October 23, 2009 and February 15, 2010; accepted June 16, 2010. Date of publication July 12, 2010; date of current version December 3, 2010. This work was supported by the Ministerio de Ciencia e Innovación de España under Project TEC2008-02077/TEC. The Associate Editor for this paper was M. M. Trivedi.

The authors are with the Departamento de Teoría de la Señal y Comunicaciones, Escuela Politécnica Superior, Universidad de Alcalá, 28805 Alcalá de Henares, Spain (e-mail: hilario.gomez@uah.es; saturnino.maldonado@uah.es; pedro.gil@uah.es; sergio.lafuente@uah.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2010.2054084

Traffic signs are normally classified according to their color and shape and should be designed and positioned in such a way that they can easily be noticed while driving. Inventory systems must take advantage of these characteristics. However, various questions need to be taken into account in an automatic traffic sign-recognition system. For example, the object's appearance in an image depends on several aspects, such as outdoor lighting conditions, camera settings, or the camera itself. In addition, deterioration of a traffic sign due to aging or vandalism affects its appearance, whereas the type of sheeting material used to make traffic signs may also cause variations. Finally, traffic sign images taken from a moving vehicle can suffer from blurring because of vehicle motion.

These problems particularly affect the segmentation step, which is usually the first stage in high-level detection and recognition systems. In this paper, the goal of segmentation was to extract the traffic sign from the background, as this is crucial to achieving good recognition results. Segmentation can be carried out using color information or structural information. Many segmentation methods have been reported in the literature since the advent of digital image processing. For example, in [17] and [18], extensive revisions of color-segmentation methods are presented. In [17], many color-segmentation methods are described and classified into different groups.

- Feature-space-based techniques: These are based on the color of each pixel with no consideration of the relationship to spatial information and include clustering, histogram thresholding techniques, and some neural network methods used only to classify colors.
- 2) Image-domain-based techniques: These aim to aggregate pixels in a region, using a measure of similarity based on color characteristics. These techniques include splitand-merge, region growing, edge detection, and neuralnetwork-based classification techniques that use color and space information.
- 3) *Physics-based techniques:* These techniques use physical models for the propagation and reflection of surfaces to look for the color regions in an image.

In [18], 150 references were presented on color segmentation. If the grayscale methods were to be included, this number would increase to about 1000 references, making an exhaustive study of the field beyond the scope of a paper such as this. Therefore, the discussion here focuses on the segmentation techniques previously used in traffic sign recognition, although some methods that have not been used before for this task but display good characteristics [such as our support vector machine (SVM) segmentation or reduced lookup tables (LUTs)] are also presented and discussed.

The segmentation methods used in earlier works about sign recognition employed different color spaces and techniques to separate the sign from the background. In [11] and [13], color normalization of the Red Green Blue (RGB) components with respect to the sum of those components was performed to detect strong colors in the image. A different relation between RGB components was used in [19], where the red component was used as a reference. Other studies used different color spaces rather than directly using RGB. The YUV color space was used in [20] to detect blue rectangular signs. Nevertheless, most researchers [14], [16], [21]–[24] have used Hue Saturation Intensity (HSI) family spaces, which focus on the hue and saturation components to prevent lighting dependencies and sometimes include intensity information to reduce hue and saturation instabilities. All previous methods can be classified among feature-space-based techniques.

Other algorithms used structural information based on edge detection, rather than color information. For example, a Laplacian filter with previous smoothing was used in [25] for grayscale images, grayscale images were also used with a Canny edge detector in [26], and a color image gradient was used in [7]. These are examples of image-domain-based techniques.

Several segmentation possibilities are thus available for the present study. The question is how to identify the best. In this case, the best segmentation method is considered to be that which gives the best recognition results. The criteria for good recognition results include high recognition rate, low number of lost signs, high speed, and low number of false alarms. To control these variables, a complete recognition system, which enables the segmentation procedure to be easily changed, was necessary. In addition, a set of images was needed to test the performance. For this study, more than 100000 images were obtained from different captured sequences while driving at normal road speed, i.e., with no disturbance to traffic. A database was constructed from the images, which were taken by different cameras using different settings and under different lighting conditions, with the camera positioned both inside and outside the car. However, not all the images have been used for this comparison. Relevant frames were extracted from several sequences identified as posing possible problems in the segmentation step. These frames have been made publicly available.

Although we focused on the Spanish traffic sign set, many of the properties are similar to those in other countries, at least in Europe, since traffic signs from European countries have similar pictograms to Spanish traffic signs, although some colors and legends are different.

This paper is organized as follows: Section II presents an overview of the entire traffic-recognition system used to measure the performance of each segmentation procedure. Section III presents the algorithms implemented for comparison purposes. Section IV describes some results and the method designed to measure them. Finally, in Section V, the results obtained are discussed to identify the best segmentation method.



Fig. 1. Block diagram of a traffic sign-recognition system.

## **II. SYSTEM OVERVIEW**

The traffic sign-recognition system that we have implemented, which was described in detail in [14], was used to evaluate segmentation algorithms presented in this paper. The system consists of four stages (see Fig. 1).

- 1) *Segmentation:* This stage extracts objects from the background, which are, in this case, traffic signs using color information.
- 2) *Detection:* Here, potential traffic signs are located through shape classification.
- Recognition: Traffic sign identification is effected using SVMs.
- 4) *Tracking:* This stage grouped multiple recognitions of the same traffic sign.

These stages have already been presented in [14], [15], and [27] and are now summarized for detection, recognition, and tracking.

## A. Shape Detection

The detection stage is described in detail in [27] and will now be briefly reviewed. This step uses the output masks obtained from the segmentation stage and gives the position and shape of any possible traffic sign. Blobs can be obtained by thresholding, color classification, or marking an edge into the image, as will be presented later in this paper.

The shapes considered are triangle, circle, rectangle, and semicircle. The octagon (from stop signs) was considered as a circle. Detection was carried out by comparing the signature of each blob with those obtained from the reference shapes.

To reduce projection distortions, each blob was normalized using the minimum inertia axis angle as a reference, and eccentricity was reduced through a method based on secondorder moments. Occlusion was overcome through interpolating the signature when the blob was opened.

To reduce scaling and rotation problems, absolute values of the discrete Fourier transform (DFT) were considered, and the total energy of the signature was normalized. The signature was sampled from 64 different angles, and the output of the

TABLE I NUMBER OF RECOGNIZED SIGNS BY TYPE

Туре	Number
Red Circular	61
Red Triangular	45
White Circular	110
White Triangular	44
White Rectangular	114
Blue Rectangular	97
Yellow Circular	47
Yellow Triangular	25
Yellow Rectangular	9
Total	552

DFT was compared with that of previously calculated models to determine to which shape category each blob belonged.

Finally, the blob was reoriented to a reference position to simplify the recognition stage, except in the case of circles, as they have no reference point.

## B. Recognition

This stage was described in detail in [14]; here, only the most important aspects are presented. Once the candidate blobs have been classified according to their shape, the recognition process is initiated. Rather than using the entire set of signs, the recognition task is divided into different colors and shapes. This way, a total of 552 traffic signs (see Table I) is reduced to a maximum of 114 signs per type, thus improving speed.

Training and testing are carried out according to the color and shape of each candidate region. Thus, to reduce the complexity of the problem, each candidate blob is only compared with those signs that have the same color and shape.

The recognition stage input is a normalized-size block of  $31 \times 31$  pixels in grayscale for every candidate object. The interior of the bounding box was therefore normalized to these dimensions, and only pixels of interest were taken into account, depending on the shape.

Different one-versus-all SVM classifiers (see the Appendix) with a Gaussian kernel were used. In the test phase, the traffic sign class with the highest SVM decision function output was assigned to each blob.

## C. Tracking

The tracking stage [15] identifies correspondences between recognized traffic signs to give a single output for each traffic sign in the sequence. If a newly detected traffic sign displays no correspondence with other previously detected signs, a new track process is initiated. The track data structure containing the objects to be tracked is then updated, taking into account the new information. This ensures that sequential detections from the same object were processed together to estimate the parameters of the object. At least two detections are required to consider the object as a traffic sign. Information such as position coordinates, size, color, type category, and the mean gray level of the region occupied by the object is evaluated to establish correspondences between traffic signs in different images.

 TABLE II

 Compared Segmentation Methods With Its Abbreviations

SEGMENTATION METHODS						
	CAD	Chromatic/Achromatic Index				
	RGB	RGB Differences				
Achromatic	RGBN	Normalized RGB Differences				
	SI	Saturation and Intensity				
	Ohta	Ohta Components				
	RGBNT	RGB Normalized Thresholding				
Color Space Thresholding	HST	Hue and Saturation Thresholding				
	HSET	Hue and Saturation Enhancement Thresholding				
	OST	Ohta Space Thresholding				
	GER	Gray-Scale Edge Removal				
Edge Detection	Canny	Canny Edge Removal				
	CER	Color Edge Removal				
Other	SVMC	Support Vector Machines Color Segmentation				
	LUT	Look up Table Speed Enhancement				

#### **III. SEGMENTATION ALGORITHMS**

This section describes the segmentation algorithms evaluated (see Table II). The implementation of these algorithms generates binary masks, thus enabling objects to be extracted from the background. One mask was obtained for each color of interest, i.e., red, blue, yellow, and white. However, some algorithms are unable to obtain all the masks needed for the system, and only one mask is obtained, which is then used with all the colors of interest. This is the case of edge-detection techniques.

At this point, a problem arises with white segmentation since this is not a chromatic color but an achromatic color. In some related works on traffic sign detection, white information is not considered, and only red or blue colors are used to detect signs. However, much information is lost when this approach is employed since a large number of signs incorporate white content, i.e., prohibition or danger signs. Furthermore, some color spaces, such as HSI, are unstable near achromatic colors and cannot directly be used over those pixels. To improve white segmentation and reduce color space problems in this study, chromatic/achromatic decomposition is carried out. Thus, only those pixels considered chromatic are classified into different colors, whereas for achromatic pixels, those over a given threshold of brightness are considered to be white. This idea, based on saturation and intensity values, was used in [28], but we adapt each color space to identify achromatic pixels. Thus, we distinguish between color-specific segmentation algorithms and those devoted only to chromatic/achromatic decomposition, although, in some cases, both are closely related.

# A. Color Space Thresholding (CST)

One extended color-segmentation technique consists of thresholding the components in a color space. The existing

TABLE III THRESHOLD VALUES FOR RGBNT, HST, AND OST METHODS

Method	Threshold Values
RGBN	ThR = 0.4, ThG = 0.3, ThB = 0.4, ThY = 0.85
HSI	$ThR_1 = 10, ThR_2 = 300, ThB_1 = 190, ThB_2 = 270, ThY_1 = 20, ThY_2 = 60, ThY_3 = 150$
Ohta	$ThR_1 = 0.024, ThR_2 = -0.027, ThB_1 = -0.04, ThB_2 = 0.082, ThY_1 = 0.071, ThY_2 = 0.027$

variations of this technique [18] are related to different spaces or different means to identify the thresholds. The election of the color space is a key point in this technique [29], and therefore, several of them are compared in this work. It is possible to obtain an automatic threshold based on the histogram of the image [30], but this approach identifies color regions, instead of a given color. Thus, the thresholds used for segmentation are established by looking for the desired colors in the space used. The distribution of these colors gives an idea of the thresholds needed, which are empirically set to obtain the best classification results. The empirical election of the thresholds cannot guarantee the best results; thus, we performed an exhaustive search around the empirical thresholds to validate them. This procedure and some results are shown in Section IV-C3. As previously stated, white classification was performed using the achromatic/chromatic techniques explained in Section III-B.

1) RGB Normalized Thresholding: The RGB space is one of the basic color spaces (such as XYZ) and, furthermore, is usually the initial space, as it is used by capture cameras and monitors. If simplicity of the segmentation process is the aim, the best choice will be the use of RGB with no transformation. However, the high correlation between the three color components and the effect of illumination changes on color information makes it difficult to find the correct thresholds in this space using empirical methods. One solution could be the use of a normalized version of RGB with respect to R + G + B, as in [11] and [13], which uses three normalized components called r, g, and b. This way, illumination changes have less effect on color; in addition, given that the sum of the new components is r + g + b = 1, only two components are needed to carry out classification as the other component can directly be obtained from these. In addition, thresholds are easily located in this space. However, some problems arise with this normalized space since, with low illumination (low RGB values), the transformation is unstable, and at near-zero values, noise is amplified.

The masks for each color in this space are obtained using the following expressions for each color mask:

$$\operatorname{Red}(i, j) = \begin{cases} \operatorname{True}, & \text{if } r(i, j) \geq ThR \\ & \text{and } g(i, j) \leq ThG \\ \operatorname{False}, & \text{otherwise} \end{cases}$$
$$\operatorname{Blue}(i, j) = \begin{cases} \operatorname{True}, & \text{if } b(i, j) \geq ThB \\ \operatorname{False}, & \text{otherwise} \end{cases}$$
$$\operatorname{Yellow}(i, j) = \begin{cases} \operatorname{True}, & \text{if } (r(i, j) + g(i, j)) \geq ThY \\ \operatorname{False}, & \text{otherwise.} \end{cases}$$
(1)

The threshold values used are shown in Table III.

2) *Hue and Saturation Thresholding (HST):* In [31], the HST technique was presented and generalized for red, blue, and

yellow. The HSI color space has two color components, i.e., hue and saturation, which are closely related to human perception and an illumination component that is close to brightness. Hue represents the dominant color value, and saturation represents the purity of color, with high values belonging to pure colors and low values belonging to colors containing a high mix of white. HSI components can be obtained from RGB [28]. The hue obtained *H* is within the interval [0, 360], and the saturation *S* is within [0, 255]. It can be seen that hue is undefined when saturation is null (grayscale with R = G = B) and that saturation is undefined when intensity is null.

The output masks for each color using hue/saturation thresholding are thus obtained as

$$\operatorname{Red}(i,j) = \begin{cases} \operatorname{True}, & \operatorname{if} H(i,j) \leq ThR_1 \\ & \operatorname{or} H(i,j) \geq ThR_2 \\ \operatorname{False}, & \operatorname{otherwise} \end{cases}$$
$$\operatorname{Blue}(i,j) = \begin{cases} \operatorname{True}, & \operatorname{if} H(i,j) \geq ThB_1 \\ & \operatorname{and} H(i,j) \leq ThB_2 \\ \operatorname{False}, & \operatorname{otherwise} \end{cases}$$
$$\operatorname{Yellow}(i,j) = \begin{cases} \operatorname{True}, & \operatorname{if} H(i,j) \geq ThY_1 \\ & \operatorname{and} H(i,j) \leq ThY_2 \\ & \operatorname{and} S(i,j) \geq ThY_3 \\ & \operatorname{False}, & \operatorname{otherwise}. \end{cases}$$
(2)

In [14], the thresholds were set after an analysis of the hue and saturation histogram of manually selected traffic sign parts, where the hue/saturation thresholding had been applied to extract the red, yellow, and blue components. Table III shows the threshold values employed in the evaluation presented in this paper since they are different from those used in [14], and saturation was used for yellow only.

This method is simple and almost immune to illumination changes since hue is used, but the main drawbacks include the instability of hue and the increase in processing time due to the RGB-to-HSI transformation.

3) Hue and Saturation Color Enhancement Thresholding (HSET): In [21], a different method for thresholding the hue and saturation components was presented. The distribution of hue and saturation in red and blue hand-segmented signs was studied, and subsequently, the values associated with each color are identified. To prevent the problems of a rigid threshold, a soft threshold based on the LUTs shown in Fig. 2 was used. This procedure was denominated "color enhancement" but in fact constitutes a soft threshold, where different values are assigned using linear functions, as shown in Fig. 2. The LUTs for the transformation of hue and saturation are described in this figure, where four parameters are used:  $h_{\min}$ ,  $h_{top}$ ,  $h_{max}$ , and  $s_{\min}$ . In all the cases, hue and saturation are normalized within the interval [0, 255].



Fig. 2. Color LUTs for the HSET method [21]. Color enhancement was achieved by using three LUTs for the hue and saturation components. Red uses the left LUT, blue and yellow use the central LUT, and saturation uses the right LUT. Hue and saturation were normalized in the interval [0, 255].

TABLE IV LUT AND THRESHOLD PARAMETERS FOR HSET

	$h_{\min}$	$h_{ m top}$	$h_{\max}$	$s_{\min}$	Thresholds
red	11	N.A.	224	23	28,160
blue	128	150	180	84	6,257
yellow	10	20	25	84	14,080

De la Escalera *et al.* [21] did not include yellow, whereas we have extended the method to incorporate this color by using the same enhancement as for blue (see Fig. 2) but employing different parameters.

The method can be summarized here.

- 1) Obtain hue and saturation from the RGB image.
- 2) Transform hue and saturation with two LUTs.
- Normalize the product of the transformed hue and saturation. This step was not performed in our study to increase speed.
- 4) Threshold the normalized product.

After the new hue and saturation values have been obtained, the product of both values can be performed, and the result was directly thresholded without normalization to reduce calculations and enhance speed. The different thresholds used for each color are shown in Table IV. The values used were obtained through modification of those from [21], as the product was not normalized, and the aim was to obtain the best empirical results. The algorithm for blue objects is extended using the LUT presented in [21], whereas that for yellow objects uses the results from [14].

4) Ohta Space Thresholding (OST): The search for an effective space in color segmentation has generated a large number of color features, such as XYZ, YIQ, Lab, and LUV. Among these spaces, that proposed by Ohta *et al.* [32] displays some desired characteristics, including simplicity and the fact that it can be used without high computational cost. Another characteristic is that this space is derived from trying to find the best uncorrelated components, thus making them almost independent. Furthermore, this space is included in the Opponent Color Spaces family, which is inspired by the physiology of the human visual system [28].

Based on extensive experiments [32] and the use of the Karhunen–Loeve transform, the author identified a set of three color features derived from RGB, which are effective for the segmentation of color images, i.e.,

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$
 (3)

As we can see, the  $I_1$  component is related to illumination; thus, only  $I_2$  and  $I_3$  are needed for color classification. Although these components can be used for direct classification, in this study, we use the normalization presented in [33], which reduces color variations due to illumination changes. The new normalized components  $P_1$  and  $P_2$  are given by

$$P_{1} = \frac{1}{\sqrt{2}} \frac{R-B}{R+G+B} = \frac{1}{3\sqrt{2}} \frac{I_{2}}{I_{1}}$$
$$P_{2} = \frac{1}{\sqrt{6}} \frac{2G-R-B}{R+G+B} = \frac{2}{3\sqrt{6}} \frac{I_{3}}{I_{1}}.$$
(4)

Using these normalized components, the colors can be classified as follows:

$$\operatorname{Red}(i,j) = \begin{cases} \operatorname{True}, & \text{if } P_1(i,j) \ge ThR_1 \\ & \text{and } P_2(i,j) \le ThR_2 \\ \operatorname{False}, & \text{otherwise} \end{cases}$$
$$\operatorname{Blue}(i,j) = \begin{cases} \operatorname{True}, & \text{if } P_1(i,j) \le ThB_1 \\ & \text{and } |P_2(i,j)| \le ThB_2 \\ \operatorname{False}, & \text{otherwise} \end{cases}$$
$$\operatorname{Yellow}(i,j) = \begin{cases} \operatorname{True}, & \text{if } P_1(i,j) \ge ThY_1 \\ & \text{and } |P_2(i,j)| \le ThY_2 \\ & \text{False}, & \text{otherwise.} \end{cases}$$
(5)

The threshold values used are shown in Table III.

#### B. Chromatic/Achromatic Decomposition

Chromatic/achromatic decomposition tries to find the image pixels with no color information, i.e., gray pixels. The grayscale is obtained when the R, G, and B values are equal; however, if these values are close rather than equal, the colors are perceived as being near gray. The methods presented extract gray pixels, and then, the brighter pixels are treated as white ones. All of the methods are different since each one is applied to different color spaces, but all of them are based on the idea of closeness of the R, G, and B components.

1) Chromatic/Achromatic Index: In [34], a decomposition for chromatic and achromatic pixels for the detection of white signs was presented. This method was used in [14] for such detection, together with hue/saturation thresholding. A chromatic/ achromatic index is thus defined as

$$CAD(R,G,B) = \frac{(|R-G| + |G-B| + |B-R|)}{3D}$$
(6)

where R, G, and B represent the color components for a given pixel, and D is the degree of extraction of an achromatic color. Accordingly, a pixel was considered achromatic when

$$Achr(i, j) = \begin{cases} True, & \text{if } CAD(i, j) \le 1\\ False, & \text{otherwise} \end{cases}$$
(7)

and white when

White
$$(i, j) = \begin{cases} \text{True}, & \text{if Achr}(i, j) = \text{True} \\ & \text{and } (R + G + B) \ge ThW \\ \text{False, otherwise.} \end{cases}$$
 (8)

Method	Thresholds Values
CAD	D = 30, ThW = 180
RGB Differences	$ThA_1 = 32, ThA_2 = 40, ThA_3 = 40, ThW = 180$
Normalized RGB Differences	ThA = 0.17, ThW = 180, ThL = 60
Achromatic SI	ThA = 48, ThW = 60, ThL = 60
Achromatic Ohta	$ThA_1 = 0.51, ThA_2 = 0.882, ThW = 60, ThL = 20$

TABLE V Threshold Values for Achromatic Methods

The threshold values for this method are shown in Table V.

2) *RGB Differences:* Although the previous index is useful, the use of a threshold to measure the difference between every pair of components is more realistic. Thus, we mark colors as being achromatic when the three differences between components are below a fixed threshold, which is different for each one. Accordingly, a pixel is considered achromatic when

$$\operatorname{Achr}(i,j) = \begin{cases} \operatorname{True}, & |R(i,j) - G(i,j)| \le ThA_1 \text{ and} \\ & |G(i,j) - B(i,j)| \le ThA_2 \text{ and} \\ & |B(i,j) - R(i,j)| \le ThA_3 \\ \end{cases}$$
(9)  
False, otherwise

and white when

White
$$(i, j) = \begin{cases} \text{True}, & \text{if Achr}(i, j) = \text{True} \\ & \text{and } (R + G + B) \ge ThW \\ & \text{False, otherwise.} \end{cases}$$
 (10)

The values for the thresholds are shown in Table V.

*3)* Normalized RGB Differences: In normalized RGB space, the achromatic pixels can be found in a similar way to that shown in the previous section. However, working in a normalized space requires only two differences, instead of three, since the third component can be obtained from the other two (see Section III-A1). Thus, the output image with each pixel marked as achromatic or chromatic is given by

$$\operatorname{Achr}(i,j) = \begin{cases} \operatorname{True}, & \text{if } |r(i,j) - g(i,j)| \le ThA \\ & \text{and } |r(i,j) - b(i,j)| \le ThA \\ & \text{False}, & \text{otherwise} \end{cases}$$
(11)

and white is obtained with

$$\text{White}(i, j) = \begin{cases} \text{True,} & \text{if Achr}(i, j) = \text{True} \\ & \text{and } (R + G + B) \ge ThW \quad (12) \\ \text{False,} & \text{otherwise.} \end{cases}$$

Since this is a normalized space, with near-low intensity values, instability exists, and to prevent this, when a pixel previously considered as chromatic has the sum of its RGB components below a given threshold (ThL), it is directly considered as black and thus achromatic. The threshold values are shown in Table V.

4) Saturation and Intensity: When HSI or similar spaces are employed, the achromatic detection presented in [28] can be used. This method is based on the fact that low saturation values mark pixels as achromatic since, with R, G, and B being equal (gray colors), saturation is null. However, not only zero saturation but low values are also considered achromatic. In

addition, hue is undefined for zero saturation and unstable for low intensity. Thus, the expression used is

$$Achr(i, j) = \begin{cases} True, & \text{if } S(i, j) \le ThA\\ False, & \text{otherwise.} \end{cases}$$
(13)

Intensity must be considered in two ways. Those pixels considered chromatic but with an intensity below a threshold called ThL are directly considered as black, thus preventing the instability of hue for low intensity. High values will be considered as white when a pixel is achromatic, i.e.,

White
$$(i, j) = \begin{cases} \text{True,} & \text{if Achr}(i, j) = \text{True} \\ & \text{and } I(i, j) \ge ThW \\ \text{False,} & \text{otherwise.} \end{cases}$$
 (14)

The threshold values are shown in Table V.

5) Ohta Components: In Ohta space, achromatic pixels can be located by looking at  $I_2$  and  $I_3$  components or normalized  $P_1$  and  $P_2$  [see (4)]. Low values for  $P_1$  and  $P_2$  are obtained when R, G, and B components are similar. Therefore, low values of  $P_1$  and  $P_2$  mark the achromatic pixels. Consequently, achromatic pixels are given by

$$\operatorname{Achr}(i,j) = \begin{cases} \operatorname{True}, & \text{if } |P_1(i,j)| \le ThA_1 \\ & \text{and } |P_2(i,j)| \le ThA_2 \\ & \text{False, otherwise} \end{cases}$$
(15)

and white pixels are given by

White
$$(i, j) = \begin{cases} \text{True,} & \text{if Achr}(i, j) = \text{True} \\ & \text{and } I_1(i, j) \ge ThW \\ \text{False,} & \text{otherwise.} \end{cases}$$
 (16)

For low values of  $I_1$ , components  $P_1$  and  $P_2$  are unstable; therefore, chromatic pixels below a ThL threshold are marked as black and, thus, are achromatic. The threshold values for this method are shown in Table V.

## C. Edge-Detection Techniques

Other extended segmentation techniques are based on the use of edge-detection algorithms to locate the different objects within an image. The idea is to mark the edge points as "no object," so that the points inside a closed edge are automatically marked as "object." With this method, color information is not needed, and problems of color spaces can be prevented. In [25], the authors reported that the use of the hue component showed two main problems.

- 1) There is a larger computational cost to obtain the hue from RGB.
- 2) The hue component can be affected by illumination changes, distance from the camera, or weather conditions.

TABLE VI PARAMETERS FOR CANNY EDGE DETECTION

	Sigma	Tlow	Thigh
Values	0.5	0.5	0.95

Therefore, they used only the brightness of the images to effect segmentation, using a Laplacian method.

In [26], the authors reported that, while color provides faster focusing on searching areas, precision was lower due to confusion of colors (particularly red and blue), segmentation problems in predominantly white signs, and changes in lighting conditions. Thus, methods based on shape analysis are more robust when changes in lighting occur. Therefore, the Canny method was used for edge detection since this method preserves closed outlines, which is a desirable characteristic in shapedetection systems.

One common problem with these methods is that, although they are simple and fast, they produce numerous candidate objects, which burden the detection and recognition steps with more work.

1) Grayscale Edge Removal (GER): This method was presented in [25] and comprises the classical two-step secondorder derivative (Laplacian) method: first, smoothing the image and, second, applying a Laplacian filter that performs a second derivative over the image to extract the edges. After this process, the result is an image called L(i, j).

Following this, the edge image is obtained by thresholding the results as follows:

$$O(i,j) = \begin{cases} \text{Edge}, & L(i,j) \ge T\\ \text{No-edge}, & L(i,j) < T \end{cases}$$
(17)

where T is the threshold, which is set as T = 3 as in [25].

2) Canny Edge Removal (Canny): The previous method used a very simple edge detector, and while implementation is fast, quality is not the best. Among the algorithms proposed for edge detection, the Canny edge-detection method [35] is commonly recognized [36] as a "standard method" used for comparison by many researchers. Canny edge detection uses linear filtering with a Gaussian kernel to smooth noise and then computes the edge strength and direction for each pixel in the smoothed image. After differentiation and nonmaximal suppression and thresholding, the edges are marked. This method tends to give connected shapes, and isolated points are minimal.

For this study, we used adaptive thresholds based on histograms of the image. The parameters used are given here.

- 1) Sigma or the parameter for the Gaussian kernel used.
- 2) The high threshold value was the (100 \* Thigh) percentage point in magnitude of the gradient histogram of all the pixels, which passed nonmaximal suppression.
- 3) The low threshold was calculated as a fraction of the computed high threshold value.

These parameters are shown in Table VI.

3) Color Edge Removal: The methods previously described do not use color information, but to take advantage of this information, an edge-extraction technique based on detection in the RGB color space is proposed. This method measures the distance between one pixel and its  $3 \times 3$  neighbors in the RGB

color space. The process starts with an edge detector applied over the color image, according to the following equation:

$$D_{ij} = \sum_{k=1}^{8} (R_{ij} - R_{ijk})^2 + (G_{ij} - G_{ijk})^2 + (B_{ij} - B_{ijk})^2$$
(18)

where  $R_{ij}$ ,  $G_{ij}$ , and  $B_{ij}$  are the red, green, and blue values of pixel ij, respectively, and  $R_{ijk}$ ,  $G_{ijk}$ , and  $B_{ijk}$  are the red, green, and blue values of the kth neighbor, respectively. The value obtained for each pixel is not the distance but the square of the distance to its neighbors; the square root is not necessary as it only increases the computational cost. After the final value  $D_{ij}$  is computed for each pixel, those pixels with values below a given threshold are considered as belonging to the foreground, whereas those above the threshold are considered as belonging to the edges separating the objects from the foreground.

#### D. SVM Color Segmentation (SVMC)

When CST is used, two problems are identified. There are a great number of thresholds to be adjusted, and the adjustment of these thresholds depends only on the images used, with no confirmation about generalization of the results obtained. Trying to reduce the work in parameter adjustment and obtain good generalization, a color-segmentation procedure based on SVMs is presented.

As can be seen, segmentation is a classification task where every pixel in the image is classified or labeled into several groups. Thus, segmentation can be carried out using any of the several well-known classification techniques. One of these is the SVM, which provides some improvements over other classification methods (see Appendix). SVMs yield a unique solution since the optimality problem is convex. This is an advantage, compared with neural networks, which have multiple solutions associated with local minima and, for this reason, may not be robust enough over different samples. In addition, this solution exhibits good generalization, and only a few parameters are needed to tune the learning machine.

In [37], an algorithm based on SVMs was presented to classify the pixels of an image using color information. Samples of the targeted color to be detected, in addition to other colors from training images, were labeled and used to train the SVM. The color space used was the RGB for simplicity, but other spaces could also be used. The parameters of the SVM were obtained with an exhaustive search by using tuning tools provided with the library LIBSVM [38]. The values obtained were  $\gamma = 0.0004$  and C = 1000 for all the colors.

One advantage of this segmentation method is that SVMs can be trained to find only those colors that are of interest to our application in an easy way, taking pixel samples from images. Obviously, to generalize to different sequences (different cameras or illuminations), the number of training vectors must be increased, but the generalization capacity of SVMs does not increase the number of support vectors in the same amount.

The main problem presented by this algorithm is its speed. Although the number of support vectors obtained is not high, the speed was lower than that of other segmentation algorithms. Because of this, it is not possible to directly use it in the recognition system of this study. However, this problem can be overcome using the LUTs proposed in Section III-E.

## E. Speed Enhancement Using a LUT

Sometimes, a good segmentation algorithm is available, but it cannot be used in a real application because of its slowness. Generally, algorithms that require color space transformations or complex calculations (such as SVM) present the problem of low speed. However, this can be prevented by making a precalculated lookup table to assign a color to each possible RGB value. With this approach, a table containing  $2^{24}$  positions is needed to accomplish with the whole equivalence. The number of operations is thus reduced (only one access to the table and an assignation are needed), but the table required is too long. Therefore, in the table mentioned, the RGB components were quantized, which gives a table with  $2^{18}$  positions, with 6 bits, instead of 8 bits, per component being used.

This quantization should not significantly affect detection performance, because only the two least significant bits of each component are eliminated. The reduction in computational time is significant enough to overlook this loss of information.

In the experiments carried out for the present study, this LUT is used with three particularly lengthy methods: 1) HST, which requires the computation of hue and saturation; 2) HSET, which computes a modified hue and saturation; and 3) the SVMC method, which requires the computation of several kernel distances to effect the classification. Thus, three LUTs were used: one for HST, one for HSET, and one for an SVMC that had been trained using images taken on sunny and rainy days.

## **IV. EXPERIMENTS**

## A. Traffic Sign Set

Many sequences on different routes and under different lighting conditions were captured. Each sequence included thousands of images. With the aim of analyzing the most problematic situations, we extracted several sets, which included those frames that were particularly problematic with regard to segmentation. Each set presented different segmentation problems, such as low illumination, rainy conditions, array of signs, similar background color, and occlusions. Table VII show details for each set.

Fig. 3 shows representative frames of such sets. For the results presented in this paper, a total of 313 images selected from thousands of  $800 \times 600$  pixel images were analyzed. These sets were considered representative of the main segmentation problems that may arise. Using the entire set of sequences obtained was not practical, as the total number of images involved was too high to carry out the inspection necessary to identify where a sign appears and how many signs there are in a sequence.

## B. Goal Evaluation

The problem that arises when we want to measure the performance of different segmentation methods is that there is no established method for carrying this out. There are many studies

TABLE VII SETS USED TO TEST THE SEGMENTATION ALGORITHMS (SETS CAN BE ACCESSED AND DOWNLOADED AT http://agamenon.tsc.uah.es/Segmentation)

Name	Characteristics	Num. of Frames	Sign Types
SET 1	Different segmentation colors and different shapes	36	1:♥ 2:♥ 3:♥ 4:▲
SET 2	Complex white signs	11	5:0
SET 3	Different segmentation colors and different shapes	27	6:♥ 7:● 8:▲
SET 4	Complex white sign	9	9:0
SET 5	Different shapes and an array of signs	29	10:▲ 11:▲ 12:@ 13:☺
SET 6	Signs with yellow back- ground	20	14: <b>&amp;</b> 15:😁
SET 7	Two signs with a near reddish background, and the signs were small com- pared to size of images	5	16: <mark>©</mark>
SET 8	Arrays of signs with red crossed ones	14	17:▽ 18: <b>⊘</b>
SET 9	Partially occluded sign	11	19:🗢
SET 10	Arrays and complex back- ground. Rainy day	10	20: <b>▲</b> 21: <b>1</b>
SET 11	Arrays and complex back- ground. Rainy day	17	22:🛆
SET 12	Different segmentation colors, some rotated signs	27	23:♥ 24:♥ 25:▲
SET 13	Red crossed signs which are complex to recognize	57	26: <b>☜</b> 27:●
SET 14	Cloudy day. Low illumi- nation.	40	28: 🛆 29: 😁

that measure segmentation performance [39]–[41], but none of them represents a standard. The main problem of these methods is that those with unsupervised measures are not good enough in specific scenarios, whereas supervised ones need an image segmentation background that must manually be constructed. Both types of methods suffer from an excessive execution time.

In this paper, we propose an evaluation method based on the performance of the whole recognition system. That is, we count the signs correctly recognized using different segmentation methods, whereas the rest of the system blocks (detection and recognition) remain unchanged. The images used in the tests were presented in the previous section. Evidently, following an inspection, the number and type of signs to be recognized in each set are known. The number of correctly recognized signs is not the only parameter that gives information about the performance of segmentation; however; speed or lost signs are also important parameters.

The parameters we evaluated are given here.

1) *Number of signs recognized:* For each individual sign, we counted the number of times it was correctly recognized.



Fig. 3. Some frames from each testing set. These frames show the most important signs in each set.

(One sign can be recognized more than once, depending on the colors it has.) However, instead of giving this number, we give a normalized version related to the maximum number of times that a sign can be detected. This way, a value near 1 is the best scenario since all possible signs have correctly been recognized. In addition, the sum of all the scores obtained (*Total score*) by every method is presented to show a unique performance measure. Since we used 29 different kinds of signs, the ideal score will be 29.

- Global rate of correct recognition: The sum of all correctly recognized signs was related to the number of signs that could globally be recognized. A value of 100 indicates that all possible signs in all sequences were correctly recognized.
- 3) *Number of lost signs*: This refers to the number of signs that were not recognized in any way.
- 4) *Number of maximum*: This parameter gives the number of times that a method achieved the maximum score.
- 5) *False recognition rate*: This figure represents the percentage of signs erroneously recognized by a method with respect to the number of total signs recognized.
- 6) *Speed*: Measure of execution time per frame. That is, the total execution time in seconds is divided between the number of frames used.

These parameters and all the tests were carried out using an automatic tool that compared the results obtained by every method with the known ground truth of all the sequences. This tool uses Matlab and bash commands to extract and process

TABLE VIII Results of the Different Achromatic Methods Studied. Each Row Represents a Sign With White Content in the Sets Studied

Sign	RGBN	Ohta	SI [28]	CAD [14]	RGB
1	0.72	0.44	0.78	0.00	0.67
4	0.67	0.67	0.44	0.44	0.00
5	0.14	0.14	0.14	0.14	0.14
6	1.00	0.93	0.93	0.13	0.60
8	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00
10	0.88	0.75	0.75	0.62	0.75
11	0.62	0.62	0.50	0.38	0.75
12	0.62	0.62	0.62	0.25	0.62
13	0.45	0.45	0.45	0.36	0.36
16	0.80	0.90	0.90	0.80	0.90
17	1.00	1.00	1.00	1.00	0.57
18	0.36	0.71	0.57	0.43	0.36
20	0.75	0.62	0.62	0.75	1.00
21	1.00	0.62	0.75	0.88	0.62
22	0.88	0.81	0.81	0.69	0.88
25	0.00	0.00	0.00	0.00	0.00
26	0.82	0.89	0.82	0.61	0.41
28	0.90	0.90	0.90	0.90	0.90
29	0.68	0.68	0.68	0.68	0.68

data. All the measures were obtained in a Linux environment with a 2.6.27 kernel.

Some example raw results<sup>1</sup> obtained for achromatic methods are presented in Table VIII, where each column represents the results obtained using a specific method, and each row of results corresponds to a sign in the sequence. The row number identifies a sign in Fig. 3. The best results for each sign are in bold type to facilitate interpretation.

# C. Results

1) Achromatic Decomposition Methods: First, it is necessary to ascertain whether the proposed achromatic decomposition methods are good enough and which of them are the best since, for some segmentation methods, no related achromatic decomposition exists, and a decision must be reached concerning the different options.

The data are presented in Table VIII. Since achromatic decomposition applies to white segmentation, only signs with white information are presented in the results. Additional information is given in Table IX.

Upon inspection of Table IX, it can be seen that the original CAD index used in [14] is not an option for obtaining the best results. The modified CAD index separating the three differences in RGB improves the results but does not produce the best performance. The method that achieved the best data for the total score, recognition percentage, number of lost signs, and maximum obtained is the RGB Normalized method, but it should not be concluded from this as the best since the Ohta and SI methods are not clearly worse. In false positives, the RGB

<sup>&</sup>lt;sup>1</sup>Result images including segmentation, detection, and recognition can be found at http://agamenon.tsc.uah.es/Segmentation.

Measures	RGBNT	Ohta	SI [28]	CAD [14]	RGB
Total score	12.29	11.78	11.69	9.07	10.22
Recognition (%)	65.66	64.53	64.15	47.55	51.32
Lost	3	3	3	4	4
Max	11	10	8	4	8
False (%)	0.00	0.00	0.00	1.56	2.86
Speed	0.1478	0.1356	0.1400	0.1130	0.1340

TABLE IX Analysis of the Results in Table VIII. See Section IV-B

Normalized, SI, and Ohta methods perform best with no false signs, but the other two methods get good scores. For speed, all methods are similar, but CAD achieves the best score.

Based on these data, we decided to use the achromatic RGB Normalized and Ohta methods in conjunction with its related color method and the SI achromatic method with color HST and HSET. Although we could have used the RGB Normalized method with the other methods, this would have implied the use of two color spaces, and as the results are similar, it was not considered to be the best option.

2) Color Segmentation Methods: In this section, the data obtained for color plus achromatic methods are presented. The data refer to all existing signs in the sets, including red, white, blue, and yellow data. Although raw data were obtained in a similar way to that presented in Table VIII, in this case, the most important information is summarized in Fig. 4(a) to improve data presentation.

It can be seen that the best methods are the CST methods. Edge-detection methods are not the best in all the cases, but for signs such as 5, which is only white (end of prohibition) and thus problematic for achromatic decomposition, they represent the best option. Of all these latter methods, Canny performs the best.

Among CST methods, although the RGB Normalized method obtains the best score for the total score and recognition percentage, the scores for other methods are not sufficiently different to justify discarding them. The best results for lost and the number of maximum are obtained by the LUT SVM method. The false percentage is very similar among CST methods, except for LUT SVM, and is excessive with edge detection methods.

In execution speed, the best scores are obtained by CST methods, whereas edge detection methods are significantly worse. Within CST methods, HST has the worst speed, al-though this could be improved by using a LUT. The RGB Normalized method and OST without a LUT show good speed behavior. The differences observed between LUTs are due to differences in the number of objects detected by each method; as the number increases, the detection and recognition steps take longer.

Speed enhancement using an LUT as presented in Section III-E improves speed and does not significantly reduce quality, as can be seen by a comparison of HST and LUT HST or HSET and LUT HSET.

3) Threshold Adjustment and Sensitivity: The results presented clearly depend on how the different parameters have been adjusted. Although the experiments were intended to get



Fig. 4. Results for the color segmentation methods. (a) Analysis of the results obtained for the image set presented in Section IV-A. In this case, the total score can reach 29. (b) Analysis of the results for the validation sets. The total score can reach 43. See Section IV-B for details about each measure presented.

the best results, it may be possible that better results could be reached with other parameters. Thus, a more complex adjustment method was implemented for those methods that are heavily dependent on thresholds, such as Achromatic or CST.

The method consists of two steps.

 Initial empirical adjustment: The first adjustment is started by looking for the thresholds in a graphical way, i.e., when the thresholds related to the red color are to be adjusted, an image with red signs is chosen, and the thresholds are set until a good visual segmentation for this color is obtained. After all the colors have been adjusted this way, recognition results are obtained using all the frames in every set. Then, the thresholds are refined using the recognition results in a "trial-and-error" procedure.



Fig. 5. Variation of the recognition performance. Examples for the RGBN color space and thresholds ThA, ThW, ThL, and ThR. The graphs have two ordinate axes since each plot has different scales. The left ordinate axis is for the recognition rate, and the right ordinate axis is for the lost and false rates.

2) *Exhaustive search*: In this second step, a sweep around the empirical threshold values is performed. Recognition results are obtained for multiple values for one of the thresholds while keeping the others unchanged. This way, the recognition performance evaluation function is obtained for each threshold. Plotting these results allows a simple visual inspection to find the best threshold value.

Fig. 5 shows some examples of the graphs obtained with the exhaustive search procedure. These examples correspond to RGBN space, and the evolution of recognition percentage, lost signs, and false percentage with respect to thresholds ThA, ThW, ThL, and ThR is plotted.<sup>2</sup> Three performance measurements are plotted together since a tradeoff between false/correct detection and signs lost is desired.

Looking, for example, to the plot of ThA evolution, it can be seen that ThA = 0.17 gives good recognition percentage, with only three signs lost and no false detection. However, the best recognition percentage is obtained for ThA = 0.13 but with some false percentage and one additional sign lost. In this case, the initial empirical adjustment gave ThA = 0.14, i.e., close to the optimum but not the best. Thus, this threshold was modified to the optimum value. The inspection of every graph for the different color spaces used gives similar information, and although sometimes the initial empirical adjustment was modified, most of them were maintained. 4) Validation: The data presented in previous sections are significant enough to generate some conclusions about what segmentation method should be used for sign recognition. However, since it was necessary to adjust the parameters for each method to obtain the best results using the sets presented in Section IV-A, doubt may arise about generalizing the results to other sets. Therefore, more sets were created to validate the results, using images captured with different cameras, under different lighting conditions, and in different locations from those used in the previous sections. These sets include 43 different signs.<sup>3</sup>

Data obtained for validation sets are shown in Fig. 4(b). From a comparison with Fig. 4(a), it can be seen that validation roughly confirms previous data since the RGB Normalized method remains as the best method for most measures, and results for CST and Edge detection methods are the same. The only important difference concerns the SVM method since the results for validation sets are poor, compared with the previous ones. (SVM was applied in this validation test with the same training as in previous tests.) A deeper analysis of data separating color and white results shows that there is a reduction in the recognition rate for white information, whereas for color, performance is maintained. This reduction may have been the consequence of poor SVM training for white. Color training shows good generalization, but white does not. This may be considered a drawback of the SVM method since, to obtain good results with white information, this method requires more training than for color.

<sup>&</sup>lt;sup>2</sup>More graphs for different methods and parameters can be accessed at http://agamenon.tsc.uah.es/Segmentation.

Measures	GER [25]	Canny	CER (1400)	CER (1500)	RGBNT	HST [14]	HSET [21]	OST	LUT HST	LUT HSET	LUT SVM
Total	47	20	21	23	46	45	45	47	42	42	42
Correct	41	17	18	18	46	44	45	47	42	42	42
False	6	3	3	5	0	1	0	0	0	0	0
Lost	6	30	29	29	1	3	2	0	5	5	5

TABLE X Results Using Tracking Information

5) Tracking Results: Finally, the entire system, including tracking, was tested for each segmentation method. To this purpose, we used a sequence of 7799 images recorded in mixed urban and road environments over 12 km with no relation to the images and sequences used in previous tests. In this test, it was not possible to carry out such exhaustive data collection as in previous tests since the number of images was higher. Therefore, the tracked signs for each segmentation method were used as the result. In Table X, the collected data are shown. The first row gives the number of signs tracked for each method, the second row indicates the number of signs that were correctly identified, the third row indicates the number of false tracked signs, and the last row displays the number of lost signs with respect to the method that gave the highest number of correct signs. From these results, it can be seen that OST is the best method since it has no loss and no false scores. However, the RGB Normalized method obtained very similar results, with only one lost sign. HST and HSET gave good results, with two lost signs and no false for HSET and three lost signs and one false for HST. LUTs obtained the worst results within CST methods, losing five signs.

Once again, CST methods performed better than edgedetection methods, and although the GER method gave comparatively good results, it produced too many lost and false signs. The Canny method, which obtained good results (among edgedetection methods) in previous tests, performed poorly here. A possible explanation may be that this method excessively depends on parameter settings.

## V. CONCLUSION

This paper has presented research aimed at identifying the best segmentation methods for its use in automatic road sign-recognition systems. Different methods employed in previous studies have been implemented, although they have been modified and improved to obtain the best results. Furthermore, other new methods not previously used for this task are proposed, such as SVM, in addition to color spaces not previously tested, such as normalized Ohta. The use of an LUT with some loss of information (2 bit/channel) is also suggested to improve the speed of the slowest methods. Finally, achromatic decomposition in different color spaces has also been presented since the treatment of color and achromatic information can be separated.

Analysis of the data obtained has led to six conclusions.

 The recognition percentage results for the best method are 69.49% for the test sets and 78.29% for the validation sets. These results may seem low, but they were obtained by taking into account all the times that a traffic sign could be recognized in a sequence. Moreover, the image sequences were selected for their complexity, and therefore, the percentage tends to be low.

- 2) For test and validation sequences, the RGB Normalized method performed the best, whereas, for tracking, the best performance was obtained with OST. LUT SVM may be an option, but it needs more refined training in several scenarios.
- 3) Edge-detection methods may be used as a complement to other color-segmentation methods, but they cannot be used alone.
- 4) The use of the LUT method improves speed, and quality was similar to the original method.
- 5) No method performs well in all the contexts.
- 6) Normalization, as in the RGB Normalized method or OST, improves performance and represents a low-cost operation. Although HST or HSET gives good results, their cost in speed and their performance render them unnecessary. Why use a nonlinear and complex transformation if a simple normalization is good enough?

Although it is not possible to identify one method as being the best in all the cases, our main conclusion is that a colorspace-threshold method incorporating illumination normalization constitutes a good choice. In addition, the use of an LUT with some loss of information improves speed and implies that some more lengthy methods could be used with good results.

Achromatic decomposition has been a good choice since most problems of color classification arise when treating achromatic pixels due to instabilities. Moreover, white identification in signs has been improved using this decomposition.

It would appear that the end of prohibition signs has been particularly difficult to detect due to segmentation problems with achromatic colors and the fact that these signs are split into two semicircles by their central bar. To improve the detection of these signs, not only good segmentation but also a more refined detection and recognition method is needed.

The primary aim of this paper was to carry out an exhaustive study to identify the best segmentation method for this particular task, i.e., traffic sign recognition. The main tool for this study has been our traffic-recognition system, which can be improved in all of its stages but was the same for all the segmentation methods used. Consequently, the results presented here should be considered as being relative between methods and in no way absolute.

# Appendix

# SUPPORT VECTOR MACHINE CLASSIFIERS

The principles of SVMs have been developed by Vapnik [42] and presented in several works, such as [43].

The classification task is reduced to finding a decision frontier that divides the data into the groups chosen. The simplest decision case is where the data can be divided into two groups. The work presented here used this type of classification, without loss of generality since multiclassification can be obtained using the SVMs in several ways.

The simplest decision problem comprises a number of vectors divided into two classes, and the optimal decision frontier, which divides these classes, must be encountered. The optimal decision will be the one that maximizes the distance from the frontier to the data. In a 2-D case, the frontier will be a line, whereas in a multidimensional space, the frontier will be a hyperplane. The desired decision function has the following form:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b$$
(19)

where x is the input vector, and the y values that appear in this expression are +1 for one-class training vectors and -1 for the others. In addition, the inner product is performed between each training input and the vector that must be classified. Thus, a set of training data (x, y) is needed to find the classification function. The  $\alpha$  values are the Lagrange multipliers obtained in the minimization process, and the *l* value will be the number of vectors  $(x_i)$  that, during the training process, contribute to forming the decision frontier. These vectors are those with an  $\alpha$  value that is not equal to zero and are known as support vectors.

When the data are not linearly separable, this scheme cannot directly be used. To prevent this problem, the SVMs can map the input data into a high-dimensional feature space using the well-known kernel method. An optimal hyperplane is constructed by the SVMs in a high-dimensional space and then returns to the original space, transforming this hyperplane into a nonlinear decision frontier. The nonlinear expression for the classification function is given in

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$
(20)

where K is the kernel that performs the nonlinear mapping.

The choice of this nonlinear mapping function or kernel is very important for the performance of the SVMs. One kernel that is generally used with good results is the radial basis function. This function has the expression given in

$$K(\mathbf{z}, \mathbf{w}) = \exp\left(-\gamma |\mathbf{z} - \mathbf{w}|^2\right).$$
(21)

The  $\gamma$  parameter in (21) must be chosen to reflect the degree of generalization applied to the data used. When the input data are not normalized, this parameter also performs a normalization task.

When some data within the sets cannot be separated, the SVMs can include a penalty term (C) in the minimization, which renders misclassification more or less important. The greater this parameter is, the more significant the misclassification error in the minimization procedure.

#### ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the editors for their many helpful suggestions, which have greatly improved the presentation of this paper.

#### REFERENCES

- N. Barnes, A. Zelinsky, and L. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 322–332, Jun. 2008.
- [2] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Siegmann, H. Gomez-Moreno, and F. J. Acevedo-Rodriguez, "Traffic sign recognition system for inventory purposes," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 4–6, 2008, pp. 590–595.
- [3] C. Nunn, A. Kummert, and S. Muller-Schneiders, "A novel region of interest selection approach for traffic sign recognition based on 3D modelling," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 4–6, 2008, pp. 654–659.
- [4] C. F. Paulo and P. L. Correia, "Traffic sign recognition based on pictogram contours," in *Proc. 9th WIAMIS*, May 7–9, 2008, pp. 67–70.
- [5] B. Cyganek, "Road signs recognition by the scale-space template matching in the log-polar domain," in *Proc. 3rd Iberian Conf. Pattern Recog. Image Anal.*, vol. 4477, *Lecture Notes in Computer Science*, 2007, pp. 330–337.
- [6] W.-J. Kuo and C.-C. Lin, "Two-stage road sign detection and recognition," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Beijing, China, Jul. 2007, pp. 1427–1430.
- [7] Y. Liu, T. Ikenaga, and S. Goto, "Geometrical, physical and text/symbol analysis based approach of traffic sign detection system," in *Proc. IEEE Intell. Vehicle Symp.*, Tokyo, Japan, Jun. 2006, pp. 238–243.
- [8] G. K. Siogkas and E. S. Dermatas, "Detection, tracking and classification of road signs in adverse conditions," in *Proc. IEEE MELECON*, Málaga, Spain, 2006, pp. 537–540.
- [9] H. Ishida, T. Takahashi, I. Ide, Y. Mekada, and H. Murase, "Identification of degraded traffic sign symbols by a generative learning method," in *Proc. 18th ICPR*, Hong Kong, Aug. 2006, vol. 1, pp. 531–534.
- [10] W. Wu, X. Chen, and J. Yang, "Detection of text on road signs from video," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 4, pp. 378–390, Dec. 2005.
- [11] H. Kamada, S. Naoi, and T. Gotoh, "A compact navigation system using image processing and fuzzy control," in *Proc. IEEE Southeastcon*, New Orleans, LA, Apr. 1990, vol. 1, pp. 337–342.
- [12] W. Ritter, "Traffic sign recognition in color image sequences," in Proc. Intell. Vehicles Symp., Detroit, MI, Jul. 1992, pp. 12–17.
- [13] R. Janssen, W. Ritter, F. Stein, and S. Ott, "Hybrid approach for traffic sign recognition," in *Proc. IEEE Int. Conf. Intell. Vehicles*, 1993, pp. 390–397.
- [14] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-Moreno, and F. López-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 264–278, Jun. 2007.
- [15] S. Lafuente-Arroyo, S. Maldonado-Bascón, P. Gil-Jiménez, H. Gómez-Moreno, and F. López-Ferreras, "Road sign tracking with a predictive filter solution," in *Proc. 32nd IEEE IECON*, Nov. 2006, pp. 3314–3319.
- [16] P. Arnoul, M. Viala, J. P. Guerin, and M. Mergy, "Traffic signs localisation for highways inventory from a video camera on board a moving collection van," in *Proc. IEEE Intell. Vehicles Symp.*, Tokyo, Japan, Sep. 1996, pp. 141–146.
- [17] L. Lucchese and S. K. Mitra, "Color image segmentation: A state-of-theart survey," *Proc. Indian Nat. Sci. Acad.*, vol. 67-A, no. 2, pp. 207–221, 2001.
- [18] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognit.*, vol. 34, no. 12, pp. 2259–2281, Dec. 2001.
- [19] A. de la Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road traffic sign detection and classification," *IEEE Trans. Ind. Electron.*, vol. 44, no. 6, pp. 848–859, Dec. 1997.
- [20] J. Miura, T. Kanda, and Y. Shirai, "An active vision system for real-time traffic sign recognition," in *Proc. IEEE Intell. Transp. Syst.*, Oct. 1–3, 2000, pp. 52–57.
- [21] A. de la Escalera, J. M. Armingol, J. M. Pastor, and F. J. Rodríguez, "Visual sign information extraction and identification by deformable models for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 2, pp. 57–68, Jun. 2004.
- [22] E. De Micheli, R. Prevete, G. Piccioli, and M. Campani, "Color cues for traffic scene analysis," in *Proc. Intell. Vehicles Symp.*, Detroit, MI, Sep. 1995, pp. 466–471.

- [23] C. Fang, S. Chen, and C.-S. Fu, "Road sign detection and tracking," *IEEE Trans. Veh. Technol.*, vol. 52, no. 5, pp. 1329–1341, Sep. 2003.
- [24] H. Fleyeh, "Shadow and highlight invariant colour segmentation algorithm for traffic signs," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Jun. 2006, pp. 1–7.
- [25] Y. Aoyagi and T. Asakura, "A study on traffic sign recognition in scene image using genetic algorithms and neural networks," in *Proc. 22nd IEEE Int. Conf. Ind. Electron., Control Instrum.*, Taipei, Taiwan, Aug. 1996, vol. 3, pp. 1838–1843.
- [26] M. Garcia-Garrido, M. Sotelo, and E. Martin-Gorostiza, "Fast traffic sign detection and recognition under changing lighting conditions," in *Proc. IEEE ITSC*, M. Sotelo, Ed., 2006, pp. 811–816.
- [27] P. Gil-Jiménez, S. Maldonado-Bascón, H. Gómez-Moreno, S. Lafuente-Arroyo, and F. López-Ferreras, "Traffic sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies," *Signal Process.*, vol. 88, no. 12, pp. 2943–2955, Dec. 2008.
- [28] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. New York: Springer-Verlag, 2000.
- [29] P. Kumar, K. Sengupta, and A. Lee, "A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst.*, 2002, pp. 100–105.
- [30] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst., Man, Cybern., vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
- [31] S. Lafuente-Arroyo, P. García-Díaz, F. J. Acevedo-Rodríguez, P. Gil-Jiménez, and S. Maldonado-Bascón, "Traffic sign classification invariant to rotations using support vector machines," in *Proc. Adv. Concepts Intell. Vision Syst.*, Brussels, Belgium, Aug./Sep. 2004.
- [32] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Comput. Graph. Image Process.*, vol. 13, no. 3, pp. 222–241, Jul. 1980. [Online]. Available: www.ri.cmu.edu/pubs/pub\_2487.html
- [33] C. Vertan and N. Boujemaa, "Color texture classification by normalized color space representation," in *Proc. Int. Conf. Pattern Recog.*, Barcelona, Spain, 2000, pp. 580–583.
- [34] H. Liu, D. Liu, and J. Xin, "Real-time recognition of road traffic sign in motion image based on genetic algorithm," in *Proc. 1st Int. Conf. Mach. Learning Cybern.*, Nov. 2002, pp. 83–86.
- [35] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [36] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE J. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1338–1359, Dec. 1997.
- [37] H. Gómez-Moreno, P. Gil-Jiménez, S. Lafuente-Arroyo, R. Vicen-Bueno, and R. Sánchez-Montero, *Recent Advances in Intelligent Systems and Signal Processing*. Corfu, Greece: WSEAS, 2003, ch. Color images segmentation using the Support Vector Machines, pp. 151–155.
- [38] C. Chang and C. Lin, LIBSVM: A Library for Support Vector Machines, 2001. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm
- [39] L. Yang, F. Albregtsen, T. Lnnestad, and P. Grttum, "A supervised approach to the evaluation of image segmentation methods," in *Proc. CAIP*, vol. 970, *Lecture Notes on Computer Science*, 1995, pp. 759–765.
- [40] Y. J. Zhang, "A review of recent evaluation methods for image segmentation," in *Proc. 6th Int. Symp. Signal Process. Appl.*, Aug. 2001, vol. 1, pp. 148–151.
- [41] S. Chabrier, B. Emile, H. Laurent, C. Rosenberger, and P. Marche, "Unsupervised evaluation of image segmentation application to multi-spectral images," in *Proc. 17th ICPR*, Aug. 23–26, 2004, vol. 1, pp. 576–579.
- [42] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 2000.
- [43] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Methods. Cambridge, U.K.: Cambridge Univ. Press, 2000.



Hilario Gómez-Moreno (M'99) received the B.S. degree in telecommunication engineering from the Universidad de Alcalá, Alcalá de Henares, Spain, in 1994 and the M.S. degree in telecommunication engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 1999. He is currently working toward the Ph.D. degree with the Universidad de Alcalá.

Since 1997, he has been with the Departamento de Teoría de la Señal y Comunicaciones, Escuela Politécnica Superior, Universidad de Alcalá, where

he became a Full-Time Lecturer in 2002. His research interests are image processing and support vector machines.



Saturnino Maldonado-Bascón (S'98–A'00–M'04– SM'10) received the Telecommunication Engineering degree from the Universidad Politécnica de Madrid, Madrid, Spain, in 1996 and the Ph.D. degree from the Universidad de Alcalá, Alcalá de Henares, Spain, in 1999.

Since 1997, he has been a member of the faculty of the Universidad de Alcalá, becoming an Associate Professor with the Departamento de Teoría de la Señal y Comunicaciones, Escuela Politécnica Superior, in 2001. His research interests are image

processing and pattern recognition.



**Pedro Gil-Jiménez** received the Telecommunication Engineering and Ph.D. degrees from the Universidad de Alcalá, Alcalá de Henares, Spain, in 2001 and 2009, respectively.

In 1999, he joined the Departamento de Teoría de la Señal y Comunicaciones, Escuela Politécnica Superior, Universidad de Alcalá, where he has been a Lecturer since 2001. His research interests include computer vision, video-surveillance systems, and intelligent transportation systems, particularly those involving signal and image processing.



**Sergio Lafuente-Arroyo** received the B.S. and M.S. degrees in telecommunication engineering in 2000 and 2004, respectively, from the Universidad de Alcalá, Alcalá de Henares, Spain, where he is currently working toward the Ph.D. degree.

He is currently an Instructor with the Departamento de Teoría de la Señal y Comunicaciones, Escuela Politécnica Superior, Universidad de Alcalá. His research interests include digital image processing and computer vision.