# Shape Classification Algorithm Using Support Vector Machines for Traffic Sign Recognition

P. Gil-Jiménez, S. Lafuente-Arroyo, S. Maldonado-Bascón, and H. Gómez-Moreno

Dpto. de Teoría de la señal y Comunicaciones, Universidad de Alcalá, 28871 Alcalá de Henares (Madrid), Spain
{pedro.gil, sergio.lafuente, saturnino.maldonado, hilario.gomez}@uah.es

**Abstract.** In this paper, a new algorithm for traffic sign recognition is presented. It is based on a shape detection algorithm that classifies the shape of the content of a sign using the capabilities of a Support Vector Machine (SVM). Basically, the algorithm extracts the shape inside a traffic sign, computes the projection of this shape and classifies it into one of the shapes previously trained with the SVM. The most important advances of the algorithm is its robustness against image rotation and scaling due to camera projections, and its good performance over images with different levels of illumination. This work is part of a traffic sign detection and recognition system, and in this paper we will focus solely on the recognition step.

## 1 Introduction

A traffic sign recognition system basically consists on an image processing system mounted over a vehicle recording the road, and the goal of the system is the detection of all traffic signs present on the scene, and also its classification according with its shape, colors and meaning. Table 1 shows the meaning of traffic signs according with its color and shape. All the signs and properties described are for the Spanish traffic signs.

In most cases, traffic sign research is divided into two basic steps, namely detection an recognition. The detection step is the process that determines which parts of the images are candidate to be a traffic sign. In many works [1], [2], [3], [4], the detection is based on a color-based segmentation, taking advance of the main colors used on traffic signs, as we can see in table 1. For that reason, red, blue, yellow and/or white are the most frequently colors used on the segmentation process.

Once candidate blobs have been extracted from the image, some approaches implement a pre classification step according with its shape [1], [5]. From table 1, the shapes used in traffic signs are the equilateral triangle, the circle, the octagon and the square. With this additional step, we reduce the classification problem to a smaller number of classes, therefore reducing the time employed in the classification stage.

J. Cabestany, A. Prieto, and D.F. Sandoval (Eds.): IWANN 2005, LNCS 3512, pp. 873–880, 2005.

**Table 1.** Meaning of traffic signs according to its color and shape

| Color | Shape | Meaning |
|---|---|---|
| Red Rim | Circle | Prohibition |
| Red Rim (Up) | Triangle | Danger |
| Red Rim (Down) | Triangle | Yield |
| Red | Octagonal | Stop |
| Blue | Square | Recommendation |
| Blue | Circle | Obligation |
| White | Circle | End of prohibition |
| Yellow | Circle | End of prohibition (construction) |

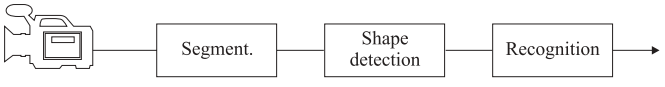



**Fig. 1.** Block diagram

In figure 1 we can see the block diagram of the traffic sign recognition system we have described. The algorithm described in this paper is related with the last step in figure 1, that is, once every possible blob has been classified according with its shape and color, the last step consists on the recognition of the content of the sign, and so, the identification of the sign. A complete description for the rest of the system can be found in [5].

Many works have been proposed for the recognition step. In [4], a NN is used to perform the classification of every possible blob. In [3], the identification of signs is carried out by a normalized correlation-based pattern matching using a traffic sign image database. In this work the classification task is performed by a SVM [6] as it will be described later. We have focused our algorithm on circular signs, since on these kind on signs, object rotations appear as a problem than can not be solved as easily as with other kind of shapes, where reorientation of the blob can be done from the edges of the sign.

## 2 Shape Classification

Since the inside part of a typical traffic sign is normally composed of one or two colors, plus the background, a new segmentation process can be performed over the blob to separate the content of the sign from the background. After this process, the problem is reduced to classify the shape of the segmented object into one of all possible objects that can be found inside a traffic sign, that is, classify the sign according with its meaning.

### 2.1 Segmentation

Figure 2 shows an example of sign segmentation for circular images with white background. In 2(a) we can see the blob corresponding to the sign. This blob is


(a) (b) (c)

**Fig. 2.** Segmentation example. (a) Original blob, (b) Original blob without the red rim, (c) Segmented image

the output of the previous steps, as mentioned above. Since the previous steps has classified the sign into the circular group, we can approximately determine the center and the radius of the sign. With this two parameter, removing the red rim from the sign is straightforward. Obviously, this step must be different for every kind of sign, according with its shape and colors, but the procedure is the same for signs from the same category. In 2(b), the same sign is shown, where the red circular rim has already been removed. Once the rim is removed, the last process is the segmentation of the content of the sign. In figure 2(c) the segmented image is shown, where the segmentation has been performed over the luminance matrix, using an adaptive threshold. This process gives a certain robustness against illumination changes.

### 2.2 Object Projection

The classification of the segmented object now is performed computing the projections of the objects. These projections are computed at several values of $\theta$, according with figure 3. As we will see later, this procedure give a great robustness against object rotations. For each value of $\theta$, a new coordinates system, denoted $uv$, is generated, where the transformation matrix between this system and the image coordinates system $(xy)$ is given by:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Since the projection of the object is computed for each value of $\theta$, the whole operation will yield a two dimension matrix of size M x N (see figure 4), being M the number of values of $\theta$ where the projections are computed, and N is the number of samples of each projection. For a particular value of M, the value of $\theta$ for each iteration is incremented according with the following step:

$$\Delta\theta = \frac{2\pi}{M}$$

The projections are computed over the coordinate $u$, (see figure 3). To make the results independent of the object size, and hence, robust to image scaling, N is a constant in the system, and the step between samples in the coordinate $u$ are

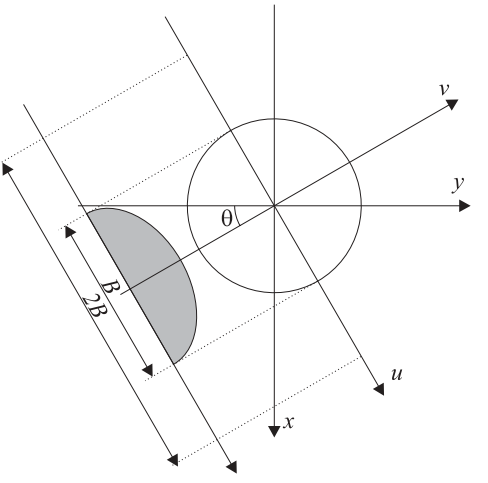



**Fig. 3.** *uv* and *xy* coordinates system

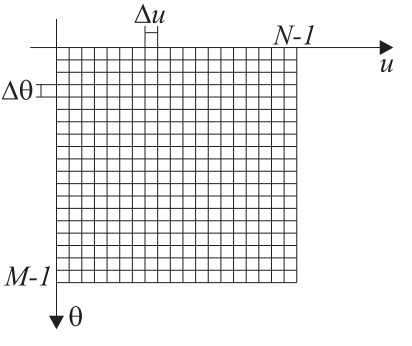



**Fig. 4.** Projection matrix

determined taking into account the size of the object. The maximum width of the object can be estimated from the second order moments in the angle of the least moment of inertia [7]. For a rectangle of height $H$, width $B$, and $B > H$, the second order moments are:

$$\mu_{20} = \frac{B^3 H}{12}$$
$$\mu_{02} = \frac{H^3 B}{12}$$

From these expressions, we can compute the width of the rectangle, $B$:

$$\mathrm{B} = \sqrt[8]{\frac{12^2 \cdot \mu_{20}^3}{\mu_{02}}}$$

For an arbitrary shape, the previous expression allows us to estimate the maximum width of the object from its second order moments. In this case, the previously computed value of $B$ is not its maximum width, but it leads us to determine the starting and ending values in the coordinate $u$ where the projections are to be computed. These limits will be:

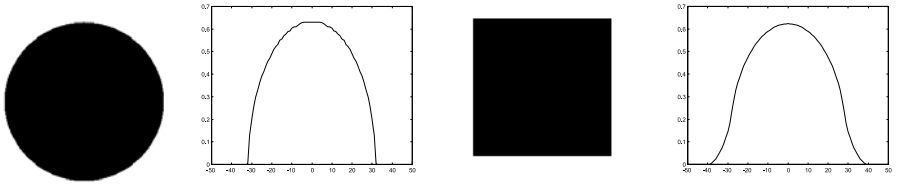
$$-k\frac{B}{2} < u < k\frac{B}{2}$$

where $k$ is a constant chosen according with the following discussion. Large values of $k$ implies that many samples of the projections will be null for values of $u$ near $-kB/2$ and $kB/2$, because we are evaluating beyond the object. With small values of $k$, samples of the projection can be loosen because we are not analyzing the whole object. The value actually chosen was $k = 2$, and so the projections are computed for $-B < u < B$ and $-B < v < B$ (see figure 3). With this value, the step between samples will be:

$$\Delta u = \frac{2 \cdot B}{N}$$

From now on, we will call this matrix the *projection matrix*. The next step consists on the computation of the mean value for every column in the projection matrix to obtain a vector of N elements. As with the previous matrix, we will call this vector the *projection vector*. With this step we make this vector invariant to object rotation, since a rotation on the object is reflected as a circular shift on the projection matrix in coordinate $\theta$, and since we compute the mean value for every value of $\theta$, the result is independent of the orientation of the object. In figure 5 it is shown the projection vector for a circle (5(a) and 5(b)) and the projection vector for a square (5(c) and 5(d)).

As it can be seen easily, the projection vector is symmetric, and hence, only half of the vector is used. This new projection vector, of size $N/2$, will be the vector used in the classification process. Figure 6 shows the projection vector for some different shapes used in the evaluation section, along with its corresponding image and its segmented mask.



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

**Fig. 5.** Projection vector examples

(a) Category A (Maximum speed limit 120)

(b) Category B (Maximum speed limit 40)

(c) Category C (Maximum speed limit 90)

(d) Category D (No overtake)

(e) Category E (Maximum speed limit 100)

**Fig. 6.** Projection vector examples for the test images

## 3 Experimental Results

A set of 154 blobs were chosen to test the proposed scheme. These blobs were extracted from an image database we have created for the evaluation of different traffic sign detection and recognition algorithms, and it is available at http://roadanalysis.uah.es. In this set we have included signs from 5 different types, as can be seen in figure 6. The algorithm basically extracts the projection vector as described above, taking N equal 100, that lead to a final vector of $N/2 = 50$ samples. The value of N was chosen as a tradeoff between precision and speed of the algorithm. We also take M equal 90, leading to a step of 4 degrees in $\theta$. This value of M is again a tradeoff between precision and speed. Higher values of M implies more computational load, with no significant improvement in the results. Smaller values of M may reduce the robustness of the algorithm to object rotations.

In the experiment, an exhaustive search over the parameters $\gamma$ and $C$ was performed in order to find the values where the total number of errors in the experiment were minimum. The results for this search show that the optimal values lay near the values $\gamma = 1$ and $C = 100$. Once the optimal values were found, from now on, all remaining experiments use these two particular values for parameters $\gamma$ and $C$. We used 4 blobs of each category, which account for a total of 20 blobs, in the search step. For the rest of the set, one third of the blobs of each category was taken as the training set, leaving the rest of the blobs as the testing set. This experiment was repeated three times, taking as the training set for a particular experiment a different set of blobs than the ones used on the other experiments. The processes were performed with a SVM [6] with a Gaussian Kernel. The implementation uses the library LIBSVM [8].

Table 2 shows the results for the optimal values. This table shows, for each category, the success probability as the number of blobs properly classified with respect to the total number of blobs in the prediction set. Note that the experiment was repeated three times, and these values are the mean values for the three experiments. It also shows the number of blobs used for the training step and the number of blobs used in the prediction step.

From the results, we must conclude that the success probability is not good enough for categories with a small number of samples, especially for categories $A$ and $B$, whereas for categories with enough number of samples are satisfactory, which makes the overall success probability acceptable. We also have to take into account that the images used in the experiment are realistic images, that

**Table 2.** Results for $\gamma = 1$ and $C = 100$

| Category | A | B | C | D | E | Total |
|---|---|---|---|---|---|---|
| Number of blobs | 18 | 18 | 29 | 46 | 23 | **134** |
| Training | 6 | 6 | 9 | 15 | 7 | **43** |
| Prediction | 12 | 12 | 20 | 31 | 16 | **91** |
| Success prob. (%) | 80 | 78 | 89 | 97 | 87 | **90** |

includes different illuminations, occlusions, deteriorated signs and other kind of problems that make the success probability decrease.

## 4 Conclusions

This paper describes a new algorithm for the recognition of traffic signs. It is based on a shape detector that focuses on the content of the sign to perform the recognition of traffic signs. The classification is done by a SVM where the input is a vector computed from the projections of the object at several angles to overcome orientation problems. Other advantages of the algorithm is its robustness against different illumination and scaling, and also its simplicity.

The direction of our future work must include the increase of the test set database in two ways: first, increase the number of categories, and second, increase the number of blobs for each category. We also need to focus our work in the previous steps, especially the segmentation, since this step is crucial for the correct operation of the whole system. We also have to deal with other kind of problems, like partial occlusions, shadows, bad illuminations, etc.

## Acknowledgement

This work was supported by the project of the Ministerio de Educación y Ciencia de España number TEC2004/03511/TCM.

## References

1. de la Escalera, A., Moreno, L., Salichs, M.A., Armingol, J.: Road traffic sign detection and classification. IEEE trans. on industrial electronics **44** (1997) 848–859
2. Kamada, H., Naoi, S., Gotoh, T.: A compact navigation system using image processing and fuzzy control. Southeastcon Proceedings **1** (1990) 337 – 342
3. Miura, J., Kanda, T., Shirai, Y.: An active vision system for real-time traffic sign recognition. Proc. IEEE Inteligent transportation systems (2000) 52–57
4. de la Escalera, A., Armingol, J.M., Mata, M.: Traffic sign recognition and analysis for intelligent vehicles. Image and vision computing **21** (2003) 247–258
5. Lafuente-Arroyo, S., García-Díaz, P., Acevedo-Rodríguez, F.J., Gil-Jiménez, P., Maldonado-Bascón, S.: Traffic sign classification invariant to rotations using support vector machines. ACIVS'04 (2004)
6. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (2000)
7. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice Hall (1989)
8. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.