Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision, Malta, September 15-17, 2005 (pp216-221)

Recognition of Mandatory Traffic Signs using the Hausdorff distance

R. J. LÓPEZ-SASTRE, S. LAFUENTE-ARROYO, P. SIEGMANN, P. GIL-JIMÉNEZ, A. VAZQUEZ-REINA
Department of Signal Theory and Communications
Universidad de Alcalá de Henares
Escuela Politécnica Superior. Campus Universitario. 28805 Alcalá de Henares

http://www.uah.es

Abstract: - This paper proposes a new recognition algorithm of mandatory traffic sings using the Hausdorff distance. This algorithm has been designed to detect arrows on traffic signs, especially direction signs. The arrows on these mandatory signs appear in a multitude of different forms and positions. Due to this variety this algorithm uses a structural approach to recognize the arrows. First the sign is transformed into a binary bitmap. The second stage consists in computing the skeleton of the arrows that appear in the mandatory signs of the image. The algorithm calculates de Hausdorff distance between some models of skeletons of arrows and the skeleton obtained. The algorithm recognizes the model that produces the smallest Hausdorff distance.

Key-Words: - ideogram, arrow, mandatory, traffic signs, recognition, Hausdorff distance, Voronoi surface, skeleton.

1 Introduction

Traffic sign detection and recognition have been an important issue for research recently: [1], [2], [3] and [4] are some of these works. One kind of traffic signs are the mandatory traffic signs. This paper proposes a recognition algorithm of mandatory traffic signs using the Hausdorff distance. This algorithm works with the skeleton of the arrow that is inside of every mandatory sing. The Hausdorff distance allows us to determine the difference between shapes. A technique for comparing images using the Hausdorff distance can be found in [5]. This new algorithm uses this distance for comparing the skeleton obtained from the arrows of real mandatory signs with the skeletons obtained from the arrows of ideal mandatory signs.

This paper contains the next sections. In section 2 it makes a global description of the new algorithm implemented. Section 3 contains the experimental results and section 4 shows the conclusions and the direction of the future works.

2 Traffic Sign Recognition Algorithm

The algorithm proposed in this paper has four steps:

- Step 1: Segmentation.
- Step 2: Shape Classification.
- Step 3: Thinning of the objects.
- Step 4: Arrow Recognition.

Fig.1 shows the four steps of the algorithm. First the algorithm makes a segmentation of the original image and generates a binary image. The skeleton of the image can be calculated in the next stage and then, a recognition process based on Hausdorff distance, can be run. This kind of implementation, without de shape classification step, takes more time. In order to make an efficient algorithm the second step it is necessary. Now we explain the four steps of the algorithm.

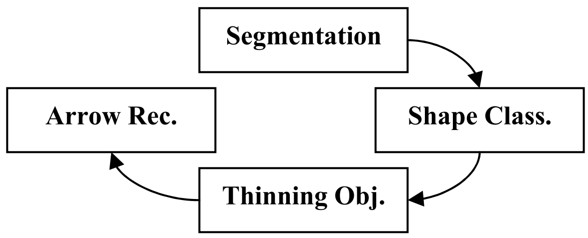


Fig.1 Four steps of the algorithm.

In the first step, a segmentation of the original image it's done. The algorithm creates a mask where pixels of the image that may belong to a mandatory traffic sign were marked as object pixels, whereas pixels that may not belong to a mandatory traffic sign were marked as background pixels. In this way, after this first step we have a binary image. To achieve this task, some different colour-based segmentations are performed over the original image, taking advance of the most frequently colours used in mandatory traffic signs A complete description of a segmentation process applied to traffic signs can be found in [1]. Some examples of mandatory traffic signs are

Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision, Malta, September 15-17, 2005 (pp216-221)

represented in Fig.2. Fig.3 shows an original image which contains a mandatory traffic sign and the white based segmented mask obtained in the first stage of the algorithm. Once this binary image has been generated the second step can start.

Fig.2 Two examples of mandatory traffic signs.

2.1 Shape Classification

This stage of the algorithm is fundamental for the correct and efficient arrow recognition process. Not all the zones of the white based segmented mask are candidates to be a mandatory traffic sign. With the traffic sign shape classification the algorithm generates a list *LObject* that contains the form and the position of every object that appears in the binary image. In [2] is described all the process implemented for this shape classification.

(a)

(b)

Fig.3 Segmentation step. (a) Original image, (b) White based segmented mask.

After the segmentation process this algorithm computes the connected components, in order to get an initial list of all possible objects in the image. A bounding rectangle and the mask area for each object it is computed too. The final step of this shape classification requires a completely filled mask to get a correct result. After this mask filling process the decision step is performed. This decision is made in base of comparisons between the signature of the blob and the signature of the theoretical shapes we are looking for. To make the algorithm robust to object rotations, which become circular shift signature, it compares the absolute value of the FFT of the signatures instead of the signatures themselves, taking advance of the property of the DFT in the presence of shifts.

This Shape classification is one of the most important steps in the process of traffic sign recognition. In fact, shape classification can be considered as the step where traffic signs are located in the image.

2.2 Thinning Algorithm

The goal of this stage is to obtain the skeleton of every arrow that is inside of each mandatory traffic sign. We can calculate the skeleton of all the binary image but this strategy takes more computational cost than another one that only computes the skeleton of a part of the binary image. In this algorithm we apply the thinning method designed only to the circular objects that have been detected after the traffic sign shape classification stage. With the skeleton of every object the algorithm can start the recognition process.

The thinning algorithm starts with the binary image and the list *LObject* that contains all the objects detected. Each element of this list contains the following information:

- the area in pixels of the object.
- the bounding box of the object.
- the form of the object.

The next thinning algorithm is applied for every object of the list. Thinning is a morphological operation that is used to erase selected pixels from binary images. We have implemented this thinning algorithm to obtain the skeleton of an object. Like other morphological operations the thinning algorithm uses binary structuring elements. For the special application of skeletonization, structuring elements have been designed which appear in Fig.4. In every iteration of the algorithm, the binary image is first thinned by the structuring element Fig.4 (a), and then by the structuring element represented in Fig.4 (b). After these two structuring elements our algorithm uses the remaining six 90º rotations of the two previous elements. These six structuring elements are represented in Fig.5. The algorithm consists in successive iterations until none of the thinning processes produces any further change. In each iteration the algorithm verifies all of the eight structuring elements for every pixel marked as object pixel. For each structuring element, the algorithm executes a function that calculates if a pixel verifies the conditions of the structuring element that has been analysed or not. This function is called *verifymask* . The thinning function can be defined as

$$thinning(I,E) = I - verifymask(I,E) \quad (1)$$

where I is the binary image and E is the structuring element. The subtraction is a logical subtraction defined by

$$I - J = I \cap NOT(J) \quad (2)$$

where I and J are binary images. The algorithm computes (1) for every pixel of I marked as object pixel, and for each structuring element E defined for this thinning algorithm. The results obtained with this algorithm are presented in Fig.6 and Fig.7 for some different images.

0	0	0
	1	
1	1	1

(a)

	0	0
1	1	0
	1	

(b)

Fig. 4 Structuring elements for skeletonization by morphological thinning.

In [6] and [7] other thinning algorithms can be found. The algorithm presented here doesn't produce errors with horizontal edges like other thinning algorithms. Fig. 6 shows the skeleton obtained from an ideal mandatory traffic sign. This model is an image of 215 x 217 pixels and the thinning algorithm only needs 23 iterations to obtain the skeleton that the Fig. 6 shows. Fig. 7 represents some skeletons that belong to the arrows of some real mandatory traffic signs. The computation of this skeleton allows us an efficient implementation for a recognition algorithm.

The arrows on these mandatory traffic signs appear in a multitude of different forms. Due to this variety this recognition algorithm uses a structural approach based on the skeleton of the arrow, so this thinning algorithm is a fundamental stage for the recognition process.

0		1
0	1	1
0		1

1	1	1
	1	
0	0	0

1		0
1	1	0
1		0

(a)

0	0	
0	1	1
	1	

	1	
0	1	1
0	0	

	1	
1	1	0
	0	0

(b)

Fig.5 Six 90º rotations of the two structuring elements of Fig.4.

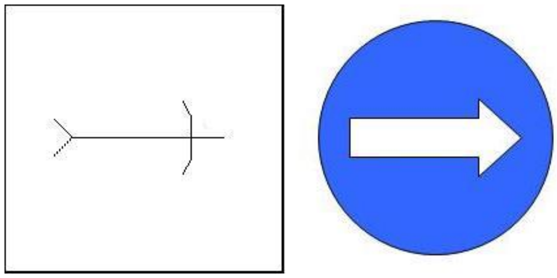


Fig.6 Skeleton of the ideal mandatory traffic sign.

Fig.7 Some skeletons of arrows obtained from real mandatory traffic signs.

2.3 The Hausdorff Distance

Our algorithm uses a method based on the Hausdorff distance for the mandatory traffic signs recognition process. This kind of distance measures the extent to which each point of a model set lies near some point of an image set and vice versa. Thus, this distance can be used to determine the degree of resemblance between two objects that are superimposed on one another. The geometric comparisons of shapes is a fundamental tool for model-based object recognition in images, where many of the methods used refer to a similarity measure between the model features and features of the image. In our particular case, of mandatory traffic sign recognition, the algorithm calculates the Hausdorff distance between some models of skeletons of ideal arrows, and the skeletons of arrows obtained from real images of mandatory traffic signs. In [5] and [8] there are some descriptions about how to use the Hausdorff distance in the pattern recognition problem in images. Algorithm presented in this paper uses the Hasudorff distance described in [5] but our approach differs in one important way. Due to the structural description of the arrow for which it has been designed, and the defined possible placement for mandatory traffic signs in the images, our algorithm will not compute the Hausdorff distance between all possible relative positions of the model in the real images. This strategy allows us a faster implementation for this recognition process. The algorithm designed computes the Hausdorff distance between 5 models of skeletons and a real skeleton. These models represent the real positions of the mandatory traffic signs. For each model 15 masks have been designed and every one has a different size (from 450 x 450 pixels to 60 x 60 pixels). Fig. 8 shows the five models constructed for this application.

With the Hausdorff distance we can compare portions of shapes. This property is very important for our particular application for traffic signs recognition. In a lot of situations it is probable that a traffic sign appears only partly visible due to occlusions. For this reason our algorithm for recognition uses the Hasudorff distance.

The method for computing the Hausdorff distance for this article is similar in many ways to binary correlation, except that the Hausdorff distance is a nonlinear operator.

2.2.1 The Hausdorff distance

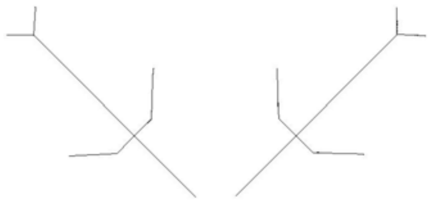
Given two finite point sets $I = \{i_1, i_2, i_3, \ldots, i_p\}$ and $M = \{m_1, m_2, m_3, \ldots, m_q\}$, the Hausdorff distance is defined as

$$H(I,M) = \max(h(I,M), h(M,I)) \quad (3)$$

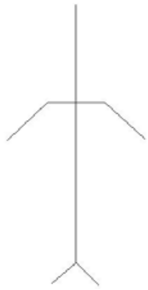
where

$$h(I,M) = \max_{i \in I} \min_{m \in M} \|i - m\| \quad (4)$$

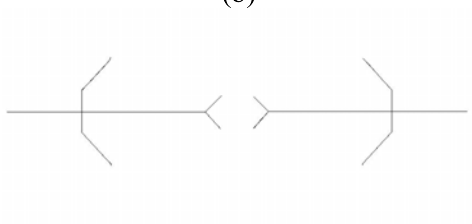
and $\|\cdot\|$ is some underlying norm on the points of I and M, for example the L_2 or Euclidean norm. In our application I contains the points of the object of the image, and M the points of the model.



(a)



(b)



(c)

Fig. 8 Five models for recognition using the Haussdorff distance.

The function $h(I,M)$ is called the directed Hasudorff distance from I to M. It identifies the point i of I that is farthest from any point of M and measures the distance from i to his nearest neighbour in M. The description presented in (3) is made for the continuous case. For pattern recognition in images the points sets lie on an integer grid. Each point i of an integer grid I has Cartesians integer coordinates (i_x , i_y), and analogously (m_x, m_y) Cartesians integer coordinates for every pixel m of M . For computing the Hausdorff distance $H(I, M)$ between these two binary mask I and M, the algorithm follows the next steps. First computes for the two sets I and M the rasterized approximations to their respective Voronoi surfaces, called $d(x)$ and $d'(x)$. The algorithm computes two arrays: $D[x,y]$ and $D'[x,y]$. With the notation $D[x,y]$ we denote an array that specify the rasterized approximation for the set I, and with $D'[x,y]$ for the set M. Each element of the previous arrays contains the distance to the nearest pixel nonzero from the position with coordinates (x, y). There are so many methods for computing these arrays. In every method the norm $\|\cdot\|$ used determines the specified value of each pixel of D or D'. The algorithm proposed from this paper uses a norm based in the Chessboard distance which is defined as follows. Given two pixels i and j with Cartesians coordinates (x_i, y_i) and (x_j, y_j) respectively, the Chessboard distance $d(i,j)$ between pixels i and j is defined as

$$d(i,j) = \max(|x_i - x_j|, |y_i - y_j|) \quad (5)$$

Then, the array $D[x,y]$ is zero where the pixel $I(x,y)$ is marked as an object pixel, and if it is marked as a foreground pixel the algorithm computes the Chessboard distance to the nearest pixel marked as an object pixel. It is not necessary to compute the distance from the pixel to all the other nonzero pixels and then take the minimum distance. The algorithm first computes the distance from the pixel to his first 8 neighbours, if none of them is a nonzero pixel, then it computes the distance to his next 16 neighbours, etc. Fig. 9 shows this process which allows an efficient computation for the Voronoi surfaces for each pixel.

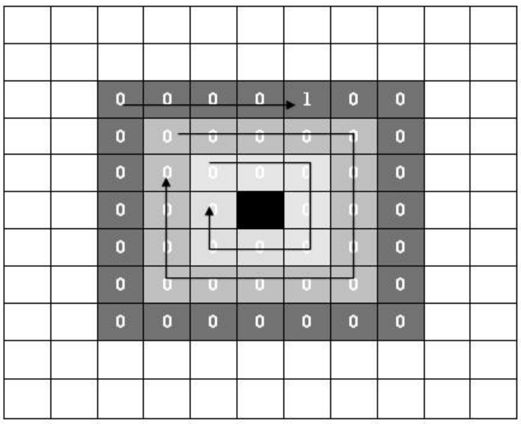


Fig. 9 Steps for computing the Voronoi surfaces for each pixel.

Other methods for computing these D and D' arrays are described in [9] and [10].

Once these arrays are computed the next step is to compute the directed $h(I,M)$ and the inverted $h(M,I)$ Hausdorff distance. We explain the process to compute $h(I,M)$ and for computing $h(M,I)$ the process is analogous. $h(I,M)$ is obtained by the next algorithm. The binary image array I is covered. For every pixel which is marked as object pixel, with (x, y) coordinates, the algorithm looks for the value of the array D' in the same position, defined by the (x,y) coordinates too, if this distance is lower than the previous looked for, the algorithm updates $h(I,M)$ with this new value. To obtain $h(M,I)$ the process is analogous, but in this case the distance array that is used for computing the minimum distance is D. Once these two partial distances are obtained the Hausdorff distance is computed as described in (3).

This described process is executed for every model of skeleton of arrow, and that which produces the minimum Hasudorff distance is the recognized model. In this section it has been showed that the distance is not computed for all the image and that it is not considered for all the possible relative positions of the model over the real image. Instead of this strategy, the Hausdorff distance is computed between five models and each object detected in the stage of shape classification. This kind of process produces and efficiently computing Hasudorff distance. Once the Hasudorff distance has been computed for every model and the lowest distance has been selected, the algorithm compares this distance with a threshold τ, and if $H(I,M) \geq \tau$ the algorithm determines that the object analyzed isn't a mandatory traffic sign.

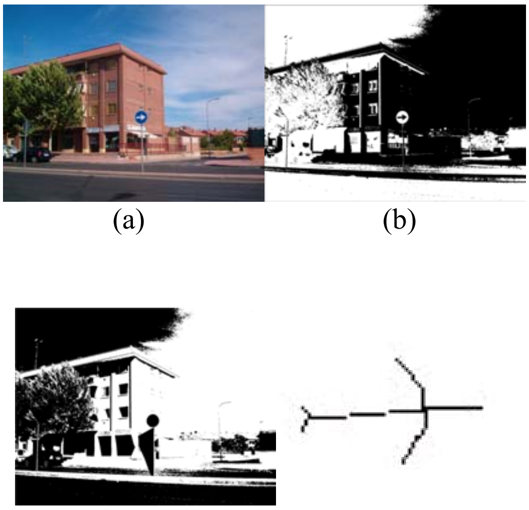
2.4 Traffic sign recognition process.

The algorithm described allows to recognize the mandatory traffic signs. A complete recognition process is described and showed in this section:

- The image is captured by a camera and it is segmented like it has been described (Fig.10 (a)).
- The shape classification is executed. Fig. 10 (b) shows the results of this step. In this image of Fig. 10 (b) the shapes that have been detected appear filled.
- The next step cleans all the image zones that don't belong to any object detected Fig. 10 (c).
- The thinning algorithm is executed (Fig. 10 (d)).
- Then the Hasudorff distance is computed and a model is recognized.

3 Experimental Results

In this section we present some real experimental results obtained with the algorithm designed. The algorithm presents very good results for all the signs captured in normal conditions of illumination. Fig. 10 shows a correct process of recognition. Fig. 11 shows an other real image and that the mandatory signal recognized is correct. These two images, and all the others computed, produce a Hausdorff distance for the model recognized lower than 15. After these experimental results we have assigned to the threshold τ the value 20. If a recognition process produces a Hausdorff distance higher than τ we can assume that this sign is not a mandatory traffic sign.



(a) (b)

(c) (d)

Fig. 10 Results of each stage of the recognition process.

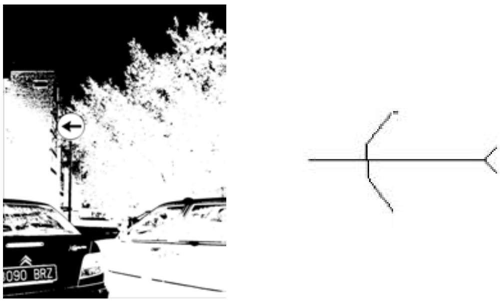


Fig. 11 Results obtained for a real image.

4 Conclusions

This work presents a new efficient method for mandatory signal recognition. The experimental results obtained are very satisfactory. The problem of this method is the time inverted for computing the Hausdorff distance when the objects are too big. The direction of our future work will be, first, the reduction of computational time for big objects performing improvements in the Hausdorff distance algorithm. The second future work will be to prepare the Hausdorff distance algorithm for comparing portions of shapes, and then the algorithm will allow us to recognize signs with occlusions. Other future work will be to elaborate another algorithm based on the Hausdorff distance for other kind of traffic signs.

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

References

[1] 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, *ACIVS'04,* August 2004

[2] P.Gil-Jiménez S.Lafuente-Arroyo H.Gómez-Moreno F.López-Ferreras and S.Maldonado-Bascón, Traffic Sign Shape classification evaluation II: FFT applied to the signature Blobs, *Proc. IEEE Intelligent Vehicles Symposium Las Vegas U.S.A.,* 2005, pp. 607-612.

[3] A. de la Escalera J.M. Armigol and M. Mata , Traffic sign recognition and analysis for intelligence vehicles, *Image and vision computing,* Vol. 21, 2003, pp.247-258.

[4]C.Y. Fang C.S. Futh P.S. Yen S. Cherng and S.W. Chen, An automatic road sign recognition system based on a computational model of human recognition processing, *Computer Vision and Image Understanding,* Vol. 26, 2004, pp. 237-268.

[5] D.P. Huttenlocher G.A. Klanderman and W.J. Rucklidge, Comparing Images Using the Hausdorff Distance, *Pattern Analysis and Machine Intelligence IEEE* ,Vol 15, Issue 9, Sept. 1993, pp. 850-863.

[6] R.C. González and R.E. Woods, *Digital Image proccesing,* Addison Wesley, 1993.

[7] E.R. Davies and A.P.N. Plummer, Thinning algorithms: a critique and a new methodology, *Pattern Recognition 14,* 1981, pp 53-63.

[8] D.W. Paglieroni, Distance transform: Properties and machine vision applications, *Comput. Vision Graphics Image Processing: Graphical Models Image Processing*, Vol. 54, No. 1, 1992, pp. 56-74.

[9] G.Borgefors, Distance transforms in digital images, *IEEE Trans. Patt. Anal. Machine Intell.,* Vol. PAMI-8, 1986,pp. 344-371.

[10] P.E. Danielsson, Euclidean distance mapping, *Comput. Graphics Image Processing,* Vol. 14, 1980, pp. 227-248.