Automatic Recognition of Bidimensional Models Learned by Grammatical Inference in Outdoor Scenes

Alberto Sanfeliu(*) and Miguel Sainz(**)
(*) Instituto de Robotica e Informatica Industrial, (**)Instituto de Cibernética
Universidad Politécnica de Catalunya - CSIC
Diagonal 647, 08028 Barcelona
(*)sanfeliu@ic.upc.es, (**)sainz@ic.upc.es

Abstract

Automatic generation of models from a set of positive and negative samples and a-priori knowledge (if available) is a crucial issue for pattern recognition applications. Grammatical inference can play an important role in this issue since it is one of the methodologies that can be used to generate the set of model classes, where each class consists on the rules to generate the models. In this paper we present the recognition methodology to identify models in a outdoor scenes generated through a grammatical inference process. We will summarize how the set of model classes are generated and will explain the recognition process. An example of traffic sign identification will be shown.

1 Introduction

Techniques for automatically acquiring shape models from sample objects are presently being researched. At present, a vision developer requires to select the appropriate shape representation, design the reference models using the chosen representation, introduce the information and program the application. This methodology is used in industrial applications, since there is not any other available. However, it is cumbersome and impractical when dealing with large set of reference models. The recognition systems in the future must be capable of acquiring objects from samples with limited human assistance.

There exist few approaches to automatically acquire generic models. Some of them are based on neural networks [6], appearance representation [9] and grammatical inference [13], [12].

In this paper we will explain the method to recognize objects acquired by grammatical inference, where the model is a bidimensional grammar or language.

2 Summary of the model representation and generation

In previous works [13] and [12], we explain how we automatically generate models of bidimensional objects in outdoor scenes, from true color images and through a two step process based on the *Active Grammatical Inference* methodology. The output of the process is a two level context sensitive language which represent each model class.

The formal representation of a bidimensional model is:

Definition 2.1 A pseudo-bidimensional Augmented Regular Expression (or PSB-ARE) is a four-tupla (Σ_R, V, T, L), where Σ_R is the set of the row *ARE's* [1], V is the associated set of *star variables*, T is the associated *star tree*, and L is a set of independent linear relations $l_1 \ldots l_{nc}$, each envolving the variables in V. If the set L is given by partitioning the set of star variables V into two subsets V^{ind}, V^{dep} of independent and dependent star variables, respectively, and expressing the latter as linear combinations of the former:

$$l_i \equiv v_i^{ind} = a'_{i1} \cdot v_1^{ind} + \ldots + a'_{ij} \cdot v_{1j}^{ind} + \ldots + a'_{i(ni)} \cdot v_{ni}^{ind} + a'_{i0}, \quad for 1 \le i \le nc$$

where *ni* and *nc* are the number of independent and dependent star variables, respectively. R, V and T are described in detail in [1].

The definition of V restricts the allowed values for the star variables to natural numbers, $\forall\ k \in [1,ns]$: $v_k \in N$. Consequently, the set of linear relations L is only well-defined when the involved variables take natural numbers as values. This also implies that some of the star variables may be implicitly constrained to a smaller range inside the natural numbers (e.g. $v_k \ge z$, $z \in N$; v_k always odd; v_k always even; etc..). Moreover, the coefficients a_{ij} (or a'_{ij}) of the linear relations will always be rational numbers.

The PSB-ARE can be seen as a column ARE of the ARE's of each row. Fig. 1 shows the PSB-ARE of the a traffic sign. We call this representation as pseudobidimensional ARE due that there are two levels of ARE. The first level it is the row level where each row of the model is represented by an ARE. The second level is the column level, where all the columns are represented only by one ARE which terminal symbols are the row ARE's. This type of representation is a nonsymmetric representation which can be automatically generated by string grammatical inference methods.

From the point of view of identification of models in a scene, this representation allows to describe models irrespectively of the scale and position of the object in the scene. Moreover it permits to identify partially occluded objects as well as objects with distorsions. However the representation does not allow to represent models at different angles of rotation. The generation of the PSB-ARE models is based in a five step procedure which is explained elsewhere [12]

3 Recognition process

The recognition process is based in two criteria: (1) minimize the preprocessing phase;

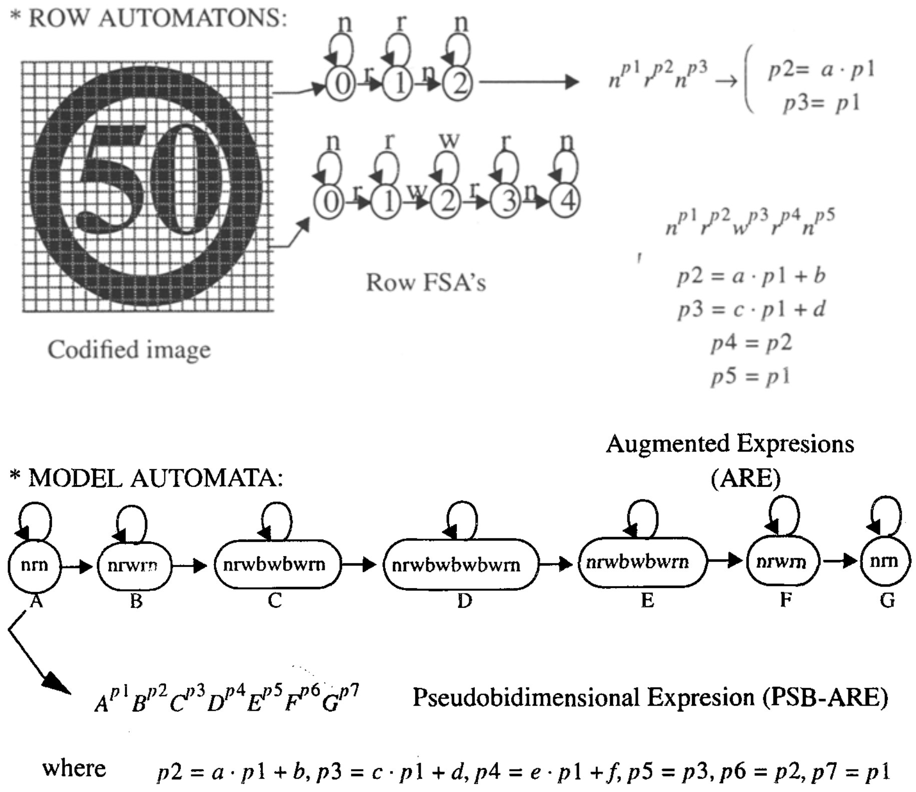

* ROW AUTOMATONS:
Codified image
Row FSA's
$n^{p1} r^{p2} n^{p3} \rightarrow p2 = a \cdot p1,\ p3 = p1$
$n^{p1} r^{p2} w^{p3} r^{p4} n^{p5}$
$p2 = a \cdot p1 + b$
$p3 = c \cdot p1 + d$
$p4 = p2$
$p5 = p1$
Augmented Expresions (ARE)
* MODEL AUTOMATA:
nrn
nrwrn
nrwbwbwrn
nrwbwbwbwrn
nrwbwbwrn
nrwrn
nrn
A
B
C
D
E
F
G
$A^{p1} B^{p2} C^{p3} D^{p4} E^{p5} F^{p6} G^{p7}$
Pseudobidimensional Expresion (PSB-ARE)
where $p2 = a \cdot p1 + b, p3 = c \cdot p1 + d, p4 = e \cdot p1 + f, p5 = p3, p6 = p2, p7 = p1$

Figure 1 Model Description

and (2) maximize the analysis phase. We can apply these criteria since the model learning process is done at the raw level without requiring any process to extract specific structural relations of the models. We understand that one way to reduce the distorsions introduced by the preprocessing phases and speed up the identification of models is to eliminate as much as possible the early phases and maximize the identification phase. For this reason we do a very simple preprocessing to the image and we work harder in the analysis phase.

We have reduced the preprocessing phases to a low level segmentation, location of the object to be identified and codification. Each one of these phases and the analysis phase is explained below.

3.1 Segmentation

The objective of the segmentation phase is to leave the image prepared to detect potential locations of candidate objects. For this reason we can use very simple methods and a coarse segmentation process. This last issue is very important for the identification of objects in outdoors images since the segmentation process is usually very sensitive to uncontrollable variables.

We use a three layered neural network to do the segmentation process. As inputs we consider a window of 4x4 color pixels (each one represented by the three channels: red, blue and green), which implies 48 inputs to each neural unit. The neural network has 10 neural units in the first layer, and 10 and 5 neural units in the second and third layers respectively. The five outputs corresponds to 5 significant different image features: road, sky, white colors(clouds and road lines), grass and traffic signs (in this work we learn and detect traffic lines with red ring). We apply the neural recognition processing to every window of 4x4 pixels and the result is set in the segmented image. Examples of this process can be seen in [13]. The pixels detected as traffic sign elements will be used in the next processes. The other classified elements can be used to eliminate uncertainties in the location of the traffic signs, however they have not been used in this work.

3.2 Location of the potential objects in the scene

The next phase is to locate the areas, denoted as *candidate-area*, where a potential object can be identified as one of the reference models. Since in this case we are looking for traffic sign reference models, the image pixels identified as traffic sign pixels will be the seed of a candidate-area. We look for the limits of each candidate-area by looking a closed area with a minimum number of pixels. Once located, the rectangle area which circumscribe the area is computed. Although the candidate-area could be ill located, for example a small portion of the total candidate-area is located, the matching process will try to find the best match.

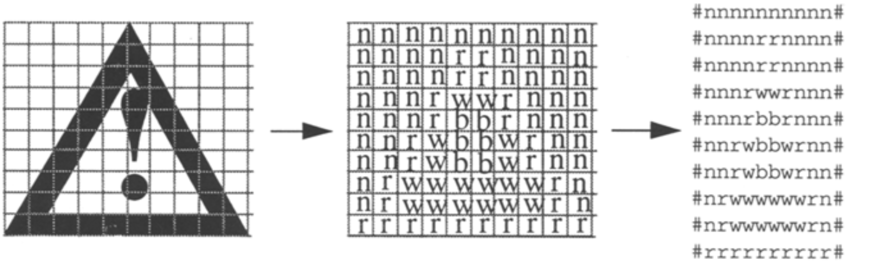
3.3 Codification of the candidate-area

The segmentation phase codifies the image in a set of classes which in principle can be used for the codification of the candidate-object. However, our tests have shown that this segmentation does not allow to do a fine codification of the candidate-area. For this reason we do a new segmentation on the candidate-area with the objective to find the best codification of the candidate symbols. This segmentation is again done by identifying each pixel by means of a neural network which has been previously trained with a large number of samples (the number of classes depends on the number of terminal symbols used to describe the reference model). We use the same architecture described in the segmentation phase and the same learning algorithm.

An example of this codification for a traffic sign can be seen in Figure 2

3.4 Looking for the candidate pattern seed transitions

In this work we represent each reference model by means of a context sensitive



n	n	n	n	n	n	n	n	n	n
n	n	n	n	r	r	n	n	n	n
n	n	n	n	r	r	n	n	n	n
n	n	n	r	w	w	r	n	n	n
n	n	n	r	b	b	r	n	n	n
n	n	r	w	b	b	w	r	n	n
n	n	r	w	b	b	w	r	n	n
n	r	w	w	w	w	w	w	r	n
n	r	w	w	w	w	w	w	r	n
r	r	r	r	r	r	r	r	r	r

```
#nnnnnnnnnn#
#nnnnrrnnnn#
#nnnnrrnnnn#
#nnnrwwrnnn#
#nnnrbbrnnn#
#nnrwbbwrnn#
#nnrwbbwrnn#
#nrwwwwwwrn#
#nrwwwwwwrn#
#rrrrrrrrrr#
```

Traffic sign window Primitive found Chain code extraction

Figure 2 Codification of a trafic sign

language, which have the language structure to generate the samples of a specific model. However the language do not incorporate the potential distortions at the low and high level (for example due to noise or partial occlusion of the reference model). In order to taken into account these distortions, the parser or the matching process must compute a similarity measure [3], [4]. Since our language is context sensitive and at present there is not an efficient error correcting parser, we have developed a new strategy. The strategy consist on three steps: (1) look in the image for the candidate pattern seed transitions; (2) generate the best reference models in accordance with the first step; and (3) do the matching process using the Leveshtein [5] measure distance. With this strategy we overcome the problem of distortions and partial occlusions with a low computational processing time.

In this section we explain how we use candidate pattern seed transitions to detect potential objects in the image. One or several seed transitions are preestablished for each reference model in the learning phase and are selected based on several criteria, basically to find robust transitions in the image (robustness is required since that the image noise is very common in outdoors images). A seed transition is a portion of the context sensitive language of the reference model which encompasses several rows. The left side of Figure 3 shows the seed transition parameters and the clue generated of the model shown in Figure 1. The parameter *k* is used to determine the number of adjoining transitions considered in the seed transition.

The process tries to find the seed transitions although there exist noisy pixels in the image. The image is scanned from left to right and top to bottom looking for a row portion which matches with the row seed transition. The seed transitions which have got a good match (surpass a threshold) continue the same process with the other rows. A seed transition can be found in several locations in an image, and this number depends very much on the selection of them. The right side of Figure 3 shows the two zones where the seed transition shown in the left has been found. Pay attention in the resulting found targets, the seed transitions are found and the number of repetitions are computed (the noisy pixels are automatically withdrawn and substituted by seed symbols). This targets will be play an important role in the following sections.

3.5 Generation of the reference models for matching

In order to find the reference objects in the image and since we use the Levesthein distance measure to obtain the best candidate match, the algorithm must generate the reference models in accordance with the results of the previous section. In the last section we got as found targets (see Figure 3) the seed transitions with the number of symbol repetitions as exponent, for example $r^4\, w^6$. The seed exponents are used in this section to estimate the size (width and height) of the reference model which has to be generated. Since a seed can be found in several locations with diverse exponents, the outcome of this process must be all the sizes of the reference models detected at the previous section. Moreover the location of the object in the image with respect to the candidate reference model is again recomputed to obtain it more precisely. The number of scaled references models to be used in the matching process depends again basically in the number of seed transitions. If the seeds do not make a clear distinction between reference models and diverse portions of them, the method will find a large number of candidate reference scaled models for the matching process.

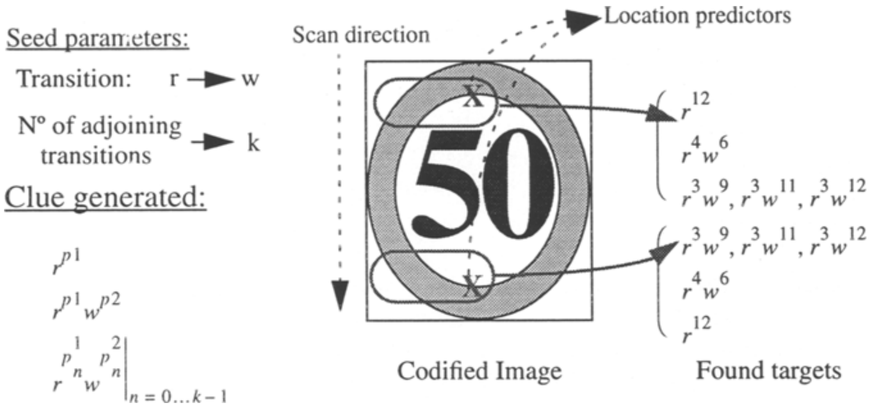


Figure 3 Seed transitions and their location in a candidate-ARE

3.6 The matching process

In this phase each one of the candidate reference models obtained in the previous section is compared with the object identified at the recomputed locations. The method computes the Levensthein distance measure $d(row^a_{ki}, row^m_{ji})$ row by row and obtains the following measures (where row^a_{ki} is the row i of the image in the candidate-area a_k, and row^m_{ji} is the row i of the reference model m_j):

Pattern error ε_P: this error computes the average error through all the rows of a candidate-area a_k and a model m_j,

$$\varepsilon_P = \frac{d\left(row_i^{a_k}, row^{m_j}\right)}{Nrows_{a_k}}$$

where $Nrows_{ak}$ is the number of rows of a_k.

Area error ε_A: this error computes the average error of a consecutive set of rows which overcomes a predetermined threshold t_A,

$$\varepsilon_A = \frac{\sum d\left(row_i^{a_k}, row^{m_j} \right)}{Nrows_{a_{k,}}}$$

where Σ is over a limited area a_{kl} which rows have a $\varepsilon_P > t_A$

The comparison is done first at the *Pattern error* level and then at the *Area error* level. If two candidate reference models have similar ε_P values, the decision is taken using the ε_A error. Since the reference model has been represented only through the rows, there could be that some objects which difference at the column level be erroneously identified. This problem can be solved by doing two types of representation, one by rows and another by columns. The same global process can be applied by taken into account that the rows are columns and viceversa.

Some results of the segmentation process are shown in Figure 4. On the left side, there are 3 color images of 512x512 pixels and on the right side there results of the neural net segmentation process. In Figure 5, an error matching table is shown. The values shown are the shape error and the most significant area errors and their location (is percentual area location with its origin at the top of the candidate).

4 Conclusions

This work presents a flexible method to identify reference models in outdoors color images where the models are represented by a pseudobidimensional context sensitive language and that they are learned by an automatic grammatical inference procedure. The procedure emphasizes the analysis process against the preprocessing in order to avoid the classical problems that appear in outdoor images, lack of robustness in the preprocessing phases and difficulty to describe outdoors models. The method identify objects irrespective of their size, location or partial occlusion. The method is based on the use of error correction at the level of prototypes instead of languages, although the reference models are represented by context sensitive languages. The key issue is the use of seeds to find the candidate reference models in image and from them to estimate the reference model. The method has been applied to identify traffic sign which where previously learned by a grammatical inference procedure and the outcome looks very promising.

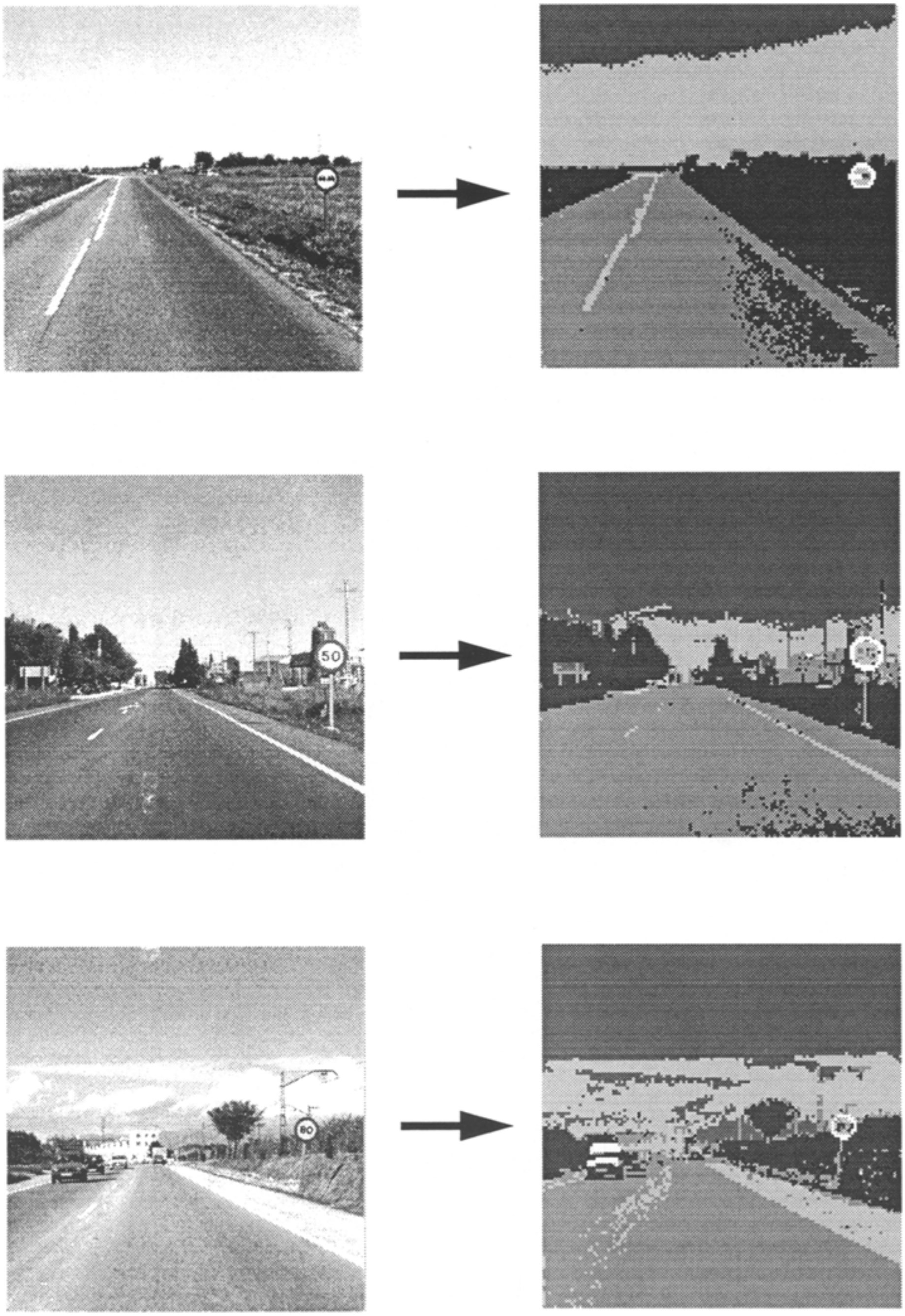


Figure 4 Segmentation Results

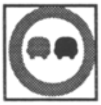
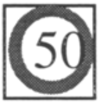
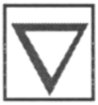
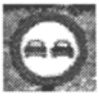
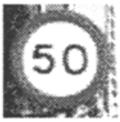
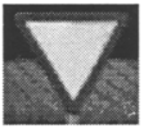
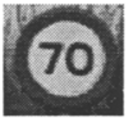
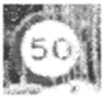
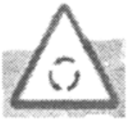
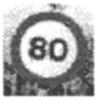
MODELS CANDIDATES						
	SHAPE	AREA	SHAPE	AREA	SHAPE	AREA
	18.301	(5%-11%) 13.514 (33%-38%) 13.514 (41%-55%) 23.784 (78%-100%) 36.036	32.821	(0%-15%) 14.000 (25%-100%) 39.250	38.318	(4%-27%) 56.481 (32%-75%) 54.929
	25.379	(0%-100%) 25.379	5.775	(42%-63%) 12.857	79.942	(4%-65%) 80.713
	47.306	(5%-100%) 48.488	34.930	(0%-60%) 61.127	25.859	(0%-7%) 28.852 (8%-15%) 22.678 (16%-60%) 48.256
	36.217	(4%-100%) 37.041	25.022	(0%-10%) 18.878 (40%-100%) 32.177	39.047	(0%-67%) 60.072
	32.508	(0%-13%) 17.568	24.671	(0%-13%) 17.949 (16%-27%) 14.744 (37%-100%) 30.983	34.218	(3%-76%) 46.559
	35.000	(0%-100%) 35.000	38.295	(0%-23%) 43.213 (32%-100%) 41.832	52.894	(0%-100%) 52.894
	23.799	(16%-33%) 24.320 (36%-69%) 43.443	15.681	(20%-31%) 13.194 (40%-74%) 24.074 (85%-94%) 19.444	62.892	(0%-100%) 62.892

Figure 5 Model Matching Results

5 References

[1] R. Alquezar and A. Sanfeliu, "Augmented regular expressions: a formalism to describe, recognize and learn a class of context -sensitive languages", *Research Report LSI-95-17-R*, Universitat Politecnica de Cataluyna, Barcelona (1995).

[2] R. Alquezar and A. Sanfeliu, "An algebraic framework to represent finite-state machines in single-layer recurrent neural networks", *Neural Computation*, **7**, Sept.(1995).

[3] K.S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, New York, (1982).

[4] H.Bunke and A. Sanfeliu, *Syntatic and Structural Pattern Recognition: Theory and Applications*, World Scientific, (1990).

[5] V.I. Levensthein, "Binary codes capable of correcting deletions, insertions and reservals", *Sov. Phys. Dokl, 10 (8), 707-10,* Feb (1966).

[6] W. Lei and N. M. Nasrabadi, "Invariant object recognition on neural network of cascaded RCE nets", *Int. Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7, No.4, pp 815-829, (1993).

[7] L. Miclet, "Grammatical inference," in *Syntatic and Structural Pattern Recognition: Theory and Applications*, H.Bunke and A.Sanfeliu, Eds., World Scientific, 1990.

[8] H. Murase and S.K. Nayar, "Learning object models from apearance", Proc. of AAAI, Washington D.C., July 1993.

[9] H. Murase and S.K. Nayar, "Visual learning and recognition of 3D objects from appearance", *International Journal of Computer Vision*, Vol. 14, No.1, pp 5-24, January, 1995.

[10] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning internal representations by error propagation", in D.E. Rumelhart & J.L. McClelland (eds.) *Parallel Distributed Processing: Explorations in Microstructure of Cognition Volo 1: Foundations*, MIT Press (1986).

[11] A. Sanfeliu and R. Alquezar, "Active Grammatical Inference: a new learning methodology", in *Shape and Structure in Pattern Recognition*, D. Dori and A. Bruckstein (eds.), World Scientific Pub., Singapore (1995).

[12] A. Sanfeliu and M.Sainz, "Aprendizaje automatico de modelos en vision por computador", *Proceedings of the XVI Jornadas de Automatica*, San Sebastian, 27-29 Sept, (1995).

[13] M. Sainz and A. Sanfeliu, "A first approach to learn the model of traffic signs using connectionist and syntactic methods", *Proceedings of the VI Simposium de Reconocimiento de Formas y Analisis de Imagenes*, Cordoba, 3-6 April (1995).