

# Automatic Extraction of Linguistic Models for Image Description

Illuminada Baturone and Andrés Gersnoviez

**Abstract**—This paper describes a methodology to extract fuzzy models that describe linguistically the low-level features of an image (such as color, texture, etc.). The methodology combines grid-based algorithms with clustering and tabular simplification methods to compress image information into a small number of fuzzy rules with high linguistic meaning. All the steps of the methodology are carried out with the help offered by the tools of Xfuzzy 3 environment, so we can define, simplify, tune and verify the fuzzy models automatically. Several examples are included to illustrate the advantages of the methodology.

## I. INTRODUCTION

THE inherently imprecision and ambiguity of images make them very suitable for fuzzy logic-based solutions. This happens at low level, due to noise and possibly low resolution and quality of the images, as well as at high level, because edges of objects or regions, their distances, relative positions, etc. can be modeled by fuzzy concepts [1]-[2]. Another relevant advantage of fuzzy logic is that heuristic knowledge expressed linguistically can be easily translated into fuzzy rule bases and, vice versa, given fuzzy rules, a strong linguistic meaning can be obtained. The latter issue has been exploited to describe images with natural or natural-like language, which has been proven to be feasible and effective in many applications such as image indexing and retrieval and pattern recognition. In image retrieval, linguistic descriptions may fill the gap between the information stored about the images and that required to retrieve [3]-[4]. In recognition processes, a linguistic model repository may cope with the variability of the patterns. Linguistic descriptions of images have also been exploited to construct cognitive vision systems for surveillance purposes or autonomous robotics [5].

Most linguistic models deals with abstract concepts (features, objects, etc.) that have been extracted from the images after a preprocessing process. Other few models deal with low-level image features, such as color or grey values. The latter is the case of the works in [6]-[7], which apply this solution for handwritten character recognition.

The focus of this paper is also to extract linguistic models

from low-level features. The idea is to understand the image as a set of pixels that contain numerical features: a luminance value (in case of grey images); red, green, and blue values (in case of RGB images), a texture value (in case of images processed by texture filters), etc [8]. Hence, an image is a set of numerical data ordered by the horizontal, x, and vertical, y, pixel positions. In this sense, the models addressed herein are models with two inputs (x and y) and one or more outputs. They consist of simple ‘if-then’ rules capable of summarizing which image regions has certain luminance (‘dark’, ‘clear’, ...), colour (‘skin-like’, ‘blood colour’, ...), and/or texture (‘coarse’, ‘high-contrast’, ...).

The paper is organized as follows. Section II summarizes the methodology proposed to obtain a linguistic model from an image. Peculiarities of fuzzy linguistic modeling are depicted. Section III describes briefly how the proposed methodology can be carried out automatically with the CAD tools of Xfuzzy 3, a design environment developed at the Microelectronics Institute of Seville and University of Seville [9]. Section IV illustrates the methodology with several examples of colour images, and describes the use of the generated models into a pattern recognition system. Finally, conclusions are given in Section V.

## II. LOW-LEVEL LINGUISTIC MODELS

Considering an image as a set of numerical data, the problem of extracting a fuzzy logic-based model for the image is the problem of extracting fuzzy rules from numerical data, which is a well-known problem in the field of knowledge discovering.

### A. Clustering versus grid-based modeling

Two main strategies can be followed to extract rules from data: clustering and grid-based fuzzy rule learning. Clustering techniques (in particular fuzzy clustering) have been widely applied for images, mostly for processing rather than for modeling purposes. For creating models, clustering techniques organize the numerical data into clusters and use them to generate rules. Each cluster generates a rule by projections into each variable [10]-[11], so that rules and membership functions are extracted simultaneously. Clustering techniques offer the advantage of generating usually a low number of rules. As a drawback, the resulting rules do not contain a high linguistic meaning. This is why several authors have proposed techniques to enhance its linguistic interpretability [10], [12].

In the other side, grid-based algorithms generate a partition of the input and output variables and then create the

This work was supported in part by the Spanish Ministerio de Ciencia y Tecnología under the Projects TEC2008-04920 and DPI2008-03847, and by Junta de Andalucía under the Project P08-TIC-03674.

I. Baturone is with the University of Seville and the Microelectronics Institute of Seville (CNM-CSIC), c/ Américo Vespucio s/n, 41092, Seville, Spain (phone: +34-954466666; fax: +34-954466600; email: [lumi@imse-cnm.csic.es](mailto:lumi@imse-cnm.csic.es)).

A. Gersnoviez is with the University of Córdoba, Córdoba, Spain (e-mail: [andresgm@uco.es](mailto:andresgm@uco.es)).

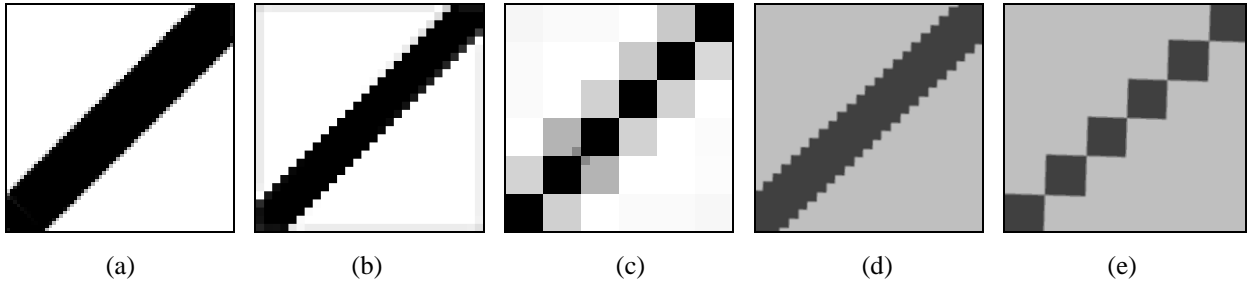


Figure 1: (a) Original image. The same image with 28x28 (b) 6x6 pixels (c). Models obtained from the original image with Wang-Mendel technique with 28x28 (d) and 6x6 rules.

rules [13]. Each numerical data generates the rule which is the most activated by it. If the rules generated share the same antecedents but have different consequents the conflict is solved by selecting the rule with the highest activation value and eliminating the rest. These algorithms obtain rules with high linguistic meaning. The drawback is that the number of rules grows exponentially with the number of inputs, so that they are adequate for models with a few inputs.

Since our objective is to obtain linguistically expressive models with two inputs, grid-based techniques have been employed.

### B. Grid-based modeling of images

Grid-based techniques (in particular Wang-Mendel technique [13]) have been widely applied in control applications to interpolate the actions of a human expert. Two advantages of describing a control action by a rule base are its linguistic meaning and interpolation capability, so that the rules summarize the most relevant control actions to take. In the control domain, defuzzification methods such as center of gravity or Takagi-Sugeno inference methods are employed so that the fuzzy rules activated by the controller inputs are aggregated accordingly to their activation degree in order to provide a smooth control action.

In the case of images, the situation is different. No smoothing is usually pursued in the model since this could perform some kind of filtering and lost of relevant information (such as edges). Hence, the defuzzification method usually employed provides as output the consequent of the most activated rule, which will be called MaxLabel method herein. Wang-Mendel technique applied to an image (considered as a set of  $x$ ,  $y$ , and pixel values) extracts a complete rule base, since there are numerical data for each pixel position  $(x, y)$ . A complete rule base with the MaxLabel method as defuzzification is equivalent to a complete non-fuzzy rule base, that is, to a rule base whose antecedents are rectangular membership functions. As a matter of fact, the shape of the membership functions has not influence on the result but only the intersection point between membership functions, that is, the value for which one function has higher membership degree than its neighbors (this is why they can be rectangular). Moreover, the t-norm employed to connect the antecedents has neither influence on

the result (since a t-norm is a monotonous function).

As a result, given an image of resolution  $M_o \times N_o$  pixels, a model obtained with Wang-Mendel technique with  $M_m \times N_m$  rules ( $M_m$  membership functions for the  $x$  variable and  $N_m$  functions for the  $y$  variable) is similar to a low-resolution version of the image with  $M_m \times N_m$  pixels (since  $M_m < M_o$  and  $N_m < N_o$ ). This is illustrated in Fig. 1 for a simple image of a diagonal line. Fig. 1a shows the original image with a resolution of 280x280 pixels. Fig. 1b and c show the same image but with 28x28 and 6x6 pixels, respectively. Fig. 1d and e show the model obtained with Wang-Mendel technique with 28x28 and 6x6 rules, respectively (the level of grey is only symbolical). The latter model contains linguistically interpretable rules such as:

IF  $x$  is 'quite at the left' AND  $y$  is 'quite at the bottom' THEN the image is 'dark'

Where the concepts 'quite at the left' and 'quite at the bottom' can be represented by either fuzzy sets or crisp intervals.

### C. Non-complete grid-based models

In case of classification-type fuzzy models, such as the commented above, a complete rule base contains so much information that there is no place for ambiguity or uncertainty. Hence, using fuzzy sets in the antecedents is not required and would be more complicated than using crisp intervals.

Let us consider again the example in Fig. 1a and the model in Fig. 1e with 6x6 rules. It can be seen that not all the 36 rules of the model provide true information. As depicted in Fig. 1c, the 10 squares in the frontier between the diagonal line and the background, should be neither quite dark nor quite clear. In the case of Fig. 1e, those squares are clear because the pixels that achieve the highest activation degree of each rule responsible of each square region have a clear value. Wang-Mendel technique only takes into account the pattern with the highest activation degree and does not consider that many other patterns provide dark results for the same region. In order to consider this fact, several authors have proposed rule extraction techniques that qualify the rules by performance indexes [14]-[15]. For example, the

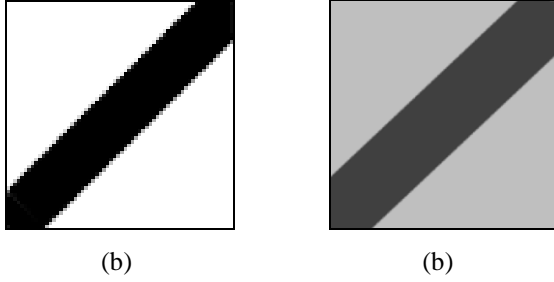


Figure 2: (a) Original image. (b) Model obtained from the original image with 26 instead of 36 rules.

index proposed in [14] is:

$$P_R = \frac{1}{|L|} \sum_{p \in L} a_R(p) \cdot e_R(p) \quad (1)$$

$$e_R(p) = \begin{cases} 1 & \text{if } p \text{ is well classified by } R \\ -1 & \text{otherwise} \end{cases}$$

Where  $L$  is the pattern set from which extracting the rules and  $a_R(p)$  is the activation degree of rule  $R$  for pattern  $p$ .

If rules are ordered by their performance indexes, a non complete rule base can be obtained by selecting the best rules or, similarly, eliminating the worst rules. As a matter of fact, rules with low performance index do not model correctly the image and should be better eliminated. If the model is crisp, a non complete rule base contains regions with no output. However, if the model is fuzzy, the regions with no specific rules are absorbed by the neighbor ones. Hence, fuzzy sets are interesting for the antecedents of non-complete rule bases. In the example of Fig. 1e, if the 10 rules of the frontier are eliminated, and the antecedents are fuzzy, the result is shown in Fig. 2. The fuzzy model with 26 rules, being simpler, provides a result more similar to the original image. In this sense, non complete fuzzy rule bases can provide better models than crisp systems.

#### D. Simplification of grid-based models

Even using non complete rule base and only 2 inputs, the

$x_1$	$x_2$	$x_1$	$x_2$	$x_1$	$x_2$	$x_1$	$x_2$	$x_1$	$x_2$	$x_1$	$x_2$	
1	1	✓	1	1,2	✓	1	1,2,3	✓	1,2	1,2,3	1,2,3,4,5	1
			1,2	1	✓	1,2	1,2	✓	1,2,3,4	1	✓	
						1,2,3	1	✓				
1	2	✓	1	2,3	✓	1,2	2,3	✓	2,3,4,5	1	✓	
2	1	✓	1,2	2	✓	2	1,2,3	✓				
			2	1,2	✓	2,3,4	1	✓				
			2,3	1	✓							
1	3	✓	1,2	3	✓	3,4,5	1	✓				
2	2	✓	2	2,3	✓							
3	1	✓	3,4	1	✓							
2	3	✓	4,5	1	✓							
4	1	✓										
5	1	✓										

Figure 4: Minimization table of tabular simplification.

$x_1 \backslash x_2$	(1)	(2)	(3)	(4)	(5)
	L	QL	FRN	QR	R
(1) VCL	S	S	S	B	B
(2) CL	S	S	S	B	B
(3) MED	S	M	M	VB	VB
(4) FAR	S	VS	VS	VS	VB
(5) VFAR	S	VS	VS	VS	VB

Figure 3: Example of tabular simplification.

number of rules extracted by grid-based techniques is high. A method to simplify them is to apply tabular simplification to each set of rules with the same consequent. This method, proposed in [16], is an extension of Quine-McCluskey algorithm employed in Boolean design because neither consequents nor antecedents are bivalued in fuzzy design.

Fig. 3 illustrates how this tabular simplification transforms a 2-input rule base with 25 rules into 7 rules with more linguistic content. In particular, the procedure to transform the 9 rules with the same consequent 'S' into 2 simpler rules is shown in Fig. 4. This procedure consists of the following steps:

1.- The membership functions of the input variables are represented by ordered natural numbers, so that the antecedent part of each rule can be represented by the sum of its constituent antecedents. Those antecedent parts with the same sum form a group, and the groups are listed in an increasing order of sums. Each antecedent part is like a minterm in the Boolean design.

2.- A search is performed between adjacent groups to combine those antecedent parts that only differ in the membership function of one input and it is the neighbor at the right or left. Each combination is similar to an implicant in Boolean design. The implicants obtained are grouped in a new list and a new combination is searched to generate another list (five lists are shown in Fig. 4). The final result

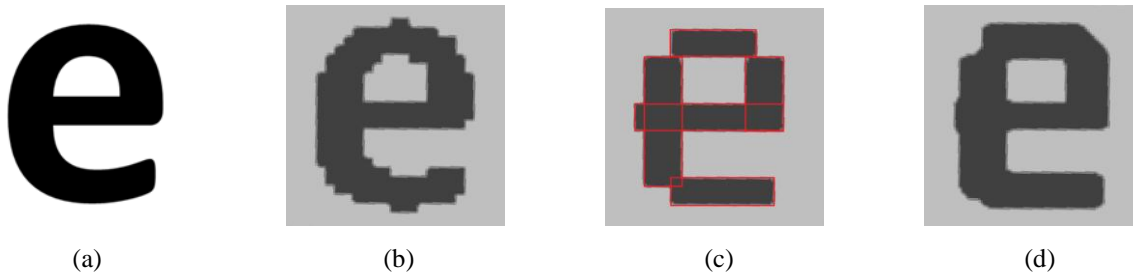


Figure 5: (a) Original image. (b) Model obtained from the original image with 625 rules. (c) Selection of the 5 more relevant prime implicants for the dark region. (d) Model with 12 rules.

are the prime implicants (the two groups of ‘S’ depicted in Fig. 3).

3.- The minimum number of prime implicants covering all minterms are selected. This step begins with the biggest prime implicants so as to find the most generic rules (in the example of Fig. 4, the 2 prime implicants found should be selected).

4.- The resulting rules can be expressed with a high linguistic meaning by using linguistic hedges such as ‘equal or greater than’, ‘equal or smaller than’, and ‘not equal to’. In the example of Fig. 4, the 2 selected rules will be expressed as: ‘if  $x_2$  is L then out is S’, and ‘if  $x_1$  is equal or smaller than CL and  $x_2$  is equal or smaller than FRN then out is S’.

After performing tabular simplification, a further simplification can be carried out if a simpler model is pursued. This simplification consists in qualifying the prime implicants with a performance index and selecting a given number of the best of them or, similarly, eliminating a given number of the worst. The best prime implicants are those which cover the maximum number of minterms (as commented in the step 3 of the procedure). The following in importance are those which cover more minterms not already covered. Fig. 5 illustrates the result of this procedure for an image of the letter ‘e’. The initial model with 625 rules (25x25) is transformed by tabular simplification into a model with 45 rules (18 for the dark regions and 27 for the clear regions). Further simplification on prime implicants allows a model with only 12 rules (5 for the dark and 7 for the clear

regions). The resulting model contains the rules that define clearly which regions belong to the ‘e’ and which others do not clearly belong to the ‘e’. For instance, two rules of this model are:

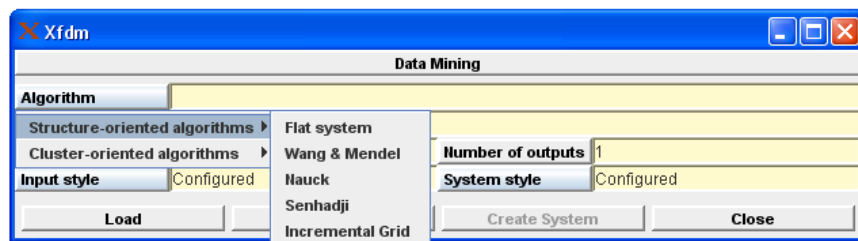
```
IF y is 'quite at the bottom' OR y is
'quite at the top' THEN the image is 'clear
(not belong to 'e')
IF y is 'at the middle' and x is 'from the
left to the right' THEN the image is 'dark'
(belong to 'e')
```

The model is fuzzy because both groups of rules are not complement, that is, there are regions that neither belong nor not belong explicitly.

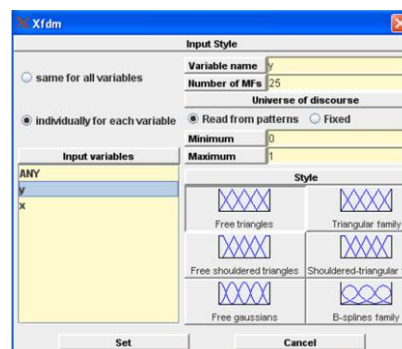
### III. AUTOMATIZATION OF THE TECHNIQUE

Extraction of linguistic models as described above can be performed automatically with the CAD tools of Xfuzzy 3 design environment, developed at the Microelectronics Institute of Seville and University of Seville [9].

Extracting fuzzy rules from numerical data can be carried out by the tool *xfdm*. The graphical user interface of this tool allows selecting several clustering and grid-based algorithms (Fig. 6a). Among the later, the user can select Wang-Mendel technique and techniques that consider performance index of the rules, such as those proposed by Nauck and Senhadji [14]-[15], which do not extract complete rule bases. The tool requires a file with the numerical information from which extract the rules, and definition of the input style (number of



(a)



(b)

Figure 6: Windows of *xfdm*.

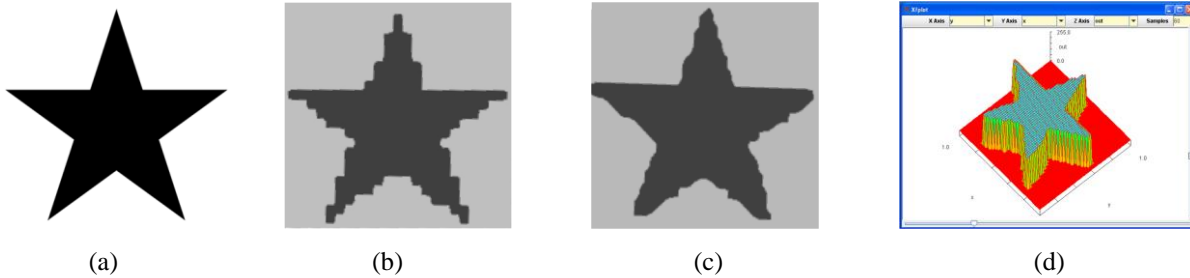


Figure 7: (a) Original image. Models with a (b) complete and (c) non complete rule base. (d) Window of *xfplot*.

membership per input and its type) and the system style (type of antecedent connective and defuzzification method) (Fig. 6b).

Evaluation and comparison of the models extracted can be done with the verification tool *xfplot*, which allows representing the output versus two or one of the inputs. A color palette to represent the output values facilitates visualization. The model results shown above in this paper has been obtained with this tool. As another example, Fig. 7b-c illustrate the result of using a complete rule base of  $25 \times 25$  (625) rules and a non complete rule base with 572 rules for modeling the star in Fig. 7a. They have been shown as 2D results because the model only contains 2 output values, but they can also be shown as 3D graphics, as illustrated in Fig. 7d.

Another verification tool very useful to analyze the linguistic models obtained is *xfmt*, which allows monitoring how the rules are activated by an (x,y) input and how is obtained the output value. It is particularly useful to understand how grid cells with no rule associated specifically are absorbed by neighbor rules. Fig. 8 illustrates the windows of this tool.

The simplification tool, *xfsp*, offers different algorithms to simplify the membership functions (defined for inputs and outputs) as well as the rules. Among the later, the user can select the tabular simplification described above. Since tabular simplification is performed per each output membership function, it is recommended to firstly simplify the

number of output functions. This can be done with *xfsp* by applying clustering or similarity-based algorithms. For example, Fig. 9 shows how the 76 grey levels of the pixels in a image are clustered into 8 grey levels. After tabular simplification, it is recommended to use again *xfsp* to purge those input membership functions that are not used by the rules.

Finally, the learning tool, *xfsl*, can be employed to tune the rule parameters so as to adjust the model to the numerical data.

#### IV. APPLICATION EXAMPLES

The proposed approach to modeling images have been

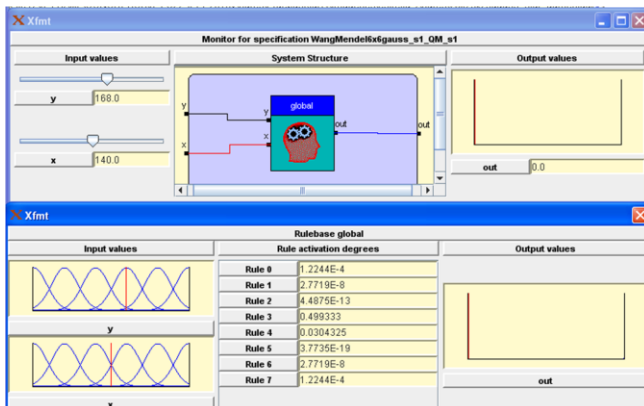


Figure 8: Windows of *xfmnt*.

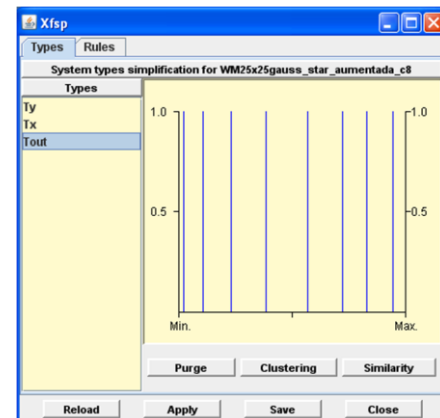
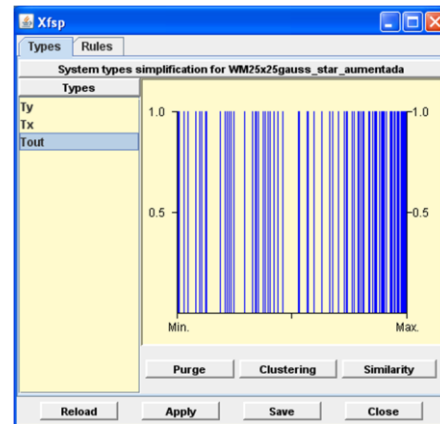


Figure 9: Windows of *xfsp*.

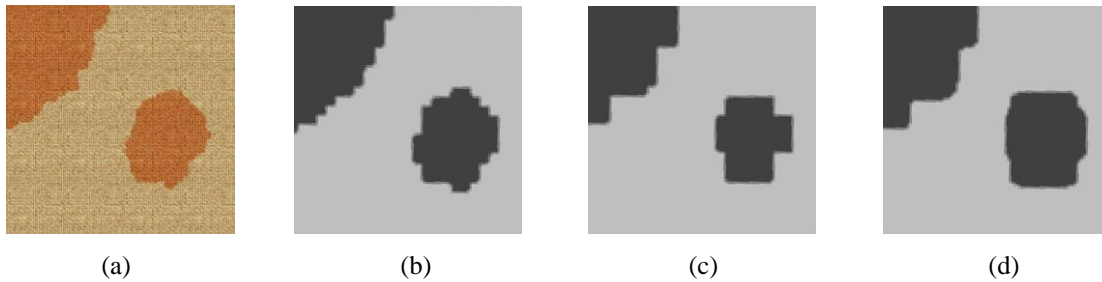


Figure 10: (a) Original image. (b) Model with 38 rules. (c) Model with 6 rules for the spots and an 'else' rule for the background. (d) Model with 12 rules.

applied to colour and texture images with different visual appearance. Fig. 10a shows an image corresponding to a cloth stained with wine. Using the tool *xfdm*, Wang-Mendel technique with a initial partition of 25x25 was selected to extract 625 rules. Using *xfsp*, the consequent values of the rules were clustered into 2 values that correspond to what visually is appreciated as stained cloth and non stained cloth. Tabular simplification applied with *xfsp* resulted in a system with 38 rules (15 modeling the spots and 23 modeling the background). Fig. 10b shows the output of this model obtained with *xplot*. Further simplification of the model is performed by selecting the most significant prime implicants. Fig. 10c illustrates the result obtained with *xplot* when the model contains 6 prime implicants (6 rules) to describe the spots and an else rule to describe the background (what is not spot is background). It can be seen how this model (which is crisp for using the 'else' rule) provides a result with a quadrilateral aspect. In the other side, Fig. 10d illustrates the result obtained with *xplot* when the model contains 6 rules to describe the stained cloth and other 6 rules to describe the non stained cloth. This description is not complete since it contains 12 instead of the original 38 rules. Simplification process leads to identify 5 relevant zones in *x* and *y* pixels positions that can be denoted as 'quite at the top', 'top', 'middle', 'bottom', 'quite at the bottom' (for the *y*), and 'quite at the left', 'left', 'middle', 'right', 'quite at the right'. The model has a clear linguistic meaning and is similar to what a human would say about the image. Examples of the rules are the following:

```

If y is quite at the top and x is quite at the left then the cloth is stained;
If y is quite at the bottom then the cloth is not stained;

```

Another example that illustrates the technique is shown in Fig. 11. The image in Fig. 11a represents a crack in the asphalt of a road. The pixel values have been clustered into 2 values representing the crack and the background. The obtained fuzzy model, represented in Fig. 10d contains 7 rules for describing the crack and 9 for describing the background. They have a high linguistic meaning as can be appreciated in the following examples:

```

If x is everywhere at the left then there is not crack;
If y is from the middle to the bottom and x is in the middle then there is crack;

```

As commented in the introduction, linguistic models can be employed in pattern recognition systems. The purpose of the last example included in this section is to illustrate the use of the models described herein in this kind of systems. A repository has been created with simple linguistic descriptions of each pattern so that given the image of an unknown pattern, the system finds which pattern in the repository gives the highest matching score with the unknown pattern. In order to evaluate matching, several approaches can be implemented. One of them is to model the unknown pattern and to compare it with those in the

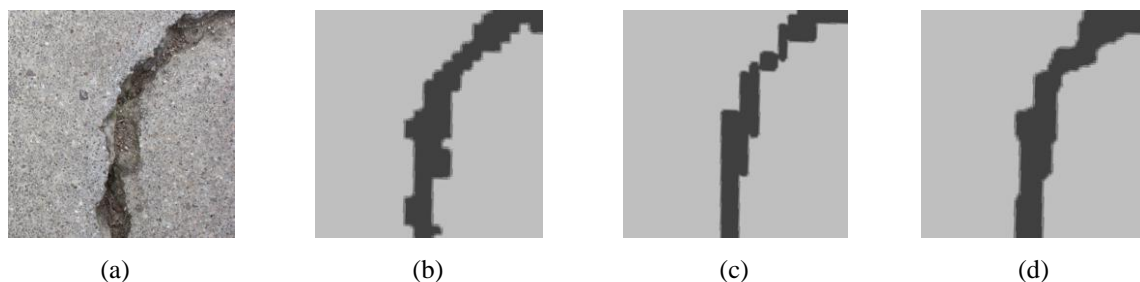


Figure 10: (a) Original image. (b) Model with 39 rules obtained from simplifying a model with 625 rules. (c) Model with 7 rules for the crack region and an 'else' rule for the background. (d) Model with 16 rules.

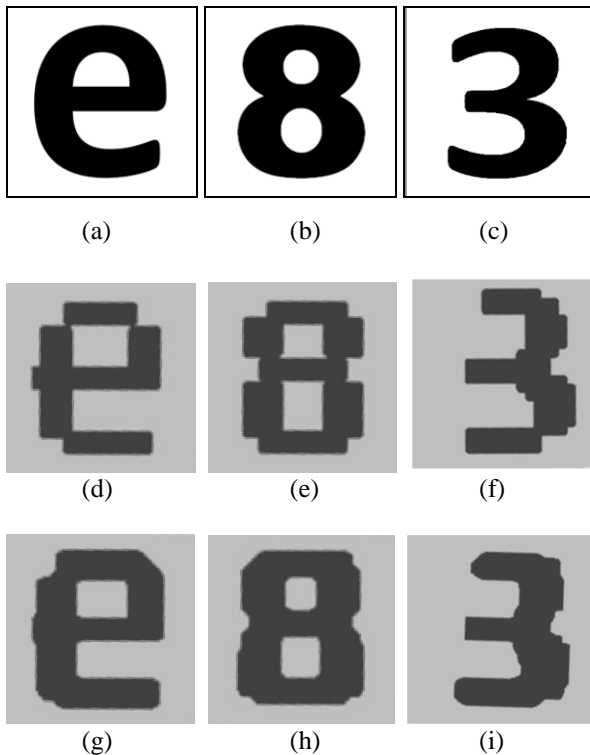


Figure 11: Similar patterns to recognize: (a) letter ‘e’, (b) number ‘8’, and (c) number ‘3’; (d)-(f) Relevant rules of the models for the dark regions and an ‘else’ rule for the background; (g)-(i) Relevant rules of the models for the dark and clear regions.

repository. This would require to sweep the pixels in the image (all or a significant part) to generate the numerical data required by the model. The approach that has been implemented is to sweep the pixels (all or a significant part) but directly to evaluate which model is the most adequate. Since the models are simple, few operations are required per pixel. The patterns considered have been letters and numbers. The matching score is calculated by summing, for each model in the repository, the activation of its constituent rules, and subtracting the activation of its contrary rules (again both sets of rules are not complement but fuzzy complement). The activation value of a rule for an input pixel is calculated as the product of the antecedents’ (position) and consequent’s (pixel value) membership

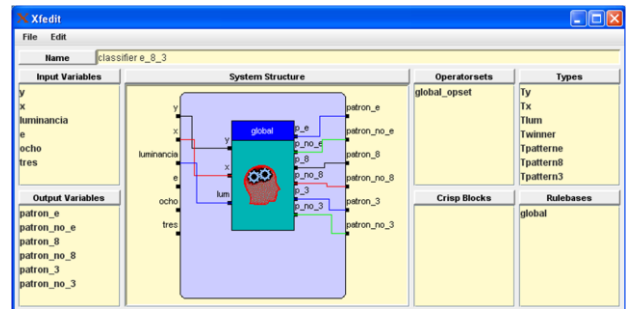


Figure 12: Window of *xfedit* with part of the recognition system.

degrees of that input pixel. Product is used because connective ‘and’ is represented by the product.

This recognition system has been described with the description tools of Xfuzzy 3 (*xfedit* and *xfpkg*) and simulated with the tool *xfsim* of Xfuzzy 3. To illustrate its performance, let us focus on three patterns that are similar, such as those illustrated in Fig. 11. Fig. 12 illustrates the *xfedit* window with the rule base associated with these three patterns.

Since the models stored summarizes the visual appearance of the patterns, the recognition system performs similarly to what a human would recognize. This does not happen in other kind of recognition systems, such as those based on neural networks. Another advantage, is that the matching scores of the different patterns can be provided so as to evaluate the fiability of the recognition.

As examples, let us consider the six different input patterns illustrated in Fig. 13. They are more or less noisy version patterns of the stored ones. Table I shows the matching scores provided by the recognition system when it is simulated with *xfsim* and the input patterns are those in Fig. 13. The matching score is expressed as percentage of a full similarity between the input pattern and the model stored. It can be seen how the score diminishes as the input pattern is more noisy. The system recognizes the pattern as that with the maximum matching score (depicted in bold in Table I). Similarly to what a human would decide, there are patterns, such as that in Fig. 13c, where the decision is not clear at all.

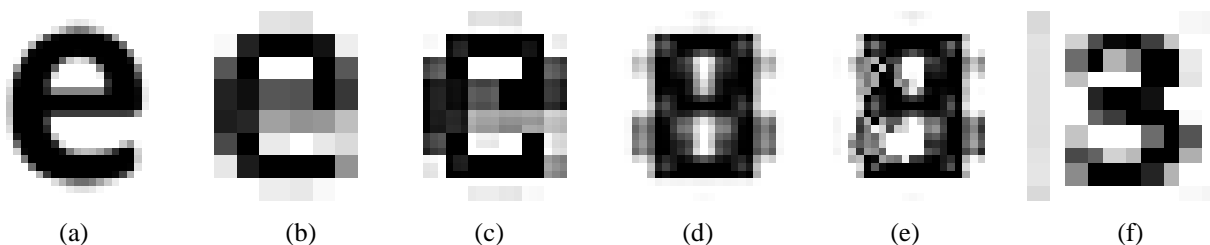


Figure 13: Different input patterns analyzed.

## V. CONCLUSION

The proposed methodology extracts models that describe the low-level features on an image similarly to what a human could provide. Apart from its linguistic interpretability, compression of the image information into a true fuzzy rule base provides better modeling than a crisp rule base. The methodology can be applied easily thanks to the CAD tools of Xfuzzy 3 environment, which allow extracting grid-based rules from numerical data, applying clustering to consequent values of the rules, simplifying the result by tabular methods, monitoring its behavior and simulating its performance within a particular application.

## REFERENCES

- [1] M. Nachtegael and E. E. Kerre. "Classical and fuzzy approaches towards mathematical morphology". Chapter 1 in *Fuzzy techniques in image processing*, Physica-Verlag, 2000.
- [2] M. Nachtegael, D. Van der Weken, and E. E. Kerre. "Soft Computing in ImageProcessing: Recent Advances" (Series Studies in Fuzziness and Soft Computing). Springer-Verlag, 2007.
- [3] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years". *IEEE Trans. on PAMI*, 22 (12), pp. 1349-1379, 2000.
- [4] Q. Li, Z. Shi, and Z. Shi, "Linguistic expression based image description framework and its application to image retrieval", Chapter 4 in *Soft Computing in Image Processing*, Vol. 210/2007, Springer-Verlag.
- [5] H.-H. Nagel, "Steps toward a cognitive vision systems", *AI Magazine*, vol. 25, no. 2, pp. 31-50, 2004.
- [6] Z. Chi, H. Yan, "Handwritten numeral recognition using self organizing maps and fuzzy rules. *Pattern Recognition* 28 (1), pp. 59-66, 1995.
- [7] B. Lazzerini, F. Marcelloni, "A linguistic fuzzy recogniser of off-line handwritten characters", *Pattern Recognition Letters*, 21, pp. 319-327, 2000.
- [8] J. Chamorro-Martínez, and P. Martínez-Jiménez, "Fuzzy Sets for Image Texture Modelling Based on Human Distinguishability of Coarseness", *Lectures Notes in Computer Science*, Vol. 5571/2009, pp. 229-236.
- [9] Xfuzzy 3 available at: <http://www.imse-cnm.csic.es/Xfuzzy>.
- [10] M. Sugeno, T. Yakusawa, "A Fuzzy-Logic-Based Approach to Qualitative Modeling", *IEEE Trans. On Fuzzy Systems*, Vol. 1, N° 1, pp.7-31, Feb. 1993.
- [11] F. Klawonn, R. Kruse, "Constructing a Fuzzy Controller from Data", *Fuzzy Sets and Systems*, 85, pp. 177-193, 1997.
- [12] M. Setnes, R. Babuska, U. Kaymak, H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification", *IEEE Trans. on Systems, Man and Cybernetics*, part B, vol. 28, no. 3, pp. 376-386, Junio 1998.
- [13] L. Wang, J. M. Mendel, "Generation Rules by Learning from Examples". *Proc. Int Symp. on Intelligent Control*, pp. 263-268, 1991.
- [14] D. Nauck, "Data Analysis with Neuro-Fuzzy Methods", PhD. Dissertation, Univ. of Magdeburg, Faculty of Computer Science, Alemania, 2000.
- [15] R. Senhadji, S. Sánchez-Solano, A. Barriga, I. Baturone, F. J. Moreno-Velo, "NORFREA: An Algorithm for non-redundant fuzzy rule extraction", *Proc. IEEE SMC'2002*, vol. 1, pp. 604-608, Tunisia, Oct. 2002.
- [16] I. Baturone, F. J. Moreno-Velo, A. A. Gersnoviez, "A CAD Approach to Simplify Fuzzy System Descriptions", *Proc. FUZZ-IEEE'2006*, pp. 2392-2399, Vancouver (Canada), July 2006.

TABLE I  
MATCHING SCORES FOR THE INPUT PATTERNS IN FIGURE 13

Input Pattern	Pattern 'e'	Pattern '8'	Pattern '3'
Fig. 13a	<b>77.1 %</b>	68.6 %	51.4 %
Fig. 13b	<b>65.4 %</b>	59.4 %	46.1 %
Fig. 13c	<b>62.3 %</b>	<b>62.3 %</b>	48.7 %
Fig. 13d	59.9 %	<b>71.8 %</b>	57.7 %
Fig. 13e	57.9 %	<b>69.4 %</b>	64.4 %
Fig. 13f	52.3 %	62 %	<b>66 %</b>