8 Fuzzy Inference Systems Type-1 and Type-2 for Digital Images Edge Detection

Edges detection in digital images is a problem that has been solved by means of the application of different techniques from digital signal processing, also the combination of some of these techniques with Fuzzy Inference System (FIS) has been experienced. In this chapter a new FIS Type-2 method is implemented for the detection of edges and the results of three different techniques for the same intention are compared.

8.1 Introduction

In the area of digital signal processing, methods have been proven that solve the problem of image recognition. Some of them include techniques like binarization, bidimensional filtrate, detection of edges and compression using banks of filters and trees, among others.

Specifically in methods for the detection of edges we can find comparative studies of methods like: Canny, Narwa, Iverson, Bergholm y Rothwell. Others methods can group in two categories: Gradient and Laplacian (Heath, 1996).

The gradient methods like Roberts, Prewitt and Sobel detect edges, looking for maximum and minimum in first derived from the image. The Laplacian methods like Marrs-Hildreth do it finding the zeros of second derived from the image (Mendoza and Melin, 2005).

This work is the beginning of an effort for the design of new pre-processing images techniques, using Fuzzy Inference Systems (FIS), that allows feature extraction and construction of input vectors for neural networks with aims of image recognition.

Artificial neural networks are one of the most used objective techniques in the automatic recognition of patterns, here some reasons:

- Theoretically any function can be determined.
- Except the input patterns, it is not necessary to provide additional information.
- They are possible to be applied to any type of patterns and to any data type.

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 95–107, 2008.
springerlink.com

The idea to apply artificial neuronal networks for images recognition, tries to obtain results without providing another data that the original images, of this form the process is more similar to the form in which the biological brain learns to recognize patterns, only knowing experiences of past.

Models with modular neural networks have been designed, that allow recognizing images divided in four or six parts. This is necessary due to the great amount of input data, since an image without processing is of 100x100 pixels, needs a vector 10000 elements, where each one corresponds to pixel with variations of gray tones between 0 and 255 (Mendoza and Melin, 2005).

This chapter shows an efficient Fuzzy Inference System for edges detection, in order to use the output image like input data for modular neural networks. In the proposed technique, it is necessary to apply Sobel operators to the original images, and then use a Fuzzy System to generate the vector of edges that would serve as input data to a neural network.

8.2 Sobel Operators

The Sobel operator applied on a digital image, in gray scale, calculates the gradient of the intensity of brightness of each pixel, giving the direction of the greater possible increase of black to white, in addition calculates the amount of change of that direction.

The Sobel operator performs a 2-D spatial gradient measurement on an image. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

The Sobel edges detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The Sobel masks are shown in equation (8.1) (Green, 2002):

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{8.1}$$

Where Sobely y Sobelx are the Sobel Operators throughout x-axis and y-axis.

If we define I as the source image, g_x and g_y are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as in equations (8.2) and (8.3).

$$g_x = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobel_{x,i,j} * I_{r+i-2,c+j-2} \tag{8.2}$$

$$g_y = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobel_{y,i,j} * I_{r+i-2,c+j-2} \tag{8.3}$$

Where *gx* and *gy* are the gradients along axis-x and axis-y, and * represents the convolution operator.

The gradient magnitude *g* is calculated with equation (8.4) (Fan et al., 2004).

$$g = \sqrt{g_x^2 + g_y^2} \tag{8.4}$$

8.3 Edge Detection by Gradient Magnitude

Although the idea presented in this chapter, is to verify the efficiency of a FIS for edges detection in digital images, from the approaches given by Sobel operator, is necessary to display first the obtained results using only the gradient magnitude.

It will be used as an example the first image of the subject number one of the ORL database (figure 8.1). The gray tone of each pixel of this image is a value of between 0 and 255.

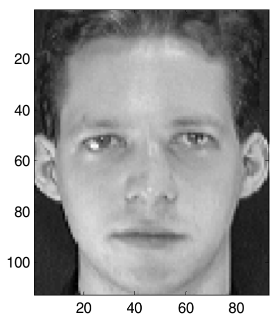


Fig. 8.1. Original Image 1.pgm

In figure 8.2 appears the image generated by *gx*, and figure 8.3 presents the image generated by *gy*.

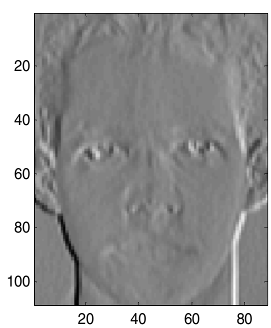


Fig. 8.2. Image given by *gx*

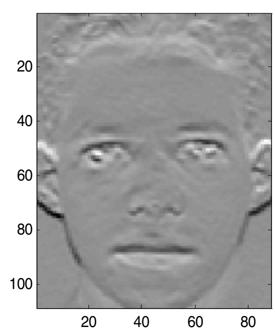


Fig. 8.3. Image given by *gy*

An example of maximum and minimum values of the matrix given by *gx*, *gy* and *g* from the image 1.pgm is shown in table 8.1.

Table 8.1. Maximum and minimum values from 1.pgm, gx, gy y g

Tone	1.pgm	gx	gy	g
Minimum	11	-725	-778	0
Maximum	234	738	494	792

After applying equation (8.4), g is obtained as it is in figure 8.4.

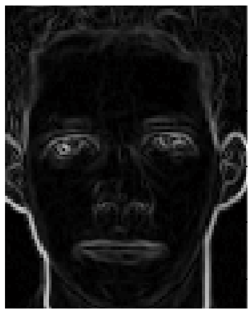


Fig. 8.4. Edges image given by g

8.4 Edge Detection Using Type-1 Fuzzy Logic

A Mamdani FIS was implemented using Type-1 Fuzzy Logic, with four inputs, one output and 7 rules, using the Matlab Fuzzy Logic Toolbox, which is shown in figure 8.5.

For the Type-1Fuzzy Inference System, 4 inputs are required, 2 of them are the gradients with respect to x-axis and y-axis, calculated with equation (2) and equation (3), to which we will call DH and DV respectively.

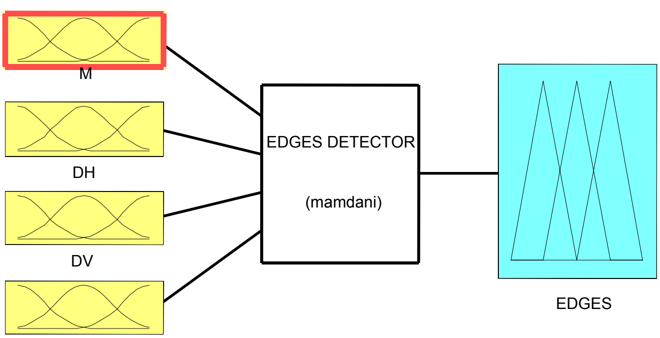


Fig. 8.5. FIS in Matlab Fuzzy Logic Tool Box

The other two inputs are filters: A high-pass filter, given by the mask of the equation (8.5), and a low-pass filter given by the mask of equation (8.6). The high-pass filter hHP detects the contrast of the image to guarantee the border detection in relative low contrast regions. The low-pass filter hMF allow to detects image pixels belonging to regions of the input were the mean gray level is lower. These regions are proportionally more affected by noise, supposed it is uniformly distributed over the whole image.

The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favor false edges by effect of noise (Miosso and Bauchspiess, 2001).

$$hHP = \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \tag{8.5}$$

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{8.6}$$

Then the inputs for FIS type 1 are:

DH=g_x

DV=g_y

HP= $hHP*I$

M= $hMF*I$

where * is the convolution operator.

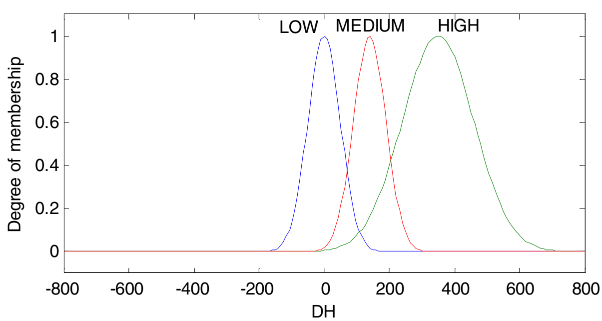


Fig. 8.6. Input variable DH

For all the fuzzy variables, the membership functions are of Gaussian type. According to the executed tests, the values in DH and DV, go from -800 to 800, then the ranks in x-axis adjusted as it is in figures 8.6, 8.7 and 8.8, in where the membership functions are:

LOW: gaussmf(43,0),
MEDIUM: gaussmf(43,127),
HIGH: gaussmf(43,255).

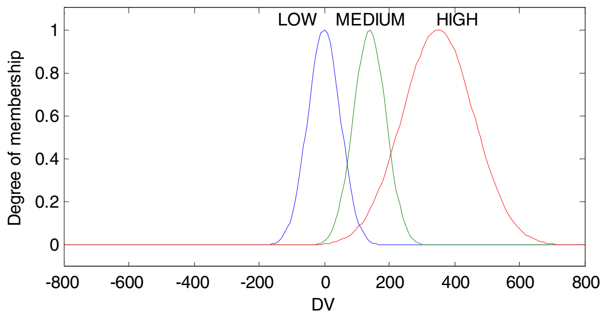


Fig. 8.7. Input variable DV

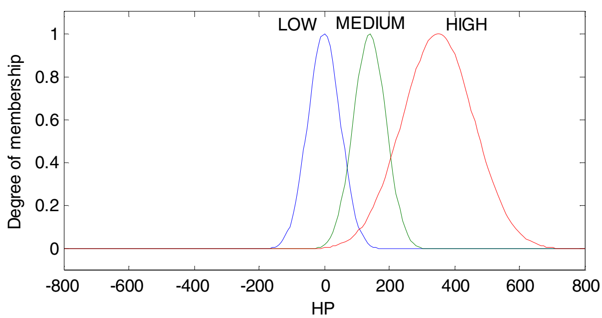


Fig. 8.8. Input variable HP

In the case of variable M, the tests threw values in the rank from 0 to 255, and thus the rank in x-axis adjusted, as it is appraised in figure 8.9.

In figure 8.10 is the output variable EDGES that also adjusted the ranks between 0 and 255, since it is the range of values required to display the edges of an image.
The seven fuzzy rules that allow to evaluate the input variables, so that the exit image displays the edges of the image in color near white (HIGH tone), whereas the background was in tones near black (tone LOW).

1. If (DH is LOW) and (DV is LOW) then (EDGES is LOW)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is HIGH)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is HIGH)

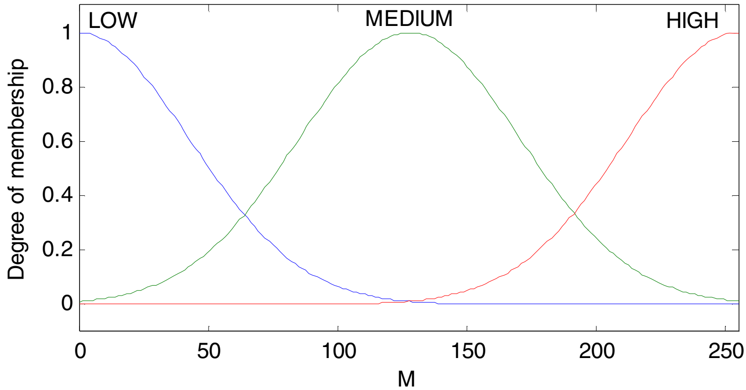


Fig. 8.9. Input variable M

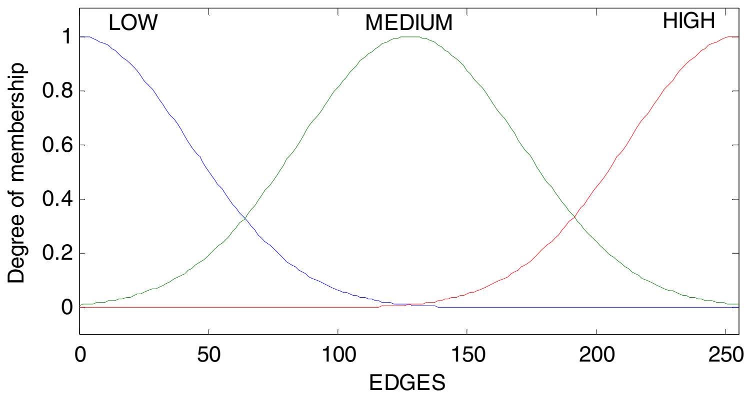


Fig. 8.10. Output variable EDGES

Fig. 8.11. EDGES Image by FIS Type 1

4. If (DH is MEDIUM) and (HP is LOW) then (EDGES is HIGH)
5. If (DV is MEDIUM) and (HP is LOW) then (EDGES is HIGH)
6. If (M is LOW) and (DV is MEDIUM) then (EDGES is LOW)
7. If (M is LOW) and (DH is MEDIUM) then (EDGES is LOW)

The result obtained for image of figure 1 is remarkably better than the one than it was obtained with the method of gradient magnitude, as it is in figure 8.11.

Reviewing the values of each pixel, we see that all fall in the rank from 0 to 255, which is not obtained with the method of gradient magnitude.

8.5 Edge Detection Using Type-2 Fuzzy Logic

For the Type-2 FIS, the same method was followed as in Type-1 FIS, indeed to be able to make a comparison of both results. The tests with the type-2 FIS, were executed using the computer program imagen_bordes_fis2.m, which creates a Type-2 Inference System (Mamdani) by intervals (Mendel, 2001).

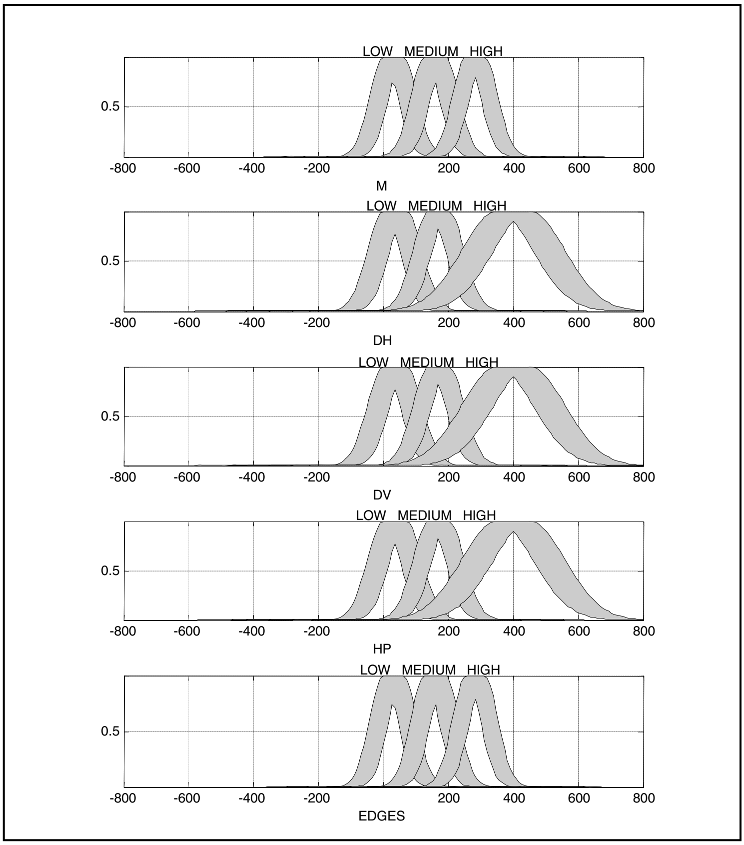


Fig. 8.12. Type-2 fuzzy variables

Fig. 8.13. EDGES Image by FIS Type 2

The mentioned program creates the fuzzy variables type 2 as it is seen in figure 8.12. The wide of the FOU chosen for each membership function was the one that had better results after several experiments.

The program imagen_bordes_fuzzy2.m was implemented to load the original image, and to apply the filters before mentioned. Because the great amount of data that the fuzzy rules must evaluate, the image was divided in four parts, and the FIS was applied to each one separately. The result of each evaluation gives a vector with tones of gray by each part of the image, in the end is the complete image with the edges (figure 8.13).

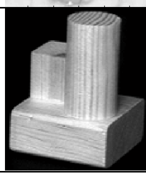
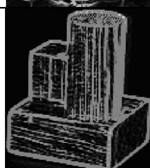
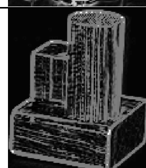
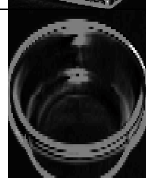
8.6 Comparison of Results

The first results of several tests conducted in different images can be appreciated in table 8.2.

At first, the results with FIS Type-1 and FIS Type2 are seen very similar. However thinking about that to show the images with a dark background it could confuse the contrast of tones, tests were done inverting the consequent of the rules, so that the edges take the dark tone and the bottom the clear tone, the rules changed to the following form:

1. If (DH is LOW) and (DV is LOW) then (EDGES is HIGH)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is LOW)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is LOW)
4. If (DH is MEDIUM) and (HP is LOW) then (EDGES is LOW)
5. If (DV is MEDIUM) and (HP is LOW) then (EDGES is LOW)
6. If (M is LOW) and (DV is MEDIUM) then (EDGES is HIGH)
7. If (M is LOW) and (DH is MEDIUM) then (EDGES is HIGH)

Table 8.2. Results of Edge Detection by FIS1 y FIS2 (dark background)

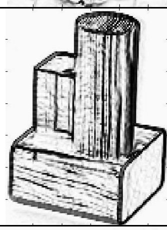
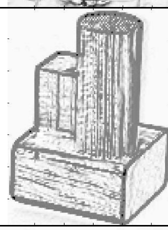
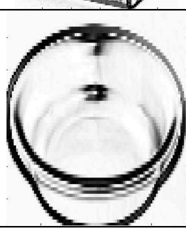
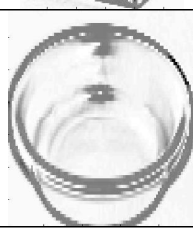
Original Image	EDGES (FIS 1)	EDGES (FIS 2)
		
		
		
		
		

Fuzzy Systems were tested both (Type-1 and Type-2), with the new fuzzy rules and same images, obtaining the results that are in table 8.3.

In this second test can be appreciated a great difference between the results obtained with the FIS 1 and FIS 2, noticing at first a greater contrast in the images obtained with the FIS 1 and giving to the impression of a smaller range of tones of gray in the type-2 FIS.

In order to obtain an objective comparison of the images, histograms were elaborated respectively [14] corresponding to the resulting matrices of edges of the FIS 1 and FIS 2, which are in table 8.4.

Table 8.3. Results of Edge Detection by FIS1 y FIS2 (clear background)

EDGES (FIS 1)	EDGES (FIS 2)
	
	
	
	

The histograms show in the y-axis the range of tones of gray corresponding to each image and in x-axis the frequency in which he appears pixel with each tone.

As we can observe, unlike detector FIS1, with FIS2 the edges of an image could be obtained from very complete form, only taking the tones around 150 and 255.

Like a last experiment, in this occasion to the resulting images of the FIS Type-2 the every pixel out of the range between 50 and 255 was eliminated.

Table 8.5 shows the amount of elements that was possible to eliminate in some of the images, we see that the Type-2 Edges Detector FIS allows to using less than half of the original pixels without losing the detail of the images. This feature could be a great advantage if these images are used like input data in neural networks for detection of images instead the original images.

Table 8.4. Histograms Of The Resulting Images Of The Edges by Gradient Magnitud, Fis 1 And Fis 2 Methods

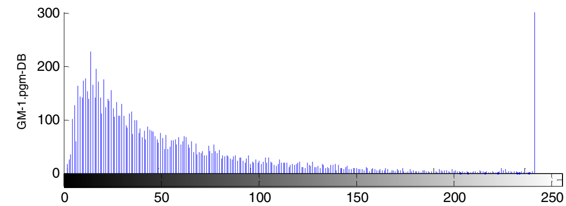
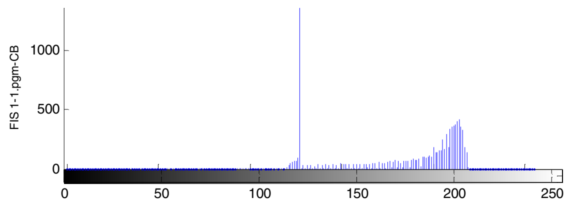
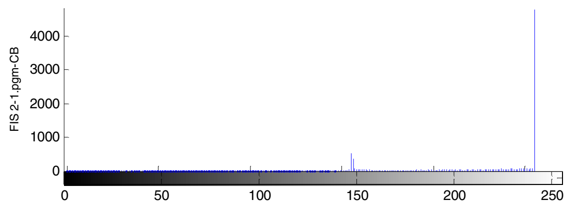
IMAGE: 1.PGM
METHOD: GRADIENT MAGNITUDE

METHOD: FIS 1 (CLEAR BACKGROUND)

METHOD: FIS 2 (CLEAR BACKGROUND)


Table 8.5. Type-2FIS Edges Images Including Only Pixels With Tones Between 150 And 255

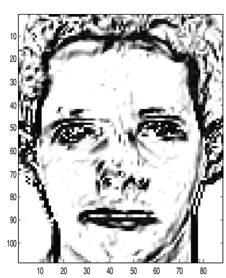
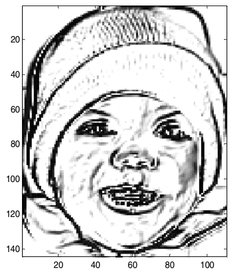
BORDERS IMAGE	DIMENSION (pixels)	PIXELS INCLUDED
	108x88 (9504)	4661 49 %

Table 8.5. (*continued*)

	144x110 (15840)	7077 44.6 %

8.7 Summary

The application of Sobel filters was very useful to define the input vectors for the Type-1 FIS and the Type-2 FIS, although in future works we will try to design Neuro-Fuzzy techniques able to extract image patterns without another data that the original image and to compare the results with traditional techniques of digital signal processing.

Thanks to the histograms of the images it was possible to verify the improvement of results of the FIS Type-1 with respect to the FIS Type-2, since with only the appreciation of the human eye was very difficult to see an objective difference.

The best result was obtained by the Type-2Fuzzy Inference System, because it was possible to clear more than half of the pixels without depreciating the image, which will reduce in drastic form the cost of training in a neural network.