Competitive Fuzzy-Classifier Image Edge Detection

Lily Rui Liang* and Carl G. Looney*
Computer Science Department
University of Nevada, Reno, NV 89557 USA
liang@cs.unr.edu, looney@cs.unr.edu

Abstract *The competitive fuzzy classifier operates on the set of four features extracted from the 3x3 neighborhood of each pixel. These features are the magnitudes of differences between that pixel and its neighboring pixels on four directions. They are input into the competitive fuzzy classifier inputs that connect to five fuzzy set membership functions that represent "white background" or one of the four different classes of "black edge". Each pixel in the image is classified on its highest fuzzy truth and mapped to white or black accordingly. Class oriented magnitude competition is done before a "black edges" pixel is mapped to black to output fine lines. The paradigm is simple, computationally efficient, has low sensitivity to noise and is isotropic. The competitive fuzzy classifier yields thin black lines on a white background.*

1. Introduction

In computer graphics, edges are defined as locations in an image where there is a significant variation in the gray level or color of pixel in some direction [1]. They are one of the most important visual clues for interpreting images [2]. Edge detection reduces an image to show only its edges details, which appear as the outline of objects within the image that can be used in subsequent image analysis operations for feature or object recognition [3].

Although there are many different ways to do edge detection, such as Sobel 3x3 filtering, Prewit 3x3 filtering, Laplacian of Gausian filtering, moment based operator, Shen & Castan operator, and Canny & Deriche operator, some common problems are a large volume of computation, sensitivity to noise and anisotropy. Russo, and Ramponi [4-6], designed fuzzy rules for the edge detection. Such rules can smooth while sharpening edges, but require a rather large rule set [7].

A fuzzy classifier system with two classes : edge and non-edge has been proved of easy modeling, small computation, low sensitivity to noise, and isotropy [8]. But same as the common thresholding method in the edge localization step of other edge detection algorithms, rather than detecting a single point of maximum gray level gradient associated with an edge , which is the summit of a ridge, it detects a portion of that ridge. The ridge can be rather broad in the case of diffuse edges, resulting in a thick band of pixels in the edge map [1].

* Supported by ARO Grant DAAD19-99-1-0089

The competitive fuzzy classifier thins the wide ridges around local maxima in difference magnitude down to edges that are only one pixel wide, and yields thin black lines on a white background.

2. Methodology

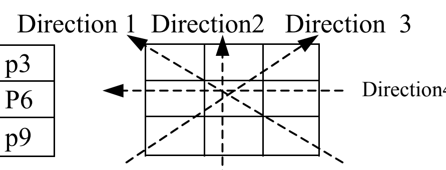
For a center pixel p5 in a 3x3 neighborhood, we define four directions on its neighborhood: horizontal, vertical, and two diagonals. The gray-level difference magnitudes between p5 and its neighbors on these directions are designated by X1, X2, X3, X4, and calculated by

$$X1 = |P1-P5|+|P9-P5|$$
$$X2 = |P2-P5|+|P8-P5|$$
$$X3 = |P3-P5|+|P7-P5|$$
$$X4 = |P4-P5|+|P6-P5|$$

Figure 1 shows the 3x3 neighborhood of gray-levels and the directions defined above.

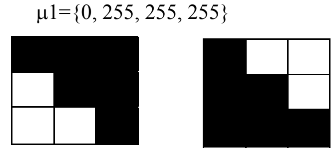
Figure 1. Pixels and Directions in a 3x3 Neighborhood.

p1	p2	p3
p4	p5	P6
p7	p8	p9

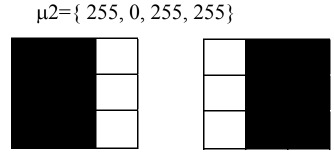


Four classes of edges and a non-edge class are differentiated in our model. Each class represents a pair of typical edges, which share a same feature vector μ. μ is 4-dimentional, and made up by X1, X2, X3 and X4, the directional gray level differences of those edges in that class. The edges represented by each class and their feature vectors are given below.

Class 1: $\mu1=\{0, 255, 255, 255\}$

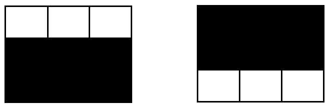


Class 2: $\mu2=\{255, 0, 255, 255\}$

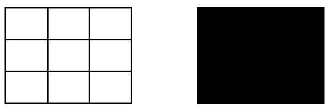


Class 3: $\mu 3=\{255, 255, 0, 255\}$

Class 4: $\mu 4=\{255, 255, 255, 0\}$



Class White Background: $\mu 0=\{0, 0, 0, 0\}$



Now, we use constant *Low* to substitute "0", and use constant *High* to substituted "255". With L denotes *Low* and H denotes *High*, the feature vectors become:

Class White Background: $\mu 0=\{L, L, L, L\}$
Edge Class 1: $\mu 1=\{L, H, H, H\}$
Edge Class 2: $\mu 2=\{H, L, H, H\}$
Edge Class 3: $\mu 3=\{H, H, L, H\}$
Edge Class 4: $\mu 4=\{H, H, H, L\}$

In practice, the values of *Low* and *high* can be defined by user for each particular image to achieve a desirable result.

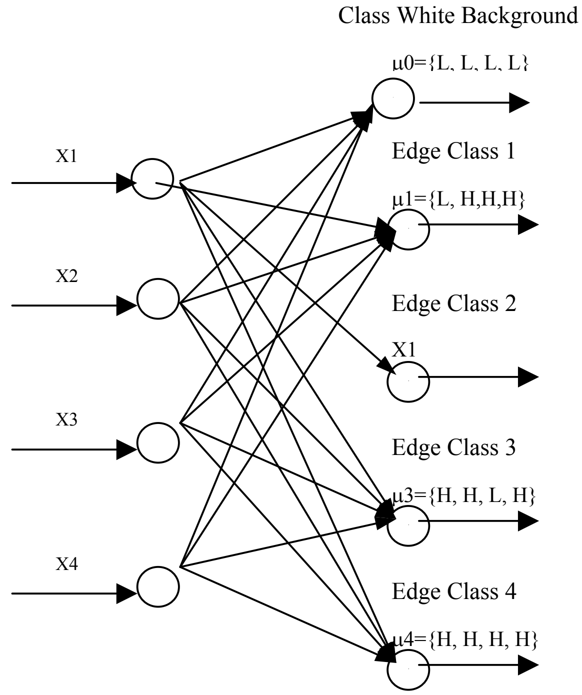
A fuzzy classifier is a system that accepts either: i) input feature vectors; or ii) fuzzy truths for features belonging to fuzzy set membership functions. It outputs a codeword that provides the class to which the feature vector belongs or else outputs a fuzzy value at the k-th output node to designate the fuzzy truth of the k-th class. Usually the individual output components are fuzzy truths. The criterion is that fuzzy truths are used in the decision making process[9].

For each pixel in the image there is an 4-dimensional feature vector {X1, X2, X3, X4} that contains the gray-level differences on the four directions of its 3x3 neighborhoods: horizontal, vertical, and two diagonals. The competitive fuzzy classifier operates on this feature vector to determine whether or not this pixel is an edge and what kind of edge it is by providing fuzzy truths for five classes of pixels shown as above. These classes are respectively mapped to black or white for the center pixel in the new output image. Thus every image pixel is mapped to black or white in the output image, which is a line drawing image of black lines on a white background.

3. System Model

Figure 2 shows the competitive fuzzy classifier.

Figure 2. Competitive Fuzzy Classifier Model



4. Fuzzy Set Membership Functions

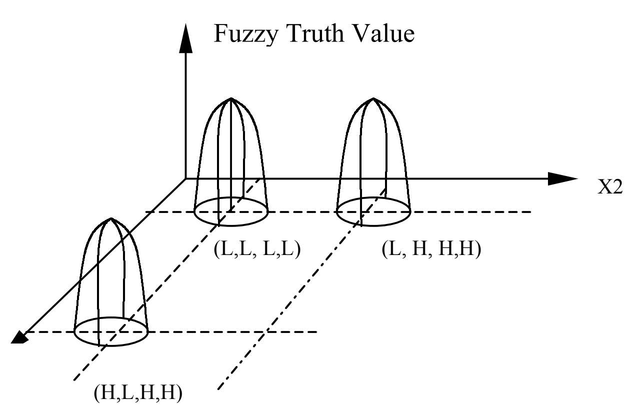
On the 4-dimensional feature space, we define the fuzzy membership functions for the five classes with Epanechnikov [9] functions as given below.

Class White Background: $\text{Max}\ \{0,\ 1-\|X-\mu 0\|^2/\beta\}$
Edge Class 1: $\text{Max}\ \{0,\ 1-\|X-\mu 1\|^2/\beta\}$
Edge Class 2: $\text{Max}\ \{0,\ 1-\|X-\mu 2\|^2/\beta\}$
Edge Class 3: $\text{Max}\ \{0,\ 1-\|X-\mu 3\|^2/\beta\}$
Edge Class 4: $\text{Max}\ \{0,\ 1-\|X-\mu 4\|^2/\beta\}$

$\mu 0, \mu 1 \dots \mu 4$ are defined as previous. Thus the quality of the edge detection depends upon parameters "L", "H", and "β", and thus upon the particular image.

Figure 3 is a view with 2 dimensions of feature plus the dimension of fuzzy truth. We present this for easy visualization.

Figure 3. a Three-Dimensional View of FMSF.



The upside-down cups are the Epanechnikov functions. Each input feature vector falls into one or the other fuzzy set membership functions because we enlarge them so that they overlap to cover all cases. The maximum fuzzy truth-value of the five fuzzy set membership functions is the winner, that is, determines the class of the pixel.

Before the pixel is changed to either white or black in the output image, a competition with its neighboring pixels is done. So that, for the pixels neighboring each other and belongs to the same class of edge, only the one with the largest difference magnitude on the direction associated with that class will be consider an edge, and turned to black. Rules for the competition are given below.

If Class White wins: change to white (1)

If Edge Class 1 wins: (2)
Compete X3 with pixels neighboring on Direction 3, if win, change to black, else change to white.

If Edge Class 2 wins: (3)
Compete X4 with pixels neighboring on Direction 4, if win, change to black, else change to white.

If Edge Class 3 wins: (4)
Compete X1 with pixels neighboring on Direction 1, if win, change to black, else change to white.

If Edge Class 4 wins: (5)
Compete X2 with pixels neighboring on direction 2, if win, change to black, else change to white.

5. Experimental Results

All of the results are obtained by using a 3x3 neighborhood of the center pixel, the fuzzy membership functions and the rules established above. The threshold parameters (L&H) are adjusted to achieve good results. In practice, since different people expect to see different details in the same image, those parameters should be input by users to order the type of edge they want [10].

Figures 4 and 7 show the respective image of a building and a space shuttle. We show the results of using the competitive fuzzy classifier on these images in Figures 5 and 8. For comparison, we show the results of using the popular Unix based tool Xview (xv) in Figures 6 and 9. We note that our lines are thin and making clear outline, more detail can be seen.

6. Analysis and Conclusions.

We have put the neighborhood difference magnitudes into a new type of competitive fuzzy classifier to classify a pixel as being an edge or not and we then changed the pixel to black or white in the output image. The results are a line drawing of thin lines on a white background.

The benefits of using a competitive fuzzy classifier model in edge detection are: i) yields thin black lines even the edge in the input image is diffuse ii) it is simple and easy to compute with only five fuzzy set membership functions iii) the Epanechnikov functions reduce the computation further without degrading the quality of the edge detection; iv) the method is not vulnerable to noise because of the fuzzy smoothing process where there can be noise on some of the feature values; v) the process is isotropic in that lines of all directions are detected equally well.

Future work should include using localized parameters of L and H or dynamic parameters to achieve more details.

References

[1] Nick Efford, Digital Image Processing, Addison Wesley, pp.164-173, 2000

[2] Earl Gose, Richard Johnsonbaug & Steve Jost, Pattern Recognition and Image Analysis, Prentice Hall PTR, pp. 298, 1996

[3] Gregory A. Baxes, Digital Image Processing, Principles and Applications, John Wiley & Sons, Inc.

[4] Russo. F. "A new class of fuzzy operators for image processing", IEEE Int.Conf. on Neural Networks, pp.815-820, 1993

[5] Russo.F. "A user-friendly research tool for image processing with fuzzy rules", Proc. First IEEE Int. Conf. Fuzzy Systems, San Diego, pp. 561-568, 1992

[6] Russo, F & Ramponi, G., "Fuzzy operator for sharpening of noisy images", IEE Electron. Lett., 28: pp.1715-1717, 1992

[7] Carl G. Looney, "Nonlinear Rule-based Convolution for Refocusing", Real-Time Imaging 6, pp.29-37, 2000

[8] Lily Rui Liang, Ernesto Basallo, and Carl G. Looney, "Image Edge Detection with Fuzzy Classifier", Proc. Of the ISCA 14th international Conference, Las Vegas, pp. 279, 2001

[9] www.cs.unr.edu/~looney/cs791j/unit4

[10] http://prettyview.com/edge/

Figure 4 Original Building Image

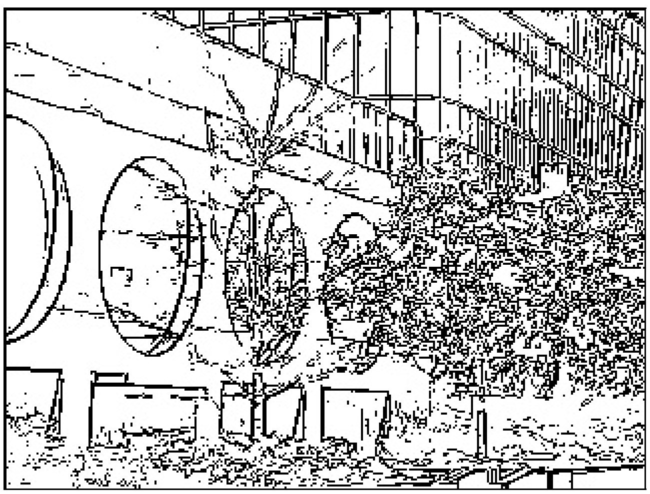


Figure 5 Edge-detected Building (L=0, H=35, β=1000000)

Figure 6 Edge-detected Building by Xview (threshold=35)

Figure 7 Original Shuttle Image

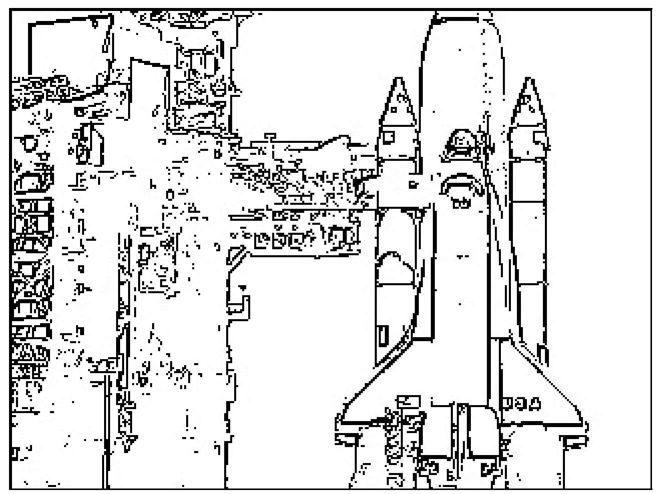


Figure 8 Edge-detected Shuttle (L=0, H=35, β=1000000)

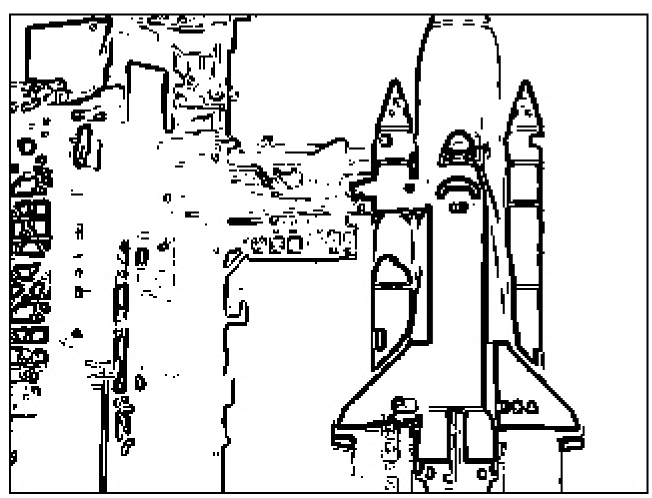


Figure 9 Edge-detected Shuttle by Xview (threshold=35)