Image Edge Detection with Fuzzy Classifier

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

Abstract. *Our special fuzzy classifier operates on the set of eight features extracted from the 3x3 neighborhood of each pixel. These features are the magnitudes of differences between that pixel and the eight neighboring pixels. They are input into the fuzzy classifier inputs that connect to two fuzzy set membership functions that represent "white background" or "black edge." The paradigm is simple, computationally efficient, has low sensitivity to noise and is isotropic. Each pixel in the image is mapped to white or black. The fuzzy classifier yields bold black lines on a white background.*

1. Introduction

1.1 Background.

Digital image processing performs digital signal processing in two dimensions. An M×N image is an array of locations of M rows and N columns of pixels (picture elements) where each has a grayscale value f(m,n)at each location (m, n).

Edges, in an image, are defined as locations where there is a significant variation in the gray level or color of pixel in some direction [1]. Edge detection is the most common approach for detecting meaningful discontinuities in gray level [2]. Edge detection algorithms locate and accentuate edges. A main purpose is to segment blobs for identifying objects in an image. A second purpose is to convert an image into a black and white line drawing. The latter is the purpose here. Edge delineation is useful in medicine, surveying, surveillance, etc.

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 [3-5], designed fuzzy rules for the edge detection. Such rules can smooth while sharpening edges, but require a rather large rule set. [6]

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

The benefits of employing a fuzzy classifier for edge detection are its small computation, low sensitivity to noise, isotropy and easy modeling, We discuss these in more detail after introducing the methodology.

Methodology.

For a 3x3 neighborhood of a center pixel p5, the gray-level difference magnitudes between p5 and its neighbors are designated by X1, X2,…,X8 and calculated by

X1=p1-p5	X2=p2-p5
X3=p3-p5	X4=p4-p5
X5=p6-p5	X6=p7-p5
X7=p8-p5	X8=p9-p5

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

Figure 1. Pixels and differences in a 3x3 neighborhood.

p1	p2	p3
p4	p5	P6
p7	p8	p9

X1	X2	X3
X4	0	X5
X6	X7	X8

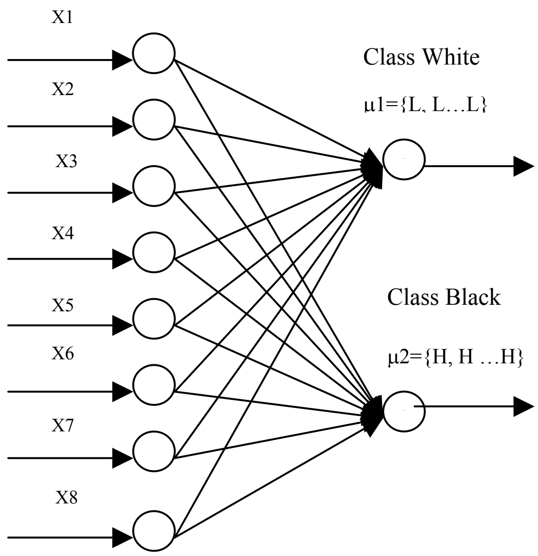
A fuzzy classifier is a system that accepts either: i) input feature vectors; or ii) fuzzy truths for features belonging to fuzzy set membership functions. It outputs a codeword that provides the class to which the feature vector belongs or else outputs a fuzzy value at the k-th output node to designate 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[7].

For each pixel in the image there is an 8-dimensional feature vector (X1, X2,…,X8) that contains the gray-level differences on its 3x3 neighborhoods. The fuzzy classifier operates on this feature vector to determine whether or not this pixel is an edge by providing fuzzy truths for two classes of pixels: i) an edge pixel, or ii) a non-edge pixel. These two 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.

The System Model.

Figure 2 shows the fuzzy classifier.

Figure 2. Fuzzy Classifier Model.



2. Fuzzy Set Membership Functions

The feature space is 8-dimensional. On it we define the fuzzy membership functions for Class White and Class Black that include constants and Epanechnikov [1] functions as given below.

Class White:

$$\begin{cases} 1 & \text{for } X1 \leq L \text{ and } X2 \leq L \text{ and} \ldots \text{and } X8 \leq L \\ \text{Max}\{0,\ 1-\|X-\mu 1\|^2/\beta\} & \text{otherwise} \end{cases}$$

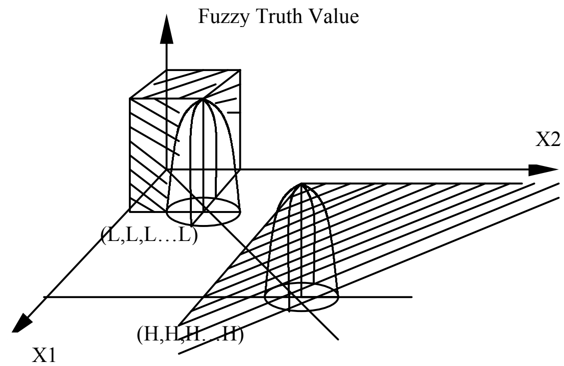
Class Black:

$$\begin{cases} 1 & \text{for } X1 \geq H, \text{ and } X2 \geq H, \text{ and} \ldots \text{and } X8 \geq H \\ \text{Max}\{0,\ 1-\|X-\mu 2\|^2/\beta\} & \text{otherwise} \end{cases}$$

L denotes the threshold variable *Low* and H denotes *High*. These are constants and their values depend on the particular image. The quality of the edge detection depends upon these parameters and thus upon the particular image.

A view with 2-dimentional feature vectors plus the dimension of fuzzy truth is shown in Figure 3. We present this for easy visualization.

Figure 3. A Three-Dimensional View of the FMFs.



The upside-down cups are the Epanechnikov functions that are combined with the constant 1 to form the complete fuzzy set membership function defined above. 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 two fuzzy set membership functions is the winner, that is, determines the class of the pixel. Then the pixel is changed to either white or black in the output image. The edge detection rules are given below.

If Class White wins: change to white (1)
If Class Black wins: change to black (2)
If there is a tie: change to white (3)

3. 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 and 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[8].

Figures 4 and 7 show the respective image of a building and a space shuttle. We show the results of using the 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 bold and stronger and that the background is clean. We could perform a line thinning technique now if we wanted strong thin lines.

4. Analysis and Conclusions.

We have put the neighborhood difference magnitudes into a new type of 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 thick bold lines on a white background.

The benefits of using a fuzzy classifier model in edge detection are: i) it is simple and easy to compute with only two fuzzy set membership functions, which greatly reduces the fuzzy truth value computation; ii) the Epanechnikov functions reduce the computation further without degrading the quality of the edge detection; iii) the method is not vulnerable to noise because of the fuzzy smoothing process where there can be noise on some of the feature values; iv) the process is isotropic in that lines of all directions are detected equally well.

Future work should include refinements to reduce the width of the lines. It may be worthwhile to also allow more than two shades of gray in the final drawing, which could be, for example, two or three shades of black on a white background, depending on the edge strength.

References

[1] Carl G. Looney, “Fuzzy and rule-based image convolution”, Mathematics and Computers in Simulation 51, pp.209-219, 2000

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

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

[4] 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

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

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

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

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

Figure 4 Original Building Image

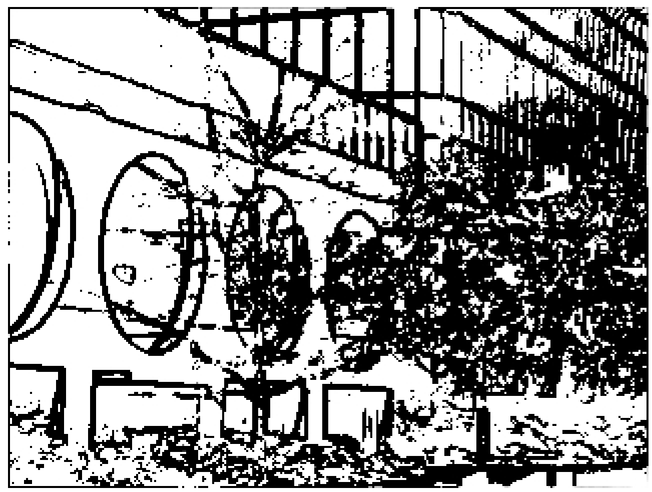


Figure 5 Edge-detected Building (L=6, H=15, β=80000)

Figure 6 Edge-detected Building by XView

Figure 7 Original Shuttle Image

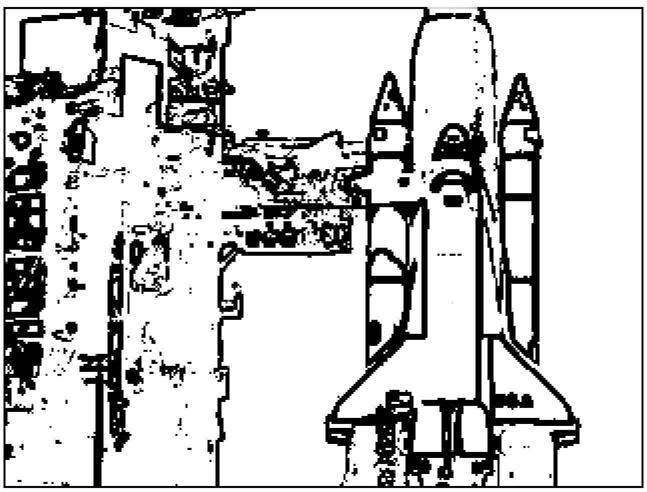


Figure 8 Edge-detected Shuttle
(L=6, H=15, β=80000)

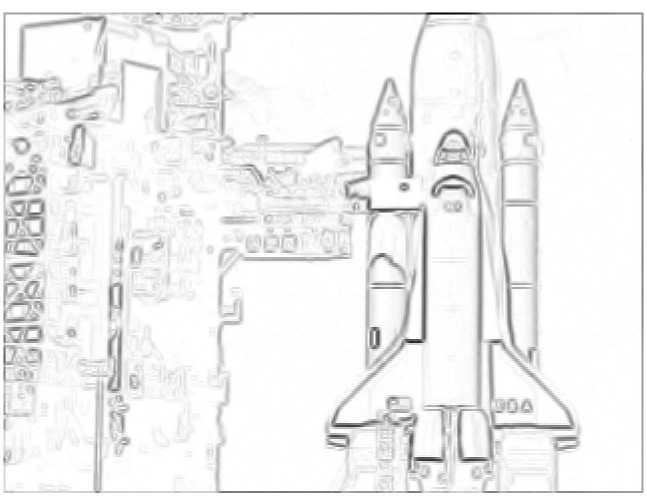


Figure 9 Edge-detected Shuttle by XView