Fuzzy Inference System For Edge Detection In True Colour Images

Mrs. K. Mekala, II[nd] ME., (Communication Systems),
Mrs. S. Saraswathi, ME, Assistant Professor / ECE,
Mahendra Engineering College (Autonomous),
Mahendirapuri, Mallasamudram, Namakkal, Tamilnadu, India

Abstract

This paper reports the implementation in MATLAB environment, of a very simple but efficient Fuzzy logic(FL) based algorithm to detect the edges of an input image by scanning it throughout using a 2*2 pixel window using Fuzzy Inference System(FIS). An edge is a boundary between two uniform regions. We can detect an edge by comparing the intensity of neighboring pixels. As uniform regions are not crisply defined, small intensity differences between two neighboring pixels do not always represent an edge. Instead, the intensity difference might represent a shading effect. Also, a Graphical User Interface (GUI) in MATLAB has been designed to aid the loading of the image, and to display the resultant image at different intermediate levels of processing. Threshold level for the image can be set from the slider control of GUI. It is simulated using MATLAB 7.6 R2008a tool. The fuzzy rule based algorithm has been successful in obtaining the edges that are present in an image after its implementation and execution with various sets of images. Thus developed algorithm exhibits tremendous scope of application in various areas of digital image processing.

1. Introduction

Modern time is an era of technology in which we now believe in the vision based intelligence. Penetration of computers into each area of the market and living has forced the designers to add the capability to see and analyze and to innovate more and more into

the area of electronic vision or image processing. At the level of computational intelligence for electronic vision, many of the algorithms have been developed to extract different types of features from the image such as edges, segments and lot many other types of image features. Edge detection is a terminology in electronic vision, particularly in the areas of feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities. The goal of edge detection is to locate the pixels in the image that correspond to the edges of the objects seen in the image. This is usually done with a first and/or second derivative measurement following by a comparison with threshold which marks the pixel as either belonging to an edge or not. The result is a binary image which contains only the detected edge pixels. The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. Discontinuities in image brightness are likely to correspond to discontinuities in depth, discontinuities in surface orientation, changes in material properties or variations in scene illumination. Shashank Mathur and Anil Ahlawat, presented a fuzzy relative pixel value algorithm for edge detection by checking the relative pixel values in 3*3 pixels mask for scanning of image using the windowing technique, which is subjected to a set of fuzzy conditions for the comparison of pixel values with adjacent pixels to check the pixel magnitude gradient in the window. Their technique was not rule based Yinghua Li, Bingqi Liu and Bin Zhou presented Fuzzy technology as a newly rising technology used in many fields, especially in the image domain, and fuzzy enhancing technique as one important portion of the fuzzy technology. Based on this technology, they firstly set the image fuzzy characteristic plane of original image, secondly preceded the fuzzy enhancement, and then detected the edge by Sobel differential arithmetic. Yasar Becerikli1 and Tayfun, Turkey proposed that edge detection is one of the most important tasks in image processing. They studied that image segmentation; registration and identification are based on edge detection. They proposed that fuzzy rules based algorithm is more flexible in handling thickness of edges in the final image. Cristiano Jacques Miosso, Adolfo Bauchspiess presented that First-order linear filters constitute the algorithms most widely applied to edge detection in digital images but they don't allow good results to be obtained from images where the contrast varies a lot, due to non-uniform lighting, as it happens during acquisition of most part of natural images. Dong-Su

Kim, Wang-Heon Lee presented an edge magnitude and direction scheme that uses 3 x3 ideal binary pixel patterns and described a lookup table. They concluded that their algorithm didn't require any manual online threshold adjustment and was more suitable to the dynamic environment.

2. Related Work

To solve the problem of existing technique, a color edge detection technique which isbased on the fusion of principal component analysis and hue component are proposed by Lie and Fan (2013).Initially a computational method of hue difference is defined, to obtain accurate edges for hue component which is then applied to classical gradient operators [10]. Moreover by using edge fusion of huecomponent and principal component of color image with lesser computational complexity, complete object edges can be obtained [3]. Wang and Yan(2012) [4] proposed a new edge detection in color image processing which is based on vector morphological operators. A new vector ordering has proposed in RGBcolor space. And then by determining the characteristics of noise contaminating image, we can proposed a vector morphological operators and applied in color edge detection. The new ideal is noise removing and pixels judgement that can be used to represent color morphological dilation and erosion. How edges are efficiently preserved in noise appearance for this, a novel approach of edge detection for color image was proposed by Xu et al.(2012)[5] . Morphological gradient operator is constructed with performance of noise latter which is consistent with human vision perception, for this a multi-structure elements are designed. An improved canny algorithm has proposed byXin and Ki (2012)[6] to detect edges in color images. Algorithm is consistingof vectorsobel gradient computation, non-maxima suppression based on edge detection,quaternion weighted average filter, interpolation and connection. An algorithm is applied to deal with color image of transmission line. This results show that from grey image processing method our algorithm is still better., XIAO et al. (2016)[7] has proposed a multi-scale edge detection algorithm which took soft threshold method to implement detail enhancement and reduction in noise of the true color image. A novel color edge detection algorithm is presented by Chen et al.(2010) [8] for the improvement of efficiency and the performance of color edge detection. To smooth the original image,an improved kuwahara filter is used. In RGB color spacing, after edge detection in each channel independently, an adaptive threshold selection method is used to determine the ideal threshold value and this algorithm is known as edge thinning and it is applied to get accurate edge. Based on the concept of self-organizing map (som), Jordan et al.(2011)[9] has presented multispectral images of edge detection. To produce a global ordering of spectral vectors. A one to one correspondence between pixels values and scalars is guaranteed, with global ordering`. The edge probability is only finding by the adjacent human pixels. This method eliminates linearization and uses the SOM more effectively for edge detection while holding greater flexibility. Singh et al.(2013)[1] has presented architecture to reduce the usage of FPGA resources and it use only one processing element for computing gradient of all three R, G and B color component. The FPGA resource usage is decreased by 35% from standard implementations that uses three gradient computational blocks. B. Gao et al.(2012)[2] has proposed method of edge detection which is built on 32 fuzzy rules. For MRI head scans edge detection is the pre-segmentation processes. It observes more accurate edges than the traditional canny and sobel edge detection operator and time taken for detection of edges is also less. It generates sharp and clear edges for segmentation of brain portions in MRI of head scans of human.

3. Canny Edge Detection Method

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection includes:

1. Detection of edge with low error rate, which means that the detection should accurately

catch as many edges shown in the image as possible

2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations – a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

Among the edge detection methods developed so far, Canny edge detection algorithm is one of the most strictly defined methods that provides good and reliable detection. Owing to its optimality to meet with the three criteria for edge detection and the simplicity of process for implementation, it became one of the most popular algorithms for edge detection.

4. Problem Statement

The following problems are identified from the current edge detection methods.

- Very Sensitive to noises from the surrounding area
- It uses 3x3 pixel window for Masking
- Difficult to implement to reach real time response.
- More time consuming
- Inaccurate and Sensitive to Noise

5. Fuzzy based Edge Detection Scheme

Fuzzy logic represents a good mathematical framework to deal with uncertainty of information. Fuzzy inference system designed has four inputs, which corresponds to four pixels of instantaneous scanning matrix, one output that tells whether the pixel under consideration is "black", "white" or "edge" pixel. Rule base comprises of sixteen rules, which classify the target pixel. Algorithm for the noise removal has been implemented at different levels of processing. The resultant image from FIS is subjected to first and second derivative to trace the edges of the image and for their further refinement. Main feature of the algorithm is that it has been designed by the smallest possible mask i.e. 2*2 unlike 3*3 or bigger masks.

5.1 Edge Detection

Edge detection is a terminology in electronic vision, particularly in the areas of feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities. The goal of edge detection is to locate the pixels in the image that correspond to the edges of the objects seen in the image.

5.2 Noise Removal

Noise removal is performed at different intermediate levels of processing. The idea of noise removal is to remove the pixels which have been falsely recognized as edge by the processing. Size of the scanning mask for this task is 2*2 pixels window. 2*2 pixels mask is slid over the whole image pixel by pixel row wise and the process continues till the time whole image is scanned for unwanted edge pixels. Fig. 5.1 shows p5 as falsely marked edge pixel as all the surrounding pixels i.e. p1, p2, p3, p4, p6, p7, p8 & p9 are white. Such types of falsely marked edge pixels are changed to White by the noise removal algorithm.

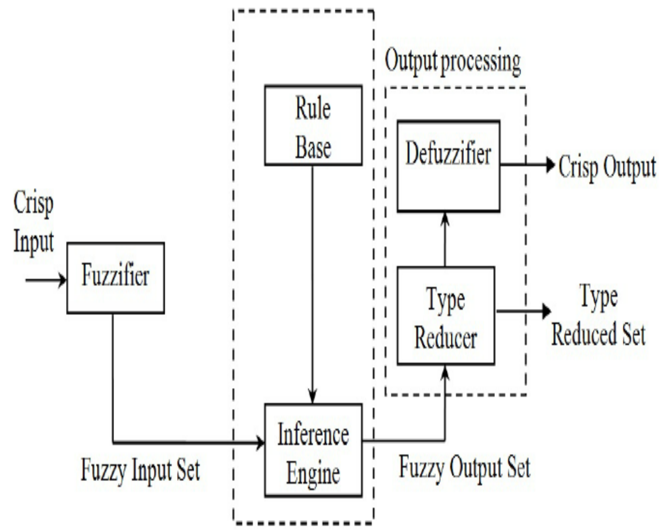

Fig :5.1 Fuzzy Block Diagram

5.3 Fuzzy Algorithm

- * Import RGB Image and Convert to Grayscale
- * Convert Image to Double-Precision Data
- * Obtain Image Gradient
- * Define Fuzzy Inference System (FIS) for Edge Detection
- * Specify FIS Rules
- * Evaluate FIS
- * Plot Results

In this section, we give a brief review on the main technique of our algorithm.

* **Import RGB Image and Convert to Grayscale**

Import the image into MATLAB.

Irgb = imread('peppers.png');

Irgb is a 384 x 512 x 3 uint8 array. The three channels of Irgb (third array dimension) represent the red, green, and blue intensities of the image.

Convert Irgb to grayscale so that you can work with a 2-D array instead of a 3-D array. Use the standard NTSC conversion formula to calculate the effective luminance of each pixel.

Igray = 0.2989*Irgb(:,:,1)+0.5870*Irgb(:,:,2)+0.1140*Irgb(:,:,3);

figure; image(Igray,'CDataMapping','scaled'); colormap('gray');

title('Input Image in Grayscale')

Alternatively, you can use the rgb2gray function in the Image Processing Toolbox software to convert Irgb to grayscale.

* **Convert Image to Double-Precision Data**

The Fuzzy Logic Toolbox software operates on double-precision numbers only. So, convert Igray, a uint8 array, to a double array.

I = double(Igray);

Because uint8 values are in the [0 2^8-1] range, all elements of I are in that range too. Scale I so that its elements are in the [0 1] range.

classType = class(Igray);

scalingFactor = double(intmax(classType));

I = I/scalingFactor;

Alternatively, you can use the im2double function in the Image Processing Toolbox software to convert Igray to a scaled, double-precision image.

* **Obtain Image Gradient**

The fuzzy logic edge-detection algorithm for this example relies on the image gradient to locate breaks in uniform regions. Calculate the image gradient along the *x*-axis and *y*-axis.

Gx = [-1 1];

Gy = Gx';

Ix = conv2(I,Gx,'same');

Iy = conv2(I,Gy,'same');

figure; image(Ix,'CDataMapping','scaled'); colormap('gray'); title('Ix');

figure; image(Iy,'CDataMapping','scaled'); colormap('gray'); title('Iy');

Gx and Gy are simple gradient filters. You convolve I with Gx, using the conv2 function, to obtain a matrix containing the *x*-axis gradients of I. The gradient values are in the [-1 1] range. Similarly, you convolve I with Gy to obtain the *y*-axis gradients of I. You can use other filters to obtain the image gradients, such as the Sobel operator or the Prewitt operator. For information about how you can filter an image using convolution.

Alternatively, if you have the Image Processing Toolbox software, you can use the imfilter, imgradientxy, or imgradient functions to obtain the image gradients.

* **Define Fuzzy Inference System (FIS) for Edge Detection**

Create a Fuzzy Inference System (FIS) for edge detection, edgeFIS.

edgeFIS = newfis('edgeDetection');

Specify the image gradients, Ix and Iy, as the inputs of edgeFIS.

edgeFIS = addvar(edgeFIS,'input','Ix',[-1 1]);

edgeFIS = addvar(edgeFIS,'input','Iy',[-1 1]);

Specify a zero-mean Gaussian membership function for each input. If the gradient value for a pixel is 0, then it belongs to the zero membership function with a degree of 1.

sx = 0.1; sy = 0.1;

edgeFIS = addmf(edgeFIS,'input',1,'zero','gaussmf',[sx 0]);

edgeFIS = addmf(edgeFIS,'input',2,'zero','gaussmf',[sy 0]);

sx and sy specify the standard deviation for the zero membership function for the Ix and Iy inputs. You can change the values of sx and sy to adjust the edge detector performance. Increasing the values makes the algorithm less sensitive to the edges in the image and decreases the intensity of the detected edges.

Specify the intensity of the edge-detected image as an output of edgeFIS.

edgeFIS = addvar(edgeFIS,'output','Iout',[0 1]);

Specify the triangular membership functions, white and black, for Iout.

wa = 0.1; wb = 1; wc = 1;

ba = 0; bb = 0; bc = .7;

edgeFIS = addmf(edgeFIS,'output',1,'white','trimf',[wa wb wc]);

edgeFIS = addmf(edgeFIS,'output',1,'black','trimf',[ba bb bc]);

As you can with sx and sy, you can change the values of wa, wb, wc, ba, bb, and bc to adjust the edge detector performance. The triplets specify the start, peak, and end of the triangles of the membership functions. These parameters influence the intensity of the detected edges.

Plot the membership functions of the inputs/outputs of edgeFIS.

```
figure
subplot(2,2,1); plotmf(edgeFIS,'input',1); title('Ix');
subplot(2,2,2); plotmf(edgeFIS,'input',2); title('Iy');
subplot(2,2,[3 4]); plotmf(edgeFIS,'output',1); title('Iout')
```

* **Specify FIS Rules**

Add rules to make a pixel white if it belongs to a uniform region. Otherwise, make the pixel black.

```
r1 = 'If Ix is zero and Iy is zero then Iout is white';
r2 = 'If Ix is not zero or Iy is not zero then Iout is black';
r = char(r1,r2);
edgeFIS = parsrule(edgeFIS,r);
showrule(edgeFIS)
ans =
1. If (Ix is zero) and (Iy is zero) then (Iout is white) (1)
2. If (Ix is not zero) or (Iy is not zero) then (Iout is black) (1)
```

* **Evaluate FIS**

Evaluate the output of the edge detector for each row of pixels in I using corresponding rows of Ix and Iy as inputs.

```
Ieval = zeros(size(I));% Preallocate the output matrix
for ii = 1:size(I,1)
   Ieval(ii,:) = evalfis([(Ix(ii,:));(Iy(ii,:));]',edgeFIS);
End
```

* **Plot Results**

```
figure; image(I,'CDataMapping','scaled'); colormap('gray');
title('Original Grayscale Image')
figure; image(Ieval,'CDataMapping','scaled'); colormap('gray');
title('Edge Detection Using Fuzzy Logic')
```

we detected the edges in an image using a FIS, comparing the gradient of every pixel in the *x* and *y* directions. If the gradient for a pixel is not zero, then the pixel belongs to an edge (black). You defined the gradient as zero using Gaussian membership functions for your FIS inputs

Fuzzy rule based algorithm provides better edge detection and has an exhaustive set of fuzzy conditions which helps to extract the edges with a very high efficiency.. Threshold level setting is done through the slider control of GUI. More the value of the slider, more of the edges will be traced have better visual appearance than the standard existing. The fuzzy logic approach for image processing allows you to use membership functions to define the degree to which a pixel belongs to an edge or a uniform region.

6. Conclusion and Future Work

A very simple & small but a very efficient fuzzy rule based edge detection algorithm to abridge the concepts of artificial intelligence and digital image processing has been developed. The algorithm and associated GUI has been developed in MATLAB environment. Displayed results have shown the accuracy of the edge detection using the fuzzy rule based algorithm over the other algorithms. The fuzzy rule based algorithm has been successful in obtaining the edges that are present in an image after its implementation and execution with various sets of images. Sample outputs have been shown to make the readers understand the accuracy of the algorithm. Thus developed algorithm exhibits tremendous scope of application in various areas of digital image processing. In future it can be further enhanced by modifying some essential features for application in Medical field.

References

[1]G. Mercier, Dezert,Z.Liu,Jean. and Singh et al., "Edge detection in color images based on DSmT". Information Fusion (FUSION), IEEE, 2011.

[2] C. B,Gao. "Edge detection of colour image based on quaternion fractional differential". Image Processing, IET, pp.261-272, 2012.

[3]Lei , Fan.“ Modifiedcolor based edge detection of remote sensing images using fuzzy logic”,international journal of emerging trends & technology in computer science, vol 3, Issue 4, 2014.

[4] Lei, Y.Fan,Tao and Yi Wang. "Color edge detection based on the fusion of hue component and principal component analysis." IET Image Processing 8.1 ,pp.44-55,2014.

[5] Silvia, et al "Clifford Algebra Based Edge Detector for Color Images". Complex, Intelligent and Software Intensive Systems (CISIS), IEEE, 2012.

[6] L. Yan and Wang, Lei, "Edge detection of color image using vector morphological operators". Computer Science and Network Technology (ICCSNT), IEEE, 2012.

[7] Xu,H. Zhao, Hui and Y. Zhang, "Edge detection of color image using mathematical morphology in HSV color space". Computer Science and Network Technology (ICCSNT), IEEE, 2012.

[8] Xin,Geng,C.Keand H.Xiaoguang."An improved Canny edge detection algorithm for color image". Industrial Informatics (INDIN), IEEE, 2012.
[9] Xiao,G.Geng,M.Zhou, Feng. "Detail enhancement and noise reduction with true color image edge detection based on wavelet multi- scale". Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), IEEE, 2011.
[10]Ramandeep Kaur, Manjit Singh and Butta Singh, "Comparative Analysis of Color Edge Detection Techniques Based on Fuzzy Logic", An International Journal of Engineering Sciences, January Vol. 17, 2016.