DOI: 10.54254/2755-2721/4/2023422

A comparison of canny edge detection algorithm and edge detection algorithm based on fuzzy logic

Wang Yujie

North University of China

915649685@qq.com

Abstract. Edge detection techniques in digital image processing are a very valuable area of research. In recent years, there are two more popular edge detection algorithms, one is Canny edge detection algorithm,and the other is edge detection algorithm based on Fuzzy Logic. Both algorithms have been used in various fields. The main work of this paper is to do a brief introduction of two algorithms and compare the results from them.

Keywords: Canny operator, Fuzzy logic, Edge detection, comparison.

1. Introduction

With the development of all kinds of computer application, the importance of image edge detection is also increasing. Edge detection is the process of identifying and locating sharp discontinuities in an image, and the edge can provide the most basic and important information of an image.

At present, there are a variety of edge detection operators that can satisfy different kinds of requirements. These common edge detection operators can be classified as two forms. The one is first order differential which refers to Roberts operator[1], Sobel operator[2] and Prewitt operator[3];The other is the second-order differential operators which usually refers to Laplacian operator and LOG operator. All of these operators are simple, low computationally intensive and less time-consuming. However, in the presence of noise, these algorithms results suffer from missing edges and false edges. With further development of the edge detection operator later, Canny proposed the Canny operator[4], optimizing the performance of the algorithm with the effect of noise. The Canny algorithm is always developing, for example, statistical filtering is used to replace Gaussian filtering[5];wavelet transform is used to enhance the edge details[6];double thresholds are selected automatically by gray histogram concavity analysis[7];Canny operator has better noise immunity and produces good results, and it is still widely used in many field now.

What's more, there is now an edge detection algorithm based on the idea of fuzzy logic, using the Fuzzy Inference System(FIS),which has the ability to recognize image with the idea of artificial neural networks. It manages to get the result without other external data, like the brain of a living creature recognizing an image with only knowing the past experiences. It is a new image edge detection algorithm based on theories from the field of artificial intelligence, which plays a great role in the field of edge detection with its superior performance.

In this work, the main work has focused on the comparison of Canny edge detection algorithm and edge detection algorithm based on Fuzzy Logic. Section wise paper described as follows: In section 2,

DOI: 10.54254/2755-2721/4/2023422

principles and basic steps of both algorithms will be described, Section 3 provides the implementation and results in MATLAB environment and some discussions about them and finally the conclusion is in Section 4.

2. Principles and basic steps

2.1. Canny operator

The canny operator which were presented in 1986 showed three criteria[4] and detects images according to these three criteria. These three criteria focus on reducing noise interference and lowering the possibility of detecting not true boundaries, ensuring the edges processed by the algorithm is similar to the edges of original image and ensuring that the algorithm processes a unique boundary and that the boundary is a single pixel.

Based on the three criteria, there are four steps to implement the canny algorithm. A Gaussian smoothing filter is firstly needed to convolve with image to remove the noise from the image. Noise usually corresponds to the high frequency part of the image, So filtering out the high frequency part of the image with a Gaussian filter can have the effect of removing noise. The Canny operator first filters the image by convolving it with a two-dimensional Gaussian function, and the two-dimensional Gaussian function is shown in formula(1)

$$h(x,y) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{1}$$

Afterwards the magnitude and direction of the gradient will be computed. Gradients of two direction are shown in formula(2).

$$\nabla f = grad(f) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{2}$$

the vector amplitude of gradient can also be expressed in another form, which can be shown in formula(3),what’ s more, gradient direction can be shown in formula(4).

$$|\nabla f_{(1)}| = |G_x| + |G_y| \tag{3}$$

$$\varphi(x,y) = tan^{-1}\frac{G_y}{G_x} \tag{4}$$

Then, non-maximum suppression is needed to retain the true edge points. The reason for the implementation of non-maximum method is to compare pixel at the centre with neighbouring pixels in gradient direction. If the central pixel has a larger value, it is recognized as a edge pixel . If the central pixel value is smaller, the central pixel value will be set to zero. Lastly, based on the experience from previous experiments, two thresholds are set manually to identify the true edge pixels and remove the false edge, and then remove the non-connection. The edge detection image is obtained later.

2.2. Edge detection based on fuzzy logic

In this process, a sobel operator is required to applied to the original image to calculate the gradient of gray level of each pixel. The sobel edges operator uses a pair of 3x3 convolution masks, calculating the gradient respectively in the x-direction and in the y-direction.

The masks slide over the image, as it is much smaller than the picture, calculating a 3x3 square of pixels at one time. The masks are shown in formula(5)[8].

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{5}$$

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

DOI: 10.54254/2755-2721/4/2023422

Two images which include the derivative approximations in x-direction and y-direction at each point can be calculated as (6) and (7).

$$g_x = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobel_{x,i,j} * I_{r+i-2,c+j-2} \quad (6)$$

$$g_y = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobel_{y,i,j} * I_{r+i-2,c+j-2} \quad (7)$$

Where and are the gradients along axis-x and axis-y, I is the matrix of original image and * represents the convolution operator.

Then 4 inputs are required for the FIS, and the and computed with (6) and (7) will be called DH and DV respectively and they are the two of the inputs[9].The other two inputs are filters which use the masks to convolve with the original image. Two masks of the filters are shown as (8) and (9),and the filters are respectively high-pass filter and low-pass filter.

$$hHP = \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \quad (8)$$

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (9)$$

The purpose of the filter is to design a system that makes it easier to include edge information from low contrast areas without suffering from false edges.

What comes next is the fuzzy variables, for this part, all the membership functions are all Gaussian. Then the ranks are shown in figure 1, 2 and 3.

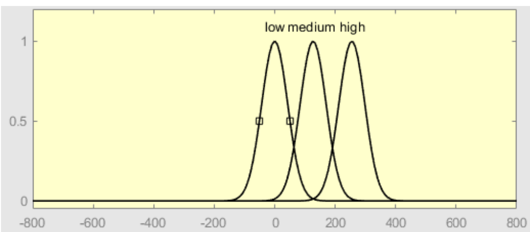

Figure 1. Input variable DH.

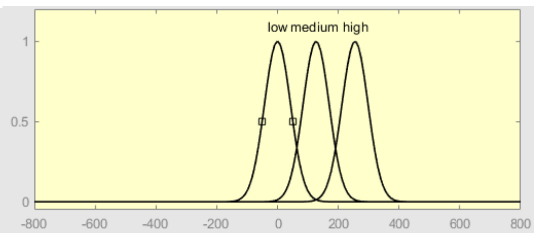

Figure 2. Input variable DV.

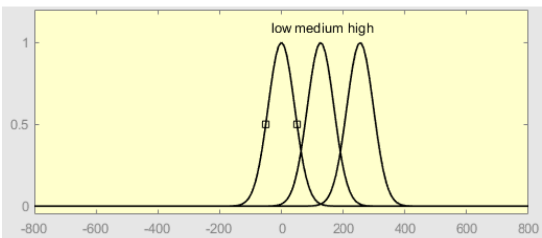

Figure 3. Input variable HP.

In case of the variable M, the ranks of x-axis of it is from 0 to 255 with advanced experiment, and it is shown in figure 4.

DOI: 10.54254/2755-2721/4/2023422

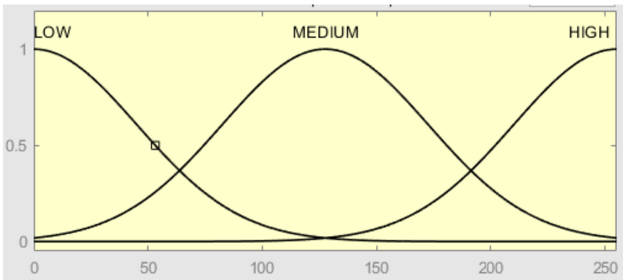

Figure 4. Input variable M.

Output EDGES with the rank from 0 to 255 is shown in figure 5.

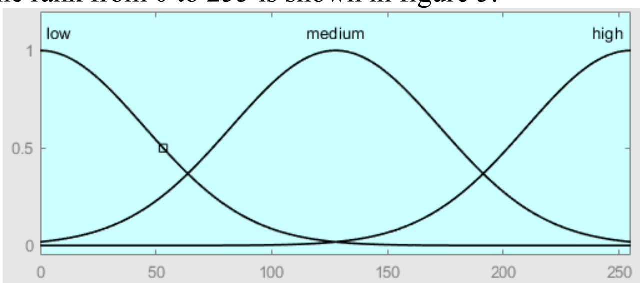

Figure 5. Input variable EDGES.

Finally the seven fuzzy rules are set to evaluate the input variables to identify which are the edges, and the result image shows edges near white in color and the background near black in color. The fuzzy rules are shown in figure 6.

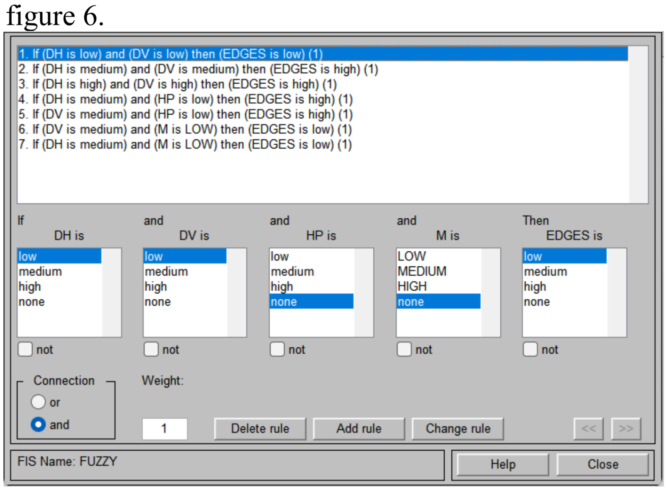

Figure 6. Fuzzy rules.

3. Implementation and results

For the purpose of implementation and promotion, the original image is in colour portraits of people, and it is shown in figure 7.

DOI: 10.54254/2755-2721/4/2023422

Figure 7. Original image.

All implementations are performed in the MATLAB (R2021b) environment, and for the fuzzy rules part, seven rules are set with the Matlab Fuzzy Logic Tool Box [10]. The simple steps of Canny operator and edge detection with fuzzy logic are shown respectively in figure 8 and 9.

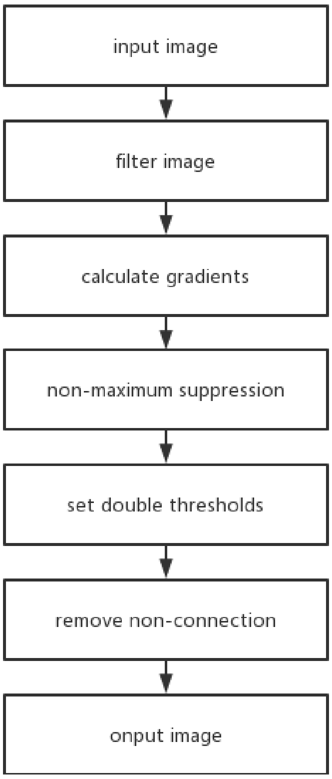

Figure 8. Steps of Canny operator.

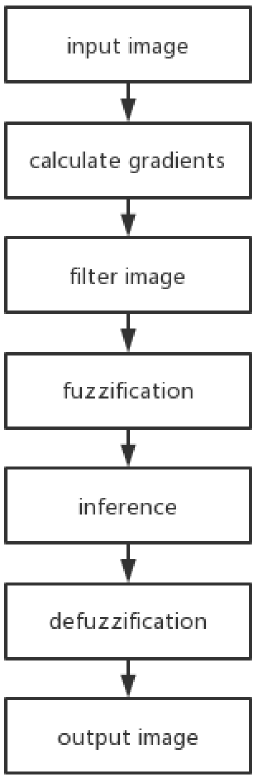

Figure 9. Steps of edge detection with fuzzy logic.

For the result obtained from Canny operator, shown in figure 10, it has more details in the edge part than that from FIS, and it has clearer edges. However, the performance of Canny operator can be bad when there are plenty of contents in image. What's more, for the result from edge detection with fuzzy logic, which is shown in figure 11, it has wider range of gray level, which makes the background brighter, so that it has a better visual effect. In addition, it has a thicker and smoother edge to the picture. But the result is that some detailed information is lost to a certain extent.

Figure 10. Result from canny.

Figure 11. Result from edge detection with fuzzy logic.

4. Conclusion

In this paper, two algorithms from different ideas are introduced. For the Canny operator, the gradients of x-direction and y-direction are calculated firstly, what comes next is the non-maximum suppression, and finally double thresholds and non-connection removal are used to improve the accuracy of edge extraction. For the edge detection based on fuzzy logic, the gradients of image are still calculated firstly, then the inputs of FIS should be calculated, and finally the results are obtained according to the fuzzy rules. For the results from Canny operator, they have more detailed edge but bad performance when there are too many contents in image. Results from the other idea seem to be good in visual effects, but they lost some detail in edge information to a certain extent. The results have been shown to facilitate the reader's understanding of the superiority of the each algorithm so that different algorithm can be chosen to suit different situations.

References

[1] Roberts L. Machine perception of 3-D solids[J]. Optical & Electro Optical Information Processing, 1962.
[2] None. Pattern classification and scene analysis[J]. Artificial Intelligence, 1973.
[3] Prewitt J. Object enhancement and extraction, Picture Processing and Psychopictorics. 1970.
[4] Canny J . A Computational Approach to Edge Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, PAMI-8(6):679-698.
[5] Duan Jun, Zhang Bo. Research on edge of improved Canny operator [J]. Software guide, 2018,17 (10): 68-71
[6] Duan Hongyan, Shao Hao, Zhang Shuzhen, Zhang Xiaoyu, Wang Xiaohong. An improved image edge detection algorithm based on Canny operator [J]. Journal of Shanghai Jiaotong University, 2016,50 (12): 1861-1865.
[7] Xu Liang, Wei Rui. Optimization algorithm of image edge detection based on Canny operator [J]. Science and Technology Bulletin, 2013,29(07): 127-131+150.
[8] Miosso C J, Bauchspiess A. Fuzzy inference system applied to edge detection in digital images.
[9] Mendoza O, Melin P, Licea G. A New Method for Edge Detection in Image Processing Using Interval Type-2 Fuzzy Logic[C]// 2007 IEEE International Conference on Granular Computing (GRC 2007). IEEE, 2007.
[10] The MathWorks, Inc. Fuzzy Logic Toolbox 2.2.4. http://www.mathworks.com/products/matlab/, 2006.