International Core Journal of Engineering Volume 5 Issue 9, 2019
ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

Design and Implementation of Edge Extraction Algorithm for Digital Image

Ji Zhang [a], Yajing Zhang [b]

Department of Computer Science, North China Electric Power University, Baoding 071000, China.

[a]72zhangji@163.com, [b]ncepuzyj@163.com

Abstract

With the development of mathematics and artificial intelligence, various types of edge algorithms are emerging, and theories such as neural networks are applied to image edge detection. However, due to the contradiction between detection accuracy, edge positioning accuracy and anti-noise, there are still some inconsistencies in edge detection. How to achieve accurate positioning and extraction of image edges has become an important issue that people must face. In this paper, the principle method of image edge extraction is deeply studied and explored. The main work is reflected in the following aspects:First, the meaning and prospect of image edge extraction are analyzed and discussed. Then introduce the basic theory of graphics extraction, enumerate and explain the current various extraction algorithms, focus on the canny algorithm and the robot algorithm, and obtain their extracted images through experimental simulation, from image extraction effect, peak signal-to-noise ratio and time overhead. Several parties have made comparisons. Secondly, the MATLAB experimental simulation platform is introduced, and the user interface design and development is carried out on this platform, and the code is given for some functions. Finally, this paper gives a comprehensive summary of the work of the thesis, and looks forward to the further research on image edge extraction.

Keywords

Vdeo denoising; salt and pepper noise; spatial filtering; wavelet filtering.

1. Introduction

Edge extraction refers to a process for a picture outline in a digital image processing [1]. In my opinion, the edge shows where the grayscale change is the largest image, that is, where the image grayscale changes most strongly. The discontinuity of the image gray scale on the surface changes the edges. As far as the general definition is concerned, the so-called edge extraction is the part that preserves the most dramatic change in the gray level in the image. If you use mathematical methods to solve this problem, the most straightforward and feasible method is to use differentiation (differential is used when the object is a continuous function, and when the object is in a discontinuous form).

With digital products and various digital devices entering people's daily lives, digital image processing technology has gradually penetrated into various fields such as industry, agriculture, animal husbandry, science and technology, military, geography, etc., and is closely related to people's daily life. When scientists use satellites to obtain Earth's resources and weather conditions, the hub of information is photo images. Similarly, doctors use X-ray or CT to analyze tomographic images of various organs of the human body. Images have become a very common and effective information

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

carrier in human activities. They can store a large number of object attributes and features, making them the primary means of obtaining raw information from the outside world. Therefore, technologies such as digitization of images, feature extraction of images, storage and transmission of images, and the like have been developed.

Image edge extraction is very useful for image recognition and computer analysis. On the one hand, the edge of the image can outline the target object, so that the observer can understand the target that he or she needs to observe most intuitively and clearly; at the same time, the edge of the image contains abundant connotation information (such as step nature and direction change). , the shape of the research object in the image, etc., these are extremely important features in image recognition technology. In essence, the edge of the image is the most intuitive and true reflection of the local characteristics of the image (grayscale, color, texture, etc.). In image analysis, the extraction and analysis of image edges is particularly important and important.

2. Related work

At present, the existing image edge extraction has formed a relatively mature research method system.

In 1963, Roberts first published an algorithm that explored the edges of images. The operator he proposed is actually a flat template, the principle of which is to use the adjacent pixel difference in the diagonal direction [2]. Such an operator using a local difference to find the edge of an image has the advantage of being more accurate in view of the processing effect of the image, but it also has the disadvantage of being sensitive. But it is undeniable that the presentation of such an operator is still a huge asset in the field of image research. For convenience of reference, this operator is named after its theoretical proposer and is called the Roberts operator.

In 1968, at the Doctoral thesis exchange seminar, the "A 3x3 Isotropic Gradient Operator for Image Processing" theory was proposed, but at the time this theory did not publish a paper as a molding theory. Etc. [3]. It was not until five years later that this theory emerged and became public in a footnote to a published monograph. Since then, this theoretical algorithm (the sobel operator mentioned later) has been learned and used as an important method in the field of image edge extraction.

In 1980, the famous scientist David Courtnay Marr and the famous computer related field scientist Ellen Hildreth proposed a new edge detection algorithm, LoG edge detection operator, which combines Laplacian algorithm and Gaussian algorithm. The algorithm first performs Gaussian filtering on the image and then finds the second-order derivative. Therefore, in order to facilitate the memory, this algorithm is simply referred to as the Laplacian-of-Gaussian operator, which is the LoG operator often mentioned today [4].

One of the proponents of the LoG operator, David Marr, is a well-known British neuroscientist and psychologist. When he was alive, he combined a series of cutting-edge human research results such as Psychology, Artificial Intelligence, and Neurophysiology to study a series of new visual processing models. His research has had a tremendous impact on the development of computational neuroscience. But unfortunately, this great scientist died in 1980, when he was only 35 years old.

Another contributor to the LoG algorithm, Ellen C. Hildreth, Ben, and Bo, studied at the Department of Computer Science at MIT. Hildreth is currently a professor of computing at Wellesley College and a scientist at the Massachusetts Institute of Technology.

The papers co-authored by the two were finally revised and accepted at the end of the 1970s. The following year, the paper was published in the journal. In 1980, when the paper was published, Marr, one of the creators, died of leukemia. The paper co-authored by them has been cited more than 6,000 times. According to the habit of naming algorithms by the name of the presenter, the LoG edge detection operator is also called Marr & Hildreth operator.

Then in 1986, John F. Canny discovered a algorithm for edge detection based on multi-level computational analysis. This algorithm is used for the convenience of memory, and in order to

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

commemorate the great achievements of its proponents, also the name of its presenter. Named the Canny operator. The proposed Canny algorithm provides a very convenient method for image edge extraction.

In addition, typical algorithms in the field of image edge detection include forming algorithms such as Prewitt operator and Laplacian operator. In the development of image edge extraction technology, many molding algorithms are not explicitly published and declared in papers or other forms, but the generation of each algorithm greatly promotes the development of image edge recognition and extraction technology.

3. Algorithm analysis

3.1 Canny algorithm

The Canny algorithm is the best edge detection algorithm recognized in all known mature edge detection algorithms. Although the time and space complexity of the algorithm is relatively higher than other operators, with the development of computer computing power, the current computing equipment is completely no problem for real-time implementation of Canny.

For the definition of the edge of digital images, Canny first proposed three basic conditions [11]:

1) Accurate detection: Improve the signal to noise ratio. This requires that you do not miss the edge points and add non-edge points as little as possible.

2) Accurate location: In the positional relationship, the edge points detected by the algorithm should be as close as possible to the true edge center.

3) Single response: For one edge, only one response can be generated.

The above is the guiding ideology of the Canny algorithm.

3.1.1 Image denoising using Gaussian convolution

There are usually two ways to achieve image Gaussian filtering: one is to use a one-dimensional Gaussian kernel for two convolutions; the other is to use a two-dimensional Gaussian kernel for one convolution.

1) Gaussian convolution kernel

$$K = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{3-1}$$

Equation (3-1) is a discretized one-dimensional Gaussian function. As long as the parameter value can be determined, a one-dimensional kernel can be obtained.

$$K = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3-2}$$

Equation (3-2) is a discretized two-dimensional Gaussian function. Similarly, as long as the value of the parameter is determined, a two-dimensional kernel vector can be obtained.

2) Image Gaussian filtering

Gaussian filtering of the image is actually an operation of adding the weight to the gray value in the current point and its neighbors and adding an average weight according to certain rules. This operation can effectively remove noise.

Under normal circumstances, filter denoising and edge detection are two mutually exclusive operations. The result of suppressing noise is often that the edges of the image are seriously damaged and the blurring is unclear, so that the edge uncertainty is enhanced. The certainty of the edge and the increased accuracy of the edge detection inevitably enhance the recognition sensitivity of the noise. However, actual experimental tests have shown that this method does enhance the accuracy of image edge detection while preventing noise interference. This is image Gaussian filtering.

3.1.2 Calculation of the amplitude and direction of the gradient

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

As mentioned above, after the image is digitized, a discrete matrix is formed instead of a continuous function. Therefore, if you want to obtain an image gradient, you can choose to use the first-order difference method for approximate calculation. By the first-order difference, two matrices composed of partial derivatives of the target image in the horizontal and vertical directions can be obtained. Expressed as formula (3-3) and formula (3-4):

$$S_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \tag{3-3}$$

$$S_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \tag{3-4}$$

The horizontal first-order partial directional array (ie, P[i,j] of equation (3-5)) and the vertical first-order partial directional array (ie, Q[i,j] of equation (3-6)) The gradient magnitude ((M[i, j]) of the equation (3-7)) and the gradient direction (ie Θ[i, j] of the equation (3-8)) are expressed as follows:

$$P[i,j] = \frac{(f[i,j+1]-f[i,j]+f[i+1,j+1]-f[i+1,j])}{2} \tag{3-5}$$

$$Q[i,j] = \frac{(f[i,j]-f[i+1,j]+f[i,j+1]-f[i+1,j+1])}{2} \tag{3-6}$$

$$M[i,j] = \sqrt{P[i,j]^2 + Q[i,j]^2} \tag{3-7}$$

$$\theta[i,j] = \arctan(Q[i,j]/P[i,j]) \tag{3-8}$$

After calculating these matrices by calculation, the calculation of the gradient amplitude and direction has been completed, and the next detection process can be performed.

3.1.3 Non-maximal suppression of gradient amplitude

In the paper of the Canny operator proposed by John anny, the gradient non-maximum suppression is performed in only four directions, that is, for each pixel currently studied, its gradient direction is similar to the four The gradient direction is selected and replaced by the final gradient direction. In this case, the points compared to the non-maximum suppression are:

1) 0: left and right;

2) 45: upper right and lower left;

3) 90: upper and lower;

4) 135: upper left and lower right.

The most intuitive benefit of this approach is that it is simple to operate and computationally intensive, but since the gradient directions in natural images do not all follow these four directions, this simplified approach can hardly achieve the best results. Therefore, it is necessary to interpolate the pixels in the neighborhood at this time.

At the same time, in practice, the resulting image is digitized to form a matrix of pixels, which means that the resulting image is a discrete point rather than a continuous function. For this reason, the gradient of the pixel points currently studied by difference is actually largely unable to continue its extra gradient point. In most cases, the point on both sides of the gradient direction at point c (ie, the current research object point) at the center position is a subpixel. The so-called sub-pixel point is the point where the theory exists and does not actually exist.

According to the gradient definition, it is known that the maximum point can only appear at c and the two sub-pixel points, so the determination of these two sub-pixel points is particularly important.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

In fact, by means of interpolation, these two sub-pixel points can be obtained.

Now suppose that the pixel point of interest is c point, and the gradient direction is divided into a horizontal direction g_x and a vertical direction g_y in the area. If $|g_y|>|g_x|$, this indicates that the gradient direction of point c is closer to the vertical direction (the direction of the Y-axis), you can use Fig. 3-1 to illustrate the two cases (the direction is the same and the direction is different):

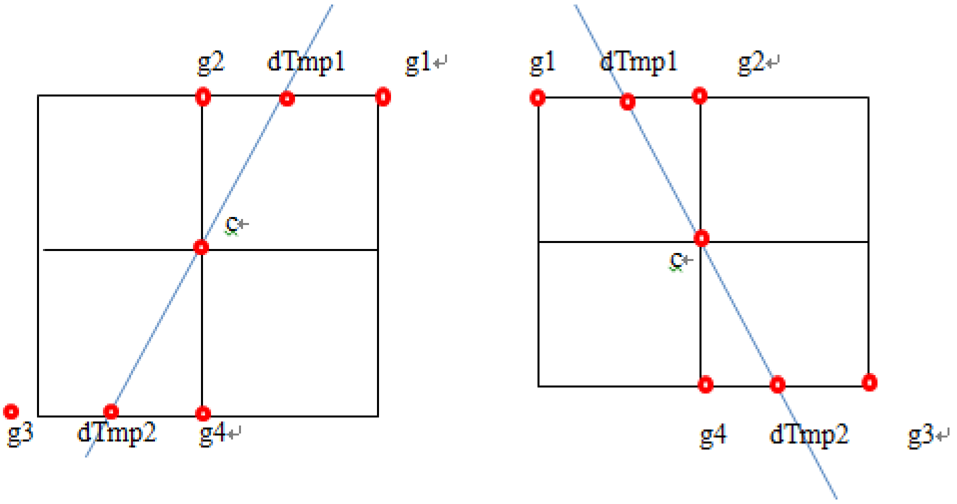

Fig 3-1. Gradient amplitude non-maximum suppression (1)

In Fig. 3-1, we can see that C represents the center position point; the slanted line represents the gradient of the current point c, the first one simulates the anomaly of g_y and g_x, and the second one simulates the same direction of g_y and g_x. At this time, the weight w is as shown in equation (3-9):

$$w = |g_x| / |g_y| \quad (3\text{-}9)$$

For the same reason, you can get the case of $|g_x|>|g_y|$, where the gradient of the point is closer to the horizontal (X-axis) direction, which is illustrated in Fig. 3-2:

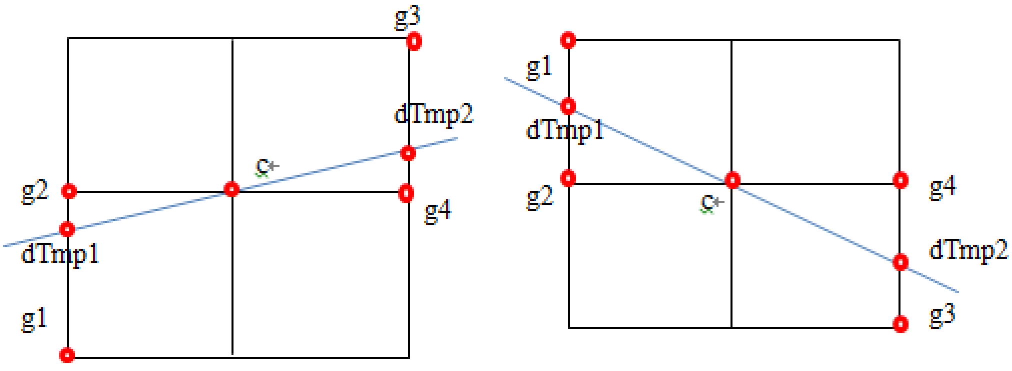

Fig 3-2. Gradient amplitude non-maximum suppression (2)

In Fig. 3-2, C still represents the center position point, the gradient is the gradient of c, the first simulation g_y and g_x square, and the second simulation g_y and g_x, the weight w as (3-10):

$$w = |g_y| / |g_x| \quad (3\text{-}10)$$

Through the above analysis, we can understand the necessary interpolation in the Canny operator as follows:

1) Four directions are used in the Canny algorithm to determine the gradient. However, since the edge gradient direction does not necessarily follow a predetermined direction in the actual image, interpolation must be performed to find the value that best fits both sides of the current point gradient.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

2) Since the images obtained by digitizing in practice are all discrete matrices, the points on both sides of the gradient at c are not necessarily present, and the gradient values of this point must be obtained by interpolating the points on both sides.

The suspect points dTemp1 and dTemp2 are obtained by the difference. Next, it is only necessary to compare the gradient magnitude at the central position C with the gradient at the two interpolation points to determine whether it is an extreme point.

3.1.4 Double threshold algorithm to determine the edge

Although the non-maximum suppression of the gradient magnitude has been performed, there are still many "false edges" in the resulting "gradient magnitude" maximal matrix, ie, the region is extremely large and not the edge interference value. It should be noted that these “false edges” are not completely meaningless. Their existence is crucial for the edge connection, but these “false edges” are not true edges, and their existence must satisfy “do not interfere with the real The premise that the edge determines the image.

The Canny algorithm chooses to reduce the "false edge" by using the dual threshold algorithm. The double-threshold algorithm simply selects two thresholds (high and low, respectively). By filtering the high value, you can get an edge image (almost all edges selected by the high threshold are edges), but due to the threshold. If the edge is selected, the edge of the image generated by the edge is not connected. In order to ensure the continuity of the edge, a low value is selected while selecting the high value.

In the high-value image, all the points are connected into a contour. When the breakpoint is reached, the algorithm finds a point that can satisfy the low value in the adjacent area of the breakpoint as a supplementary point of the breakpoint added to the edge set, so that the loop is repeated until The entire image is closed.

3.2 Sobel algorithm

Sobel operator is one of the most important operators in pixel image edge detection, and plays an important role in information technology such as machine learning, digital media and computer vision.

3.2.1 Calculation method

The algorithm of the sobel operator is: 1. After adding weights, the mean is obtained; 2. For differentiation. The specific operation is as shown in equations (3-11) and (3-12):

$$\Delta_x f(x,y) = [f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1)] - [f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)] \quad (3\text{-}11)$$

$$\Delta_y f(x,y) = [f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)] - [f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)] \quad (3\text{-}12)$$

The template for the Sobel operator to detect edges in the vertical and horizontal directions of the current pixel is shown in Figure 3-4. The former is used to detect the lateral edges in the image and the latter is used to detect the longitudinal edges in the image.

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

Fig. 3-4 Sobel operator template

ISSN: 2414-1895
DOI: 10.6919/ICJE.201908_5(9).0016

In fact, in practical applications, the maximum value of the two template operations is taken as the value of the point for each point. The result of the operation after such processing is the edge image.

3.2.2 Edge processing

Before introducing the concept of refining edges, we must first understand two other concepts: goals and background.

The so-called "target" is again referred to as "foreground", which refers to the part of the content that people are interested in. Conversely, the content of the rest of the image that is not of interest is referred to as the "background."

Because of the existence of the target and the background, image segmentation is involved in image processing. The so-called image segmentation is the operation of dividing an image into a target and a background. Image segmentation is an indispensable key step in both image processing techniques and image analysis techniques.

In recent years, image segmentation technology has been highly valued by researchers in related fields (such as computer vision, etc.). To date, more than a thousand mature and reliable algorithms have been proposed in the field of image segmentation.

The advantage of the Sobel operator is that it is simple, the image is processed fast, and the edges are smooth and continuous. This is due to the good choice of the image segmentation technique by the sobel operator. This kind of operator chooses to use the threshold method for image segmentation, which makes the time and space complexity occupied by the operator for image processing is extremely low. However, the algorithm also shows obvious shortcomings, such as the edge is relatively thick, and because the processing needs to be binarized, the edge extraction is greatly affected by the threshold selection.

3.2.3 Edge processing

The Sobel operator contains two sets of cubic scales, horizontal (horizontal) and vertical (vertical). The two arrays are convolved with the image to obtain the difference between the two directions. Now suppose that if A represents the original image, and respectively detect the array, then the calculation method is as shown in equations (3-13) and (3-14):

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \tag{3-13}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{3-14}$$

The gradient value of each point of the image can be calculated by the formula (3-15):

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{3-15}$$

The gradient direction can then be calculated using equation (3-16).

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{3-16}$$

4. Experiment procedure

The graphical user interface is designed to facilitate various processing and operations on the image, and to implement various image edge extraction algorithms through the interface, and obtain edge results. This would be more efficient for using the Command Window to run m files and execute Matlab commands, reducing many manual operations.

1) Visualization in MATLAB

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

MATLAB contains a complete package of visualization functions. In this paper, only the uigetfile() function is used to achieve simple human-computer interaction to achieve the purpose of autonomously selecting images [16].

The usage of this function is: [FN, PN, FI] = uigetfile(FS, DT, DN);

Where: FN: "file name"; PN: "file path name"; FI: "selected file type"; FS: "file type"; DT: "dialog title"; "DN: "default file name" (under with).

There are several typical uses when using the uigetfile() function:

i. Set a single file type, such as: [FN, PN] = uigetfile('*.jpg', 'choose your file');

ii. Set multiple file types, such as: [FN, PN] = uigetfile({'*.jpg';'*.png';;'*.*'},' choose your file ');

iii. Set the appropriate description for the file type, such as:

```
[FN, PN] = uigetfile(
{'*.jpg;*.fig;*.avi;*.mdl','Files (*.jpg,*.fig,*.avi,*.mdl)';
‘*.jpg','files1 (*.jpg)';
'*.fig',' files2 (*.fig)';
‘*.mat','files3 (*.mat)';
‘*.mdl',' files4 (*.mdl)';
‘*.*',' Files (*.*)'},
'Pick a file');
```

(This method should be noted when using it. It should be written strictly in this format!)

iv. Return the file type and set the file multiple selection. Such as:

```
[FN, PN,FI]   = uigetfile(
{'*.mat','Mfiles (*.mat)';
'*.mdl','Models (*.mdl)';
'*.*','all (*.*)'},'oxb','find', 'on');
```

(Please pay attention to the writing format when using this method!)

v. You can set the default file name, the implementation code is as follows:

```
uigetfile({'*.tif;*.png;*.PDF;*.avi','who pa who';
'*.*','all' },'biaoti','D:\Work\picture.jpg')
```

The code that uses this function in the experiment is as follows:

```
[FN, PN] = uigetfile('*.*', 'Read Image');
if isequal(filename,0)
   msgbox(' No pictures were selected ');
else
    p=fullfile(pathname,filename);
    ImageCheck=imread(p);
end
```

2) Graphic display interface

The function used in the graph used in the experiment is the built-in function imshow() in MATLAB.

imshow() is a function used to display images in MATLAB. When the function is used, the image will be displayed according to the specified range [lh]. When the gray value in the image is equal to or lower than the value of l, the function will be processed to display in black; and when the gray value in the image is greater than or equal to h, the image will be used. The processing will be white for all; when the gray value in the image is between l and h, the function will be displayed with the midtone of its default value of gray level.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

Note the following when using this function:

imshow(K): Display K directly, without any processing;

imshow(K,[]): Set K's h and l values to pure white and pure black respectively when K is displayed (the gray value of pure white in MATLAB is 255, and the gray value of pure black is 0), each value of K is mapped to a gray value between 0 and 255. Use the imshow() function to display the image as follows:

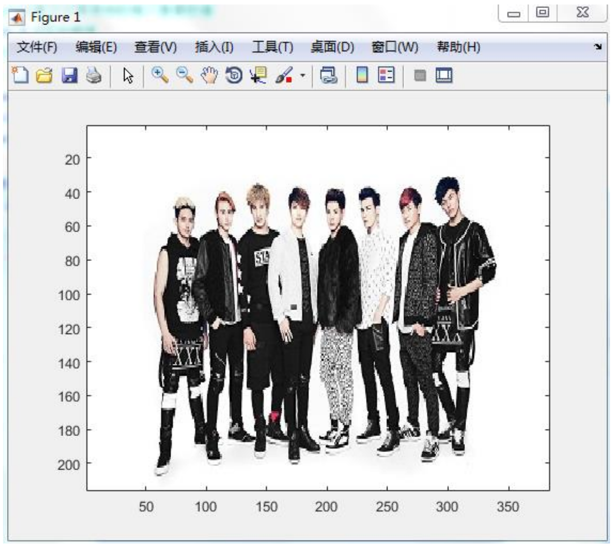

Fig 4-1. Image display

3) Matrix normalization

In the implementation of algorithms such as sobel operator and LoG operator, it is necessary to normalize the matrix formed by image transformation. The mat2gray() function is used here to achieve normalization.

"Normalization" means that the value of each element in the object matrix is between 0 and 1. The function is used as follows:

i. I = mat2gray(A, [m1 m2]): normalizes the data between m1 and m2 in image matrix A. All elements smaller than m1 are updated to 0, and elements larger than m2 are adjusted to 1.

ii. I = mat2gray(A): Normalize the image matrix A. After normalization, each element is in the range 0 to 1 (including 0 and 1). Here 0 is black and 1 is white.

5. Conclusion

In the field of digital image, the theory of image edge extraction has important significance, and it has been receiving much attention. The main goal of image edge extraction is to preserve image features as much as possible, thereby providing maximum image information for further analysis of the image. Over the years, the theory of image edge extraction has been greatly developed and widely used, and many mature algorithms have emerged. This paper has carried out related research and analysis on image edge extraction. The main work is reflected in the following aspects:

1) Introduced the practicality of image edge extraction, development at home and abroad. Emphasis is placed on the scientific research in this direction.

2) The basic theoretical process of image edge extraction is briefly analyzed, and the image denoising part and image gradient value calculation part are analyzed.

3) Several algorithms with wide application and strong practicability in image edge extraction, such as canny algorithm, sobel operator and LoG algorithm, are introduced in detail. And make a reasonable theoretical analysis of several algorithms and a visual comparison of the final extraction results.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

4) Explain the MATLAB experimental simulation platform used in this paper, and use this platform to implement the algorithm implementation and interactive interface. A brief introduction to its development process and a brief code for some of the features.

The research content involved in image edge extraction is very extensive. Due to limited time, this paper is only theoretically elaborated and verified by the experimental platform. In view of the current lack of theoretical level and practical experience, this paper There are still many places where work needs to be improved and improved. I think the following topics need further study:

1) In the image edge extraction algorithm, this paper only analyzes and studies the existing forming algorithms, and does not propose a reasonable solution to the shortcomings of the algorithm.

2) In the interactive interface design of MATLAB, I failed to implement a complete interactive interface package, and only realized a simple picture self-selection window.

3) In the process of algorithm simulation, the determination of the threshold value of some algorithms and the determination of the range of the final edge extraction error value of the image are still not accurate enough.

Due to the above problems, I hope to study further in future study.

References

[1] Ji Hu, Sun Jixiang, Shao Xiaofang, Mao Ling. Image Edge Extraction Method and Prospects[J]. Computer Engineering and Applications, 2004(14): 70-73.

[2] Wang Bing. Edge Processing with Roberts Operator[J]. Gansu Science and Technology, 2008(10): 18-20.

[3] Popular Science: LoG Edge Detection Operator. Science Network Blog, 2014. 05. 02

[4] YUAN Chunlan, XIONG Zonglong, ZHOU Xuexue, PENG Xiaohui. Research on image edge detection based on Sobel operator[J]. LASER & IR, 2009, 39(01): 85-87.

[5] Zhang Anding. Remote sensing principle and application problem solution [M]. Beijing: Science Press, 2016.

[6] Liu Zhihui, Li Hongwei, Bian Jiawen. Two-Dimensional Harmonic Frequency Estimation in Multiplicative and Additive Noise[J]. Signal Processing, 2012, 28(4): 495-499.

[7] Wang Jianfeng, Zhao Xiaorong, Li Li, et al. Image Smoothing Experiment Design Based on Median Filtering and Matlab Implementation[J]. Journal of Nanjing Institute of Industry Technology,

[8] Meng Rui. Research and implementation of image denoising based on wavelet transform [D]. Northeastern University, 2009.

[9] Song Guohua, Cui Jingan. The second edition of Advanced Mathematics: Petroleum Industry Press, 2013.

[10] Department of Applied Mathematics, Tongji University. Under Advanced Mathematics [M]. Beijing: Higher Education Press, 2007.

[11] Qiu Chunting, Liu Hongyan, Qi Jing. An Improved Canny Edge Detection Method[J]. Journal of Basic Science in Textile Universities, 2014, 27(03): 380-384.

[12] Jair Trapé Goulart, Rosana Almada Bassani, José Wilson Magalhães Bassani. Application Based on The Canny Edge Detection Algorithm for Recording Contractions ff Isolated Cardiac Myocytes [J]. Computers in Biology and Medicine, 2017, 81: 106-110.

[13] Yan Yanhong. Research on image edge detection based on improved canny algorithm [D]. Chongqing Normal University, 2011.

[14] Marr D, Hildreth E. Theory of Edge Detection [J]. Proceedings of the Royal Society of London, 1980, 207 (1167): 187-217.

[15] Fu Peng, Sun Quansen, Ji Zexuan, Chen Qiang. A Signal to Noise Ratio Estimation and Measurement Criterion for Remote Sensing Images[J]. Acta Geodaetica Sinica, 2013, 42(04): 559-567.

[16] Lan Hong, Tian Jin, Li Shuzhi, Liu Lihui. Design of Image Processing Platform Based on Matlab GUI[J]. Journal of Jiangxi University of Science and Technology, 2014, 35(03): 79-84.

[17] Tu Wangming, Wei Youguo, Shi Shaomin. Application of MATLAB in Digital Image Processing[J]. Microcomputer Information, 2007(06): 299-300+23.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201908_5(9).0016

[18] Kong Yinchang. Implementation of Digital Image Edge Extraction Operator Based on Matlab[J]. Value Engineering, 2012(15): 171-172.

[19] Zhou Chao. An edge extraction algorithm based on Canny theory[J]. Science & Technology Information, 2012(8): 8-10.

[20] [US] Rafael C. Gomzalez, et al., Qiu Qiuqi translation. Digital Image Processing (Matlab Edition) [M]. Beijing: Publishing House of Electronics Industry, 2012.

[21] Jiangnan Yanyu. Detailed explanation of Canny edge detection [Z]. CSDN, 2011.

[22] Wang Xiaojun, Liu Xumin, Guan Yong. Image Edge Detection Algorithm Based on Improved Canny Operator[J]. Computer Engineering, 2012, 38(14): 196-198+202.

[23] Jing Yu. Research on edge detection algorithm for marine oil spill remote sensing image [D]. Dalian Maritime University, 2011.

[24] Fen Zhang, Naosheng Qiao, Jianfeng Li. A PCB Photoelectric Image Edge Information Detection Method [J]. Optik-International Journal for Light and Electron Optics, 2017, 144: 642-646.

[25] Wen Jiang, Haoyu Zhou, Yue Shen, Bo Liu, Zigang Fu. Image Segmentation with Pulse-Coupled Neural Network and Canny Operators[J]. Computers and Electrical Engineering, 2015, 46:528-532+538

[26] Long Xiang, Wu Xiaoqing. Motion Segmentation Based on Fusion of MSRF Segmentation and Canny Operator [J]. Procedia Engineering, 2011, 15: 1637-1641

[27] WeiQing Wang, HuaFeng Ding, XingJiang Xiao, Jun Xu, Yuan Zhang. The Improved Algorithm of Edge Detection Based on Canny Operator[J]. Advanced Materials Research, 2011, 1105(179): 97-102

[28] Ye Tian, Hua Qian, QiuHong Ke. A Bubble Detection Algorithm Based on Wavelet Transform and Canny Operator for Deinked Pulp Flotation Column [J]. Applied Mechanics and Materials, 2013, 2218(278): 1162-1166

[29] TAO Xiaoming, LI Yunhong, LIU Xu, et al. Edge detection algorithm based on Canny operator and threshold segmentation[J]. Journal of Xi'an Engineering University, 2014, 28(06): 745-749.

[30] Wang Zuocheng, Liu Xiaodong, Xue Lixia. An improved method for edge detection of Canny operator[J]. Computer Engineering and Applications, 2010, 46(34): 202-204+248.