

ACCURATE DETECTION OF IMAGE EDGE: COMPARISON OF DIFFERENT ALGORITHMS

Ronghui Fu

College of Computer Science, Neijiang Normal University, Neijiang 641100, Sichuan, China

Email: frongh@yeah.net

Abstract - In this paper, three edge detection algorithms based on LOG operator, Canny operator and wavelet transform were introduced briefly. Then, three edge detection algorithms were simulated and analysed by MATLAB. The subjects of analysis were three kinds of pictures, vehicles, ships and animals, and each group had three pieces. The results showed that the image edge obtained by the algorithm based on LOG operator had general outline but insufficient details, the image edge contour and details obtained by the Canny operator algorithm were relatively complementary, and the image edge obtained by the wavelet transform algorithm had more abundant contour and details. In terms of performance, the algorithm based on wavelet transform had the highest peak signal-to-noise ratio (PSNR), followed by the algorithm based on Canny operator and the algorithm based on LOG operator; in terms of detection time, the algorithm based on LOG operation was the longest, followed by the algorithm based on Canny operator and the algorithm based on wavelet transform.

Keywords: Edge Detection, LOG Operator, Canny Operator, Wavelet Transform.

1. Introduction

Fast and accurate image recognition has been widely used in various fields, such as traffic license plate recognition, face recognition, medical image marking, etc., and for fast image recognition, image edge detection is a very important part [1].

For an image, edges contain a lot of important information: edges play a role of segmenting regions, and edges represent the beginning and end of the regions on both sides [2].

In the region of an image itself, the change of the pixel value is relatively stable, while the change of the pixel value at the edge is usually intensive and transitional. Through the fast and accurate detection of the edge in the image, the relevant image recognition system can quickly and accurately divide the image into regions and then analyze different regions according to the needs. Moreover, the contour formed by the edge of the image can also reflect the details in the image [3].

At present, the basic principle of the commonly used edge detection algorithm comes from "the change of the pixel value at the edge is usually intensive" mentioned above.

Therefore, the method of deriving the gray value of images is used to find the edge point. Gu et al. [4] carried out image edge detection using artificial neural network and parallel computing, compared it with the traditional edge detection algorithm, and found that the new method had higher detection efficiency and accuracy.

Yu et al. [5] proposed a microscopic image edge detection algorithm based on Sum and Difference of Neighborhoods along Axis (SDNNA) and found through the experiment that the algorithm had strong anti-noise ability and high positioning accuracy and was suitable for images with different resolutions.

Shehu et al. [6] proposed an improved fast fuzzy algorithm to deal with images with multi-level edges, and the algorithm could adaptively select parameters according to different gray areas. In this paper, three edge detection algorithms based on LOG operator, Canny operator and wavelet transform were introduced, and then three edge detection algorithms were simulated by MATLAB.

2. Three Edge Detection Algorithms

• Edge detection algorithm based on Log operator

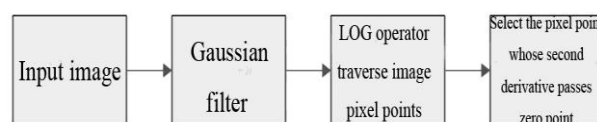


Figure 1: The flow of edge detection algorithm based on Log operator

The calculation flow of edge detection based on LOG operator is shown in Figure 1.

Firstly, image $f(x, y)$ is input.

Secondly, as the second derivative is the second derivation of the image function, part of the "noise" contained in the image will be amplified.

When using the zero point of the second derivative to find the edge point, the "noise" will seriously interfere with the accuracy of the search, and when it is serious, the "double edge image" will be detected. In this paper, the image is denoised using the Gaussian filter [7], and the calculation formula is:

$$G(x, y) = \frac{\exp(-\frac{x^2 + y^2}{2\sigma^2})}{2\pi\sigma^2} \quad (1)$$

where $G(x, y)$ is the image processed by denoising, σ is the standard deviation following Gaussian distribution, and (x, y) is the pixel coordinates of the image.

$$\nabla^2 G(x, y) = G(x+1, y) + G(x-1, y) + G(x, y+1) + G(x, y-1) - 4G(x, y) \quad (3)$$

The LOG operator is:

$$\nabla^2 h(x, y) = \frac{(\frac{x^2 + y^2}{\sigma^2} - 2) \cdot \exp(-\frac{x^2 + y^2}{2\sigma^2})}{2\pi\sigma^4} \quad (4)$$

After all the pixels in the image are traversed by the above LOG operator, the second derivative of the image pixel is obtained.

The pixel with the second derivative crossing the zero point is selected, and the pixel point is the edge of the image.

• Edge detection algorithm based on Canny operator

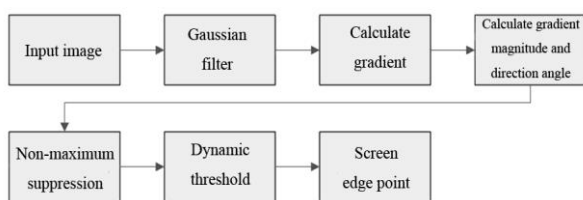


Figure 2: The flow of edge detection algorithm based on Canny operator

The calculation flow of the edge detection algorithm based on Canny operator is shown in Figure 2. Firstly image $f(x, y)$ is input. In order to improve the accuracy of edge detection and reduce noise interference, the image is denoised by Gaussian filter, and the denoising formula is shown in equation (1) above.

After image denoising, the horizontal and vertical gradients of each pixel in the image are calculated using a 3×3 Sobel operator template [9]. The calculation formula is:

LOG operator is the product of combining Laplace operator with Gaussian function. Laplace operator and Gaussian function are:

$$\begin{cases} \nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} \\ h(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \end{cases} \quad (2)$$

where ∇^2 is the Laplace operator [8] and $h(x, y)$ is a Gaussian function. As the image used for computer processing is a digital image composed of pixels, the image function is the discrete function.

In addition to the Laplace operator and Gaussian function mentioned above, it is also need to approximately solve LOG operator based on image neighborhood and equation (3):

$$\begin{cases} G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \otimes f'(x, y) \\ G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \otimes f'(x, y) \end{cases} \quad (5)$$

where G_x is the horizontal gradient of image pixels, G_y is the vertical gradient of image pixels, $f'(x, y)$ is the denoised image. After obtaining the set of horizontal and vertical gradients of image pixels, the gradient amplitude and direction angle of each pixel point are calculated respectively using the following equation [10]:

$$\begin{cases} |G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| \\ \theta = \arctan\left(\frac{G_y}{G_x}\right) \end{cases} \quad (6)$$

where $|G|$ is the gradient amplitude of the pixel. Theoretically, the quadratic sum of the horizontal and vertical gradients of the pixel needs to be squared; in order to reduce the calculation amount and improve calculation efficiency, the gradient amplitude is approximately calculated by the sum of the absolute values of the horizontal and vertical gradients. θ is the gradient direction angle of the pixel.

Then the edge points are extracted using non-maximum suppression algorithm. The calculation method is to compare the gradient amplitude of the pixel points on the same gradient direction angle, keep the pixel point with the largest gradient amplitude as the edge point, and set the rest pixel

points to zero. Then the dynamic threshold is calculated:

$$T = avg + 2\sigma + o \quad (7)$$

where T is a threshold, avg is the average gradient of pixels higher than the average gradient of the whole image, σ is the standard deviation between the gradient of pixel point larger than avg and avg , and o is a constant term.

The pixels of the image are screened by the calculated dynamic threshold, so as to supplement the edge points extracted by the non-maximum suppression algorithm and make the edges as continuous as possible. If the gradient of a pixel is greater than the threshold, it is considered as an edge point; if a pixel whose gradient is smaller than threshold is connected with an edge point, it is considered as an edge point; otherwise, it is considered as a non-edge point.

• Edge detection algorithm based on wavelet transform

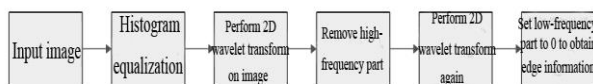


Figure 3: The flow of the edge detection algorithm based on wavelet transform

In this paper, the edge detection algorithm based on wavelet transform uses histogram equalization to reduce the impact of noise in the image and then uses wavelet transform to remove noise, and the effective edge information is remained. The algorithm flow is shown in Figure 3.

Firstly, the image is input, and then histogram equalization is performed to enlarge the gray range of the image and reduce the influence of noise. The formula of histogram equalization [11] is:

$$s = \int_0^r p_r(\omega) d\omega \quad (8)$$

where s is the gray value after histogram equalization, r is the gray value before histogram equalization, and $p_r(\omega)$ is the probability density that gray level is r in the image. In practical application, the image processed by computer is a digital image composed of pixel points, which is a discrete image. Therefore, equation (8) needs to be discretized, which is expressed as:

$$s_k = \sum_{j=0}^k \frac{n_j}{n} \quad k = 0,1,2,3,\dots,L-1 \quad (9)$$

where s_k is the gray value after transformation, n_j is the number of gray value as r_k , n is the number of the total pixels in the image, and L is the number of gray values in the image.

After the image histogram is equalized, the image is processed by two-dimensional wavelet transform. Firstly, the image horizontal information is processed by wavelet transform to obtain low-frequency signals and high-frequency signals by wavelet transform. Then the image longitudinal information of low-frequency signal and high-frequency signals is processed by wavelet transform respectively.

The formula of wavelet transform [12] is as follows:

$$\begin{cases} c_{j-1}(k) = \sum_m h(m-2k)c_j(m) \\ d_{j-1}(k) = \sum_m g(m-2k)c_j(m) \end{cases} \quad (10)$$

where $c_{j-1}(k), d_{j-1}(k)$ are low frequency and high frequency signals obtained after wavelet decomposition, $c_j(m)$ is the signal before wavelet decomposition, and $h(\bullet)$ and $g(\bullet)$ are low and high-pass filters.

After the two-dimensional wavelet transform, the high-frequency part is removed, and then the image is processed by two-dimensional wavelet transform again. Four groups of signals are obtained.

After the low-frequency part is set to zero, the edge information of the image is obtained.

3. Simulation Analysis

• Experimental environment

In this paper, the above three edge detection algorithms were simulated and analyzed using MATLAB software [13]. The experiment was carried out in a laboratory server.

The server configurations were Windows 7 system, I7 processor and 16 G memory.

• Experimental data

In this paper, three images with different contents were selected, including vehicle, ship and animal. The pictures were all in bmp format, and the size was 600×512 . In order to detect the edge detection performance of three edge detection algorithms under noise interference, each image was added with a Gaussian noise of $\sigma^2 = 0.02$ [14].

• Experimental method

Firstly, three edge detection algorithms were written with MATLAB software, and then gray-scale preprocessing was carried out for the pictures with Gaussian noise added. Then, three edge detection algorithms were used to detect the edge of three kinds of group pictures (nine pictures in total).

The pictures after edge detection were obtained, and the detection time was recorded.

• **Evaluation indicator**

In this paper, the performance of three edge detection algorithms was detected using the peak signal-to-noise ratio (PSNR) [15], and its formula is:

$$PSNR = 10 \times \lg \frac{255 \times 255}{\left(\sum_{x=1}^m \sum_{y=1}^n (I'(x,y) - I(x,y))^2 \right) / (m \times n)} \quad (11)$$

where *PSNR* is the peak signal-to-noise ratio, $m \times n$ is the image size, $I(x, y)$ is the original image after adding Gaussian noise, and $I'(x, y)$ is the image processed by edge detection algorithm.

• **Experimental results**

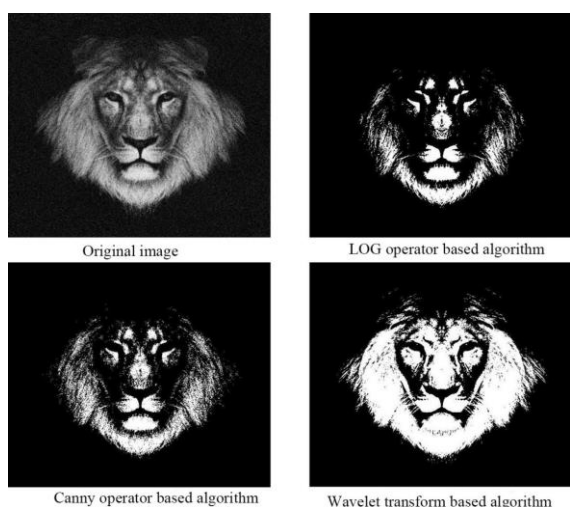


Figure 4: The original image after noise addition and images processed by three edge algorithms

Limited by the space, this paper only shows a group of animal images: the original image with noise and images processed by three edge detection algorithms.

As shown in Figure 4, the noisy original image used for algorithm performance detection is a lion's face, which becomes fuzzy after Gaussian noise was added. As to the noisy original image processed by the edge detection algorithm based on the LOG operator, the naked eye could see the approximate shape of the lion's front face directly, the lion's front face could be barely recognized at the first sight, but there were many details not detected compared with the original image. As to the noisy original image processed by the edge detection algorithm based on the Canny operator, it was similar to the result of the algorithm based on LOG operator, i.e., the lion's face could be distinguished at the first sight; compared with the result of the algorithm based on LOG operator, the details of edge image were richer, especially the area of mane was larger. As to the noisy original image processed by the edge detection algorithm based on wavelet transform, the front face of a lion could be clearly seen; compared with the other two algorithms, the details were more abundant; compared with the original image, the

extracted edge was also closer, especially the mane was very close to the area of the original image.

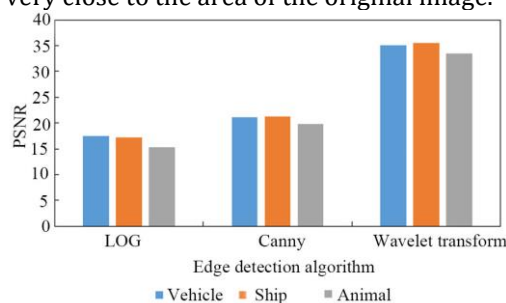


Figure 5: PSNR comparison between three edge detection algorithms

As shown in Figure 5, the average PSNR of the edge detection algorithm based on LOG operator was 17.45 for vehicle group image detection, 17.21 for ship group image detection and 15.36 for animal group image detection; the average PSNR of the edge detection algorithm based on Canny operator was 21.15 for vehicle group image detection, 21.31 for ship group image detection and 19.88 for animal group image detection; the average PSNR of the edge detection algorithm based on wavelet transform was 35.12 for the vehicle group image detection, 35.48 for ship group image detection and 33.54 for animal group image detection. It was seen from Figure 5 that the average PSNR of the three edge detection algorithms based on LOG operator was the lowest, that based on Canny operator was the second, and that based on wavelet transform was the highest.

After comparing the PSNR between the three groups of images under the same edge detection algorithm, it was found that the PSNR of the vehicle group and the ship group images was relatively close to that of the original image after processing, while the PSNR of the animal group images processed by the algorithm was significantly smaller compared to the other two kinds of group images.

The reason lied in the fact that vehicles and ships which are artificial creation with relatively regular appearance and details and few changes in complex details had relatively clear edge, but there were the arc-shaped or irregular edge lines in the images of animals, and the animal hair under the effect of light increased more irregular edges to the image, which affected the performance of the edge detection algorithm.

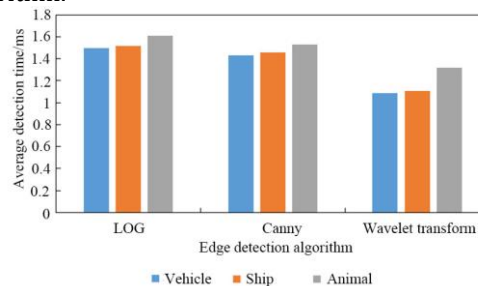


Figure 6: Comparison of detection time between three edge detection algorithms

As shown in Figure 6, the average time of the edge detection algorithm based on Log Operator for vehicle, ship and animal group images was 1.5 ms, 1.52 ms and 1.61 ms respectively; the average time of the edge detection algorithm based on Canny operator for vehicle, ship and animal group images was 1.43 ms, 1.46 ms and 1.53 ms respectively; the edge detection algorithm based on wavelet transform for vehicle, ship and animal group images was 1.09 ms, 1.11 ms and 1.32 ms, respectively. It was seen from Figure 6 that for vehicles, boats and animals, the average detection time of the edge detection algorithm based on wavelet transform was the shortest, that based on Canny operator was the second, and that based on LOG operator was the longest, but it was close to that based on Canny operator. The comparison of the average detection time of three groups of images under the same edge detection algorithm demonstrated that the detection time of vehicles and ships was short, and the detection time of animals was long. The reason for the difference was similar to the former one: vehicles and ships with regular appearance had many linear edges in the image, and moreover the total number of edges was small, while there were many non-linear edges in the animal group images, and moreover the thick hair made the unit area of the image produce more edges.

4. Conclusion

In this paper, three edge detection algorithms based on LOG operator, Canny operator and wavelet transform were introduced briefly, and then three edge detection algorithms were simulated and analyzed by MATLAB. The subjects of analysis were three kinds of group pictures of vehicles, ships and animals, three images each group. The results are as follows: (1) from the visual perspective, the image obtained by the algorithm based on wavelet transform had the best edge effect and rich details, and the algorithm based on the LOG operator had the poorest effect, but the rough shape was available; (2) the average PSNR of the algorithm based on wavelet transform was the largest, the average PSNR of the algorithm based on Canny operator was the second, and the average PSNR of the algorithm based on LOG operator was the smallest; (3) the average detection time of the algorithm based on LOG operator was the longest, the Canny operator based algorithm was the second, and the wavelet transform based algorithm was the shortest.

References

[1] Farbod M, Akbarizadeh G, Kosarian A, Rangzan K. "Optimized fuzzy cellular automata for synthetic aperture radar image edge detection," *Journal of Electronic Imaging*, 2018, 27(1):1.

[2] Singla D, Juneja M. "Hybrid Edge Detection-Based Image Steganography Technique for Color Images," *Advances in Intelligent Systems and Computing*, 2015, 309:277-280.

[3] Cai L C, Xu B, Su H T, Lyu M Z, Guo X F. "Edge Detection Analysis of High-Temperature Composite Based on Finite-Element Mesh of Binary Image," *Journal of Computational and Theoretical Nanoscience*, 2016, 13(12):9460-9463.

[4] Gu J, Pan Y, Wang H. "Research on the improvement of image edge detection algorithm based on Artificial Neural Network," *Optik - International Journal for Light and Electron Optics*, 2015, 126(21):2974-2978.

[5] Yu J D, Zhang X M. "Edge detection algorithm for lines on microscopic image," *Optics and Precision Engineering*, 2015, 23(1):271-281.

[6] Shehu A, Hulaj A, Bajrami X. "An Algorithm for Edge Detection of the Image for Application in WSN," *Journal of Data Acquisition & Processing*, 2017:207-213.

[7] Qiao N S. "Edge Detection of Printed Circuit Board Defect Image," *Acta Photonica Sinica*, 2016, 45(4).

[8] He Y B, Zeng Y J, Chen H X. "Research on improved edge extraction algorithm of rectangular piece," *International Journal of Modern Physics C*, 2018, 29(1).

[9] Ahmed Asif Fuad K, Rizvi S M. "Hardware Software Co-Simulation of Canny Edge Detection Algorithm," *International Journal of Computer Applications*, 2015, 122(19):7-12.

[10] Xu D Q, Wang X Y, Sun G, Li H. "Towards a novel image denoising method with edge-preserving sparse representation based on laplacian of B-spline edge-detection," *Multimedia Tools & Applications*, 2015, 76(17):17839-17854.

[11] Pyka K. "Detection of orthoimage mosaicking seamlines by means of wavelet transform," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016, XLI-B4:45-50.

[12] Wang Z, Wu L, Yang W. "Application of the directional wavelet transform in edge detection of airfield runway," 2016, 164:57-64.

[13] Yang Y, Kou K I, Zou C M. "Edge Detection Methods Based on Differential Phase Congruency of Monogenic Image," *Multidimensional Systems & Signal Processing*, 2016, 29(1):1-21.

[14] Wang Q, Zeng Q M, Zhang H Z, Jiao J. "Edge Detection of PolSAR Image Based on Stochastic Distance," *Acta Geodaetica Et Cartographica Sinica*, 2015, 44(7).

[15] Amoako-Yirenkyi P, Appati J K, Dontwi I K. "Performance Analysis of Image Smoothing Techniques on a New Fractional Convolution Mask for Image Edge Detection," *Open Journal of Applied Sciences*, 2016, 06(7):478-488.