

Research Article

An Improved NMS-Based Adaptive Edge Detection Method and Its FPGA Implementation

Enzeng Dong, Yao Zhao, Xiao Yu, Junchao Zhu, and Chao Chen

Key Laboratory of Complex System Control Theory and Application, Tianjin University of Technology, Tianjin 300384, China

Correspondence should be addressed to Chao Chen; cccovb@hotmail.com

Received 6 August 2015; Accepted 29 November 2015

Academic Editor: Vincenzo Paciello

Copyright © 2016 Enzeng Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For improving the processing speed and accuracy of edge detection, an adaptive edge detection method based on improved NMS (nonmaximum suppression) was proposed in this paper. In the method, the gradient image was computed by four directional Sobel operators. Then, the gradient image was processed by using NMS method. By defining a power map function, the elements values of gradient image histogram were mapped into a wider value range. By calculating the maximal between-class variance according to the mapped histogram, the corresponding threshold was obtained as adaptive threshold value in edge detection. Finally, to be convenient for engineering application, the proposed method was realized in FPGA (Field Programmable Gate Array). The experiment results demonstrated that the proposed method was effective in edge detection and suitable for real-time application.

1. Introduction

Due to the broad application fields, such as in industry, spaceflight, medicine, and military [1–3], the study on edge detection became a hot issue in image processing. At present, the researches on edge detection are mainly focused on mathematical morphology methods [4, 5] and gradient methods [6, 7]. The morphology based edge detection methods can effectively detect edges; however, complex iterative operations in these methods are the main obstacle for real-time applications.

On the other hand, the gradient based edge detection methods are more suitable for engineering implementation due to relatively simple calculation. Many improved edge detection algorithms based on gradient calculation such as Roberts operator [8], Sobel operator [9], Canny operator [10], and Laplacian operator [11] were proposed successively. For example, based on the Canny operator, Yu et al. [12] designed an edge directional interpolation method to process the MRI (magnetic resonance image) of fetal spinal column. Wang et al. [13] improved the Sobel edge detection algorithm by using NMS algorithm. Wiehle and Lehner [14] proposed an edge detection algorithm for high definition satellite images based on Sobel operator.

Owing to the advantages in high calculation speed and parallel processing, the FPGA became a useful hardware realization tool for real-time images processing [15–18]. Li et al. [19] designed a data block accelerator to optimize the Sobel operator in FPGA, and this method could be applied in real-time edge detection.

In previous works, the accuracy and speed of edge detection methods were improved from different aspects. Nevertheless, edge detection threshold has not been given much concern. In this paper, an adaptive edge detection algorithm based on the NMS was proposed. In this method, the gradient image was processed by utilizing NMS method. Then, the elements values of gradient image histogram were mapped into a wider value range by a certain power map, which can help calculate the threshold accurately.

According to the maximal between-class variance, the threshold was calculated and inverse mapped to original gradient map. The inverse mapped threshold was considered as adaptive threshold value in edge detection. Finally, for application in engineering fields, the proposed method was realized in FPGA. The experiment results showed that the proposed method has higher processing speed and detection accuracy.

TABLE 1: 3×3 Sobel operator.

(a) 0° template		
-1	0	1
-2	0	2
-1	0	1
(b) 45° template		
0	1	2
-1	0	1
-2	-1	0
(c) 90° template		
-1	-2	-1
0	0	0
1	2	1
(d) 135° template		
-2	-1	0
-1	0	1
0	1	2
(e) The original grey value		
Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

2. The Adaptive Edge Detection Algorithm

The flow chart of edge detection algorithm was shown in Figure 1. Gradient image was computed from original image data by Sobel operator. Then, NMS method was employed to process gradient image. Because gradient values in processed gradient image clustered in low value range, histogram mapping was performed to obtain a more spread distribution. After histogram mapping, the threshold \tilde{T} in mapped histogram was calculated by using OTSU method. By inverse mapping, adaptive threshold T was achieved. Double thresholds computed from threshold T were applied to determine potential edges.

2.1. Gradient Calculation and NMS Processing. The original image was divided into 5×5 subregion. The 3×3 Sobel operator of vertical, horizontal, and two diagonals (0° , 45° , 90° , and 135°) was employed to calculate the gradient, as shown in Tables 1(a), 1(b), 1(c), and 1(d).

The gradient of a 3×3 image region (Table 1(e)) can be computed by multiplying the element value with the corresponding value in four directional Sobel operators. Then, four gradient values with direction were calculated as in the following equation:

$$g_0 = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7),$$

$$g_{45} = (Z_2 + 2Z_3 + Z_6) - (Z_4 + 2Z_7 + Z_8),$$

TABLE 2: 3×3 gradient amplitude template.

G_1	G_2	G_3
G_4	G_5	G_6
G_7	G_8	G_9

$$g_{90} = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3),$$

$$g_{135} = (Z_8 + 2Z_9 + Z_6) - (Z_4 + 2Z_1 + Z_2), \quad (1)$$

where g_0 , g_{45} , g_{90} , and g_{135} are the gradient amplitude with 0° , 45° , 90° , and 135° , respectively. The maximal value and its direction were selected as gradient amplitude as shown in (2) and gradient direction, respectively, in the following process:

$$G_i = \max(g_0, g_{45}, g_{90}, g_{135}). \quad (2)$$

For 5×5 subregion, nine gradient amplitudes and the directions according to each gradient amplitude can be computed, as shown in Table 2.

The nonmaximum suppression method set the gradient values G_5 of central point to 0 if the gradient amplitude G_5 of central point is not local maximal, which indicates the location with the sharpest change of intensity value. According to the direction, the final gradient of the central point can be calculated by (3) as follows:

$$G_{5f} = \begin{cases} G_5, & (G_4 < G_5), (G_6 < G_5), (g_d = 0^\circ), \\ G_5, & (G_3 < G_5), (G_7 < G_5), (g_d = 45^\circ), \\ G_5, & (G_2 < G_5), (G_8 < G_5), (g_d = 90^\circ), \\ G_5, & (G_1 < G_5), (G_9 < G_5), (g_d = 135^\circ), \\ 0, & \text{others,} \end{cases} \quad (3)$$

where g_d is the direction of central point and G_{5f} is the gradient amplitude of the central point after NMS processing.

By traversing the whole image, the NMS processed gradient image was achieved. The larger value of gradient amplitude means more possibility of edge. To improve the accuracy of edge detection, threshold of gradient amplitude should be calculated to determine whether one point is edge point or not.

2.2. The Adaptive Strategy of Threshold Calculation. After NMS processing, gradient amplitude mainly concentrated in the smaller values ranges. The accuracy of edge detection is very sensitive to threshold. Thus, the calculation of threshold is a key step in the proposed method. In this subsection, calculation of adaptive threshold consists of three parts: histogram mapping, calculation of threshold according to maximal between-class variance, and inverse mapping of threshold as shown in Figure 2.

In the step of histogram mapping, the gradient amplitudes were mapped into more widely values ranges. Histogram mapping function was defined as follows:

$$j = \text{round}(16 \cdot \sqrt{i}), \quad (4)$$

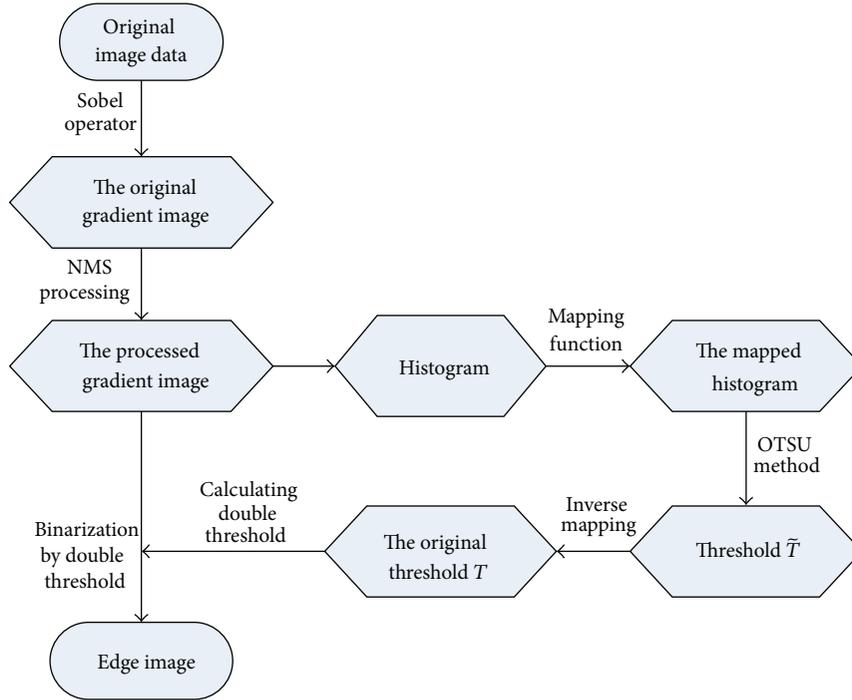


FIGURE 1: Flow chart of edge detection algorithm.

where i is the value in original histogram and j is the value in mapped histogram. Function round rounds decimal to the nearest integer.

In OTSU method [13], the optimum threshold separating the two classes was calculated by maximizing the between-class variance. Assuming that the NMS processed image contains two classes of point (edge point or not), the between-class variance can be computed by the following equation:

$$\sigma_B^2 = \frac{N_1}{N} (u_1 - u)^2 + \frac{N - N_1}{N} (u_2 - u)^2, \quad (5)$$

where σ_B^2 is the between-class variance, N is the total number of pixels of NMS processed gradient image, u is the average gradient amplitude, N_1 is the number of edge points, and u_1 and u_2 are the average values of edge points and background points, respectively, in the mapped gradient image.

Equation (5) can be simplified into (6)

$$\sigma_B^2 = \frac{N_1}{N - N_1} (u_1 - u)^2. \quad (6)$$

Furthermore,

$$N_1 = \sum_{i=1}^t n_i, \quad (7)$$

where n_i is the number of points whose gradient value is i .

The average gradient value u can be calculated as (8)

$$u = \frac{\sum_{i=1}^{255} i \cdot n_i}{\sum_{i=1}^{255} n_i}. \quad (8)$$

u_1 and u_2 can be calculated as follows:

$$u_1 = \frac{\sum_{i=1}^t i \cdot n_i}{\sum_{i=1}^t n_i} = \frac{u(t)}{N_1}, \quad (9)$$

$$u_2 = \frac{\sum_{i=t+1}^L i \cdot n_i}{\sum_{i=t+1}^L n_i} = \frac{u \cdot N - u(t)}{N - N_1},$$

where $u(t)$ is equal to $\sum_{i=1}^t i \cdot n_i$.

By traversing t from 1 to 254, the value t corresponding to the maximum between-class variance σ_B^2 is the threshold \tilde{T} . Finally, the adaptive threshold T in NMS processed image can be obtained by using inverse mapping as follows:

$$T = \left(\frac{\tilde{T}}{16} \right)^2. \quad (10)$$

2.3. The Processing of Double Threshold. The double thresholds were designed to determine the true or false edge points in NMS processed image. The threshold value T was defined as the mean value between the high threshold and the low threshold. Then, the high threshold and the low threshold can be calculated as the following equations:

$$T_h = \frac{T}{0.7} \approx 1.43T, \quad (11)$$

$$T_l = \frac{0.4 \cdot T}{0.7} \approx 0.57T,$$

where T_h is the high threshold and T_l is the low threshold.

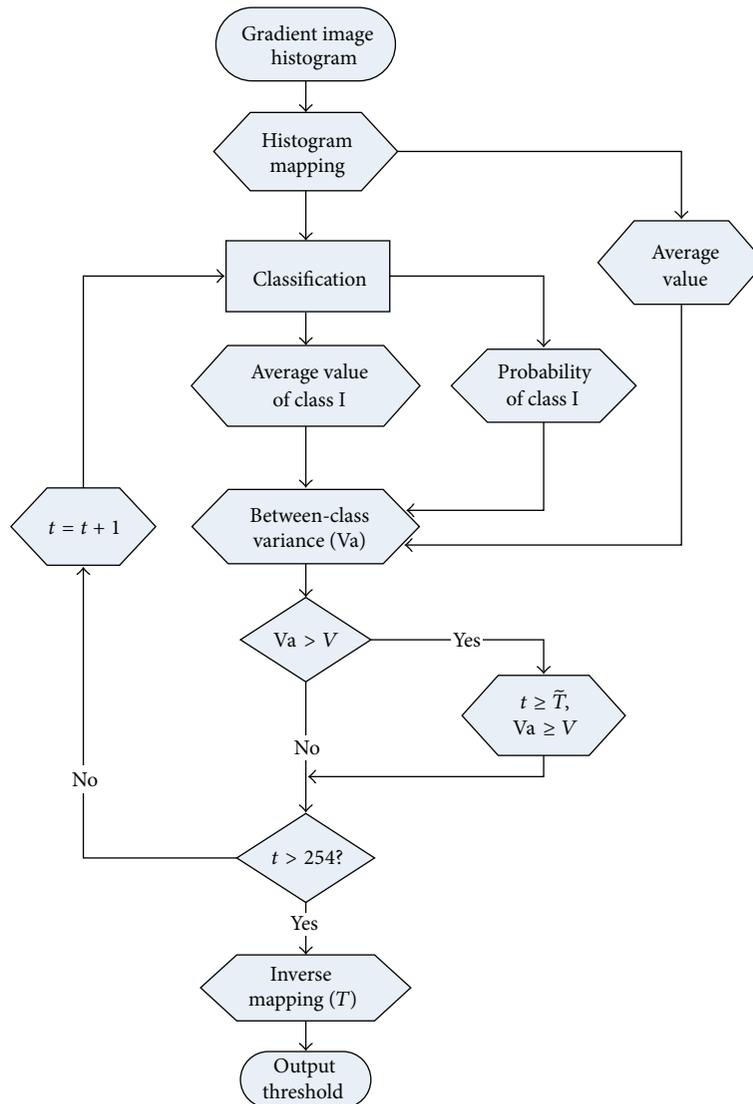


FIGURE 2: Flow chart of adaptive threshold calculation.

In edge detection process, if the gradient value is greater than T_h , it was edge point. If the gradient value is smaller than T_l , it was not considered as edge point. In addition, if the gradient value is between high threshold and low threshold, at least one gradient value of neighborhood points (Table 2) is greater than the high threshold; this point was edge point; otherwise, it was not the edge point.

3. FPGA Implementation

The FPGA implementation scheme was mainly divided into three modules: image storage module, VGA display module, and algorithm executive module. The whole structure diagram was shown in Figure 3.

3.1. Image Storage Module. Image storage module consists of two on-chip ROMs. One was used to store address of pixels, and the other one was used to store the grayscale of

the corresponding pixel. This design can reduce image storage space as shown in Figure 4.

3.2. VGA Display Module. VGA display module was designed to realize two main functions. Firstly, it can generate clock signal, blanking signal, line sync signal, and field sync signal for the display [20]. Secondly, it can produce the correct address signal for reading the image data stored in ROM. The module circuit was shown in Figure 5.

3.3. Algorithm Executive Module. Six submodules were designed to realize the edge detection algorithm, including gradient amplitude calculation submodule, gradient direction calculation submodule, NMS processing submodule, histogram mapping submodule, maximal between-class variance calculation submodule, and double threshold processing submodule.

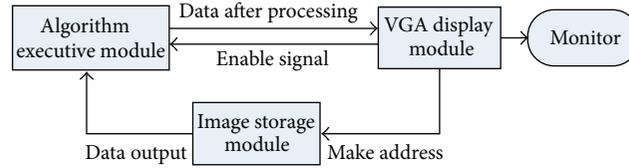


FIGURE 3: Block diagram of the system.

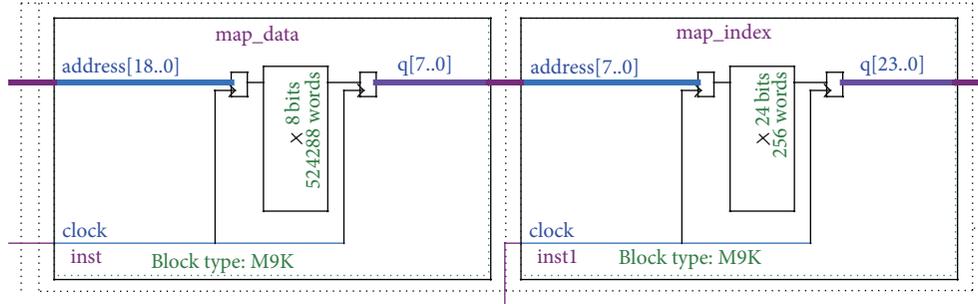


FIGURE 4: Memory module.

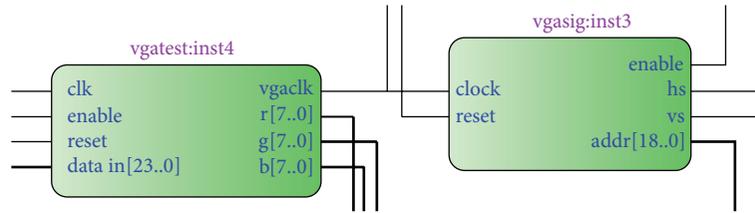


FIGURE 5: VGA display module.

Gradient amplitude calculation submodule and gradient direction calculation submodule were used to calculate the eight neighborhood gradient amplitudes and gradient direction of center point, respectively. NMS processing module was designed to process gradient image. These three submodules were shown in Figure 6.

Histogram mapping submodule was used to map the histogram of the gradient image. Maximal between-class variance calculation submodule was designed to find the threshold corresponding to maximal between-class variance. Double threshold processing submodule was used to inverse map the threshold and calculate double threshold. Threshold calculation related three submodules and connection were shown in Figure 7.

4. The Experiment Results and Analysis

In this section, the experimental results were analyzed in three aspects: resource usage rate, running duration, and accuracy of edge detection.

4.1. Resource Usage Rate. The DE2-115 education development board (Altera Corporation) with EP4CE115F29C7N chip was used in this experiment. The size of image was

TABLE 3: Resource usage of edge detection system.

	Logical elements	Registers	Multiplier
Gradient calculation	3800	2032	0
Adaptive threshold calculation	9263	3999	7
The whole system	12461	5654	7

640×480 pixels. The system resource usage was shown in Table 3. Note that the resource usage of the whole system is less than the sum of the two subsystems, because of the repetition usage of the logical elements in subsystems. The logical elements used were 11% of the whole elements in running time, which suggested that the resource usage rate of proposed method is relatively low.

4.2. Running Duration of Each Process. In previous works, adaptive edge detection algorithm [21] and a simple Canny algorithm [22] were realized by FPGA. Comparison of running duration with those works was shown in Table 4. Although the working frequency of proposed method was relatively lower, the running duration of each process was

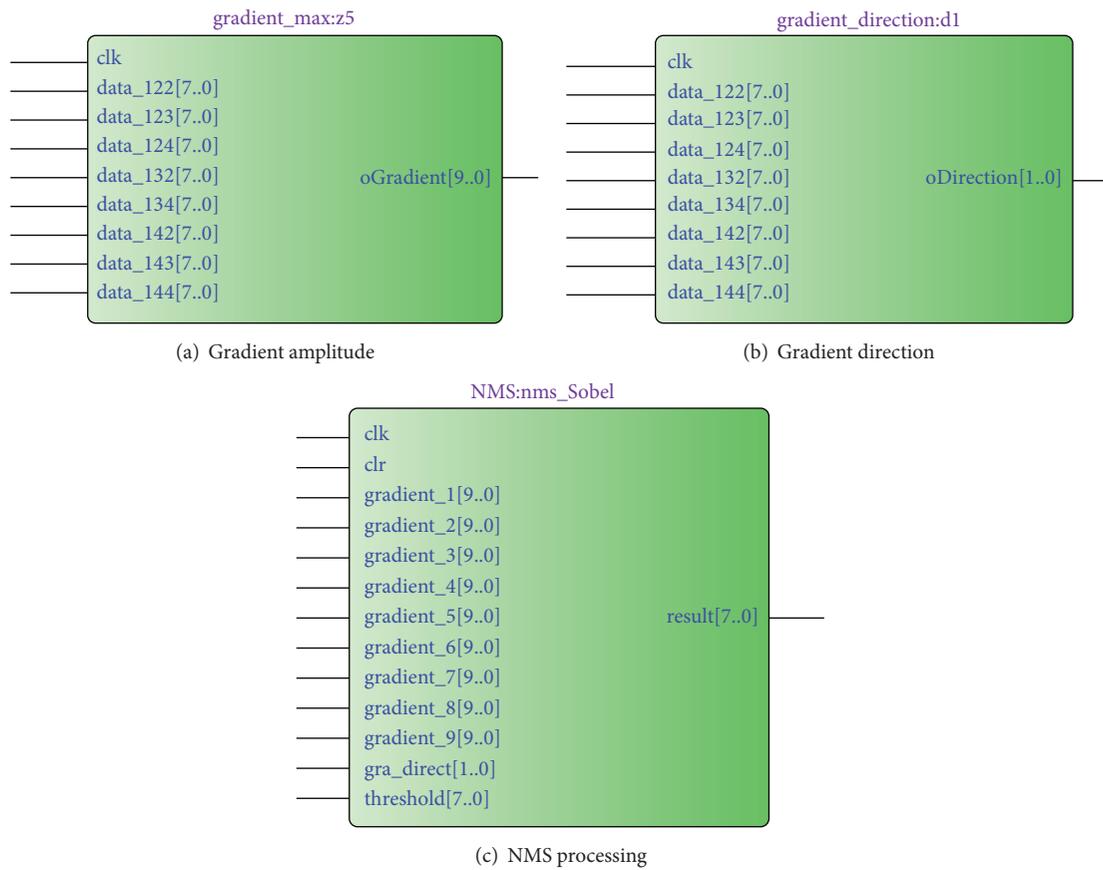


FIGURE 6: Gradient calculation related modules.

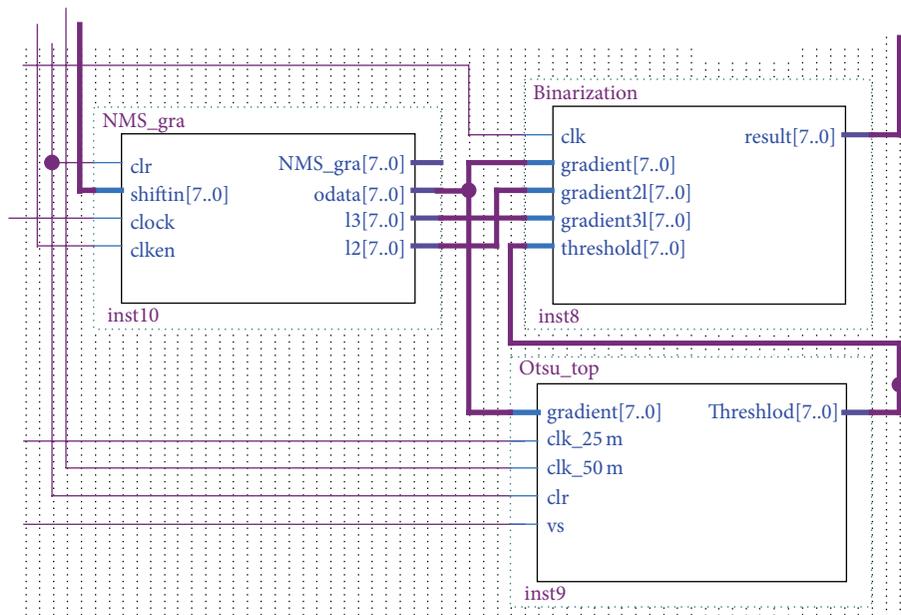


FIGURE 7: Adaptive threshold calculation related modules.

TABLE 4: System running duration.

Algorithm	Platform	Image size	Uptime
Adaptive threshold algorithm in [21]	Cyclone II, working frequency: 50 MHz	$512 \times 512 \times 8$ bits	0.0313 s
Simple Canny algorithm in [22]	Virtex-5, working frequency: 100 MHz	$512 \times 512 \times 8$ bits	0.0287 s
The proposed algorithm based on FPGA	Cyclone IV, working frequency: 50 MHz	$640 \times 480 \times 8$ bits	0.0245 s

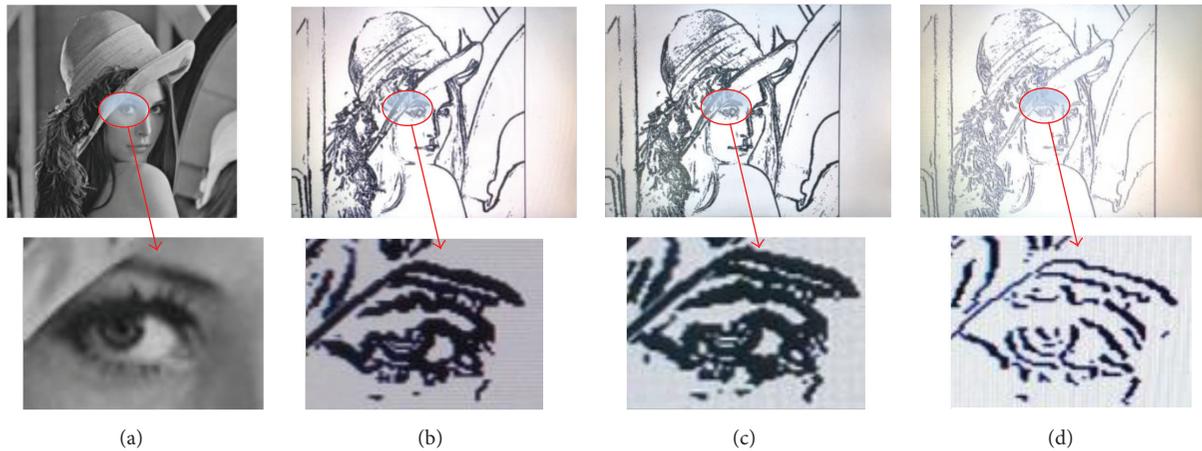


FIGURE 8: The edge detection results with image Lena, (a) original image, (b) method in [23], (c) Prewitt method, and (d) proposed method.

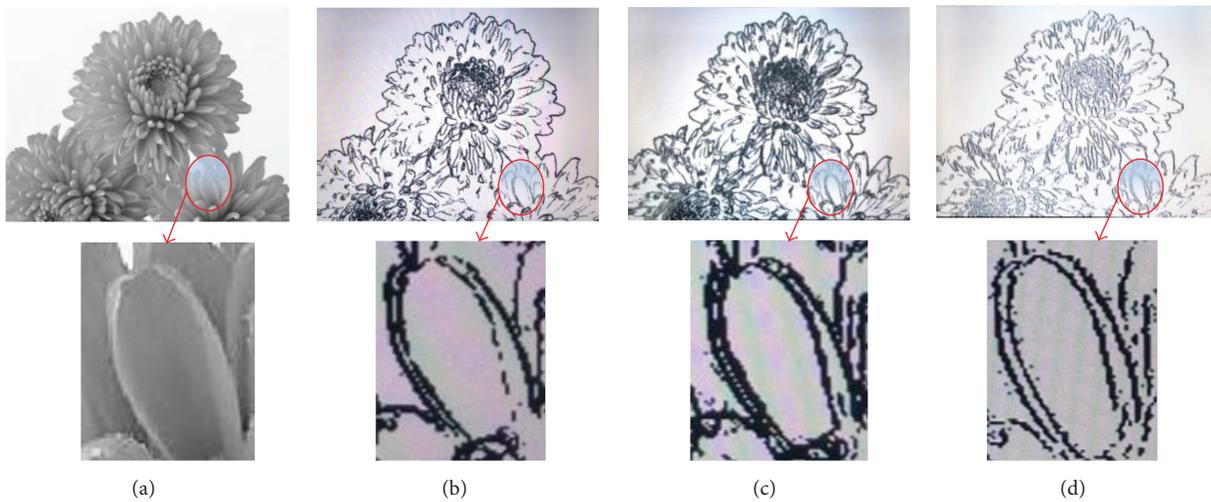


FIGURE 9: The edge detection results with image Flower, (a) original image, (b) method in [23], (c) Prewitt method, and (d) proposed method.

shorter, which demonstrated the higher speed of proposed algorithm.

4.3. Accuracy of Edge Detection. Image Lena (Figure 8(a)) and image Flower (Figure 9(a)) were used in edge detection experiments. The detection results of the two images were shown in Figures 8(b), 8(c), and 8(d) and Figures 9(b), 9(c), and 9(d), respectively. Compared with the other two

methods, the detected edge by proposed method is thinner than the other methods.

The positive rates of edge detection were calculated to evaluate the effect of the proposed method, as shown in Table 5. Compared with Roberts operator, Prewitt operator, Sobel operator, LoG operator, and Canny operator, true positive rate by proposed method is the highest and false positive rate is relatively lower. These results suggested that the proposed method can detect the edge more accurately.

TABLE 5: Comparison of the positive rate.

	True positive rate (%)	False positive rate (%)
Roberts operator	31.26	1.02
Prewitt operator	34.08	0.58
Sobel operator	32.73	0.53
LoG operator	26.83	2.39
Canny operator	45.17	5.2
Proposed method	48.16	1.19

5. Conclusion

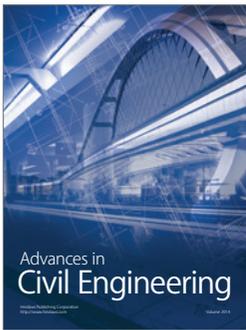
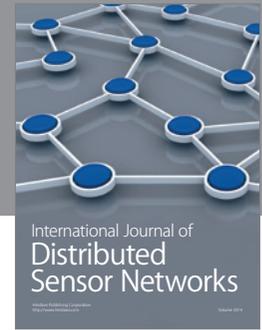
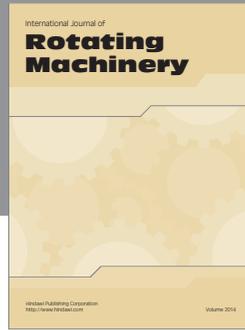
An improved adaptive edge detection algorithm based on the NMS method was proposed in this paper. In the proposed method, a power map function was defined to map the NMS processed gradient image. Then, adaptive threshold corresponding to maximal between-class variance was calculated based on the mapped histogram. Additionally, the proposed method was realized in FPGA. Experimental results showed that this method has higher processing speed and better accuracy, which was suitable for application in real-time image processing.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] F. Ren, B. Li, and Q. Chen, "Single parameter logarithmic image processing for edge detection," *IEICE Transactions on Information & Systems*, vol. E96, no. 11, pp. 2437–2449, 2013.
- [2] P. R. Possa, S. A. Mahmoudi, N. Harb, C. Valderrama, and P. Manneback, "A multi-resolution FPGA-based architecture for real-time edge and corner detection," *IEEE Transactions on Computers*, vol. 63, no. 10, pp. 2376–2388, 2014.
- [3] C. Lopez-Molina, B. D. Baets, H. Bustince, J. Sanz, and E. Barrenechea, "Multiscale edge detection based on Gaussian smoothing and edge tracking," *Knowledge-Based Systems*, vol. 44, pp. 101–111, 2013.
- [4] W. Shi, X. Zhu, and G. Zhu, "Adaptive edge detection algorithm of MRI image based on morphology," *Chinese Journal of Scientific Instrument*, vol. 34, no. 2, pp. 408–414, 2013.
- [5] Y. Wu, Y. Song, and H. Zhou, "Edge detection of combustion flame images based on anisotropic mathematical morphology," *Chinese Journal of Scientific Instrument*, vol. 34, no. 8, pp. 1818–1825, 2013.
- [6] Y. Liu and X. Cheng, "A new edge detection algorithm using wavelet and threshold method," *Journal of Information & Computational Science*, vol. 6, no. 4, pp. 1759–1766, 2009.
- [7] E. Chen, J. Wang, L. Qi, and W. Lv, "A novel multiscale edge detection approach based on nonsubsampling contourlet transform and edge tracking," *Mathematical Problems in Engineering*, vol. 2015, Article ID 504725, 14 pages, 2015.
- [8] L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-Optical Information Processing*, vol. 20, pp. 31–39, MIT Press, 1963.
- [9] I. Sobel, *Camera Models and Machine Perception*, Department of Computer Science, Stanford University, Stanford, Calif, USA, 1970.
- [10] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [11] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London*, vol. 207, no. 1167, pp. 187–217, 1980.
- [12] S. Yu, R. Zhang, S. Wu, J. Hu, and Y. Xie, "An edge-directed interpolation method for fetal spine MR images," *BioMedical Engineering Online*, vol. 12, no. 1, article 102, 2013.
- [13] W. Wang, S. Zhu, and W. Tang, "Adaptive non-maxima suppression edge detection algorithm and FPGA design to achieve," *Chinese Journal of Scientific Instrument*, vol. 35, no. 8, pp. 1926–1930, 2014.
- [14] S. Wiehle and S. Lehner, "Automated waterline detection in the wadden sea using high-resolution terrasar-X Images," *Journal of Sensors*, vol. 2015, Article ID 450857, 6 pages, 2015.
- [15] P.-Y. Hsiao, H. Wen, C.-H. Chen, and S.-J. Chen, "Real-time realisation of noise-immune gradient-based edge detector," *IEE Proceedings: Computers and Digital Techniques*, vol. 153, no. 4, pp. 261–269, 2006.
- [16] Z. Hu, D. Wang, Q. Zhao, and Q. Zhu, "Design of portable intelligent image acquisition system based on SOPC," *Chinese Journal of Scientific Instrument*, vol. 31, no. 2, pp. 371–376, 2010.
- [17] Y. Guo, Y. Su, and C. Gao, "Design and implementation of real time infrared image collection system based on FPGA," *Chinese Journal of Scientific Instrument*, vol. 32, no. 3, pp. 514–519, 2011.
- [18] L. Zhang and C. Ling, "Application of sobel edge detection based on FPGA," *Embedded Technology*, vol. 37, no. 8, pp. 26–29, 2011.
- [19] Z. Li, C. Liu, and M. An, "The design of sobel image edge detection based on FPGA," *Modern Scientific Instruments*, vol. 2012, no. 4, pp. 84–86, 2012.
- [20] J. Jiang, D. Yu, and J. Wang, "The CCD color image acquisition and displaying system of electronic endoscope based on FPGA," *Chinese Journal of Scientific Instrument*, vol. 21, no. 1, pp. 50–53, 2000.
- [21] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A distributed canny edge detector: algorithm and FPGA implementation," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2944–2960, 2014.
- [22] G. Chaple and R. D. Daruwala, "Design of Sobel operator based image edge detection algorithm on FPGA," in *Proceedings of the 3rd International Conference on Communication and Signal Processing (ICCSP '14)*, pp. 788–792, IEEE, April 2014.
- [23] L. Huang, Y. Fang, X. Zuo, and X. Yu, "Automatic change detection method of multitemporal remote sensing images based on 2D-Otsu algorithm improved by firefly algorithm," *Journal of Sensors*, vol. 2015, Article ID 327123, 8 pages, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

