Image Denoising and Image Contour Detection

Yi Yang [1], Chengzhen Jia [2,*]

[1] Southampton Ocean Engineering Joint Institute at Harbin Engineering University, Harbin, China

[2] School of Electrical Engineering and Artificial Intelligence, Xiamen University Malaysia, Sepang, Malaysia

* Corresponding Author Email: eee2209290@xmu.edu.my

Abstract. With the development of various fields in society, the demand for digital images has increased significantly. These digital images require image processing in many fields before they can be put into use, such as transportation, healthcare and remote sensing. After denoising and contour detection, digital images can play their role. So far, many image denoising and image contour detection methods have been proposed by researchers, each with different characteristics. In the aspect of image denoising, this paper reviews some image-denoising methods and FPGA implementations based on denoising filters provides the principles of these methods and discusses the advantages and disadvantages of these denoising methods and the advantages of FPGA implementations based on filters. In the aspect of image contour detection, traditional methods have prominent limitations when performing edge detection on complex scenes. This paper systematically reviews the improvement techniques of Canny and Sobel's classic edge detection methods, as well as the optimization of Gaussian filtering for detection performance, and explores their practicality in complex scenes.

Keywords: Image denoising, spatial domain methods, transform domain methods, image contour detection, Canny, Sobel detection, gaussian filtering, FPGA.

1. Introduction

With the continuous development of society, the output of digital images is increasing, and in many professional fields, digital images need to undergo image processing before they can be put into use. For example, in the field of automatic driving, the auto drive system needs to accurately identify pedestrians and other various objects in a complex environment, and these images are usually interfered with by noise and cannot be used directly. To accurately identify and classify various objects, first denoise the images to ensure image clarity, and then conduct contour detection. In the field of medical imaging, medical images are often affected by noise, which can mask or distort the boundaries of the lesion area. Therefore, image denoising is needed to improve image quality, and contour detection is then performed to accurately identify the lesion area, thereby helping doctors make more accurate diagnoses [1]. Image denoising and image contour detection play an important role in image processing.

Image denoising is a fundamental step in image processing, which aims to remove noise from the image while preserving its true details as much as possible, thereby improving the quality of the image [2]. This helps to enhance the accuracy of subsequent image contour detection. Image contour detection is an important component of image analysis, aimed at identifying and extracting edge and structural information in images, thereby providing high-quality feature information for subsequent image processing and analysis tasks [3]. Image denoising and image contour detection are interrelated, jointly improving the usability and analysis effectiveness of images.

In the aspect of image denoising, this paper summarizes some classical denoising methods in spatial domain and transform domain, and analyses their principles, advantages and disadvantages. Subsequently, two FPGA-based filtering methods were summarized, which can greatly improve denoising speed while completing denoising tasks, they have the characteristics of parallelism, real-time performance, low power consumption, and flexibility.

In the aspect of image contour detection, the conventional Canny algorithm has a poor detection effect and is prone to creating false edges in complicated scenarios (such as photos with rich textures or uneven illumination). By using adaptive thresholds, the enhanced approach put forth by Weibin Rong et al. decreases edge loss and increases detection robustness [4]. In the same way, only simple edge situations can use the conventional Sobel operator. Zhiyong Zhang et al. suggested a multi-directional Sobel method to enhance edge-capturing capabilities to overcome this limitation [5]. Furthermore, a Gaussian filtering technique based on changing standard deviation is presented to reduce the influence of Gaussian noise [6]. In addition to analysing the idea of picture edge contour detection and its use in complicated scenarios, this study provides an overview of the Canny algorithm, Sobel algorithm, Gaussian filtering, and other relevant enhancement technologies. It attempts to increase the accuracy and real-time performance of contour detection in complicated scenarios by putting these enhanced algorithms on FPGA [7, 8].

2. Image Denoising

Digital images usually produce noise in the process of acquisition or transportation, noise refers to the introduction of irrelevant signals or random changes in the image. These noises will interfere with the real information of the image and reduce the quality of the image, thus affecting the visual effect of the image and subsequent image processing tasks [9].

Generally speaking, image-denoising methods can be divided into spatial domain methods and transform domain methods according to the working domain.

2.1. Spatial Domain Methods

Spatial domain methods are a kind of method that directly processes the pixel values of images, they reduce noise by analysing and adjusting the pixels of images [10]. Spatial filters can be divided into linear and nonlinear filters.

Linear filters are commonly used to remove Gaussian noise by processing pixel values in images through linear operations. Specifically, it combines the weighted sum of each pixel value of the image with its neighbouring pixel values, but it often cannot effectively preserve the texture details of the image when removing noise. Common linear filters include the mean filter and the Wiener filter.

Mean filtering smooths the image by calculating the average value of the pixel neighbourhood. However, mean filtering may excessively smooth the image under high noise conditions, resulting in loss of details [11]. Wiener filtering can overcome this drawback. Wiener filtering achieves denoising by minimizing the mean square error between the restored image and the original image. It can remove noise while preserving the edges and details of the image, but it may still blur the sharp edge of the image [12].

Nonlinear filters can be used to remove noise from images without recognizing it. They do not rely on linear operations but rather determine the output value of pixels through more complex operations, such as sorting, minimum value, maximum value, etc. [13]. Common nonlinear filters include median filters, weighted median filters, and bilateral filters.

The working principle of the median filter is to sort all pixel values within the pixel neighbourhood, replacing the central pixel value with the median. It can effectively suppress noise and preserve the edges of the image well [14]. The weighted median filter is an extension of median filtering, which considers the weights of pixels in different adjacent regions, making the filtering results more accurate [15].

The basic idea of bilateral filter mainly includes two parts: domain filtering and range filtering [16]. The domain filter acts as a low-pass filter, which can smooth the image and remove noise. The range filter is the nonlinear part of the bilateral filter, which has the function of preserving edges. It averages pixels by comparing the similarity of pixel values, only similar pixel values will be averaged, and pixel values with significant differences will not affect the filtering process.

The shift-variant transformation operation of bilateral filtering can be represented under Gaussian noise as:

$$\bar{\phi}(\bar{\mathbf{m}}_0) = \frac{1}{k(\mathbf{m}_0)} \sum_{\mathbf{m} \in F} \phi(\mathbf{m}) \cdot s(\phi(\mathbf{m}_0), \phi(\mathbf{m})) \cdot c(\mathbf{m}_0, \mathbf{m}) \tag{1}$$

In the formula, $\mathbf{m}$=($\mathbf{m}$,$\mathbf{n}$) represents the pixel coordinates of the image to be filtered. $\mathbf{m}_0$ and $\bar{\mathbf{m}}_0$ represent the centre pixel coordinates in the noisy image and the filtered image, respectively. F is the filtering window. $\bar{\phi}(\bar{\mathbf{m}}_0)$ represents the grayscale value of the pixel being filtered, $\phi(\mathbf{m}_0)$ represents the grayscale value of the central pixel $\mathbf{m}_0$, $\phi(\mathbf{m})$ represents the grayscale value of the neighbouring pixel $\mathbf{m}$.

The operation of bilateral filters can be divided into two parts: the photometric component and the geometric component. The photometric component compares the difference in grayscale values between the central pixel and neighbouring pixels. Its function is to adjust the weight based on the difference in grayscale values. The larger the grayscale value difference, the smaller the corresponding filtering coefficient; The smaller the difference in grayscale values, the larger the filtering coefficient. The formula is:

$$s(\phi(\mathbf{m}_0), \phi(\mathbf{m})) = \exp\left(-\frac{1}{2}\left(\frac{\|\phi(\mathbf{m}_0) - \phi(\mathbf{m})\|}{\sigma_{\mathrm{ph}}}\right)^2\right) \tag{2}$$

In the formula, σ_{ph} is a parameter that controls the impact of grayscale differences.

The geometric component compares the spatial distance between the central pixel and neighbouring pixels. Its function is to adjust weights based on spatial distance, the greater the spatial distance, the smaller the weight, the closer the distance, the greater the weight. The formula is:

$$c(\mathbf{m}_0, \mathbf{m}) = \exp\left(-\frac{1}{2}\left(\frac{\|\mathbf{m}_0 - \mathbf{m}\|}{\sigma_{\mathrm{c}}}\right)^2\right) \tag{3}$$

In the formula, σ_{c} is a parameter that controls the influence of spatial distance.

Finally, the normalization operation is used to avoid the large change of image brightness, which can ensure the stability of the filtered image gray value range. The formula is:

$$k(\mathbf{m}_0) = \sum_{\mathbf{m} \in F} s(\phi(\mathbf{m}_0), \phi(\mathbf{m})) \cdot c(\mathbf{m}_0, \mathbf{m}) \tag{4}$$

2.2. Transform Domain Methods

The transformation domain method is different from the spatial domain method. The transformation domain method transforms the image from the original pixel space to another domain through mathematical transformation, thereby forming a new representation form, and then processing the noise within that domain. In the transform domain, due to the different characteristics of signals and noise, noise usually manifests as smaller coefficients, concentrated in the high-frequency part, while the energy of signals is usually concentrated in fewer coefficients (low-frequency part). Therefore, by processing the transform domain coefficients, the goal of removing noise can be achieved. Finally, the processed image is restored to the spatial domain through inverse transformation [17].

PCA method preserves the main information of the image through dimensionality reduction and feature extraction while removing noise. This is a data-adaptive method, but its computational complexity is high, it is suitable for images with a relatively single type of noise [18].

The wavelet transform method decomposes an image into multiple different frequency bands, each containing information about the image at different scales and directions. The low-frequency part generally includes the overall structure and contour of the image, while the high-frequency part includes the details, texture, and noise of the image. Denoising different frequency bands of an image can achieve the goal of removing noise while preserving the details of the image. This method also has good localization ability and can analyze images at different scales, but there are also issues with selecting wavelet bases and high computational complexity [19].

BM3D is an image-denoising algorithm based on block matching and 3D filtering, which is a classical algorithm of transform domain methods. Its basic idea is to stack similar image patches into three-dimensional data blocks through block matching and apply hard thresholding or Wiener filtering to the coefficients of these three-dimensional data blocks in the transform domain, thereby effectively preserving the details and edge information of the image while removing noise. But when the noise increases, the denoising effect of this method on the image is not good [20].

3. Image Edge Detection

3.1. Enhancements in Contour Detection

With the continuous development of science and technology, our requirements for the electronic products we use are constantly increasing, especially in the field of image contour detection, which has important applications in transportation, medical imaging and satellite remote sensing. The original various algorithms are gradually being eliminated under the existing needs of contour detection, and it is imperative to improve the original algorithms according to the needs.

3.1.1 Adaptive Canny Edge Detection

As a classic edge detection algorithm, the Canny algorithm works well for simple, low-noise images, but performs poorly for complex images, such as scenes with a lot of textures and shadows. An improved Canny edge detection algorithm proposed by Weibin Rong et al. in 2014 uses an adaptive threshold selection method that can automatically simulate different situations and applies to a wider range of needs [4].

For images with less edge information details, the gradient amplitude distribution is more concentrated, most pixels have low gradient amplitudes, and only a small number of pixels have large gradient values. To effectively distinguish edge pixels from non-edge pixels, double threshold selection is selected, and the high threshold T_h is determined by the formula $T_h = E_{ave} + k \cdot \sigma$, where$k$ is the coefficient obtained experimentally, generally in the range of 1.2 to 1.6. E_{ave} is the gradient mean and σ is the standard deviation. Pixels with gradient amplitudes greater than the high threshold T_h are marked as edge pixels.

Compared with images with more edge information, the gradient distribution in different regions is quite different. Using a global threshold may lead to the loss of some edge information. First, select a small $N*N$ area around each pixel and calculate the gradient mean E_{ij} and standard deviation σ_{ij} of the area. Calculate the high threshold of the local area according to the formula $T_h = E_{ave} + k \cdot \sigma_{ij}$, which can avoid the edge loss problem caused by over-reliance on the global threshold in traditional methods.

This improved algorithm not only retains the advantages of the traditional Canny algorithm but also greatly improves the ability to suppress noise. However, its calculation speed is relatively slow, and the calculation complexity is high, which is not conducive to hardware implementation, as shown in fig 1 and fig 2.

Figure 1. (a) Original Image (b) Traditional Canny Algorithm (c) Method for Images with Less Edge Information [4]

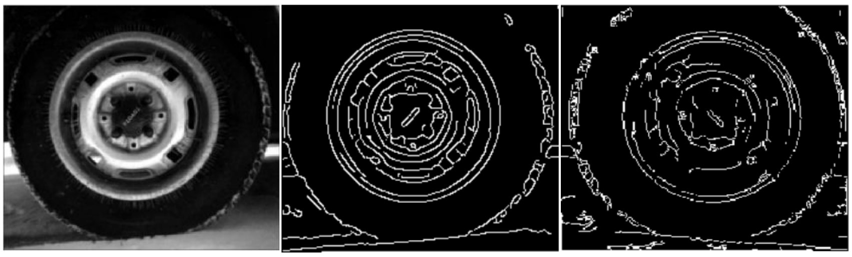

Figure 2. (a) Original Image (b) Traditional Canny Algorithm (c) Method for Images with More Edge Information [4]

3.1.2 FPGA-Based Improved Sobel Operator

As an edge operator based on the first-order derivative, the Sobel algorithm calculates the magnitude of the image gradient by neighbourhood averaging or weighted averaging to detect image edge details. This method is relatively simple to calculate fast, and easy to implement in hardware. However, since it only uses templates in two directions, it can only detect edges in the horizontal and vertical directions. For images with more complex textures, the edge detection effect is average [26].

In FPGA-Based Improved Sobel Operator Edge Detection proposed by Guangxiao Zhou et al. in 2023, it is mentioned that the convolution calculation dimension of the Sobel operator is expanded from two directions to eight directions, namely 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° [26]. At the same time, the parallel processing capability of FPGA is used to improve the detection efficiency to obtain more accurate edge information. The gradient amplitude G used is obtained by summing the squares of the convolution results in eight directions. This algorithm has higher accuracy, richer edge detail capture, and faster calculation speed, and has important use value in the field of image processing.

$$G_1 = G_{0^\circ}^2 + G_{45^\circ}^2 + G_{90^\circ}^2 + G_{135^\circ}^2 \tag{4}$$

$$G_2 = G_{180^\circ}^2 + G_{225^\circ}^2 + G_{270^\circ}^2 + G_{315^\circ}^2 \tag{5}$$

$$G = \sqrt{G_1 + G_2} \tag{6}$$

3.1.3 Multidirectional Sobel Operator

In the research of Zhiyong Zhang et al. in 2024, the traditional 3×3 Sobel operator was expanded to the 5×5 direction, and 45° and 135° operators were introduced to it so that it has higher performance in the central row and column. Weight can better capture image details and be more sensitive to edges.

Regarding the edge detection results, as can be seen from the figure below, the 3×3 image is relatively darker than the image detected by the 5×5 operator, indicating that the 3×3 Sobel has a weaker ability to calculate edge gradients. As shown in Fig3, the image in Figure (d) has no obvious shadows in the horizontal and vertical directions, while the images shown in Figures (b) and (c) all have significant shadows in both directions, indicating that the 5×5 Sobel calculation in four directions is better than 3×3 and the 5×5 operator has good detection ability, as shown in fig3.

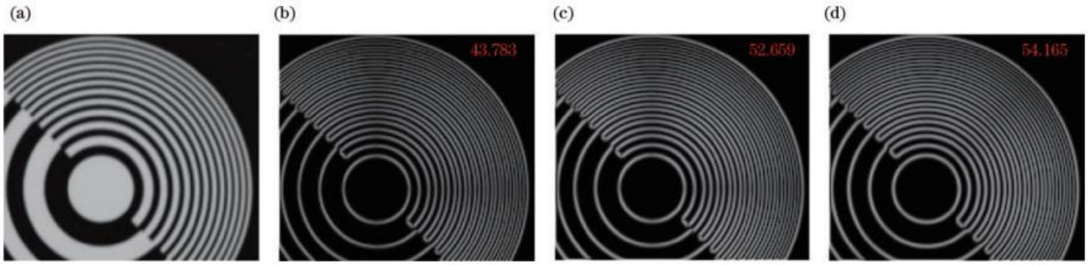

Figure 3. The edge detection results using Sobel operators of different sizes and directions. (a) Original Image; (b) 3×3 Sobel Detected Image; (c) 5×5 Sobel Detected Image; (d) 4-Direction 5×5 Sobel Detected Image [6]

3.2. Noise Reduction Techniques

During the image acquisition process, the image is often affected by various noises, which reduces the effect of edge detection. One of them is Gaussian noise. Gaussian filtering effectively reduces the interference of noise by smoothing the image, while retaining the edge information of the image.

3.2.1 Gaussian Filtering for Edge Preservation

Gaussian filter is a linear filter used for smoothing. The smoothing filter replaces the grey value of each pixel in the image with the average grey value of the pixels in the neighbourhood determined by the filter mask, thereby achieving a smoothing effect. To reduce the impact of image noise on clarity calculation, it is necessary to filter the image [27]. Applying Gaussian filtering to the region of interest (ROI) of the image and performing weighted averaging on the neighbourhood pixels, where the weight of the central pixel is higher than that of the surrounding pixels, can effectively overcome the boundary effect and remove Gaussian noise while maintaining the clarity of the edge information, which helps the Sobel operator better capture the edge information in the image [6].

3.2.2 ROI-based Gaussian Filtering with Variable Standard Deviation

As shown in Fig 4, in the research of Zhiyong Zhang et al. in 2024, a Gaussian filter method introducing variable standard deviation was proposed to determine the weight by calculating the distance between the central pixel of the image and the surrounding pixels, and the weight decreased as the distance increased, the weight of pixels in the central area is larger, and the weight of pixels in the edge area is smaller [6]. In addition, this method divides the ROI into different blocks, and calculates the weight distribution within the area based on the pixel value of each block, ultimately forming a weighted filter matrix. This method significantly improves the accuracy of edge detection, is especially suitable for processing images with complex textures and uneven brightness distribution and is particularly outstanding in edge information retention and noise suppression [6].

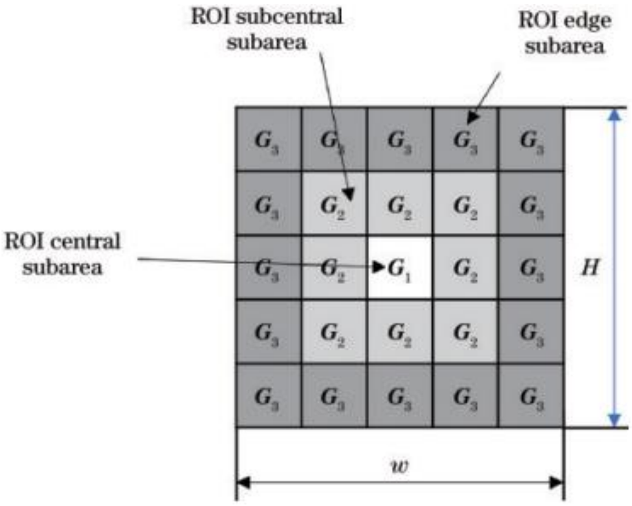

Figure 4. ROI, ROI central subarea, ROI subcentral subarea [6]

4. FPGA implementation of image processing

Image processing algorithms are widely used in practical applications, so improving their running speed is particularly important. There are currently two main ways to improve speed: optimizing image processing algorithms; Improve processing speed through hardware.

4.1. FPGA implementation of image denoising

4.1.1 An FPGA Design of a Bilateral Filter for Real-Time Image Denoising

Anna Gabiger Rose et al. proposed that real-time performance can only be achieved when hardware is used to accelerate the filter [21]. In hardware comparison, FPGA has lower power consumption and latency compared to GPA, so Anna Gabiger Rose et al. proposed an architecture for FPGA implementation based on a bilateral filter, which is characterized by real-time image denoising.

The architecture design of this method mainly consists of three parts: register matrix, photometric component, and geometric component [21]. Among them, the register matrix divides the image data into multiple groups for processing. This means the data in the whole filtering window can be passed to the next level of processing in a pixel clock cycle, and allows multiple filtering operations to be carried out at the same time to achieve parallel processing. The photometric component is a non-linear component, it calculates the difference in pixel grayscale values and generates corresponding weighting coefficients. The smaller the grayscale value difference, the larger the corresponding weighting coefficient. This design calculates all possible photometric weight values in advance based on formulas and possible grayscale differences and stores these weight values in LUT, which reduces the complexity of each weight calculation and improves the calculation speed. Geometric components utilize the separability and symmetry of filters to decompose two-dimensional filtering into one-dimensional filtering in both vertical and horizontal directions. This approach not only improves computational efficiency but also reduces the occupation of hardware resources. Due to the symmetry of the filtering window, geometric components can reduce computational complexity by accumulating pixel pairs while maintaining the accuracy of the filtering results.

In this architecture design, the internal clock frequency of the filter is set to four times the input pixel clock frequency, which can complete multiple computing tasks within a one-pixel clock cycle, thereby improving computing speed. In the final stage, the filtering results are normalized to ensure that the dynamic range of the output image is consistent with the input image. The architecture design can support filter cores of different sizes, which can be expanded by adjusting the size of the register matrix, filtering components, and multiplexer.

In the aspect of experiments, the authors added Gaussian noise of different intensities to an 8-b grayscale image and filtered the image using Matlab and ModelSim, respectively. Then, the PSNRdB and MSSIM of the filtered image were compared. The experimental results show that the FPGA-implemented bilateral filter can effectively denoise at different noise levels while preserving the details of the image. The image denoised by this method has almost no difference from the image filtered using Matlab. The authors concluded that by running filters on a Virtex-5 FPGA and comparing them with other bilateral filter implementations, this design achieved high clock frequency (220 MHz), good image frame rate, and fewer multipliers than standard requirements with a larger filter core (5×5).

The experimental results show that the design can efficiently and in real-time complete denoising tasks while having significant advantages in image quality, computational performance, and resource utilization.

4.1.2 FPGA Implementation of Modified Recursive Box Filter-Based Fast Bilateral Filter

Gollamandala Udaykiran Bhargava and Sivakumar Vaazi Gangadharan proposed an FPGA implementation of a modified recursive box filter (MRBF)-based fast bilateral filter (FBF) architecture, which reduces computational complexity, power consumption, and hardware area by using FPGA and optimizing the filtering algorithm [22].

MRBF optimizes traditional box filters recursively, its core is to reduce the computational complexity to O (1) through recursive computation, rather than increasing with the square of the filter window size ω. This optimization enables the filter can efficiently handle larger windows while maintaining real-time performance.

In traditional hardware implementations, such as circuit designs based on CMOS technology, power consumption, area, and latency are the main factors affecting its performance, and the use of hardware resources is also high. To solve these issues, the authors employed quantum-dot cellular automata (QCA) technology, using MAJ instead of traditional logic gates to design a QCA-based full adder (QCA-FA), which can significantly optimize hardware performance. After designing QCA-FA, multiple QCA-FAs can be connected in series to construct an N-bit QCA-RCA. Modified carry select adder (MCSA) based on QCA reduces power consumption and hardware area by optimizing the traditional dual RCA structure into a combination of QCA-RCA and QCA-BEC. In addition, the authors designed an MRBF-FBF architecture on FPGA, using pipeline and parallel computing to achieve further acceleration. Its core modules include initialize block, update F-G block, update P-Q block, delay MRBF block and division block This modular design allows each module to focus on its computing tasks.

The authors conducted a performance analysis of the FPGA architecture of MRBF-FBF and compared it with existing filter and adder architectures. The experimental results obtained indicate that this method is better than existing methods in terms of hardware resource utilization, running speed, and power consumption.

In the aspect of experiments, the authors first added Gaussian white noise (AWGN) to the images, they used FPGA architecture of MRBF-FBF, RBF-based FBF, GFIE, ADF, and MATLAB to denoise noisy images [23 - 25]. The experimental results showed that the images processed by the MRBF-FBF FPGA architecture were highly like the MATLAB simulation results. The PSNR and SSIM of the FPGA architecture of MRBF-FBF are better than other methods.

The experimental results show that the FPGA architecture of MRBF-FBF not only completes denoising tasks with high quality but also achieves optimization in hardware resource utilization and power consumption.

4.2. FPGA Implementation of Image Contour Detection

In the study by Zhengyang Guo et al. in 2010, they used the FPGA platform to carry out parallel calculations using the Sobel operator increased to four directions, effectively reducing the impact of noise while maintaining very accurate edge detection efficiency, and the speed and effect of image processing were faster and better than on the software platform [7].

In the study by Laigong Guo et al. in 2023, they also used the FPGA platform to implement the Canny algorithm that reduced the gradient amplitude and direction calculation complexity, effectively improving the calculation speed and the edge detection effect [8].

5. Conclusion

This article categorizes some image-denoising methods: spatial domain methods and transform domain methods. Spatial domain methods directly process image pixel values, and commonly used linear filters such as mean filters and Wiener filters can smooth out noise but may lead to blurry details. Nonlinear filters such as median filtering and bilateral filtering can better preserve image edge details. The transform domain method uses frequency domain analysis, such as the PCA method, Wavelet transform method and BM3D, to process noise in different frequency bands while preserving the structure and texture of the image. These denoising methods have more advantages through hardware implementation, and FPGA-based architectures are widely used due to their high processing speed and low power consumption. Such as FPGA implementation based on bilateral filters and FPGA implementation of a modified recursive box filter based on fast bilateral filters. These FPGA architectures can efficiently denoise and demonstrate significant advantages in computational performance, image quality, and hardware resource utilization.

Edge detection has made great progress in algorithm optimization in recent years. For example, the adaptive threshold selection Canny algorithm enables it to perform more accurate edge detection on images with different edge information, while the multi-directional Sobel operator can help the algorithm capture edge details with higher accuracy on more complex images, and the variable standard deviation Gaussian filter can improve the accuracy of edge information retention by overcoming boundary effects and removing Gaussian noise. However, the existing improved algorithms still have problems such as high computational complexity when processing complex texture images. Future research can further optimize the hardware implementation efficiency of the algorithm and explore more adaptive methods to improve detection performance in complex scenes.

Acknowledgements

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Acharya T. Image Processing—Principles and Applications. Wiley-Interscience, 2005.

[2] Goyal B., Dogra A., Agrawal S., Sohi B. S., Sharma A. Image denoising review: From classical to state-of-the-art approaches. Information Fusion, 2020, 55: 220-244.

[3] Gong X. Y., Su H., Xu D., Zhang Z. T., Shen F., Yang H. B. An overview of contour detection approaches. International Journal of Automation and Computing, 2018, 15: 656-672.

[4] Rong W., Li Z., Zhang W., Sun L. An improved CANNY edge detection algorithm. Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, 2014: 577-582.

[5] Li Changsheng, Wang Yajuan, Huang Qijun, Chang Sheng. Design of face detection system based on FPGA (Doctoral dissertation), 2011.

[6] Zhang Zhiyong, Pan Ninghui, Zhao Tingyu. A method for calculating the expected clarity value of the central subregion image of the region of interest based on standard deviation weighted Gaussian filtering function and multi-directional Sobel operator. Laser & Optoelectronics Progress, 2024, 61 (18): 1837008-1837008.

[7] Guo Z., Xu W., Chai Z. Image edge detection based on FPGA. Proceedings of the 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science, 2010: 169-171.

[8] Guo L., Wu S. FPGA implementation of a real-time edge detection system based on an improved Canny algorithm. Applied Sciences, 2023, 13 (2): 870.

[9] Verma R., Ali J. A comparative study of various types of image noise and efficient noise removal techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 2013, 3 (10).

[10] Li X., Hu Y., Gao X., Tao D., Ning B. A multi-frame image super-resolution method. Signal Processing, 2010, 90 (2): 405-414.

[11] Zohair A. A., Shamil A. A., Sulong G. Latest methods of image enhancement and restoration for computed tomography: a concise review. Applied Medical Informatics, 2015, 36 (1): 1-12.

[12] Benesty J., Chen J., Huang Y. Study of the widely linear Wiener filter for noise reduction. Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010: 205-208.

[13] Motwani M. C., Gadiya M. C., Motwani R. C., Harris F. C. Survey of image denoising techniques. Proceedings of GSPX, 2004, 27: 27-30.

[14] Pitas I., Venetsanopoulos A. N. Nonlinear digital filters: principles and applications. Springer Science & Business Media, 2013, 84.

[15] Yang R., Yin L., Gabbouj M., Astola J., Neuvo Y. Optimal weighted median filtering under structural constraints. IEEE Transactions on Signal Processing, 1995, 43 (3): 591-604.

[16] Tomasi C., Manduchi R. Bilateral filtering for gray and color images. Proceedings of the Sixth International Conference on Computer Vision, 1998: 839-846.

[17] Jain P., Tyagi V. Spatial and frequency domain filters for restoration of noisy images. IETE Journal of Education, 2013, 54 (2): 108-116.

[18] Zhang L., Dong W., Zhang D., Shi G. Two-stage image denoising by principal component analysis with local pixel grouping. Pattern Recognition, 2010, 43 (4): 1531-1549.

[19] Hou J. H. Research on image denoising approach based on wavelet and its statistical characteristics. Dissertation, Huazhong University of Science and Technology, 2007.

[20] Dabov K., Foi A., Katkovnik V., Egiazarian K. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on Image Processing, 2007, 16 (8): 2080-2095.

[21] Gabiger-Rose A., Kube M., Weigel R., Rose R. An FPGA-based fully synchronized design of a bilateral filter for real-time image denoising. IEEE Transactions on Industrial Electronics, 2013, 61 (8): 4093-4104.

[22] Bhargava G. U., Gangadharan S. V. FPGA implementation of modified recursive box filter-based fast bilateral filter for image denoising. Circuits, Systems, and Signal Processing, 2021, 40 (3): 1438-1457.

[23] Dabhade S. D., Rathna G. N., Chaudhury K. N. A reconfigurable and scalable FPGA architecture for bilateral filtering. IEEE Transactions on Industrial Electronics, 2017, 65 (2): 1459-1469.

[24] Mukherjee D., Mukhopadhyay S. Fast hardware architecture for fixed-point 2D Gaussian filter. AEU-International Journal of Electronics and Communications, 2019, 105: 98-105.

[25] Nair R. R., David E., Rajagopal S. A robust anisotropic diffusion filter with low arithmetic complexity for images. EURASIP Journal on Image and Video Processing, 2019: 1-14.

[26] Jin Pengfei. An improved Sobel image edge detection algorithm. Applied Optics, 2008, 29 (4): 625-628.

[27] Liu Zongming, Zhang Yu, Cao Shuqing, Lu Shan, Ye Dong. Research on image edge feature extraction technology based on hardware description language. Infrared, 2015, 36 (11): 19-24.