

FPGA-Based Improved Sobel Operator Edge Detection

Guangshao Zhou, Shuai Guo, Zhibo Chen

College of Electronic Information, Southwest Minzu University, Chengdu 610225, China

Abstract: Image edge detection is an important field in image processing, with its primary goal being to identify edge information of various objects in images. Traditional edge detection programs are mostly executed serially, resulting in low detection efficiency. This paper proposes an FPGA-based improved Sobel operator edge detection method, harnessing the parallel processing capability of FPGA to enhance detection efficiency. Building upon the Sobel operator in edge detection algorithms, this method increases the convolution calculation dimensions of the Sobel operator, expanding it from 2 directions to 8 directions for more precise edge information. Images captured by OV5640 are processed within the FPGA and displayed via VGA. Compared to other edge detection algorithms, the FPGA-based improved Sobel operator edge detection method efficiently and rapidly detects image edges, ensuring stable image display and accurate edge information.

Keywords: FPGA; Image Processing; Sobel Operator.

1. Introduction

With the widespread application of digital images in modern society, image edge detection has become increasingly important as a core task in the fields of computer vision and image processing. From autonomous driving to medical image analysis, image edge detection provides critical information for various applications, helping us understand the shapes and structures of objects. However, traditional edge detection methods face challenges when dealing with various types of images and noise [1].

In this context, this paper presents an FPGA-based improved Sobel operator edge detection method to enhance the performance and efficiency of edge detection. The Sobel operator is a common edge detection method, but it may yield suboptimal results in certain situations. Our research aims to address these issues and provide new insights for further research in the field of image edge detection.

The structure of the FPGA-based improved Sobel operator edge detection, as shown in Figure 1, consists mainly of four modules: the image capture module, data storage module, image processing module, and VGA display module.

The FPGA section uses ALTERA's Cyclone IV series chip, model EP4CE10F17C8. The image capture module employs an OV5640, which features a 5-megapixel high-definition camera with a fixed focus lens. The VGA display module utilizes VGA driver control to transmit the processed images to the monitor.

2. System Design

2.1. Image Capture Module

The image capture primarily utilizes the OV5640 high-definition camera, which has an actual photosensitive array of 2592×1944 (5 megapixels), allowing it to capture a wealth of image information and support VGA display output of image data [2]. The sensor integrates numerous image processing circuits internally, including automatic exposure control, automatic white balance, ISP (Image Signal Processor), etc., enabling basic processing of the captured images.

The operational principle of OV5640 is as follows: Firstly, the camera's clock is provided by XVCLK, triggering the

reset signal and initiating the camera to begin image capture after initializing registers. Secondly, it coordinates the PCLK (pixel clock), HREF (horizontal reference signal), and VSYNC (vertical synchronization signal) through timing logic to achieve synchronized transmission with the camera's digital image signal. Finally, the image is transmitted to the host computer for display via the 8 data output ports D7-D0 on the DVP interface. The functions of OV5640 are depicted in Figure 2.

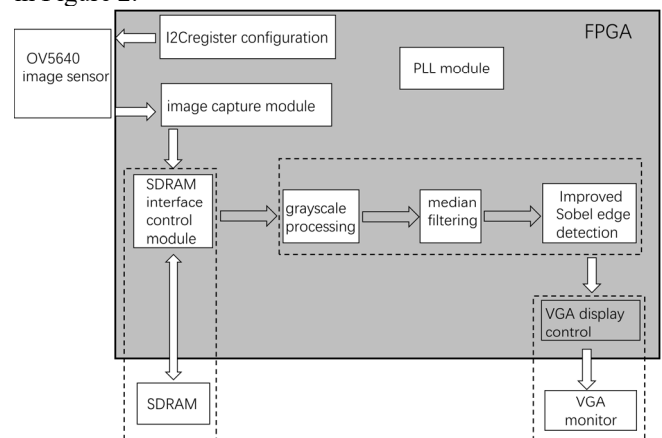


Figure 1. FPGA-Based Improved Sobel Operator Edge Detection System Block Diagram

2.2. Image Storage Module

SDRAM is a type of Synchronous Dynamic Random-Access Memory, where 'synchronous' means that the clock frequency matches the clock frequency of the processor system bus. To ensure that data is not lost, the memory array needs to be constantly refreshed. SDRAM allows for non-linear storage of data, and read/write addresses can be freely chosen [3]. As shown in Figure 3, the storage space of SDRAM is divided into 4 L-Banks. When addressing, one L-Bank needs to be specified, and then the desired row and column can be selected within the chosen L-Bank.

The SDRAM chip model used in this case is 'HY57V2562GTR,' with a capacity of 256Mbit and a 16-bit data bus width. The Storage Structure is shown in Figure 4.

2.3. Image Processing Module

2.3.1. Grayscale Processing

To perform edge detection, it is necessary to process the pixel

grayscale values of an image. There are several mainstream methods for this:

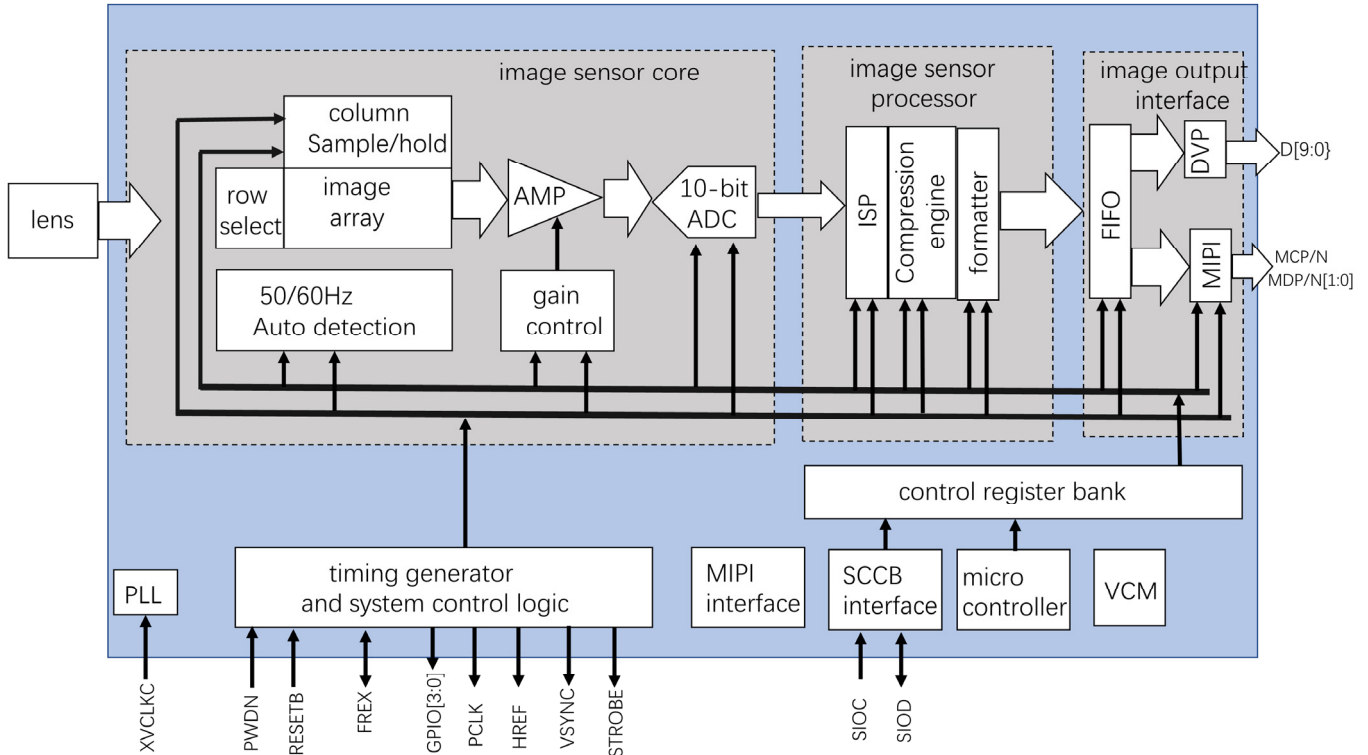


Figure 2. OV5640MRSR block diagram

- (1) **Luminance Method:** This approach involves calculating the luminance by linearly combining the red, green, and blue channels with weighted coefficients. It is designed based on the varying sensitivity of the human eye to different colors. Typically, higher weight is assigned to the green channel because the human eye is more sensitive to green. Its mathematical expression is:

$$Y = 0.299 \times R + 0.587 \times G + 0.144 \times B \quad (1)$$

In this context, where Y represents brightness, and R, G, B respectively represent the values of the red, green, and blue channels.

- (2) **Component Method:** In the component method, only one-color channel from a color image is used to generate a grayscale image. The pixel values of the selected channel are used as the pixel values of the grayscale image. In this case, each pixel's grayscale value contains information from only one-color channel. As needed, the generated grayscale values are then mapped to the range of 0 to 255, resulting in a grayscale image.
- (3) Convert the three-channel RGB image data into a Ycbr image using addition and multiplication operations [4], and then display the image using the Y component. The image transformation algorithm is defined by equations (2), (3), and (4) as shown. If a calculated value is negative, it is set to 0.

$$Y = 0.183 \times R + 0.614 \times G + 0.062 \times B + 16 \quad (2)$$

$$Cb = 0.568(B - Y) + 128 \quad (3)$$

$$Cr = 0.713(R - Y) + 128 \quad (4)$$

Due to the fact that the grayscale image displayed using the Y component in the third method exhibits stronger contrast and a more realistic display compared to the grayscale images displayed in the first two methods, this system adopts the third method for grayscale image processing.

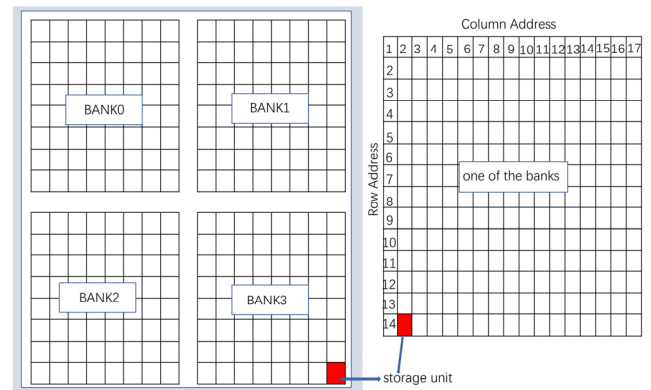


Figure 3. SDRAM Storage Structure

2.3.2. Median Filtering

Median filtering is a common spatial domain filtering technique primarily used for noise reduction and image smoothing. Unlike pixel-based weighted averages, it relies on pixel value sorting [5]. The working principle can be divided into three steps:

The first step involves defining a sliding window, typically in the shape of a square or rectangle, and then sliding this window across the image. In this case, a 3x3 window size is used, as shown in Figure 5c).

In the second step, at each window position, the pixel values within the window are sorted, usually in ascending or

descending order, resulting in an ordered list of pixel values. In this case, an ascending order is used.

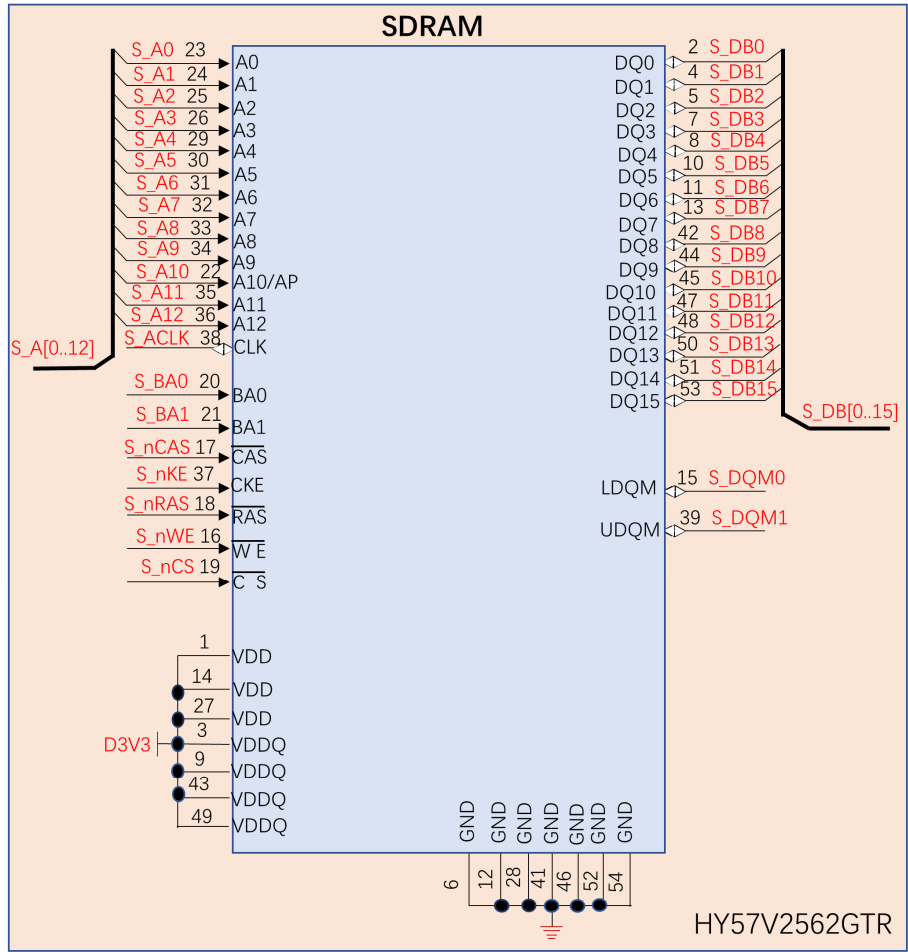


Figure 4. HY57V2562GTR Hardware Circuit Diagram

The third step entails selecting the middle value from the sorted list as the new pixel value for the current pixel position. Its mathematical expression is:

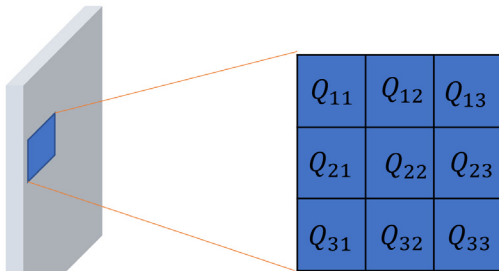
$$F(x, y) = \text{median}\{g(s, t)\}, (s, t) \in S_{xy} \quad (5)$$

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

a) x direction

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

b) y direction



c) 3x3 image Matrix

Figure 5. Traditional Sobel Operator and 3x3 Image Matrix

2.3.3. Improved Sobel Algorithm

The Sobel operator is a commonly used convolution kernel

in image processing and edge detection. It calculates gradient magnitude and compares it to a threshold, making it a classic edge detection algorithm [6-10]. The traditional Sobel edge detection uses operators that are typically divided into two directions, as shown in Figure 5a) and Figure 5b): the x-direction for horizontal edge detection and the y-direction for vertical edge detection.

Performing convolution operations with the two directional operators on the 3x3 image matrix shown in Figure 5c), you can obtain the gradients G_{90° (in the x-direction) and G_{180° (in the y-direction) as described by equations (6) and (7):

$$G_{90^\circ} = -1 \times Q_{11} + 1 \times Q_{13} + (-2) \times Q_{21} + 2 \times Q_{23} + (-1) \times Q_{31} + 1 \times Q_{33} \quad (6)$$

$$G_{180^\circ} = 1 \times Q_{11} + 2 \times Q_{12} + 1 \times Q_{13} + (-1) \times Q_{31} + (-2) \times Q_{32} + (-1) \times Q_{33} \quad (7)$$

By squaring G_{90° and G_{180° separately and then taking the square root, you obtain the gradient magnitude G as shown in equation (8).

$$G = \sqrt{(G_{90^\circ})^2 + (G_{180^\circ})^2} \quad (8)$$

When the gradient magnitude G exceeds the predetermined threshold T , the image edge, denoted as Edge, is set to 1. This signifies that the pixel point is an edge point. Otherwise, if G is less than or equal to the threshold T , the image edge, Edge, is set to 0, indicating that the pixel point is not an edge point, as described in equation (9).

$$Edge = \begin{cases} 1 & G \geq T \\ 0 & else \end{cases} \quad (9)$$

Traditional Sobel edge detection is known to have limitations in terms of the limited dimensionality of convolution factors, which can result in lower accuracy in image detection and a lack of richness in capturing fine edge details.

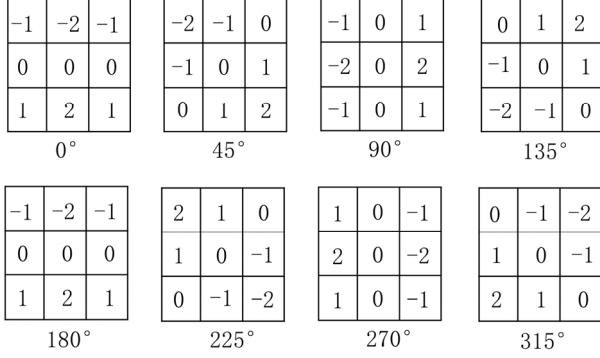


Figure 6. Improved 8-Direction Sobel Operator

To address the issue of limited convolution factor dimensionality, this design increases the convolution factor dimensions to eight directions, as shown in Figure 6. These directions include 0° , 45° , 90° , 135° , 180° , 225° , 270° , and 315° .

Using the parallel processing capabilities of FPGA, convolution operations can be performed with the eight-directional operators on the 3×3 region centered around the pixel to be detected, as described in equation (10) to (17).

$$G_{0^\circ} = (-1) \times Q_{11} + (-2) \times Q_{12} + 1 \times Q_{13} + 1 \times Q_{31} + 2 \times Q_{32} + 1 \times Q_{33} \quad (10)$$

$$G_{45^\circ} = (-1) \times Q_{11} + (-2) \times Q_{12} + (-1) \times Q_{21} + 1 \times Q_{23} + 1 \times Q_{32} + 2 \times Q_{33} \quad (11)$$

$$G_{90^\circ} = (-1) \times Q_{11} + 1 \times Q_{13} + (-2) \times Q_{21} + 2 \times Q_{23} + (-1) \times Q_{31} + 1 \times Q_{33} \quad (12)$$

$$G_{135^\circ} = 1 \times Q_{12} + 2 \times Q_{13} + (-1) \times Q_{21} + 1 \times Q_{23} + (-2) \times Q_{31} + (-1) \times Q_{32} \quad (13)$$

$$G_{180^\circ} = 1 \times Q_{11} + 2 \times Q_{12} + 1 \times Q_{13} + (-1) \times Q_{31} + (-2) \times Q_{32} + (-1) \times Q_{33} \quad (14)$$

$$G_{225^\circ} = 1 \times Q_{11} + 2 \times Q_{12} + 1 \times Q_{21} + (-1) \times Q_{23} + (-1) \times Q_{32} + (-2) \times Q_{33} \quad (15)$$

$$G_{270^\circ} = 1 \times Q_{11} + (-1) \times Q_{13} + 2 \times Q_{21} + (-2) \times Q_{23} + 1 \times Q_{31} + (-1) \times Q_{33} \quad (16)$$

$$G_{315^\circ} = (-1) \times Q_{12} + (-2) \times Q_{13} + 1 \times Q_{21} + (-1) \times Q_{23} + 2 \times Q_{31} + 1 \times Q_{32} \quad (17)$$

The gradient magnitude G is obtained by summing the squares of the convolution results from all eight directions, as described in equation (18)-(20):

$$G_1 = G_{0^\circ}^2 + G_{45^\circ}^2 + G_{90^\circ}^2 + G_{135^\circ}^2 \quad (18)$$

$$G_2 = G_{180^\circ}^2 + G_{225^\circ}^2 + G_{270^\circ}^2 + G_{315^\circ}^2 \quad (19)$$

$$G = \sqrt{G_1 + G_2} \quad (20)$$

The comparison of the gradient magnitude G with the preset threshold T , where if G is greater than or equal to the threshold T , $Edge$ is set to 1; otherwise, the image $Edge$ is set to 0. In this case, the chosen threshold T is 250, as shown in equation (21).

$$Edge = \begin{cases} 1 & G \geq 250 \\ 0 & else \end{cases} \quad (21)$$

2.4. VGA Display

The VGA (Video Graphics Array) video graphics array is a computer display standard introduced by IBM in 1987 that uses analog signals for data transmission. The VGA interface is a dedicated interface for outputting data that conforms to the VGA standard. This interface consists of 15 pins, organized into 3 rows with 5 holes each. It is used to transmit analog signals for red, green, blue, as well as horizontal and vertical synchronization signals. VGA monitors are known for their low cost, simplicity of structure, and versatility [11-12].

The operating principle of a VGA monitor is to start scanning from the upper-left corner of the screen, moving from left to right, and repeating this left-to-right scan at the beginning of each new row after completing a row. After scanning each row, it uses a row synchronization signal for synchronization. Once all rows are scanned, it forms a frame and uses a field synchronization signal for field synchronization. The timing diagram for row synchronization is shown in Figure 7, and the timing diagram for field synchronization is shown in Figure 8.

2.5. PLL Module

In FPGA design, due to the different frequency requirements of various modules, a PLL (Phase-Locked Loop) module is commonly used to perform clock signal division or multiplication on existing clock signals. In the context of improving the Sobel edge detection algorithm, there are the following clock signal requirements: The I2C register configuration module requires a 50MHz clock. The SDRAM controller requires a 100MHz signal, A 65MHz phase-offset clock is also needed. The VGA display controller module also requires a 50MHz clock signal. To meet the clock signal requirements of the entire system, a PLL IP core is employed. This PLL IP core uses the external 50MHz clock signal as a reference to generate three different clock signals with distinct frequencies.

3. System Testing and Validation

To validate the effectiveness of the improved Sobel algorithm proposed in this paper for image edge detection, Verilog HDL code was developed using the Quartus II development environment. The code was compiled and loaded through JTAG for programming the FPGA. The experimental environment and results are shown in Figure 9.

In Figure 9: A) Represents the original image. In this experiment, the OV5640 camera was used for edge detection of flowers. B) Shows the grayscale image after grayscale processing. C) Displays the grayscale image after median filtering. D) Illustrates the result of traditional unimproved Sobel algorithm applied to flower image edge detection. It can be observed that the image contains a considerable amount of noise and false edges, and some edge information is lost or discontinuous in both the x and y directions. E) Presents the result of this design. It is evident that both edge continuity and image detail richness are superior to the traditional Sobel

algorithm. Additionally, local noise and false edges are effectively controlled, resulting in better edge continuity.

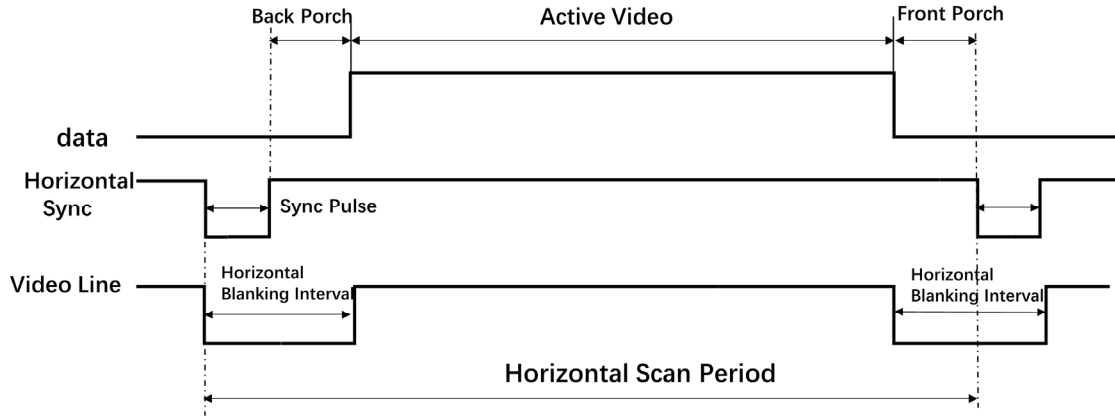


Figure 7. Horizontal Sync Timing Diagram

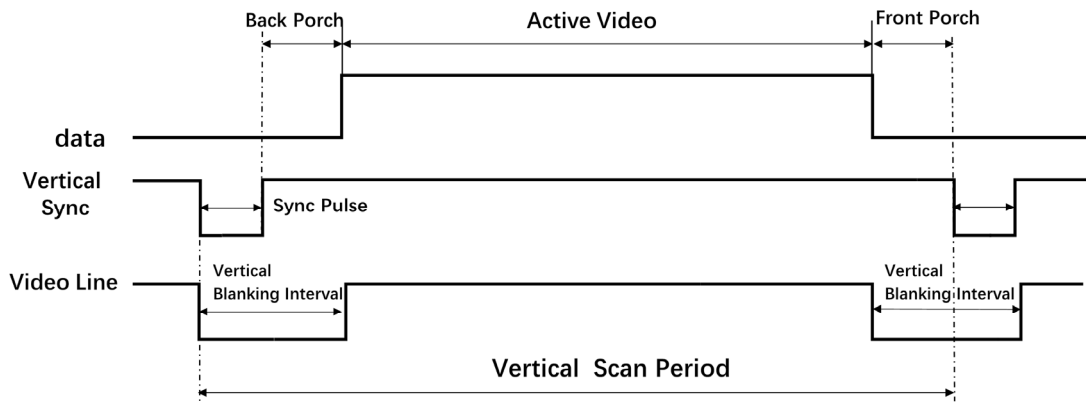


Figure 8. Vertical Sync Timing Diagram

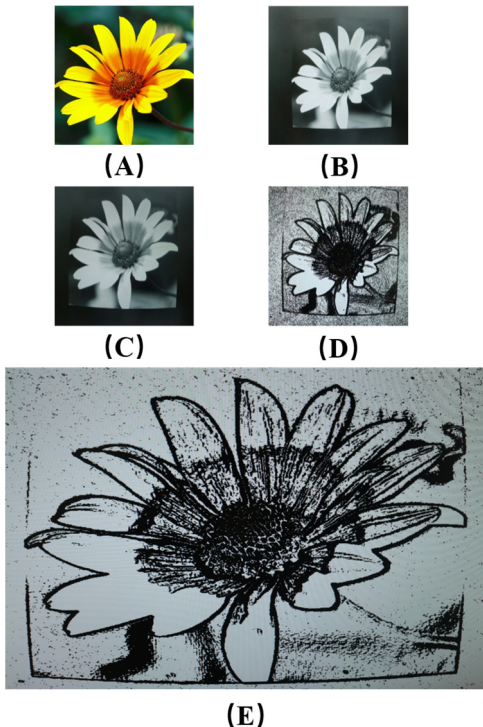


Figure 9. Edge Detection Comparison Chart

4. Conclusion

This paper addresses issues with traditional Sobel edge detection, such as limited convolution factor dimensions, resulting in lower accuracy in image detection and a lack of richness in capturing fine edge details. To overcome these

limitations, an improved Sobel edge detection algorithm is proposed, which increases the dimensionality to eight directions and is implemented on an FPGA. Through on-board experiments, the feasibility of this improved algorithm is verified. Compared to traditional algorithms and software implementations, this paper achieves higher accuracy, captures richer image edge details, and offers improved computational speed. It holds significant practical value in the field of image processing.

References

- [1] Chen Lei, Ye Yan. Multi-Directional Adaptive Threshold Edge Detection Algorithm and FPGA Parallel Implementation. *Wireless Communication Technology*, 2019, 28(04): 21-26.
- [2] Chen Yueyu. Comparative Analysis of Edge Detection Algorithms. *Agricultural Network Information*, 2012(06): 31-33.
- [3] Li Hongyan, Fu Yirong, Qiao Jiawei, et al. An Adaptive Image Scaling Algorithm Based on Edge Operators. *Software Engineering*, 2021, 24(07): 12-14.
- [4] Sun Baiyang, Leng Jianwei, Zhao Jiaqi. Research and Implementation of FPGA-Based Sobel Edge Detection Algorithm. *Chemical Engineering Automation and Instrumentation*, 2018, 45(03): 180-183+231.
- [5] Liu Tengfei, Liu Wei. Design of Image Acquisition and Binary Image Display System with FPGA. *Microcontrollers and Embedded Systems Applications*, 2020, 20(10): 39-42.
- [6] Chen Xianrui. Design of Real-Time Image Edge Detection System Based on FPGA. Northeast Petroleum University, 2023. DOI: 10.26995/d.cnki.gdqsc.2023.000365.

- [7] Liu Mingchuang, Sun Haijiang, Wang Jiacheng, et al. Design of Image Acquisition and Storage System Based on FPGA. Mechanical and Electrical Engineering Technology, 2023, 52(05): 193-199.
- [8] Li Xu. Research and FPGA Implementation of Image Edge Detection Algorithms. Shenyang Jianzhu University, 2021. DOI: 10.27809/d.cnki.gsjgc.2020.000076.
- [9] Ma Shuye. Research on Droplet Image Edge Detection Algorithm Based on FPGA. North University of China, 2023. DOI: 10.26926/d.cnki.gbfgu.2022.000403.
- [10] Qi Xin, Chen Jianfeng, Luo Weilin. FPGA Implementation of Image Preprocessing Algorithms. Foreign Electronic Measurement Technology, 2021, 40(02): 102-107.
- [11] Gui Huaimeng. FPGA-Based VGA Image Display Design. Computer Programming Skills and Maintenance, 2019(08): 136-138.
- [12] Guo Jingfeng. FPGA and SD Card-Based VGA Image Display and Switching Device. Electronic Testing, 2022, 36(07): 98-100.