International Journal of Science and Engineering Applications
Volume 14-Issue 03, 31 – 39, 2025, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1403.1008

Real-Time Detection of Strip Surface Defects Based on FPGA

Yanding Wang
Yangtze University
School of Electronic Information and Electrical Engineering
Jingzhou City, Hubei Province
China

Zhong Peng
Yangtze University
School of Electronic Information and Electrical Engineering
Jingzhou City, Hubei Province
China

Jiang Liu
Yangtze University
School of Electronic Information and Electrical Engineering
Jingzhou City, Hubei Province
China

Abstract: The steel industry is a pillar industry of the national economy, accounting for about 5% of China's GDP. As one of the important raw materials in the iron and steel industry, the surface quality of strip steel directly determines the quality of the final product. However, in the actual production process, due to the limitations of the process flow and production environment, the surface of the strip steel often appears scratches, punching, silk spots, inclusions and other defects. These defects not only affect the appearance quality of the product, but also may cause the steel to be more prone to corrosion and rupture, thereby affecting the safety and reliability of the product. The traditional strip defect image detection system is difficult to meet the needs of millimeter real-time processing because of its limited processing speed. In contrast, FPgas have powerful parallel processing capabilities and low latency characteristics, which are particularly suitable for applications such as strip surface defect detection requiring real-time feedback and high precision. Therefore, whether in order to ensure the stable supply of the domestic market, or in response to the national "Belt and Road" initiative to promote foreign exports, the FPGA-based strip surface defect image detection system has important theoretical significance and broad practical application prospects.

According to the requirements of strip surface defect detection, Intel Cyclone IV EP4CE10F17C8 is selected as the FPGA chip, OV5640 CMOS camera is used for image acquisition, and W9825G6KH-6 SDRAM is used for image data storage. PCF8591T is used for AD/DA conversion of video data and successfully builds a complete hardware platform. The platform realizes the whole process from the acquisition, processing, storage, transmission to the final display of the strip defect image. Implement the FPGA top-down design process, first of all, design the IIC communication protocol interface, which is used to configure the camera and realize its acquisition function. According to the SDRAM manual, the initialization, automatic refresh, read and write operations and arbitration commands of SDRAM are completed in order to efficiently store and manage image data. In order to coordinate the reading and writing of image data better, an asynchronous FIFO control module is designed to ensure the stability and high efficiency of data transmission. Finally, based on VGA sequence diagram, VGA driver circuit is designed to realize the real-time display of image data on the display screen.

In view of the high sensitivity of traditional Canny algorithm in the face of salt-and-pepper noise, an adaptive median filter is designed to replace the Gaussian filter. By dynamically adjusting the size and form of the convolutional template, the method can adaptively select the most suitable filtering mode according to different noise levels, effectively suppress the interference of salt and pepper noise, and improve the image quality. In order to solve the problem of false detection that may occur in the recognition of edge points by the traditional Canny algorithm, this paper uses a four-direction 3×3 Sobel operator to replace the original two-direction 2×2 convolution template. This improvement enables edge detection in four directions: horizontal, vertical, top left to bottom right, top right to bottom left, making edge detection more comprehensive and clear. In addition, to solve the problem that the uncertainty of image display may be caused by manually setting the threshold in the traditional Canny operator, an adaptive threshold method is proposed, and the relationship between high and low thresholds is set to double.

Finally, the system test and analysis are carried out. Through the comprehensive comparison of each filter algorithm template, the results show that the adaptive median filter algorithm has the best performance in both subjective visual effect and objective indicators PSNR and SSIM, which is suitable for this system. Combined with the joint simulation of MATLAB and ModelSim, the improved algorithm is superior to the traditional method in terms of subjective visual effect, and can capture image edges more accurately. Moreover, the real-time transmission effect in the on-board experiment is good, and it can support real-time image processing and defect detection tasks. System resource consumption is low, the total time of FPGA processing a 640×480 resolution image is 13.03 milliseconds, 10 times faster than the software algorithm

Keywords: Steel strip surface defect detection；FPGA; Canny algorithm;adaptive median filtering; Adaptive threshold

1. Edge detection operator and FPGA hardware platform are introduced

Image edge detection is a key technology in computer vision and image processing, which aims to identify edges or object contours in images. Edges are often areas of dramatic changes in grayscale, color, [1]or texture, and are critical for tasks such as image segmentation, object detection, and recognition. According to the variation form of pixel gray value, edge can be divided into two types: step type and roof typ, as shown in Figure 1, in which Figure a is step type and Figure b is roof type. In the step edge, the gray value suddenly changes from one flat region to another flat region, and there is an obvious peak value in the first derivative, which represents the place where the brightness changes the fastest, that is, the edge point. For the second derivative, there is a positive and negative value on both sides of the point, and the zero intersection point is 0, [2]which corresponds to the edge point. In the roof edge, the gray value presents a symmetrical circular arc change, and the gray value gradually rises to a peak value and then decreases, similar to a Gaussian function.

The first derivative is a symmetrical waveform. Similarly, the position of the second step edge point is the edge point where the obvious minimum value of the second derivative appears.

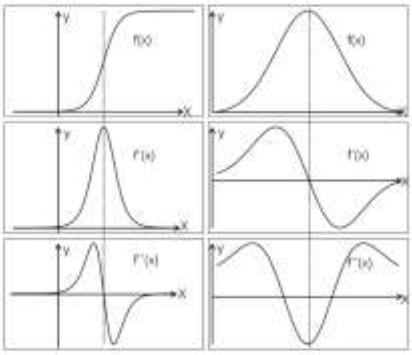

Figure1-1 Step and roof type

Canny operator is more superior in detection performance and is considered to be one of the best edge detection methods widely used at present. The objective of Canny operator is to find all the true edges in the image, [3]while minimizing the influence of false edges to ensure the positioning accuracy and continuity of edges, mainly based on the idea of gradient and Non-Maximum Suppression. The procedure is shown in Figure 2.

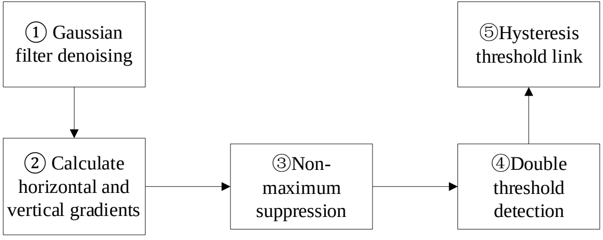
① Gaussian filter denoising
② Calculate horizontal and vertical gradients
③Non-maximum suppression
④Double threshold detection
⑤Hysteresis threshold link

Figure2 Canny algorithm flow

Non-maximum suppression precisely locates edge points by looking for a local maximum in the gradient direction, that is, comparing G(x,y) on the current pixel with its neighborhood pixels. If the current pixel is a non-local maximum, its intensity is set to 0 and only the true edge points are preserved. The maximum threshold value of Thigh and the minimum threshold value of Tlow are set. Edges are classified into strong edges, weak edges and non-edges according to different thresholds, so as to effectively reduce false edges and reduce the omission rate.

Table 1 lists various edge detection algorithms for comprehensive analysis. In view of the high precision requirement of strip surface defect detection and the interference of salt and pepper noise, it is very important to choose a stable and excellent edge detection method. Canny operator ensures accurate edge positioning because of its own smooth filtering and non-maximum suppression, double threshold detection and hysterical connection enhance edge continuity and integrity, making its detection effect closer to the real image, which meets the requirements of this study.

Table1 Summary of five types of edge detection algorithms

algorithm	peculiarity
Roberts	Simple, fast and sensitive to noise
Sobel	Good robustness and edge enhancement
Canny	Accurate detection and good edge continuity
Laplacian	The calculation is simple and direction-independent
Prewitt	Excellent robustness, suitable for smooth images

On the basis of theoretical analysis, the above algorithm is simulated by MATLAB to verify its actual effect, as shown in Figure 3.

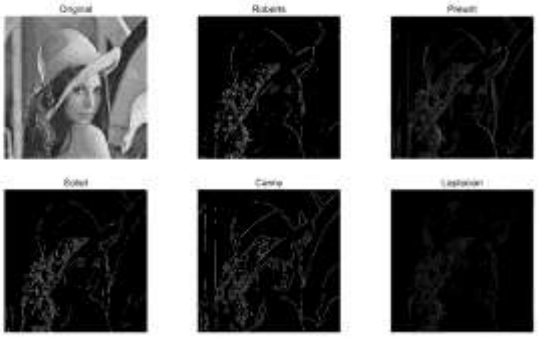

Figure3 Comparison of five algorithms

2. the design of real-time image acquisition and display system

Figure 4 shows the overall system scheme. The system is mainly divided into image acquisition module, image processing module, image display and storage module. OV5640 camera is used as image input device to collect original image data, and the output image data is transmitted to FPGA in the form of digital signal. SCCB[4] is responsible for configuring the camera register protocol. The collected images are processed by the image processing module. In the gray level conversion step, the color images are converted to gray level images to reduce the amount of data while retaining the main information of the images. The improved Canny edge detection algorithm is used to extract the edge of the gray level images. The image data processed by the algorithm is stored by SDRAM memory, and the read and write FIFO in SDRAM controller controls the read and write rate of image data, which plays an important role in the frame rate and synchronization of the image. Finally, the read data is displayed by VGA driver module to drive VGA display. The PLL clock embedded in FPGA is used to manage and allocate the system clock to ensure the synchronization of image acquisition, processing and display. This chapter will introduce the design of image acquisition module and image display storage module in detail.

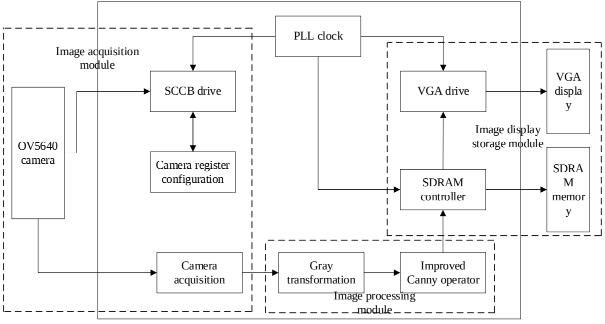

Figure 3 Overall system design

2.1 OV5640 Camera

OV5640 is a CMOS camera module produced by OmniVision. The sensor can realize all the functions of a single chip 5-megapixel camera and is controlled through the Serial Camera Control Bus (SCCB) interface.[5] Figure 4 shows the functional block diagram of OV5640 camera in the official manual. It can be seen that the camera module integrates a number of key functional components, including image sensor core (ISC), image signal processor (ISP), [6]image output interface, clock management and automatic lens focus control, etc. The camera has high integration and flexibility, making it the first choice in this system. Table 3-1 describes how the OV5640 connects to the FPGA hardware interfacezed.

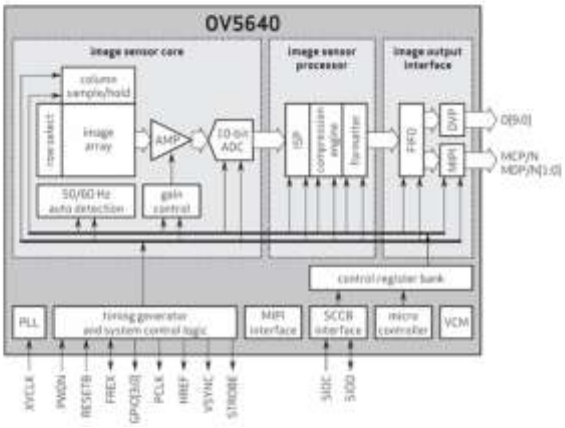

Figure 4 OV5640 block diagram

(1)ISC module is the core acquisition unit of OV5640 camera, which is responsible for converting optical signals into electrical signals, including Image Array, Gain Control, analog amplifier (AMP), 10-bit ADC and 50/60Hz automatic detection. The image array consists of 5 megapixel units that capture light signals; Gain control adjusts the output signal strength to adapt to low light environments; AMP amplifies weak light signal to enhance subsequent digital processing effect; A 10-bit ADC converts an analog signal to a digital signal for ISP processing; Automatic detection at 50/60Hz optimizes image quality and eliminates stroboscopic effects.[7]

(2)ISP is the core processing module of OV5640, which is responsible for processing the original data output by the sensor and improving the image quality. Its core functions include noise reduction, white balance, gamma correction and chromatic aberration, and the image is converted to RGB, YUV, JPEG and other formats through the compression engine to adapt to different storage and transmission needs.

(3) Image Output Interface (Image Output Interface) mainly transmits the processed image data to the external device through FIFO buffering, supporting two main interfaces: Digital Video Port (DVP) and Mobile Industry Processor Interface (MIPI). DVP works with HREF, VSYNC, and PCLK for parallel communication with FPgas.

(4) The control logic module includes PLL(Phase Locked Loop), SCCB interface, timing generation control and VCM(Voice Coil Motor), wherein PLL mainly provides the required working clock for the camera; SCCB interface reads and writes registers through SCCB interface protocol, time sequence generation control is to control and manage line-field synchronization signal and pixel clock, VCM is to control the lens autofocus function.

2.2 IIC Protocol

The IIC protocol also uses two communication lines, one SCL for the serial clock line and one SDA for the serial data line. The SCL is generated by the master device to provide a synchronous clock signal, while the SDA is used for two-way data transmission, supporting the sending and receiving of data. Similar to the SCCB protocol, the IIC protocol requires the data line to remain stable during the high level of the clock signal, and the switch of the data line state can only be performed during the low level of the clock. The timing diagram of the IIC protocol is basically the same as that of the SCCB, including the complete process of the START signal (START), data transmission, reply signal (ACK/NACK) and STOP signal (STOP), to ensure the reliability and consistency of communication.The biggest advantage of the IIC protocol is that it supports the communication structure of one master and many slaves. The master device selects the target device by sending the address of the slave device, supports up to 127 slave devices (7-bit address mode), and distinguishes the transmission direction of the data through the read/write flag bit (R/W bit, R=1 indicates read, W=0 indicates write). [8]The address of the OV5640 device is 7'b1010_000, and the write sequence is shown in Figure5.

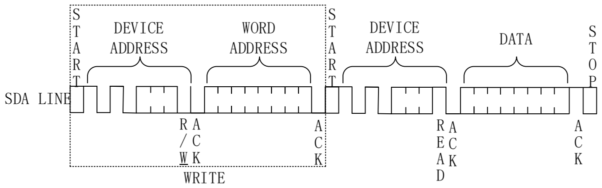

Figure 5 IIC read sequence

2.3 IIC protocol driver design

According to the IIC read and write sequence diagram, IIC communication can be divided into multiple stages and involve a variety of signal variables, so it is very suitable for using state machines to implement I²C protocol drivers. The state machine can accurately complete the functions of start signal generation, data transmission, response detection, single read word writing, 8/16-bit register address switching and stop signal, etc. Figure 6 shows the IIC[9] protocol driver state machine design.

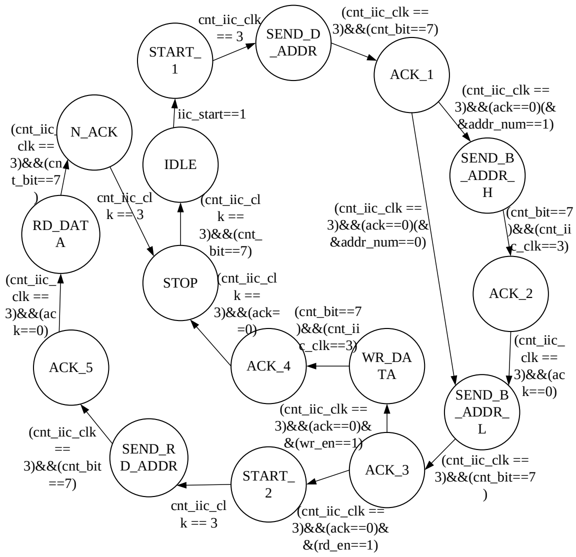
IDLE
iic_start==1
START_1
cnt_iic_clk == 3
SEND_D_ADDR
(cnt_iic_clk == 3)&&(cnt_bit==7)
ACK_1
(cnt_iic_clk == 3)&&(ack==0)(&&addr_num==1)
SEND_B_ADDR_H
(cnt_bit==7)&&(cnt_iic_clk==3)
ACK_2
(cnt_iic_clk == 3)&&(ack==0)
SEND_B_ADDR_L
(cnt_iic_clk == 3)&&(ack==0)(&&addr_num==0)
(cnt_iic_clk == 3)&&(cnt_bit==7)
ACK_3
(cnt_iic_clk == 3)&&(ack==0)&&(wr_en==1)
WR_DATA
(cnt_bit==7)&&(cnt_iic_clk==3)
ACK_4
(cnt_iic_clk == 3)&&(ack==0)
STOP
(cnt_iic_clk == 3)&&(cnt_bit==7)
(cnt_iic_clk == 3)&&(ack==0)&&(rd_en==1)
START_2
cnt_iic_clk == 3
SEND_RD_ADDR
(cnt_iic_clk == 3)&&(cnt_bit==7)
ACK_5
(cnt_iic_clk == 3)&&(ack==0)
RD_DATA
(cnt_iic_clk == 3)&&(cnt_bit==7)
N_ACK
cnt_iic_clk == 3

Figure 6 IIC driver state machine design

As can be seen in the figure above, through the design of the state machine with 16 states, the single write operation and random read operation of the I²C device can be realized, and the driver of the IIC protocol interface can be realized. Each state is introduced below:

(1) Initial state: After power-on, the state machine enters the idle state, waiting for the host (FPGA) to receive a valid read/write start signal, and jumps to the START1 state after receiving the signal.

(2) Start signal state: The host sends a start signal to mark the start of communication, and then jumps to the SEND_D_ADDR_W state.

(3) Sending device address:host sending device address (high 7 bits) and write flag bit (minimum 0, indicating the write operation), and then jump to ACK1 state.

(4) Answer signal status: If the host correctly receives the answer signal (ACK) returned from the device, jump according to the register address bytes: 16-bit register address: jump to SEND_R_ADDR_H (send high byte address). 8-bit register address: Jump to SEND_R_ADDR_L (send low byte address). If no response signal is received, the state machine returns to IDLE.[10]

(5) Send high byte address: The host sends the high 8 bits of the register address, then jumps to the ACK2 state. ACK2: jumps to SEND_R_ADDR_L after receiving the response signal.

(6) Send low byte address: The host sends the low 8 bits of the register address, and then jumps to the ACK3 state.

(7) Answer signal state 3: According to the read/write control signal (read =1, write =0) Jump: Write operation: jump to WR_DATA state. Read operation: Jump to START2 state.

(8) Write data state:The host writes single-byte data to the slave device, and then jumps to ACK4 state.

(9) Response signal state 4 :After receiving the response signal, jump to STOP (stop state).

(10) The second start signal state :The host sends the start signal again and jumps to the SEND_D_ADDR_R state.

(11) Send device addres : Host send device address (high 7 bits) and read flag bit (lowest 1, indicating the read operation), then jump to ACK5 state.

(12) Answer signal state 5:If the answer signal from the device is received, jump to the RD_DATA state.

(13) Read data state : The host reads 8-bit data from the bus, and then jumps to the NACK state.

(14) No answer state :The host sends no answer signal, marking the completion of data reading, and then jumps to the STOP state.

(15) STOP state: The host sends a stop signal to end the current communication and the state machine returns to IDLE.

(16) Error handling: In any response state (ACK1-ACK5), if no correct response signal is received, the state machine immediately jumps back to IDLE state and waits for restart.

According to the description of the state machine, the IIC driver code and test code are written, and the simulation test results through ModelSim are shown as follows, in which Figure 7 is the overall simulation sequence diagram of IIC read and write.

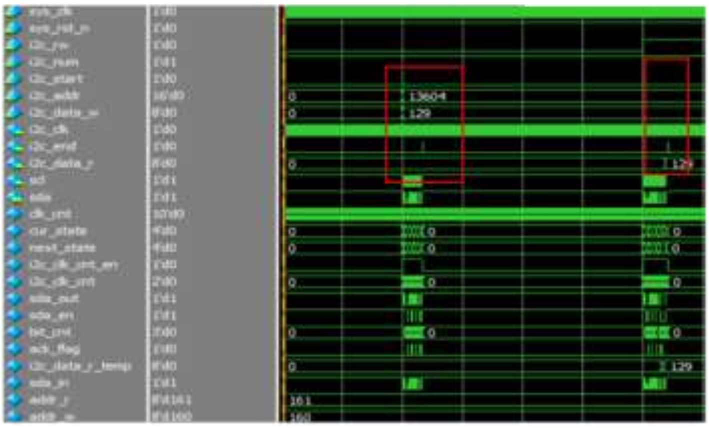

Figure 7 Overall IIC read/write simulation waveform

2.4 SDRAM Top-level design

Figure 8 shows the top layer structure of SDRAM. The overall design mainly includes two modules: FIFO control module and SDRAM control module. Among them, the SDRAM control module is further divided into five sub-modules, which are SDRAM initialization module, SDRAM automatic refresh module,[11] SDRAM read module, SDRAM write module and SDRAM arbitration module. Each of these modules performs its own functions, and jointly completes SDRAM initialization, data reliability maintenance, read and write operations, and multi-request arbitration and timing control.

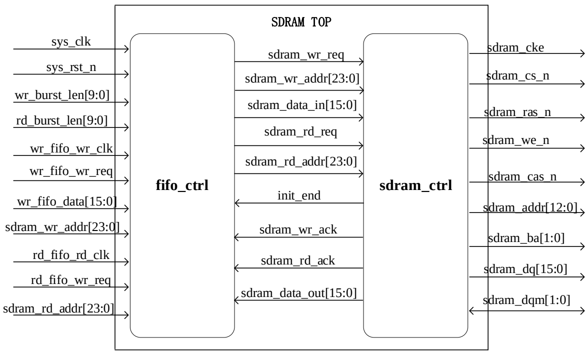

Figure 8 Top-level design of SDRAM

Figure 9 shows the complete flow of the SDRAM simulation model. It includes the whole process of SDRAM initialization, automatic refresh, data writing, pre-charging and data reading. In the initialization phase, pre-charge (PRE), multiple automatic refresh (AREF) and mode register load (LMR) were completed successively, and the working mode of SDRAM was successfully configured (CAS delay =3, burst length =10, sequential access). Then, it enters the automatic refresh phase and sends automatic refresh commands periodically to ensure data reliability. In the WRITE operation, 10 data are written through activation (ACT), [12]burst write (WRITE) and pre-charge (PRE). Finally, in the READ operation, the correctness of the written data is successfully verified by activation (ACT), burst read (READ) and pre-charge (PRE). The whole simulation process is consistent with SDRAM operation principle.

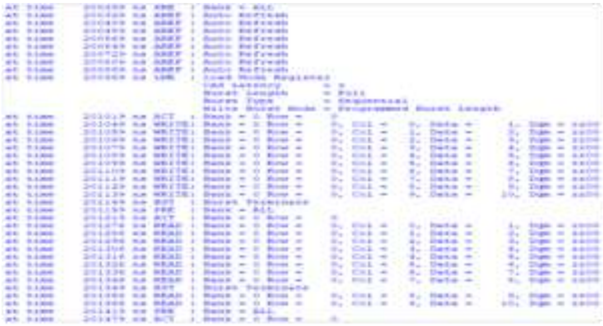

Figure 9 Flowchart of the SDRAM simulation model

2.5 VGA Display Module Design

Figure 10 shows the timing structure of the line synchronization signal (HSYNC). In a complete line scan cycle, there are six stages: sync, back edge, left Border, valid image, Right Border, and front,[14] corresponding to pixel values. In the process of line scanning, the image data is scanned and displayed through the synchronization of HSYNC signal. Among them, the image information is valid only in the effective image stage, and is invalid in the synchronization, back edge, left border, right border and front stage. During the synchronization phase, the HSYNC signal is kept low to mark the start of a new line. In other stages, the HSYNC signal remains high. When entering the next line scan cycle, the HSYNC signal is pulled down again in the synchronization phase, and then pulled up in other phases, forming a periodic change to ensure the correct synchronization and display of each line of images.

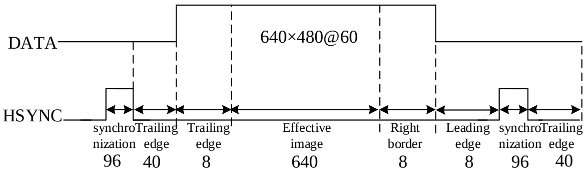

Figure 10 HSYNC timing structure

Figure 11 shows the RTL view of the VGA driver.[15] The input signal pix_data[15:0] is the pixel data obtained from the read FIFO of SDRAM, which is used as the source of the display content of VGA display. vga_clk is a 25MHz drive clock, which is used to control the operation of VGA module and the generation of pixel timing. The output signals include line synchronization signal HSYNC and field synchronization signal VSYNC, which are used to accurately control line scanning and frame refresh of VGA display. The pixel data request signal pix_data_req is used to dynamically request pixel data from SDRAM to ensure the continuity of data flow; At the same time, the module outputs formatted pixel rgb data through rgb[15:0] to VGA display to realize real-time image display. Figure 12shows the simulation waveform. It can be seen that the line field signal is counted and pulled up under the two counters, the line counter technology is pulled up to 799 line synchronization signal, and the other stages are low, the field counter is to count and pull up the line scan period, and the final output effective signal is the output of 16'd5535 for the effective signal in the line field, which is consistent with the above principle.

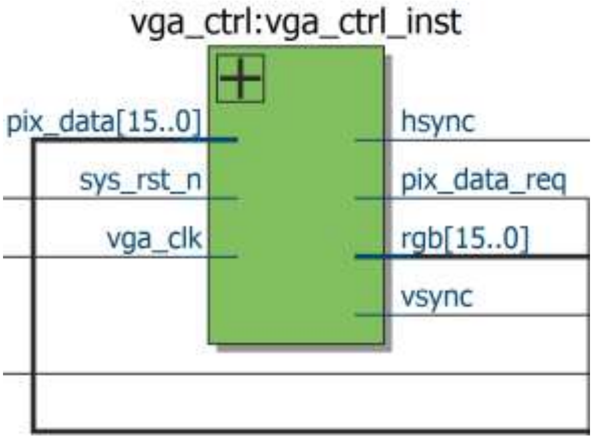

Figure 11 RTL view of the VGA driver

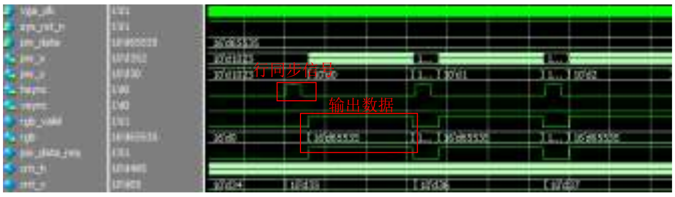

Figure 12 VGA driver simulation waveform

3. Canny algorithm improvement strategy and its FPGA implementation

In order to solve the problem that the Gaussian filter in the traditional Canny operator will blur the edge of the image, which may lead to the loss of image details, and for some high frequency noise denoising effect is limited, the adaptive median filter is proposed to replace the Gaussian filter. In view of the problem that only the gradient amplitude and direction of X axis and Y axis are calculated in the traditional Canny operator, and the collected edge points may be incomplete, an improved Sobel operator is proposed to calculate the gradient amplitude and direction in four directions. In view of the problems such as the uncertainty of image display caused by manual setting of threshold values in the traditional Canny operator, an adaptive threshold method is proposed. Set the relationship between high and low thresholds to double. Figure 13 shows the improved image processing flow diagram. It includes image input[16], gray scale conversion, adaptive median filtering, improved Sobel operator to calculate gradient amplitude and direction, non-maximum suppression, adaptive double threshold, hysteresis threshold segmentation and image output.

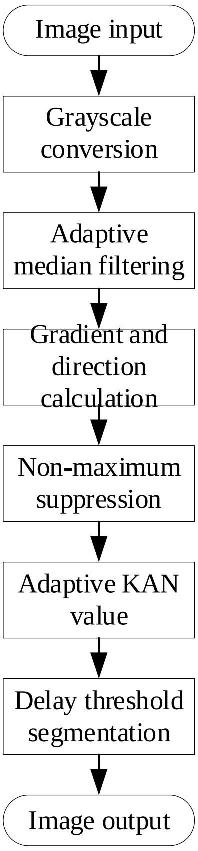
Image input
Grayscale conversion
Adaptive median filtering
Gradient and direction calculation
Non-maximum suppression
Adaptive KAN value
Delay threshold segmentation
Image output

Figure 13 Flowchart of the improved algorithm

Adaptive median filtering (AMF) is a nonlinear filtering technology widely used in image processing, which is mainly used to remove pulse noise and protect image edge and detail information. Compared with the traditional median filter, the adaptive median filter has the ability to dynamically adjust the filter window size, and can determine the filter region adaptively according to the noise distribution characteristics, thus improving the noise removal effect[17]. Figure 14 shows the logical flow diagram of the selection control module for the adaptive median filter. The main function of the module is to judge the noise point adaptively and select the appropriate pixel value as the output according to the pixel value in the sliding window and the sorted statistical result. Where G33xy is the pixel value of the center point of the image sampling window, G33 and G55 are the output of the 3×3 median filter and the output of the 5×5 median filter respectively.

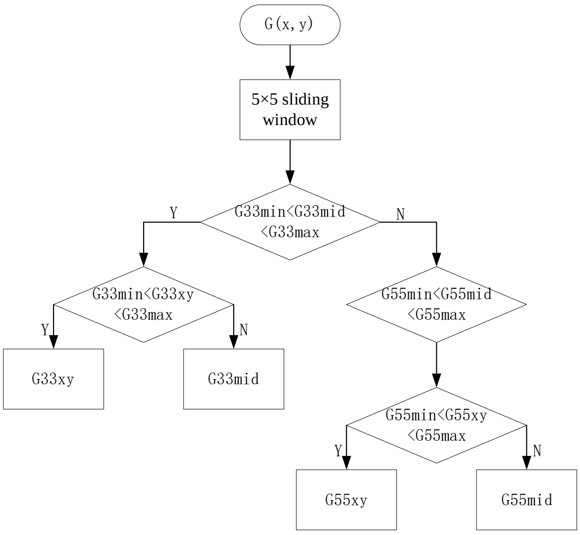

Figure 14 Flowchart of the adaptive median filtering algorithm

Figure 15 Test simulation waveform of the adaptive median filter select controller part. It can be seen that the output results after the algorithm processing and comparison are 193, 198, 192, 204, which is consistent with the output effect of the above algorithm flow chart.

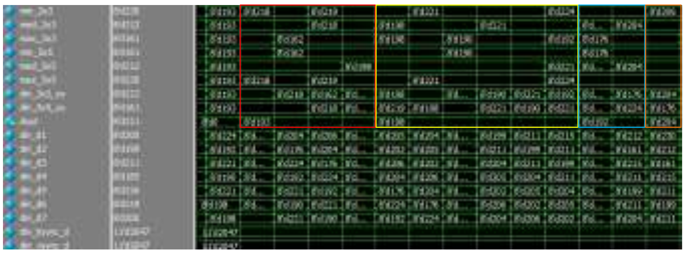

Figure 15 Waveform simulation by adaptive median filtering algorithm

Traditional Canny operator uses fixed threshold value to determine the edge, which has two inherent defects: First, high threshold value can easily lead to missing weak edge detection and damage the continuity of the target contour; Second, low threshold is easy to introduce noise interference, resulting in pseudo-edge response. In order to overcome the above problems, this study adopts dynamic threshold adjustment strategy and proposes an adaptive double threshold method based on local gray statistics. [44] Specifically, the algorithm calculates the local gray mean value of Thigh in real time through the sliding window (3×3 neighborhood) as the reference value of the high threshold, and sets the low threshold value as Tlow=0.5Thigh according to the empirical formula of noise suppression.This dynamic mechanism enables the threshold parameters to be adjusted adaptively with the local contrast of the image: the threshold is automatically raised in the high-texture area to suppress oversegmentation, and the threshold is lowered in the smooth area to enhance edge sensitivity. Compared with the global fixed threshold, this method significantly improves the coherence of weak edges while maintaining strong edge detection capability. The input data is the 9 pixel values of the 3×3 sliding window (8'd36, 8'd129, 8'd9, 8'd99, 8'd13, 8'd141, 8'd141, 8'd101, 8'd18). After the clock rising edge is triggered, the module completes the pixel accumulation operation through a one-level pipeline, outputs the gray value and 10'd687 of the window, and then dynamically generates a high threshold value of Thigh=8'd76 based on the mean value algorithm (∑/9=76.33→ rounded down), and synchronously calculates the low threshold value Tlow=8'd38. Both sum and Thigh have a clock delay, which is consistent with the theoretical results above.

4. System verification and analysis

Figure 16 shows the establishment of the image defect detection system platform in the laboratory environment. Altera's FPGA chip model EP4CE10F17C8 development board was used for the experiment, the program was downloaded and debuggable through JTAG interface, the OV5640 camera was configured using IIC bus, and the Winbond SDRAM chip model W9825G6KH-6 was used. Capacity 256Mbit for image data storage and ADAPTS to the monitor through the on-board VGA port.

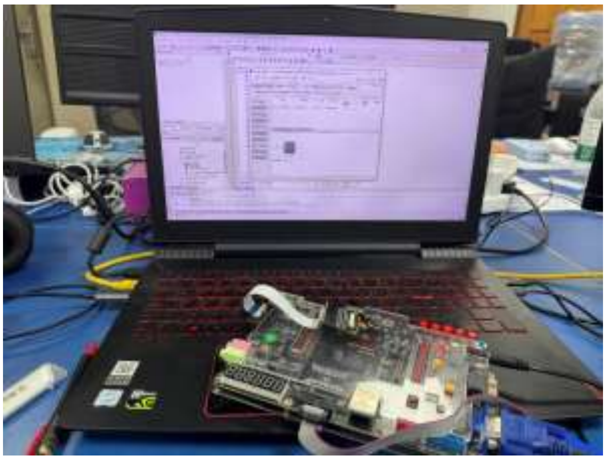

Figure 16 Establishment of the defect detection system platform

In this paper, OV5640 image sensor is adopted in the acquisition system, the output image resolution is set to 640×480, and the output pixel clock frequency of the camera is 72MHz. After the camera is moved, the image display on the display screen is smooth and does not lose frames. Since the camera data acquisition and SDRAM storage data are managed through the FIFO in the system, the image data writing and image data output can be realized at the same time, which greatly reduces the running time and provides stable real-time performance for the subsequent image processing.

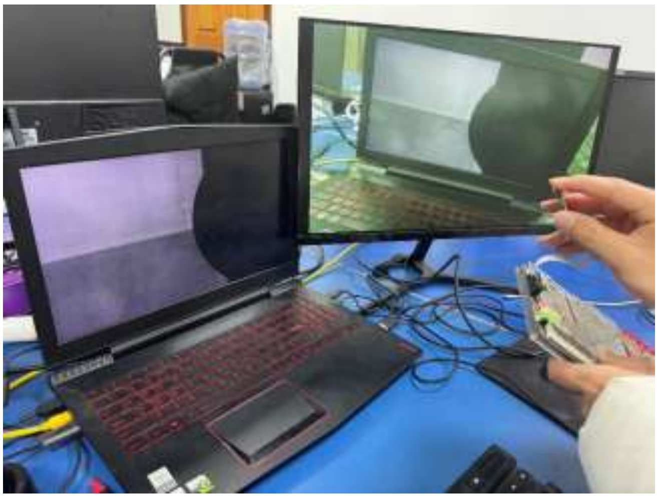

Figure 17 Real-time collection and display demonstration of the upper panel

The adaptive median filter obtains the optimal PSNR (32.6dB) and SSIM (0.934) when the salt-and-pepper noise density is 0.1, and the PSNR increases by 5.84% and 16.01% compared with the 5×5 median filter and 3×3 median filter, respectively. Compared with the 5×5 median filter and the 3×3 median filter, SSIM increased by 2.41% and 5.78% respectively. In order to ensure the comparability and scientifeness of the experimental results, PSNR and SSIM of the two key indicators of each algorithm under different noise density scenarios were not compared.Figure 18 shows the changes of PSNR values of three different algorithms under different noise densities of salt and pepper. In general, with the increase of noise density, the PSNR values of all algorithms show a decreasing trend. Under the same noise density condition, the adaptive median filtering algorithm has the highest PSNR value and the best image quality. In addition, compared with the other two algorithms, the PSNR value of the adaptive median filter algorithm decreases relatively little, showing its advantage in processing images with high noise density.

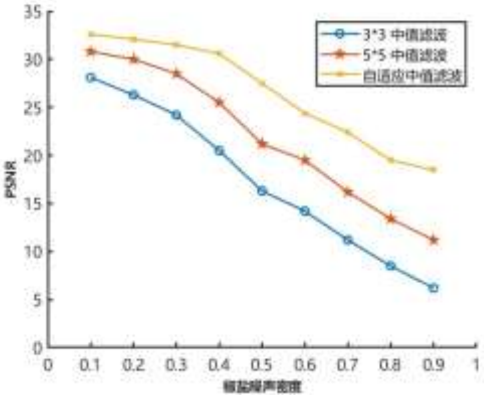

Figure 18 PSNR comparison at different noise densities of salt and pepper

Quartus tool is used to complete RTL synthesis and layout, and the generated bitstream file is burned to FPGA through

JTAG interface. As shown in Figure 5-8, the real-time detection system captures four typical strip surface defects: moon-shaped defects, oil contamination, creases and punching. The detection results of these defects show that the effectiveness and accuracy of the developed algorithm in hardware implementation have been verified.

Figure 19 Display of four types of defects detected by FPGA in real time

This experiment includes camera acquisition, adaptive median filter processing, improved Canny algorithm processing, and writing into SDRAM memory. Finally, VGA display is used to simulate camera acquisition data by converting input image data into byte data, and three-stage pipeline algorithm design is adopted. Figure 20 shows the simulation time for processing a strip steel defect image with a resolution of 640×480

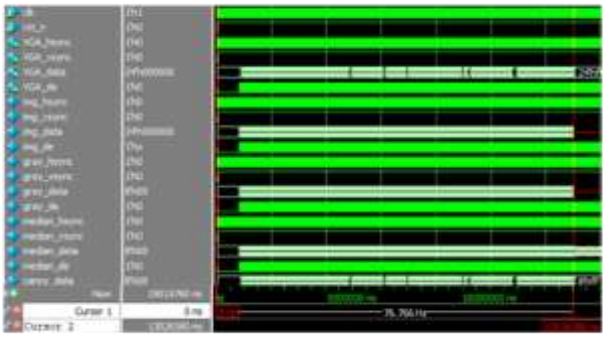

Figure 20 Simulation time of a 640x480 strip steel defect image

As can be observed from the figure, the total time of the entire image processing is 13.02658 milliseconds. In view of the requirement of real-time processing performance in this experiment, in order to evaluate the gap between the running time of the whole system and the running time based on traditional software algorithms, ten picture samples were selected for comparative experiments. The tests were performed in a MATLAB environment using an Intel i7-14650HX processor, an NVIDIA RTX 4060 graphics card, a 24-core CPU, and 16GB DDR5 memory. The test is performed on a PC running Windows 11.

4.1 Establishment and analysis of strip surface defect detection model

In this experiment, YOLOv11, an advanced convolutional neural network model, was adopted to build the target detection system. YOLOv11 is the latest iteration version of YOLO series real-time target detector, which has been optimized in terms of speed and accuracy. YOLOv11 adopts the classic single-stage detector design paradigm, and its core structure is shown in Figure 5-13. It contains the following key components: Backbone, Neck, and Head. The backbone network of YOLOv11 adopts C3K2 block, which replaces the C2f block in the previous version. The design of C3K2 block improves the computing efficiency. The neck network of YOLOv11 has added C2PSA module to further enhance the feature extraction capability. YOLOv11's detection head uses a depth-separable approach to reduce redundant calculations.

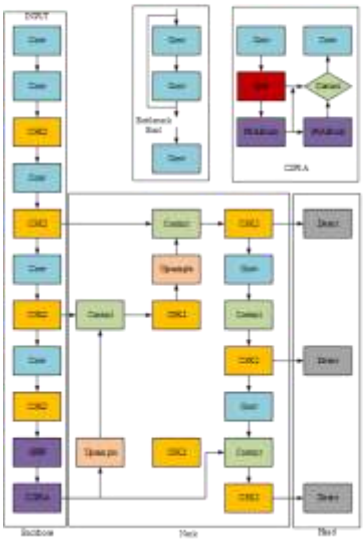

Figure 21 Core architecture of YOLOv11

From the confusion matrix of the original dataset, it can be seen that the accuracy of the six categories in this dataset is higher than 50%, among which the accuracy of the identification punch (d) is 90%, and the accuracy of the moon gap (a) is 80.8%. However, there are still some misclassification, especially for category scratches (c), dents (e) and inclusions (i), which have an accuracy of 43.75%, 14.2% and 30% respectively. This suggests that the features of the original dataset may not be significant enough or overlap in some categories, making it difficult for the model to distinguish these categories correctly. In contrast, the confusion matrix of data set 2 after algorithm processing shows that the number of correct classifications of categories moon gap (a), water spot (h), inclusion (i) and weld (j) has decreased, indicating that the classification accuracy of these categories is still lower than that of the original data set after Sobel edge detection processing. This may indicate that Sobel edge detection does not significantly improve the classification performance of these categories. In the confusion matrix of data set 3 processed using Canny edge detection, the classification effect of class pits (e), fold (f) and inclusions (i) is improved compared with the original data set. In particular, the number of correct categories of pits (e) is significantly increased, and the accuracy rate is 42.9%, which is nearly two times higher than the original data set. This shows that Canny edge detection performs better in reducing misclassification, and the categories of misclassification become more concentrated, further indicating that Canny edge detection processing enhances the stability and accuracy of the model.Compared to dataset 2, dataset 3 shows a more significant improvement. For example, the number of correct classifications for category scratches (c), dents (e) and inclusions (i) increased, with an accuracy of 66.7%, 53.3% and 46.2% respectively, while the number of misclassifications decreased significantly. For lunar gap (a), punch (d) and weld (j), the accuracy rates were 87.5%, 92% and 86%, respectively, indicating that the improved Canny edge detection also achieved good optimization results in these categories of feature extraction.By comparing the values on the main diagonal of the confusion matrix for the four data sets, we can clearly see that dataset 3 shows the deepest color depth, which means that it performs best in classification accuracy. Further analysis shows that the deepening of color depth reflects the increase of more correctly classified samples and the decrease of misclassified samples, indicating that the overall performance of the model has been improved to some extent through the algorithm proposed in this paper. It also proves that the optimization research based on Canny edge detection algorithm is effective and meaningful.

5.Reference

[1] ZHANG Ting. Design of Signal Generator based on MCU [J]. Shanxi Electronic Technology,2018(05):21.

[2] LI Longzhou, Zhou Biying, Zhang Peng. Design of Low Frequency Signal Generator based on AT89C52 [J]. Electronic Design Engineering,2018,26(15):147.

[3] REN Yingjie, Huang Jianqing, Guo Kai, LI Yajun. Design of Simple function Signal Generator based on STC89C51 MCU [J]. Electronic Design Engineering,2018,26(14):91.

[4] Zhu Kaiwang, Yu Jiankun. Design of signal generator based on AT89C51 MCU [J]. Electronic World,2017(09):110.

[5] ZHU Zhaoxu. Design of Function Signal Generator Based on 51 MCU [J]. Digital Technology and Application,2017(02):11.

[6] GUO Hui. Design and Analysis of Simple function Signal Generator based on 51 single chip microcomputer [J]. Electronic Testing,2016(23):3.

[7] Yu Meng, Wang Chao. Design of function Signal generator System based on single chip Microcomputer [J]. Electronic Manufacturing,2015(24):13.

[8] Li Bo, Zu Jing. Digital Synthesis Waveform Generator based on single chip Microcomputer [J]. Instrument Users,2008(04):99-100

[9] ZHANG Xin. Principle and Application of Single Chip Microcomputer [M]. Beijing: Publishing House of Electronics Industry, 2005,8.

[10] Mei Li-feng, WANG Yan-Qiu, WANG Yu-duo. Principle of Single Chip Microcomputer and its Interface Technology [M]. Beijing: Tsinghua University Press, 2006,8.

[11] ZHANG Yigang, Peng Xiyuan, Tan Xiaoyun, Qu Chunbo. MCS-51 MCU Application design [M]. Harbin: Harbin Institute of Technology Press, 1997.

[12] Zhang Hongrun, Yi Tao. Microcontroller application technology course [M]. Beijing: Tsinghua University Press, 2006,10.

[13] DING Xiangrong, Xie Jun, Wang Caishen. Microcontroller C language programming and practice [M]. Beijing: Publishing House of Electronics Industry, 2009,8.

[14] Zhu Dinghua, Dai Ruhua. Principle and Application of Single Chip Microcomputer [M]. Beijing: Beijing Jiaotong University Press, 2003.

[15] Li Qingpeng, Lu Jun, Li Junhua et al. High Precision frequency Signal realization based on single chip Microcomputer and DDS [J]. Application of Electronic Technique, 2002,28 (9) : 50.

[16] Xu Aijun. Principle and Design of intelligent measurement control instrument [M]. 2nd edition. Beijing: University of Aeronautics and Astronautics Press, 2004.

[17] Zhang Peng, Chen Jian. Design of a high-precision waveform generator [J]. SCM and Embedded System Applications, 2005, (1) : 62.