

DOI:10.19651/j.cnki.emt.2416286

# 基于 Sobel 的 FPGA 图像边缘检测系统设计

宋倩男 刘光柱 武乐林 盖明均

(合肥工业大学微电子学院 合肥 230009)

**摘要:** 随着机器视觉相关领域的研究与发展,对图像处理的要求变得更加复杂和多样,而处理实时图像时,边缘信息检测变的尤为重要。本文设计了一种基于 Sobel 算法的 FPGA 图像边缘检测系统,实时进行视频图像的采集、处理和显示。采用自适应阈值和非极大值抑制算法,结合 8 方向 Sobel 边缘检测算法以提升检测精度,进行改进前后的 Sobel 边缘检测算法的仿真验证和硬件实现。采用流水线设计产生滑窗加速图像处理,增强图像处理的实时性。硬件综合实验表明,设计的基于 Sobel 算法的 FPGA 图像边缘检测系统,能高效地实现视频流的图像边缘检测,处理图像速度提升 57%,边缘细节检测全面,增强视频图像处理效率,可用于目标识别及跟踪研究。

**关键词:** FPGA; Sobel 边缘检测; 自适应阈值; 非极大值抑制

中图分类号: TN79 文献标识码: A 国家标准学科分类代码: 510.4050

## Design of FPGA image edge detection system based on Sobel

Song Qiannan Liu Guangzhu Wu Lelin Gai Mingjun

(School of Microelectronics, Hefei University of Technology, Hefei 230009, China)

**Abstract:** As research and development in the field of machine vision continue to advance, the requirements for image processing have become more complex and diverse. Edge information detection is particularly important when processing real-time images. This paper designs an FPGA image edge detection system based on the Sobel algorithm, capable of real-time video image acquisition, processing, and display. Adaptive threshold and non-maximum suppression algorithms are used, combined with an 8-direction Sobel edge detection algorithm to improve detection accuracy. The Sobel edge detection algorithm is validated and implemented in hardware before and after improvement. A pipeline design is adopted to generate a sliding window to accelerate image processing and enhance the real-time performance of image processing. Hardware synthesis experiments show that the FPGA image edge detection system based on the Sobel algorithm can efficiently achieve image edge detection of video streams, improving image processing speed by 57%, providing comprehensive edge detail detection, enhancing video image processing efficiency, and can be used for target recognition and tracking research.

**Keywords:** FPGA; Sobel edge detection; adaptive threshold; non-maximum suppression

## 0 引言

事物的轮廓边缘提供了大量重要信息,这些信息可用于图像识别、目标检测及目标跟踪等图像处理系统中,通过提取图像边缘信息,能显著减少后续图像处理的数据量,这使得边缘检测的应用越来越普遍。随着对图像处理实时性要求的不断提高,优化处理算法以增强边缘检测信息的精确度受到了广泛关注。通过设计实时硬件平台可提升处理速度,以满足图像处理的实时性需求。

经典的边缘检测算法有 Robert 算子<sup>[1]</sup>、Sobel 算子<sup>[2]</sup>和 Canny 算子<sup>[3]</sup>。Robert 算子和 Sobel 算子都是一阶微分

算子,计算简单易于实现,Robert 算子是最早被提出来用作边缘检测的算子,它只能检测边缘却难以定位边缘,边缘检测效果不佳。Canny 算子基于二阶微分算子被提出,利用高斯函数模型来平滑噪声的影响,在处理复杂图像时具有较高的鲁棒性,但其计算复杂度高,硬件实现较为困难。Sobel 算子边缘检测效果远胜于 Robert 算子,检测计算量远小于 Canny 算子,所以本文基于 Sobel 算子进行边缘检测系统的设计。

现场可编程逻辑门阵列 (field programmable gate array, FPGA) 提供了稳定的开发环境,对加速图像处理效果显著<sup>[4]</sup>,基于 FPGA 平台的边缘检测系统被大量研究。

文献[2]使用 Vertex-5 器件设计了一种基于 Sobel 的边缘检测系统,结果表明,该系统使用的硬件资源明显减少。文献[5]对边缘检测算法中的卷积运算加以改进,使用加减和移位运算来代替其中的乘法运算,以减少对硬件资源的需求,但边缘检测仍难以达到实时处理图像的要求。文献[6]对 Canny 边缘检测算法提出改进方法,用 Sobel 边缘检测算子近似计算边缘的梯度,缩短了原有算法运行时间的 6.8%。文献[7]和[8]满足了边缘检测系统的实时性要求,其中,文献[7]运用形态学改进 Sobel 边缘检测算法,在 FPGA 上实现了实时准确定位的要求。随着人工智能的发展,对图像的边缘检测处理提出了新的挑战,要求基于 FPGA 的边缘检测系统可以满足实时性要求的同时,尽可能的提高检测精度,以满足图像处理的实际需要。

本文基于 FPGA 硬件平台,提出了包含图像采集、处理和显示单元的系统架构<sup>[9]</sup>,对传统的 Sobel 边缘检测算子提出了改进,采用 8 方向的 Sobel 算子结合自适应的阈值变化,引用 4 方向的非极大值抑制边缘算法,使得边缘显示更加准确。用流水线硬件实现方式对图像处理进行加速,最终将边缘检测结果显示在显示屏上,系统实现了准确对图像进行实时边缘检测的效果。

## 1 Sobel 边缘检测算法

### 1.1 传统的 Sobel 算法

对图像进行 Sobel 边缘检测时,通过  $3 \times 3$  大小的滑窗依次获得图像像素,与卷积因子进行卷积运算,水平梯度  $\mathbf{G}_x$  和垂直梯度  $\mathbf{G}_y$ ,结果分别如下:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} y_1 & y_2 & y_3 \\ y_4 & y_5 & y_6 \\ y_7 & y_8 & y_9 \end{bmatrix} \quad (1)$$

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \times \begin{bmatrix} y_1 & y_2 & y_3 \\ y_4 & y_5 & y_6 \\ y_7 & y_8 & y_9 \end{bmatrix} \quad (2)$$

化简得:

$$\mathbf{G}_x = y_3 + 2 * y_6 + y_9 - y_1 - 2 * y_4 - y_7 \quad (3)$$

$$\mathbf{G}_y = y_7 + 2 * y_8 + y_9 - y_1 - 2 * y_2 - y_3 \quad (4)$$

再求该像素点对应的梯度幅值,

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \quad (5)$$

则梯度幅度可变换为  $|\mathbf{G}| = |\mathbf{G}_x| + |\mathbf{G}_y|$ , 梯度方向为  $\theta = \arctan \frac{\mathbf{G}_y}{\mathbf{G}_x}$ , 为了保持梯度幅度位宽依然为 8 位, 变换式(5)得:

$$\mathbf{G} = \frac{1}{2}(|\mathbf{G}_x| + |\mathbf{G}_y|) \quad (6)$$

再将  $\mathbf{G}$  的值与设定的阈值比较大小,判断输出是否为边缘,若输出大于阈值,则为 1,反之为 0。本文对式(6)的硬件实现如图 1 所示,对  $\mathbf{G}_x$  和  $\mathbf{G}_y$  进行计算时,  $\mathbf{G}_x$  和  $\mathbf{G}_y$  均为 8 位有符号数,红框内的电路可进行取绝对值运算<sup>[7]</sup>,当

符号位为 1 时,绝对值为其本身,当符号位为 0 时,绝对值为其求反再加 1。

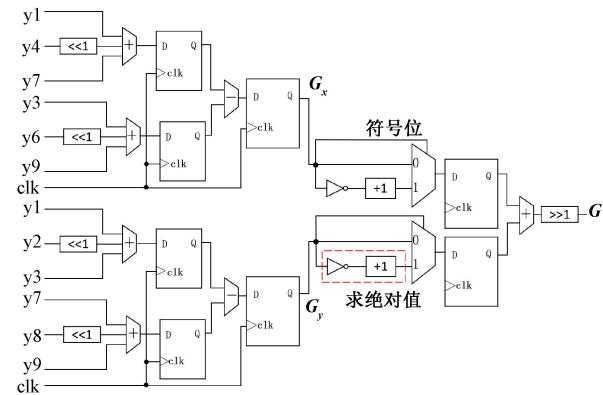


图 1 Sobel 算法实现的硬件图

Fig. 1 Hardware diagram implementing the Sobel algorithm

### 1.2 8 方向 Sobel 算法及自适应阈值的实现

传统的 Sobel 边缘检测算法只能检测图像水平方向和垂直方向边缘,不能检测多方向边缘。文献[10]使用改进的坐标旋转数字计算算法 (coordinate rotation digital computer,CORDIC) 对传统的 Sobel 算法进行硬件实现,虽然提升了图像处理速度,但是检测效果受到限制。为了使得边缘检测的效果在多方向都能达到较好的检测效果,本文对 Sobel 算子进行改进,引入 8 方向的  $3 \times 3$  卷积因子模板如图 2 所示<sup>[11]</sup>。

$\begin{bmatrix} -1 & -2 & -1 \\ -2 & 0 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$
$0^\circ$	$45^\circ$	$90^\circ$	$135^\circ$
$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$
$180^\circ$	$225^\circ$	$270^\circ$	$315^\circ$

图 2 8 方向卷积因子模板

Fig. 2 Eight-direction convolution kernel templates

将  $3 \times 3$  滑窗取得的像素值与 8 方向卷积因子模板进行卷积运算得到梯度值,对计算得到的各个方向的梯度求绝对值均值,作为该像素的梯度幅值。梯度幅值计算公式如下式:

$$\mathbf{G} = \frac{1}{8}(|\mathbf{G}_0| + |\mathbf{G}_1| + |\mathbf{G}_2| + |\mathbf{G}_3| + |\mathbf{G}_4| + |\mathbf{G}_5| + |\mathbf{G}_6| + |\mathbf{G}_7|) \quad (7)$$

改进后的 Sobel 边缘检测算法可以充分捕捉不同方向的边缘信息,从而提高边缘检测的鲁棒性,提高检测结果的稳定性。对各个方向梯度值的计算依然采取图 1 中的硬件实现方式,再将绝对值之和右移 8 位即得到最后的梯度幅值。

在设定判断是否为边缘的阈值时,过大的阈值会漏掉边缘,过小的阈值会误识非边缘点。为了更准确的识别边缘,本文采用自适应阈值来提高检测的精度。自适应阈值就是每个像素点对应一个阈值,这里充分利用中值滤波模块生成阈值,中值滤波模块输出的 $3 \times 3$ 窗口中心值即为滑窗中心像素点对应的阈值大小<sup>[12]</sup>。

### 1.3 非极大值抑制算法

为了减少噪声并有效保留图像中的边缘信息,本文提出了 4 方向的非极大值抑制算法 (non-maximum suppression, NMS)。NMS 通过在像素点的梯度方向寻找局部极大值点进行边缘的提取,对剩余的非极大值点进行边缘的抑制。本系统采用 4 方向的 NMS,将每个像素点的梯度幅值与其梯度方向上的 2 个相邻像素点的梯度幅值比较大小,若大于其相邻像素的梯度幅值,则保留该像素,反之,则设置为 0。

Sobel 算法计算得到了 8 个方向的梯度,梯度方向分别如图 2 所示,在 NMS 中将这些方向简化为 4 个方向,分别是 $0^\circ$ 、 $45^\circ$ 、 $90^\circ$ 和 $135^\circ$ 。将梯度方向 $\theta$ 量化为 4 个方向中的 1 个,对应的量化范围如表 1 所示,根据量化后的方向,选择两个邻近像素进行比较。例如,如果量化方向为 $45^\circ$ ,则需要比较当前像素点与左下和右上的两个像素点的梯度值,将 8 方向的梯度信息与 4 方向的 NMS 进行对应,在去除伪边缘信息的同时有效提取出边缘信息。

表 1 NMS 量化方向

Table 1 NMS quantized direction

$\theta$ 所在范围	量化 $\theta$
$[-22.5^\circ, 22.5^\circ]$ 或 $[157.5^\circ, 202.5^\circ]$	$0^\circ$
$[22.5^\circ, 67.5^\circ]$ 或 $[202.5^\circ, 247.5^\circ]$	$45^\circ$
$[67.5^\circ, 112.5^\circ]$ 或 $[247.5^\circ, 292.5^\circ]$	$90^\circ$
$[112.5^\circ, 157.5^\circ]$ 或 $[292.5^\circ, 337.5^\circ]$	$135^\circ$

## 2 系统架构

在 FPGA 上的系统架构主要由 4 部分组成:图像采集单元、图像缓存单元、图像处理单元和图像显示单元。系统架构如图 3 所示,OV5640 图像传感器通过串行摄像头控制总线协议(serial camera control bus, SCCB)接口进行寄存器配置<sup>[13]</sup>,摄像头 OV5640 采集并输出 RGB565 格式的彩色图像数据,灰度转换模块将 RGB565 图像数据转化为 YCbCr 格式的 8 位灰度图像数据,图像缓存单元将灰度图像数据通过先入先出(first in first out, FIFO)数据缓存器写模块缓存到同步动态随机存储器(synchronous dynamic random access memory, SDRAM)中,然后通过 FIFO 读模块将图像数据从 SDRAM 中读取出来,依次通过中值滤波处理和改进后的 Sobel 边缘检测算法处理后,最终在薄膜晶体管(thin film transistor, TFT)显示器上显示图像的边缘检测结果。

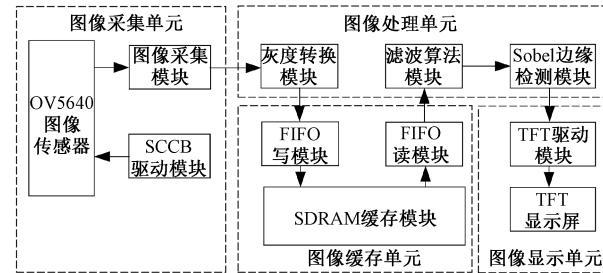


图 3 系统架构

Fig. 3 System architecture diagram

### 2.1 灰度转换模块

将 RGB565 图像数据转化为 YCbCr 格式的灰度图像数据时,需要对 RGB565 末尾补 0 变成 RGB888 格式的图像数据,再转换到 YCbCr 颜色空间。YCbCr 是 8 位 3 通道格式,Y 表示亮度分量,Cb 和 Cr 分别表示蓝色和红色的色度信息,此处进行灰度转换时,只需保留 Y 分量的信息即可,以此完成图像从三通道到单通道的灰度转换<sup>[14]</sup>。其变换关系为:

$$Y = 0.299R + 0.587G + 0.114B \quad (8)$$

FPGA 在进行硬件实现时,浮点数的运算会消耗过多逻辑资源,因此先把浮点数扩大 256 倍并舍去小数部分,再右移 8 位将其还原。

$$Y = 77R + 150G + 29B \quad (9)$$

$$Y = (77R + 150G + 29B) \gg 8 \quad (10)$$

### 2.2 滤波算法模块

图像中不期望出现的随机误差被称为噪声,包含重要信息的分量被称为细节,图像中的噪声和细节会对边缘检测的结果产生重大影响。噪声会对图像包含的信息产生干扰,影响图像的边缘提取,所以需要对图像进行滤波处理,减少图像噪声的同时尽可能保留图像细节。准确区分图像噪声和细节变得十分重要,通常采用频域分析、统计分析、深度学习和空间域滤波等方法区分图像中的噪声和细节,本文采用空间域滤波的方法来平滑噪声保留细节。

常用的对图像进行空间域滤波的算法有中值滤波、均值滤波和高斯滤波等。尽管均值滤波简单易实现,但它会模糊图像的边缘和细节,无法区分噪声和细节。高斯滤波处理后的图像平滑且效果自然,更能保留图像结构信息,但计算复杂度很高且速度较慢。综合来看,中值滤波算法不仅能很好地去除异常值,且具有较强的抗干扰能力,能充分利用 FPGA 的并行性高效地去除噪声且不损失边缘细节。

本文采用中值滤波算法,使用大小为 $3 \times 3$  的窗口在图像上滑动,采用流水线方式生成滑窗,用 FIFO 模块对图像数据实现行缓存,先用 FIFO1 和 FIFO2 分别缓存前两行数据,即 ROW1 和 ROW2,将正在输入的像素值当作第 3 行数据,即 ROW3<sup>[15]</sup>。因图像的分辨率大小为 $800 \times 480$ ,

数据位宽为 8 位,则所用 FIFO 的深度为 800,位宽大小为 8。滑窗的流水线设计如图 4 所示,FIFO 依次取出该行所有像素,每个时钟周期数据移位一次,当完成所有行的操作后,则完成了对一帧图像的处理,图 4 箭头指向的第 3 个时钟周期生成了一个  $3 \times 3$  滑窗,通过  $3 \times 3$  滑窗获得图像像素值。

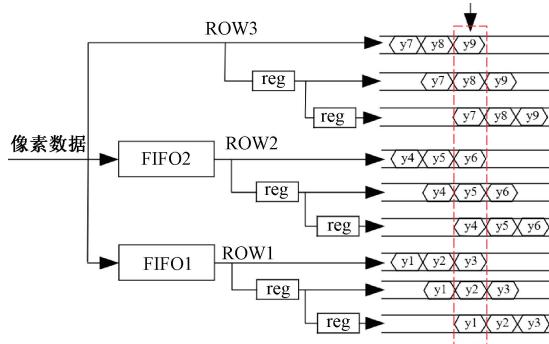


图 4 流水线设计生成滑窗

Fig. 4 Pipeline design generating sliding window

在对中值滤波算法进行硬件实现时,采用三级流水线即可计算出滑动窗口的中值大小。首先,对每行由大到小进行排序;其次,对每列由大到小进行排序;最后,对中间行由大到小进行排序,则滑窗中心的值即为所求。

### 2.3 图像缓存单元

由于 FPGA 片内块随机存储器(block random access memory, BRAM)资源有限,采用 SDRAM 模块来对图像数据进行缓存<sup>[16]</sup>。摄像头采集到的图像数据经灰度转换模块处理后,通过 FIFO 写模块写入 SDRAM 存储器中,通过 FIFO 读模块进入图像处理单元<sup>[17]</sup>。SDRAM 通过 RAS、CAS、CS、WE 四个引脚来控制并实现读写、刷新和预充电等操作。当 FIFO 写模块中的数据达到设定的长度时,向 SDRAM 发送写请求,SDRAM 响应写请求后,FIFO 写模块中的数据写入 SDRAM。当 FIFO 读模块的数据小于设定的长度时,向 SDRAM 发送读请求,从 SDRAM 中读取缓存的图像数据。在 SDRAM 内部采用乒乓操作的存储方式,写数据和读数据分别对不同的 bank 进行操作,当读完且写完数据后,对 2 个 bank 进行切换,以此实现 2 个模块同时工作的功能。

## 3 实验结果与分析

进行实验的硬件平台如图 5 所示,在 Xilinx 公司的 FPGA 开发板上搭建系统,对 OV5640 摄像头采集到的图像进行实时的边缘检测,将检测边缘显示在 TFT 显示屏上。

### 3.1 基于 Matlab 的仿真测试

用 Matlab R2021b 对传统的 Sobel 算法和改进后的 Sobel 算法进行软件仿真,在传统的 Sobel 算法中,阈值大小设为 80,分别对传统的 Sobel 算法,改进后具有自适应阈

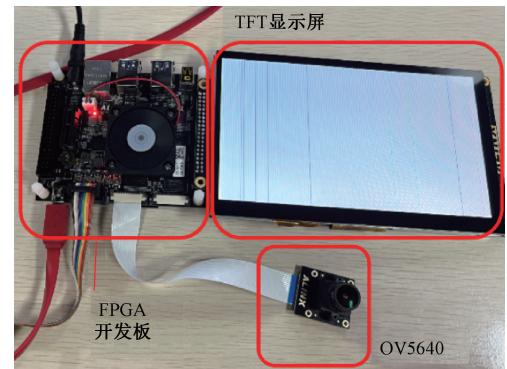


图 5 系统硬件平台

Fig. 5 System hardware platform

值的 8 方向 Sobel 算法以及本文设计的加入 NMS 的改进 Sobel 算法进行仿真,仿真效果如图 6 所示。

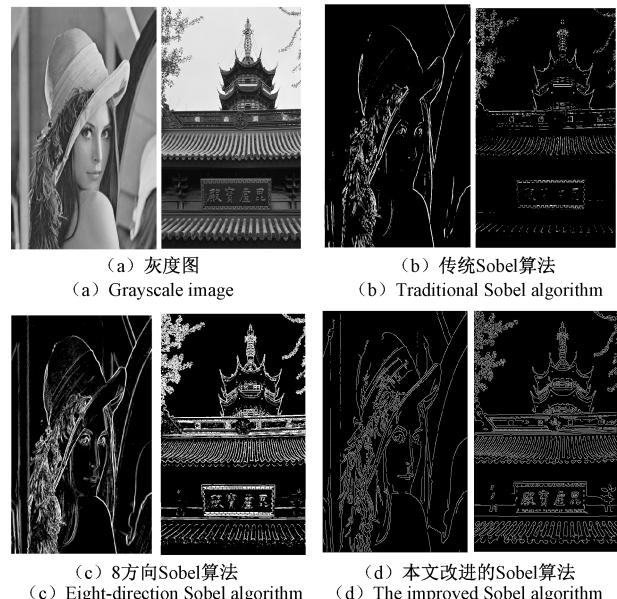


图 6 Matlab 仿真图

Fig. 6 Matlab simulation diagram

图 6(a)和(b)分别为原图的灰度图、传统的 Sobel 算法的边缘检测结果,显然有许多边缘信息未被准确提取;图 6(c)为 8 方向 Sobel 算法的边缘检测结果,比图 6(b)提取到的边缘细节更为清晰,显示的边缘信息更为全面;图 6(d)为改进阈值自适应并添加 NMS 的 Sobel 算法处理结果,其不仅显示了更为全面的边缘细节,且去除了许多抑制边缘点,使边缘得到细化,边缘定位更加的准确。

用 Matlab 分别统计 3 种算法处理图 6(a)右图后图片的黑白像素点数量,其黑白像素数量及白色像素点占比如表 2 所示,白色像素点代表边缘信息,黑色像素点代表非边缘信息,由表 2 可知,8 方向的 Sobel 算法较传统 Sobel 算法多检测到了 14.35% 的边缘信息,这其中也包含了部分为边缘信息,本文改进的 Sobel 算法在 8 方向的 Sobel 算法

上去除了 5.25% 的伪边缘信息,易得改进后算法的边缘检测精度得到提升。

表 2 黑白像素点数量表

Table 2 Black and white pixel count table

边缘检测	图像(白/黑/白占比)
传统 Sobel 算法	69 380/314 620/18.07%
8 方向 Sobel 算法	124 497/259 503/32.42%
本文改进 Sobel 算法	104 332/279 668/27.17%

在 Matlab 上搭建图 3 中的图像处理单元模块,记录处理一帧图像所需时长如表 3 所示,Matlab 处理一帧图像所需平均时间为 26.1 ms。

表 3 Matlab 处理图像时间

Table 3 Matlab image processing time

次数	时间/ms	次数	时间/ms
1	45.1	5	28.9
2	24.6	6	23.5
3	17.2	7	27.4
4	22.1	8	19.8

### 3.2 硬件的综合实现

本设计采用的 FPGA 芯片为 Xilinx 公司的 Zynq UltraScale+MPSoCs CG 系列的芯片,型号为 XCZU2CG-1SFVC784E,该芯片包含 252 个 IOBs、47 232 个 LUTs,240 个 DSP,94 464 个 FF 和 150 个 36 Kb 的 BRAMs 等硬件资源,在 Vivado 2019.2 上对该系统架构进行综合实现,其硬件资源利用率如图 7 所示。由图 7 可知,该芯片硬件资源足以满足系统设计需求。

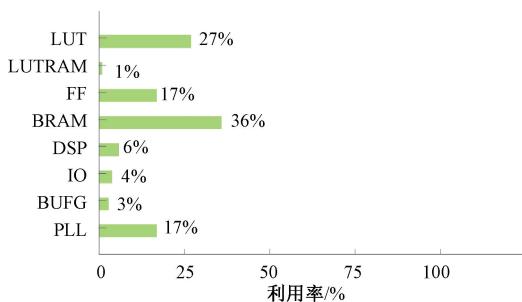


图 7 FPGA 硬件综合结果

Fig. 7 FPGA hardware synthesis results

对本系统进行实时视频流的边缘检测测试,通过 OV5640 获得的灰度图如图 8(a)所示,经过传统 Sobel 边缘检测处理后如图 8(b)所示,经 8 方向 Sobel 算法处理后如图 8(c)所示,经本文改进 Sobel 算法处理后如图 8(d)所示。由图 8 相比较容易得出,本文改进的 Sobel 算法边缘检测系统在处理实时视频时,对边缘细节检测全面且准确,具有较高的检测精度,取得了软件仿真的预期效果。

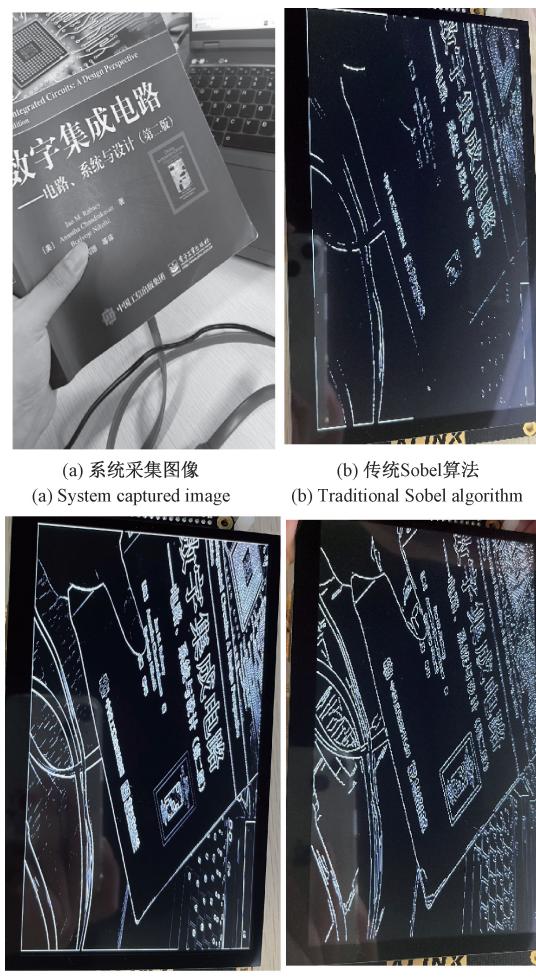


图 8 FPGA 边缘检测效果图

Fig. 8 FPGA edge detection results

系统图像显示单元的时钟频率为 33 MHz,而 TFT 显示屏的分辨率为  $800 \times 480$ ,系统可以每秒显示 27 帧图像,而每秒显示超过 20 帧就能满足人眼实时性的要求,因此,该系统实现了对实时视频流的边缘检测功能。在 FPGA 上实现的边缘检测系统处理一帧相同图像的时长为 11.3 ms,较 Matlab 仿真速度提升 57%,相较于文献[18]处理一帧图像时间减少 0.3 ms,完成了对图像处理的硬件加速,结合边缘检测效果来看,本系统在对实时视频检测的基础上,边缘检测的精度得到了有效提升。

## 4 结论

本文充分利用 FPGA 的并行性,设计了一种基于 FPGA 的图像采集与实时边缘检测系统,通过对 Sobel 算法进行 8 方向卷积因子的改进及阈值的自适应,采用 NMS 算法,获得更细节、更清晰的边缘检测结果。硬件上,设计了一种流水线架构产生滑窗,图像的缓存采用了乒乓存储方式,有效减少了图像处理的时间。最终,实现了对实时图

像进行边缘检测的预期效果。在接下来的研究中,可以对各模块的硬件实现进行优化,从而减少对 FPGA 资源的使用,也可以加入更复杂的图像处理算法,以实现更多功能,如目标跟踪和目标识别等。本系统使用的是 TFT 显示屏,未来可改用分辨率更高清的 HDMI 接口,也可设计通过以太网传输来增大传输距离,使其能在更多领域发挥重要作用。

## 参考文献

- [1] 沈杨. 基于 Robert 算子的车牌图像边缘检测[J]. 电子制作. 电脑维护与应用, 2014(24): 46.
- [2] SHEN Y. Vehicle license plate edge detection based on Robert operator [J]. Electronic Production. Computer Maintenance and Applications, 2014(24): 46.
- [3] CHAPLE G, DARUWALA R D. Design of Sobel operator based image edge detection algorithm on FPGA [C]. 2014 International Conference on Communication and Signal Processing, 2014: 788-792.
- [4] GAURAV K, GHANEKAR U. Image steganography based on Canny edge detection, dilation operator and hybrid coding[J]. Journal of Information Security and Applications, 2018, 41: 41-51.
- [5] MENAKA R, JANARTHANAN R, DEEBA K. FPGA implementation of low power and high speed image edge detection algorithm[J]. Microprocessors and Microsystems, 2020, 75: 103053.
- [6] JIANG J, LIU CH, LING S R. An FPGA implementation for real-time edge detection[J]. Journal of Real-Time Image Processing, 2018, 15: 787-797.
- [7] SANGEETHA D, DEEPA P. FPGA implementation of cost-effective robust Canny edge detection algorithm[J]. Journal of Real-Time Image Processing, 2019, 16: 957-970.
- [8] 谢杰, 陈政, 傅建军. 基于 FPGA 的车牌定位系统研究[J]. 电子与封装, 2022, 22(1): 90-95.
- [9] XIE J, CHEN ZH, FU J J. Research on license plate location system based on FPGA [J]. Electronics & Packaging, 2022, 22(1): 90-95.
- [10] 唐文豪. 基于 FPGA 低照度视频采集与边缘检测算法研究[D]. 西安: 西安科技大学, 2021.
- [11] TANG W H. Research on low illumination video acquisition and edge algorithm based on FPGA [D]. Xi'an: Xi'an University of Science and Technology, 2021.
- [12] XU L, ZHENG D CH. A novel Sobel edge detection accelerator based on reconfigurable architecture [J]. Traitement Du Signal, 2022, 39(4): 1421.
- [13] 吴昊, 刘楠, 丁朋, 等. 改进 CORDIC 算法实现及其在边缘检测中的应用[J]. 电子测量技术, 2023, 46(16): 148-157.
- [14] WU H, LIU N, DING P, et al. Implementation of improved CORDIC algorithm and application in edge detection [J]. Electronic Measurement Technology, 2023, 46(16): 148-157.
- [15] WANG L OU. A new fast median filtering algorithm based on FPGA [C]. IEEE 10th International Conference on ASIC, 2013: 1-4.
- [16] 夏桂书, 何前, 魏永超, 等. 基于以太网与 FPGA 图像处理的边缘检测系统[J]. 仪表技术与传感器, 2023, (10): 56-59.
- [17] XIA G SH, HE Q, WEI Y CH, et al. Edge detection system based on ethernet and FPGA image processing[J]. Instrument Technique and Sensor, 2023, (10): 56-59.
- [18] 陈浩林, 王擎宇, 卢振坤, 等. 基于 FPGA 的高速低噪声目标检测系统[J]. 南京理工大学学报(自然科学版), 2022, 46(5): 623-631.
- [19] CHEN H L, WANG Q Y, LU ZH K, et al. Design of high speed and low noise target detection system based on FPGA[J]. Journal of Nanjing University of Science and Technology, 2022, 46(5): 623-631.
- [20] 金韬韬. 基于 FPGA 的图像采集与处理系统设计[D]. 南京: 南京理工大学, 2021: 29-31.
- [21] JIN T T. Design of image acquisition and processing system based on FPGA [D]. Nanjing: Nanjing University of Science and Technology, 2021: 29-31.
- [22] KASHYAP S, BHANDARI A K, GIRI P. Low resource FPGA implementation based efficient image edge detector architecture [J]. Multimedia Tools Applications, 2024, 83: 25595-25615.
- [23] 王骞, 丁铁夫. 高速图像存储系统中 SDRAM 控制器的实现[J]. 液晶与显示, 2006, 21(1): 48-52.
- [24] WANG Q, DING T F. Realization of SDRAM controller in high-speed image storing system [J]. Chinese Journal of Liquid Crystals and Displays, 2006, 21(1): 48-52.
- [25] 查志勋. 基于 FPGA 的实时边缘检测系统研究[D]. 合肥: 合肥工业大学, 2021.
- [26] CHA ZH X. Research on real-time edge detection system based on FPGA [D]. Hefei: Hefei University of Technology, 2021.
- [27] 邢志华. 基于 FPGA 的图像实时边缘检测系统研究[D]. 上海: 东华大学, 2023.
- [28] XING ZH H. Research on real-time image edge detection based on FPGA [D]. Shanghai: Donghua University, 2023.

## 作者简介

宋倩男, 硕士研究生, 主要研究方向为数字系统设计和图像处理。

E-mail: 2326302242@qq.com

刘光柱(通信作者), 博士, 讲师, 主要研究方向为智能计算。

E-mail: guangzhuliu@hfut.edu.cn