

# 基于改进的 Canny 算子实时视频边缘检测系统在 FPGA 上的设计与实现

周克良, 周利锋, 刘太钢, 章祖忠

(江西理工大学 电气工程与自动化学院, 江西 赣州 341000)

**摘要:** 为了提高图像边缘检测的性能, 缩短处理时间, 提出了一种基于 FPGA 的实时视频边缘检测系统; 该系统以 EP2C8Q208C8 为实验硬件平台, 首先采用摄像头 OV7670 获取模拟视频数据, 双端口 SDRAM 实现对图像数据的缓存, 利用 FPGA 并行处理的特点, 采用 Verilog HDL 硬件描述语言实现改进的 Canny 边缘检测算法, 最终实现在 VGA 显示屏上显示图像边缘的效果; 实验结果表明, 较传统的边缘检测算法, 该系统边缘检测定位精度高, 对噪声的抗干扰能力强, 能够准确快速的输出图像边缘信息。

**关键词:** FPGA; 边缘检测; Canny 算子; VGA

## Design and Implementation of Real-time Video Edge Detection System Based on Improvement of Canny Algorithm on FPGA

Zhou Keliang, Zhou Lifeng, Liu Taigang, Zhang Zuzhong

(School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China)

**Abstract:** In order to improve the performance of image edge detection and shorten the processing time, we proposed a real-time video edge detection system based on FPGA. The system takes EP2C8Q208C8 as the experimental hardware platform, first the system receive the analog video data by using camera OV7670 and storage this data by the dual-port SDRAM, utilize the characteristic of FPGA parallel processing to realize the improved Canny edge detection algorithm by Verilog HDL hardware description language and finally displayed the image edge on the VGA monitor. The result of experiment show: compared with other traditional edge detection algorithm have high positioning accuracy, strong anti-interference ability of the noise, able to output image edge information accurately and quickly.

**Keywords:** FPGA; edge detection; Canny algorithm; VGA

### 0 引言

图像边缘包含图像的大部分信息, 主要存在于不同目标, 目标与背景, 目标与目标之间<sup>[1]</sup>, 边缘的提取是图像分析与识别的基础, 是图像分割、目标检测等所依据的主要特征, 然而在边缘的提取过程中边缘和噪声都是高频信号, 抑制噪声的同时也会对边缘的定位产生影响。传统的边缘检测算子如 sobel, roberts, kirch 和 prewitt 算子是局部窗口梯度算子<sup>[2-3]</sup>, 操作简单, 易于实现, 对噪声敏感, 抗干扰性差。Canny 边缘检测算子具有较好的检测效果, 但需要人为设定高低阈值<sup>[4]</sup>, 易检测出虚假边缘或丢失局部边缘且这些边缘检测大都在通用的 PC 或 DSP 芯片上利用软件来实现, 处理速度慢, 达不到实时性的要求<sup>[5]</sup>。

随着嵌入式系统的飞速发展, 基于传统 Canny 边缘检测的基础, 本文首先采用中值滤波代替高斯滤波提高对图像噪声的抑制能力, 然后基于图像梯度幅值直方图的特性, 自适应生成动态阈值, 克服传统边缘检测算法需人为设定阈值的不足<sup>[6]</sup>。采用可编程逻辑器件 FPGA 实现对此系统的控制运行。实验结果表明, 该系统处理速度快且具有较强的抗噪能力, 检测精度高, 实时性强。

### 1 传统 Canny 边缘检测算法的原理

边缘检测是图像处理的重要环节, 1986 年 Canny 以较低的信噪比, 较高的定位精度以及最大程度的单边缘响应的最优准则<sup>[4-7]</sup>推导出了 Canny 边缘检测算法。其算法的实现主要包括以下过程, 首先, 采用高斯滤波器实现对图像的平滑处理, 然后采用一阶偏导有限差分的方法计算梯度的幅值和方向, 再对梯度幅值进行非极大值抑制<sup>[8]</sup>, 最后利用双阈值算法检测并得出图像的边缘。

其中高斯滤波的主要目的是对原始图像进行平滑处理, 假设  $f(x, y)$  为原始图像,  $G(x, y)$  是二维高斯函数, 其中:  $G(x, y) = (1/2\pi\sigma^2)e^{-(x^2+y^2)/2\sigma^2}$   $S(x, y)$  为平滑后的图像, 则有:

$$S(x, y) = G(x, y) * f(x, y) \quad (1)$$

平滑后图像边缘方向上的法向量方向为:

$$n = \frac{\partial s / \partial n}{|\partial s / \partial n|} \quad (2)$$

平滑后图像二阶导数的零点数即是图像的边缘: 即

$$\partial^2 S / \partial n^2 = 0 \quad (3)$$

将非极大值抑制后的图像按  $Th_1$ ,  $Th_2$  进行两次阈值化处理得到图像  $H_1$  和  $H_2$ , 其中  $Th_1$  为高阈值,  $Th_2$  为低阈值, 且  $Th_1 = 2Th_2$ 。则经过高阈值处理后图像含较少的噪声和伪边缘, 但也丢失一些真实的边缘, 经过低阈值处理后的图像包含较全面的边缘信息, 但也存在部分虚假的边缘。最后以图像为基础, 图像  $H_2$  为补充得到较全面的边缘图像。

收稿日期: 2015-07-09; 修回日期: 2015-08-25。

基金项目: 江西省教育厅科技资助项目(GJJ13429)。

作者简介: 周克良(1963-), 男, 江西赣州人, 教授, 硕士生导师, 主要从事过程控制、智能控制、嵌入式等方向的研究。

## 2 改进的 Canny 边缘检测算法

### 2.1 中值滤波

传统的 Canny 边缘检测采用高斯滤波实现对图像的平滑处理, 易丢失缓变边缘。中值滤波对脉冲噪声具有较好的滤除作用<sup>[2]</sup>, 在滤除噪声的同时, 能够保护信号的边缘使之不被模糊且算法简单, 易于硬件的实现。本系统实现中值滤波的步骤如下, 读取模板对应的像素的灰度值, 然后将这些灰度值按顺序排列, 取出中间的值赋给模板中心的像素即可。

### 2.2 自适应双阈值的选择

图像受拍照条件, 外部环境等因素的影响, 传统的 Canny 边缘检测算法是人为设定高低阈值, 不能根据不同条件下的图像变换为相应的阈值, 影响对图像的边缘检测。

针对前者的不足, 本文基于梯度直方图的性质, 低阈值位于第一个大波峰和第一个小波峰之间的平坦部分<sup>[6]</sup>, 采用两点梯度幅值作差分的方法, 将梯度幅值直方图转化成差分直方图, 如式 (4):

$$\text{Diff}(i) = \begin{cases} \text{Diff}, & \text{abs}(NMS(i+1) - NMS(i)) > \text{th}, \\ 0, & \text{abs}(NMS(i+1) - NMS(i)) \leq \text{th} \end{cases}$$

$$i = 1, 2, 3, \dots \quad (4)$$

则第一个过零点的梯度幅值就是高阈值, 即有

$$Th_1 = \text{Arg}(\text{Diff}(i) = 0)$$

低阈值:  $Th_2 = \frac{1}{2} Th_1$

式中, NMS 表示经过非极大值抑制后的输出, Arg 是满足  $\text{Diff}(i) = 0$  的第  $i$  像素点的梯度幅值。

## 3 系统的硬件设计

本系统选用现场可编程门阵列 FPGA 作为系统的核心芯片, 利用 FPGA 并行处理的优势, 提高了系统的处理效率。基于 EP2C8Q208C8 的微处理器, 控制各器件间的相互工作。如系统框图图 1 所示, 首先通过 I<sup>2</sup>C 配置模块, 实现对解码芯片的配置以及视频信号的采集, 经过解码芯片将模拟视频信号转化成数字视频信号并输入给 FPGA 控制系统。在 FPGA 控制系统中, 图像数据依次经过中值滤波, 梯度计算, 非极大值抑制等操作后, 进而对非极大值抑制的结果作差分直方图, 根据直方图的性质得到图像的高低阈值, 最终对非极大值抑制后的图像进行双阈值化处理, 得出边缘图像并在显示屏上显示。FPGA 芯片主要完成对外围芯片的控制以及输入数据的处理, 其中 FPGA 控制模块主要包括 I<sup>2</sup>C 配置模块, VGA 控制模块, Canny 边缘检测主要包括中值滤波模块, 梯度计算以及非极大值抑制模块和双阈值化处理模块。系统流程图如图 2 所示。

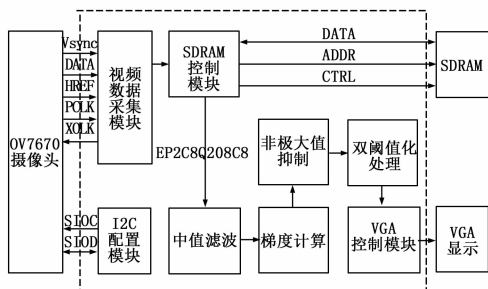


图 1 系统框图

### 3.1 图像的采集

本系统采用 CMOS 技术设计的摄像头 OV7670 完成图像

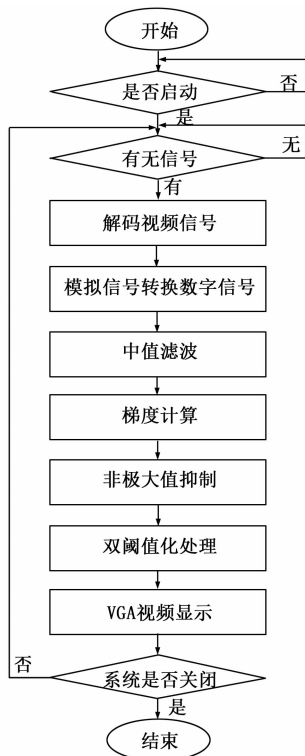


图 2 系统流程图

数据的采集, 具有低耗能, 体积小, 灵活性强的优点, 通过 SCCB 控制总线可输入子采样、整帧、取窗口等方式的 8 位分辨率数据<sup>[9]</sup>。通过 I<sup>2</sup>C 配置模块实现对寄存器的配置, 然后写入器件的 ID 地址, 寄存器地址和控制数据, 最后通过图像传感器的行场同步信号的控制实现对图像数据的采集。

### 3.2 中值滤波的实现

本系统选用 3×3 窗口模板, 以某一像素点为中心, 取出相邻的 8 个像素点及其本身, 按大小排序, 排在中间的像素值即为进行中值滤波处理后的数据, 其结构图如图 3 所示, 获得的 9 个像素位于相邻的三行内, 选用两个 FIFO 存储器存储前两列数据, 当第三行数据到来时, 同时读取缓冲器内的数据。其中行延时的作用是控制三行数据的时间, 使之被同时获取, 列延时的作用是将获取的数据进行列对齐然后再输出, 最终形成 3×3 的窗口数据  $z_1 \sim z_9$ 。

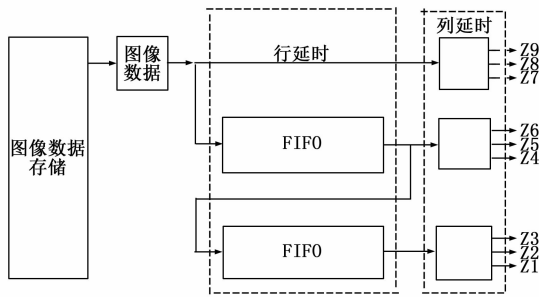


图 3 提取 3×3 模板结构图

### 3.3 自适应双阈值以及阈值化处理的实现

本系统基于梯度直方图的特性实现对双阈值的自适应选择, 其电路原理图如图 4 所示, 整幅图像的全部像素点的梯度幅值经过地址选择器, 将对应的内容经过累加器 2 加 1 后再次写入该寄存器组。待图像结束信号到来后, 在时钟有效期内, 每个

时钟到来累加器 1 的内容自动加 1, 累加结果作为寄存器的地址, 将相应寄存器的内容以及下一个地址寄存器的内容分别送入寄存器 a 和寄存器 b<sup>[6]</sup>, 然后将它们的内容进行差分, 差分结果经过比较器再与 0 比较, 按公式 (4) 即可得到高低阈值。

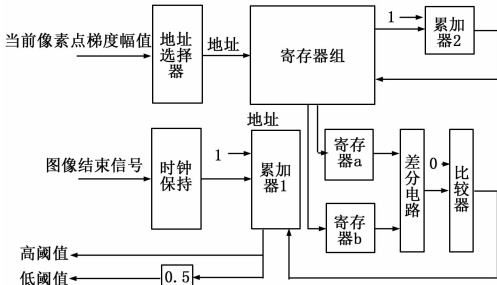


图 4 自适应双阈值选择原理图

经过非极大值抑制后的图像进行直方图统计<sup>[6]</sup>, 令经过非极大值抑制后的图像记为  $g$ , 将其中的各个点  $(i, j)$  分别通过比较器同高低阈值进行对比, 将小于高阈值的点置 0, 大于高阈值的点置 1, 得到强边缘图像  $g_1$ 。将大于低阈值小于高阈值的像素点置为 1, 其它置为 0, 得到弱边缘图像  $g_2$ 。若  $g_1$  为 1, 则此点是边缘点, 若  $g_1$  为 0,  $g_2$  为 1 且  $g_1$  的 8 邻域中有强边缘点, 此点为边缘点。边缘点输出 1, 非边缘点输出为 0, 最终得到边缘图像, 其电路框图如图 5 所示。

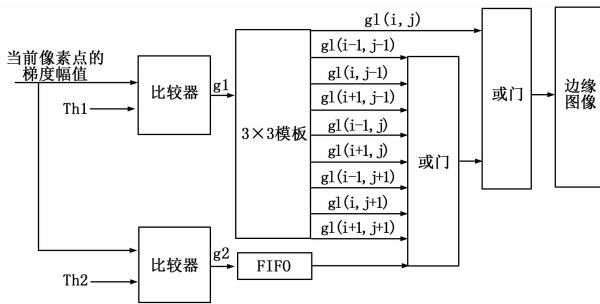


图 5 边缘判定原理图

### 3.4 VGA 显示模块

实现边缘图像在 VGA 显示屏上的显示主要是产生行同步信号, 场同步信号和 RGB 信号<sup>[10]</sup>。通过行场扫描确定图像显示的位置, 并同时产生行同步信号与场同步信号, 最后根据行场扫描的位置显示预先要显示的内容。其显示屏电路连接图如图 6 所示, 其中芯片 ADV7123 实现数据的编码, R0~R9、G0~G9、B0~B9 分别为红色、绿色、蓝色的像素数据输入, IOR、IOG、IOB 为红、绿、蓝电流输出, CLOCK 是时钟输入, SYNC 是复合同步控制输入, 通过 DB15 实现显示屏与编码芯片的连接。

## 4 系统的软件设计

对各芯片的初始化设置决定着该系统运行的成功与否, 包括对 FPGA 系统的初始化, 完成对通用 I/O 口的配置以及系统时钟的设置, 完成摄像头 OV7670 的初始化实现对所需寄存器的配置, 通过代码编写, 实现对彩色图像的灰度处理, 最终进行 VGA 的初始化, 实现边缘图像的显示。

### 4.1 灰度图像

摄像头完成对视频数据的采集, 最终实现对图像数据的边缘检测, 首先需要对被采集图像进行灰度处理, 部分代码如下:

```
assign red_data=vga_data_in[15:11];
```

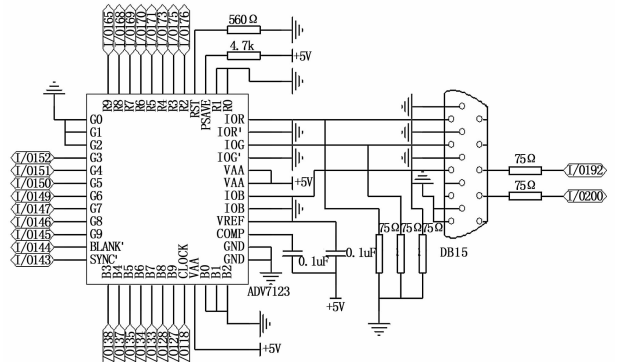


图 6 VGA 电路连接图

```
//取出红色分量
assign red_data=vga_data_in[10:5];
//取出绿色分量
assign blue_data=vga_data_in[4:0];
//取出蓝色分量
assign red_data_1=(red_data*3)/10;
assign green_data_1=(green_data*6)/10;
assign blue_data_1=(blue_data*1)/10;
assign vga_data=red_data_1[4:0]
+green_data_1[5:0]+blue_data_1[4:0];
assign vga_data_out={vga_data[5:1],
vga_data[5:0],vga_data[5:1]};
.....
```

### 4.2 改进 Canny 边缘检测

本系统采用中值滤波代替高斯滤波, 提高对图片噪声的抑制能力, 中值滤波的实现主要是产生  $3 \times 3$  的模板, 取出模板灰度像素的中间值作为模板中心的像素, 部分代码如下:

```
always@(posedge clk or negedge rst_n)
begin
if(!rst_n)
begin
mdf_a22_d1<=16'b0;
mdf_a22_d2<=16'b0;
mdf_a22_d3<=16'b0;
mdf_a22_d4<=16'b0;
mdf_a22_d5<=16'b0;
mdf_a22_d6<=16'b0;
end//初始化 mdf_a22_d*
else
begin
mdf_a22_d1<=mdf_a22;
mdf_a22_d2<=mdf_a22_d1;
mdf_a22_d3<=mdf_a22_d2;
mdf_a22_d4<=mdf_a22_d3;
mdf_a22_d5<=mdf_a22_d4;
mdf_a22_d6<=mdf_a22_d5;
end//生成 mdf_a22_d*
end
.....
always@(posedge clk or negedge rst_n)
begin
if(!rst_n)
begin
hs_d1<=1'b0;
```

```

hs_d2<=1'b0;
hs_d3<=1'b0;
hs_d4<=1'b0;
hs_d5<=1'b0;
hs_out<=1'b0;
vs_d1<=1'b0;
vs_d2<=1'b0;
vs_d3<=1'b0;
vs_d4<=1'b0;
vs_d5<=1'b0;
vs_out<=1'b0;
end // hs_out,vs_out 初始化
else
begin
hs_d1<=hs;
hs_d2<=hs_d1;
hs_d3<=hs_d2;
hs_d4<=hs_d3;
hs_d5<=hs_d4;
hs_out<=hs_d5;
vs_d1<=vs;
vs_d2<=vs_d1;
vs_d3<=vs_d2;
vs_d4<=vs_d3;
vs_d5<=vs_d4;
vs_out<=vs_d5;
end //产生 hs_out,vs_out
end
.....

```

### 4.3 VGA 初始化

实现图像边缘的显示，主要是产生行场同步信号，完成对显示屏的扫描，其代码如下：

```

assign sync=1b0;
assign blank=~(count_h<11'd160||count_v<11'd44)
assign vga_clock=~clk;
always@(posedge clk or negedge n_reset)
begin if(! n_reset)
count_h<=11'b0;
else if(count_h==11'd799)
count_h<=11'd0;
else
count_h<=count_h+1'b1;
end //行扫描
reg[6:0] frame_cnt;
always@(posedge clk or negedge n_reset)
begin if(! n_reset)
count_v<=11'b0;
else if(count_v==11'd524)
begin
count_v<=11'b0;
frame_out<=frame_cnt+1'b1;
end
else if(count_h==7b11111111)
count_v<=count_v+1'b1;
end //场扫描

```

## 5 实验验证与分析

按上述设计搭建电路，将自适应阈值的实时视频边缘检测系统在 FPGA 上实现，基于 Quartus2 软件的综合仿真，系统

的响应时间为 3.6 s，传统的实时视频边缘检测系统的响应时间为 5 s，基于 FPGA 控制的响应速度大幅提高。

实验结果如图 7 所示，其中图 7 (a) 是原始图像，图 7 (b) 是经过 Sobel 边缘检测后的图像，图 7 (c) 是经过传统 Canny 边缘检测后的图像，图 7 (d) 是本系统设计的边缘检测后的图像。实验结果表明，在图像纹理比较复杂的情况下，经典的 Sobel 边缘检测丢失了大量的真实边缘，检测效果不好，传统的 Canny 边缘检测，边缘检测较全面，但受噪声的影响，含有大量的伪边缘，本系统实现的边缘检测有效的去除了图像中的噪声，图像边缘定位精度高，能够很好的实现实时视频的边缘检测，达到了实验的目标。

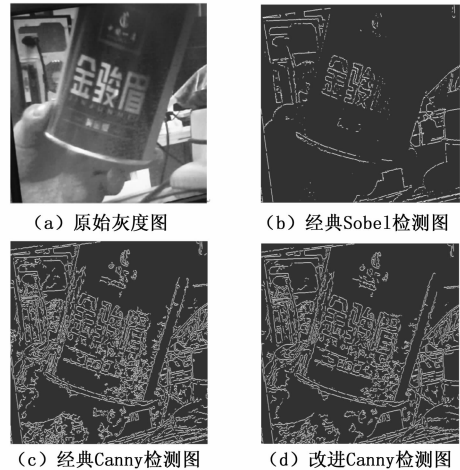


图 7 实验结果

## 6 结论

本文基于 FPGA 实现实时视频边缘检测系统，实现了视频信息的采集、边缘检测以及结果的输出，采用改进的 Canny 边缘检测算法，检测结果精确，利用 FPGA 并行处理的特性，大大提高了系统的运行速度，设计的实时视频边缘检测系统在目标追踪、智能监控领域具有广阔的应用前景。

### 参考文献：

- [1] 黄峰, 易浩. 基于 FPGA 的高速视频实时边缘检测算法设计与实现 [J]. 湖南工程学院学报, 2014, 24 (4): 5-7.
- [2] 陈虎, 凌朝东, 张浩, 等. 基于 FPGA 的实时彩色图像边缘检测算法的实现 [J]. 液晶与显示, 2014, 30 (1): 143-150.
- [3] 韦海萍, 赵保军, 唐林波, 等. Canny 算法的改进及其硬件的实现 [J]. 光学技术, 2006, 32 (2): 263-266.
- [4] 拓小明, 李云红, 刘旭, 等. 基于 Canny 算子与阈值分割的边缘检测算法 [J]. 西安工程大学学报, 2014, 28 (6): 745-749.
- [5] 张辉, 曲仕茹. 基于 FPGA 硬件实现的图像边缘检测及仿真 [J]. 计算机仿真, 2010, 27 (3): 232-236.
- [6] 何文浩, 原魁, 邹伟. 自适应阈值的边缘检测算法及其硬件实现 [J]. 系统工程与电子技术, 2009, 31 (1): 233-237.
- [7] 王纪刚, 陈家新. 基于改进滤波的 Canny 医学图像边缘检测算法 [J]. 计算及测量与控制, 2013, 21 (6): 1577-1579.
- [8] 米林, 马亚洲, 郝建军, 等. 一种基于 Canny 理论的边缘提取改进算法 [J]. 重庆理工大学学报 (自然科学), 2010, 24 (5): 54-58.
- [9] 陈伦海, 黄君凯, 杨帆, 等. 基于 FPGA 的实时边缘检测系统 [J]. 液晶与显示, 2011, 26 (2): 200-204.
- [10] 唐婷婷. 基于 FPGA+SoPC 的视频图像处理系统设计 [J]. 现代电子技术, 2014, 37 (10): 59-63.