

## SOBEL 边缘检测例程

---

### 1 实验简介

本实验将在例程“OV5640 摄像头显示例程”的基础上实现视频图像边缘检测的实验。在很多应用场合，我们只需要采集到图像的棱廓特征的信息，而不需要全部的视频图像，这样就需要用到 SOBEL 边缘检测的算法。

### 2 实验原理

#### 2.1 边缘检测原理和算法

边缘是图像最基本的特征，其在计算机视觉、图像分析等应用中起着重要的作用，这是因为图像的边缘包含了用于识别的有用信息，是图像分析和模式识别的主要特征提取手段。

在图像中，“边缘”指的是临界的意思。一幅图像的“临界”表示为图像上亮度显著变化的地方，边缘指的是一个区域的结束，也是另一个区域的开始。“边缘点”指的是图像中具有坐标  $[x,y]$ ，且处在强度显著变化的位置上的点。

常用的边缘检测算法大多是以原始图像灰度值为基础，通过考察图像的每个像素的某个邻域内灰度的变化，利用边缘一阶或二阶导数的规律来检测边缘。下图左边为原始的黑白灰度的图像，通过边缘检测算法后变成了右边的图像。



实现边缘检测有很多不同的方法，也一直是图像处理中的研究热点，人们期望找到一种抗噪强、定位准、不漏检、不误检的检测算法。其中 Sobel 算子效果较好，边缘检测算法比较简单，实际应用中效率比 canny 边缘检测效率要高，但是边缘不如 Canny 检测的准确，但是很多实际应用的场合，sobel 边缘却是首选，尤其是对效率要求较高，而对细纹理不太关心的时候。本实验就采用 Sobel 的算法来实现视频图像的边缘检测。

## 2.2 sobel 简介

sobel 是一个梯度的计算，如下图所示，是 x 和 y 方向的 3x3 窗口的卷积。

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

Gx

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

Gy

梯度计算公式  $|G| = \sqrt{Gx^2 + Gy^2}$ ，简化的近似计算  $|G| = |Gx| + |Gy|$ 。

对于图像，如下图：P1 到 P9 为 3x3 的 9 个像素点，简化公式计算：

$$|G| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)|$$

$P_1$	$P_2$	$P_3$
$P_4$	$P_5$	$P_6$
$P_7$	$P_8$	$P_9$

3x3 图像窗口

为了进一步简化计算，我们把算子进行简化，调整为如下所示

-1	0	+1
-1	0	+1
-1	0	+1

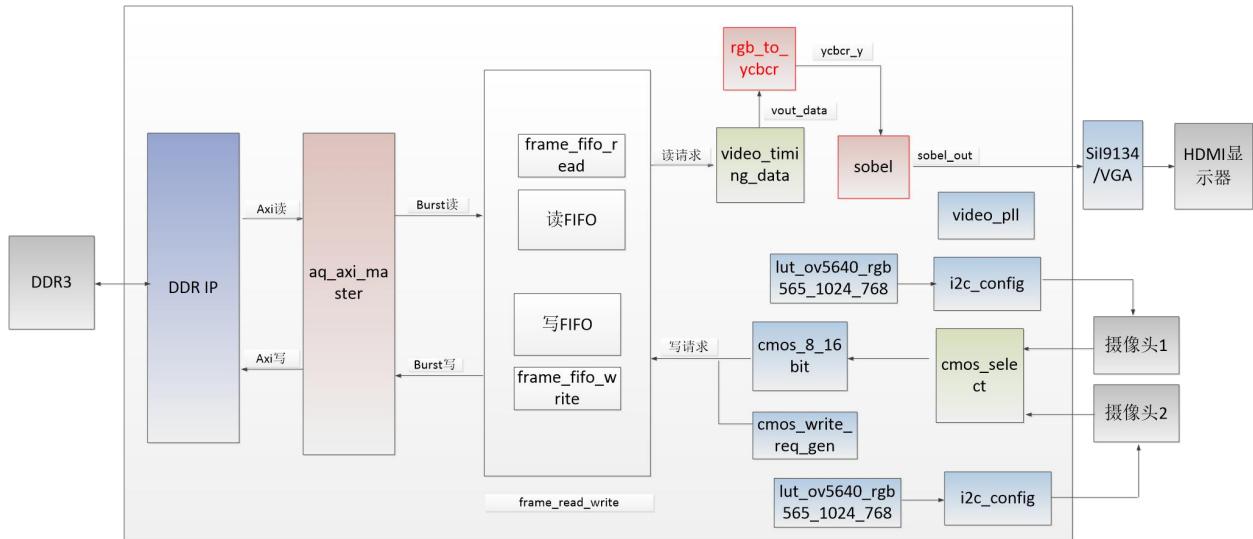
Gx

+1	+1	+1
0	0	0
-1	-1	-1

Gy

### 3 程序设计

本例程是在双目摄像头 HDMI 显示的基础上增加了两个模块，一个是 RGB 转 YCbCr 模块 `rgb_to_ycbcr`, 另一个是边缘检测模块 `Sobel`。因为边缘检测需要的是图像的亮度信息，然后通过亮度阈值判断来显示结果，所以这里需要把 RGB 的图像转换成 YCbCr，然后只取亮度信号进行 `Sobel` 转换。



## rgb\_to\_ycbcr 模块

我们这里需要把 RGB 的图像数据转换成 YUV 信号。关于 YUV 是什么？大家百度一下好了。"Y"表示亮度 (Luminance 或 Luma)，也就是灰阶值；是个基带信号。而"U"和"V"表示的则是色度 (Chrominance 或 Chroma)。在彩色转黑白图像中，我们需要的就是这个亮度的值 Y，把颜色部分去掉，Y 值越大，颜色越白，Y 值越小，颜色越暗。VGA 输出的 RGB 的值都等于这个亮度 Y 的值，VGA 显示的图像就成了黑白图像。

那如何来产生这个 "亮度"Y 信号呢？"亮度"Y 是通过 RGB 输入信号来创建的，方法是将 RGB 信号的特定部分叠加到一起。"色度"则定义了颜色的两个方面-色调与饱和度，分别用 Cr 和 CB 来表示。其中，Cr 反映了 RGB 输入信号红色部分与 RGB 信号亮度值之间的差异。而 CB 反映的是 RGB 输入信号蓝色部分与 RGB 信号亮度值之同的差异。。通过运算，YUV 三分量可以还原出 R (红) , G (绿) , B (兰)。RGB 和 YCbCr 的转换公式如下：

计算公式：

$$Y = 0.183R + 0.614G + 0.062B + 16;$$

$$CB = -0.101R - 0.338G + 0.439B + 128;$$

$$CR = 0.439R - 0.399G - 0.040B + 128;$$

在本次实验中只用到了 Y 分量, CB、CR 分量暂没有用到。得到 Y 分量后, 可以直接对 Y 分量的数据进行 sobel 算法处理, 在实验中添加了一个高斯滤波模块。大家学习过程中也可以去掉此

模块，此模块不是必要模块，高斯滤波对图像邻域内像素进行平滑时，邻域内不同位置的像素被赋予不同的权值，对图像进行平滑的同时，同时能够更多的保留图像的总体灰度分布特征。

首先需要一个  $3 \times 3$  的像素窗口，对 8 位的 Y 分量缓存两行数据后得到 3 行数据，再将 3 行数据分别打两拍就可以轻松实现  $3 \times 3$  的窗口。其计算方法如下所示：计算过程中也是使用流水线方式加快计算。

(x-1, y-1), (x, y-1), (x+1, y-1)	r0_c0, r0_c1, r0_c2
(x-1, y), (x, y), (x+1, y)	<-->   r1_c0, r1_c1, r1_c2
(x-1, y+1), (x, y+1), (x+1, y+1)	r2_c0, r2_c1, r2_c2

高斯滤波公式：  $G(x, y) = (1/16) * (f(x-1, y-1) + 2*f(x, y-1) + f(x+1, y-1) + 2*f(x-1, y) + 4*f(x, y) + 2*f(x+1, y) + f(x-1, y+1) + 2*f(x, y+1) + f(x+1, y+1))$

通过高斯滤波后，也是得到了 8 位的 gauss\_data，最后对这 8 位数据进行边沿检测。

### Sobel 模块

本实验的重点是 sobel 算法的实现，首先需要一个  $3 \times 3$  的像素窗口，利用行缓存缓存两行数据后得到三行数据，在延迟 3 个时钟得到 3 列数据，这样就可以轻松实现  $3 \times 3$  的窗口。

然后按照简化公式，采用绝对值的方式计算 sobel。

```

always@ (posedge pclk or posedge rst)begin
  if(rst)begin
    Gy_temp1 <= 10'd0;
    Gy_temp2 <= 10'd0;
    Gy_data <= 10'd0;
  end
  else begin
    Gy_temp1 <= matrix_p13 + (matrix_p23 << 1) + matrix_p33;
    Gy_temp2 <= matrix_p11 + (matrix_p21 << 1) + matrix_p31;
    Gy_data <= (Gy_temp1 >= Gy_temp2) ? Gy_temp1 - Gy_temp2 :
                                              (Gy_temp2 - Gy_temp1);
  end
end

always@ (posedge pclk or posedge rst)begin
  if(rst)begin
    Gx_temp1 <= 10'd0;
    Gx_temp2 <= 10'd0;
    Gx_data <= 10'd0;
  end
  else begin
    Gx_temp1 <= matrix_p11 + (matrix_p12 << 1) + matrix_p13;
    Gx_temp2 <= matrix_p31 + (matrix_p32 << 1) + matrix_p33;
    Gx_data <= (Gx_temp1 >= Gx_temp2) ? Gx_temp1 - Gx_temp2 :
                                              (Gx_temp2 - Gx_temp1);
  end
end

always@ (posedge pclk or posedge rst)begin
  if(rst)
    Gxy_square <= 21'd0;
  else

```

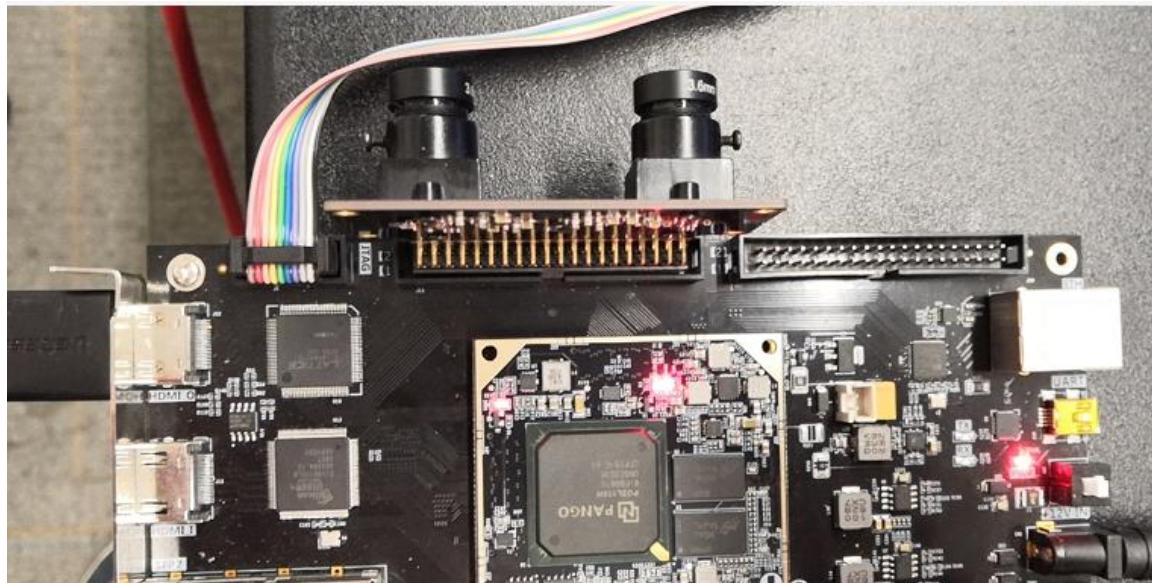
```
Gxy_square <= Gx_data + Gy_data;  
end
```

计算完成以后，要进行简单的二值化处理，将 sobel 值和阈值对比，产生黑白的二值化图像。

```
always@(posedge pclk or posedge rst)begin  
if(rst)  
    sobel_data_r <= 8'b0;  
else if(Gxy_square >= threshold)  
    sobel_data_r <= 8'h00;  
else  
    sobel_data_r <= 8'hff;  
end
```

## 4 实验现象

- (1) 将摄像头模块插入开发板，模块接入扩展口 J13。保证 1 脚对齐，1 脚在焊盘形状和其他引脚是有明显区别的，是方形的。



AN5642 摄像头模块连接图

- (2) 连接好 HDMI 显示器。
- (3) 下载实验程序，可以看到只有边沿信息的黑白视频输出。注意：**ov5640 模块焦距是可调的，如果焦距不合适，图像会模糊，旋转镜头，可以调节焦距。摄像头模块要轻拿轻放，不要用手触摸元器件。**

