作业 1(2022 年 7 月 4 日)

19200300004 黄铭瑞

作业内容:

| 课堂练习 （40分钟，用手机拍下题目）

奥比中光 ORBBEC

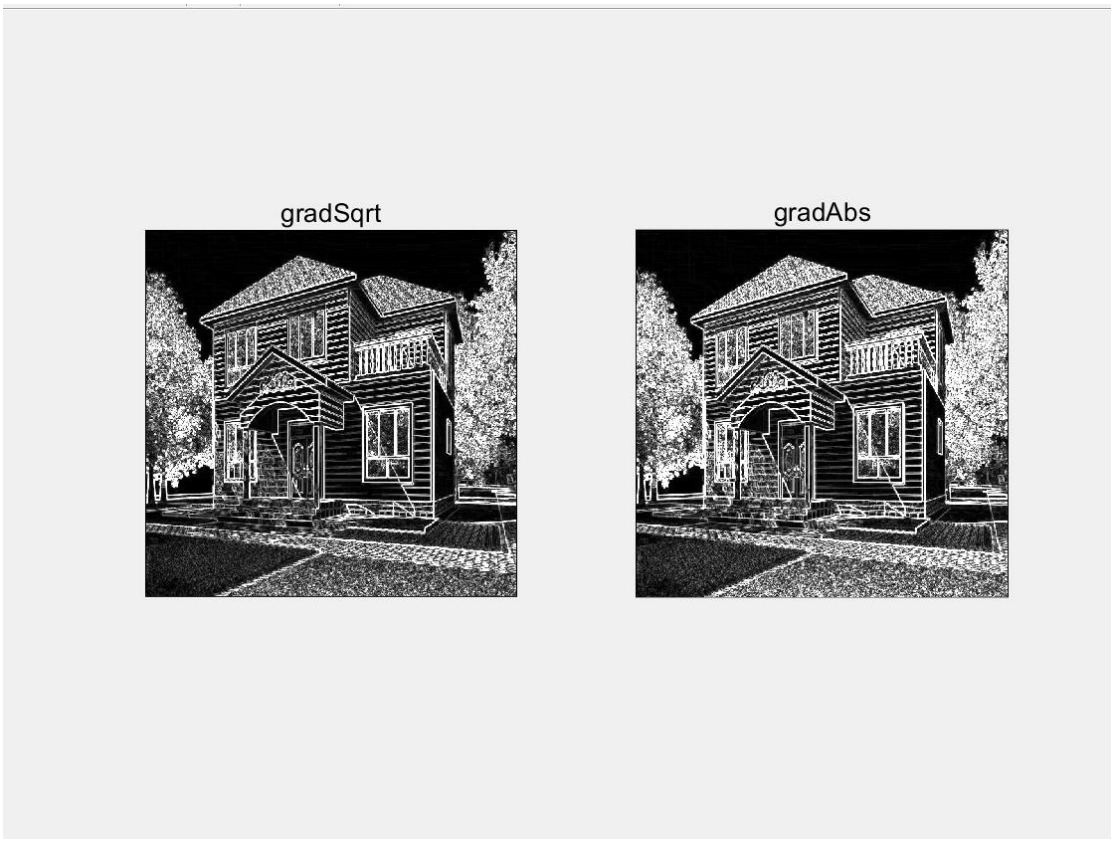
1. 基于house.png，求出x和y方向梯度Gx和Gy，并观察Gy的输出和Gx输出有何不同?
2. 用平方和开根求出矢量梯度(最终边缘图)，并尝试用绝对值法再求一次，并比较两次结果有何异同？能否plot出差异?
3. 观察最终输出的边缘图矩阵的四条边上的像素值是否有效？如果无效，请尝试通过延展原图的方法，使得矩阵的所有数值有效。

答:

1. Gx 的纵向纹理明显,Gy 的横向纹理明显.

(Gx 和 Gy 输出)

2. 绝对值法的得到的纹理要比平方和开方的到的纹理要重一些.



(平方和开方法和绝对值法)

3.可以看到,输出图像边缘的像素值为 0,为无效值.

2	157	186	134	188	0
92	156	100	255	159	0
138	158	216	74	111	0
113	99	113	169	105	0
73	67	119	45	57	0
112	53	67	67	186	0
99	11	65	94	139	0
63	84	19	81	80	0
74	7	92	85	58	0
95	95	75	63	62	0
0	0	0	0	0	0

(部分边缘像素值)

需要对原图进行 padding. 使用 padarray 函数对原图四周进行单行单列的镜像 padding,重复边缘检测操作,最后删除四周空白无效的行和列,恢复原图大小.得到边缘像素有效的输出.

2	157	186	134	188	255
92	156	100	255	159	139
138	158	216	74	111	35
113	99	113	169	105	88
73	67	119	45	57	173
112	53	67	67	186	121
99	11	65	94	139	42
63	84	19	81	80	157
74	7	92	85	58	43
95	95	75	63	62	38
48	52	143	94	62	24

(边缘像素有效)

附录

day1_sobel.m

```
img = imread('house.png'); % 读取图像转为灰度
img = rgb2gray(img);

[high,width] = size(img);   % 获得图像的高度和宽度
pix = double(img);
img_Gx = uint8(zeros(high, width));
img_Gy = uint8(zeros(high, width));
img_out1 = uint8(zeros(high, width));
img_out2 = uint8(zeros(high, width));
for i = 2:high - 1   %sobel 边缘检测
    for j = 2:width - 1
        Gx = (-pix(i-1, j-1) - 2*pix(i, j-1) - pix(i+1, j-1) + pix(i-1, j+1)
        + 2*pix(i, j+1) + pix(i+1, j));
        Gy = (-pix(i-1, j-1) - 2*pix(i-1, j) - pix(i-1, j+1) + pix(i+1, j-1)
        + 2*pix(i+1, j) + pix(i+1, j+1));
        G_sqrt = sqrt(Gx^2+Gy^2);
        G_abs = abs(Gx) + abs(Gy);
        img_Gx(i, j) = Gx;
        img_Gy(i, j) = Gy;
        img_out1(i,j) = G_sqrt;
        img_out2(i,j) = G_abs;
    end
end
% 显示灰度图
figure('Name', 'img_gray');
imshow(img);
title('imgGray');
% 显示 x 梯度和 y 梯度边缘检测图
```

```
figure('name', 'gradx & grady');
subplot(121);
imshow(img_Gx);
title('gradx');
subplot(122);
imshow(img_Gy);
title('grady');
% 显示开平方和绝对值的边沿检测图
figure('name', 'sqrt & abs');
subplot(121);
imshow(img_out1);
title('gradSqrt');
subplot(122);
imshow(img_out2);
title('gradAbs');

% 对原图边界进行扩充
img_padding = padarray(img,[1,1],"symmetric","both");
[h, w] = size(img_padding);
pix2 = double(img_padding);
img_Gx2 = uint8(zeros(h, w));
img_Gy2 = uint8(zeros(h, w));
img_out12 = uint8(zeros(h, w));
img_out22 = uint8(zeros(h, w));
for i = 2:h-1   %sobel 边缘检测
    for j = 2:w-1
        Gx2 = (-pix2(i-1, j-1) - 2*pix2(i, j-1) - pix2(i+1, j-1) + pix2(i-1,
        j+1) + 2*pix2(i, j+1) + pix2(i+1, j));
        Gy2 = (-pix2(i-1, j-1) - 2*pix2(i-1, j) - pix2(i-1, j+1) + pix2(i+1,
        j-1) + 2*pix2(i+1, j) + pix2(i+1, j+1));
        G_sqrt = sqrt(Gx2^2+Gy2^2);
        G_abs = abs(Gx2) + abs(Gy2);
        img_out12(i,j) = G_sqrt;
        img_out22(i,j) = G_abs;
    end
end
% 恢复原图大小
img_out12(all(img_out12==0,2),:) = [];
img_out12(:,all(img_out12==0,1)) = [];
img_out22(all(img_out12==0,2),:) = [];
img_out22(:,all(img_out12==0,1)) = [];
% 显示 padding 后的边缘检测图像
figure('name', 'padding: sqrt & abs');
subplot(121);
```

```
imshow(img_out12);
title('gradSqrt');
subplot(122);
imshow(img_out22);
title('gradAbs');
img_out12
```