

基于 MATLAB 平台的图像边缘检测程序的设计与实现

付 豪, ZY1601119

摘 要: 本文基于 Matlab 平台实现了图像的边缘检测程序的设计与编程。该程序具有打开一副图像, 进行 Sobel 边缘检测与 Canny 边缘检测的功能。本文重点描述了 Sobel 边缘检测与 Canny 边缘检测的原理与源码设计, 最后实验结果显示 Canny 边缘检测更为先进, 检测效果更好。

1 程序设计与实现

1.1 整体架构设计

本文利用 Matlab 进行了图像处理程序的实现。目的在于实现打开一副图像, 进行 Sobel 边缘检测与 Canny 边缘检测的功能。其中程序源码文件各类功能如下:

SobelEdge.m: 打开一副图像, 进行 Sobel 边缘检测, 并输出结果;

CannyEdge.m: 打开一副图像, 进行 Canny 边缘检测, 并输出结果。

1.2 Sobel 边缘检测原理

本程序的 Sobel 边缘检测主要分为读取图像关键帧, 转化为灰度图, 获得图像的高度和宽度, Sobel 边缘检测和输出对比图像等步骤。

边缘检测方法就是考察每个像素在某个领域内灰度的变化, 然后利用边缘临近一阶或二阶方向导数变化规律检测边缘, 即边缘检测局部算法。Sobel 边缘检测采用 Sobel 算子。Sobel 算子如图 1 所示。

| | | | |
|-----|----|----|----|
| (a) | -1 | -2 | -1 |
| | 0 | 0 | 0 |
| | 1 | 2 | 1 |

| | | | |
|-----|----|---|---|
| (b) | -1 | 0 | 1 |
| | -2 | 0 | 2 |
| | -1 | 0 | 1 |

图 1 Sobel 算子

(a) 垂直; (b) 水平;

如图 1 所示, Sobel 算子包括两组 3*3 的矩阵, 图 1 (a) 表示垂直, 图 1 (b) 的表示水平。将它们与图像作平面卷积, 即可分别得出垂直及水平的亮度差分近似值。

最后每一点灰度值大小由下式计算:

$$G = \sqrt{G_x^2 + G_y^2}$$

使用 Sobel 算子进行 Sobel 边缘检测的源码如图 2 所示。

```
f=imread('lenna.jpg');
f=rgb2gray(f);
subplot(1,2,1),imshow(f);
[x,y]=size(f);
a=[-1,0,1;-2,0,2;-1,0,1];
b=[-1,-2,-1;0,0,0;1,2,1];
gradx=imfilter(f,a,'same');
grady=imfilter(f,b,'same');
grad=gradx+grady;
subplot(1,2,2),imshow(grad)
```

图 2 Sobel 边缘检测的源码实现

1.3 Canny 边缘检测原理

canny 边缘检测, 相对于 sobel 是较为复杂的一种检测算法, 也是迄今为止最优秀的一种边缘检测算法, 同时, 它也带来了大量的运算。

本程序的 Canny 边缘检测分为四步:

第一步, 对灰度图进行高斯滤波。使用二维高斯公式, 产生高斯滤波模版。然后用模版和图像进行卷积。

第二步, 求出梯度的幅值图像和角度图像。

第三步, 对梯度幅值图形进行非最大值抑制, 并进行双阈值处理。利用求出的梯度角度图, 把各个梯度分别划分到水平, 垂直, 45°和-45°等四个方向上。然后把赋值图按所划分的四个方向, 分别同邻近的两个点比较, 若为最大值则保留, 若不是, 则变为 0; 这样就得到非最大值抑制图, 然后设置一个大阈值和一个小阈值, 再用小阈值图减去大阈值图。得到阈值图。

第四步, 连接分析来检测并连接边缘。在大阈值图中定位下一个非 0 像素 P; 在小阈值图中用 8 连通的方式, 连接到 P, 如此递归, 连接所有连线。

进行 Canny 边缘检测的源码如图 3 所示。

```

clear;
img = imread('lena.jpg');
f = rgb2gray(f);
subplot(2,2,1), imshow(f);
% 2. 边缘检测
% 2.1. 灰度阈值
img = im2bw(f);
% 2.2. 高斯滤波
img = imfilter(img, 'gaussian', 'replicate');
% 2.3. 拉普拉斯算子
img = imfilter(img, 'laplace', 'replicate');
% 2.4. 边缘检测
% 2.4.1. 梯度计算
[gy, gx] = gradient(img);
% 2.4.2. 梯度幅值
mag = sqrt(gx.^2 + gy.^2);
% 2.4.3. 梯度方向
dir = atan2(gy, gx);
% 2.4.4. 非极大值抑制
for i = 1:size(mag,1)
    for j = 1:size(mag,2)
        if (dir(i,j) >= 22.5 && dir(i,j) <= 67.5) || (dir(i,j) >= 157.5 && dir(i,j) <= 202.5) || (dir(i,j) >= 337.5 && dir(i,j) <= 382.5)
            % 非极大值抑制
            if (mag(i,j) > mag(i,j-1) && mag(i,j) > mag(i,j+1)) || (mag(i,j) > mag(i-1,j) && mag(i,j) > mag(i+1,j))
                % 保留
            else
                % 抑制
                mag(i,j) = 0;
            end
        end
    end
end
% 2.5. 边缘连接
mag = mag * 0.5;
% 2.6. 边缘检测结果
img = mag;
subplot(2,2,2), imshow(img);

```

图 3 Canny 边缘检测的源码实现

如图5 (b) 所示, Canny算子能够检测到更为完整的边缘, 定位精度也高, 相比较Sobel算子而言, 对于高噪声图像的边缘检测, Canny算子的检测效果应该会最好。但目前本程序的Canny边缘检测在非最大值抑制时的细化和最后的连接上还有改进的余地。

3 结 论

(1) 本文基于 Matlab 平台实现了图像的边缘检测程序的设计与编程。该程序具有打开一副图像, 进行 Sobel 边缘检测与 Canny 边缘检测的功能。

(2) 本文重点描述了 Sobel 边缘检测与 Canny 边缘检测的原理与源码设计, 最后实验结果显示 Canny 边缘检测更为先进, 检测效果更好。

2 实验结果

2.1 Sobel边缘检测结果展示

Sobel 边缘检测前后图像如图 4 所示。



图 4 Sobel 边缘检测结果展示

(a) 原图像; (b) Sobel 边缘检测后图像;

如图4 (b) 所示, Sobel算子采用加权滤波对图像边缘进行检测, 虽有较好的抗噪性, 但产生了宽边缘, 间断点较多, 其适用于灰度渐变、低噪声图像的边缘检测。

2.2 Canny边缘检测结果展示

Canny 边缘检测前后图像如图 5 所示。



图 5 Canny 边缘检测结果展示

(a) 原图像; (b) Canny 边缘检测后图像;