

数字图像分析第二次实验报告

院 系： 信息科学技术学院

学 号： SA19023005

姓 名： 景军元

授课教师： 李厚强、周文罡

课程助教： 欧阳剑波、刘一衡

实验名称： Canny 边缘检测算法

一、基本原理

1、最优边缘检测算子（Canny 算子）

好的边缘检测算子应具有三个指标：

（1）低失误概率

既要少将真正的边缘丢失也要少将非边缘判为边缘。

（2）高位置精度

检测出的边缘应在真正的边界上。

（3）对每个边缘有唯一的响应

得到的边界为单像素宽。

2、算法步骤

Canny 算子可以用高斯函数的一阶微分算子来近似，具体做法为：

（1）高斯平滑及差分

①先对图像进行高斯卷积平滑滤波，然后计算梯度，由于卷积算子和差分算子均为线性算子，可以交换运算顺序，得到：

$$S = \nabla(I * h) = \nabla(h * I) = (\nabla h) * I = \begin{bmatrix} h_x \\ h_y \end{bmatrix} * I = \begin{bmatrix} h_x * I \\ h_y * I \end{bmatrix}$$

其中高斯滤波器： $h = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$

高斯差分（DOG）滤波器： $h_x = \frac{\partial h}{\partial x}, h_y = \frac{\partial h}{\partial y}$

②图像梯度幅值： $G = \sqrt{(h_x * I)^2 + (h_y * I)^2}$

③图像梯度方向角： $\phi(x, y) = \arctan(G_y/G_x)$

（2）非极大值抑制

沿着梯度方向，抑制梯度值非最大的点，细化幅度图像中的屋脊带，只保留幅度局部变化最大的点，NMS 通过抑制梯度线上所有非屋脊峰值的幅值来细化边缘。

主要有以下两种方法：

①基于最近邻进行最大值消除

将梯度角 $\theta[i, j]$ 的变化范围分为四个扇区

$$\zeta[i, j] = \text{Sector}(\theta[i, j])$$

用 3×3 领域作用于幅值图像 $M[i, j]$ ，领域中心像素 $M[i, j]$ 与沿着梯度线方向的两个像素进行比较，若 $M[i, j]$ 不比沿梯度线方向的两个相邻点幅值大，则像素 $M[i, j]$ 被抑制， $M[i, j]$ 被置为 0。

②用插值进行最大值消除

通过对相邻单元的梯度幅值的插值估计梯度线上的相邻幅值，如果 P 点的梯度值小于 S_1 或 S_2 的梯度值，则 P 点的梯度值被置为 0。

$$G_{S_1} = (1 - d)G_{P_1} + dG_{P_2}$$

该方法精确但计算量大。

（3）双阈值算法

①双阈值算法采用两个阈值 τ_1 和 τ_2 ，且 $\tau_2 \approx 2\tau_1$ ；

②得到两个阈值边缘图像 $T_1[i, j]$ 和 $T_2[i, j]$ ；

- ③ $T_2[i,j]$ 含有的假边缘少，但有间断点；
- ④以 $T_2[i,j]$ 为指导，在 $T_1[i,j]$ 中相应的 8 邻域点寻找可以连接到轮廓上的点；
- ⑤不断在 $T_1[i,j]$ 收集边缘，直到将 $T_2[i,j]$ 中所有的间隙连接起来为止。

二、实验内容

1、问题重述

实现 Canny 边缘检测算法。

2、函数实现

(1) 主函数

①源代码

```
clear; close all;
filename = char('lenna.jpg','chip.tif','cameraman.jpg','airport.tif');
filedir = './test images/';
for i=1:4
    image = imread([filedir filename(i,:)]);
    if size(image,3)~=1
        image = rgb2gray(image);
    end
    [G,fai] = smooth_diff(image);
    result = NMS(G,fai);
    edge = doubleleth(result);
    figure(1);
    subplot(2,2,i);
    imshow(edge,[]);
end
```

②函数说明

对于每个图像，读入图像后检查是否为灰度图像，如果不是则转化为灰度图像，然后依次进行高斯平滑及差分操作、非极大值抑制和双阈值算法，得到输出结果进行显示。

(2) 高斯平滑及差分

①源代码

```
function [G,fai] = smooth_diff(image)
image = im2double(image);
sigma = 1;
tempsize = 3;
[M,N] = size(image);%图片尺寸
%差分模板
[y,x] = meshgrid(-(tempsize-1)/2:(tempsize-1)/2,-(tempsize-1)/2:(tempsize-1)/2);
hx =
1/sqrt(2*pi*sigma)*(exp(-(x+1).^2+y.^2)/(2*sigma^2))-exp(-(x-1).^2+y.^2)/(2*sigma^2));
hy =
1/sqrt(2*pi*sigma)*(exp(-(x.^2+(y+1).^2)/(2*sigma^2))-exp(-(x.^2+(y-1).^2)/(2*sigma^2)));
%Padding
```

```

image = [zeros(M,(tempsize-1)/2),image,zeros(M,(tempsize-1)/2)];
image = [zeros((tempsize-1)/2,N+tempsize-1);image;zeros((tempsize-1)/2,N+tempsize-1)];
hxl = zeros(M,N);
hyl = zeros(M,N);
%卷积
for i=1:M
    for j=1:N
        hxl(i,j) = sum(sum(hx.*image(i:i+tempsize-1,j:j+tempsize-1)));
    end
end
for i=1:M
    for j=1:N
        hyl(i,j) = sum(sum(hy.*image(i:i+tempsize-1,j:j+tempsize-1)));
    end
end
G = sqrt((hxl).^2+(hyl).^2);
fai = atan(hyl./hxl);
end

```

②函数说明

该函数首先设定相关参数，给出高斯平滑差分模板，然后将原图像进行补零操作，这样使得输出图像的大小和原图像大小一致。然后利用高斯平滑差分模板对补零后的图像进行卷积操作，得到沿 x 方向和沿 y 方向的梯度值，进一步求出梯度幅值和梯度角。

(3) 非极大值抑制

①源代码

```

function result = NMS(G,fai)
[M,N] = size(G);
result = G;
for i=2:M-1
    for j=2:N-1
        if abs(fai(i,j)-0)<=pi/8
            if result(i,j)<result(i-1,j) || result(i,j)<result(i+1,j)
                result(i,j)=0;
            end
        elseif abs(fai(i,j)-pi/4)<=pi/8
            if result(i,j)<result(i+1,j+1) || result(i,j)<result(i-1,j-1)
                result(i,j)=0;
            end
        elseif abs(fai(i,j)+pi/4)<=pi/8
            if result(i,j)<result(i+1,j-1) || result(i,j)<result(i-1,j+1)
                result(i,j)=0;
            end
        else
            if result(i,j)<result(i,j-1) || result(i,j)<result(i,j+1)
                result(i,j)=0;
            end
        end
    end
end

```

```

        end
    end
end
end
end

```

②函数说明

采用基于最近邻的方法进行极大值消除，比较梯度角和四个扇区的中心角，梯度角距离哪个扇区的中心角最近，则认为梯度方向即为该扇区的中心角。将中心像素和梯度方向上的像素值进行比较，小于任何一个像素值即置为 0。

（4）双阈值算法

①源代码

```

function result = doubleth(image)
[M,N] = size(image);
maxval = max(max(image));
tau1=maxval/15;
tau2=2*tau1;
T1 = image>tau1;
T2 = image>tau2;
result = T2;
iter = 100;    %设置迭代次数，由于算法按行扫描而非按边缘方向扫描，单次运行会有边缘的遗漏
for k=1:iter
    for i=2:M-1
        for j=2:N-1
            if result(i,j)==true
                result(i-1:i+1,j-1:j+1)=result(i-1:i+1,j-1:j+1) | T1(i-1:i+1,j-1:j+1);
            end
        end
    end
end
end
figure(1)
subplot(2,2,2);
imshow(T1,[]);
subplot(2,2,3);
imshow(T2,[]);
subplot(2,2,4);
imshow(result,[]);
end

```

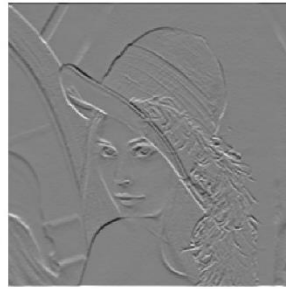
②函数说明

根据图片像素的极大值设置两个阈值，之后采用双阈值算法，以 T2 为指导在 T1 中收集点，最终得到所有轮廓。其中参数的设置可以再做进一步的调整使得效果更好，这里简单设置。由于单次计算不能完整得到所有边缘，因此进行多次迭代。

三、实验结果及分析

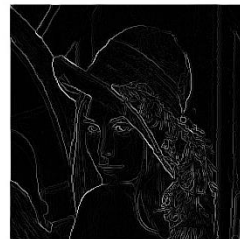
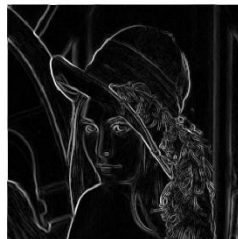
1、高斯平滑及差分

如图所示，四张图从左到右，从上到下分别是： I 、 $h_x * I$ 、 $h_y * I$ 、 G 。



2、非极大值抑制

如图所示，从左到右依次是：原图、梯度幅值图、非极大值抑制结果图。



3、双阈值算法

如图所示，从左到右，从上到下依次是：原图、弱边缘检测结果、强边缘检测结果、最后边缘检测结果图。



4、最终检测结果

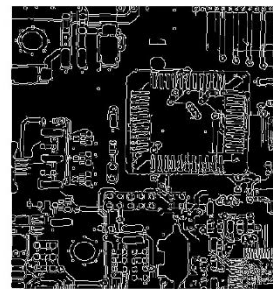
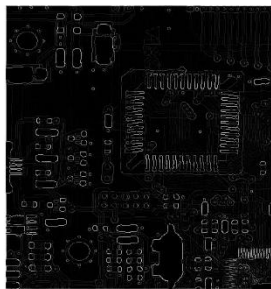
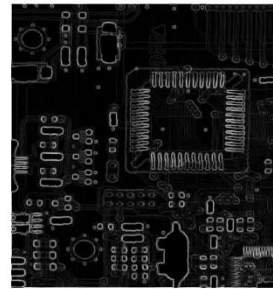
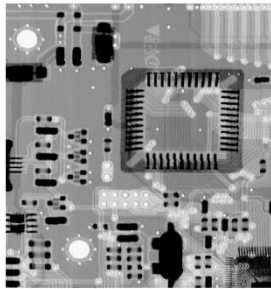
如图所示依次为 lenna.jpg、cameraman.jpg、chip.tif、airport.tif 的检测结果图。每个检测结果图中，从左到右，从上到下依次是：原图、梯度幅值、非极大值抑制结果、最终边缘检测结果。可以看到，chip.tif 的检测效果并不是很好，这可以通过调整双阈值来缓解。



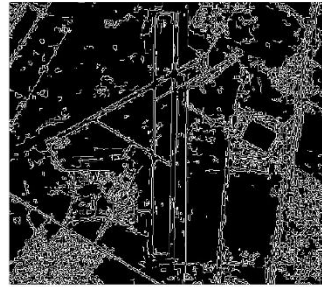
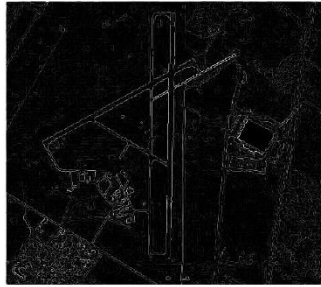
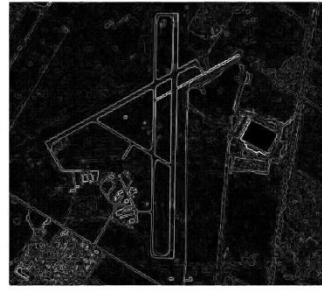
lenna.jpg



cameraman.jpg



chip.tif



airport.tif

四、实验收获

- 1、了解了 Canny 边缘检测方法的全部步骤。
- 2、加深了对于边缘检测、卷积等方法的理解。
- 3、加强了 MATLAB 编程的能力。