

***SCHOOL OF ELECTRONIC ENGINEERING***

电 子 工 程 学 院

**《图 像 分 析 与 处 理》**

**实验指导**

西安邮电大学

电 子 工 程 学 院

2025年5月

**实验四**  **基于Matlab的图像边缘检测与处理**

一、实验目的与要求

1．掌握图像边缘检测的主要原理和常用方法。

2．掌握几种常用边缘检测算子的Matlab程序设计。

3. 使学生掌握各种边缘检测算子的特征。

4. 掌握*Hough*变换的算法本质和程序设计技巧。

5. 利用Matlab软件编程实现*Hough*变换提取直线。

二、实验原理及知识点

1、边缘检测

图像理解是图像处理的一个重要分支，研究为完成某一任务需要从图像中提取哪些有用的信息，以及如何利用这些信息解释图像。边缘检测技术对于处理数字图像非常重要，因为边缘是所要提取目标和背景的分界线，提取出边缘才能将目标和背景区分开来。在图像中，边界表明一个特征区域的终结和另一个特征区域的开始，边界所分开区域的内部特征或属性是一致的，而不同的区域内部的特征或属性是不同的，边缘检测正是利用物体和背景在某种图像特性上的差异来实现的，这些差异包括灰度，颜色或者纹理特征。图像边缘检测主要用于增强图像中的轮廓边缘、细节以及灰度跳变部分，形成完整的物体边界，达到将物体从图像中分离出来或将表示同一物体表面的区域检测出来的目的。目前为止最通用的方法是检测亮度值的不连续性，用一阶和二阶导数检测的。图像边缘检测必须满足两个条件：一能有效地抑制噪声；二必须尽量精确确定边缘的位置。

由于噪声和模糊的存在，检测到的边界可能会变宽或在某些点处发生间断，因此，边界检测包括两个基本内容：首先抽取出反映灰度变化的边缘点，然后剔除某些边界点或填补边界间断点，并将这些边缘连接成完整的线。边缘检测的方法大多数是基于方向导数掩模求卷积的方法。导数算子具有突出灰度变化的作用，对图像运用导数算子，灰度变化较大的点处算得的值比较高，因此可将这些导数值作为相应点的边界强度，通过设置门限的方法，提取边界点集。

（1）微分法

微分法的目的是利用微分运算求信号的变化率，加强高频分量的作用，从而使轮廓清晰。遵循如下两个基本准则之一：找到亮度的一阶导数在幅度上比指定的阈值大的地方；找到亮度的二阶导数有零交叉的地方。

1. 差分边缘检测方法

利用像素灰度的一阶导数算子在灰度迅速变化处得到高值来进行奇异点的检测。它在某一点的值就代表该点的边缘强度，通过对这些值设置阈值来进一步得到边缘图像。差分边缘检测方法是最原始、最基本的方法。但要求差分方向与边缘方向垂直，这就需要对图像的不同方向（一般为垂直方向、水平方向和对角线方向）都进行差分运算，增加了实际运算的繁琐性，目前很少采用。

1. *Roberts*边缘检测算子

*Roberts*边缘检测算子根据任意一对互相垂直方向上的差分可用来计算梯度的原理，采用对角线方向相邻两像素之差，即：



然后根据式（*1.1*）计算出*Roberts*的梯度幅度值：



它们的卷积算子为：



*Roberts*检测器较为简单，但具有一些功能上的限制，例如，它是非对称的，而且不能检测诸如*45°*倍数的边缘。然而，它还是经常用于硬件实现中，因为它既简单又快速。

1. *Sobel*边缘检测算子

对数字图像的每个像素，考察它上下左右邻点灰度的加权差，与之接近的邻点的权大。据此，定义*Sobel*算子如下：

卷积算子为：



*Sobel*算子很容易在空间上实现，边缘检测效果较好，且受噪声的影响也较小。邻域增大抗噪性会更好，但计算量也会增大，得出的边缘也会相应变粗。*Sobel*算子会检测出许多伪边缘，边缘定位精度不够高，在精度要求不高时是一种较常用的边缘检测方法。

1. *Prewitt*边缘检测算子

*Prewitt*边缘检测算子模板如下：



图像中的每个像素都用这两个核作卷积，一个核对垂直边缘影响最大，另一个核对水平边缘影响最大。两个卷积的绝对值的最大值作为该点的输出值。不能简单的将小于*0*的值处理为*0*，这样会丢失信息。它比*Sobel*检测器在计算上要简单一些，但比较容易产生一些噪声。

1. 拉普拉斯边缘检测算子

拉普拉斯边缘检测算子是一种二阶微分算子，与其它边缘检测方法的不同之处在于，该方法是一种各向同性的检测方法，即其边缘的增强程度与边缘的方向无关，从而可以满足不同走向的边缘锐化的要求。

定义如下：



几种常用的实现*Laplace*运算的检测模板如下：



拉普拉斯算子自身很少被直接用作边缘检测，因为二阶导数对噪声具有无法接受的敏感性，它的幅度会产生双边缘，而且它不能检测边缘的方向。然而，当与其它边缘检测技术组合使用时，拉普拉斯算子是一种有效的补充方法。例如，虽然它的双边缘使得它不适合直接用于边缘检测，但该性质可用于边缘定位。

1. *Laplacian of a Gaussian(LoG)*检测器

高斯函数



其中，*r2=x2+y2*，*σ*是标准偏差。这是一个平滑函数，若和一幅图像卷积，则会使图像变模糊。模糊程度由*σ*的值决定。该函数的拉普拉斯算子为：



求二阶导数是线性运算，所以用对图像进行卷积与先用平滑函数对图像卷积再计算结果的拉普拉斯算子是一样的。这是*LoG*检测器最关键的概念，用对图像卷积会产生两个效果：是图像变平滑（从而减少噪声）；计算拉普拉斯算子，以便产生双边缘图像。然后，定位边缘就是找到两个边缘之间的零交叉。

1. *Canny*边缘检测器

*Canny*检测器的效果较好，更适合用于检测真正的弱边缘。*Canny*算子给出了一个好的边缘检测算子的*3*个指标：低失误率，即真正的边缘点尽可能提取，非边缘点尽可能不提取；高位置精度，检测的边缘应尽可能接近真实边缘；对每一个边缘点有唯一的响应，即得到单像素宽度的边缘。

1. 各种算子的Matlab实现及效果比较

在*MATLAB*中可以由*edge*函数实现各算子对边缘的检测，其调用格式为：

BW = edge(I,’METHOD’)—自动选择阈值用指定的算子进行边缘检测；

BW = edge(I,’METHOD’,THRESH)—根据给定的阈值*THRESH*用指定的算子进行边缘检测，忽略所有小于阈值的边缘。当*THRESH*为空时，自动选择阈值；

[BW, THRESH] = edge(I,’METHOD’,…)—返回*edge*使用的阈值，以确定哪个梯度值足够大到可以称为边缘点。其中*I*为输入图像。返回图像*BW*为与I同大的二值图像，*1*表示边缘，*0*表示非边缘。*I*是*unit8*型、*unit16*型，或者是*double*型，*BW*是*unit8*型。

*METHOD*：使用检测算子的类型，经常使用的有：

*sobel：*缺省值，用导数的*sobel*近似值检测边缘，那些梯度最大的点返回边缘。

*roberts：*用导数的*roberts*近似值检测边缘，那些梯度最大的点返回边缘。

*prewitt：*用导数的*prewitt*近似值检测边缘，那些梯度最大的点返回边缘。

*LoG：*用*LoG*算子检测边缘。

*Canny：*用*Canny*算子检测边缘。

*THRESH：*指定的阈值，所有不强于*thresh*的边缘都被忽略。

使用几种典型边缘检测算子得到的边缘图像如图*1.1*所示。



**(a) 原图像 (b) *Sobel*检测的边缘**



**(c)*Roberts*检测的边缘 (d) *Prewitt*检测的边缘**



**(e) *LoG*检测的边缘 (f) *Canny*检测的边缘**

**图*1.1***  使用几种典型边缘检测算子得到的边缘图

由图*1.1*可见，*Canny*算子的检测结果明显优于其它几个检测算子，除了检测到要求的特征外，还产生了最清晰的映射。

2、边缘连接

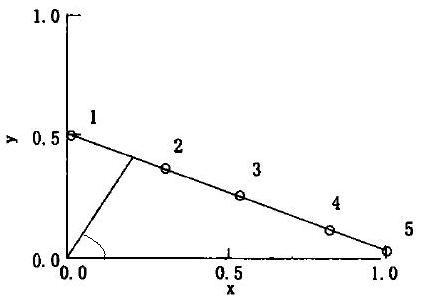
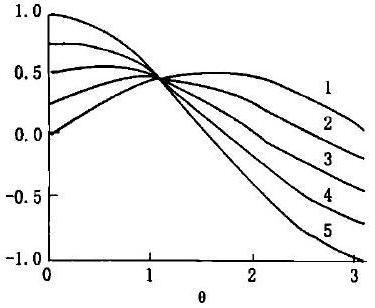
*Hough*变换是图像处理中从图像中识别几何形状的基本方法之一。*Hough*变换的基本原理在于利用点与线的对偶性，将原始图像空间的给定的曲线通过曲线表达形式变为参数空间的一个点。这样就把原始图像中给定曲线的检测问题转化为寻找参数空间中的峰值问题。也即把检测整体特性转化为检测局部特性。比如直线、椭圆、圆、弧线等。

1. *Hough*变换检测直线的原理

*Hough*变换的核心是两个坐标系的变换。总所周知，一个点或者一个线段在不同坐标系下的表现形式是不同的。图像空间中一个像素的位置坐标对应Hough空间(Hough变换以后的空间,极坐标空间较常用)中的一条曲线,即直角坐标系中的点(x,y)与极坐标系中的正弦曲线相对应。



如果图像空间中*n*个像素共一直线,那么它们*Hough*空间中的*n*条正弦曲线有一个公共交点如图*2.1*所示。把一幅图像中的所有有效像素(如二值化后的黑色像素)均对应到*Hough*空间,在*Hough*空间通过某交点的正弦曲线越多,此点所对应的图像空间的直线存在的可能性越大。将有较多正弦曲线通过的交点称做兴趣点*(Point of Intereste, POI),Hough*空间的兴趣点越多说明图像空间的直线越多。在*Hough*空间中对通过兴趣点的正弦曲线数进行简单的累加统计操作来完成直线检测任务。

(a) 图像空间的点 (b) 参数空间的正弦曲线

图2.1 *Hough*变换检测直线

图*2.1(a)*中*ρ*为从原点到直线的垂直距离，*θ*为从*x*轴算起的角度，这条直线在*ρ-θ*平面中为一点，见图*2.1(b)*。

1. *Hough*变换检测直线的算法步骤

①图像预处理，一般*Hough*变换只适应于黑白图像，即*Hough*变换之前应首先进行图像边缘提取；

②初始化一个*ρ、θ*平面的数组。一般*ρ*方向上的量化数目为对角线方向像素数，*θ*方向上的量化间距为*2°*；

③顺序搜索图像的所有黑点，对每一个黑点，按式*(2.1)*计算*ρ、θ*取不同值，分别将对应的数组元素加*1*；

④求出数组中的最大值并记录对应的*ρ、θ*；

⑤绘出*ρ、θ*对应的直线。

*Hough*变换的基本策略是：由图像空间中的边缘数据点去计算参数空间中的参考点的可能轨迹，并在一个累加器中给计算出的参考点计数，最后选出峰值。

在*Matlab*中，使用*Hough*函数计算二值图像的标准*Hough*变换。调用格式为：

[H, THETA, RHO] = HOUGH(BW);

[H, THETA, RHO] = HOUGH(BW,PARAM1,VAL1,PARAM2,VAL2);

其中*BW*是作过边缘提取的二值图像，*H*是*Hough*变换后得到的矩阵，*THETA、RHO*分别对应*Hough*变换矩阵的每一列和每一行的*θ*值和*ρ*值。

*PARAM1*即为*'ThetaResolution'*，它定义了*Hough*变换矩阵沿着*theta*轴的间隔，缺省值为*1*；

*PARAM2*即为*'RhoResolution'*，定义了*Hough*变换矩阵沿着*rho*轴的间隔，缺省值为*1。*

使用*Hough*变换进行线检测和链接的第一步是峰值检测。因为数字图像的空间中的量化、*Hough*变换的参数空间中的量化以及典型图像的边缘都不是很直这个事实，*Hough*变换的峰值一般都位于多个*Hough*变换单元中。

在*Matlab*中，函数*houghpeaks*可用于在*Hough*变换中找到一组有意义的明显峰值。调用格式为：

PEAKS = HOUGHPEAKS(H,NUMPEAKS);

PEAKS = HOUGHPEAKS(...,PARAM1,VAL1,PARAM2,VAL2);

其中*NUMPEAKS*规定了能够识别的最多峰值数，缺省值为*1；PEAKS*是一个*Q×2*的矩阵，其中*Q*的取值范围*0-NUMPEAKS*。

*PARAM1*即为*'Threshold'*，它是非负参数，缺省值为*0.5\*max(H(:))*，*H*矩阵中高于*Threshold*的值才被认为是峰；

*PARAM2*即为*'NHoodSize'，*它是一个二元向量*[m,n]*，取值为正奇数，定义了峰值附近的邻域尺寸，当峰值确定后，每一个峰值附近的邻域值被设为*0*，缺省值取最小的奇数值大于或等于*size(H)/50*。

找到的位置相关的像素用函数*houghlines*链接线段，调用格式为：

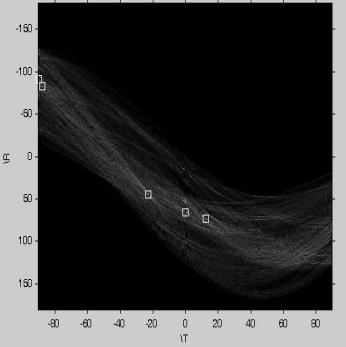
LINES = HOUGHLINES(BW, THETA, RHO, PEAKS);

LINES = HOUGHLINES(...,PARAM1,VAL1,PARAM2,VAL2);

其中*PARAM1*即为*'FillGap'*，它是正实数，缺省值为*20* ，当两线段间隔小于*'FillGap'*数值时，被合并在一起；

*PARAM2*即为*'MinLength'*，它是正实数，缺省值为*40* ，融合后线段长度小于*'MinLength'*的被舍弃。

在这个例子中，使用函数*hough, houghpeaks*和*houghlines*来寻找图像中的一组线段。注：图像须首先边缘提取得到二值图才能进行*hough*变换，结果如图*2.2*。

(a) 原始图像 (b) 带有所选*5*个峰值位置 (c) *Hough*变换峰值

的*Hough*变换 对应的线段

图2.2 使用*Hough*变换作线检测检测和连接

图*2.2(b)*显示了带有*5*个看起来很明显峰值位置重叠的*Hough*变换。图*2.2(c)*显示了结果图像，其中检测到的线段叠加为绿色的粗线。

*Hough*变换运算量太大，且不考虑各点之间的距离信息，易于将不属于直线的点也连接到直线上，产生所谓过连接。实际中，通常可把图像分成小块，对各块图像利用*Hough*变换提取直线，然后将各直线连接起来，或增加变换过程中的遍历步径，来减少运算量。

三、实验内容及步骤

1．启动*Matlab*程序，设置*Matlab*的程序组中的当前活动文件夹；

2．利用*imread( )*函数读取一幅图像，将图像数据存入一个数组中；

3．利用*imshow( )*函数来显示图像；

4．编写*Matlab*程序，利用*edge*函数实现*Sobel、roberts*和*prewitt*边缘检测算子，显示图像边缘检测结果；

5．编写*Matlab*程序，利用*edge*函数实现*log*和*Canny*边缘检测算子，显示图像边缘检测结果；

6．分析比较不同边缘检测算子的检测效果；

7．利用*Matlab*编程实现对图像提取边缘，得到边缘二值图；

8．利用相应的*Matlab*函数对该二值图像进行*Hough*变换，提取峰值点，连接峰值点得到线段；

9．将以上每一步的函数执行语句拷贝下来，写入实验报告，将得到的对应图像结果拷贝下来放入实验报告，并进行结果分析。

四、考核要点

1. 熟悉在Matlab中如何完成不同边缘检测算子的图像边缘检测。

2. 通过对比分析不同边缘检测算子的检测结果掌握不同算子的检测性能。

3. 熟悉在Matlab中如何实现基于*Hough*变换的直线提取。

五、实验报告要求

描述实验目的、实验原理及实验基本步骤，用数据和图像给出各个步骤中取得的实验结果，并进行相应的数据分析，就实验中遇到的问题及解决方法进行描述，回答思考题，给出实验总结及程序清单。

六、思考题

（1）边缘的方向是什么意思？为什么要考虑边缘的方向？

（2）*Hough*变换原理是什么？

七、实验图像

Fig.1 house.jpg Fig.2 M\_M.bmp

八、参考代码

**1、**由*edge*函数实现各算子对图像的边缘检测

clear all;

I = imread('house.jpg');

I=rgb2gray(I);

BW1 = edge(I,'sobel'); %利用Sobel算子进行边缘检测

BW2 = edge(I,'roberts'); %利用roberts算子进行边缘检测

BW3 = edge(I,'prewitt'); %利用prewitt算子进行边缘检测

BW4 = edge(I,'log'); %利用log算子进行边缘检测

BW5 = edge(I,'canny'); %利用canny算子进行边缘检测

subplot(2,3,1),imshow(I),title(‘原图’)

subplot(2,3,2),imshow(BW1),title(‘sobel’)

subplot(2,3,3),imshow(BW2),title(‘roberts’)

subplot(2,3,4),imshow(BW3),title(‘prewitt’)

subplot(2,3,5),imshow(BW4),title(‘log’)

subplot(2,3,6),imshow(BW5),title(‘canny’)

2、使用*Hough*变换作线检测和连接

clear all;

RGB = imread('M\_M.bmp');

I=RGB;

I = rgb2gray(RGB);

BW = edge(I,'canny'); % 利用Canny算子提取图像边缘

[H,T,R] = hough(BW,'RhoResolution',0.5,'ThetaResolution',0.5);

figure(1),

imshow(H,[ ],'XData',T,'YData',R,'InitialMagnification','fit'),

axis on, axis normal

xlabel('\theta');ylabel('\rho');

p = houghpeaks(H,5,'threshold',ceil(0.3\*max(H(:))));

%找到5个较明显的Hough变换峰值

hold on

plot(T(p(:,2)),R(p(:,1)),'s','color','white');

lines = houghlines(BW,T,R,p,'FillGap',10,'MinLength',10);

%查找并链接线段

figure, imshow(BW), hold on %在二值图中叠加显示这些线段

for k = 1:length(lines)

xy = [lines(k).point1; lines(k).point2];

plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

end

3、旋转图像利用*Hough*变换检测直线

I=imread('M\_M.bmp');

I=rgb2gray(I);

rotI=imrotate(I,33,'crop'); %旋转图像并用canny寻找边缘

BW=edge(rotI,'canny');

imshow(BW);

title('原图');

%对图像进行Hough变换

[H,T,R]=hough(BW); %[H,theta,rho]执行hough变换并显示hough矩阵

%显示变换域

%figure,imshow(imadjust(rescale(H)),'XData',T,'YData',R,...

% 'InitialMagnification','fit');

figure,imshow(H,[],'XData',T,'YData',R,...

'InitialMagnification','fit');

xlabel('\theta');ylabel('\rho');

axis on,axis normal,hold on

title('变换域');

%计算变换域峰值，寻找矩阵中前5个大于最大值0.3倍的峰值，并按行、列索引转换成实际坐标，并在矩阵图像中标出峰值位置

P=houghpeaks(H,5,'threshold',ceil(0.3\*max(H(:))));

x=T(P(:,2));y=R(P(:,1));

plot(x,y,'s','color','red');

%找到并绘制直线

lines=houghlines(BW,T,R,P,'FillGap',5,'MinLength',7);

figure,imshow(rotI),hold on

max\_len=0;

for k=1:length(lines)

xy=[lines(k).point1;lines(k).point2];

plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

%Plot beginning and ends of lines

plot(xy(1,1),xy(1,2),'xw','LineWidth',2);

plot(xy(2,1),xy(2,2),'xw','LineWidth',2);

%Determine the endpoints of the longest line segment

len=norm(lines(k).point1-lines(k).point2);

if(len>max\_len)

max\_len=len;

xy\_long=xy;

end

end

title('检测结果')