第七届数学中国数学建模网络挑战赛
地址：数学中国数学建模网络挑战赛组委会 网址：www.tzmcm.cn
电话：0471-4969085 邮编：010021 Email：2014@tzmcm.cn

第七届“认证杯”数学中国

数学建模网络挑战赛
承 诺 书

我们仔细阅读了第七届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们接受相应处理结果。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为：2162

参赛队员 （签名）：

队员 1：谭宁

队员 2：赵煜熙

队员 3：高有为

参赛队教练员（签名）： 邢俊英

参赛队伍组别：本科组

第七届数学中国数学建模网络挑战赛
地址：数学中国数学建模网络挑战赛组委会 网址：www.tzmcm.cn
电话：0471-4969085 邮编：010021 Email：2014@tzmcm.cn

第七届“认证杯”数学中国

数学建模网络挑战赛
编 号 专 用 页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：

2162

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

第七届数学中国数学建模网络挑战赛
地址：数学中国数学建模网络挑战赛组委会　　网址：www.tzmcm.cn
电话：0471-4969085　　邮编：010021　　Email：2014@tzmcm.cn

2014 年第七届“认证杯”数学中国数学建模网络挑战赛第二阶段论文

题　　目　对低质量图像的多种讨论

关 键 词　数字化处理　领域平均法　Robinson 算子　梯度法图像锐化

摘　　要：

本文采用领域平均法，把图片分成N个领域，在每个领域内中心的像素灰度值与其周围的24个像素灰度值取平均值，在该领域内用平均值来代替这25个像素灰度值；对原图使用一次后，像素灰度值相差变小，于是在对处理后的图像再次进行领域平均法，经过有限次的处理后，有效的阻止了某点像素值的突变与整体不合；使得某个像素点不仅与该点有关而且与其领域的像素点有关；使得各个领域融为一体，图像变的更加有整体感。

图像梯度是数字图像处理的重要内容，是图像分割、特征提取和图像识别等图像处理技术的重要前提。梯度反映图像局部亮度变化最显著的部分，梯度算法的实质是利用算法提取出图像中对象与背景之间的显著变化。

本文介绍了数字图像处理的概念及其应用领域、以及梯度域算法的典型应用。并详细介绍了常见的图像梯度计算方法，如Robert算子、sobel算子、Robinson算子等，通过理论分析和实验结果比较了他们各自的优缺点。最后提出一种新梯度域图像锐化算法，通过区分高梯度、中等梯度、低梯度和噪声区域的方法进行算法优化，取得了良好的实际效果。

参赛队号：2162

所选题目：B 题

参赛密码 _____
（由组委会填写）

§1 问题重述

图形（或图像）在计算机里主要有两种存储和表示方法。矢量图是使用点、直线或多边形等基于数学方程的几何对象来描述图形，位图则使用像素来描述图像。一般来说，照片等相对杂乱的图像使用位图格式较为合适，矢量图则多用于工程制图、标志、字体等场合。矢量图可以任意放缩，图形不会有任何改变。而位图一旦放大后会产生较为明显的模糊，线条也会出现锯齿边缘等现象。

第二阶段问题： 位图在放大时，图像质量常会有所下降，如容易产生较为明显的模糊或马赛克等现象（见图 2)。请你建立合理的数学模型，来设计一个放大位图的算法，使图像在被放大后仍能尽量保持较好的图像质量。

§2 模型的假设与符号的约定

§2.1 模型的假设与说明

1. 所处理的图片是一幅连续图像。
2. 如果位图图像存在渐变色，假设图像对比度较大，边界易分辨。

§2.2 符号的约定与说明

$g(x,y)$	平滑后的图像中的每个像素的灰度值
S	(x,y) 点领域中点坐标的集合
$f(x,y)$	一副 N*N 个像素的图像
λ	光的波长
t	时间轴坐标
I	光点（x,y,z）
（x,y,z）	表示空间某个点的坐标

§3 问题的分析

由于图像放大后会出现明显的马赛克现象，使图像模糊不清，图片上各点像素与邻近区域上的像素灰度值相差较大；特别是交界处灰度值发生突变，使得图片相邻领域看起来显得棱角分明，近距离不能肉眼不能分辨出图像中的内容。

位图是通过许多像素点表示的图像，每个像素具有颜色属性和位置属性。位图分成如下四种：二值图像 (binary images)、亮度图像 (intensity images)、索引图像(indexed images)和 RGB 图像(RGB images)。

图像有许多种分类方法；按照空间可分为平面与立体；在产生方面可分为自发光与反射（透射）,按照图像的动态特性，可以分为静止图像和运动图像；按照图像的色彩，可以分为灰度图像和彩色图像；按照图像的维数，可分为二维图像，三维图像和多维图像。对于任一幅连续图像，均可用下面的函数形式来表示：

$$I=(x,y,z,t,\lambda)$$

$f(x,y,z,\lambda,t)$表示一幅运动（t）的、彩色/多光谱（λ）的、立体(x,y,z)图像。对于静止图像，则与时间 t 无关；对于单色图像（也称灰度图像），则波长 λ 为一常数；对于平面图像，则与坐标 z 无关，故 $\begin{bmatrix} f(0,0) \\ \vdots \\ \mathrm{f}(1,0) \\ \vdots \\ f(1,\mathrm{N}-1) \\ \vdots \\ f(\mathrm{M}-1,\mathrm{N}-1) \end{bmatrix}$表示平面上的静止灰度图像，它是一般图像 $f(x,y,z,\lambda,t)$的一个特例。

在不同情况下可分为：二维图像：$I=f(x,y,\lambda,t)$；单色图像：$I=f(x,y,z,t)$；静态图像：$I=f(x,y,z,\lambda)$

数字图像在计算机中是以文件的形式存储的。常见的图像数据格式包括 BMP 格式、JPEG 格式、GIF 格式。Bmp 格式图像信息丰富，一般采取不压缩的形式。以图像左下角为起点存储图像。调色板数据结构中，RGB3 基色数据的排列顺序恰好与其他格式文件的顺序相反。　JPEG 格式图像文件的扩展名是.jpg 或是.jpeg，压 缩率很大，但是却不易觉察。适用性广泛，大多数图像类型都可以进行 JPEG 编码，现已广泛应用于 Internet 技术，节省了宝贵的网络资源。JPEG2000 的一个极其重要的特征在于它能实现渐进传输。GIF 格主要用于网络传输。最多存储 256 色，体积小，清晰度高，对灰度图像表现最佳。一个文件存储多个图像，可实现动画功能。与 JEPG2000 图像一样，也实现渐进传输。但 GIF 处理涉及动态变化，我只考虑到 bmp 和 jpeg 格式。

§4　模型的建立与求解

4.1　模型一

4.1．1　图形数字化

将代表图像的连续（模拟）信号转变为离散（数字）信号的变换过程，要解决两个问题：空间取样（空间坐标的离散化）幅度的量化（幅度的离散化，灰度值或亮度值变为若干级），数字图像（DIDITAL　IMAGE）在空间坐标和亮度上都离散化了的图像。

数字图像处理时常常是将其视为一个矩阵来处理的。首先对$f(x,y)$作等间隔采样，设取 M x N 个数据，将这些数据按采样点的对应位置排成一个矩阵，然后量化矩阵，得到一个数字矩阵$f(x,y)$，即数字图像可以表示为一个矩阵.矩阵的元素称为数字图像上的像素。图像数学表达式：

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \cdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(\mathrm{M}-1,N-1) \end{bmatrix}$$

$$\begin{bmatrix} f(0,0) \\ \vdots \\ \mathrm{f}(1,0) \\ \vdots \\ f(1,\mathrm{N}-1) \\ \vdots \\ f(\mathrm{M}-1,\mathrm{N}-1) \end{bmatrix}$$

有时为了处理方便，需要将数字图像的元素转化成向量形式。例如：

但利用 matlab 可以之间调用程序可以节省大量的运算。

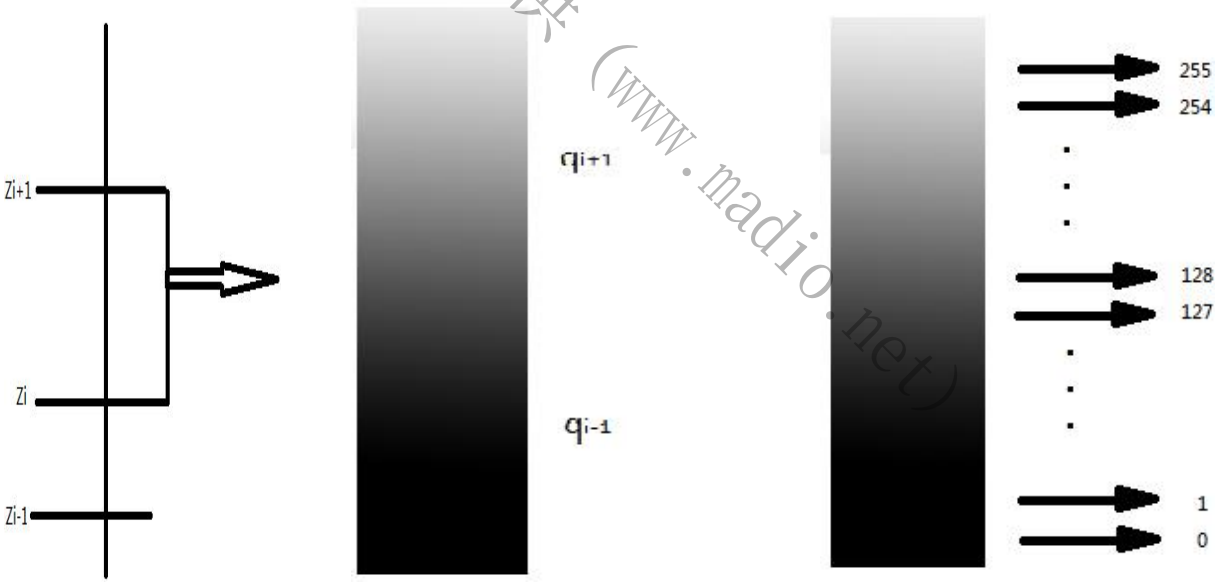
图像元素是二维空间的一个坐标，其特点是以平面上的点作为函数变量。灰度图像可以用灰度 0~255 值来表示，记为 $f(x,y)$，他表示图像在水平和垂直方向上的光照强度变化。图像在二为空间进行采样时，一般平均提取样本，构成一个离散函数 $f(i,\mathrm{j})$，若图像为彩色图，则是以三基色（RGB）的明亮度来作为函数来表示，即：

$$f(\mathrm{x},\mathrm{y})=[f_R(x,y),f_G(x,y),f_B(x,y)]^T$$

$$f(\mathrm{x},\mathrm{y})=[f_R(i,\mathrm{j}),f_G(\mathrm{i},\mathrm{j}),f_B(\mathrm{i},\mathrm{j})]^T$$

相应的离散值：

经过离散化的图像，采集的灰度值任是连续的；如图：



连续灰度值灰度标度　　整数值精确度量化

（量化）　　（量化为 8*bite*）

图像在采集时，行列的采集点与量化的级数决定了数字图像的的质量和图像数据所占空间的大小，设图像取 M xN 采样点，$f(x,y)$ 代表(x, y)点的灰度值。在一般情况下，每个像素的量化后的二进制位数 Q 和最大灰度级数 G 都取为 2 的整数幂，即：

$$Q=2^n \qquad G=2^m \ (n=0,1,2,3\cdots;m=0,1,2,3\cdots)$$

那么，一幅数字图像在计算机中所占字节数 B (*Byte*) 二进制位数为 b 为 (*bite*)：

$$B=\frac{M\times N\times Q}{8}\qquad b=M\times N\times Q$$

例如，大小为 512×512 灰度级为 256 的一幅数字图像，需要大约 210 万个储存位。

4.1. 2 领域平均法

线性低通滤波器是最常用的线性平滑滤波器。实现这种滤波的方法也称领域平均法。领域平均法是一种局部空间域处理的算法，把几个像素灰度平均值来代替每个像素。一副 N*N 个像素的图像 $f(x,y),$，平滑处理后得到一幅图像 $g(x,y),g(x,y)$ 由下式决定：

$$g(x,y)=\frac{1}{M}\sum_{(m\bullet n)\in S}f(m\bullet n)\qquad(4\text{-}1)$$

式中，$x,y=0,1,2,\cdots,N-1;S$是 (x,y) 点领域中点坐标的集合，但是其中不包括 (x,y) 点；M是集合内坐标点的总数。式（4-1）说明，平滑后的图像 $g(x,y)$ 中的每个像素的灰度值均由；含在 (x,y) 的预定领域中的 $f(x,y)$ 的几个像素的灰度值来决定。平均算法是将原图中一个像素灰度值和他周围邻近8个像素的灰度值相加，然后求得的平均值（除以9）作为新图像中该灰度值。我们用如下方法来表示该操作；　　3*3模板

$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1^{*}&1\\1&1&1\end{bmatrix}\qquad(4\text{-}2)$$

中间的黑点表示该元素为中心元素，即该个元素是要进行处理的元素，同理可得5*5的模板，如下所示：

$$\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1^{*}&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}\qquad(4\text{-}3)$$

$f(x,y)$像素与周围邻域的关系

$f(i-1,j-1)$	$f(i-1,j)$	$f(i-1,j+1)$
$f(i,j-1)$	$f(i,j)$	$f(i,j+1)$
$f(i+1,j-1)$	$f(i+1,j)$	$f(i+1,j+1)$

十字邻域

0	1	0
1	1	1
0	1	0

方形邻域

1	1	1
1	1	1
1	1	1

3点邻域

0	1	0
0	1	0
0	1	0

3	3	5	2	2	2
3	3 *0	5 *1	2 *0	2	2
4	4 *1	7 *1	2 *1	1	1
5	5 *0	2 *1	3 *0	1	1
5	5	2	3	1	1

由于图像放大后会出现明显的马赛克现象，使图像模糊不清，图片上各点像素与邻近区域上的像素灰度值相差较大；特别是交界处灰度值发生突变，使得图片相邻领域看起来显得棱角分明，近距离不能肉眼不能分辨出图像中的内容。因此采用用领域平均法，把图片分成 N 个领域，在每个领域内中心的像素灰度值与其周围的 24 个像素灰度值取平均值，在该领域内用平均值来代替这 25 个像素灰度值；对原图使用一次后，像素灰度值相差变小，于是在对处理后的图像再次进行领域平均法，经过有限次的处理后，有效的阻止了某点像素值的突变而与整体不合；使得各个领域融为一体，图像变的更加有整体感。

对局部进行分析：

1. 原图局部图像 图4-1-0和 和图片的部分灰度矩阵

Columns 1 through 16

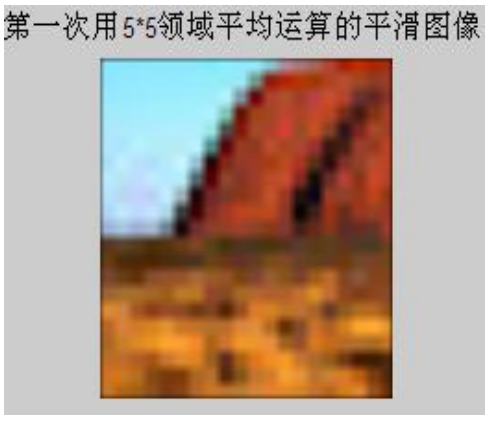
136 136 136 136 136
137 137 137 137 137
137 137 137 137 137
138 138 138 138 138
138 138 138 138 138

Columns 81 through 96

142 142 142 143 143
145 145 145 146 146
145 145 145 146 146
146 147 148 148 149
146 147 148 148 149

图4-1-0

2．局部图像进行一次领域平均法得到图1-1和用领域平均法得到的部分矩阵



Columns 1 through 16

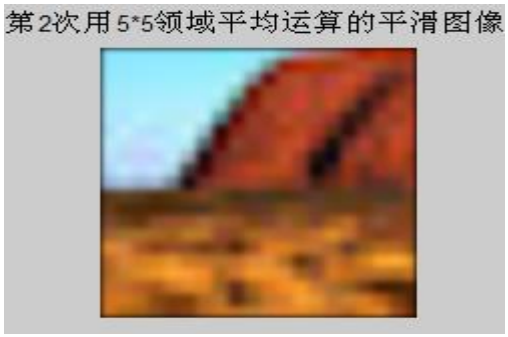
49 66 82 82 82 8
66 88 110 110 110
82 110 137 137 137
83 110 138 138 138
83 110 138 138 138

Columns 81 through 96

86 86 87 87 87 8
115 116 116 116 117
144 145 146 146 146
146 146 147 147 148
146 147 148 148 149

图4-1-1

3．局部图像进行二次领域平均法得到图4-1-2和用领域平均法得到的部分矩阵



Columns 1 through 16

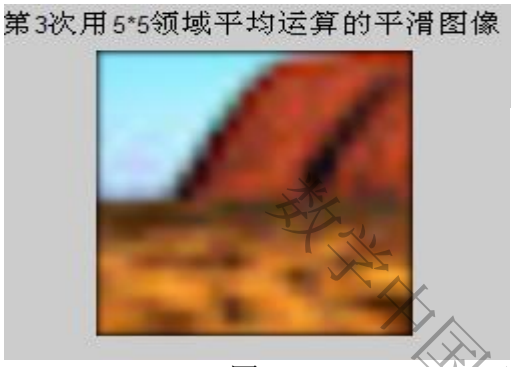
32	45	58	63	66	
45	64	82	90	93	
58	82	106	116	121	
64	90	117	127	132	
66	94	122	133	138	

Columns 81 through 96

69	69	70	70	70	6
98	99	99	99	99	9
127	128	128	129	129	
139	139	140	140	140	
144	143	143	144	143	

图4-1-2

4．局部图像进行三次领域平均法得到图1-3和用领域平均法得到的部分矩阵



Columns 1 through 16

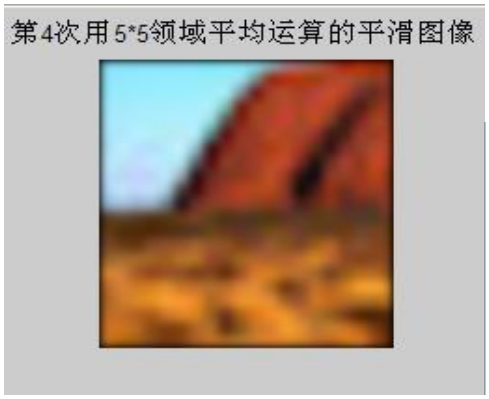
23	34	45	51	54
34	50	66	75	80
45	66	88	100	107
51	75	100	113	121
55	81	107	121	130

Columns 81 through 96

59	59	59	59	59	5
87	87	87	87	87	8
115	116	116	116	116	
130	130	130	130	130	
138	138	138	137	137	

图4-1-3

5．局部图像进行四次领域平均法得到图4-1-4和用领域平均法得到的部分矩阵



Columns 1 through 16

18	27	37	43	46
27	41	55	64	70
37	55	75	87	95
43	64	87	101	110
47	70	95	111	121

Columns 81 through 96

52	52	52	52	52	5
78	78	78	78	78	7
106	106	106	106	105	
122	122	122	121	121	
133	132	131	131	131	

图4-1-4

通过图片进行多次领域平均值法，通过比较相同地方的像素灰度矩阵的值，可看出灰度矩阵的值在减少，各像素灰度值与周围的像素灰度值的方差在不断减少，使图片放大后不会每个区域变的棱角分明；而是看上去更像一个整体。

对整体而言该方法在局部处理得到了有效地应用，对整体用领域平均法：

原图

用程序1-2第一次求领域平均值得到的图片如 图2-1：

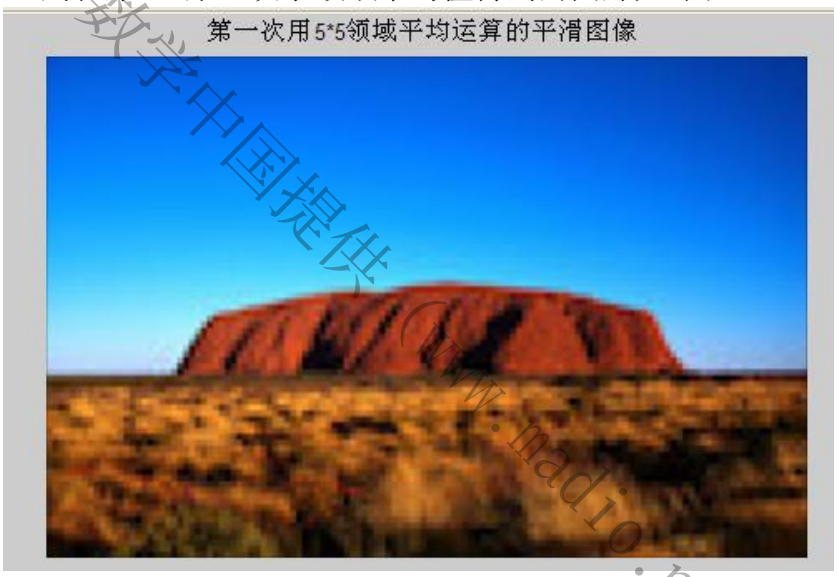


图4-2-1

使用一次领域平均法处理后，图片的马赛克现象有明显的减弱。因此:用程序4-1-2再对图2-1进行领域平均法处理，得到图4-2-2：

图4-2-2

经过有限次的处理后，图像交界处像素灰度值变化比较缓和，从而变的比较的光滑，使得图像中的内容基本可以分辨出来。

4.1．3　图像梯度的计算

4.1．3．1 梯度的概念

设体系中某处的物理参数(如温度、速度、浓度等)为 w，在与其垂直距离的 dy 处该参数为 $w+dw$，则称为该物理参数的梯度，也即该物理参数的变化率。如果参数为速度、浓度、温度或空间，则分别称为速度梯度、浓度梯度温度梯度或空间梯度。

在向量微积分中，标量场的梯度是一个向量场。标量场中某一点上的梯度指向标量场增长最快的方向，梯度的长度是这个最大的变化率。更严格的说，从欧氏空间 Rn 到 R 的函数的梯度是在 Rn 某一点最佳的线性近似。在这个意义上，梯度是雅戈比矩阵的一个特殊情况。

在单变量的实值函数的情况，梯度只是导数，或者，对于一个线性函数，也就是线的斜率。

梯度一词有时用于斜度，也就是一个曲面沿着给定方向的倾斜程度。可以通过取向量梯度和所研究的方向的点积来得到斜度。梯度的数值有时也被称为梯度。

设函数 $z=f(x,y)$ 在平面区域 D 内具有一阶连续偏导数，则对于每一点 $P(x,y)\in D$，都可定出一个向量 $\frac{\partial f}{\partial x}\vec{i}+\frac{\partial f}{\partial y}\vec{j}$，这向量称为函数 $z=f(x,y)$ 在点 $P(x,y)$ 的梯度，记为

$$gradf(x,y)=\frac{\partial f}{\partial x}\vec{i}+\frac{\partial f}{\partial y}\vec{j}. \quad (4\text{-}1)$$

类似的对三元函数也可以定义一个：

$$gradf(x,y)=\frac{\partial f}{\partial x}\vec{i}+\frac{\partial f}{\partial y}\vec{j}+\frac{\delta f}{\delta z}\vec{k} \quad (4\text{-}2)$$

梯度本意是一个向量（矢量），当某一函数在某点处沿着该方向的方向导数取得该点处的最大值，即函数在该点处沿方向变化最快，变化率最大（为该梯度的模）。

梯度是一个数学概念，引入到图像处理中，有很多计算方法。比如：Sobel、Roberts、kirsch、laplace、piewitt、robinson 算子等，但是，图像梯度的方法包括但不限于上述方法，可以根据实际需要自定义一些梯度算子。提取边界线条，需要准确的特征点，特征点能通过图像边缘的像素信息提取。特征点包括图形的边界点以及转折点。转折点是指图像比较尖锐的点，准确的特征点能提高曲线提高准确性。因此特征点的寻找在整个问题中占着很大的比重。

4.1．3.2 图像梯度的定义

图像函数 $f(x，y)$ 在点 $(x，y)$ 的梯度（即一阶微分）是一个具有大小和方向的矢量，设 Gx ，Gy 分别表示沿 x 方向和 y 方向的梯度[8]，那么这个梯度矢量可以表示为：

$$\nabla f(x,y)=[G_x,G_y]^T=[\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}]^T \tag{4-3}$$

这个矢量的幅度为：

$$mag(\nabla f)=g(x,y)=\sqrt{\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2}} \tag{4-4}$$

方向角为：

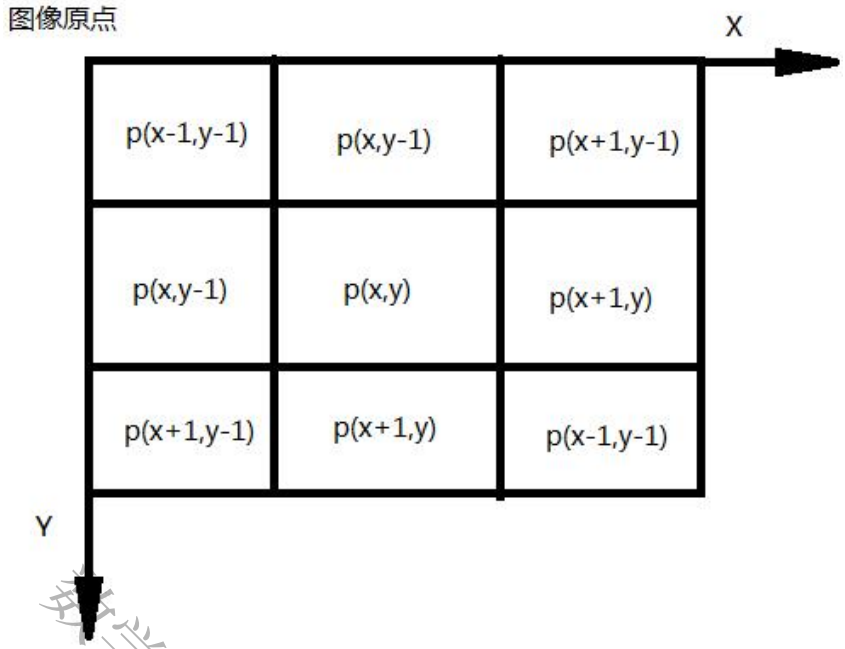
$$\varphi(x,y)=\arctan\left|\frac{(\frac{\partial f}{\partial y})}{(\frac{\partial f}{\partial x})}\right| \tag{4-5}$$

对于数字图像，导数可用差分来近似，对于图像 $f(x,y)$，用差分来近似代替导数，则在点 (x,y) 处沿 x 方向和 y 方向的一阶差分可表示为：

$$\Delta_x f(x,y)=f(x+1,y)-f(x,y)$$

$$\Delta_y f(x,y)=f(x,y+1)-f(x,y) \tag{4-6}$$

这种差分为水平垂直差分，在图像中表示如图：



水平垂直差分示意图

这是常用的水平垂直差分，还有另一种差分，罗伯特差分法，这里就不再说明了。

同时，如上图所示，对于图像来说，通常图像中最右一列和最下一行的各像素的梯度无法求得，一般用前一列和前一行的近似表示。

图像梯度可以把图像看成二维离散函数，图像梯度其实就是这个二维离散函数的求导图像梯度：$G(x,y) = dx\,i + dy\,j$；

$$dx(i,j) = I(i+1,j) - I(i,j);$$
$$dy(i,j) = I(i,j+1) - I(i,j); \qquad (4\text{-}7)$$

其中，I 是图像像素的值(如：RGB 值)，(i,j)为像素的坐标。

图像梯度一般也可以用中值差分：

$$dx(i,j) = \frac{I(i+1,j) - I(i,j)}{2}$$
$$dy(i,j) = \frac{I(i,j+1) - I(i,j)}{2}; \qquad (4\text{-}8)$$

中分差表示图像的平均梯度(meangradient)，是指图像的边界或影线两侧附近灰度有明显差异，即灰度变化率大，这种变化率的大小可用来表示图像清晰度。它反映了图像微小细节反差变化的速率，即图像多维方向上密度变化的速率，表征图像的相对清晰程度。平均梯度即图像的清晰度(definition)，反映图像对细节对比的表达能力。

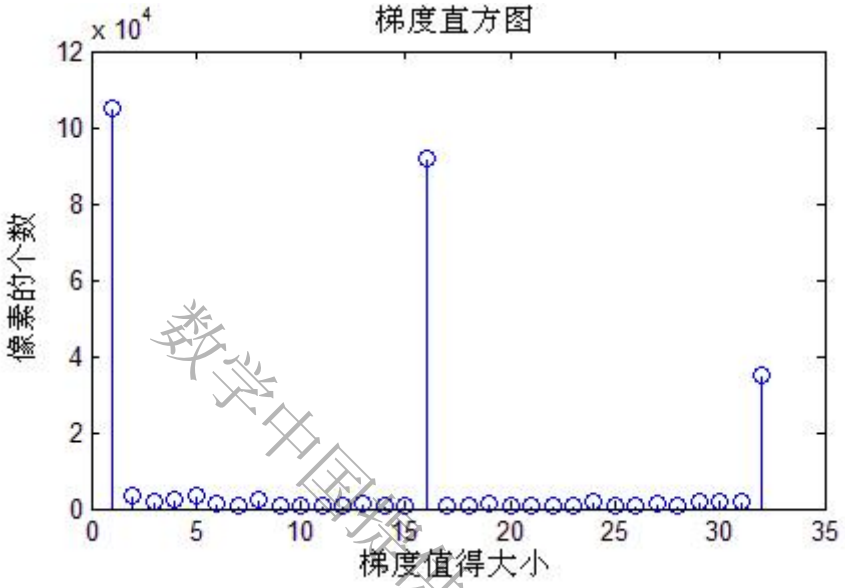
上面说的是简单的梯度定义，其实还有更多更复杂的梯度公式。最简单的梯度近似表达式为：

$$G_x = f(i,j) - f(i-1,j)$$
$$G_y = f(i,j) - f(i,j-1) \qquad (4\text{-}9)$$

有时为了提高速度，降低复杂度，可采用：　$mag(\nabla f)=|G_x|+|G_y|$ (4-10)

梯度的方向是函数 $f(x,y)$ 变化最快的地方，当图像中存在边缘时，一定有较大的梯度值；而图像中较平滑的部分，灰度值变化较小，一般有较小的梯度。图像处理中常把梯度的模简称为梯度，由图像梯度构成的图像称为梯度图像。

我通过计算每一个点的梯度值，利用 MATLAB 调用程序 The gradient histogram（见附录 7.1）的如图：



从图像看出原数字图的边缘存在极大的梯度变化。

如果将边缘认为是一定数量点亮度发生变化的地方，那么边缘检测大体上就是计算这个亮度变化的导数。为简化起见，我们可以先在一维空间分析边缘检测。在这个例子中，我们的数据是一行不同点亮度的数据。例如，在下面的 1 维数据中我们可以直观地说在第 4 与第 5 个点之间有一个边界如图：

5	7	6	4	152	148	149

例图

以上各式的偏导数需要对每个像素的位置计算，经典的图像梯度算法是考虑图像的每个像素的某个邻域内的灰度的变化，利用边缘临近的一阶或二阶导数变化规律，对原始图像中像素的某个邻域来设置梯度算子，在实际中常用小区域模板进行卷积来计算。根据模板的大小及权重的不同，人们提出了很多梯度算子，常用的算子有：Sobel 算子、Robinson 算子，Laplace 算子等。

4.1. 3.3 图像梯度常用计算算法

4.1.3.3-1 图像卷积

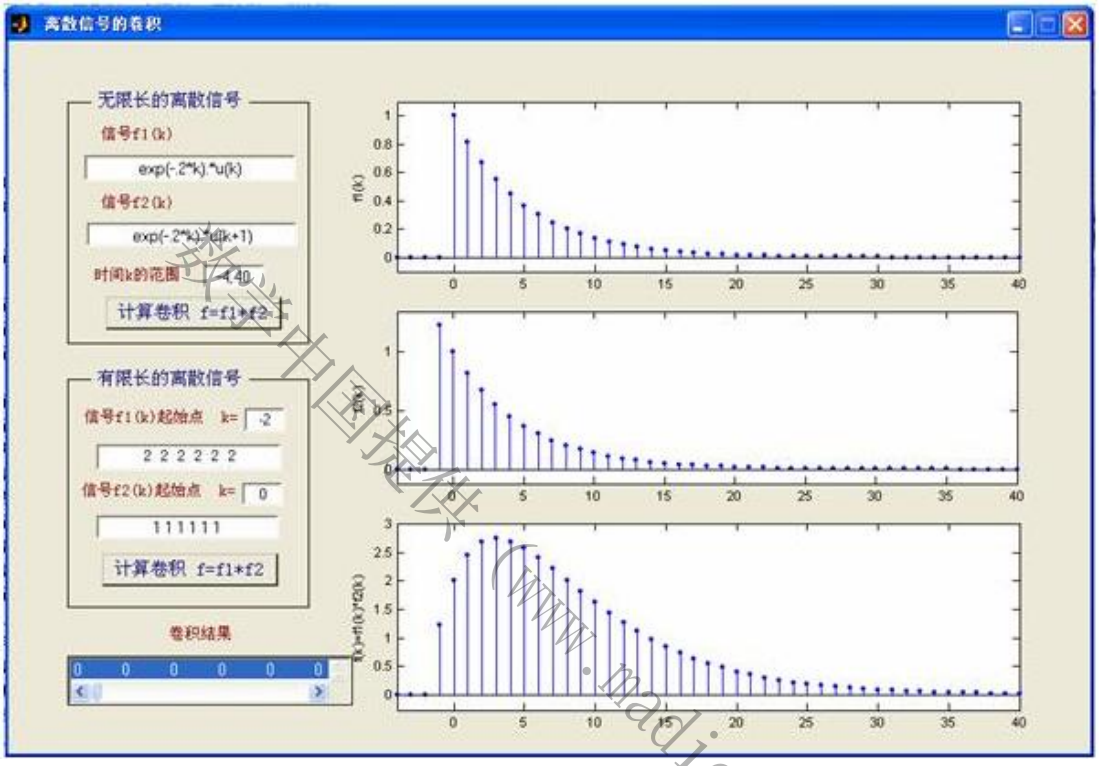
卷积是一种线性运算,图像处理中常见的 mask 运算都是卷积，广泛应用于图像滤波。 castlman 的数字图像处理对卷积讲得很详细。

$$F(X)=\frac{\sum_{K=1}^{\infty}D^{k}(x)e^{-\lambda}\lambda^{k}}{k!} \qquad (4.1\text{-}1)$$

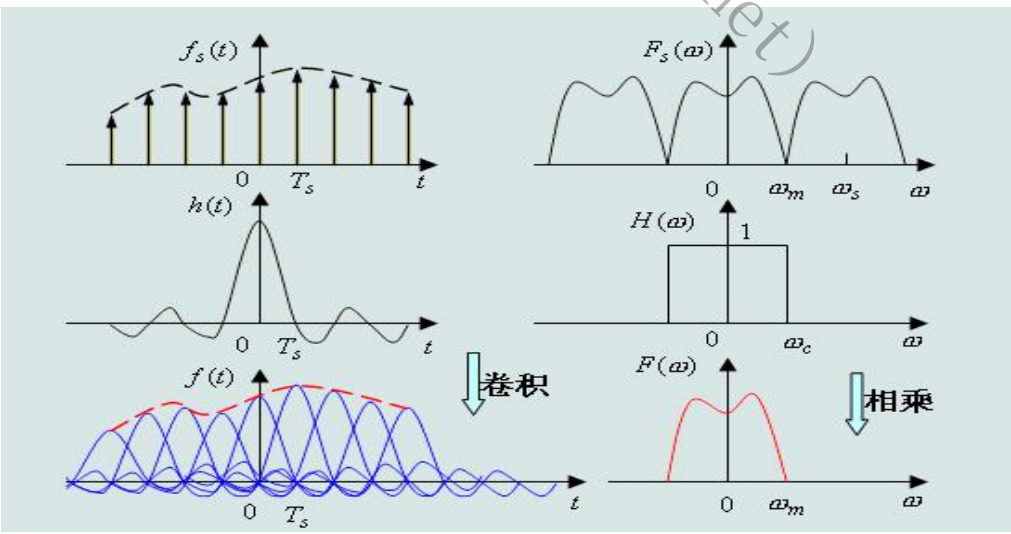
其中 $D(k)(x)$ 为 k 阶卷积。

如果将参加卷积的一个函数看作区间的指示函数，卷积还可以被看作是“滑动平均”的推广。

两个指数函数的卷积。如图



指数函数的卷积



函数叠加图

上图表明 f(t) 可以展开为正交的抽样函数的无穷级数且级数的系数等于抽样值 $f(nt_i)$，这样，若在抽样信号 $f_i(t)$ 的每个抽样值上画一个峰值为 $f(nt_i)$ 的 Sa 函数的波形，合成的波形就是 f(t)。另外，我们知道：Sa 函数的波形就是理想低通滤波器的冲激响应 h(t)，这样， 若通过理想低通滤波器，那么每一个抽样值产生一个冲激响应 h(t)，这些响应进行叠加便得到 f(t)，从而达到恢复信号的目的。

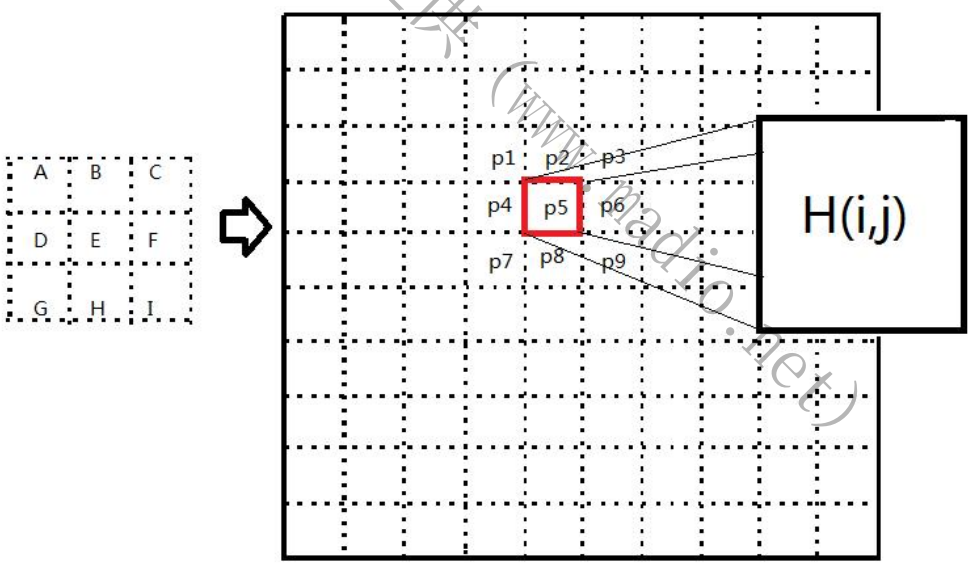
设 f(t) 是一带限连续信号，最高频率为 ω_m，根据定理一对 f(t) 进行抽样，得到 $f(nt_i)$，则 $f(nt_i)$ 经过一个频率响应为如图的理想低通滤波器后便得到 f(t)。由于它是讨论由离散信号恢复成连续信号，所以又称重建定

最简单的线性滤波器是局部卷积，即每一个像素值用其局部邻域内所有值的卷积置换：

$$h(x,y)=f(x,y)*g(x,y)$$
$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x',y')g(x-x',y-y')dx'dy'. \qquad (4.1\text{-}2)$$

若为离散函数，上式变为

$$h[i,j]=f[i,j]*g[i,j]$$
$$=\sum_{k=0}^{n-1}\sum_{l=0}^{m-1}f[k,l]g[i-k,j-l] \qquad (4.1\text{-}3)$$



积模板原点对应于位置 E，而权重 $A,B,...,I$ 是 $g[-k,-l]$ 的值，其中 $k,l=-1,0,+1$．如果 $f(x,y)$ 和 $h(x,y)$ 表示图像，则卷积就变成了对像素点的加权计算，脉冲响应(对于离散系统，脉冲响应函数是一个无穷权序列，系统的输出是输入序列 u(t) 与权序列 h(t) 的卷积和。系统的脉冲响应函数是一类非常重要的非参数模型。) $g[i,j]$ 就是一个卷积模板．对图像中每一像素点 $[i,j]$，输出响应值 $h(x,y)$ 是通过平移卷积模板到像素点 $[i,j]$ 处，计算模板与像素点 $[i,j]$ 邻域加权得到的，其中各加权值对应卷积模板的各对应值。是模板为 3×3 的示意图。卷积是线性运算 ，如下式，因为

$$g[i,j]*\{a_1f_1[i,j]+a_2f_2[i,j]\}=a_1\{g[i,j]*f_1[i,j]\}+a_2\{g[i,j]*f_2[i,j]\}$$

对任何常量a_1和a_2都成立．换句话说，和的卷积等于卷积的和，尺度变换后的图像卷积等于卷积后作相应的尺度变换．卷积是空间不变算子，因为在整幅图像中都使用相同的权重系数．但空间可变系统则在图像的不同部分要求不同的滤波权重因子，因此这种运算无法用卷积来表示．[20]

4.4.3.3-2 Sobel算子

Sobel算子主要用作边缘检测。在技术上，它是一离散性差分算子，用来运算图像亮度函数的梯度之近似值。在图像的任何一点使用此算子，将会产生对应的梯度矢量或是其法矢量。

该算子包含两组3x3的矩阵，分别为横向及纵向，将之与图像作平面卷积，即可分别得出横向及纵向的亮度差分近似值。如果以A代表原始图像，Gx及Gy分别代表经横向及纵向边缘检测的图像Sobel算子，s_x和s_y可用卷积模板来实现，其公式如下:

$$Sy=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix},Sx=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \quad (4.1\text{-}4)$$

其中的偏导数用下式计算:

$$\begin{aligned}s_x&=(a_2+ca_3+a_4)-(a_0+ca_7+a_6)\\ s_y&=(a_0+ca_1+a_2)-(a_6+ca_5+a_4)\end{aligned} \quad (4.1\text{-}5)$$

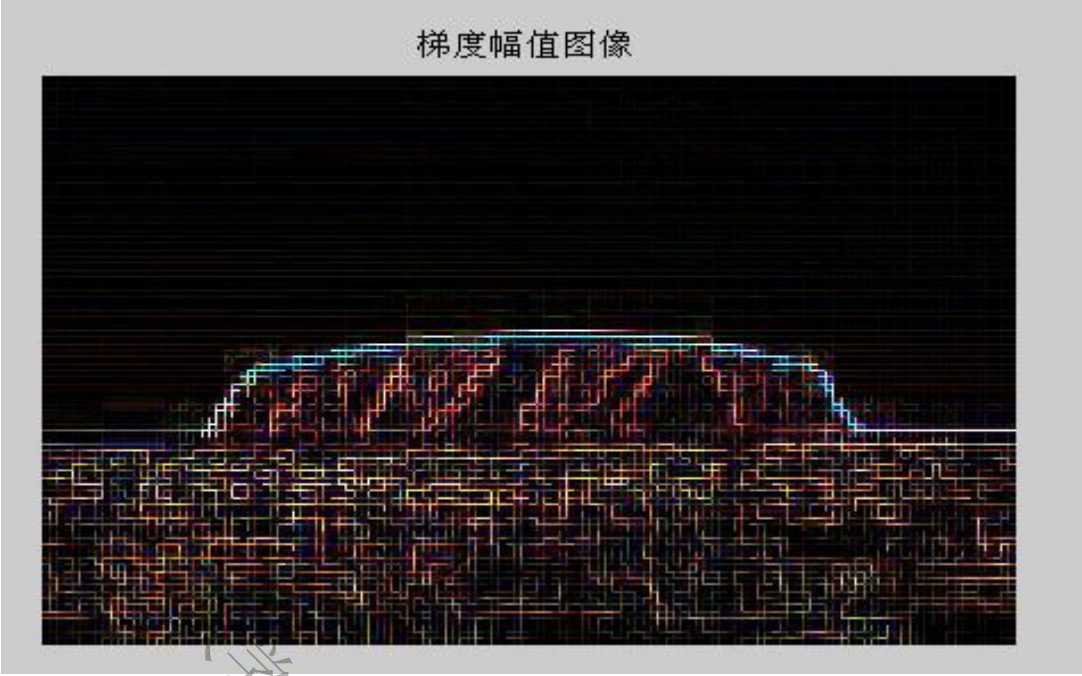
其中常数$c=2$。

Sobel提出一种将方向差分运算与局部平均相结合的方法[2]，Sobel算子将图像中的每个像素的上下左右8领域的灰度值加权求和，Sobel 算法通过2个3x3的模板，对选定而为图像中同样大小窗口进行卷积，得到图像的梯度。Sobel算子是一种梯度幅值，

图像的每一个像素的横向及纵向梯度近似值可用以下的公式结合，来计算梯度的大小:

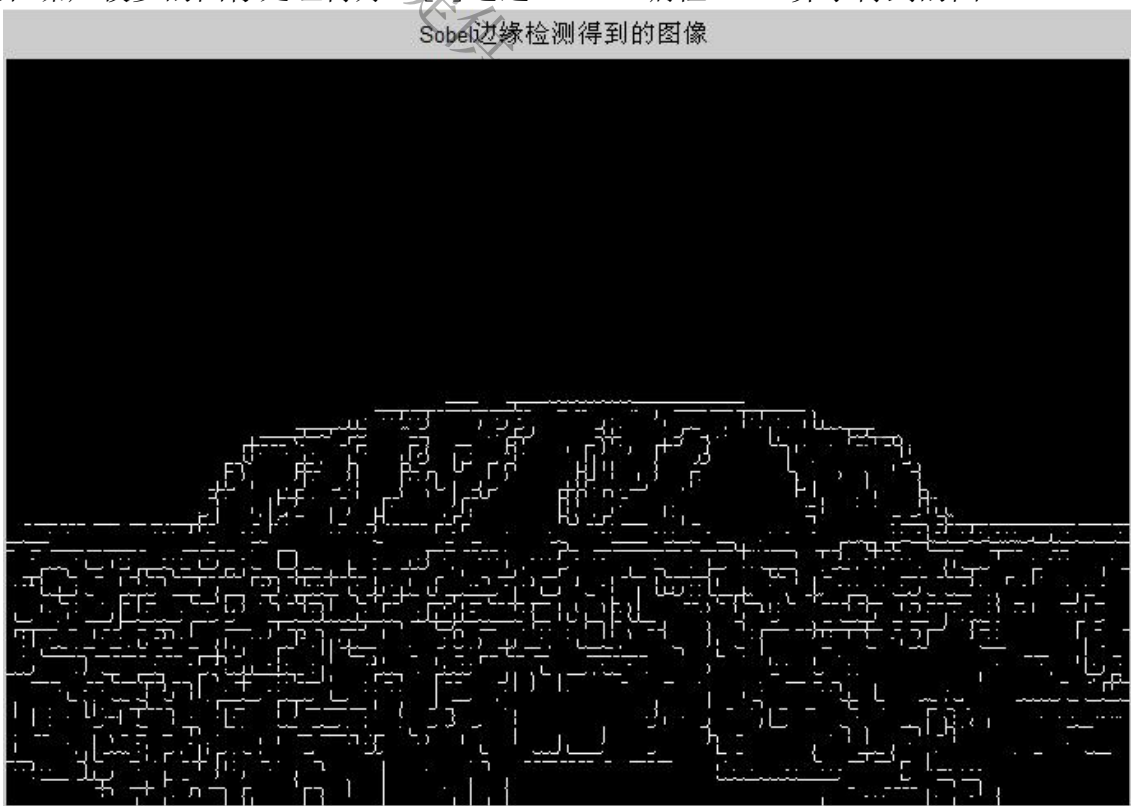
$$M=\sqrt{s_x^2+s_y^2} \quad (4.1\text{-}6)$$

利用matlab编写算法，程序1-4 Gradient magnitude运行程序得图



梯度幅度值图

用∞范数衡量梯度的幅度 $|G(x,y)|\approx \max(|G_X|,|G_y|)$ （4.1-7）。Sobel 算子对图像渐变和噪声较多的图像处理得好。[2]通过 matlab 编程 Sobel 算子得到的图



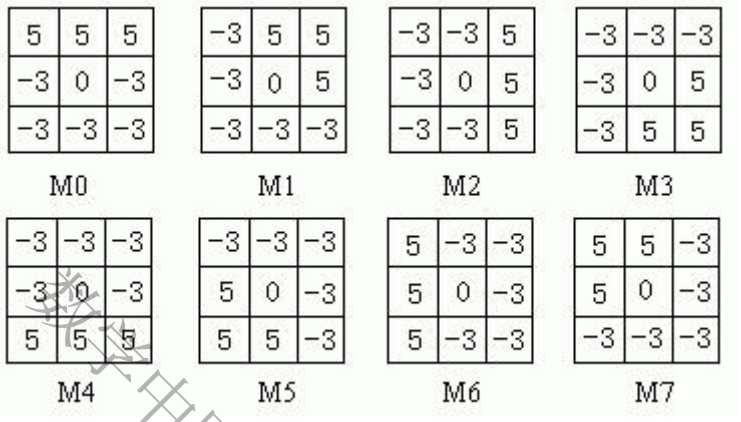
边界图

2.3-3 Robinson 算子

Robinson 算子是一种梯度样板算子[10]，Kirsch 算子依次用方向梯度算子和图像进

行卷积，用其中三个相邻点的加权和减去剩下的五个点的加权和。令 3 个临近点环绕不断以为，取其中的最大值作为梯度值。

Kirsch 算子是一种非线性算子{10]，Kirsch 算子对图像中的每个像素(x，y)，考虑他的 8 邻域灰度变化，用其中三个相邻点的加权和减去剩下的五个点的加权和。令 3 个临近点环绕不断以为，取其中的最大值作为梯度值. Kirsch 算子使用了 8 个模板来确定梯度幅度值和梯度的方向。

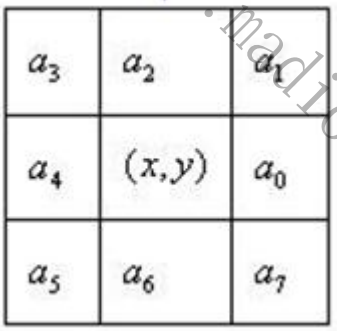


梯度幅度值和梯度的方向图

图像中的每个点都用 8 个掩模进行卷积，每个掩模对某个特定边缘方向作出最大响应。所有 8 个方向中的最大值作为边缘幅度图像的输出。最大响应掩模的序号构成了对边缘方向的编码。

Kirsch 算子的梯度幅度值用如下公式:

$$G(x,y)=\max(|M_1|,|M_2||M_3|,|M_4|,|M_5|,|M_6|,|M_7|,|M_8|) \quad (4.2\text{-}1)$$



图像中的点图

边缘的梯度幅度值：$G(x,y)=\max\{1,\max\{|5s_k-4t_k|\};k=0,1\cdots7\}$；(4.2-2)

$$s_k=a_k+a_{k+1}+a_{k+2}\text{；}\quad t_k=a_{k+3}+a_{k+4}+a_{k+7} \quad (4.2\text{-}3)$$

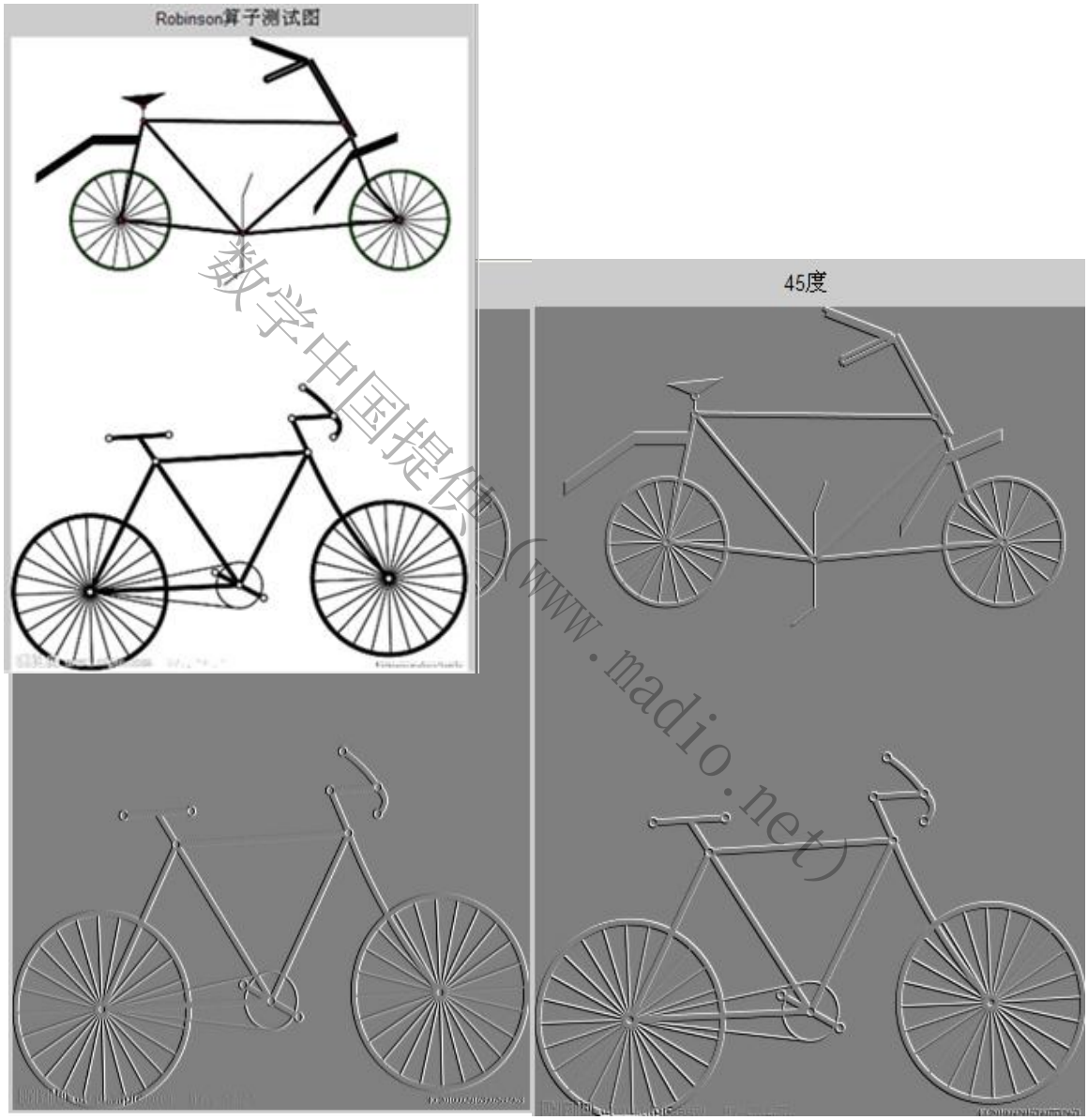
$$\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix}\quad\begin{bmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{bmatrix}\quad\begin{bmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{bmatrix}\quad\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix}$$

（1）90 度方向　　（2）0 度方向　　（3）270 度方向　　（4）180 度方向

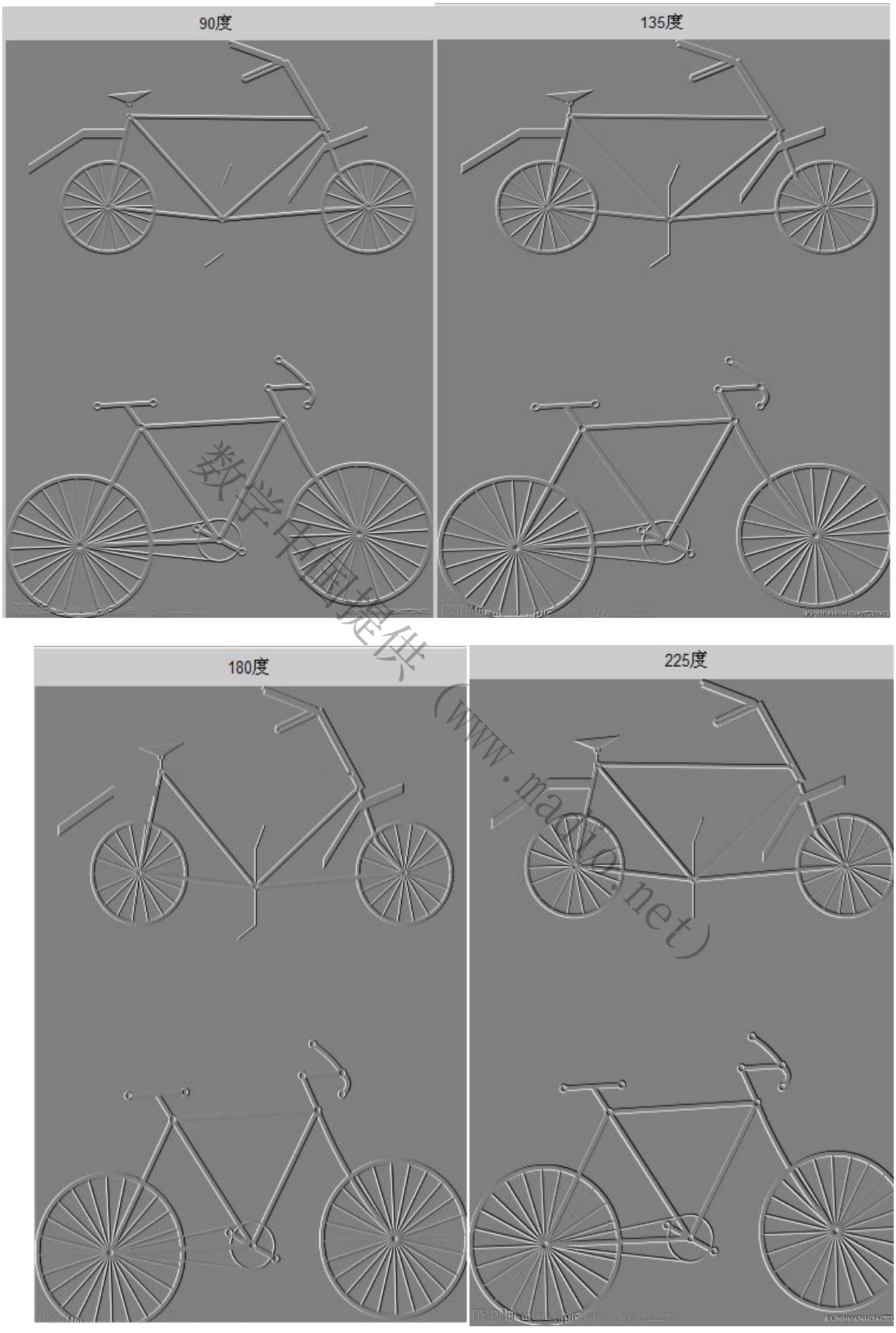
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

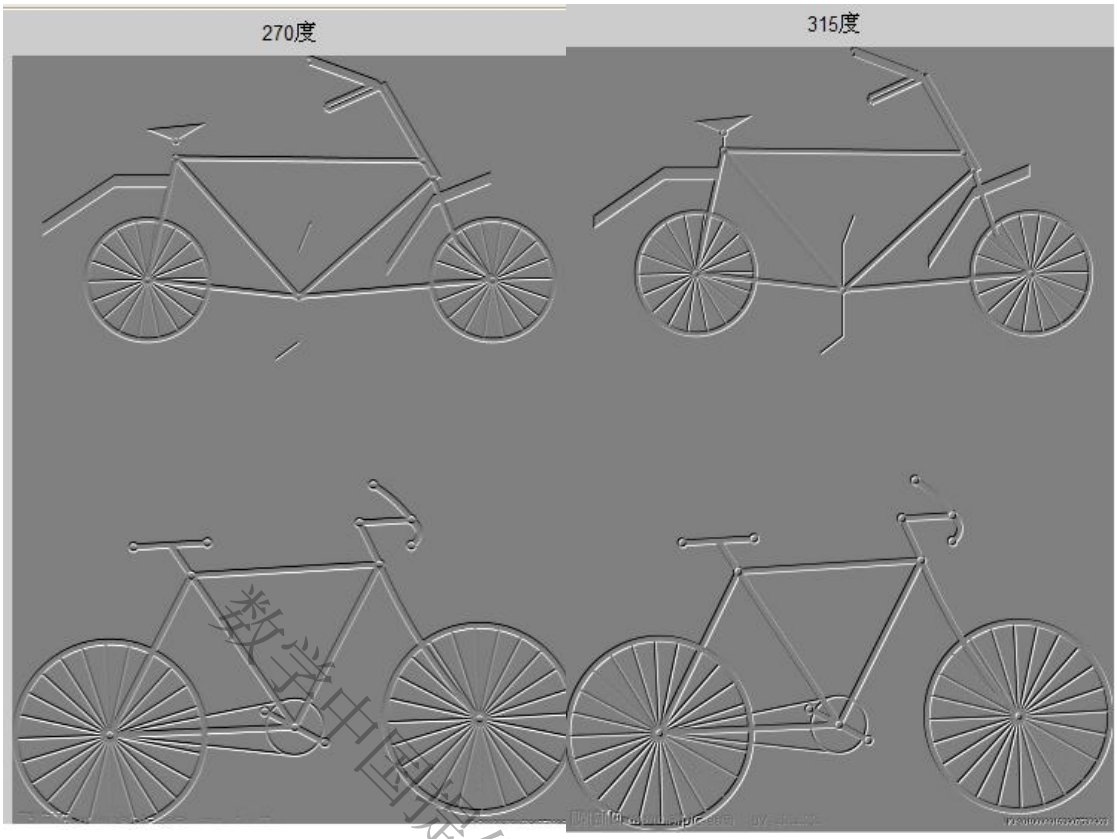
（5）45 度方向　　（6）315 度方向　　（7）225 度方向　　（8）135 度方向

由上面的卷积模板，可以依次输出 8 个值，Kirsch 算子的最后输出值 $G(x,y)$ 为上述 8 个值的最大值。为了最大化的反应各个方向算子的效果，下载了一张特殊的简画图：



#2162





Robinson 算子各方向梯度图

这张图中既有 0 度、30 度、60 度、90 度、120 度、180 度的边缘，也有 45 度、135 度的边缘；既有从黑到白的灰度变化，又有从白到黑的梯度变化，能够最大化的反应出算法的效果。同时，这里还有大小不等的圆，验证其他角度的梯度对算法的适用程度。

4.1.3 图像梯度的应用

在图像识别中，需要有边缘鲜明的图像，即图像锐化。图像如花的目的为了突出图像的边缘信息，加强图像边缘信息以便于人眼和机器的识别然而边缘模糊是图像常出现的质量问题。从图像增强的角度看，它是与图像平滑的相反的一类处理。

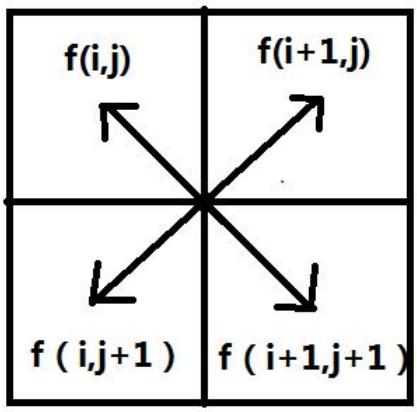
锐化是比较常见的一种算法，就是补偿图像的轮廓，增强图像的边缘及灰度跳变的部分，使图像变得清晰.但是要做好难度还是挺大的。常用的噪声算法在提升图像细节的同时，也放大了图像的噪声。这里提出的算法是一种能有应对复杂场景的图像锐化算法。在极大的提高图像细节的同时，很好的抑制了图像的噪声。

图像锐化常用的一种方法是梯度法。Roberts 梯度差分法原理如下：

$$\begin{cases} G_x = f(i+1, j+1) - f(i, j) \\ G_y = f(i, j+1) - f(i+1, j) \end{cases} \quad (4.3\text{-}1)$$

可得到 Roberts 的梯度为：

$$|G[f(i,j)]| = \nabla f(i,j) \approx \max\{|f(i+1,j+1) - f(i,j)|, |f(i,j+1) - f(i+1,j)|\} \quad (4.3\text{-}2)$$



交叉差分法

从公式可以看出，其值是与相邻像素成正比的。在图像轮廓上像素的灰度有陡然变化。梯度至很大，在于相变化相对平缓的区域梯度至较小。而在灰度区域，梯度值为零。由此可见，图像经梯度运算后，留下灰度值急剧变化的点，从而达到了图像的锐化。用 matlab 将成灰度图像，调用程序程序 1-7 Roberts 程序的图像：

Roberts 程序的图像

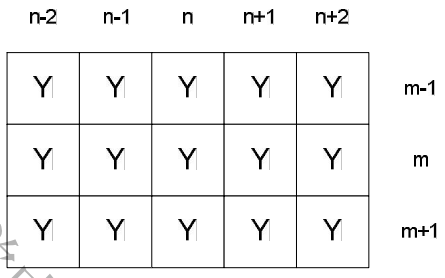
§5 模型的进一步讨论

5.1 梯度域锐化算法改进

Roberts 边缘检测算子是一种利用局部差分算子寻找边缘的算子， Roberts 算子图像

处理后结果边缘不是很平滑。经分析，由于 Roberts 算子通常会在图像边缘附近的区域内 产生较宽的响应，故采用上述算子检测的边缘图像常需做细化处理，边缘定位的精度不是很高。采用一种控制噪声的算法使用一个 3x5 的滤波器提取图像梯度：

$$\frac{\begin{bmatrix} d1 & d2 & d3 & d2 & d1 \\ d4 & d5 & d6 & d5 & d4 \\ d1 & d2 & d3 & d2 & d1 \end{bmatrix}}{Sum} \qquad (5\text{-}1)$$



提取图像

一个典型的滤波器为

$$\begin{bmatrix} -1 & 0 & -1 & 0 & -1 \\ -1 & 0 & 8 & 0 & -1 \\ -1 & 0 & -1 & 0 & -1 \end{bmatrix} \times \frac{1}{8}$$

使用下面的公式计算梯度

$$EM = \frac{(8Y_{m,n} - Y_{m-1,n-2} - Y_{m-1,n} - Y_{m-1,n+2} - Y_{m,n-2} - Y_{m,n+2} - Y_{m+1,n-2} - Y_{m+1,n} - Y_{m+1,n+2})}{8} \qquad (5\text{-}2)$$

如果直接将 EM 叠加到原图像上面，那么在提升边缘细节的同时，对噪声也会加强。为了抑制噪声的影响，使用下面的分段公式调制一下 EM（EM 为相对长度单位，相对于当前对象内文本的字体尺寸。）

$$EMLUT(x) == \begin{cases} mx & x \le -x_3 \\ mx_3 \times (\dfrac{x + x_2}{x_3 - x_2}) & -x_3 < x \le -x_2 \\ 0 & -x_2 < x \le -x_1 \\ \begin{cases} -m_2 x \\ Avg - Y \end{cases} & -x_1 < x \le x_1 \\ 0 & x_1 < x \le x_2 \\ mx_3 \times (\dfrac{x - x_2}{x_3 - x_2}) & x_2 < x \le x_3 \\ mx & x_3 < x \end{cases} \qquad (5\text{-}3)$$

从下图中可以看出，当梯度的叠加做了调制处理如下：

A：强度大于 X3 的部分做一些抑制，因为大于 X3 的部分梯度已经很大，不需要再做加强；

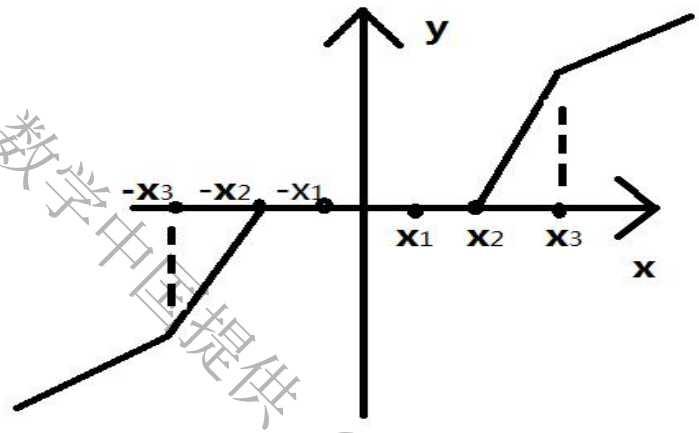
B：x2～x3 之间为梯度为重点需要加强的部分，这里采用拉伸的方法，进一步加大梯度响应；

C：x1～x2 之间的认为是很弱的边缘，不需要加强；

D：0～x1 之间的认为是噪声，此时不但不做锐化加强，相反，此时做一下区域平滑减少噪声。

E：对于负数部分做同样的方式处理。

对于上面的 D，采用平滑算法是有一定风险的，可能噪声边界的部分模糊，这里也做一些特殊处理。先判断边缘方向，使用同方向的灰度进行平滑。



5-1 梯度值图

对于上面的 D，采用平滑算法是有一定风险的，可能噪声边界的部分模糊，这里也做一些特殊处理。先判断边缘方向，使用同方向的灰度进行平滑。

$$AvgH = \frac{Y_{m,n} + Y_{m,n-1} + Y_{m,n+1}}{3} \quad ; \qquad (5\text{-}4)$$

$$AvgV = \frac{Y_{m,n} + Y_{m-1,n} + Y_{m+1,n}}{3} \quad ; \qquad (5\text{-}5)$$

$$AvgL = \frac{Y_{m,n} + Y_{m-1,n-1} + Y_{m+1,n+1}}{3} \quad ; \qquad (5\text{-}6)$$

$$AvgR = \frac{Y_{m,n} + Y_{m-1,n+1} + Y_{m+1,n-1}}{3} \quad ; \qquad (5\text{-}7)$$

DDH = abs(2*Ym,n - Ym,n-1 - Ym, n+1) ; (5-8)

DDV = abs(2*Ym,n - Ym-1,n - Ym+1,n) ; (5-9)

DDL = abs(2*Ym,n - Ym-1,n-1 - Ym+1,n+1) ; (5-10)

DDR = abs(2*Ym,n - Ym-1,n+1 - Ym+1,n-1) ; (5-11)

从 (DDH, DDV, DDL, DDR) 中找出最小值，然后使用对应的 Avg 值来替代当前像素。

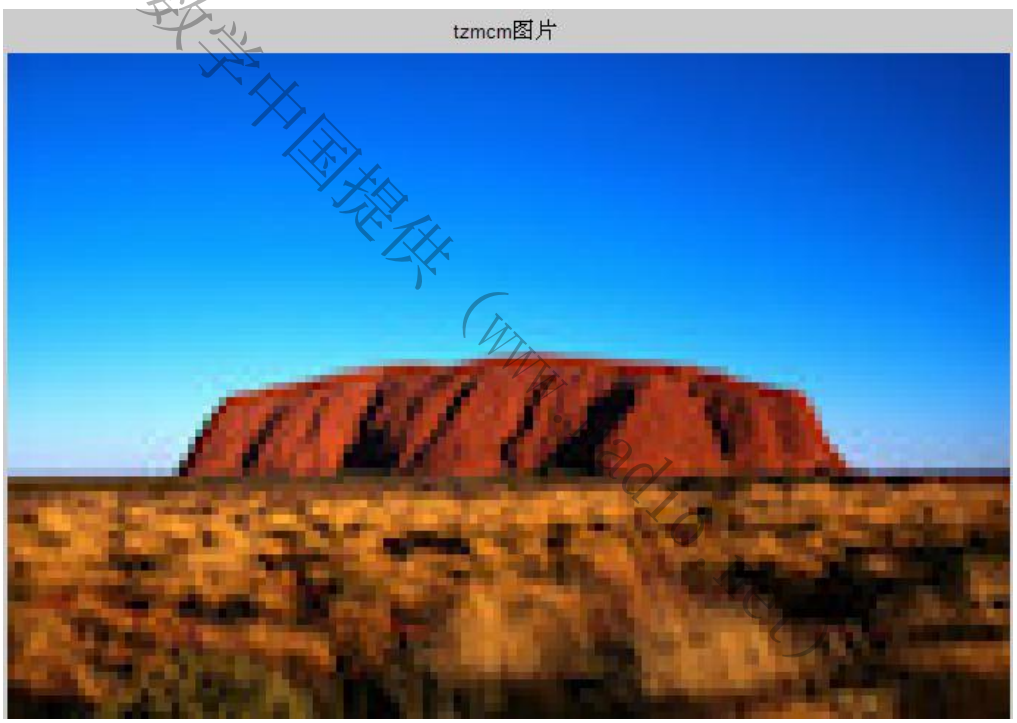
这里就是用这幅图片来说明锐化算法的效果。

对于彩色图像的锐化，通常的做法是对 RGB 三个通道分别进行锐化，这样做的一个缺点是可能导致图像边缘部分出现伪彩色。这里先将 RGB 图像转换到 YUV 色彩空间[14]，只对 Y 通道进行锐化，锐化后再转化为 RGB 图像。将 RGB 转到 YUV 使用如下公式：

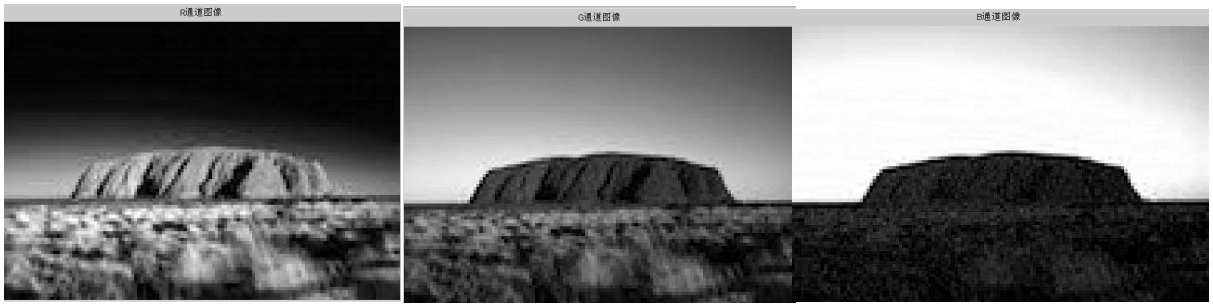
$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & 0.100 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5-12}$$

相反，YUV 转 RGB 的公式为

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.140 \\ 1.0 & -0.394 & -0.581 \\ 1.0 & 2.032 & 0.0 \end{bmatrix} * \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \tag{5-13}$$



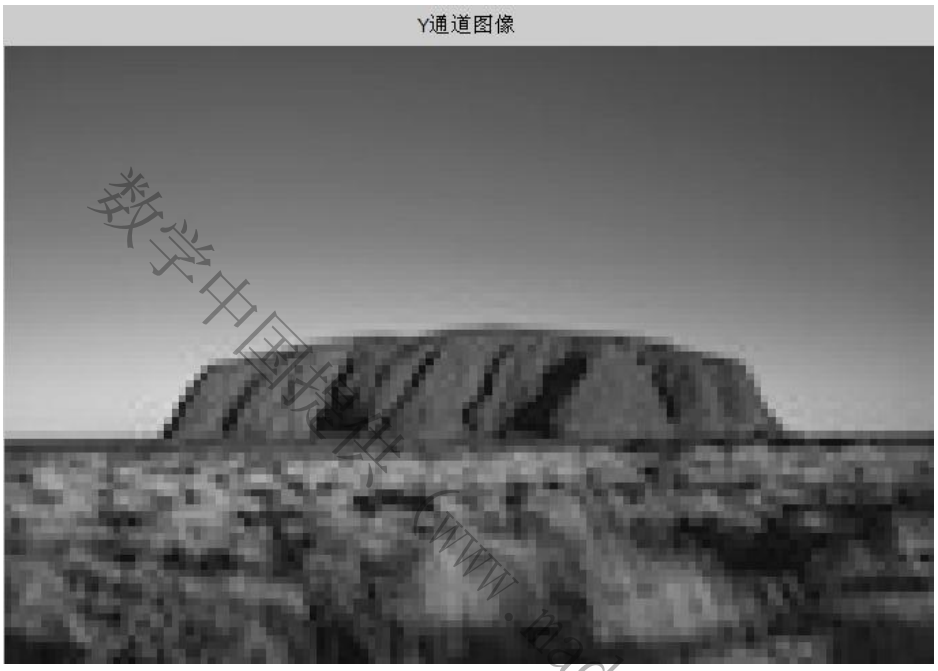
tzmcm



RGB 通道图

UVY 通道图

所有处理在 Y 上进行，Y 通道放大图如图所示：



Y 通道放大图

使用公式 5-3 所描述的 3x5 梯度过滤器与图 5-2 中图片进行卷积后得到的结果如图 5-3 所示：

5-3 亮度通道梯度图

传统锐化方法用图 5-2 和图 5-3 直接相加进行图像锐化，直接相加后得到的图像如图 5-4 所示。

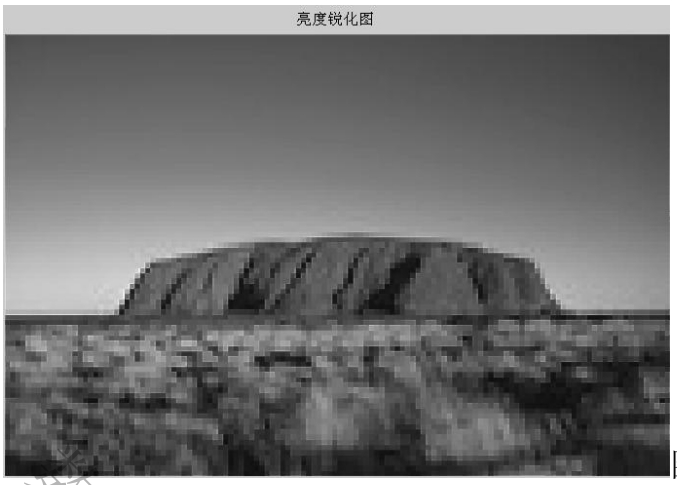


图 5-2

图 5-4

针对上面提到的缺点，我们对图像中的各个梯度部分做一个分开处理，取 X3=40，提取图 5-2 中梯度强度大于 40 的区域，如图 5-5 所示，该高梯度区域的梯度做适当抑制。

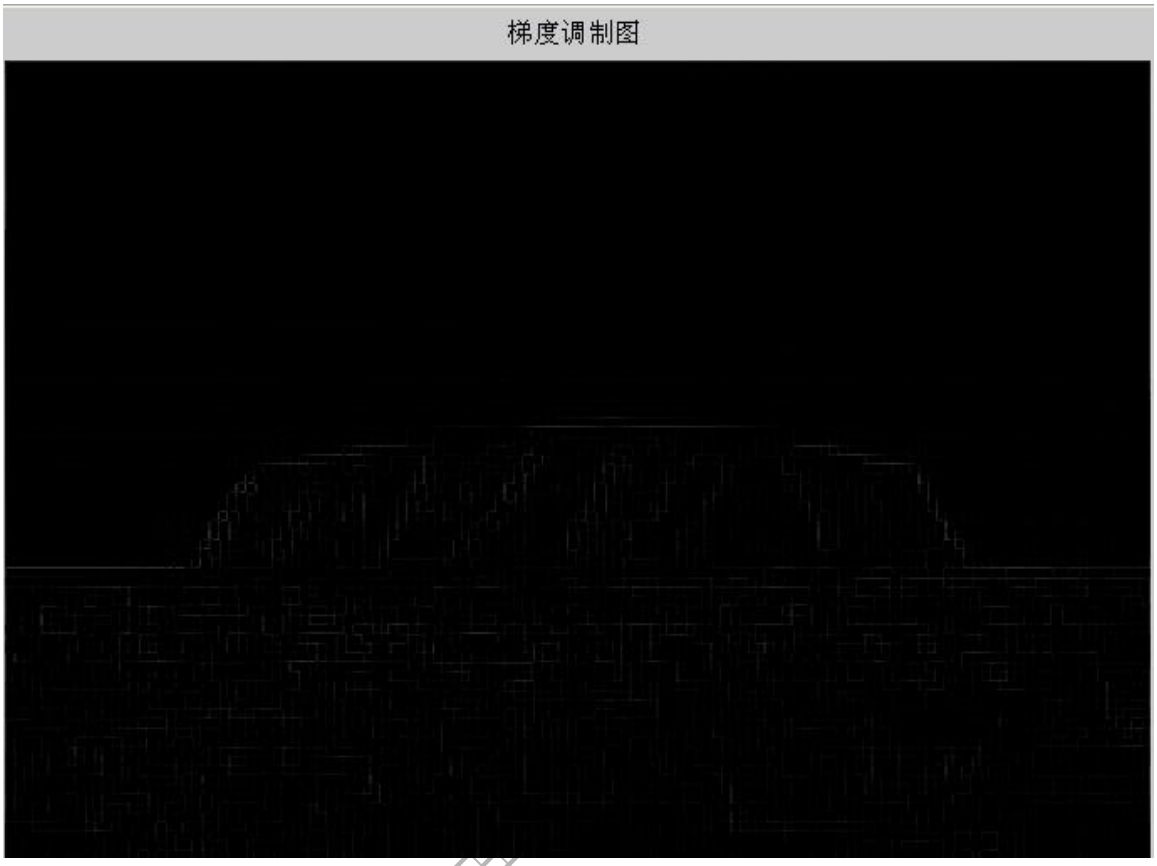


图 5-5

取 X2 为 15，则图 5-1 中在 X2（15）和 X3（40）之间的区域如图 5-6 所示，该区域所示部分的梯度需要做重点加强。

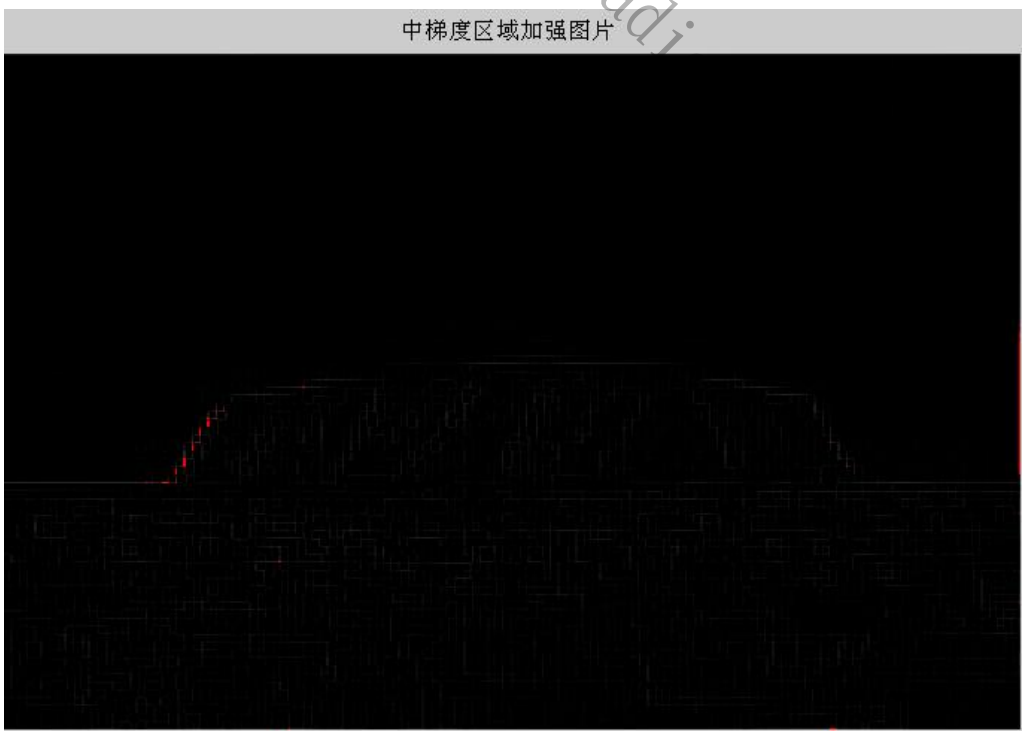


图 5-6

取 X1 为 5，则图 5-1 中在 X1（5）和 X2（15）之间的区域如图 5-7 所示。该区域所示部分的梯度保持不变

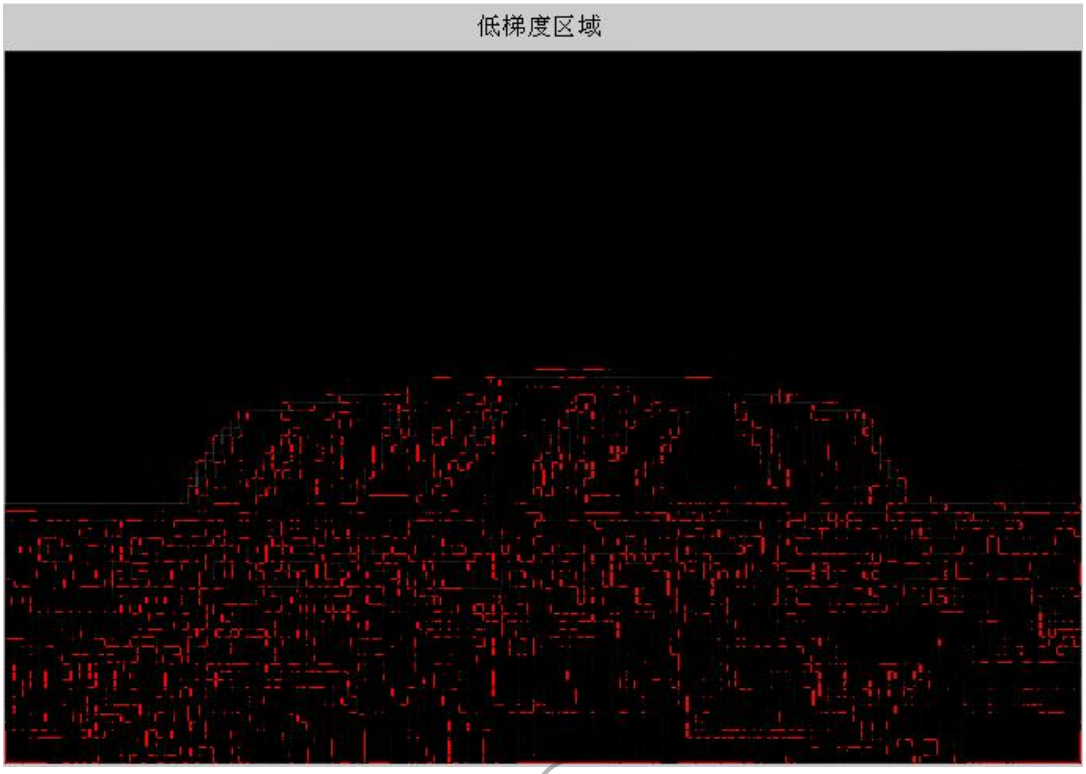
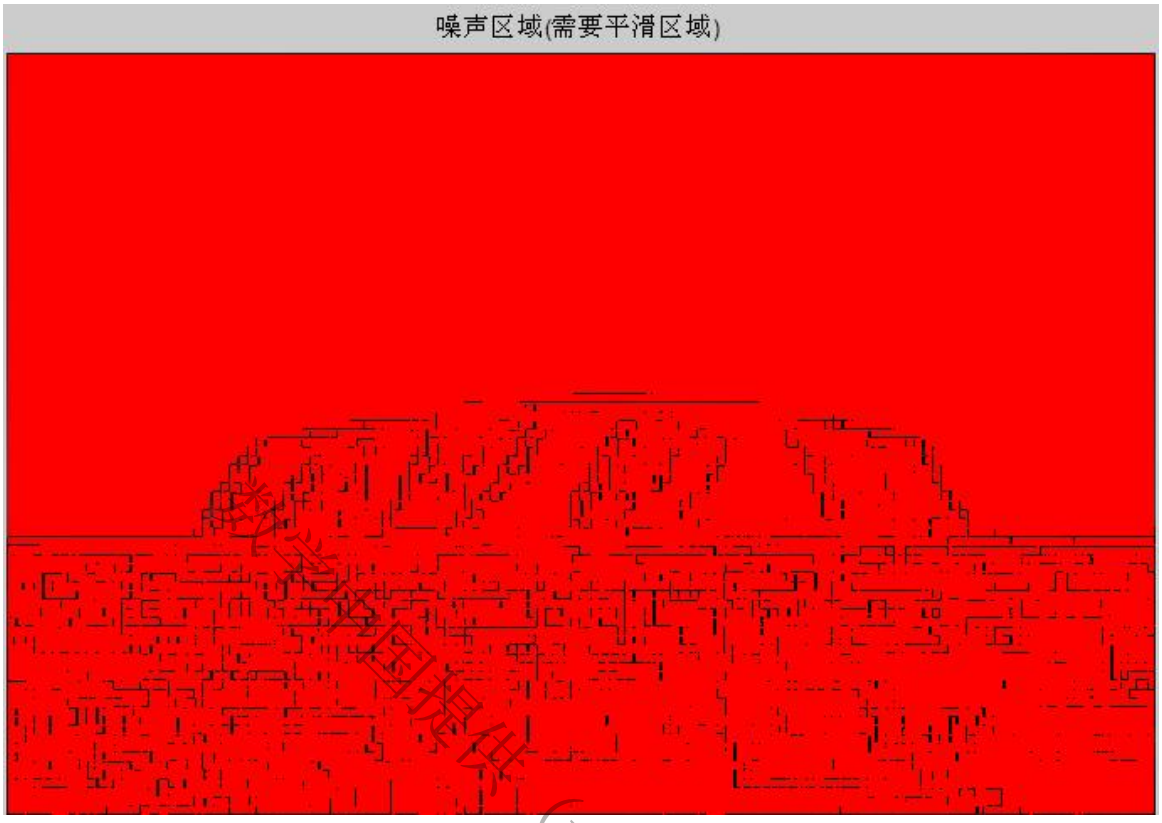


图 5-7

使用公式描述的方法，对各个级别的强度进行调制，m 取 0.3，调制后的边缘强度如图 5-8 所示。

图 5-8

在 0 到 X1 之间的区域认为是噪声区域，如图 5-9 所示，需要做适当平滑



。

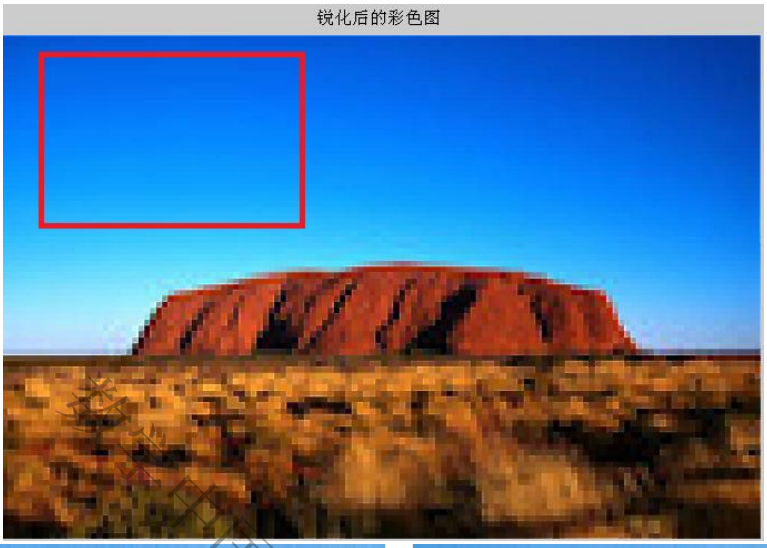
图 5-9

传统的锐化方法存在明显的弊端。原本清晰的边缘部分得到了过度加强，甚至产生对比度溢出，而对实际需要加强的不清晰边界的补偿不足，而且会成本的放大图像噪声。所以这里使用梯度域算法，提出了新的锐化方法。通过上面的效果图可以看出，新锐化算法的优势是很明显的。在如图所示区域，图像原本清晰度高的地方，通过抑制边缘梯度，防止图像边缘出现过度锐化。在如图所示的梯度相对较小区域，强化梯度叠加强度。在如图所示的平坦区域，不做梯度锐化，相反进行图像平滑，降低噪声。通过使用新的锐化方法，使图像原本清晰的边缘得以保持，进一步强化中等强度的边缘，平滑平坦区域，得到了很好的实际应用效果。

在图像中，可以明显看出，山的清晰度较高，而周围的清晰度较低。由于清晰度已经较高，锐化算法重点强化的应该是天空，草原等区域。对于天空部分的平坦区域，传统方法极大的放大了图像噪声，而新方法很好的抑制了锐化放大噪声的缺点。由于山清

晰度已经较高，锐化算法重点强化的应该是天空，草原等区域。放大图中红框显示的区域，对比传统算法和新算法的效果。

对于天空部分的平坦区域，传统方法极大的放大了图像噪声，而新方法很好的抑制了锐化放大噪声的缺点。下面通过放大图进行对比，说明新方法对噪声的抑制效果。



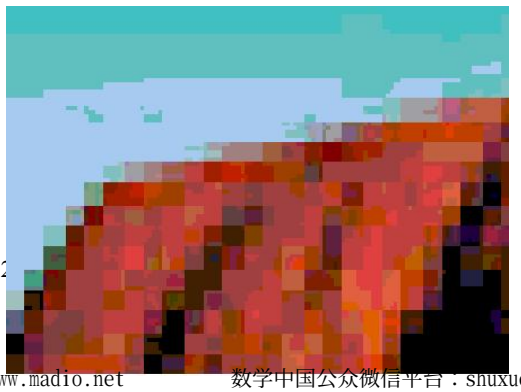
传统锐化方法噪声局部放大图

新锐化方法噪声局部放大图

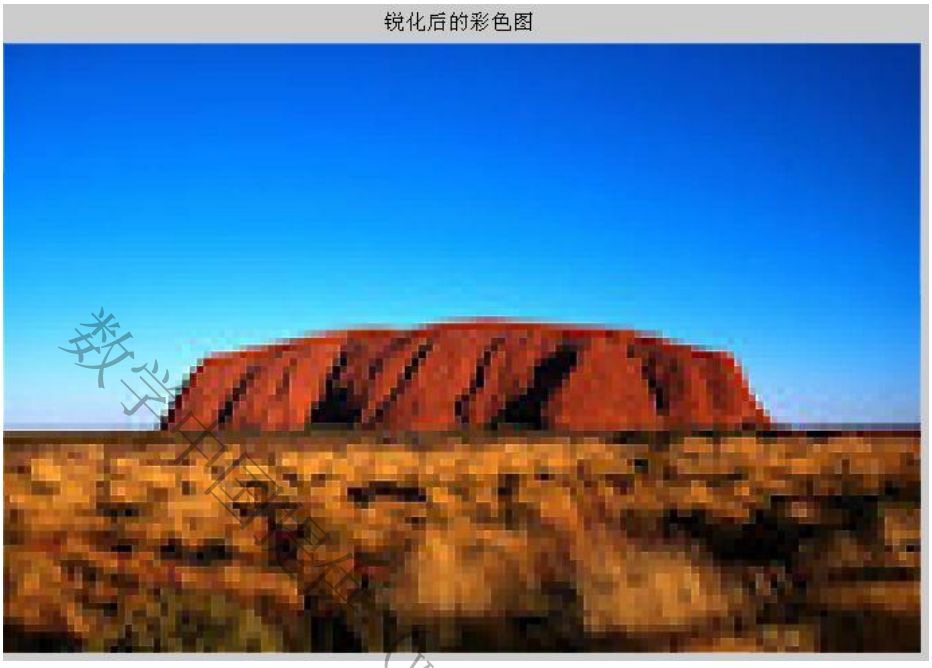
对比图可以发现新方法对噪声的抑制非常明显，取得了非常好的实际效果。

原局部放大图

传统锐化方法局部放大图



新锐化方法局部放大图



锐化后的彩色图

§6 模型的优缺点

§6.1 优点

领域平均法算法简单，对马赛克图像能得到有效的处理；

梯度锐化法对于的平坦区域，很好的抑制了锐化放大噪声的缺点。这个模型在噪声处理方面有优点，但在近处观察时还是让人产生一种失真感，如果与邻域平均法处一同使用在不同放大比例下，那么图像的视觉质量将会有很大提高。

§ 6.2 缺点

通常模板不允许移出边界，因此处理后的图像会比原来图像小；会会造成图像中物体边缘的模糊

虽然用邻域平均法处理矢量图可以在查看时对放大缩小的图像进行平滑处理，消除锯齿，但还不能保存得到不失真的矢量图，现今也存在很多直接对矢量图进行放大缩小且保存矢量图的软件，例如 Photoshop，Coreldrew，Flash 等。因此，今后将继续研究，希望能够达到这些软件的效果。

对于数字图像处理系统，梯度域算法必定会得到越来越广阔的应用。具体到本文的内容，由于时间和水平的关系编写得未尽完善，可以考虑的有价值的改进工作有：各种梯度快速计算方法，锐化参数自适应算法，车道线分割自适应算法等，使它更接近于实

际应用。

§7 参考文献

[1] 张德丰，MATLAB 数字图像处理[M]，北京：机械工业出版社，2009。

[2] 张德丰，MATLAB 数字图像处理[M]，1 版北京：人民邮电出版社，2007-01-01

[3] fatta,.generalized perona malik equation for image processing[J].ieee signal processing letter,1999.6(7)

[4] perez p.gangnet m.blake a poisson image editing[J]. IEEE Transactions on Image Processing.2003(3)

[5] A. Agarwala, M. Dontcheva, M. Agrawala.. Interactive digital photomontage[J].In Proc.SIGGRAPH04, 2004.8.

[6] A. Zomet and Y. Weiss. Seamless image stitching in the gradient domain[R].In Proc. of the European Conference on Computer Vision (ECCV04), Prague, Czech Republic.2006.5

[7] 章卫祥，周秉锋. 一种基于梯度域的彩色图像转灰度图像的方法[J].影像技术.2007.3

[8] 许录平.数字图像处理[M].北京.科学出版社.2007.10

[9] 阮秋琦.数字图像处理学[M].北京:电子工业出版社.2007

[10] Rafael c Gonzalez richard e woods digital image processing(second edition)[M].peking. publishing house of electronics industry.2003.3

[11] 郑阿奇，曹戈.matlab 实用教程[M].北京.电子工业出版社.2007.8

[12] 卢允伟，陈友荣.基于拉普拉斯算法的图像锐化算法研究和实现[J].电脑知识与技术.2009.5(6)

[13] 吴海波,刘钊.基于拉普拉斯算子的彩色图像锐化处理[J].电脑开发与应用.2008(09)

[14] 刘永勤，刘月月.基于 Verilog 的数字图像锐化研究和实现[J].科学技术与工程.2009.9(18)

[15] 赵庆军，胡青泥.基于局部图像锐化的自适应模糊边缘检测算法[J].电脑开发与应用.2004.17(9)

[16] 齐欣. 数码影像技术操作篇之六十二数码图像锐化方法[J].照相机.2009.1(2)

[17] 葛仕明，程义民，李杰，张玲.基于梯度场的拼接缝消除方法[J].计算机辅助设计与图形学学报.2007.2

[18] 王俊文，张琪，刘光杰.利用异常边缘进行图像锐化篡改取证[J].解放军理工大学学报.2009.10(3)

[19] 曾嘉亮.基于边缘检测的图像锐化算法[J].现代电子技术.2006.29(12)

[20] selim esedoglu.an analysis of the perona-malik scheme[J].comm pure appl math.2001.54(12)

[21] zhu,l. yang,y. haker,s tanenbaum,a. An Image Morphing Technique Based on Optimal Mass Preserving Mapping[J]. IEEE Transactions on Image Processing.2007.16(6)

§8 附录

程序1-1

```
clear all;
I=imread('局部1.bmp');
h=ones(5,5)/25;
I2=imfilter(I,h);
I3=imfilter(I2,h);
I4=imfilter(I3,h);
I5=imfilter(I4,h);
I6=imfilter(I5,h);
subplot(231);
 imshow(I);
title('原图')
subplot(232); imshow(I2);
title('第1次用5*5领域平均运算的平滑图像')
subplot(233); imshow(I3);
title('第2次用5*5领域平均运算的平滑图像')
subplot(234); imshow(I4);
title('第3次用5*5领域平均运算的平滑图像')
subplot(235); imshow(I5);
title('第4次用5*5领域平均运算的平滑图像')
subplot(236); imshow(I6);
title('第5次用5*5领域平均运算的平滑图像')
```

程序1-2：

```
clear all;
I=imread('1.bmp');
h=ones(5,5)/25;
I2=imfilter(I,h);
I3=imfilter(I2,h);
I4=imfilter(I3,h);
I5=imfilter(I4,h);
I6=imfilter(I5,h);
figure, imshow(I);
title('原图第一次用5*5领域平均运算的平滑图像')

figure, imshow(I2);
title('第一次用5*5领域平均运算的平滑图像')
figure, imshow(I3);
title('第2次用5*5领域平均运算的平滑图像')
figure, imshow(I4);
```

```
title('第3次用5*5领域平均运算的平滑图像')
figure, imshow(I5);
title('第4次用5*5领域平均运算的平滑图像')
figure, imshow(I6);
title('第5次用5*5领域平均运算的平滑图像')
```

程序 1-3 The gradient histogram

```
i=imread('yuan.jpg');
i=rgb2gray(i);
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(i), hy, 'replicate');   %求 x,y 方向梯度
Ix = imfilter(double(i), hx, 'replicate');

arctan=atan(Iy./Ix);   %范围在-pi/2---pi/2 之间
arctan=arctan+pi/2;    %转到 0-pi 之间
arctan(isnan(arctan)==1 & Iy~=0)=max(arctan(:));      %将 Ix =0 的统一设成最大 pi
arctan=fix(arctan/0.1);   %量化成 32 级，以统计直方图
temp=arctan(isnan(arctan)==0);
unI = sort(unique(temp));   %统计 arctan 中有多少种不同的值
nbins=length(unI);
histo=hist(temp(:),unI);
figure,stem(1:nbins,histo);
```

程序 1-4

```
close all

I = imread('1 灰度图.jpg');

hy = fspecial('sobel');

hx = hy';

Iy = imfilter(double(I), hy, 'replicate');

Ix = imfilter(double(I), hx, 'replicate');

gradmag = sqrt(Ix.^2 + Iy.^2);
figure('units', 'normalized', 'position', [0 0 1 1]);
subplot(1, 2, 1); imshow(I,[]), title('灰度增强图');
subplot(1, 2, 2); imshow(gradmag/255,[]), title('梯度幅值图像');
imwrite(gradmag,'2 梯度幅值图像.jpg')%保存图像为文件
```

程序 1-5

```
close all;
tic;

% Open UI
```

```
[filename, pathname] = uigetfile('*.bmp', 'Pick up a true color bmp file');
if isequal(filename, 0)||isequal(pathname, 0)
    return;
end
% Read in file
Image = imread(fullfile(pathname, filename));
figure;
imshow(Image);
title('Robinson 算子测试图');

Gray = Image(:,:,1);
Gray = double(Gray);

[m,n] = size(Gray);
% GradY = zeros(m, n);
h = [-1,-2,-1;0,0,0; 1,2,1];
image90 = zeros(m, n);
for r=2:1:m-2
    for c=2:1:n-2
        rb = int32(r-1);
        re = int32(r+1);
        cb = int32(c-1);
        ce = int32(c+1);
        GrayLocal = Gray(rb:re, cb:ce);
        GrayLocalFilter = GrayLocal.*h;
        image90(r, c) = sum(GrayLocalFilter(:));
    end
end
Min=min(image90(:));
Max=max(image90(:));
ImageConvert90 = uint8((image90 - Min)*255.0/(Max-Min));
figure,imshow(ImageConvert90);
title('270 度 ');

h = [-1,0,1;-2,0,2; -1,0,1];
image0 = zeros(m, n);
for r=2:1:m-2
    for c=2:1:n-2
        rb = int32(r-1);
        re = int32(r+1);
        cb = int32(c-1);
        ce = int32(c+1);
        GrayLocal = Gray(rb:re, cb:ce);
        GrayLocalFilter = GrayLocal.*h;
        image0(r, c) = sum(GrayLocalFilter(:));
    end
end
Min=min(image0(:));
Max=max(image0(:));
```

```
ImageConvert0 = uint8((image0 - Min)*255.0/(Max-Min));
figure,imshow(ImageConvert0);
title('0 度 ');
h = [0,1,2;-1,0,1; -2,-1,0];
% h = [2,1,0;1,0,-1; 0,-1,-2];
image270 = zeros(m, n);
for r=2:1:m-2
    for c=2:1:n-2
        rb = int32(r-1);
        re = int32(r+1);
        cb = int32(c-1);
        ce = int32(c+1);
        GrayLocal = Gray(rb:re, cb:ce);
        GrayLocalFilter = GrayLocal.*h;
        image270(r, c) = sum(GrayLocalFilter(:));
    end
end
Min=min(image270(:));
Max=max(image270(:));
ImageConvert270 = uint8((image270 - Min)*255.0/(Max-Min));
figure,imshow(ImageConvert270);
title('45 度 ');

h = [1,0,-1;2,0,-2; 1,0,-1];
image180 = zeros(m, n);
for r=2:1:m-2
    for c=2:1:n-2
        rb = int32(r-1);
        re = int32(r+1);
        cb = int32(c-1);
        ce = int32(c+1);
        GrayLocal = Gray(rb:re, cb:ce);
        GrayLocalFilter = GrayLocal.*h;
        image180(r, c) = sum(GrayLocalFilter(:));
    end
end
Min=min(image180(:));
Max=max(image180(:));
ImageConvert180 = uint8((image180 - Min)*255.0/(Max-Min));
figure,imshow(ImageConvert180);
title('180 度 ');

h = [1,2,1;0,0,0; -1,-2,-1];
image45 = zeros(m, n);
for r=2:1:m-2
    for c=2:1:n-2
        rb = int32(r-1);
        re = int32(r+1);
        cb = int32(c-1);
```

```
            ce = int32(c+1);
            GrayLocal = Gray(rb:re, cb:ce);
            GrayLocalFilter = GrayLocal.*h;
            image45(r, c) = sum(GrayLocalFilter(:));
        end
    end
    Min=min(image45(:));
    Max=max(image45(:));
    ImageConvert45 = uint8((image45 - Min)*255.0/(Max-Min));
    figure,imshow(ImageConvert45);
    title('90 度 ');

    h = [-2,-1,0;-1,0,1; 0,1,2];
    image315 = zeros(m, n);
    for r=2:1:m-2
        for c=2:1:n-2
            rb = int32(r-1);
            re = int32(r+1);
            cb = int32(c-1);
            ce = int32(c+1);
            GrayLocal = Gray(rb:re, cb:ce);
            GrayLocalFilter = GrayLocal.*h;
            image315(r, c) = sum(GrayLocalFilter(:));
        end
    end
    Min=min(image315(:));
    Max=max(image315(:));
    ImageConvert315 = uint8((image315 - Min)*255.0/(Max-Min));
    figure,imshow(ImageConvert315);
    title('315 度 ');

    h = [2,1,0;1,0,-1; 0,-1,-2];
    image225 = zeros(m, n);
    for r=2:1:m-2
        for c=2:1:n-2
            rb = int32(r-1);
            re = int32(r+1);
            cb = int32(c-1);
            ce = int32(c+1);
            GrayLocal = Gray(rb:re, cb:ce);
            GrayLocalFilter = GrayLocal.*h;
            image225(r, c) = sum(GrayLocalFilter(:));
        end
    end
    Min=min(image225(:));
    Max=max(image225(:));
    ImageConvert225 = uint8((image225 - Min)*255.0/(Max-Min));
    figure,imshow(ImageConvert225);
    title('135 度 ');
```

```
h = [0,-1,-2;1,0,-1; 2,1,0];
image135 = zeros(m, n);
for r=2:1:m-2
    for c=2:1:n-2
        rb = int32(r-1);
        re = int32(r+1);
        cb = int32(c-1);
        ce = int32(c+1);
        GrayLocal = Gray(rb:re, cb:ce);
        GrayLocalFilter = GrayLocal.*h;
        image135(r, c) = sum(GrayLocalFilter(:));
    end
end
Min=min(image135(:));
Max=max(image135(:));
ImageConvert135 = uint8((image135 - Min)*255.0/(Max-Min));
figure,imshow(ImageConvert135);
title('225 度 ');
```

程序 1-6

```
close all;
clc;
tic;
% Open UI
[filename, pathname] = uigetfile('xx.bmp', 'Pick up a true color bmp file');
if isequal(filename, 0)||isequal(pathname, 0)
    return;
end
% Read in file
Image = imread(fullfile(pathname, filename));
figure;
imshow(Image);
title('tzmcm 图片');
R = Image(:, :, 1);
G = Image(:, :, 2);
B = Image(:, :, 3);
figure,imshow(R);
title('R 通道图像');
figure,imshow(G);
title('G 通道图像');
figure,imshow(B);
title('B 通道图像');
% [m,n]=size(R);
% yuv=zeros(m,n,3,'uint8');
yuv=rgb2ycbcr(Image);
Y= yuv(:, :, 1);
U= yuv(:, :, 2);
V= yuv(:, :, 3);
```

```
figure,imshow(Y);
title('Y 通道图像');
figure,imshow(U);
title('U 通道图像');
figure,imshow(V);
title('V 通道图像');
h = [-1,0,-1,0,-1;
        -1,0,8,0,-1;
        -1,0,-1,0,-1];
image = imfilter(Y,h/8);
figure,imshow(image);
title('亮度通道梯度图片');
imageadd = image+Y;
figure,imshow(imageadd);
title('亮度通道直接锐化图片');

% new method
% high    suppress
 idx = find(image>40);
 imageGradAdd = image;
 imageGradAdd(idx) = imageGradAdd(idx)*0.3;
 % displey
 red = image;
 red(idx)=255;
 green = image;
 green(idx) = 0;
 blue = image;
 blue(idx)=0;
 RGB(:,:,1) = red;
 RGB(:,:,2) = green;
 RGB(:,:,3) = blue;
 figure,imshow(RGB);
 title('高梯度区域');

 % middle
 idx = find(15<=image &image<40);
 imageGradAdd(idx) = imageGradAdd(idx)*1.3;
 title('中梯度区域加强图片');
 red = image;
 red(idx)=255;
 green = image;
 green(idx) = 0;
 blue = image;
 blue(idx)=0;
 RGB(:,:,1) = red;
 RGB(:,:,2) = green;
 RGB(:,:,3) = blue;
```

```
figure,imshow(RGB);
title('中等梯度区域');

% low constrast keep same
idx = find(8<image &image<=15);
red = image;
red(idx)=255;
green = image;
green(idx) = 0;
blue = image;
blue(idx)=0;
RGB(:,:,1) = red;
RGB(:,:,2) = green;
RGB(:,:,3) = blue;
figure,imshow(RGB);
title('低梯度区域');

% mean
idx = find(0<=image &image<=8);
red = image;
red(idx)=255;
green = image;
green(idx) = 0;
blue = image;
blue(idx)=0;
RGB(:,:,1) = red;
RGB(:,:,2) = green;
RGB(:,:,3) = blue;
figure,imshow(RGB);
title('噪声区域(需要平滑区域)');

% display edge map after look up table
figure,imshow(imageGradAdd);
title('梯度调制图');

% mean noise region
filter = ones(3, 3)/9.0;
YMean = imfilter(Y,filter);
Y(idx) = YMean(idx);
% sharpness original image and convert to rgb color space
sharpnessNew = Y+imageGradAdd;
figure,imshow(sharpnessNew);
title('亮度锐化图');
yuv(:,:,1)=sharpnessNew;
yuv(:,:,2)=U;
yuv(:,:,3)=V;
RGBSharpness= ycbcr2rgb(yuv);
figure,imshow(RGBSharpness);
title('锐化后的彩色图');
```

程序 1-7

```
clc
clear
close all
ima=imread('tzmcmc.bmp');%读入图像
if isrgb(ima)
    ima=rgb2gray(ima);%如果是彩色图像，则转为灰度图像
end
ima=double(ima);

bw1 = edge(ima,'sobel'); %sobel 算子锐化
figure;subplot(121);imshow(uint8(ima));title('原始图像');%图像显示
subplot(122);imshow(bw1);title('sobel 算子锐化');

bw2 = edge(ima,'prewitt');%prewitt 算子锐化
figure;subplot(121);imshow(uint8(ima));title('原始图像');
subplot(122);imshow(bw2);title('prewitt 算子锐化');

bw3 = edge(ima,'roberts');%roberts 算子锐化
figure;subplot(121);imshow(uint8(ima));title('原始图像');
subplot(122);imshow(bw3);title('roberts 算子锐化');

bw4 = edge(ima,'log');%log 算子锐化
figure;subplot(121);imshow(uint8(ima));title('原始图像');
subplot(122);imshow(bw4);title('log 算子锐化');

bw5 = edge(ima,'canny');canny 算子锐化
figure;subplot(121);imshow(uint8(ima));title('原始图像');
subplot(122);imshow(bw5);title('canny 算子锐化');

h1=fspecial('gaussian',[9 9]);%gaussian 低通滤波器锐化
bw6 = imfilter(ima,h1);
figure;subplot(121);imshow(uint8(ima));title('原始图像');
subplot(122);imshow(uint8(bw6));title('gaussian 低通滤波器锐化');

h2=fspecial('laplacian');%laplacian 算子锐化
bw7 = imfilter(ima,h1);
figure;subplot(121);imshow(uint8(ima));title('原始图像');
subplot(122);imshow(uint8(bw7));title('laplacian 算子锐化');
```