$$s = T(r)$$

式中，r 表示图像 f 中相应点 (x, y) 的灰度，s 表示图像 g 中相应点 (x, y) 的灰度。

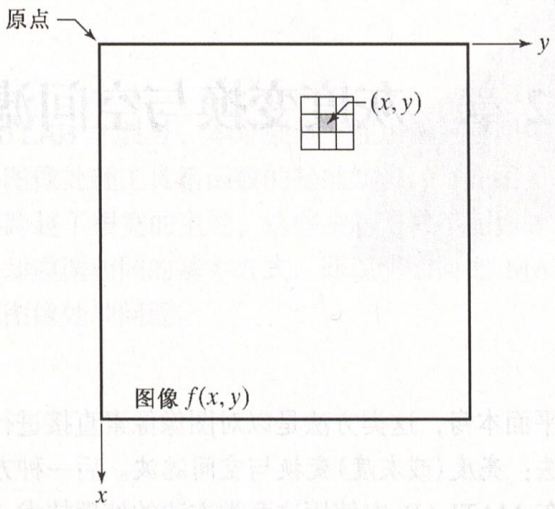
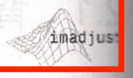


图 2.1 一幅图像中以点 (x, y) 为中心的 3×3 大小的邻域

2.2.1 函数 imadjust 和 stretchlim

函数 imadjust 是一个基本的图像处理工具箱函数，用于对灰度级图像进行灰度变换。该函数的一般语法格式为

```
g = imadjust(f, [low_in high_in], [low_out high_out], gamma)
```



如图 2.2 所示，该函数将图像 f 中的灰度值映射为图像 g 中的新值，即将 low_in 至 high_in 之间的值映射到 low_out 至 high_out 之间的值。low_in 以下与 high_in 以上的值则被截去，即 low_in 以下的值映射为 low_out，high_in 以上的值映射为 high_out。输入图像可以是 uint8 类、uint16 类、single 类或 double 类，输出图像与输入图像属于同一类。对于函数 imadjust，除了 f 和 gamma，其所有输入值都被限定在 0 和 1 之间，而与 f 的类别无关。例如，如果 f 属于 uint8 类，那么函数 imadjust 会通过把值乘以 255 来确定将要使用的实际值。对[low_in high_in]或[low_out high_out]使用空矩阵([ ])，会得到默认值[0 1]。如果 high_out 小于 low_out，那么输出灰度将被反转。

回忆 1.7.8 节的讨论可知，函数 mat2gray 可用于将一幅图像转换为 double 类，并将其灰度标定到范围[0, 1]，而这与输入图像的类别无关。

参数 gamma 指定从图像 f 中的灰度值映射生成图像 g 的曲线的形状。若 gamma 值小于 1，则映射被加权至较高(较亮)的输出值，如图 2.2(a)所示。若 gamma 的值大于 1，则映射被加权至较低(较暗)的输出值。若函数参量缺省，则 gamma 默认为 1(线性映射)。

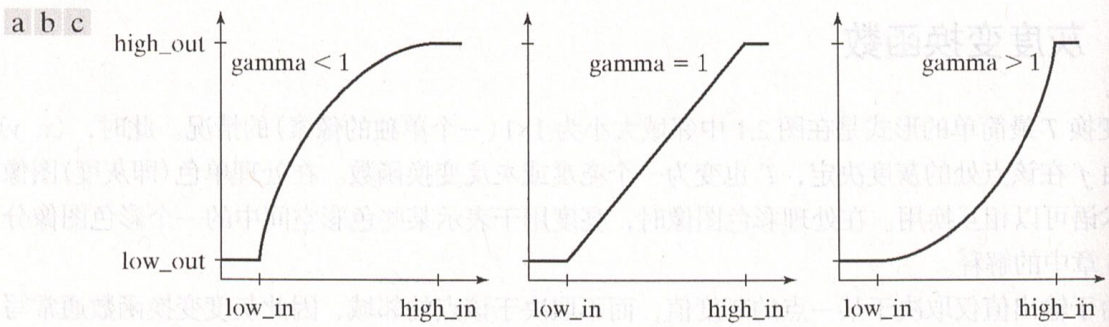


图 2.2 函数 imadjust 中各种可用的映射

使用函数 imadjust。

图 2.3(a)是一幅数字乳房图像 f，图像中显示了一处病灶，图 2.3(b)是使用如下命令得到的明暗反转(负片图像)：

```
>> g1 = imadjust(f, [0 1], [1 0]);
```

生成得明暗反转图像的过程对于增强嵌入在一大片黑色区域中的白色或灰色细节是非常有用的。例如，图 2.3(b)中就非常容易分析乳房的组织。图像的负片同样可以利用工具箱函数 imcomplement 得到：

```
g = imcomplement(f)
```

imcomplement

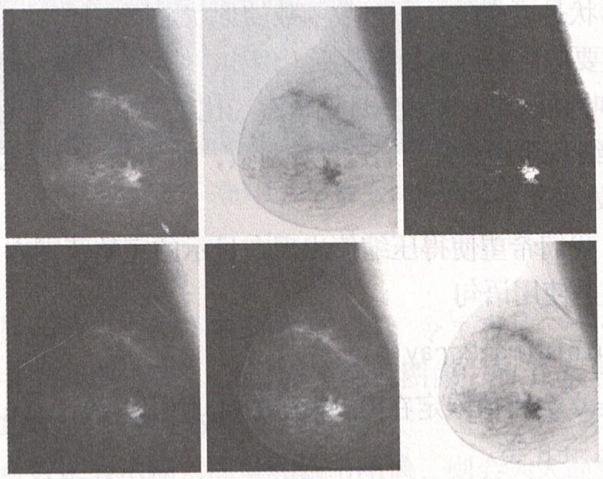
图 2.3(c)是使用如下命令的结果：

```
>> g2 = imadjust(f, [0.5 0.75], [0 1]);
```

输入包含 0.5 到 0.75 之间的灰度扩展到整个[0, 1]范围。这种类型的处理对于强调感兴趣灰度区非常有用。最后，利用命令

```
>> g3 = imadjust(f, [ ], [ ], 2);
```

压缩灰度级的低端并扩展高端［见图 2.3(d)］，得到类似于图 2.3(c)的结果(增加了更多的灰色调)。



a b c
d e f

图 2.3　(a)原始数字乳房图像；(b)负片图像；(c)灰度扩展至范围[0.5, 0.75]后的结果；(d)使用 gamma = 2 增强图像后的结果；(e)和(f)使用函数 stretchlim 作为函数 imadjust 的一个自动输入的两种结果(原图像由 G. E. Medical Systems 公司提供)

有时，能够自动地使用函数 imadjust 而不必关心上面讨论的低参数或高参数是非常有用的。这时，可使用函数 stretchlim，其基本语法是

```
Low_High = stretchlim(f)
```

stretchlim

其中，Low_High 是一个两元素向量，该向量由一个低限和一个高限组成，用于实现对比度拉伸(该术语的定义见下一节)。默认情形下，Low_High 中的值指定灰度级，这些灰度级充满 f 中底部和顶部 1%的所有像素值。该结果以向量[low_in high_in]的形式用于函数 imadjust 中，如下所示：

```
>> g = imadjust(f, stretchlim(f), [ ]);
```

图 2.3(e)显示了对图 2.3(a)执行这一操作后的结果。请观察对比度的提升。类似地，图 2.3(f)是用如下命令得到的：

```
>> g = imadjust(f, stretchlim(f), [1 0]);
```

比较图 2.3(b)和图 2.3(f)，可以看到这一操作增强了负片图像的对比度。

最后，删除先前插入的填充：

```
>> [M, N] = size(f);
>> g = g((1:M) + m, (1:N) + n);
```

以便 g 的大小与 f 的大小相同。

一些通用的非线性滤波器可以通过其他 MATLAB 和工具箱函数实现，譬如 imfilter 和 ordfilt2(见 2.5.2 节)。例如，4.3 节中的函数 spfilt，在例 2.9 中通过 imfilter 和 MATLAB 函数 log 与 exp 实现几何平均滤波器。当这种实现可能时，性能通常会快得多，且所用内存也远小于 colfilt 所要求的一小部分。但在没有可以替换实现的情形下，函数 colfilt 仍是进行非线性滤波操作的最好选择。

2.5 图像处理工具箱的标准空间滤波器

本节将讨论由工具箱支持的线性和非线性空间滤波器。其他自定义滤波器函数将在 4.3 节中实现。

2.5.1 线性空间滤波器

工具箱支持许多预定义的二维线性空间滤波器，这些滤波器可通过函数 fspecial 得到，该函数生成一个滤波模板 w，语法为

```
w = fspecial('type', parameters)
```

其中，'type'指定滤波器的类型，parameters 进一步定义规定的滤波器。由 fspecial 生成的空间滤波器汇总于表 2.5 中，表中包括了每种滤波器的适用参数。

关于梯度，请参阅5.6.1 节和7.1.3 节

表 2.5 函数 **fspecial** 所支持的空间滤波器。表中的几个滤波器用于 7.1 节中的边缘检测

类 型	语法和参数
'average'	fspecial('average', [r c])。一个大小为 r×c 的矩形平均滤波器。默认值为 3×3。若由单个数代替[r c]，则表示这是一个正方形滤波器
'disk'	fspecial('disk', r)。一个半径为 r 的圆形平均滤波器(包含在 2r + 1 大小的正方形内)。默认半径为 5
'gaussian'	fspecial('gaussian', [r c], sig)。一个大小为 r×c、标准偏差为 sig(正)的高斯低通滤波器。默认值为 3×3 和 0.5。若由单个数代替[r c]，则表示这是一个正方形滤波器
'laplacian'	fspecial('laplacian', alpha)。一个大小为 3×3 的拉普拉斯滤波器，其形状由 alpha 指定，alpha 是一个在[0, 1]范围内的数。alpha 的默认值为 0.2
'log'	fspecial('log', [r c], sig)。一个大小为 r×c、标准偏差为 sig(正)的高斯-拉普拉斯(LoG)滤波器。默认值为 5×5 和 0.5。若用单个数代替[r c]，则表示这是一个正方形滤波器
'motion'	fspecial('motion', len,theta)。当与一幅图像卷积时，输出一个滤波器来近似计算 len 个像素的线性运动(类似于相机与景物的关系)。运动的方向为 theta，单位为度，以水平方向为参考逆时针转动。默认值为 9 和 0，表示沿水平方向 9 个像素的运动
'prewitt'	fspecial('prewitt')。输出一个大小为 3×3 的 Prewitt 滤波器 wv，用来近似计算垂直梯度。水平梯度模板可以通过将结果转置来得到：wh=wv'
'sobel'	fspecial('sobel')。输出一个大小为 3×3 的 Sobel 滤波器 sv，用来近似计算垂直梯度。水平梯度滤波器可以通过将结果转置来得到：sh = sv'
'unsharp'	fspecial('unsharp', alpha)。输出一个大小为 3×3 的非尖锐的滤波器。alpha 控制形状，其值必须在[0, 1]范围内，默认值为 0.2

例 2.10 使用函数 imfilter 实现拉普拉斯滤波器。

下面通过使用一个拉普拉斯滤波器增强图像，来说明函数 fspecial 和 imfilter 的用法。图像 $f(x,y)$ 的拉普拉斯算子定义为 $\nabla^2 f(x,y)$：

$$\nabla^2 f(x,y)=\frac{\partial^2 f(x,y)}{\partial x^2}+\frac{\partial^2 f(x,y)}{\partial y^2}$$

二阶导数的常用数字近似为

$$\frac{\partial^2 f(x,y)}{\partial x^2}=f(x+1,y)+f(x-1,y)-2f(x,y)$$

关于梯度，请参阅 5.6.1 节和 7.1.3 节。

类似地

$$\frac{\partial^2 f(x,y)}{\partial y^2}=f(x,y+1)+f(x,y-1)-2f(x,y)$$

因此有

$$\nabla^2 f(x,y)=[f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)]-4f(x,y)$$

该表达式可应用于图像中的所有点，方法是使用下面的空间模板与图像卷积：

$$\begin{matrix}0 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & 0\end{matrix}$$

基于二阶导数的一种可选定义是考虑对角线元素，且可以使用下面的模板实现：

$$\begin{matrix}1 & 1 & 1\\ 1 & -8 & 1\\ 1 & 1 & 1\end{matrix}$$

两个导数有时可以用这里所示的相反符号定义，得到与前面两个模板正好相反的结果。

使用拉普拉斯算子进行图像增强的基本公式为

$$g(x,y)=f(x,y)+c[\nabla^2 f(x,y)]$$

其中，$f(x,y)$ 为输入图像，$g(x,y)$ 为增强后的图像；若模板的中心系数为正，则 c 为 1，否则 c 为 -1（Gonzalez and Woods[2008]）。因为拉普拉斯是微分算子，因此它将使图像尖锐化，但会使恒定区域为 0。把结果与原始图相加可恢复灰度级色调。

函数 `fspecial('laplacian', alpha)` 实现一个更为通用的拉普拉斯模板：

$$\begin{matrix}\frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha}\\ \frac{1-\alpha}{1+\alpha} & \frac{-4}{1+\alpha} & \frac{1-\alpha}{1+\alpha}\\ \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha}\end{matrix}$$

它可以对增强的结果进行精细的调整。但拉普拉斯算子的主要应用则基于刚刚讨论的这两种模板。

下面应用拉普拉斯算子来增强图 2.17(a) 所示的图像。这是一幅略显模糊的月球北极图像。此时，对图像的增强操作是锐化图像，同时尽可能地保留其灰度层次。首先，生成并显示该拉普拉斯滤波器：

```
>> w = fspecial('laplacian', 0)
w =
    0.0000    1.0000    0.0000
    1.0000   -4.0000    1.0000
    0.0000    1.0000    0.0000
```

注意，该滤波器是 `double` 类的，其 `alpha = 0` 的形状是前面讨论过的拉普拉斯滤波器。我们可以很容易地人为规定其形状为

```
>> w = [0 1 0; 1 -4, 1; 0 1 0];
```

下面对输入图像 f [见图 2.17(a)] 应用 w，图像 f 是 uint8 类图像：

```
>> g1 = imfilter(f, w, 'replicate');
>> imshow(g1, [ ])
```

图 2.17(b)显示了结果图像。这一结果看起来是合理的，但存在一个问题：所有的像素都是正的。由于滤波器中心系数为负，因此我们通常希望得到的是一个带有正值和负值的拉普拉斯图像。但此时，f 是 uint8 类的。如前一节所述，imfilter 给出了与输入图像类相同的输出，所以负值被截掉了。在对图像 f 滤波前，通过将其转换为浮点数可解决这一问题：

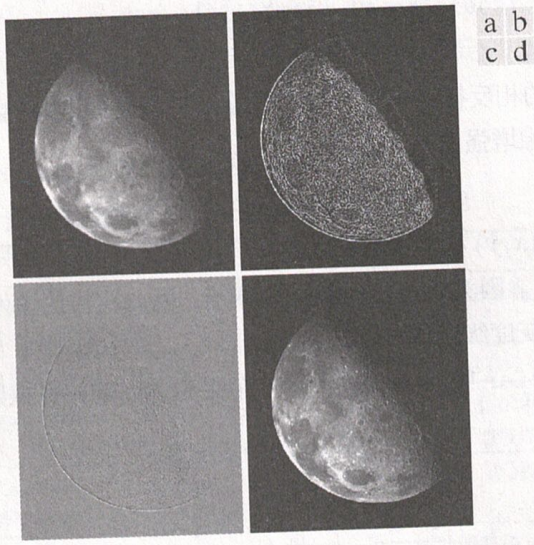
```
>> f2 = tofloat(f);
>> g2 = imfilter(f2, w, 'replicate');
>> imshow(g2, [ ])
```

图 2.17(c)显示的结果是拉普拉斯图像的典型外观。

最后，从原始图像中减去(因为中心系数为负值)拉普拉斯图像，以恢复失去的灰度层次：

```
>> g = f2 - g2;
>> imshow(g);
```

示于图 2.17(d)中的结果比原始图像要清晰。



a b
c d

图 2.17 (a)月球北极图像；(b)使用 uint8 格式经拉普拉斯滤波后的图像(因为 uint8 是无符号类型，输出中的负值被裁剪为 0)；(c)使用浮点格式获得的经拉普拉斯滤波后的图像；(d)从图(a)中减去图(c)所得到的增强后的结果(原图像由 NASA 提供)

例 2.11 手工指定滤波器及增强技术的比较。

增强问题常常需要工具箱之外的滤波器。拉普拉斯滤波器就是一个很好的例子。工具箱支持一个中心系数为 –4 的 3×3 拉普拉斯滤波器。通常，使用中心系数为 –8 、周围值均为 1 的 3×3 拉普拉斯滤波器可得到更为清晰的图像。本例的目的是手工实现这个滤波器，并比较使用这两种拉普拉斯方式得到的结果。

命令序列如下：

```
>> f = imread('Fig0217(a).tif');
>> w4 = fspecial('laplacian', 0); %Same as w in Example 2.10.
>> w8 = [1 1 1; 1 -8 1; 1 1 1];
>> f = tofloat(f);
>> g4 = f - imfilter(f, w4, 'replicate');
```