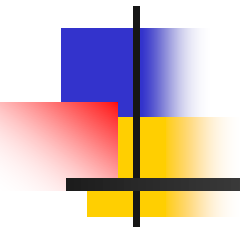


计算机视觉

图像处理基础知识

2019-3-4

数字图像处理基础知识

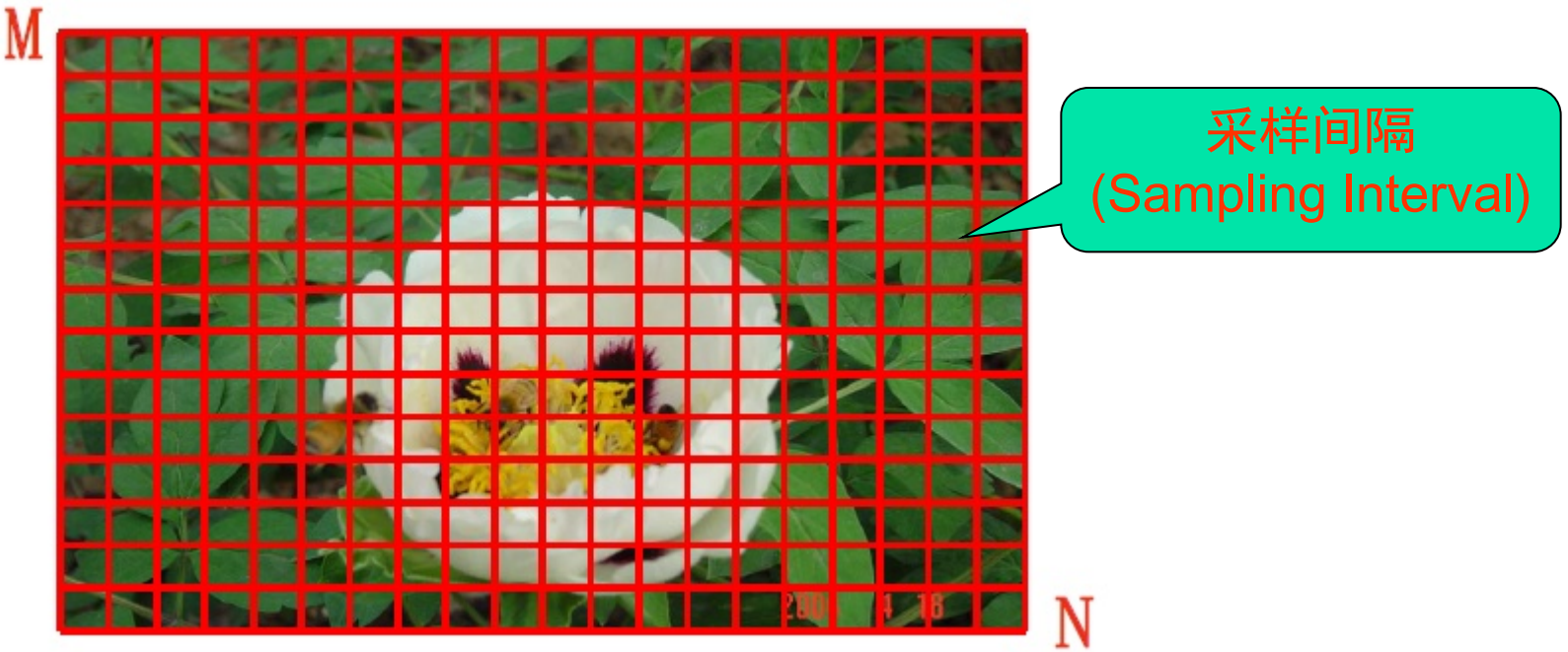
- 图像采样与量化
- 数字图像的表示和存储
- 数字图像的质量
- 像素间的一些基本关系

图像的采样和量化

- 大多数传感器的输出是连续电压
- 为了产生一幅数字图像，需要把连续的感知数据转化为数字形式
- 包括两种处理：取样和量化
- 取样：图像空间坐标(x, y)的数字化
- 量化：图像函数值f(x, y)的数字化

图像采样

- 图像采样：一个连续图像在每个采样点处被数字化。每个采样点对应于数字化图像的一个像素。
- 确定水平和垂直方向上的像素个数N和M。
- 图像采样决定了空间分辨率：单位物理空间用多少像素来表表征。



图像采样与图像质量的关系

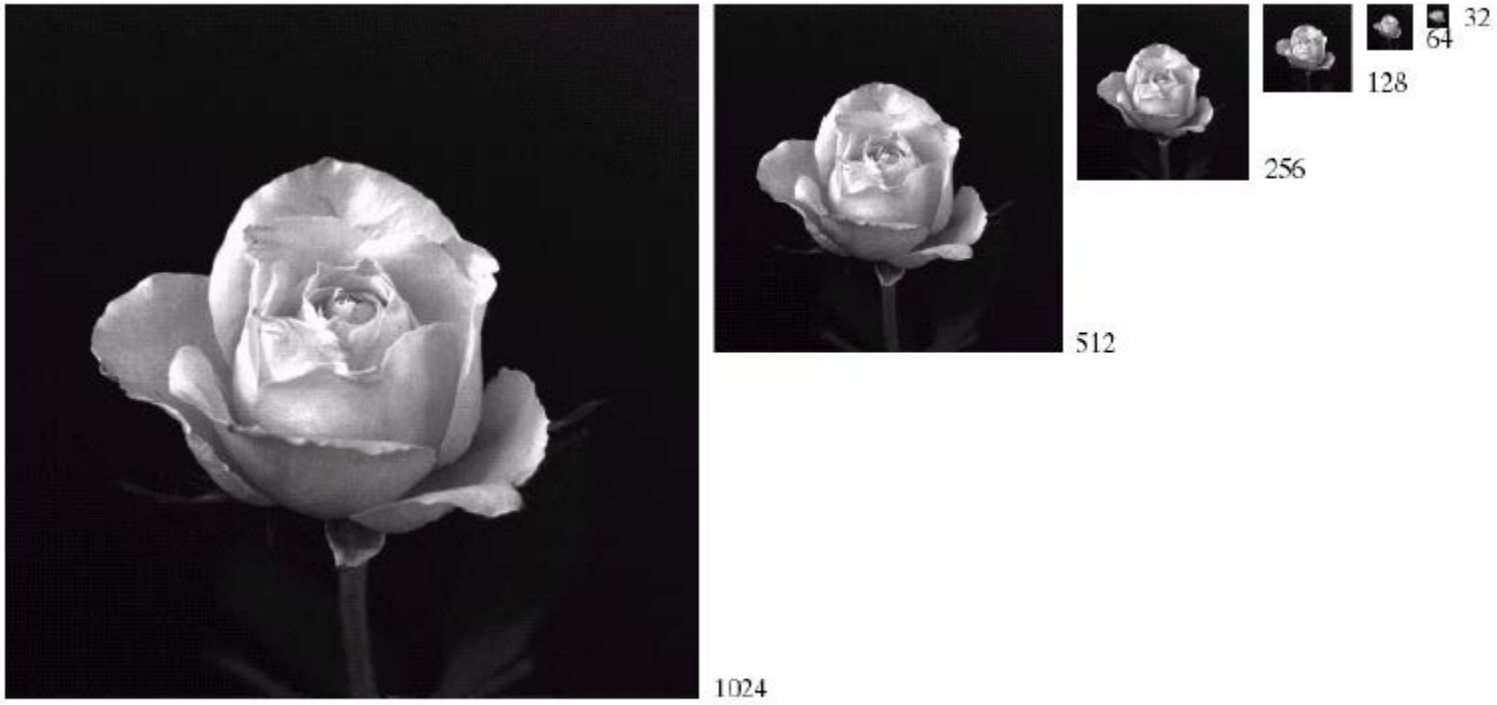
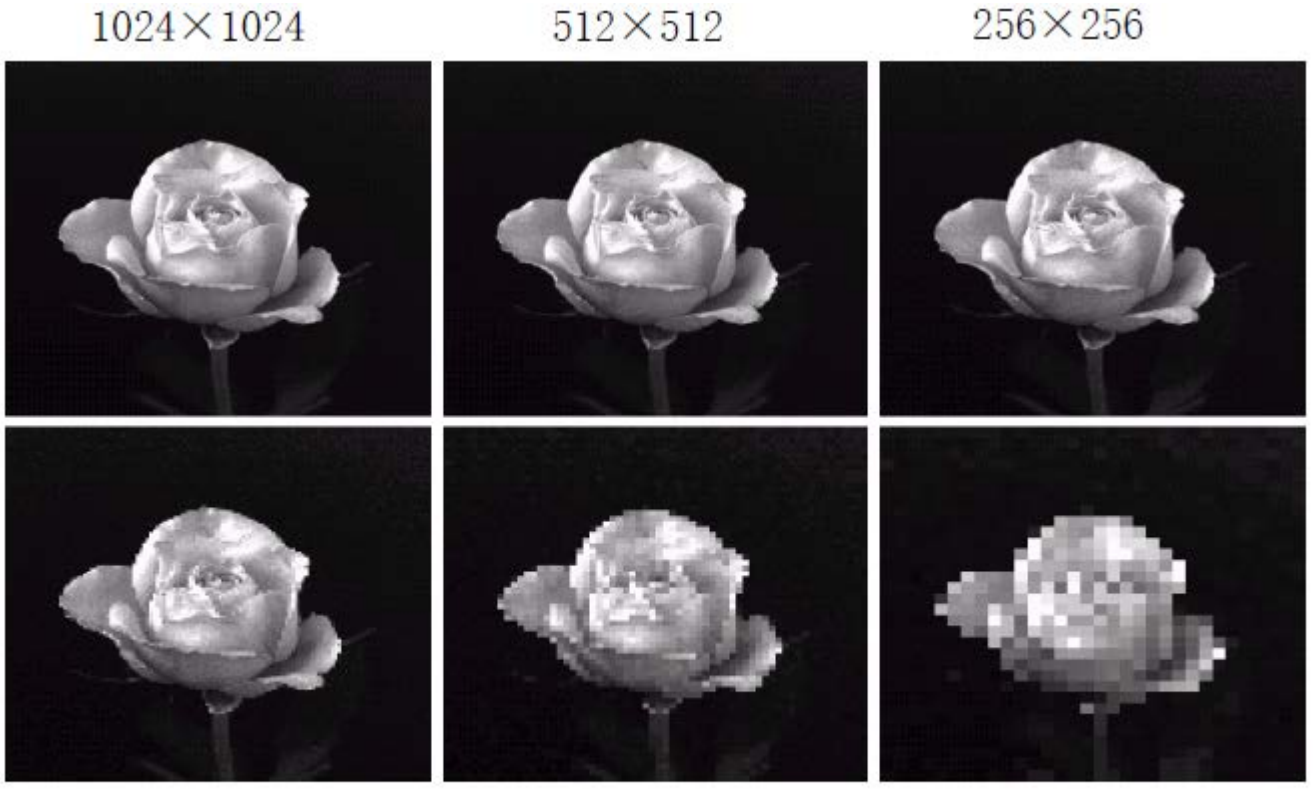


FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

图像采样与图像质量的关系

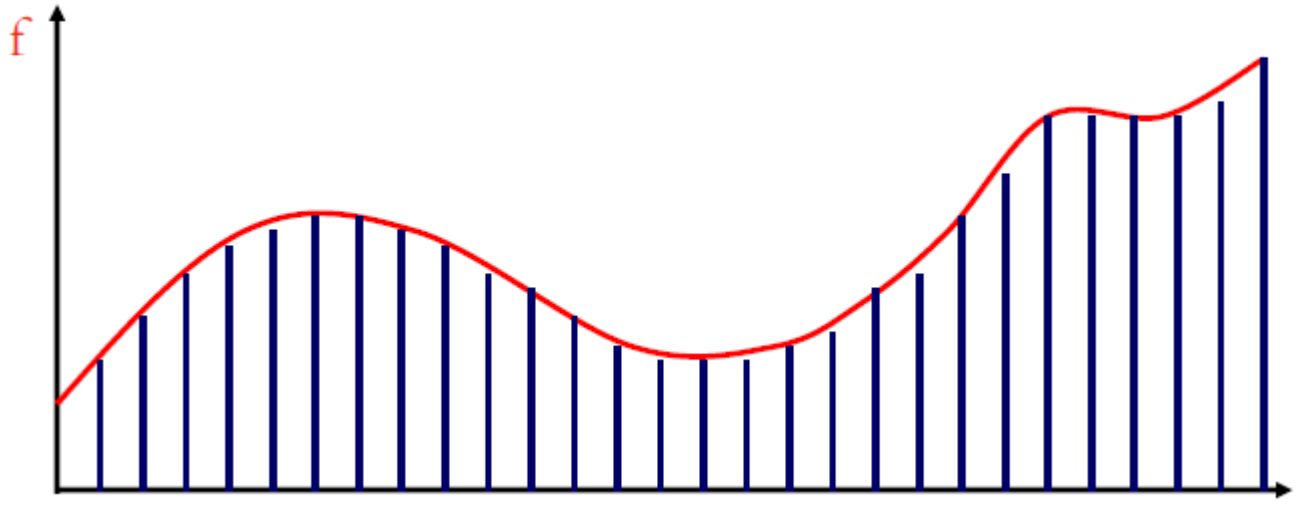


a b c
d e f

FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

图像量化

- 图像量化：给每一个像素的图像数值f(x,y) 用一个离散的数字（灰度值）来表示。
- 图像量化决定了灰度（或颜色）的分辨率。

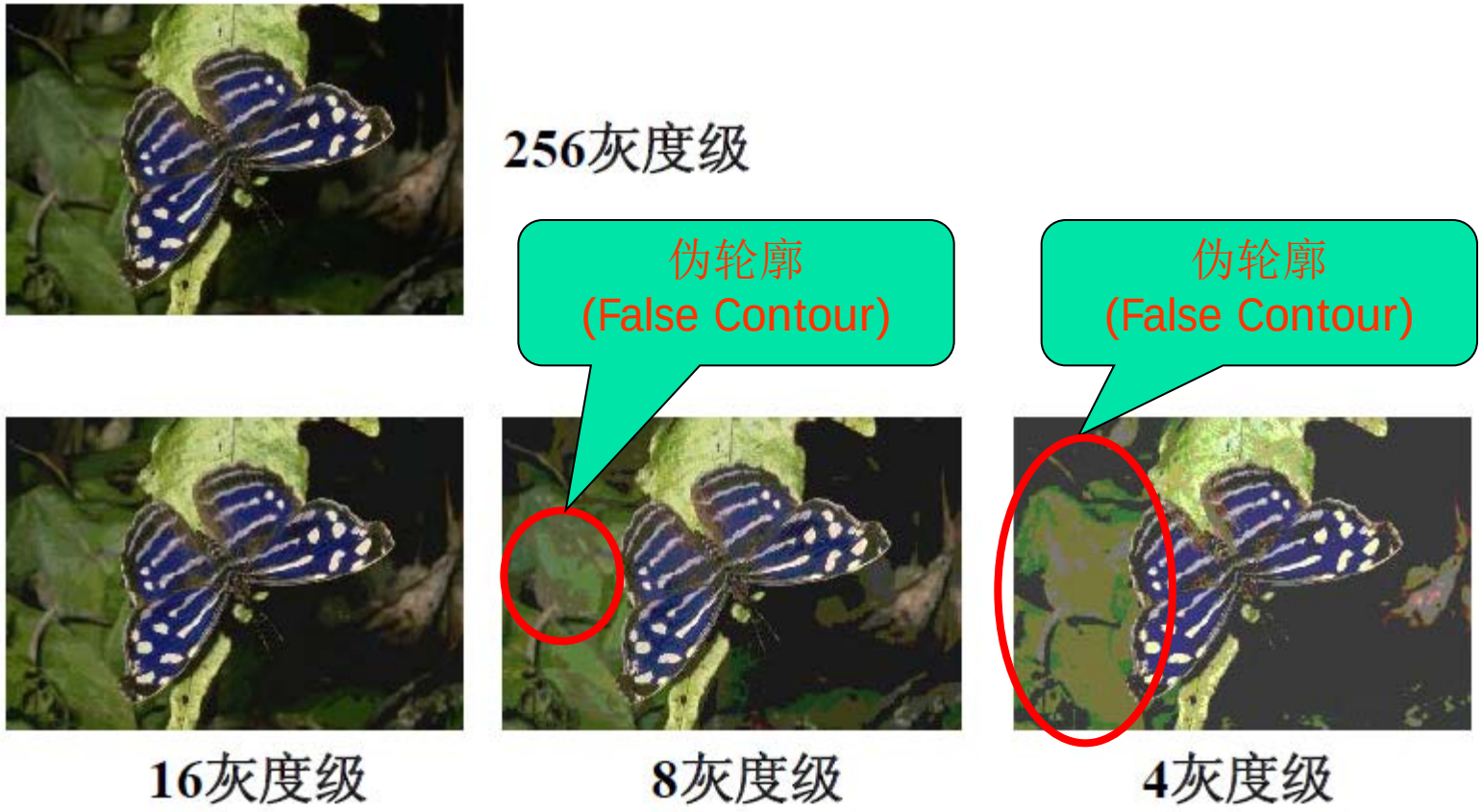


图像量化

- 大部分数字图像处理设备都采用K个等间隔的量化方式。
- 对于灰度图像而言，每个像素的亮度用一个数值来表示，该数值范围通常在0到255之间，0表示黑，255表示白，其它值表示处于黑白之间的灰度。
- 彩色图像可以用红、绿、蓝(RGB)三元组二维矩阵来表示。通常三元组的每个数值也是0到255之间，0表示相应的基色在该像素中没有，而255表示相应的基色在该像素中取得最大值。

256×256×256种颜色

图像量化与图像质量的关系



数字图像处理基础知识

- 图像采样与量化
- 数字图像的质量
- 数字图像的表示和存储
- 像素间的一些基本关系

数字图像的质量：层次

- 灰度级：表示像素明暗程度的整数量。

  例如：像素的量化范围为0～255，就称该图像为256个灰度级的图像。
- 层次：表示图像实际拥有的灰度级的数量。
- 图像数据的实际层次越多，视觉效果就越好。

数字图像的质量：层次

256个层次的图像

64个层次的图像

16个层次的图像

数字图像的质量：对比度

- 对比度：是指一幅图像中灰度反差的大小

  对比度＝最大亮度 / 最小亮度

数字图像的质量：清晰度

- 与清晰度相关的因素：
- 亮度
- 对比度
- 尺寸大小
- 颜色饱和度

影响清晰度的因素：亮度

原图

降低亮度

影响清晰度的因素：对比度

原图

降低对比度

影响清晰度的因素：尺寸大小

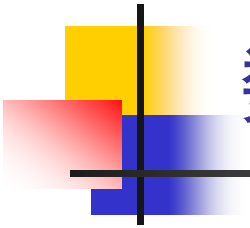
原图

缩小尺寸

影响清晰度的因素：颜色饱和度

原图

降低颜色饱和度



数字图像处理基础知识

- 图像采样与量化
- 数字图像的质量
- 数字图像的表示和存储
- 像素间的一些基本关系

数字图像的表示

- 二维离散的亮度函数： f(x,y)

  (x,y)说明图像像素的空间坐标

  函数值 f 代表了在点 (x,y) 处的像素灰度值。

- 二维矩阵： A[m,n]

  m,n 代表图像的宽和高

  矩阵元素 a(i, j) 的值表示图像在第 i 行、第 j 列的像素灰度值。

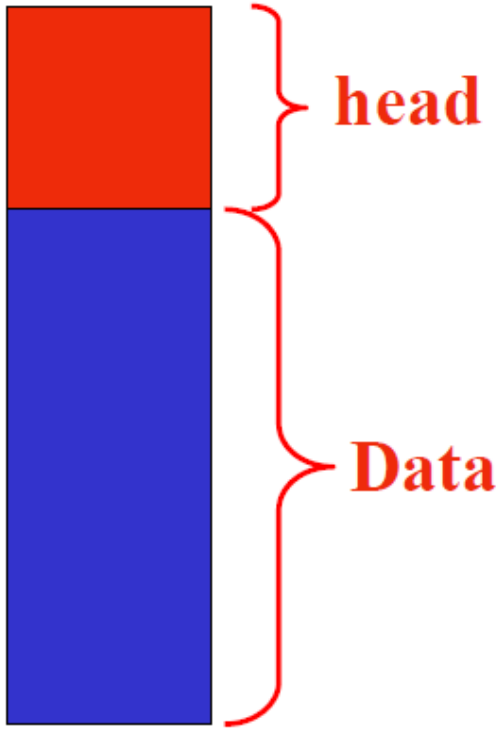
- RGB彩色图像的矩阵表示：

  三维矩阵A[m,n,3]

数字图像的存储格式

- 图像描述信息：如图像的高度和宽度等信息
- 图像数据：顺序存放的连续数据
- BMP（Bitmap）格式

1. 位图文件头
2. 位图信息头
3. 调色板
4. 图像数据



BMP格式：位图文件头

```
typedef struct tagBITMAPFILEHEADER{
    WORD    bfType;             //文件类型，必须是字符串"BM"
    DWORD   bfSize;             //指定文件大小
    WORD    bfReserved1;        //保留字，不考虑
    WORD    bfReserved2;        //保留字，不考虑
    DWORD   bfOffBits;          //从文件头到位图数据的偏移字节数
} BITMAPFILEHEADER;
```

BMP格式：位图信息头

```
typedef struct tagBITMAPINFOHEADER{
    DWORD   biSize;             //该结构的长度，40个字节
    LONG    biWidth;            //图像的宽度，单位是像素
    LONG    biHeight;           //图像的高度，单位是像素
    WORD    biPlanes;           //必须是1
    WORD    biBitCount          //颜色位数，如1，4，8，24
    DWORD   biCompression;      //压缩类型，如BI_RGB,BI_RLE4
    DWORD   biSizeImage;        //实际位图数据占用的字节数
    LONG    biXPelsPerMeter;    //水平分辨率
    LONG    biYPelsPerMeter;    //垂直分辨率
    DWORD   biClrUsed;          //实际使用的颜色数
    DWORD   biClrImportant;     //重要的颜色数
} BITMAPINFOHEADER;
```

像素/米

0表示都重要

BMP格式：图像数据

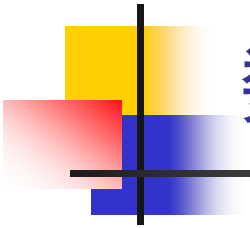
- 2色位图：图中只有2种颜色。所以一个像素只需要占用1位(bit)存储空间。相应的，一个字节(byte)可以表示8个像素。
- 16色位图：图中有16种颜色，所以一个像素需要占用4位(bit)存储空间。一个字节(byte)表示2个像素。
- 256色位图：图中有256种颜色，所以1个字节(byte)表示1个像素。
- 24位真彩色图：图中有2^24种颜色，所以3个字节(byte)表示1个像素。
- 问题：RGB颜色结构中，每个颜色通常使用256个灰度级来量化，那么如何表达2色、16色和25色？

BMP格式：调色板

```
typedef struct tagRGBQUAD{
    BYTE    rgbBlue;          //该颜色的蓝色分量
    BYTE    rgbGreen;         //该颜色的绿色分量
    BYTE    rgbRed;           //该颜色的红色分量
    BYTE    rgbReserved;      //保留值，不考虑
} RGBQUAD;
```

0～255

- 一个结构体变量表示一种颜色（占用 3 bytes）。
- 如果是16色位图，调色板需要占用16×3 bytes。
- 如果是16色位图，在其位图数据中，4位的数据为调色板中的颜色索引值（0～15）。
- 24位真彩色图中，不需要调色板。



数字图像处理基础知识

- 图像采样与量化
- 数字图像的质量
- 数字图像的表示和存储
- 像素间的一些基本关系

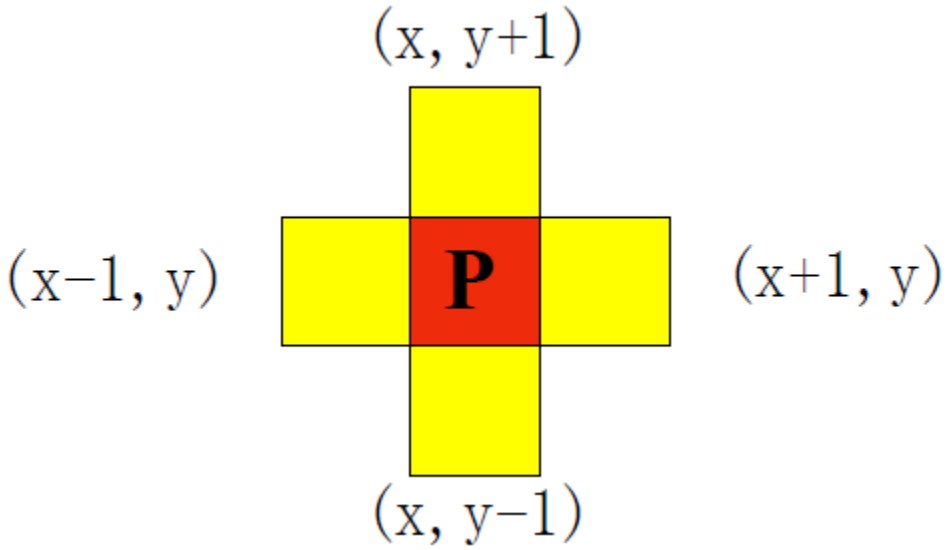
像素间的一些基本关系

- 像素之间的关系：很多图像处理任务（例如，滤波），通常需要考虑相邻像素的相互作用，而不是将每个像素孤立的来做运算。
- 相邻像素：
  - 4邻域
  - D邻域
  - 8邻域
- 连通性
  - 4联通
  - 8联通
- 距离测量

相邻像素：4邻域

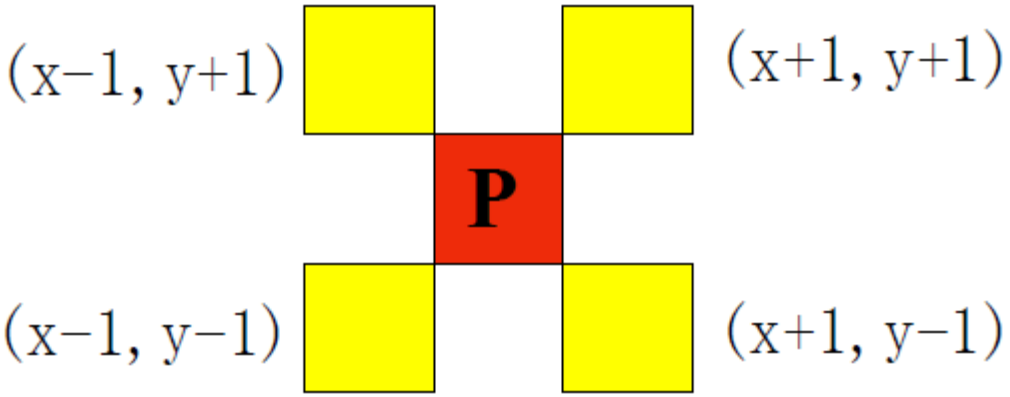
- 4邻域：像素p=(x, y)的4邻域是：

  (x+1, y)； (x-1, y)； (x, y-1)； (x, y+1)



相邻像素：D邻域

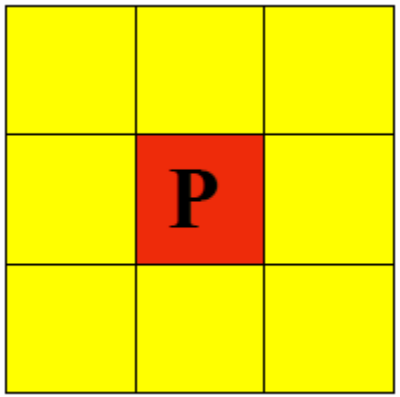
- D邻域：像素p=(x, y)的D邻域是其对角上的点：
  (x+1, y+1)； (x+1, y-1)； (x-1, y+1)； (x-1, y-1)



相邻像素：8邻域

- 8邻域：像素p=(x, y)的8邻域是：

  4邻域点 + D邻域点

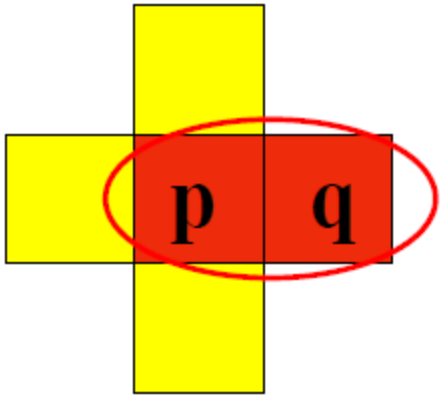


像素间的连通性

- 连通性是描述区域和边界的重要概念。
- 两个像素联通的必要条件：两个像素的位置是否相邻
- 种类：4连通和8连通

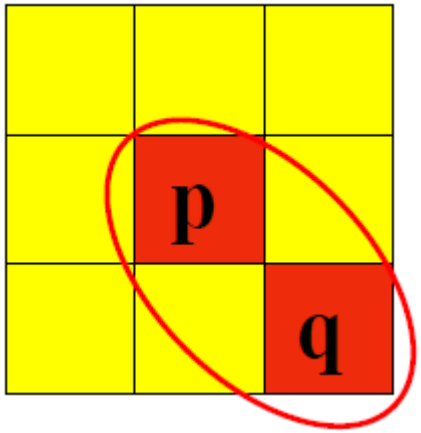
像素间的连通性：4连通

- 对于两个像素p和q，如果q在p的4邻域集合中，则称这两个像素是4连通的。



像素间的连通性：8连通

- 对于两个像素p和q，如果q在p的8邻域集合中，则称这两个像素是8连通的。



像素间的连通性：通路

- 通路：一条从像素p=(x_0, y_0) 到像素q=(x_n, y_n) 的路径。该路径是由具有坐标(x_0, y_0)，(x_1, y_1)，…，(x_n, y_n)的不同像素组成的序列。其中，(x_{i-1}, y_{i-1}) 和 (x_i, y_i)是邻接的，$1 \le i \le n$，n是路径的长度。
- 如果(x_0, y_0)=(x_n, y_n)，则该通路是闭合通路。

像素间的距离的定义

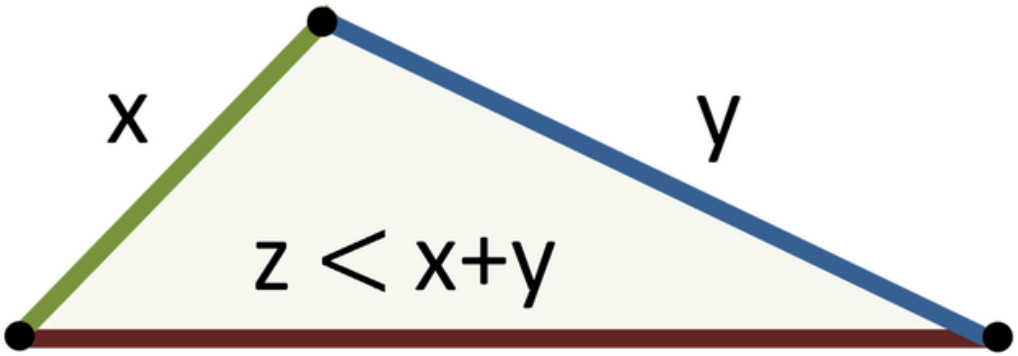
- 对于像素p，q和r，如果测量D满足以下三条特性：

  （1）同一性：D(p, q) ≥0 ，而且D(p, q)＝0，当且仅当p=q。

  （2）对称性：D(p, q) ＝ D(q, p)

  （3）三角不等性：D(p, r) ≤ D(p, q) + D(q, r)

  则称D是距离测量。



像素间的距离：欧式距离

- 像素p=(x, y)和q=(s, t)间的欧式距离 (Euclidean Distance) 定义如下：

$$D_E = \sqrt{(x-s)^2 + (y-t)^2}$$

- 优点：直观。
- 缺点：平方根的计算费时，且数值不为整数。

像素间的距离：城市距离（D_4距离）

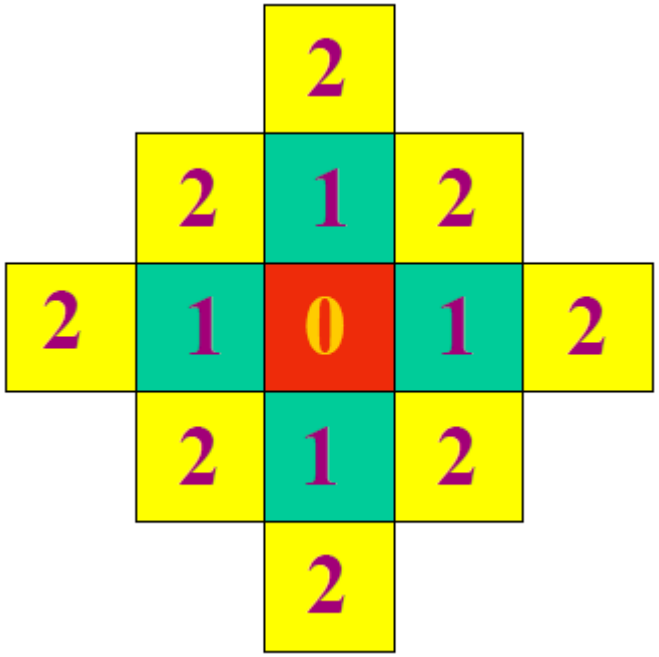
- 像素p=(x, y)和q=(s, t)间的城市距离定义如下：

$$D_4 = |x - s| + |y - t|$$

- 含义：在数字栅格中，如果只允许横向和纵向移动，城市距离表示从起点移动到终点所需的最少的步数。

像素间的距离：城市距离特性

- 到某像素的城市距离小于或者等于某个值的那些像素形成了一个菱形。



像素间的距离：棋盘距离（D_8距离）

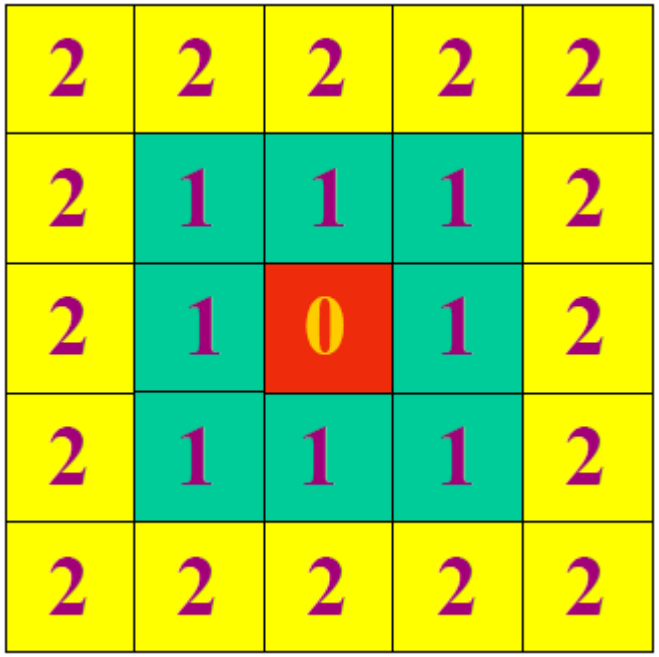
- 像素p=(x, y)和q=(s, t)间的棋盘距离定义如下：

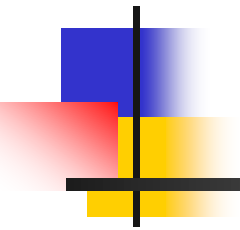
$$D_8 = \max\{|x-s|, |y-t|\}$$

- 含义：在数字栅格中，如果允许横向、纵向和对角线移动，棋盘距离表示从国王在棋盘上从一处移动到另一处所需的步数。

像素间的距离：棋盘距离特性

- 到某像素的棋盘距离小于或者等于某个值的那些像素形成了一个正方形。

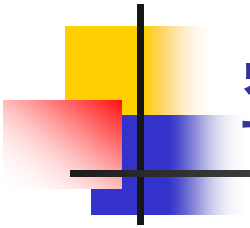




计算机视觉

空间域图像处理

2018-3-12



空间域图像处理

- 基本概念
- 基本运算
  - 点运算
  - 直方图运算
  - 代数运算
- 空间滤波器
  - 平滑空间滤波器
  - 锐化空间滤波器

基本概念

- 图像处理的目标：通过对于原始图像的处理，提取有用的信息，使得处理后的图像比原始图像更适合于特定应用。

- 因此，图像处理技术是面对任务的(task-dependent)。例如，一种适合于X射线图像的处理方法，不一定适合于火星图像的处理。

- 目前的图像处理方法主要分为两类：

  （1）空间域滤波：对图像的像素直接处理

  （2）频率域滤波：在图像的频率谱中处理

空间域图像处理的基本数学表达

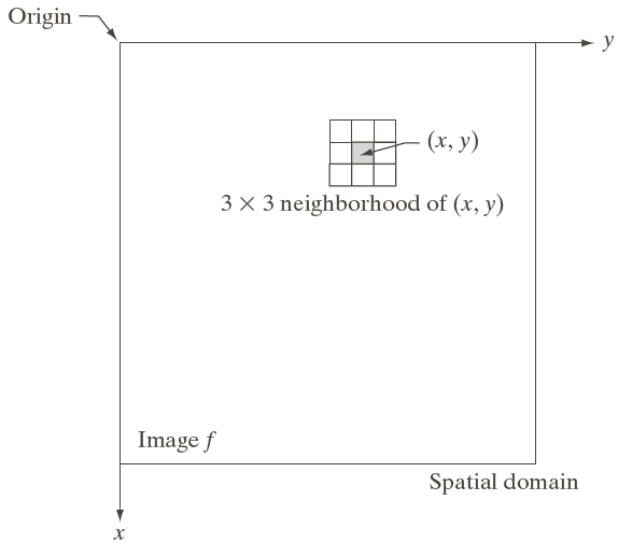
- 空间域图像处理的基本数学表达：

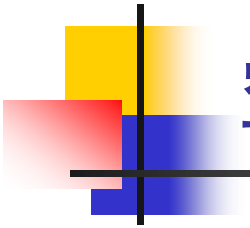
$$g(x,y)=T\left[f(x,y)\right]$$

- f(x,y)：输入图像
- g(x,y)：输出图像
- T：图像处理算子，它定义在(x, y)的邻域或者只定义在（x,y）这个像素。

空间域图像处理的基本操作

- 点(x, y)的邻域：大部分情况下是利用中心在(x, y)点的正方形或者矩形子图像(subimage)，表达为m×n邻域。该邻域被称之为掩膜(masks)、滤波器(filters)、窗口(windows)或者核(kernels)。
- 例如： 1×1邻域，3×3邻域（8邻域）。
- 基本操作：邻域的中心从一个像素向另一个像素移动，T操作应用到每一个像素得到该点的输出g。





空间域图像处理

- 基本概念
- 基本运算
- 点运算
- 直方图运算
- 代数运算
- 空间滤波器
- 平滑空间滤波器
- 锐化空间滤波器

点运算：反转变换

- 点运算：以每个像素为基础，对一幅图像进行操作。
- 图像反转(Image Negatives)

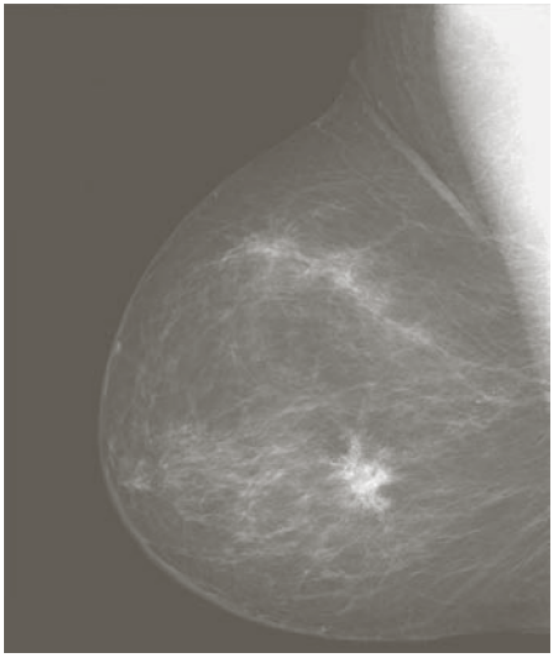
  给定图像的灰度级范围[0,L-1]，图像反转算子为：

$$T(r):\ s = L - 1 - r$$

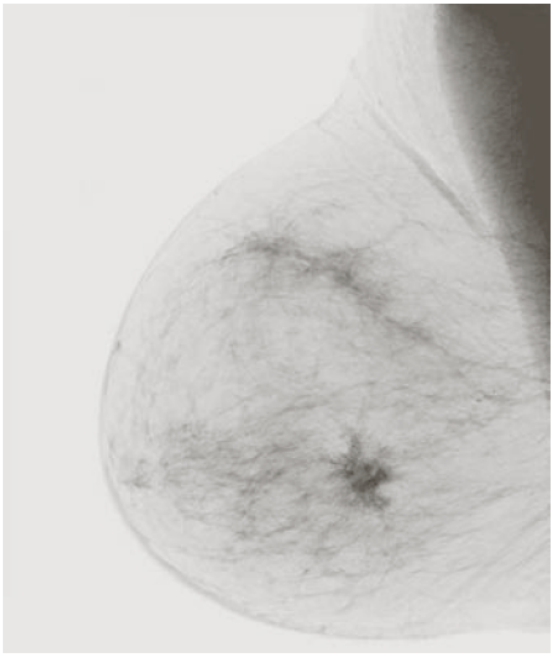
  其中 r：原始图像

  s：转换后的图像
- 作用：黑白颠倒，黑的变白，白的变黑。
- 适用范围：用于增强嵌入于图像暗色区域的白色或者灰色细节，特别是当黑色面积占主导地位时。

反转变换：示例



原始图像



反转后的图像

点运算：对数变换

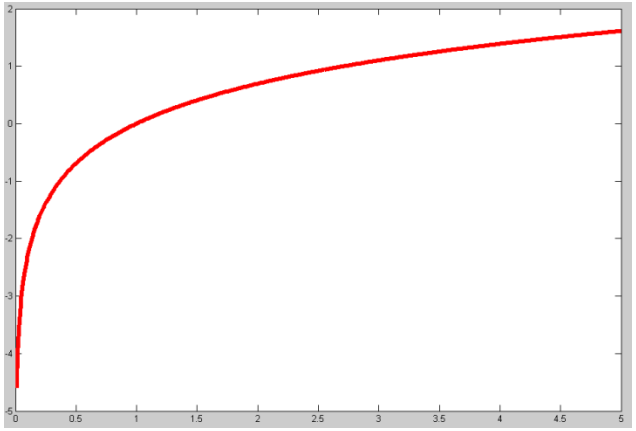
- 对数变换(Log Transformations):

$$s = c\log(1+r)$$

其中 r：原始图像，且r≥0

s：转换后的图像

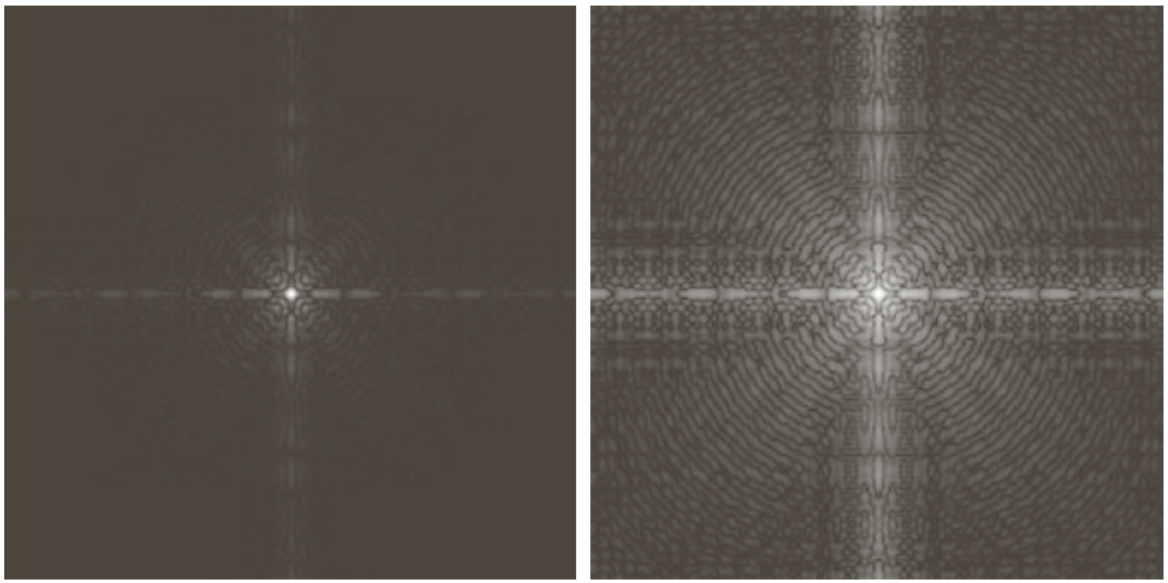
c：常数



- 作用：把位于较窄范围内的低灰度输入图像映射到较宽的范围。

对数变换：示例

- 适用范围：当原图的动态范围太大，超出了某些显示设备的允许的动态范围，如果直接使用原图，则一部分低灰度值部分的细节可能会丢失。此时，可以使用对数变换，压缩高灰度值部分、扩展低灰度值部分。



原图：傅立叶频谱(0～1.5×10^6)　　对数变换后的图像

点运算：幂次变换

- 幂次变换(Power-Law Transformations):

$$s = cr^{\gamma}$$

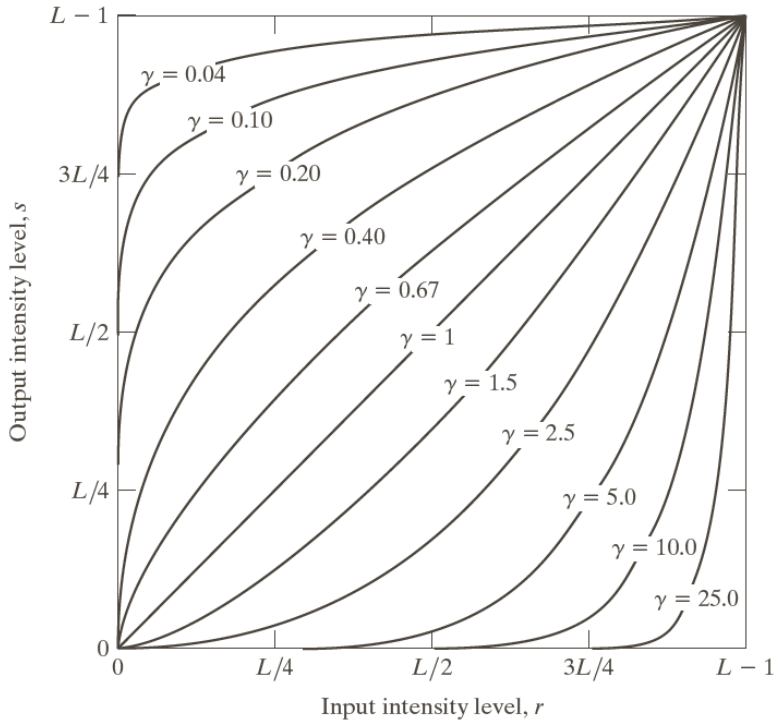
其中 r：原始图像

s：转换后的图像

c和γ：正常数

幂次变换：作用

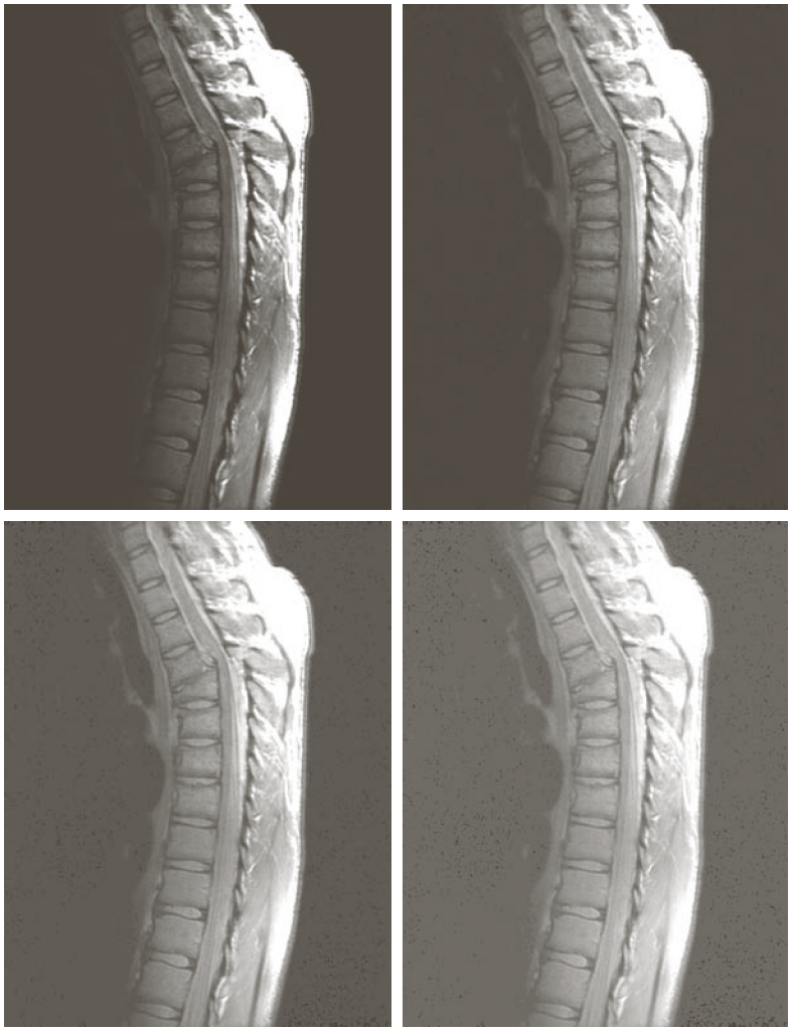
- 作用：
- γ >1：降低灰度级，使图像变暗。
- γ <1：提高灰度级，使图像变亮。



幂次变换：示例1

- 例子1：人体胸上部脊椎骨折的核磁共振图像
- 目的：使得图像变亮

- 左上方：原图
- 右上方：γ＝0.6的转换图像
- 左下方：γ＝0.4的转换图像
- 右下方：γ＝0.3的转换图像



幂次变换：示例2

- 例子2：航空地面图像
- 目的：使得图像变暗

- 左上方：原图
- 右上方：γ＝3.0的转换图像
- 左下方：γ＝4.0的转换图像
- 右下方：γ＝5.0的转换图像