哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

图像处理

Image Processing

任课教师： 杨云云

图像和数字图像

- 什么是图像？
  - ✓ 图像(image)是泛指照片、动画等等形成视觉景像的事物。图像与计算机图形学中的图形的区别是：计算机图形学是从建立数学模型到生成图形，而图像通常是指从外界产生的图形。
  - ✓ 客观世界是三维空间，但一般图像是二维的。二维图像在反映三维世界的过程中必然丢失了部分信息。即使是记录下来的信息也可能有失真，甚至于难以识别物体。因此，需要从图像中恢复和重建信息，分析和提取图像的数学模型，以至于形成人们对于图像记录下的事物有正确和深刻的认识。这个过程就称为图像处理过程。

- 定义为二维函数f(x,y)，其中，x,y是空间坐标，f(x,y)是点(x,y)的幅值。
- 灰度图像是一个二维灰度（或亮度）函数f(x,y)。
- 彩色图像由三个（如RGB，HSV）二维灰度（或亮度）函数f(x,y)组成。

图像和数字图像

- 图像的分类

  - ✓ 按灰度分类：二值、多灰度
  - ✓ 按色彩分类：单色、彩色
  - ✓ 按运动分类：静态、动态
  - ✓ 按时空分布分类：二维、三维

- 为什么需要数字图像？

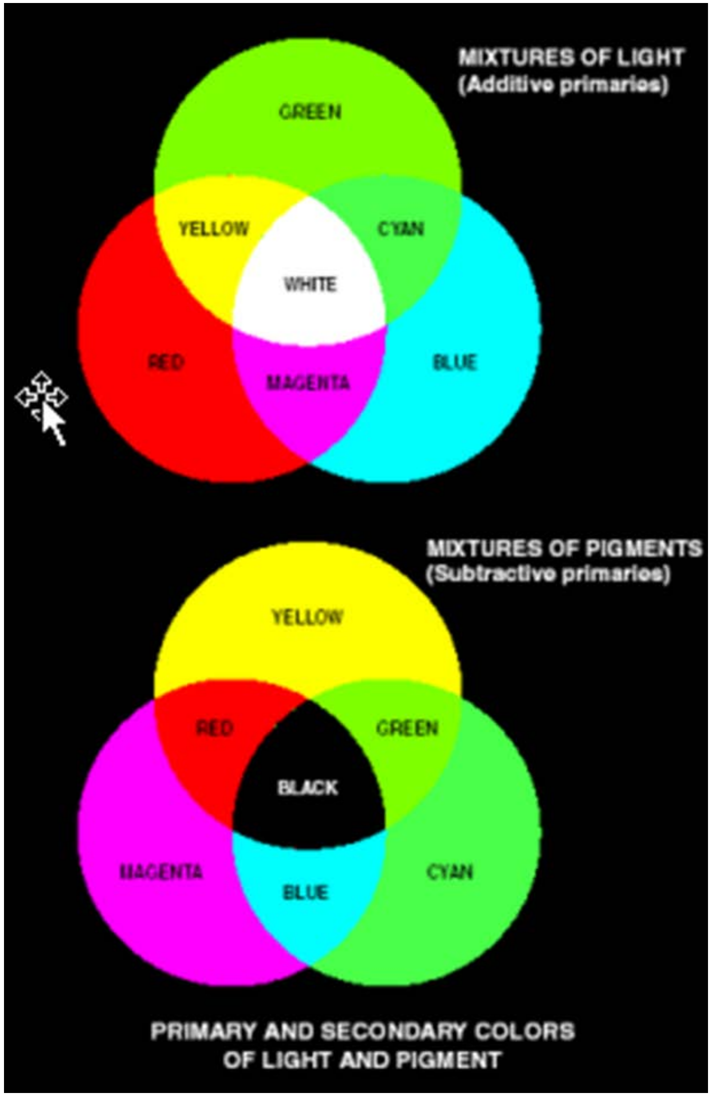
普通图像包含的信息量巨大，需要使用计算机对图像进行处理。因此，需要把普通图像转变成计算机能处理的数字图像。现在的数码相机可以直接地把视觉图像变成数字图像。数字图像类似于光栅图形，由有限行和有限列组成。每个基本单元叫做一个像素(pixel)。三维图像的像素又叫做体素(voxel)。通常的二维数字图像是一个矩型，可以用一个二维数组$I(x, y)$来表示，其中x, y是二维空间中的某坐标系的坐标，$I(x, y)$表示图像在该点处的灰度值等性质。彩色可以是红绿蓝三个单色的一定灰度值的合成。一般来说，这些坐标和灰度值是实数，不仅依赖于坐标系的选取，而且依赖于灰度值的度量单位。但是，数字计算机只可能表示有限字长的有限个数字。所以必须把灰度值离散化。简单地说，数字图像等同于一个整数值的有限矩阵。数字图像是数字图像处理和分析的对像。

图像和数字图像

左边的图像是图像处理技术中常用来检验计算机算法的实际效果的标准图像。这幅图像的名称是lenna。它是由一组数字组成的。原图像的宽和高都是256个像素，每像素有八位。它在BMP格式下有约66K字节的大小。

图像和数字图像

所有的颜色可以看作是三种基本颜色的迭加，也可以看作三种补色（从白色中除去某种颜色）的迭加。



图像技术和图像工程

- 哪些属于图像技术

图像技术是与图像有关部门的技术的总称。它是一类综合技术工程。它包括图像的采集、获取、编码、存储和传输、图像的生成、显示和输出、图像的变换、增强、恢复和重建、图像的分割、目标的检测、表达和描述、特征的提取、图像的分类、识别、图像模型的建立和匹配、图像和场景的理解。

狭义的数字图像处理是指图像的增强、恢复和重建，操作的对像是图像的像素，输出的是图像。

- 什么是图像工程？

图像工程是一门系统地研究各种图像理论、技术和应用的交叉学科。

从它的研究方法看，它与数学、物理学、生物学、心理学、电子学、计算机科学可以互相借鉴，从它的研究范围看，它与模式识别、计算机视觉、计算机图形学等学科交叉。

数字图像处理的主要研究内容

1. 图像压缩编码

- ✓ 图像压缩编码技术可以减少描述图像的数据量，以便节约图像存储的空间，减少图像的传输和处理时间。
- ✓ 图像压缩有无损压缩和有损压缩两种方式，编码是压缩技术中最重要的方法，在图像处理技术中是发展最早和应用最成熟的技术。
- ✓ 主要方法：熵编码，预测编码，变换编码，二值图像编码、分形编码……

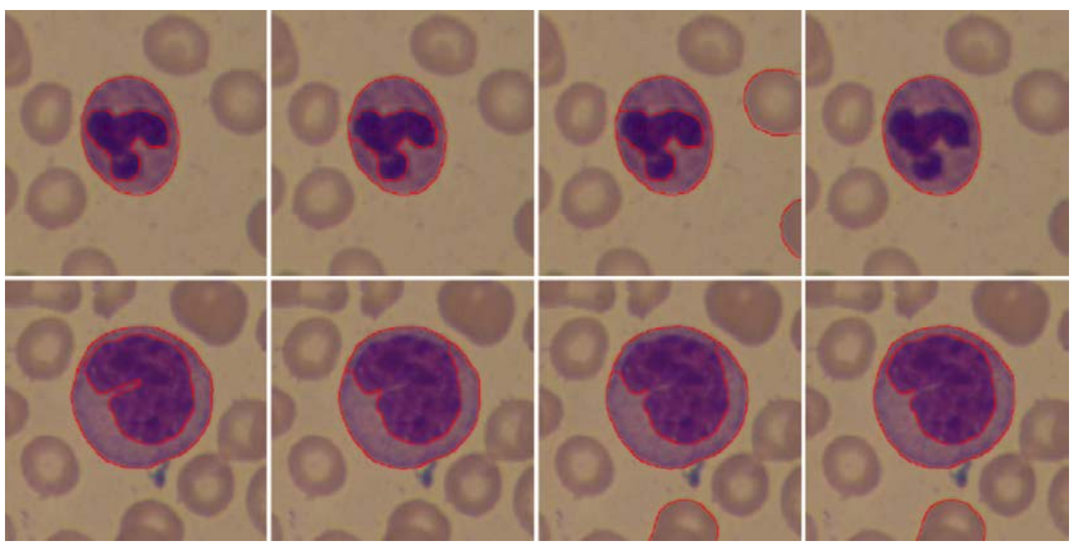
2. 图像的增强和复原

- ✓ 图像增强和复原的目的是为了改善图像的视觉效果，如去除图像噪声，提高图像的清晰度等。图像增强不考虑图像降质的原因，突出图像中感兴趣的部分。图像复原要求对图像降质的原因有所了解，根据图像降质过程建立“退化模型”，然后采用滤波的方法重建或恢复原来的图像。
- ✓ 主要方法：灰度修正，平滑，几何校正，图像锐化，滤波增强，维纳滤波……


Toronto

3. 图像分割

- ✓ 图像分割是数字图像处理中的关键技术之一。图像分割将图像中有意义的特征提取出来（物体的边缘、区域），它是进行进一步图像识别、分析和图像理解的基础。
- ✓ 虽然目前已研究出了不少边缘提取、区域分割的方法，但还没有一种普遍适用于各种图像的有效方法。对图像分割的研究还在不断的深入中，是目前图像处理研究的热点方向之一。
- ✓ 主要方法：活动轮廓模型、图像边缘检测、灰度阈值分割、基于纹理分割、区域增长……



4. 图像描述

- ✓ 图像描述是图像分析和理解的必要前提。图像描述是用一组数量或符号（描述子）来表征图像中被描述物体的某些特征。
- ✓ 主要方法：二值图像的几何特征、简单描述子、形状数、傅立叶描述子、纹理描述……

5. 图像识别

- ✓ 图像识别是人工智能的一个重要领域，是图像处理的最高境界。一副完整的图像经预处理、分割和描述提取有效特征之后，进而由计算机系统对图像加以判决分类。

6. 图像隐藏

- ✓ 是指媒体信息的相互隐藏

  数字水印
  图像的信息伪装

数字图像处理的应用领域

- 传统领域

  ✓ 医学、空间应用、地理学、生物学、军事......

- 最新领域

  ✓ 数码相机（DC）、数码摄像机（DV）
  ✓ 指纹识别、人脸识别
  ✓ 互联网、视频、多媒体等
  ✓ 基于内容的图像检索、视频检索、多媒体检索
  ✓ 水印、游戏、电影特技、虚拟现实、电子商务等

数字图像的表示

- 二维离散灰度函数——f(x, y)
  - ✓ x，y代表图像像素的空间坐标。
  - ✓ 函数值f代表了在点(x, y)处像素的灰度值。

- 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列的像素的灰度值；i，j表示几何位置。

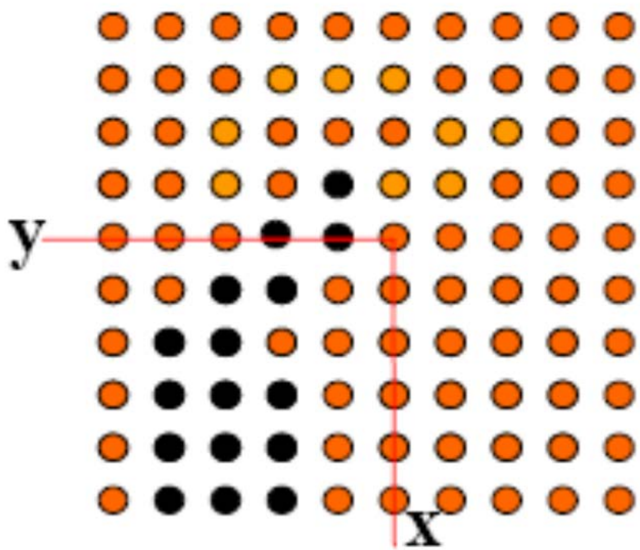
- 对于单色（灰度）图像而言，每个像素的亮度用一个数值来表示，通常数值范围在0到255之间，0表示黑、255表示白，其他值表示处于黑白之间的灰度。

- 彩色图像可以用红、绿、蓝三元组的二维矩阵来表示。通常，三元组的的每个数值也是在0到255之间，0表示相应的基色在该像素中没有，而255则代表相应的基色在该像素中取得最大值。

数字图像的像素表示

什么是像素？

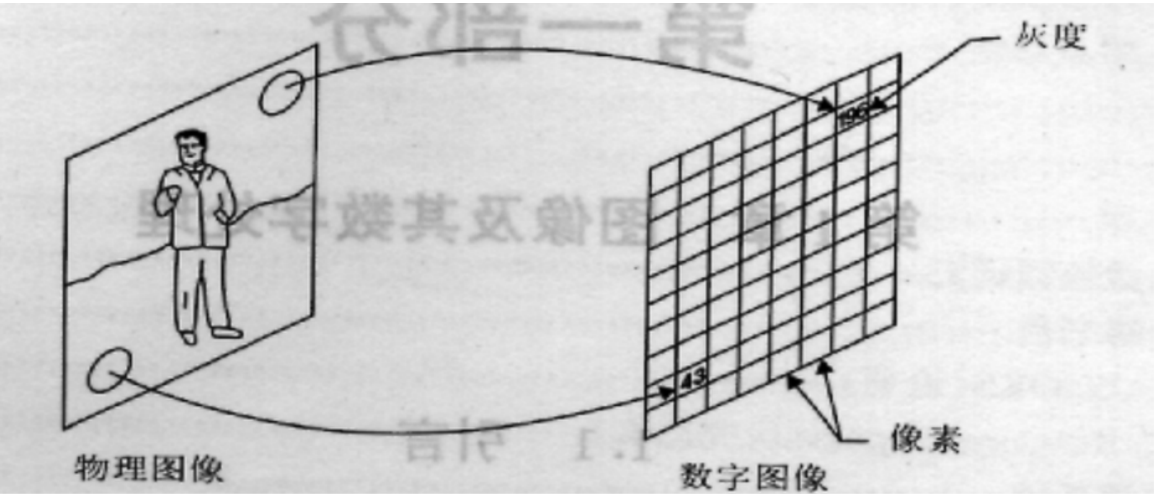
数字图像由二维的元素组成，每一个元素具有一个特定的位置(x, y)和幅值$f(x,y)$，这些元素称为像素。



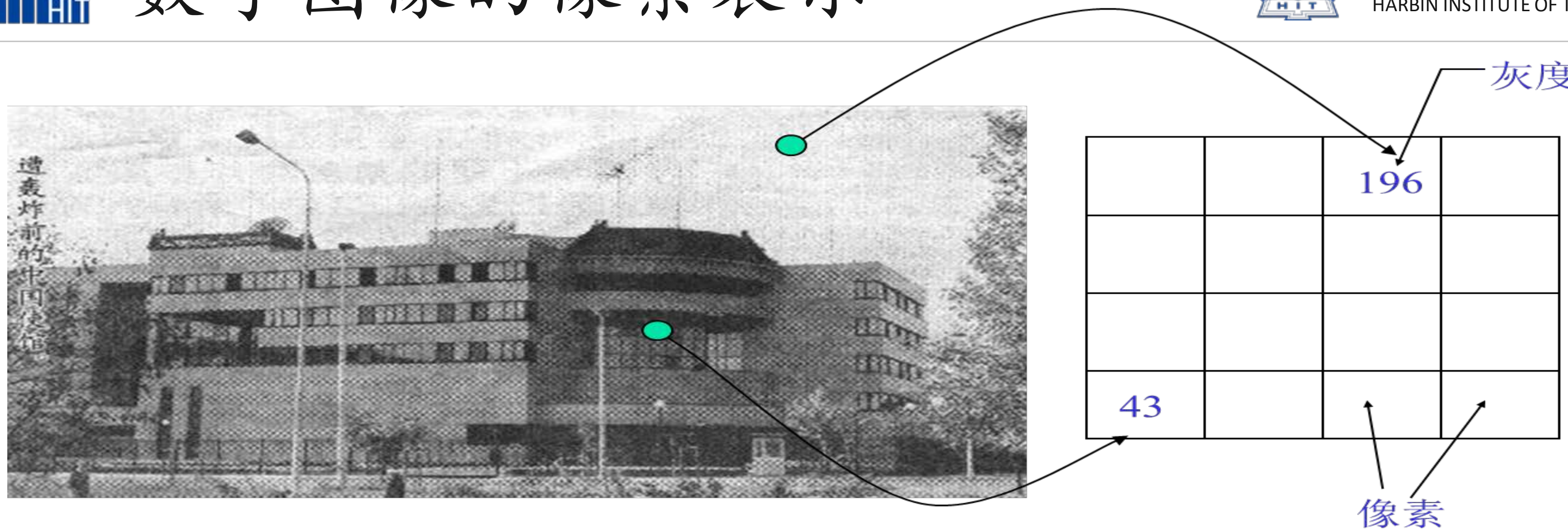
数字图像的像素表示

- ✓ 数字图像是指由被称作像素的小块区域组成的二维矩阵。将物理图像行列划分后，每个小块区域称为像素（pixel）。
- ✓ 对于单色即灰度图像而言，每个像素的的亮度用一个数值来表示，通常数值范围在0到255之间，即可用一个字节来表示。0表示黑，255表示白，而其他表示灰度级别。

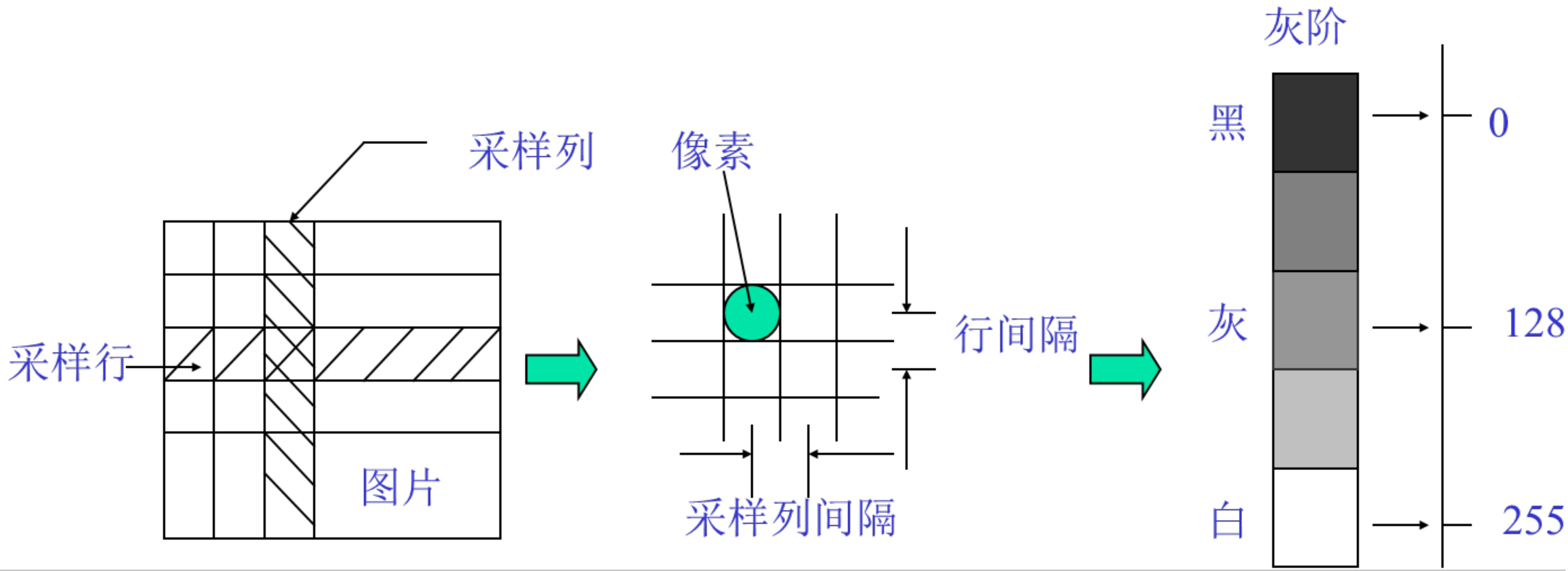
物理图像及对应的数字图像



数字图像的像素表示

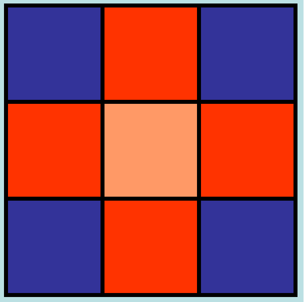


物理图像　　　　数字图像



像素间联系

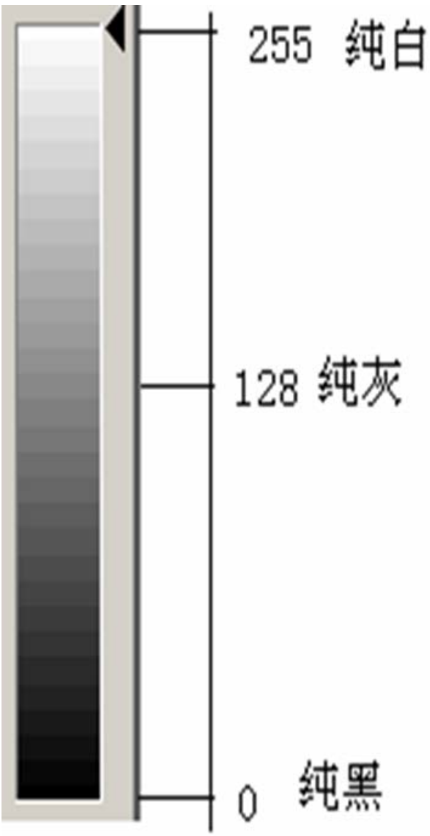
- 一个像素的邻域是指该像素周围的像素集合。
- 一个像素p的周围有八个像素，它们共同组成了该像素的8-领域N8(p)。一个像素与上下左右的四个像素组成了4-邻域N4(p)。一个像素与四个角上的像素组成了对角邻域ND(p)。



- 同类灰度的像素间的邻接，连接和连通问题：
  - ✓ 两个像素彼此落在对方的4-邻域内，称为4-连接。
  - ✓ 两个像素彼此落在对方的8-邻域内，称为8-连接。
  - ✓ 如果两个像素或者是4-连接，或者不是4-连接但落在对方的对角邻域ND(p)内，那么称为混合连接（m-连接）。

灰度图像的矩阵表示

➢ 灰度级

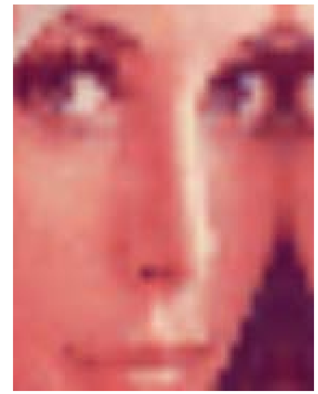


（仅列出一部分(26×31)）

97	23	97	23	23	23	84	23	84	23	84	23	84	23	84	23	23	23	97	97	3	40	40	158	158	40	3	3	97	97	97
23	97	23	23	23	23	84	23	84	23	84	23	84	23	84	23	23	23	97	97	3	3	40	158	158	40	40	3	97	97	23
97	97	97	97	23	23	84	23	23	23	23	84	23	84	23	84	23	23	23	97	3	40	40	40	158	158	3	3	3	97	97
97	97	97	97	23	23	23	23	23	84	23	23	23	23	84	23	23	23	97	97	97	3	40	158	158	40	40	3	3	3	97
97	97	97	97	23	84	23	84	23	23	23	84	23	84	23	23	23	23	23	97	3	3	40	40	158	40	40	3	3	3	97
97	97	97	23	23	23	23	23	23	23	23	23	84	23	84	23	84	23	23	97	97	3	40	40	158	40	40	40	3	3	3
97	97	23	97	23	23	23	23	23	23	84	23	23	84	23	84	23	23	23	97	97	3	40	40	158	158	40	3	3	3	3
97	97	97	23	23	23	23	23	23	23	23	23	84	23	84	23	23	23	23	97	97	3	40	40	158	40	40	40	3	3	97
97	97	97	97	23	97	23	23	23	23	23	23	23	23	84	23	84	23	23	97	97	3	3	40	40	158	40	40	3	3	3
97	97	97	23	97	23	97	23	23	23	23	23	84	23	23	84	23	23	23	23	97	97	40	40	158	40	40	40	40	3	97
97	97	97	23	97	23	97	23	97	23	23	23	23	84	23	84	23	84	23	23	97	3	3	40	158	158	158	40	3	3	97
97	97	23	97	23	97	23	97	23	23	23	23	23	23	23	23	84	23	23	23	97	97	40	40	158	158	40	40	40	3	97
97	97	23	97	23	97	23	23	97	23	23	23	23	23	84	23	84	23	23	23	97	97	3	40	158	158	158	40	3	3	97
97	97	23	97	23	23	23	97	23	23	23	23	84	23	84	23	84	23	23	23	97	97	40	40	158	40	158	40	40	3	97
97	23	97	23	23	23	23	23	23	23	23	23	23	23	84	23	84	23	23	23	97	97	3	40	40	158	158	40	40	3	97
97	97	23	97	23	23	23	23	23	23	23	84	84	84	23	84	23	84	23	23	97	97	3	40	158	40	158	40	40	3	97
23	97	23	23	23	23	23	23	23	23	23	23	84	23	84	23	84	23	23	97	23	97	97	40	40	40	158	158	3	3	97
97	23	97	23	23	23	84	23	23	84	84	84	23	84	84	84	23	23	23	23	97	97	3	3	40	158	40	158	40	97	97
97	23	23	23	23	23	23	84	23	23	23	23	84	84	84	23	84	23	23	23	23	97	97	3	40	40	40	158	40	3	23
97	23	23	23	23	84	23	23	23	84	84	84	84	23	84	84	84	23	84	23	23	97	3	3	40	40	158	158	3	3	97
97	23	23	23	23	23	23	84	23	23	23	84	23	84	84	84	23	84	23	23	23	97	97	3	40	40	158	158	40	3	97
23	23	23	23	84	23	84	23	23	84	23	84	23	84	23	84	84	23	23	23	23	97	97	3	3	40	158	158	40	3	97
97	23	23	23	23	84	84	23	84	23	84	23	84	23	84	84	23	84	84	23	23	97	97	97	3	40	158	158	40	40	3
23	23	23	84	23	84	23	84	23	84	23	84	23	84	23	84	23	84	23	23	23	97	97	3	3	40	40	165	158	40	3
23	23	84	23	84	84	84	23	84	23	23	23	84	23	84	23	84	23	23	23	23	23	97	97	3	40	158	158	158	40	3
23	23	23	84	84	84	84	84	23	23	84	23	23	23	84	23	84	23	84	23	23	23	97	97	3	40	158	158	158	40	3

彩色图像的矩阵表示

（仅列出一部分(26×31)）



(207,137,130)	(220,179,163)	(215,169,161)	(210,179,172)	(210,179,172)
(207,154,146)	(217,124,121)	(226,144,133)	(226,144,133)	(224,137,124)
(227,151,136)	(227,151,136)	(226,159,142)	(227,151,136)	(230,170,154)
(231,178,163)	(231,178,163)	(231,178,163)	(236,187,171)	(236,187,171)
(239,195,176)	(239,195,176)	(240,205,187)	(239,195,176)	(231,138,123)
(217,124,121)	(215,169,161)	(216,179,170)	(216,179,170)	(207,137,120)
(159, 51, 71)	(189, 89,101)	(216,111,110)	(217,124,121)	(227,151,136)
(227,151,136)	(226,159,142)	(226,159,142)	(237,159,135)	(237,159,135)
(231,178,163)	(236,187,171)	(231,178,163)	(236,187,171)	(236,187,171)
(236,187,171)	(239,195,176)	(239,195,176)	(236,187,171)	(227,133,118)
(213,142,135)	(216,179,170)	(221,184,170)	(190, 89, 89)	(204,109,113)
(204,115,118)	(189, 85, 97)	(159, 60, 78)	(136, 38, 65)	(160, 56, 75)
(204109113)	(227151136)	(226159142)	(237159135)	(227151136)

图像的采样和量化

一幅图像需要经过离散化成为数字图像后才能被计算机处理。图像的空间坐标的离散化叫做空间采样，灰度的离散化叫做灰度量化。

采样分为均匀采样和量化和非均匀采样和量化。

假设图像是一个长方形。在平面上取M×N个大小相同的网格，并把灰度分成G个等级。取各网格中的某点处的灰度值最接近的整数作为该网格的灰度。

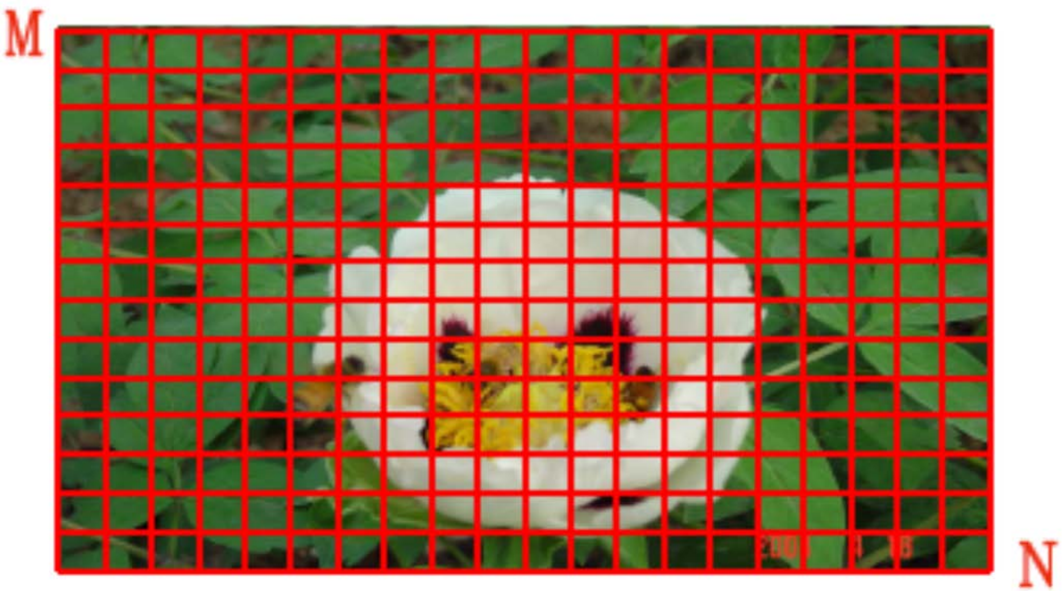
通常，取$M=2^m$, $N=2^n$和$G=2^k$。则存储一幅图像的需要的位数等于b=MNG。例如，一幅128×128、64个灰度等级的图像需要2^{20}位，512×512、256个灰度等级的图像需要2^{26}位。采样的个数和灰度等级的选取与分辨率和储存的能力两者有关，需要综合考虑。例如：

- ✓ 图像空间分辨率变化产生的效果。
- ✓ 图像灰度分辨率变化产生的效果。
- ✓ 图像空间和灰度分辨率同时变化产生的效果。

图像采样

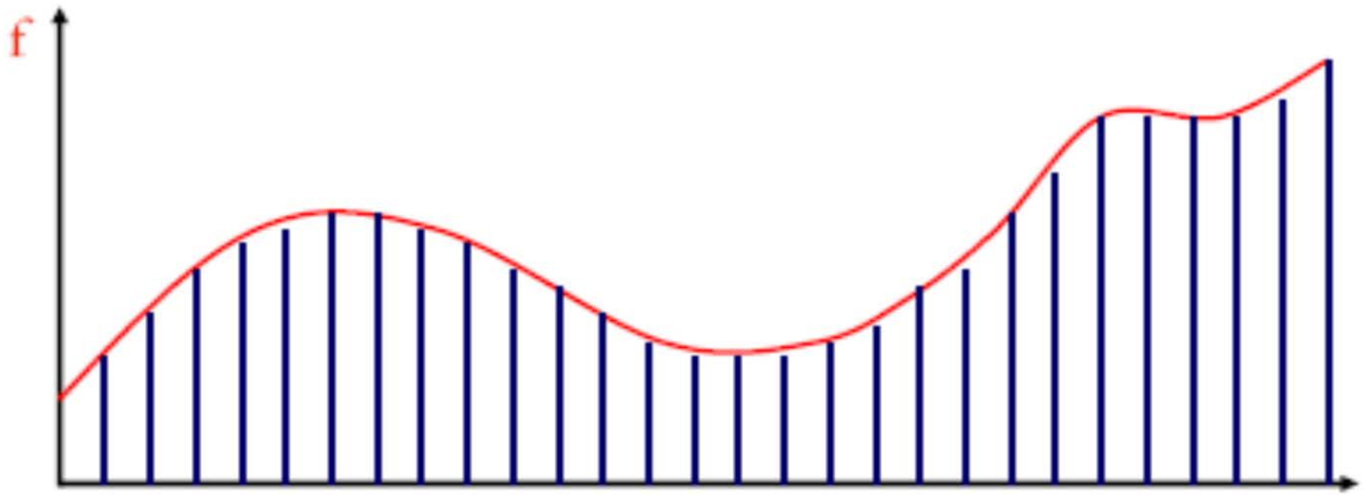
- 空间坐标（x,y）的数字化被称为图像采样。

- 确定水平和垂直方向上的像素个数N、M。

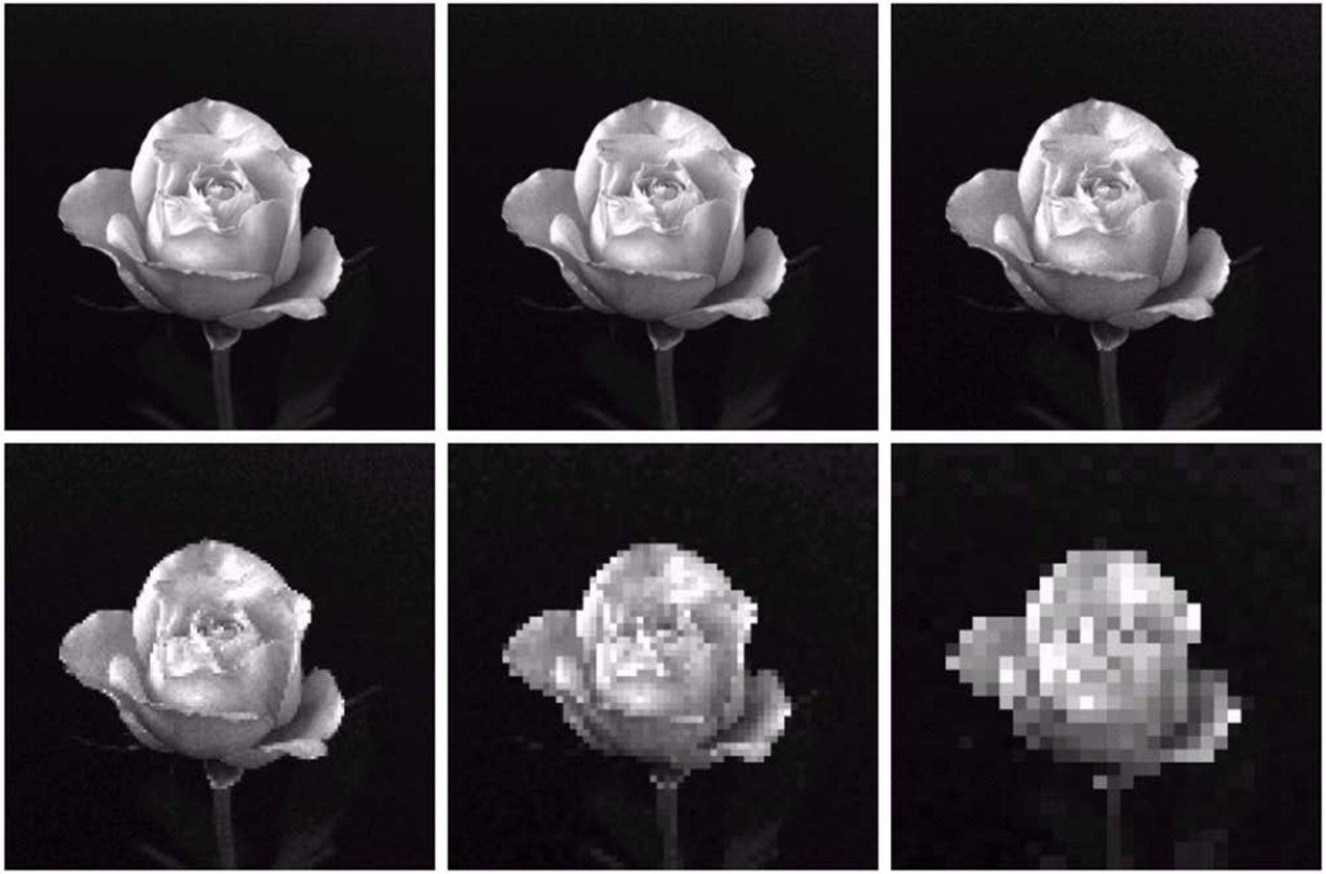


图像量化

- 函数取值的数字化被称为图像的量化，如量化到256个灰度级。



图像空间分辨率变化产生的效果



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.

不同空间分辨率图像的视觉差异

图像空间分辨率变化产生的效果

265x180

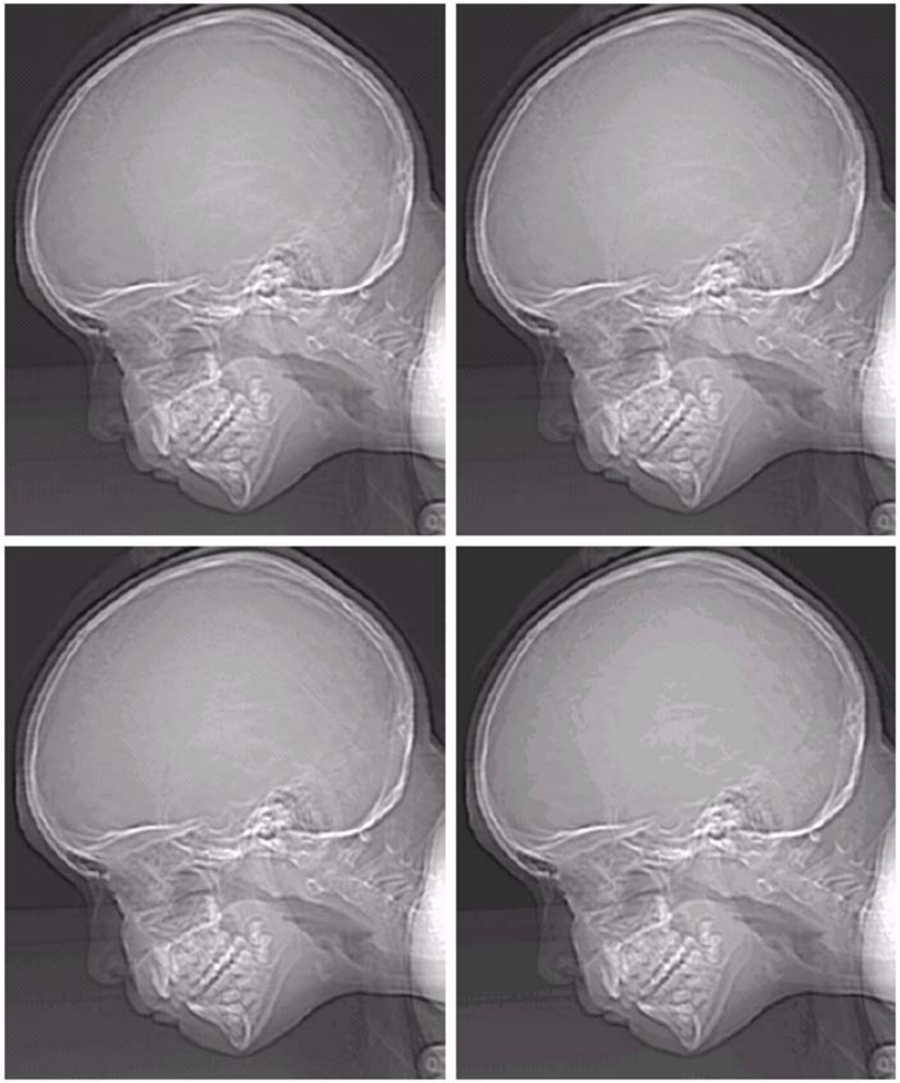
不同空间分辨率图像的视觉差异

133x90

66x45

33x22

图像灰度分辨率变化产生的效果



a b
c d

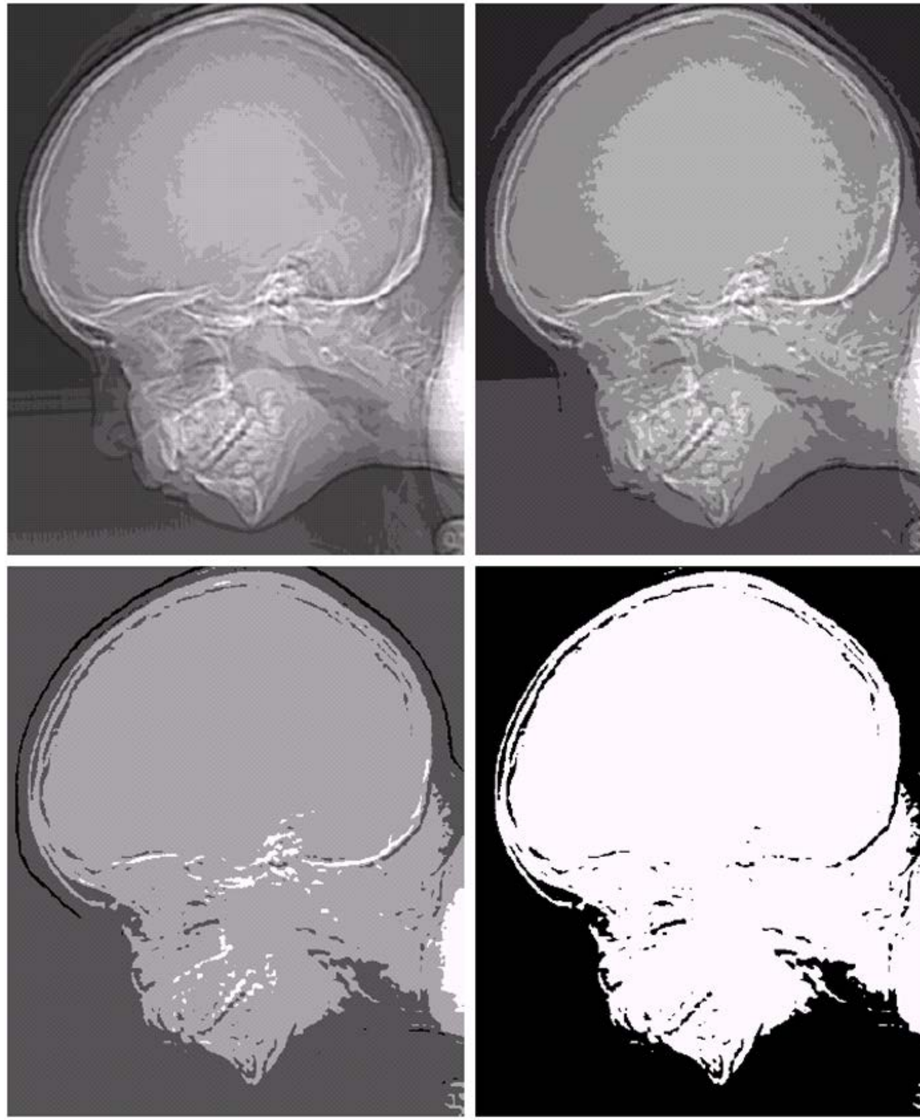
FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

不同灰度分辨率图像的视觉差异

图像灰度分辨率变化产生的效果

e f
g h

FIGURE 2.21
(Continued)
(e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



2

不同灰度分辨率图像的视觉差异

基于matlab图像处理基础

- 读入并显示一副图像

```
I=imread('cameraman.tif');
imshow(I)
```

- 改变图像的大小

```
I2=imresize(I,0.5);
figure,imshow(I2)
```

- 使用阈值操作将图像转化为二值图像

```
Threshold=graythresh(I2);
BW=im2bw(I2, Threshold);
figure,imshow(BW)
```

- 将二值图像矩阵保存成图像

```
imwrite(BW,'image.jpg');
```

- 将生成的fig图保存成图像

```
saveas(gcf,'fig.jpg')
```

基于matlab图像处理基础

- 灰度图像
  - ✓ 它是包含灰度级（亮度）的图像，在MATLAB中，灰度图像是由一个uint8，uint16或者一个双精度类型的数组描述。
  - ✓ 数据矩阵元素值分别代表了像素的颜色的灰度值。

- RGB图像
  - ✓ 每个像素由三个数值来指定红、绿和蓝颜色分量。
  - ✓ 一幅RGB图像由一个uint8、uint16或双精度类型的m×n×3数组，m和n分别表示图像的宽度和高度。
  - ✓ RGB图像是24位图像，红、绿和蓝分别占用8位，因而图像理论上可以包含2^{24}种不同的颜色，能再现图像的真实色彩。

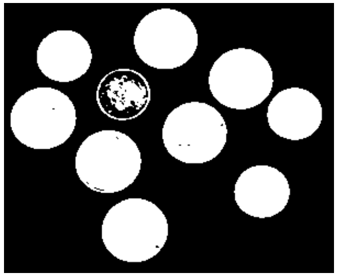
- 图像转化为二值图像

  ✓ BW=im2bw(I,level)将灰度图像转化为二值图像，其中level为归一化的阈值，取值在[0,1]之间。可由graythresh(I)得到；

  ✓ BW=im2bw(RGB,level);将RGB图像转化为二值图像。

```
clear all;
I = imread('coins.png');
subplot(1,2,1);imshow(I);
xlabel('(a)灰度图像');
level = graythresh(I);
BW = im2bw(I,level);
subplot(1,2,2);imshow(BW);
xlabel('(b)二值图像');
```

(a) 灰度图像

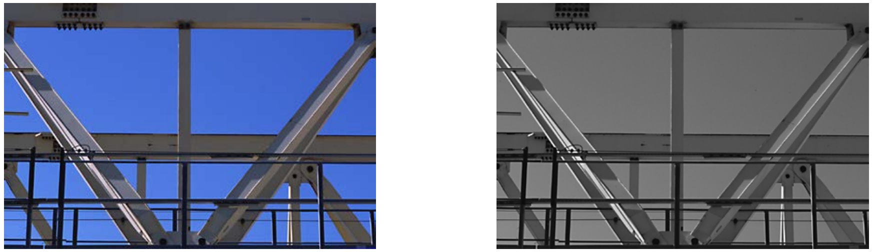


(b) 二值图像

✓ 彩色图像转化为灰度图像

I=rgb2gray(RGB);将彩色图像RGB转化为灰度图像;

```
RGB = imread('gantrycrane.png');
J = rgb2gray(RGB);
subplot(1,2,1);imshow(RGB);
xlabel('(a)彩色图像')
subplot(1,2,2), imshow(J);
xlabel('(b)灰度图像');
```

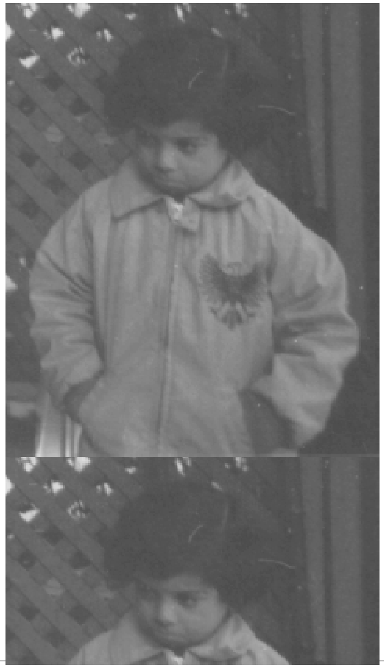


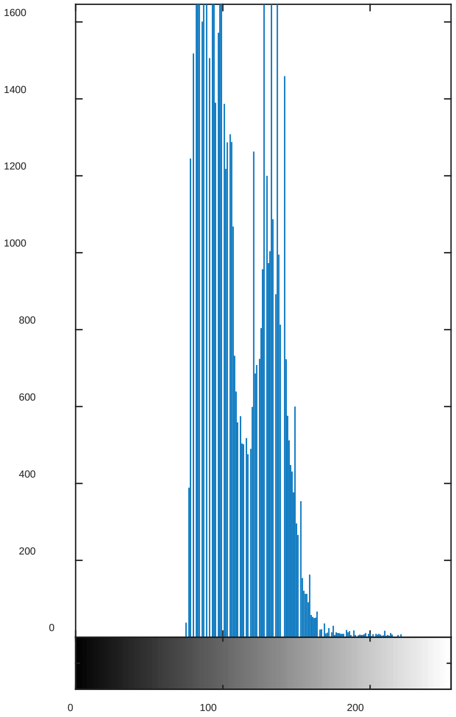
(a) 彩色图像　　(b) 灰度图像

图像的直方图

- ✓ 统计一副图像中相同像素值的个数。
  imhist(I)可以显示一幅图像的直方图

```
I= imread('pout.tif');%读取图像
subplot(121),imshow(I);%显示原图像
subplot(122),imhist(I)%显示其直方图
```

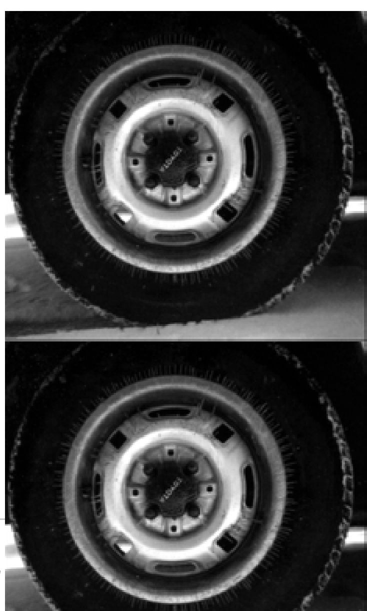


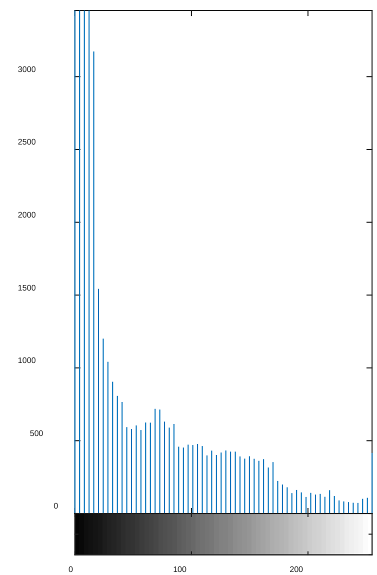


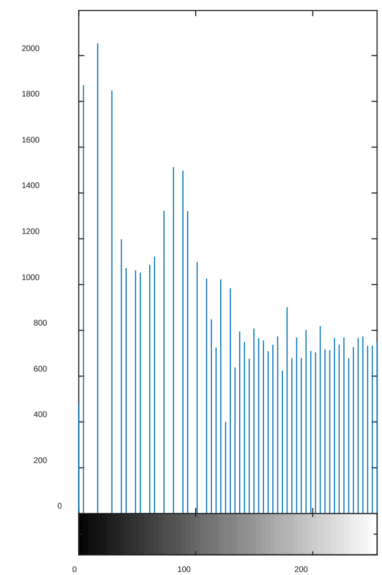
图像的直方图

- ✓ 直方图均衡化：将一幅图像的像素值均匀分布在图像的各个区间上，从而使图像的视觉效果得到改善。

```
I = imread('tire.tif');%读取图像
J = histeq(I);%直方图均衡化
figure;%显示原图像和均衡化后的图像
subplot(121), imshow(I);    subplot(122),imshow(J)
figure;%显示原图像和均衡化后图像的直方图
subplot(121), imhist(I,64);
subplot(122), imhist(J,64);
```







图像边缘检测

- sobel方法
  - ✓ sobel算子结合了高斯平滑和微分求导，对噪声具有平滑作用。
  - ✓ d_x表示水平方向,d_y表示垂直方向。

$$d_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad d_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

  - ✓ BW=edge(I,'sobel',thresh,direction, options)
  - ✓ thresth指阈值，低于该阈值的像素值被忽略，direction指检测的方法，可取horizontal，vertical或者both，默认是thinning，即边缘细化。

- prewitt方法
  - ✓ prewitt算子是利用特定区域内像素灰度值产生的差分实现边缘检测。
  - ✓ d_x表示水平方向,d_y表示垂直方向。

$$d_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad d_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

  - ✓ BW=edge(I,'prewitt',thresh,direction)

- roberts方法

  ✓ roberts方法通过局部差分计算检测边缘线条。

  ✓ d_x表示水平方向,d_y 表示垂直方向。

$$d_x = \begin{bmatrix} -1 & 0 \\ -0 & 1 \end{bmatrix} \qquad d_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

  ✓ BW=edge(I,'roberts',thresh,direction)

- canny方法

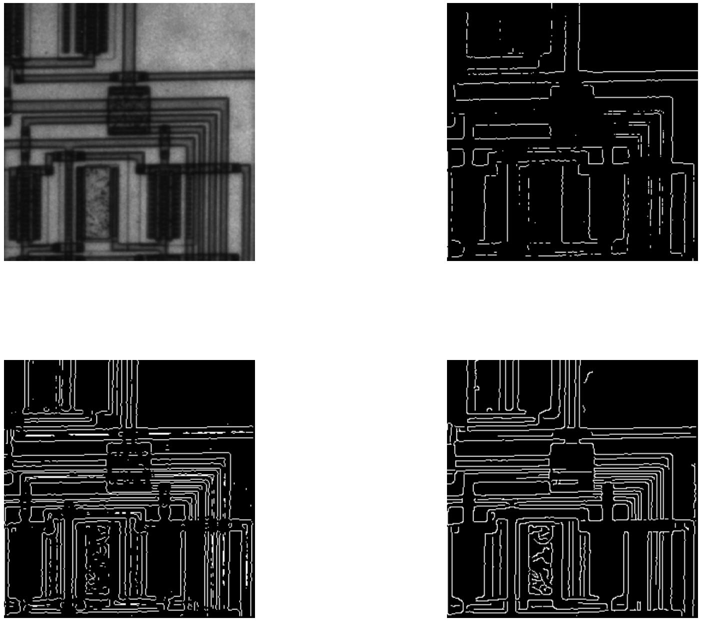
1. 使用高斯滤波器 $G(x,y)=\frac{1}{2\pi\sigma^2}e^{\frac{-(x^2+y^2)}{2\sigma^2}}$ 与图像卷积，目的是平滑图像，减少噪声。
2. 在每个点计算局部梯度和边缘方向。
3. 使用非最大值抑制来消除边缘检测带来的杂散效应。
4. 使用双阈值来确定真实和潜在的边缘。
5. 最后通过抑制孤立的弱边缘来完成边缘检测。

- canny方法

  - ✓ BW=edge(I,'canny',thresh,sigma)
  - ✓ thresh若为两个元素的向量，第一个元素为低阈值，第二个为高阈值。若为标量，则为高阈值。sigma为高斯滤波器的标准差，默认为1
  - ✓ canny方法使用了两个梯度的阈值，检测的连续性较好，是这几种方法最好的。

图像边缘检测

```
I = imread('circuit.tif');%读取图像
BW1 = edge(I,'prewitt');%使用prewitt方法检测边缘
BW2 = edge(I,'log');%使用log方法检测边缘
BW3 = edge(I, 'canny');%使用canny方法检测边缘
subplot(221), imshow(I);%显示原图像
subplot(222), imshow(BW1);%显示prewitt方法检测的边缘
subplot(223), imshow(BW2);%显示log方法检测的边缘
subplot(224), imshow(BW3);%显示canny方法检测的边缘
```



✓ 能量泛函的变分计算

设能量泛函 E 为

$$E(u)=\int_{\Omega}f(x,u(x),\nabla u(x))dx,(1.1.1)$$

其中f关于 u 和 ∇u是 C^1 的，记$\xi=\nabla u$，则该泛函的一阶变分为

$$\frac{\delta E}{\delta u}=\frac{\partial f}{\partial u}(x,u,\nabla u)-\sum_i\frac{\partial}{\partial x_i}\left(\frac{\partial f(x,u(x),\nabla u(x))}{\partial\xi}\right).(1.1.2)$$

计算过程如下：

对任意的 $v(x)\in C_0^2(\Omega)$，考虑关于 u 的变分 $u(x)+tv(x)$ 的能量泛函：

$$E(u+tv)=\int_{\Omega}f(x,u(x)+tv(x),\nabla u(x)+t\nabla v(x))dx.(1.1.3)$$

计算：

$$\left.\frac{dE(u)}{dt}\right|_{t=0}=\int_{\Omega}\left(\frac{\partial f}{\partial u}v+\frac{\partial f}{\partial\xi}\nabla v\right)dx.(1.1.4)$$

对第二项积分利用Green公式

$$\int_{\Omega}\nabla u\nabla v dx=-\int_{\Omega}v\Delta u dx+\int_{\partial\Omega}\frac{\partial u}{\partial v}v ds,(1.1.5)$$

其中v是Ω 的边界$\partial\Omega$ 的外法向量，有

$$\int_\Omega \frac{\partial f}{\partial \xi}\nabla v dx = -\int_\Omega \sum_i \frac{\partial}{\partial x_i}\left(\frac{\partial f(x,u(x),\nabla u(x))}{\partial \xi}\right) v dx + \int_{\partial\Omega} \frac{\partial f}{\partial v} v ds$$

$$= -\int_\Omega \sum_i \frac{\partial}{\partial x_i}\left(\frac{\partial f(x,u(x),\nabla u(x))}{\partial \xi}\right) v dx, \quad (1.1.6)$$

这里用到 u 在边界上为0，则

$$\left.\frac{\mathrm{d}E(u)}{\mathrm{d}t}\right|_{t=0} = \int_\Omega \left(\frac{\partial f}{\partial u} - \sum_i \frac{\partial}{\partial x_i}\left(\frac{\partial f(x,u(x),\nabla u(x))}{\partial \xi}\right)\right) v dx. \quad (1.1.7)$$

这样，得到 E 的一阶变分为

$$\frac{\partial E}{\partial u} = \frac{\partial f}{\partial u}(x,u,\nabla u) - \sum_i \frac{\partial}{\partial x_i}\left(\frac{\partial f(x,u(x),\nabla u(x))}{\partial \xi}\right). \quad (1.1.8)$$

定义1.1 假设X是一个巴拿赫空间，$E: X \to \mathbb{R}$，E在u点沿着v方向的加托导数定义为

$$E'(u;v) = \lim_{t\to 0} \frac{E(u+tv)-E(u)}{t}, (1.1.9)$$

如果对于任意的$v \in X$，此极限存在，则称E在$u \in X$有加托导数。

若E有加托导数，且极小化问题$\min_{v\in X} E(v)$有解u，那么

$$E'(u) = 0.$$

反之，若E是凸的，那么$E'(u)=0$的解u是极小化问题的解。称

$$E'(u) = 0 \text{ 或 } \frac{\delta E}{\delta u} = 0 (1.1.10)$$

为欧拉-拉格朗日（Euler-Lagrange，E-L）方程。

梯度下降流

变分问题

$$\min_{u\in X} E(u)(1.1.11)$$

所对应的梯度下降流为

$$\frac{\partial u}{\partial t} = -\frac{\delta E}{\delta u}, (1.1.12)$$

其中 $-\dfrac{\delta E}{\delta u}$ 是能量泛函 E 下降/减少的方向。

这是由于一方面，

$$E^{'}(u;v) = \lim_{t\to 0}\frac{E(u+tv)-E(u)}{t} = \int_{\Omega}\frac{\delta E}{\delta u} v dx. (1.1.13)$$

当 $E^{'}(u;v)<0$ 时，能量是减小的。
另一方面，又注意到式(1.1.7)中 v 是任意的，不妨设 $v=\dfrac{\partial u}{\partial t}$，此时若 $\dfrac{\partial u}{\partial t}=-\dfrac{\delta E}{\delta u}$，则 $E^{'}(u;v)<0$，且 E 下降得最快。

下面介绍另外一种计算能量泛函 E 所对应得梯度下降流的方法。

引入时间变量 t，则能量泛函是关于时间变量的函数，即

$$E(u(x,t)) = \int_{\Omega} f(x,u(x,t),\nabla u(x,t))dx. (1.1.14)$$

下面将$E(u(x,t))$简记为$E(t)$。

一般来说，需要求能量泛函E的极小化问题，那么能量泛函必须随着时间 t 的增大而减小，即需要$\dfrac{\partial E}{\partial t}<0$，而

$$\begin{aligned}\frac{\partial E}{\partial t}&=\int_\Omega\left(\frac{\partial f}{\partial u}u_t+\frac{\partial f}{\partial \xi}(\nabla u)_t\right)dx\\&=\int_\Omega\left(\frac{\partial f}{\partial u}u_t-\sum_i\frac{\partial}{\partial \xi}\left(\frac{\partial f}{\partial \xi}\right)u_t\right)dx+\int_{\partial\Omega}\frac{\partial f}{\partial v}u_t ds\\&=\int_\Omega\left(\frac{\partial f}{\partial u}u_t-\sum_i\frac{\partial}{\partial \xi}\left(\frac{\partial f}{\partial \xi}\right)u_t\right)dx.\,(1.1.15)\end{aligned}$$

式(1.1.15)中第二个等式用到了Green公式，其中 v 表示Ω的边界$\partial\Omega$的外法向量，第三个等式用到了Neumann边界条件

$$\frac{\partial f}{\partial v}=0.(1.1.16)$$

梯度下降法

为了使 $\dfrac{\partial E}{\partial t} < 0$，选取

$$\frac{\partial u}{\partial t} = -\left(\frac{\partial f}{\partial u} - \sum_i \frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial \xi}\right)\right). \quad (1.1.17)$$

这与先求E–L方程，再通过梯度下降的方法求能量泛函的梯度下降流所得的结果是一致的。这种方法也常常用来求能量泛函极值问题的梯度下降流。