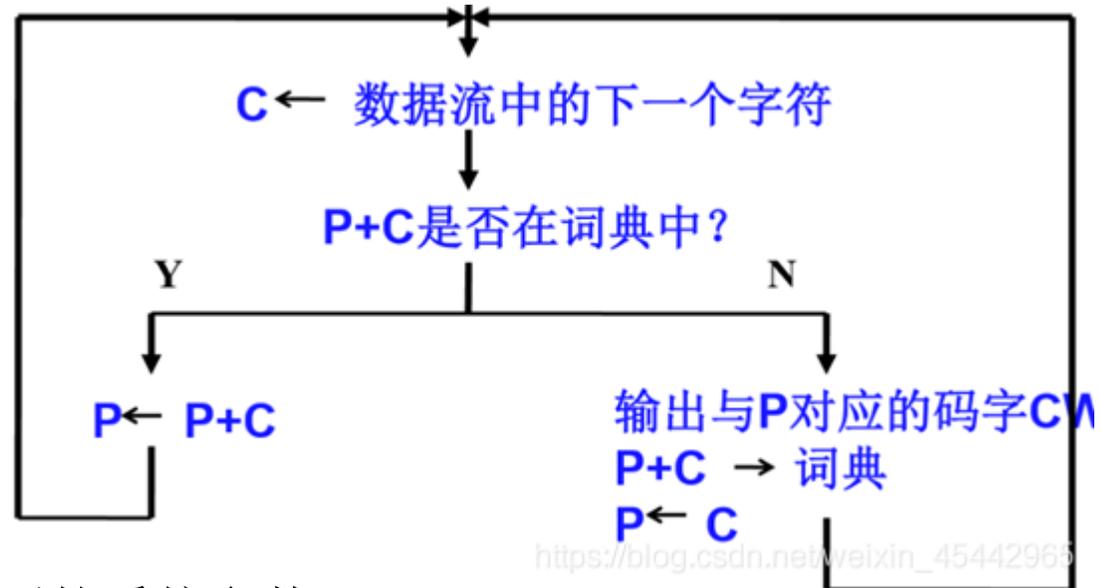


依次读入原文件的字符序列，每次碰到新的连续的字符串，就在字典中加入标示，当下次再次遇到这种字符串时，就可以用字典索引序号直接代替字符串，写入压缩文件中



□ 字典大小是一个重要的系统参数

- 如果字典太小，那么匹配灰度级序列的检测将不太可能成功
- 如果字典太大，那么码子的大小反过来会影响压缩性能

□ 编码字典是在对数据进行编码的同时创建的。在LZW解码器对编码后的数据流进行解码的同时，创建了一个同样的解压缩字典

ababcababac 初始字典

Symbol	String
0	a
1	b
2	c

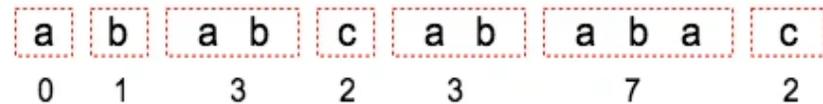
Symbol	String
0	a
1	b
2	c
3	ab
4	ba
5	abc
6	ca
7	aba
8	abac



过程中产生的字典

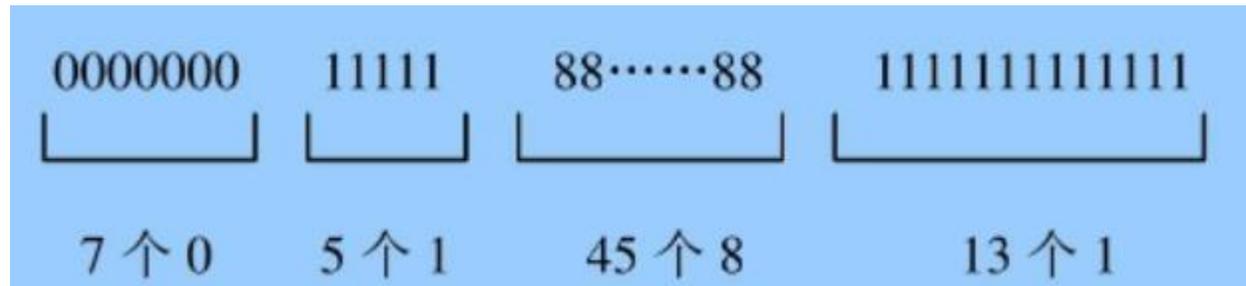
Step	P	C	P+C	P+C in Dict ?	Action	Output
1	-	a	a	Yes	更新P=a	-
2	a	b	ab	No	添加3->ab, 更新P=b	0
3	b	a	ba	No	添加4->ba, 更新P=a	1
4	a	b	ab	Yes	更新P=ab	-
5	ab	c	abc	No	添加5->abc, 更新P=c	3
6	c	a	ca	No	添加6->ca, 更新P=a	2

7	a	b	ab	Yes	更新P=ab	-
8	ab	a	aba	No	添加7->aba, 更新P=a	3
9	a	b	ab	Yes	更新P=ab	-
10	ab	a	aba	Yes	更新P=aba	-
11	aba	c	abac	No	添加8->abac, 更新P=c	7
12	c	-	-	-	-	2



行程编码

- 处理行（或列）**重复灰度**的图像
- 行程对指定**一个新灰度的开始**和**具有该灰度的连续像素的数量**
- 主要适用于压缩二值图像
- BMP文件格式的RLE是行程编码的一种形式
- 两种古老且应用最广的二值图像压缩标准CCITT Group3和4采用了行程编码





适合行程编码的图像

适用性

单一图像 — 二值图像；打印文件；地图；工程图；等高图...

比特平面编码

- 将一幅多级图像分解为一系列二值图像，然后使用二值压缩方法进行编码
- 两种分解方法
 - 基2多项式

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \dots + a_12^1 + a_02^0$$

- m比特格雷码表示图像

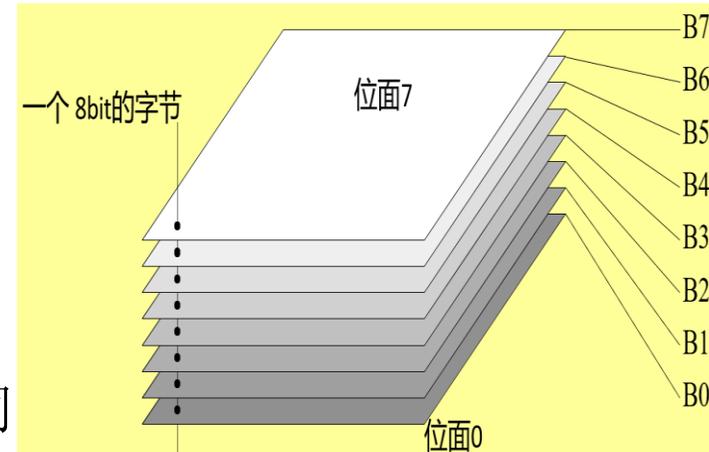
- ✓ 将m比特分解为格雷码 $g_{m-1} \dots g_2 g_1 g_0$

$$g_i = a_i \oplus a_{i+1}, 0 \leq i \leq m - 2$$

$$g_{m-1} = a_{m-1}$$

- 特点：连续码字只有一个比特位不同

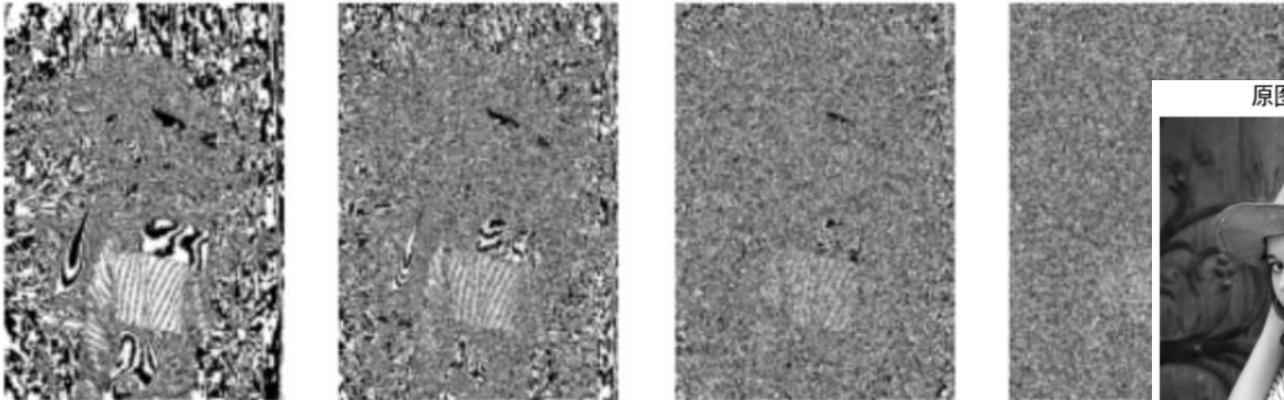
- ✓ 例如，127和128的格雷码分别是01000000和11000000



位平面第1层图像 位平面第2层图像 位平面第3层图像 位平面第4层图像



位平面第5层图像 位平面第6层图像 位平面第7层图像 位平面第8层图像



原图灰度图像



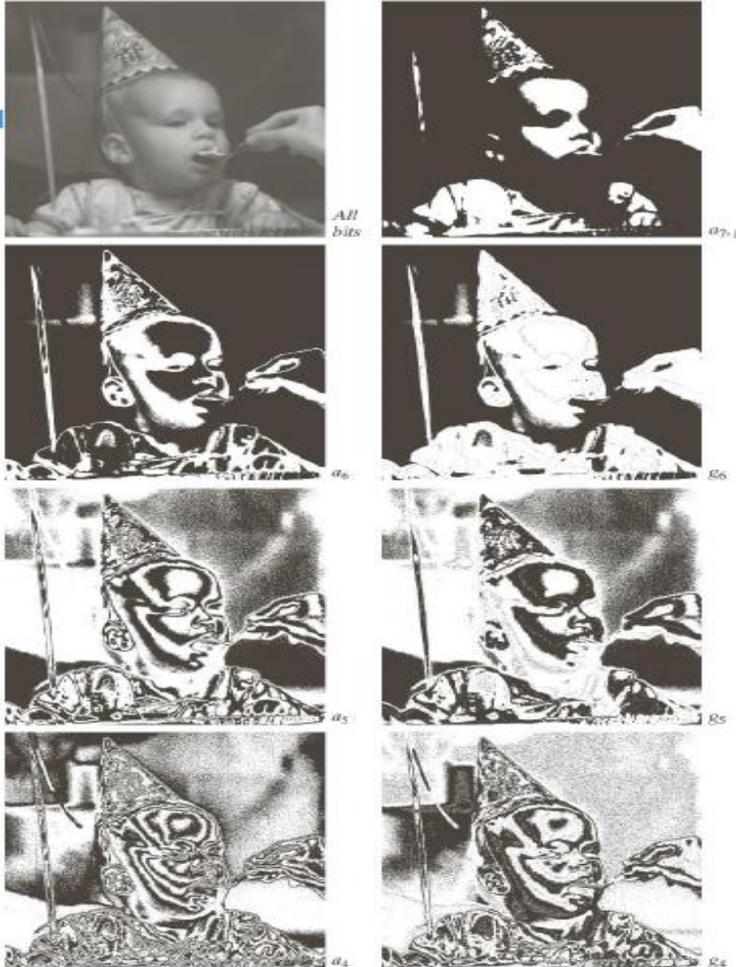
解压后



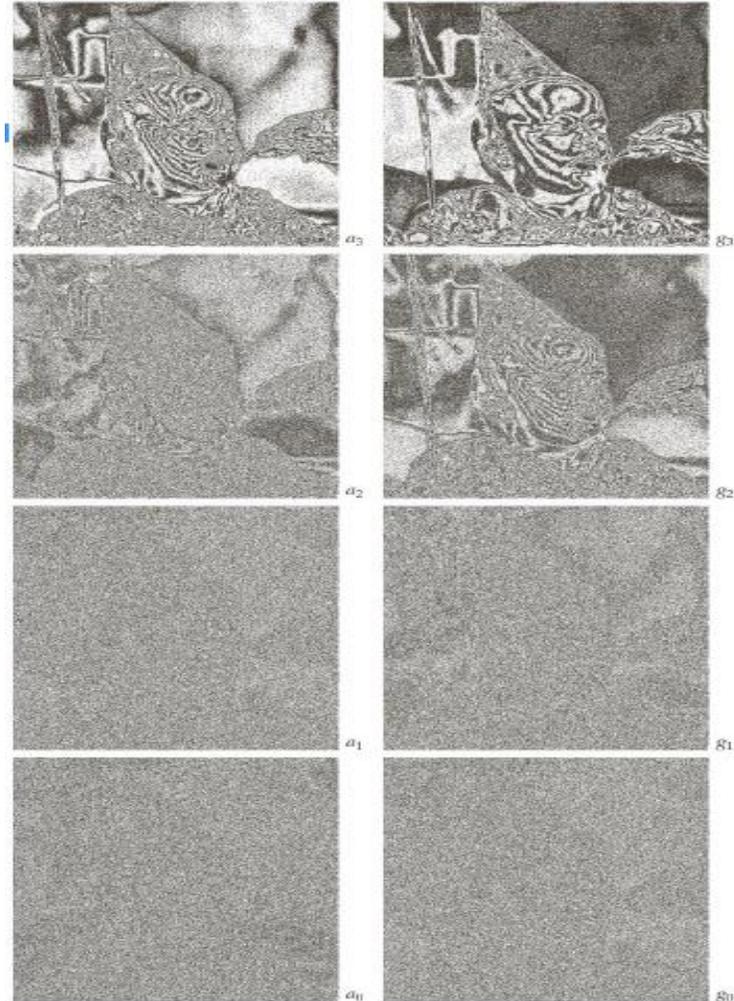
比特平面编码

(a) 一幅256比特单色图像

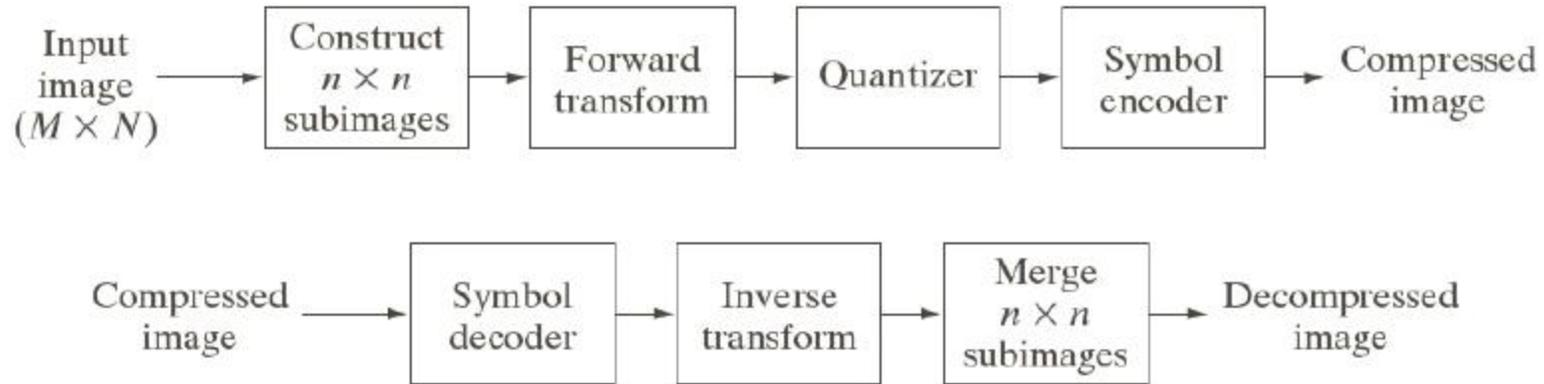
(b)-(h) 图(a)中图像的4个最高有效二进制编码和格雷编码比特平面



(a)-(h) 图(a)中图像的4个最低有效二进制编码和格雷编码比特平面



块变换编码



块变换编码系统

- 在给定的应用中，特定变换的选择取决于可容忍的重建误差大小和可用的计算资源
- 压缩是在对变换系数的量化期间实现的

$$\begin{aligned} \text{正变换} \quad T(u, v) &= \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} g(x, y) r(x, y, u, v) \\ \text{反变换} \quad g(x, y) &= \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) s(x, y, u, v) \end{aligned}$$

块变换编码

□ Walsh-Hadamard变换 (WHT)

$$r(x, y, u, v) = s(x, y, u, v) = \frac{1}{n} \sum_{i=0}^{m-1} [b_i(x) p_i(u) + b_i(y) p_i(v)]$$

$$n = 2^m$$

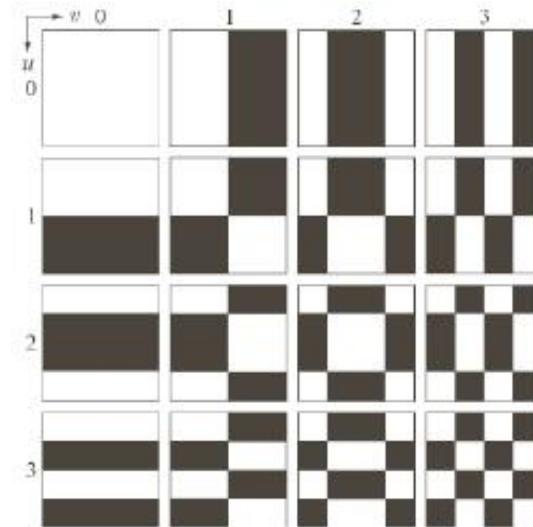
$b_k(z)$ 是 z 的二进制表示中的第 k 位

$$p_0(u) = b_{m-1}(u)$$

$$p_1(u) = b_{m-1}(u) + b_{m-2}(u)$$

...

$$p_{m-1}(u) = b_1(u) + b_0(u)$$



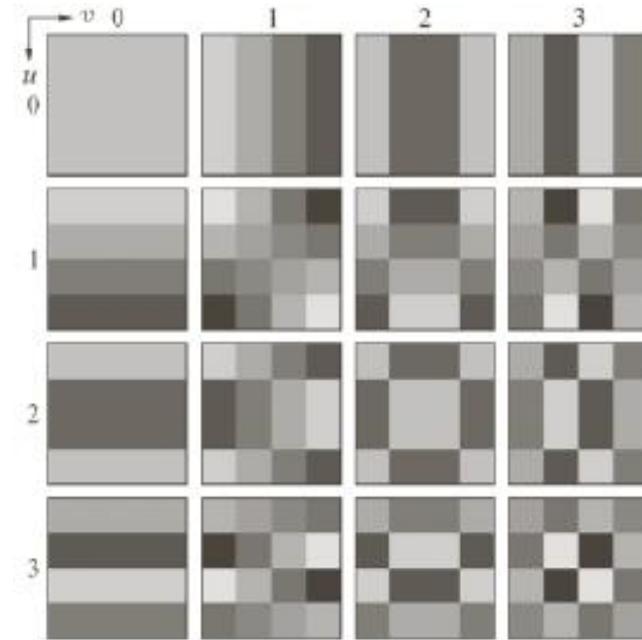
n=4时的WHT的基函数

块变换编码

□ 离散余弦变换 (DCT)

$$r(x, y, u, v) = s(x, y, u, v) = \alpha(u)\alpha(v) \cos\left[\frac{(2x+1)u\pi}{2n}\right] \cos\left[\frac{(2y+1)v\pi}{2n}\right]$$

$$\alpha(u) = \begin{cases} \sqrt{1/n}, u = 0 \\ \sqrt{2/n}, u = 1, 2, \dots, n-1 \end{cases}$$



n=4时的DCT的基函数

块变换编码

先分8*8图像块，
然后进行变换，
对变换后最小的
32个系数置零，
利用剩下的32个
大系数进行重构

重建误差：
傅里叶： 2.32；
WHT： 1.78；
DCT： 1.13



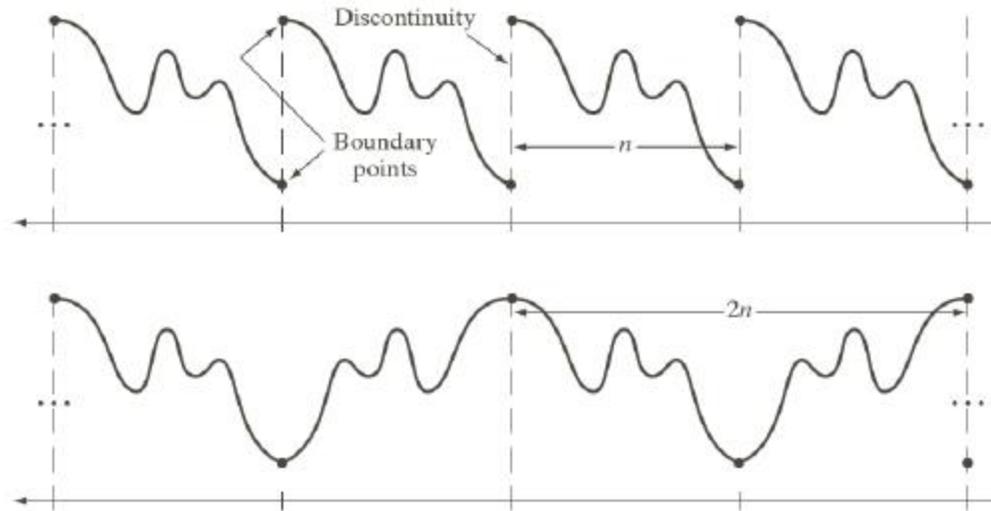
a b c
d e f

FIGURE 8.24 Approximations of Fig. 8.9(a) using the (a) Fourier, (b) Walsh-Hadamard, and (c) cosine transforms, together with the corresponding scaled error images in (d)–(f).

块变换编码

□ DCT的优点

- 用单片集成电路即可实现
- 可将最多信息装入最少系数中
- 可使块效应最小化

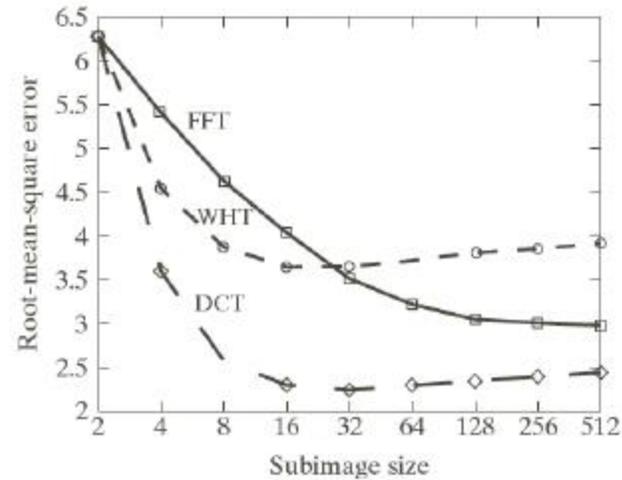


(a)一维DFT中隐含的周期性 (b)一维DCT中隐含的周期性

块变换编码

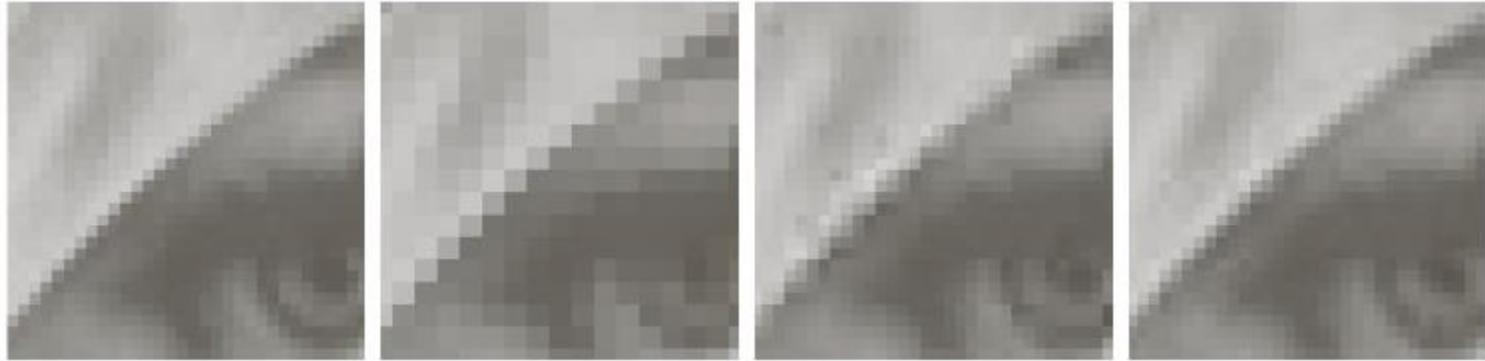
□ 子图像尺寸的选择

- 子图像的尺寸是影响编码误差和计算复杂性的重要因素
- 一般压缩效率和计算复杂度随子图像尺寸的增加而增加
- 最常用的子图像尺寸为 8×8 和 16×16



重建误差与子图像尺寸的关系

块变换编码



使用25%DCT系数对原图的近似

(a)原图

(b)使用 2×2 大小的子图的近似

(c)使用 4×4 大小的子图的近似

(d) 使用 8×8 大小的子图的近似

块变换编码

□ 编码方式

■ 区域编码

✓ 保留的系数是根据最大方差进行选择

■ 阈值编码

✓ 保留的系数是根据最大幅度进行选择

□ 比特分配

■ 对变换后的子图像的系数进行截取、量化和编码的过程



a b
c d

FIGURE 8.28
Approximations
of Fig. 8.9(a) using
12.5% of the
 8×8 DCT
coefficients:
(a)–(b) threshold
coding results;
(c)–(d) zonal
coding results. The
difference images
are scaled by 4.



块变换编码

□ JPEG标准

- 广泛使用的连续色调静止帧压缩标准
- JPEG定义了三种编码系统
 - ✓ 有损的基本编码系统，以DCT为基础
 - ✓ 扩展的编码系统，用于更大压缩、更高精度或渐进式重建
 - ✓ 面向可逆压缩的无损独立编码系统
- JPEG顺序基线系统
 - ✓ 输入和输出数据精度为8比特
 - ✓ 压缩步骤：DCT计算、量化、变长编码

块变换编码

□ JPEG基本编码与解码

■ 8 × 8子图像

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

■ 移位128个灰度级

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

块变换编码

□ JPEG基本编码与解码

■ 前向DCT

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

■ 量化、缩放和截断

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

块变换编码

□ JPEG基本编码与解码

■ Z型扫描系数重排

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

■ 完整编码阵列

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001
001 100101 11100110 110110 0110 11110100 000 1010

■ 重新生成的量化系数阵列

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

块变换编码

□ JPEG基本编码与解码

■ 反量化化

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

■ 反DCT变换

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59
-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44

块变换编码

□ JPEG基本编码与解码

■ 移位128个灰度级

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

■ 原图与重建的差别

-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	6	4	3	-2
3	8	4	-4	2	6	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

块变换编码

□ JPEG基本编码与解码



a b c
d e f

FIGURE 8.32 Two JPEG approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image.

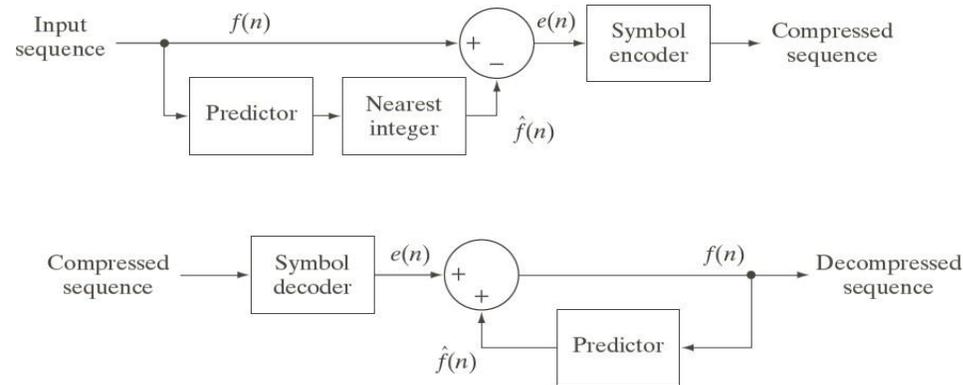
第一个结果压缩率25:1，第二个结果压缩率52:1

预测编码

□ 概述

- 预测编码通过消除**紧邻像素在空间和时间上的冗余**来实现
- 一个像素的新信息定义为实际值和预测值的差值

□ 无损预测编码



a
b

FIGURE 8.33
A lossless
predictive coding
model:
(a) encoder;
(b) decoder.

- 预测误差
$$e(n) = f(n) - \hat{f}(n)$$

- 重建
$$f(n) = e(n) + \hat{f}(n)$$

预测编码

□ 无损预测编码

- 多数情况下，预测可由m个样值进行线性组合

$$\hat{f}(n) = \text{round} \left[\sum_{i=1}^m \alpha_i f(n-i) \right]$$

- 对一幅图像，预测值可以是来自当前扫描行（一维线性预测编码）、当前行和前几个扫描行
- 一维线性预测图像编码

$$\hat{f}(x, y) = \text{round} \left[\sum_{i=1}^m \alpha_i f(x, y-i) \right]$$

预测编码

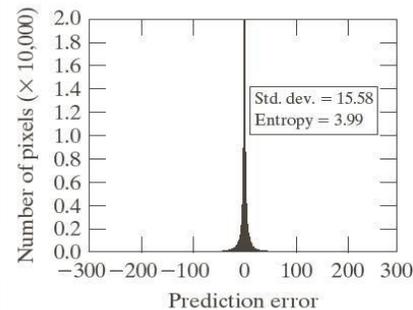
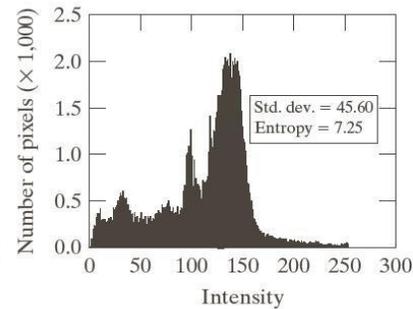
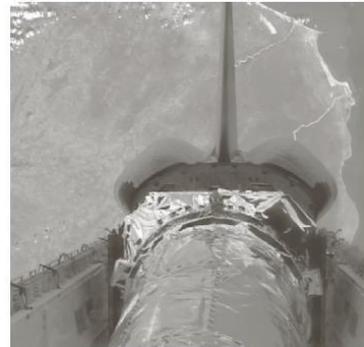
□ 无损预测编码

■ 一维线性预测图像编码示例1

$$\hat{f}(x, y) = \text{round} \left[\sum_{i=1}^m \alpha_i f(x, y - i) \right] \quad m = 1$$

a b
c d

FIGURE 8.34
(a) A view of the Earth from an orbiting space shuttle. (b) The intensity histogram of (a). (c) The prediction error image resulting from Eq. (8.2-34). (d) A histogram of the prediction error. (Original image courtesy of NASA.)



零均值拉普拉斯分布

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{-\frac{\sqrt{2}|e|}{\sigma_e}}$$

预测编码

□ 运动补偿预测残差

- 帧序列包含快速运动的物体
- 相减的方式不再有效
- 运动补偿广泛用于视频编码中

□ 运动补偿基本原理

- 将视频帧分为互不重叠的子块，称为宏块
- 每个宏块在前一帧（称为参考帧）中找到最合适的块
- 使用运动矢量表示当前宏块在参考帧的位置

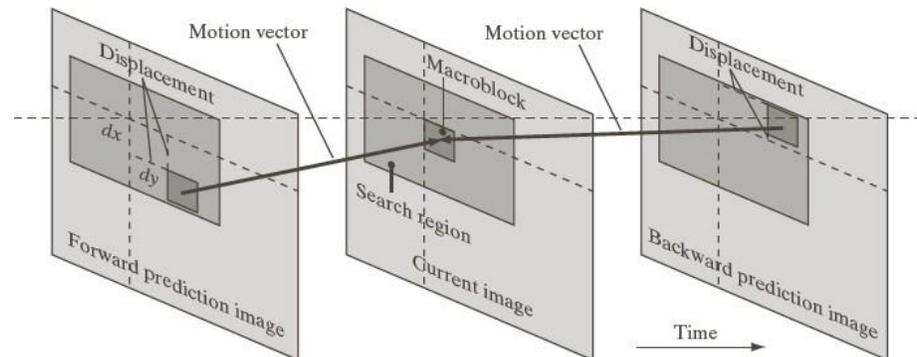
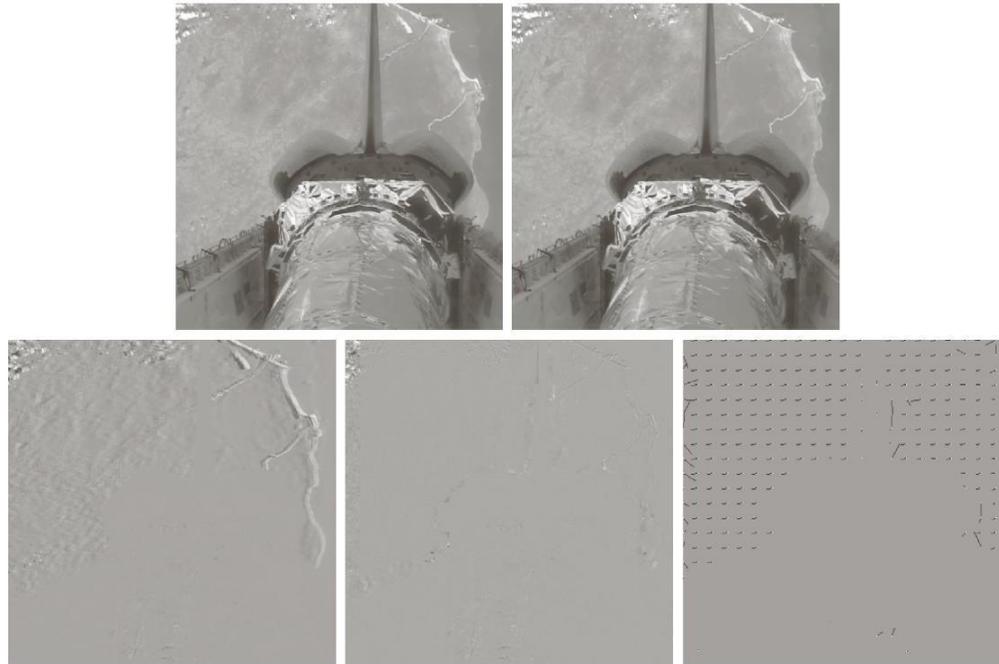


FIGURE 8.36
Macroblock
motion
specification.

预测编码

□ 运动补偿预测示例 1



c) 预测残差: 12.73灰
度级; 熵: 4.17bit/pix;
C=1.92

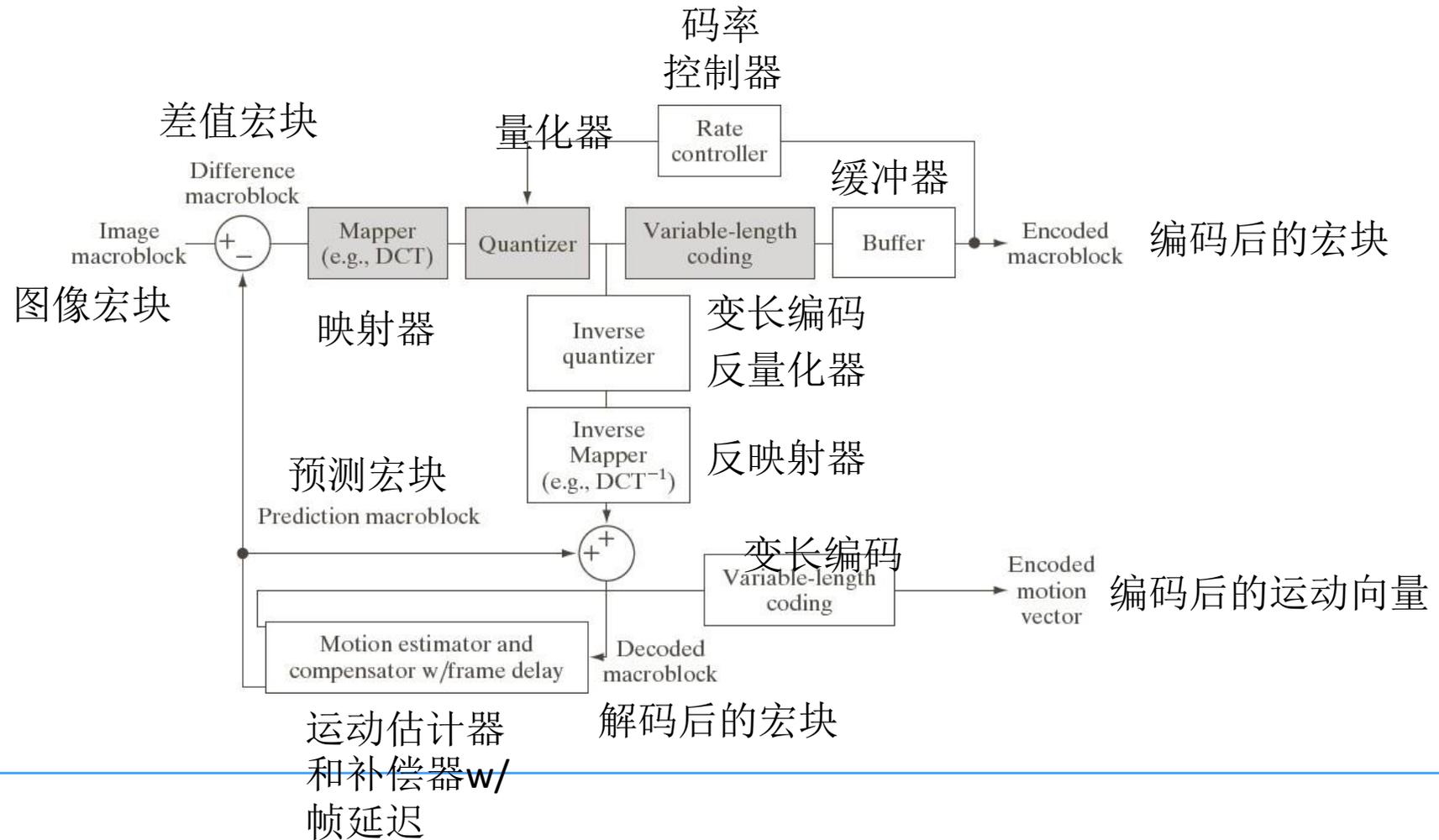
d) 预测残差: 5.62灰度
级; 熵: 3.04bit/pix;
C=2.63

a b
c d e

FIGURE 8.37 (a) and (b) Two views of Earth that are thirteen frames apart in an orbiting space shuttle video. (c) A prediction error image without motion compensation. (d) The prediction residual with motion compensation. (e) The motion vectors associated with (d). The white dots in (d) represent the arrow heads of the motion vectors that are depicted. (Original images courtesy of NASA.)

预测编码

□ 典型的基于运动补偿的视频编码器



预测编码

□ 视频压缩

5G
↓
44.56M

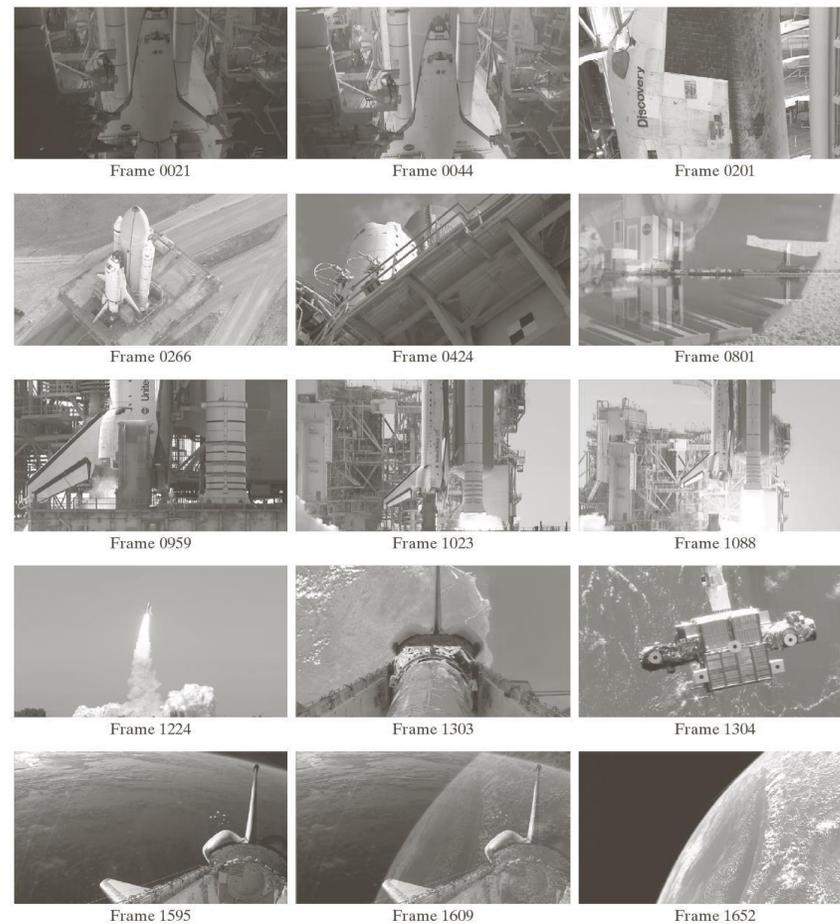
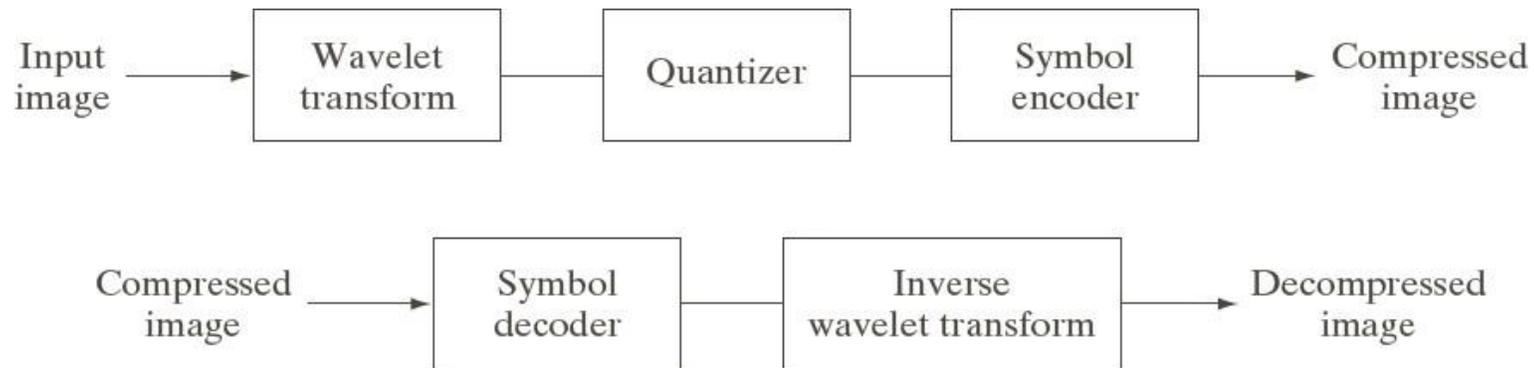


FIGURE 8.40 Fifteen frames from an 1829-frame, 1-minute NASA video. The original video is in HD full color. (Courtesy of NASA.)

小波编码

□ 概念

- 对图像像素解除相关的变换系数进行编码，比对原图像像素本身进行编码效率更高
- 选择合适的变换基函数（小波函数），将大多数重要的视觉信息包装到少量系数中，剩下的系数可以被粗略量化甚至截取为0，而图像几乎没有失真



小波编码

□ 小波基

■ 图像编码中广泛使用的小波有：

- ✓ Harr小波
- ✓ Daubechies小波
- ✓ 对称小波
- ✓ 双正交小波

TABLE 8.13

Wavelet transform filter taps and zeroed coefficients when truncating the transforms in Fig. 8.46 below 1.5.

Wavelet	Filter Taps (Scaling + Wavelet)	Zeroed Coefficients
Haar (see Ex. 7.10)	2 + 2	33.8%
Daubechies (see Fig. 7.8)	8 + 8	40.9%
Symlet (see Fig. 7.26)	8 + 8	41.2%
Biorthogonal (see Fig. 7.39)	17 + 11	42.1%

a b
c d

FIGURE 8.46
Three-scale wavelet transforms of Fig. 8.9(a) with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies Feauveau biorthogonal wavelets.



小于1.5系数截短为0



小波编码

- JPEG2000: 基于小波变换的图像编码标准
 - JPEG2000扩充了JPEG标准，在连续色调静止图像压缩和压缩数据的访问提供了更多灵活性
 - JPEG2000编码步骤
 - 对编码的图像样本进行直流电平平移和去相关性
 - 将各个分量分成多个像块，即被单独处理的像素的矩阵阵列
 - 计算每个像块分量的行和列的一维离散小波变换



小波编码

□ JPEG2000

□ JPEG2000编码步骤

- 计算每个像块分量的行和列的一维离散小波变换

Filter Tap	Highpass Wavelet Coefficient	Lowpass Scaling Coefficient
0	-1.115087052456994	0.6029490182363579
±1	0.5912717631142470	0.2668641184428723
±2	0.05754352622849957	-0.07822326652898785
±3	-0.09127176311424948	-0.01686411844287495
±4	0	0.02674875741080976

TABLE 8.15

Impulse responses of the low- and highpass analysis filters for an irreversible 9-7 wavelet transform.

小波编码

□ JPEG2000

□ JPEG2000编码步骤

- 变换生成四个子带，分别是像块分量的低分辨率近似及分量的水平、垂直和对角频率特征。将后续迭代限制为前面的分解的近似系数，把该变换重复N次，可得到N尺度小波变换

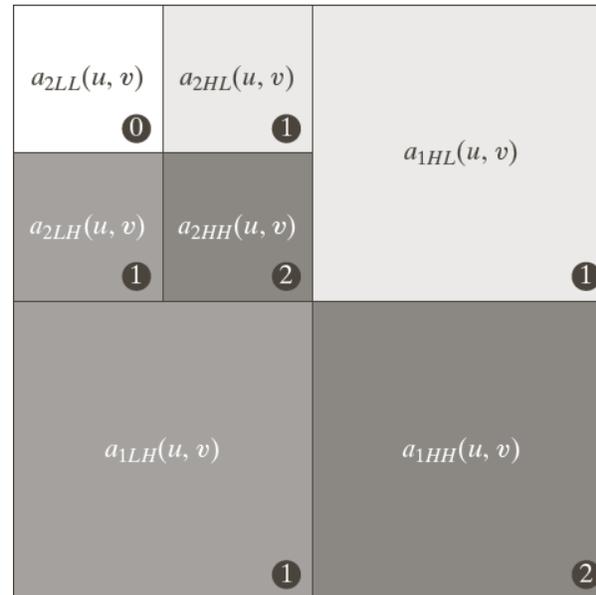


FIGURE 8.48
JPEG 2000
two-scale wavelet
transform
tile-component
coefficient
notation and
analysis gain.

小波编码

□ JPEG2000

从a)到d)压缩比C为25,
52, 75, 105;

a)误差: JPEG 5.4
J2000 3.86;

b) 误差: JPEG 10.7
J2000 5.77



FIGURE 8.49 Four JPEG-2000 approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image. (Compare the results in rows 1 and 2 with the JPEG results in Fig. 8.32.)

■ JPEG与JPEG 2000的性能比较

标准	JPEG	JPEG 2000
主要编码技术	离散余弦变换 (DCT) 知觉量化 Zigzag扫描 霍夫曼编码 算术编码	离散小波变化 (DWT) EBCOT核心算法 ROI编码 空间可扩展编码 质量可扩展编码 面向对象编码 位图形状编码 容错编码、TCQ、零数扫描
压缩比	2~30	2~50
算法效率	30:1以上急剧下降	100:1以上急剧衰减
速率失真特性		比JPEG提高30%

数字图像水印

- 数字图像水印处理是把数据插入到一幅图像中的过程，从而形成对图像的保护。
- 简单的可见水印：

$$f_w = (1 - \alpha)f + \alpha w$$

其中 f 为原图， w 为水印

Digital Image
Processing



a
b c

FIGURE 8.50
A simple visible watermark:
(a) watermark;
(b) the watermarked image; and (c) the difference between the watermarked image and the original (non-watermarked) image.

数字图像水印

□ 不可见水印

- 不可见水印肉眼是看不见的
- 使用合适的解码算法可以恢复原图
- 不可见水印是通过作为可见的冗余信息插入来保证的

$$f_w = 4 \left(\frac{f}{4} \right) + \frac{w}{64}$$

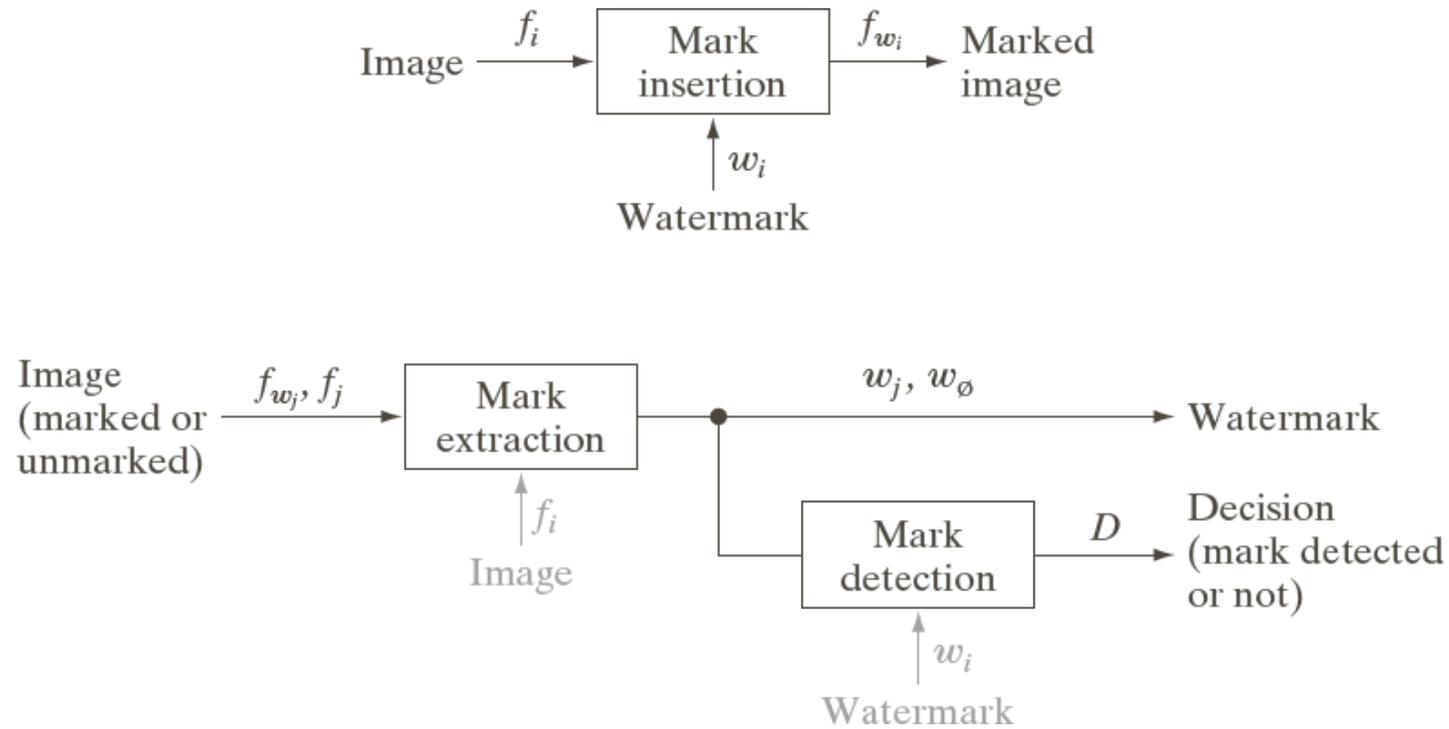


a b
c d

FIGURE 8.51 A simple invisible watermark: (a) watermarked image; (b) the extracted watermark; (c) the watermarked image after high quality JPEG compression and decompression; and (d) the extracted watermark from (c).

数字图像水印

□ 典型的图像水印处理系统







数字图像处理 (Digital Image Processing)

王正

Email: zheng_wang@seu.edu.cn
办公室: 无线谷A5415

**Image
Processing
Project Ideas**
Beginners

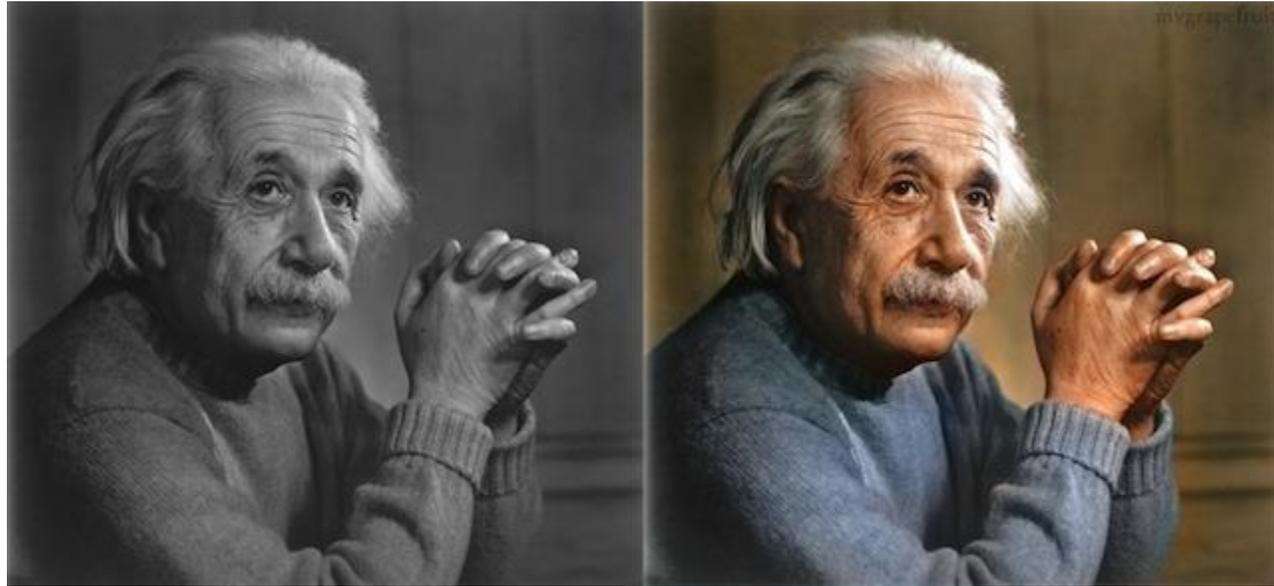


彩色图像处理

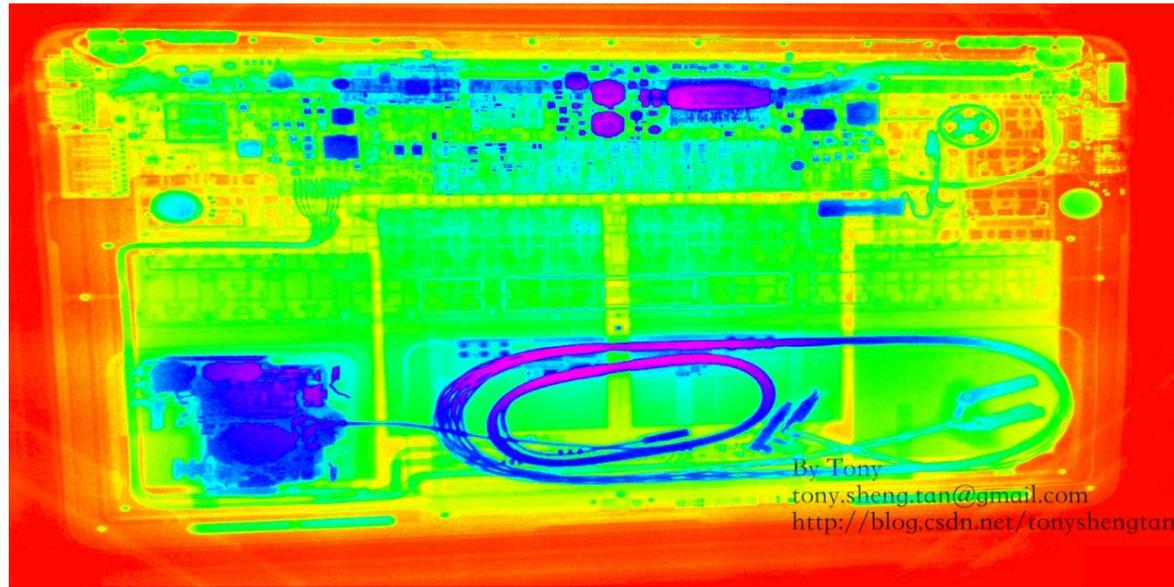
- 彩色是一个强有力的描绘子，常常可简化从场景中提取和识别目标



- 人可以辨别几千种彩色色调和亮度，但只能辨别几十种灰度色调



- 彩色图像处理可分为两个主要领域：全彩色处理和伪彩色处理
 - 全彩色处理：通常要求图像用全彩色传感器获取
 - 伪彩色处理：对一种特定的单色灰度或灰度范围赋予一种颜色



□ 前面几章介绍的一些灰度方法可直接用于彩色图像



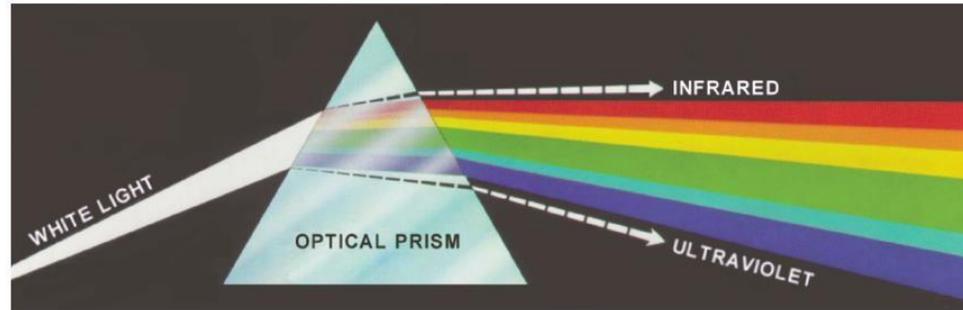


彩色图像处理

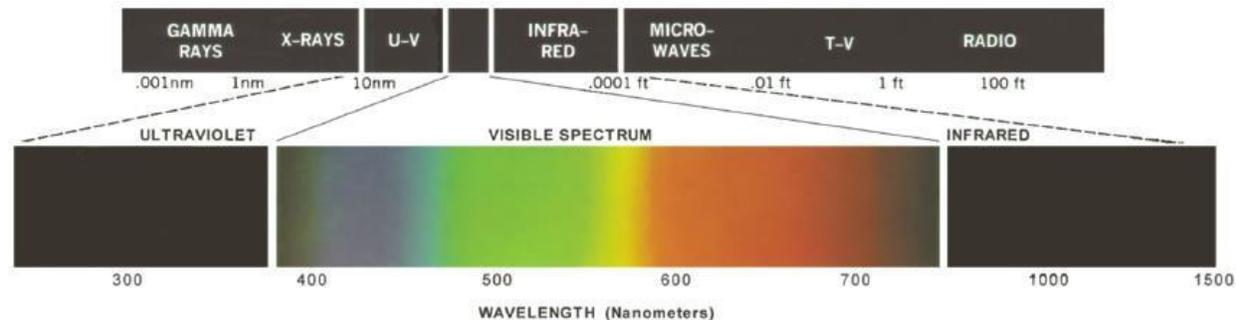
1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-

彩色基础

- 1661年，艾萨克·牛顿发现了太阳光的分解



- 彩色光大约覆盖电磁波谱中的400~700nm的范围
 - 色谱末端的颜色不是突变的，而是平滑过渡的
 - 一个物体反射有限的可见光谱时，则呈现某种颜色
 - 彩色光源质量描述：辐射，光强，亮度



辐射量是从光源流出能量的总量，通常是用瓦特度量。
光强用流明度量，它给出了观察者从光源接收的能量总和的度量。
亮度是一个主观描绘子，它实际上是不可度量的。它包含了无色的强度的概念，是描述彩色感觉的一个关键参数。



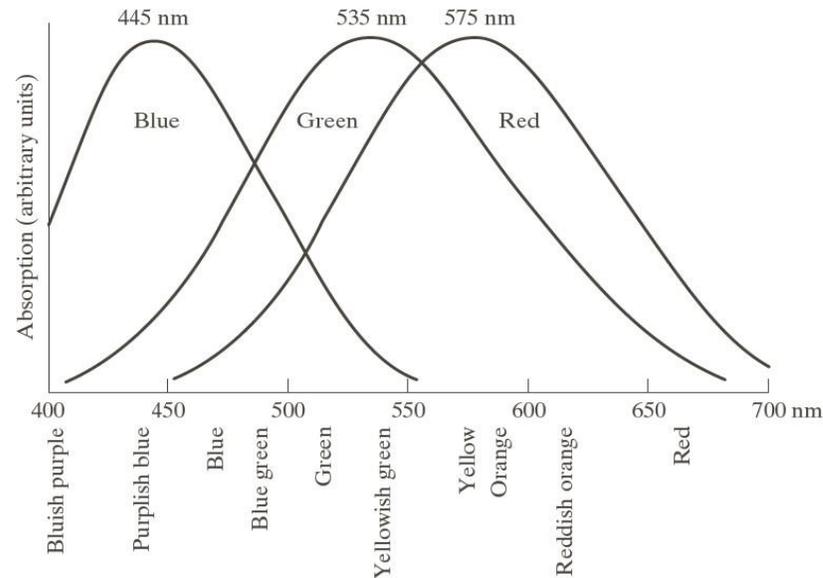
产品参数

主要参数	
投影尺寸: 30~300英寸	标准物理分辨率: 1920*1080P
真实流明: 850ANSI	投影技术: 5英寸LCD液晶屏
翻转功能: 360度翻转	投影镜头: 光学镀膜玻璃镜头
光源: LED, 3万小时寿命	四点梯形: 支持
工作电压: AC100-240V	CPU: 电视级海思64bit

彩色基础

□ 人的视觉

- 人眼中的锥状细胞是负责彩色视觉的传感器
- 人眼中600~700万个锥状细胞分为3个主要的感知类别，分别对应于红色、绿色和蓝色。65%细胞对红光敏感，33%对绿光敏感，只有2%对蓝光敏感



标准原色

蓝色: 435.8 nm

绿色: 546.1 nm

红色: 700 nm

以波长为函数，人眼中的红色、绿色和蓝色锥状细胞对光的吸收曲线

彩色基础

□ 光的原色与颜料的原色

□ 光的原色

- 红(R)、绿(G)、蓝(B)
- 满足加色原理
- 应用：CRT显示器

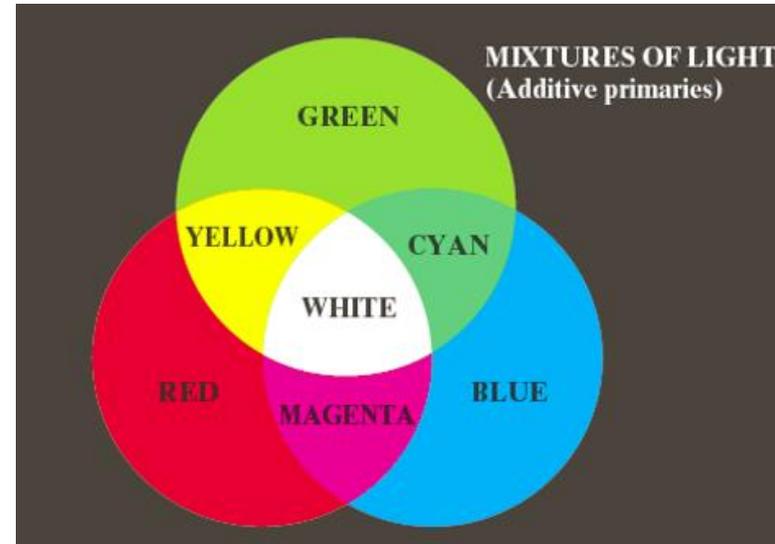
相加混色：

红 + 绿 = 黄

红 + 蓝 = 深红

蓝 + 绿 = 青

红 + 绿 + 蓝 = 白



但并不是这三个波长的原色可以产生所有的可见彩色

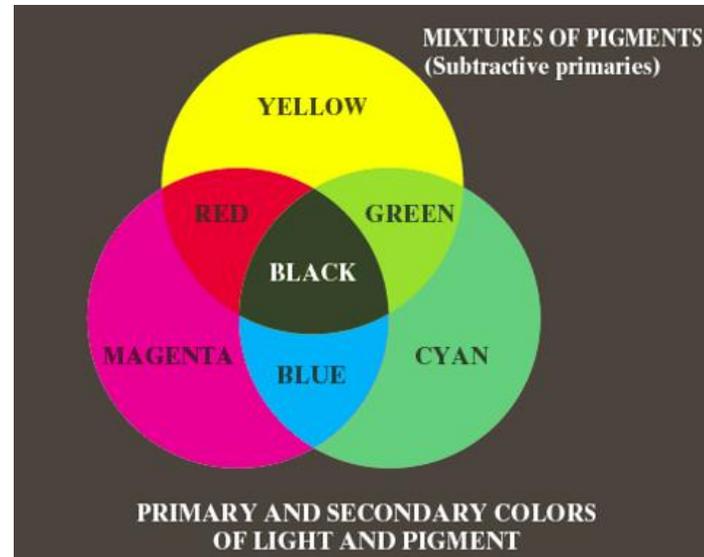
□ 颜料、着色剂的原色

- 深红色(M)、青色(C)、黄色(Y)
- 满足减色原理
- 应用：彩色打印机

减去或吸收光的一种原色并反射或传输另两种原色。

相减混色：

黄 = 白 - 蓝
深红 = 白 - 绿
青 = 白 - 红
白 - 蓝 - 绿 - 红 = 黑



色度图

□ 颜色的物理特征：亮度(Intensity)、色调(Hue)、饱和度(Saturation)

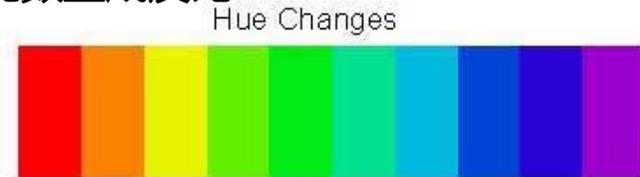
- 亮度：体现了无色的强度概念
- 色调：光波混合中与主波长有关的属性，观察者感知的主要颜色
- 饱和度：一种颜色混合白光数量，与所加白光数量成反比

亮度是色彩明亮度的概念，是光作用于人眼时所引起的明亮程度的感觉。亮度有两种特性：

同一物体因受光不同产生亮度上的变化；
强度相同的不同色光亮度感觉不同。

色调是光波混合中与主波长有关的属性，色调表示观察者接收的主要颜色。我们所说的物体呈现红色、黄色等颜色时，是指它的色调。

饱和度是指色彩纯粹的程度。纯谱色是全饱和的。像粉红（红加白）是欠饱和的，饱和度与所加白光数量成反比。



色调与饱和度一起称为色度，因此，颜色用亮度和色度表征

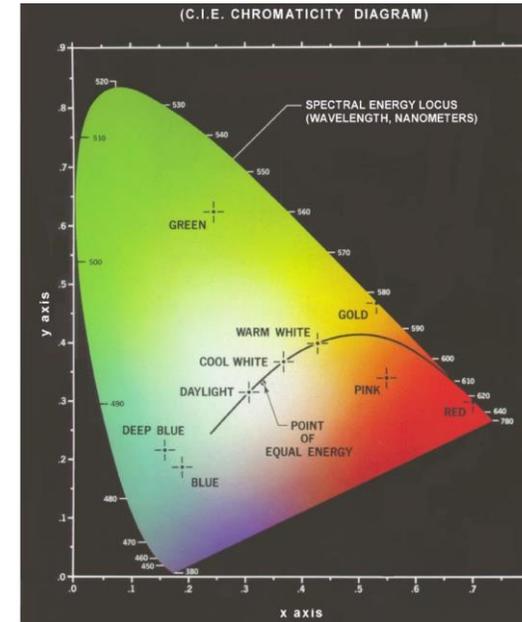
色度图

- 形成任何特殊颜色的红、绿、蓝的数量称为**三色值**，并分别表示为X、Y、Z，这样，一种颜色可由其三色值系数定义为：

$$x = \frac{X}{X + Y + Z}, y = \frac{Y}{X + Y + Z}, z = \frac{Z}{X + Y + Z}$$

- CIE色度图

- 以x和y的函数表示颜色的组成
- 相应的z值可由下式得到
 - ✓ $z = 1 - x - y$
- 由(x, y, z)即可在右图画相应颜色
- 色度图边界上的点为全饱和颜色

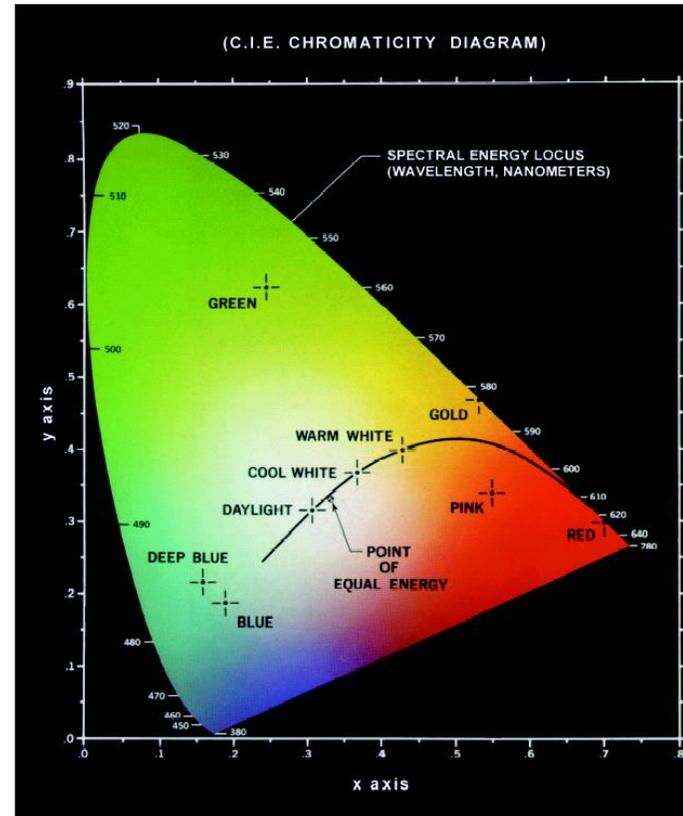


绝大部分光可由红、绿、蓝三种色光按一定比例加以合成

$$Y=0.299R+0.587G+0.114B$$

- 色度图中每点都对应一种可见光的颜色；
- 纯色在色度图边界上，任何不在边界上而在色度图内部的点都表示谱色的混合色；
- 色度图边界上的任何点都是全饱和的，越靠近中心点C饱和度越低，中心点处各种光谱能量相等而显为白色；
- 要确定三个给定的颜色所组合成的颜色，只需将三点连成三角形，三角形中任意颜色都可由这三色组成，而其他的颜色不能由这三色组成

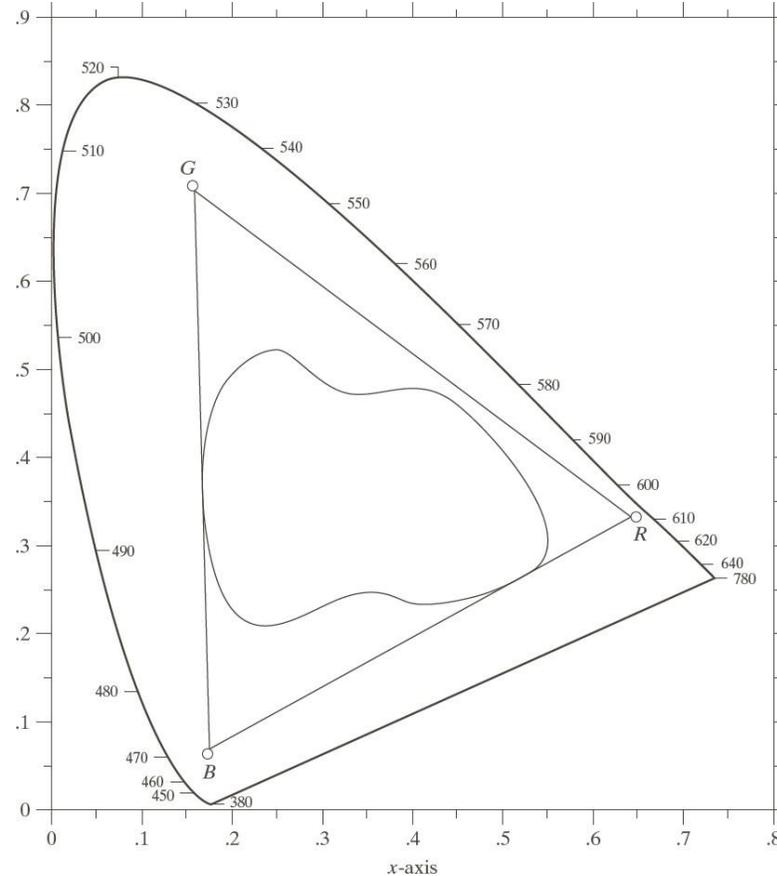
FIGURE 6.5
Chromaticity diagram.
(Courtesy of the General Electric Co., Lamp Business Division.)



彩色基础

□ 彩色监视器的彩色域

- 三个单一且固定的原色无法得到所有颜色
- 右图三角形为彩色显式器的典型彩色域
- 三角形内的不规则区域是今天高质量彩色打印设备的代表性彩色域
- 彩色打印设备的彩色域的边界是不规则的，因为彩色打印是相加色彩和相减色彩混合的组合，不易控制





彩色图像处理

1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-



彩色模型

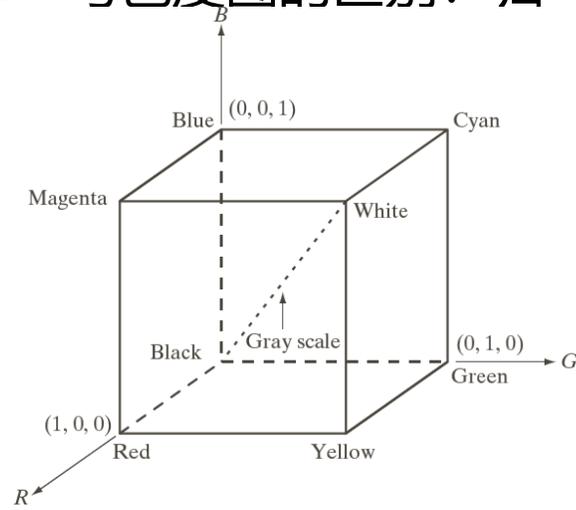
- 彩色模型（彩色空间、彩色系统）的目的：在某些标准下用通常可以接受的方式方便地对彩色加以说明
- 在数字图像处理中，最通用的模型如下：
 - 面向硬件的RGB（红、绿、蓝）模型，用于彩色摄像机
 - CMY（青、粉红、黄）模型和CMYK（青、粉红、黄、黑）模型，用于彩色打印机
 - HSI（色调、饱和度、亮度）模型，更符合人描述和解释颜色的方式

HSI模型还有一个优点，就是把图像分成彩色和灰度信息，使其更适合许多灰度处理技术

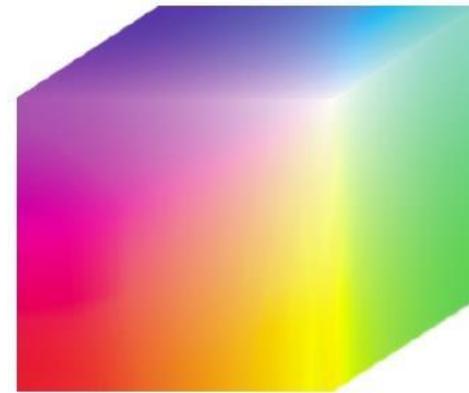
RGB彩色模型

□ RGB彩色模型

- 基于笛卡尔坐标系，三原色位于3个角上
- 在RGB空间中，表示每个像素的比特数称为像素深度
 - ✓ 考虑一幅RGB图像，其中每幅红绿蓝图像都是8比特图像，所以每个RGB像素有24比特的深度
 - **像素深度：用以表示每一个像素的比特数。**
- 与色度图的区别：归一化方式不一样



RGB彩色立方体



RGB 24比特彩色立方体 (实心)

Yves Klein





RGB 129/216/208



RGB 0/49/83



RGB 128/0/32



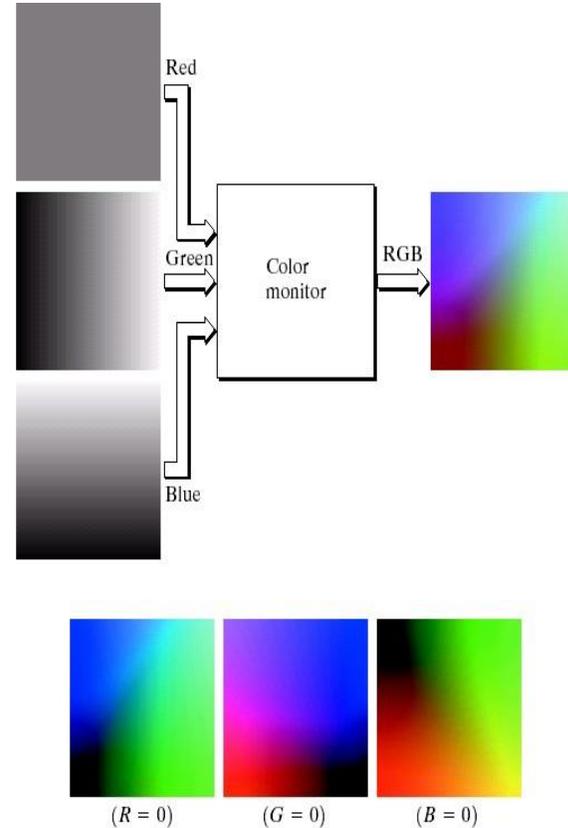
RGB 0/149/182

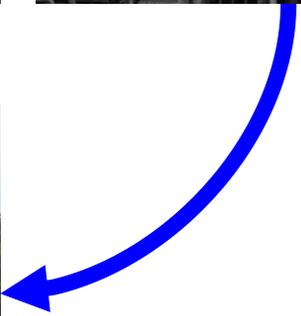
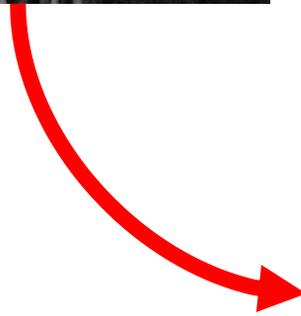


RGB彩色模型

当送入RGB监视器时，这三幅图像在荧光屏上混合产生一幅合成的彩色图像。在RGB空间，用以表示每一像素的比特数叫做**像素深度**。考虑RGB图像，其中每一幅红、绿、蓝图像都是一幅8比特图像，在这种条件下，每一个RGB彩色像素称为24比特深度。**全彩色**图像常用来定义24比特的彩色图像。24比特RGB图像中的颜色总数是 $(2^8)^3 = 16777216$ 。

a
b
FIGURE 6.9
(a) Generating the RGB image of the cross-sectional color plane $(127, G, B)$.
(b) The three hidden surface planes in the color cube of Fig. 6.8.





CMY和CMYK彩色模型

□ CMY彩色模型

- 青色、深红色、黄色是光的二次色，是颜料的原色
- 大多数在纸上沉积彩色颜料的设备，如彩色打印机和复印机，要求输入CMY数据或在内部进行RGB到CMY的转换：

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

□ CMYK模型

- 等量的颜料原色可以生成黑色，但为打印目的组合这些颜色产生的黑色是不纯的。
- 为了生成真正的黑色（在打印中起主要作用的颜色），加入了第4中颜色——黑色，提出了CMYK彩色模型

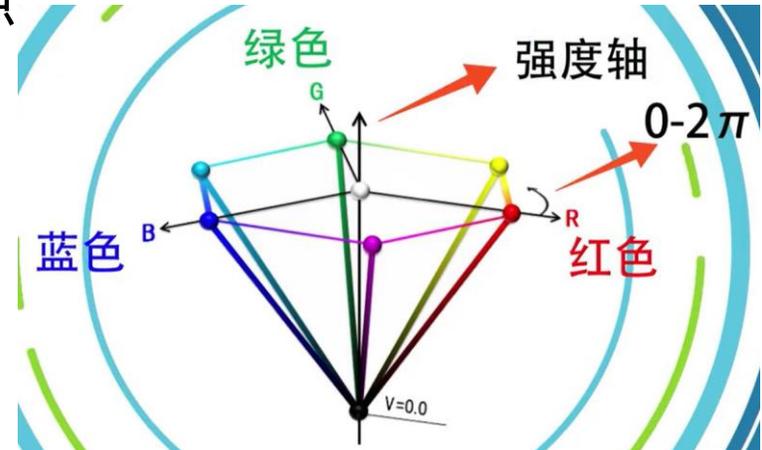
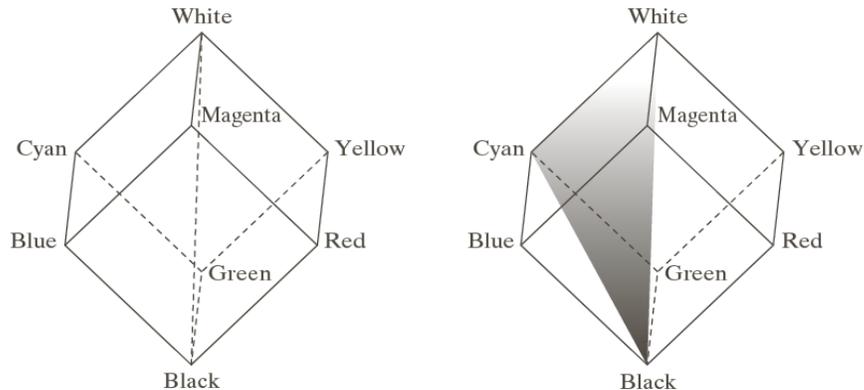
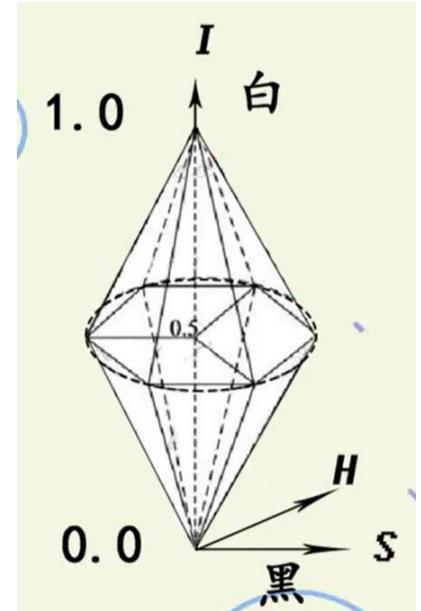
HSI彩色模型

□ HSI彩色模型

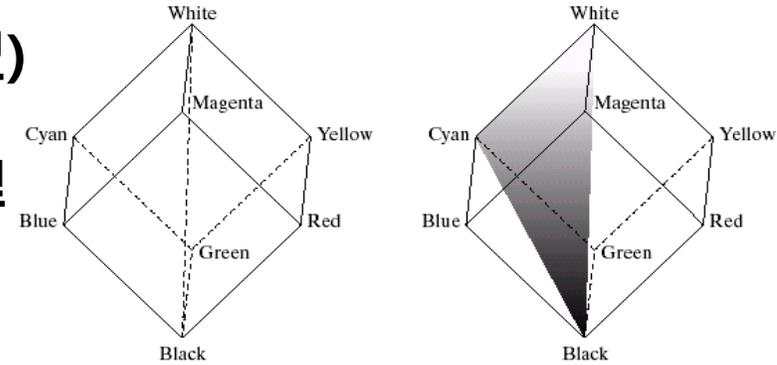
- 即色调 (hue)、饱和度 (saturation)、强度 (intensity)
- I分量与图像的彩色信息无关
- H和S分量与人感受颜色的方式紧密相连 (合成色度)

□ HSI模型与RGB模型间的概念关系

- 平面上点都有相同的色调，强度和饱和度是不同的
- 形成HSI空间所要求的**色调**、**饱和度**和**强度**值可由RGB彩色立方体得到，可将任何RGB点转换为相应的HSI彩色模型中的点



- **HSI(色调、饱和度、强度彩色模型)**
- **RGB模型对于图像彩色的产生是理想的，但对于彩色描述上的应用有较多的限制。并不适合用于彩色图像处理。**
- **由于I(Intensity)分量与图像的彩色信息无关，H(Hue)和S(Saturation)与人感觉颜色的方式紧密相连，这些特点使得HSI模型非常适合借助人的视觉系统来感知彩色特性的图像处理算法。**

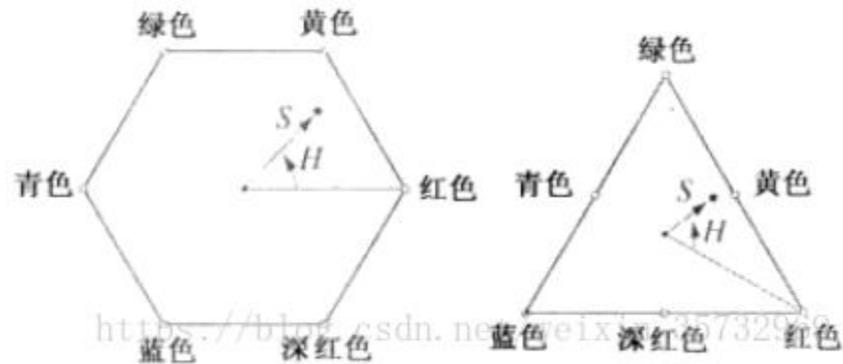
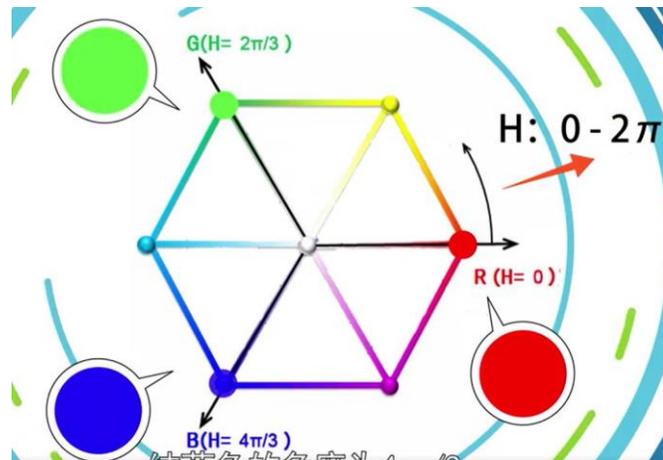


a b

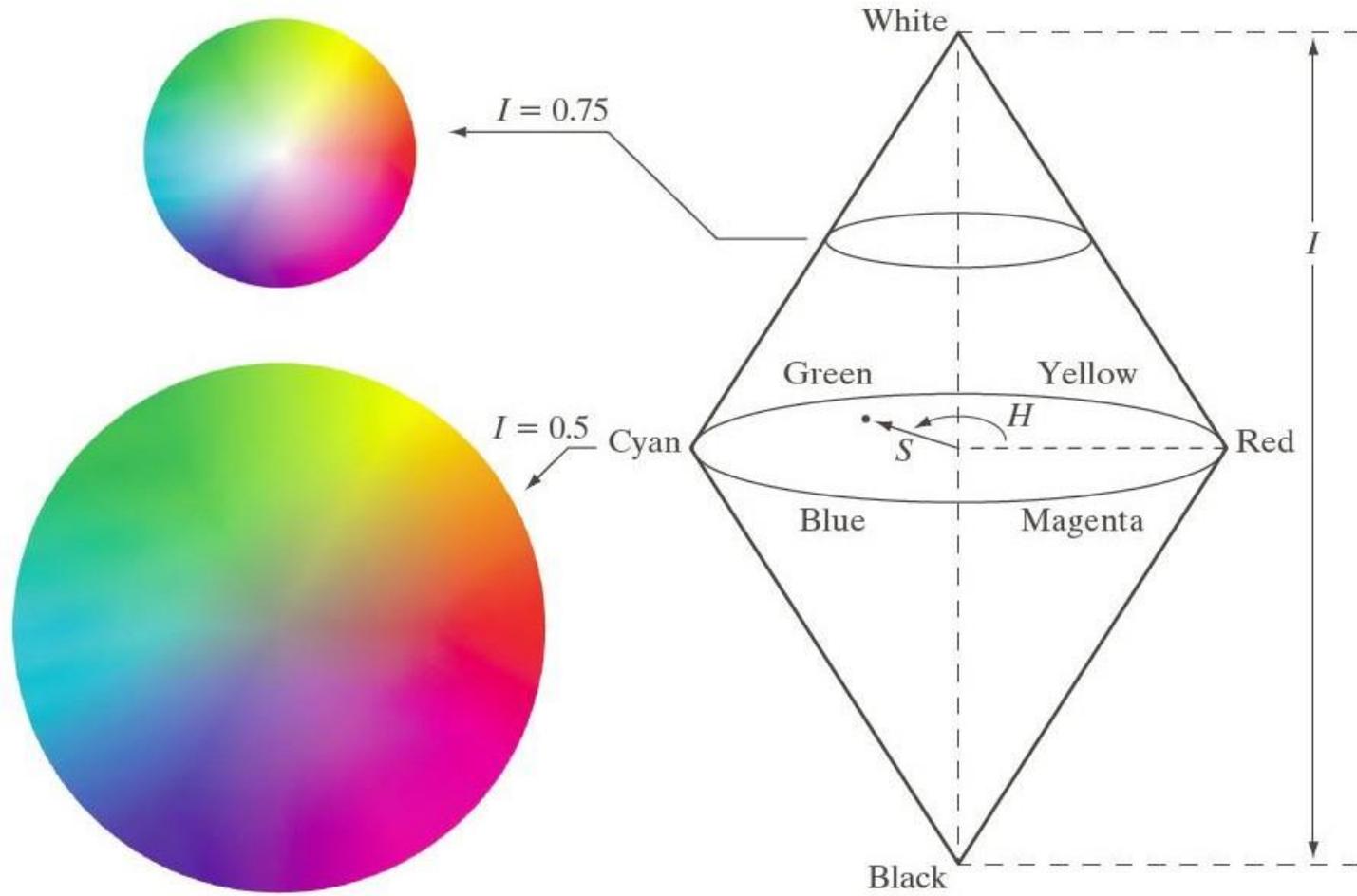
FIGURE 6.12 Conceptual relationships between the RGB and HSI color models.

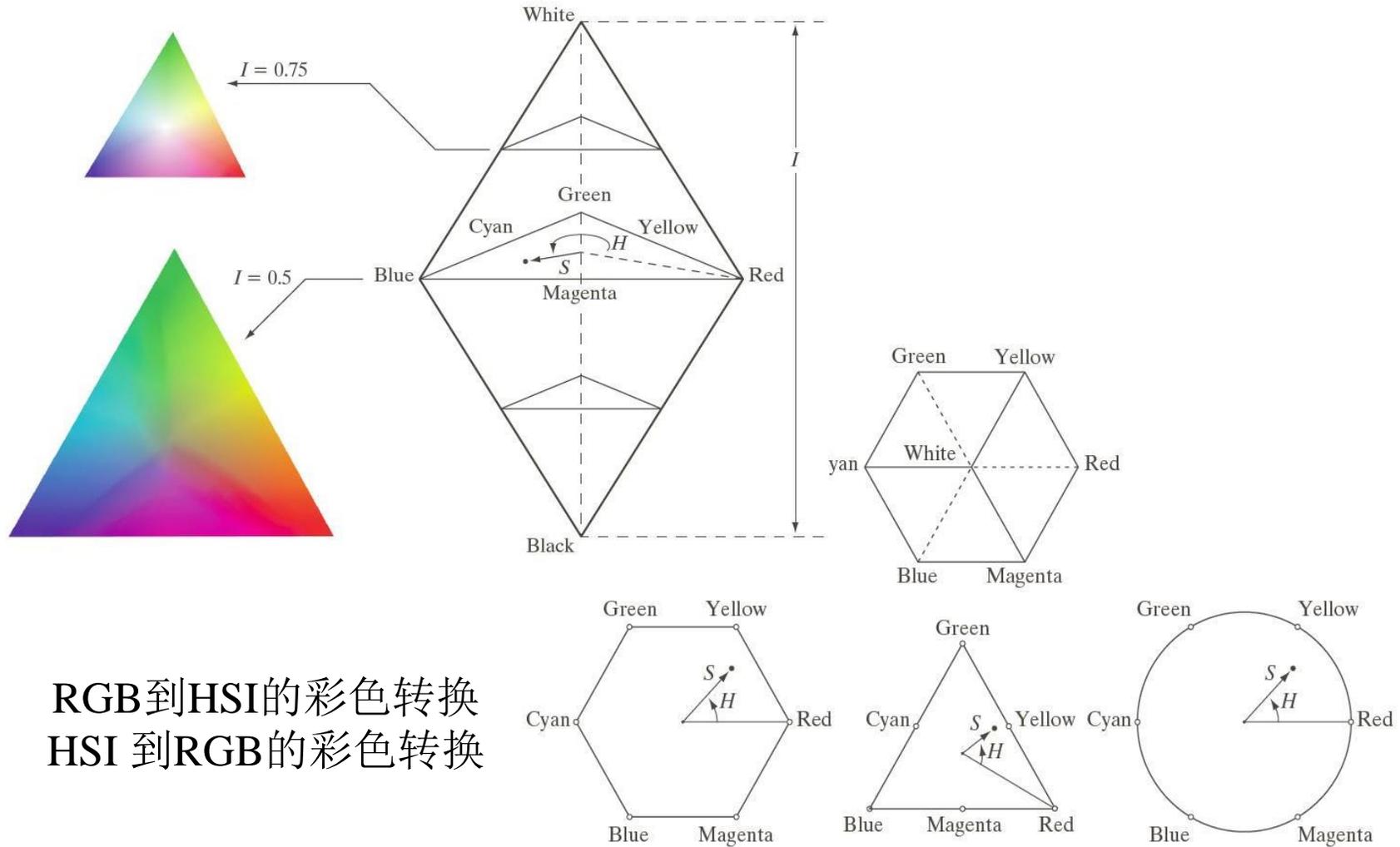
从白到黑的对角线表示强度或亮度(I)，我们可看到位于由强度轴和立方体边界决定的三角形平面内的所有点都有相同的色调。沿着垂直轴旋转色调平面，可以得到不同的色调

- HSI空间由一个强度轴和位于与该强度轴垂直的平面内的彩色点的轨迹表示
- 当平面沿强度轴上下移动时，由每个平面与立方体表面构成的横截面定义的边界不是三角形就是六边形



- 与红轴的0度角指定为0色调，从这开始色调逆时针增长
- 饱和度（距垂直轴的距离）是从原点到该点的向量长度
- HSI彩色模型的重要分量是垂直强度轴、到一个彩色点的向量长度和该向量与红轴的夹角





RGB到HSI的彩色转换
HSI到RGB的彩色转换



彩色图像处理

1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-



伪彩色图像处理

□ 特点

- 伪彩色（也称假彩色）图像处理是基于一种指定规则，对灰度值赋以颜色的处理
- 不同灰度区域 → 赋予不同颜色

□ 原理

- 人眼对颜色比对灰度有较大的分辨能力
- 对灰度：几十
- 对彩色：几千

□ 分类

- 灰度分层
- 灰度到彩色的变换

灰度分层

□ 灰度分层

- 一幅图像被描述为三维曲面
- 用一些平行于该图像的坐标平面的平面去切割图像的三维曲面，从而把灰度函数分成一些灰度区间
- 每个灰度区间赋予不同的颜色

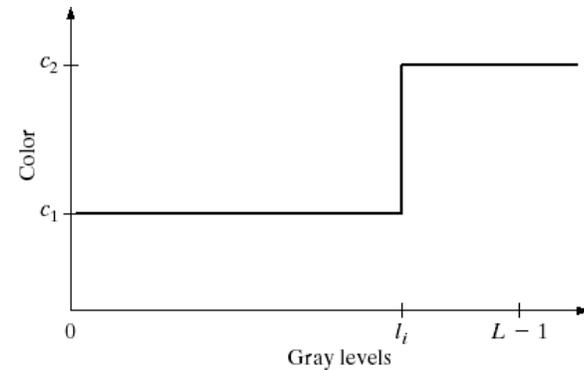
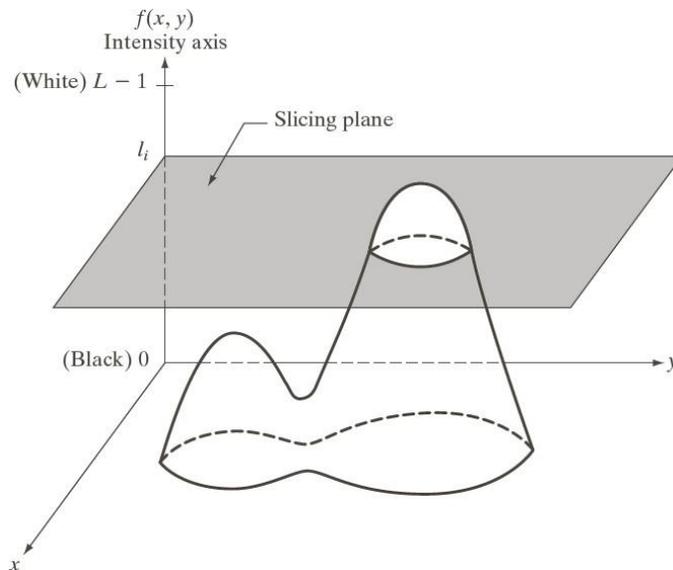


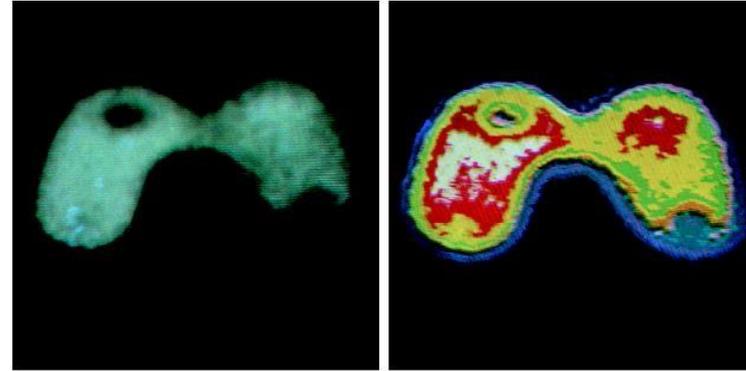
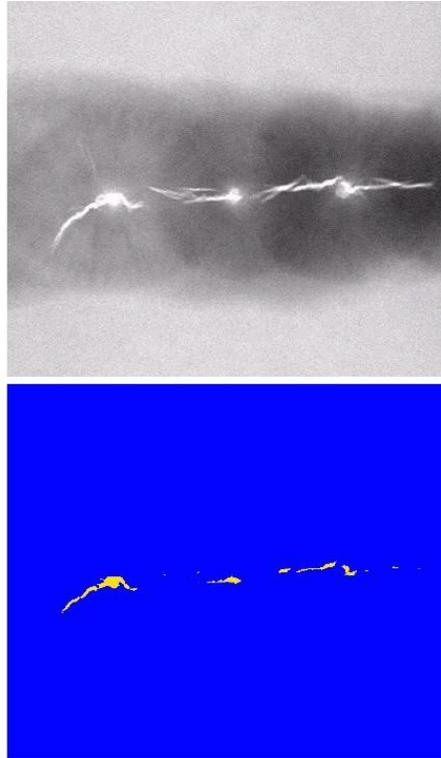
FIGURE 6.19 An alternative representation of the intensity-slicing technique.

灰度分层

□ 例

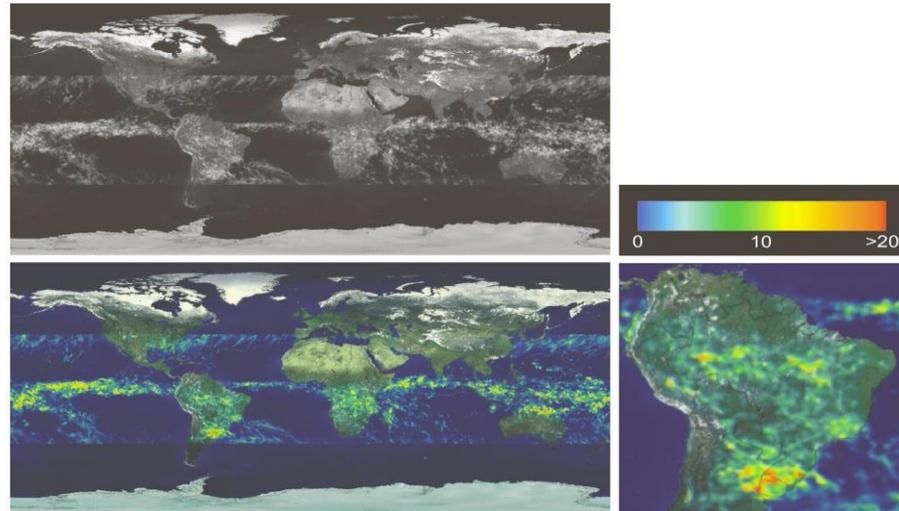
a
b

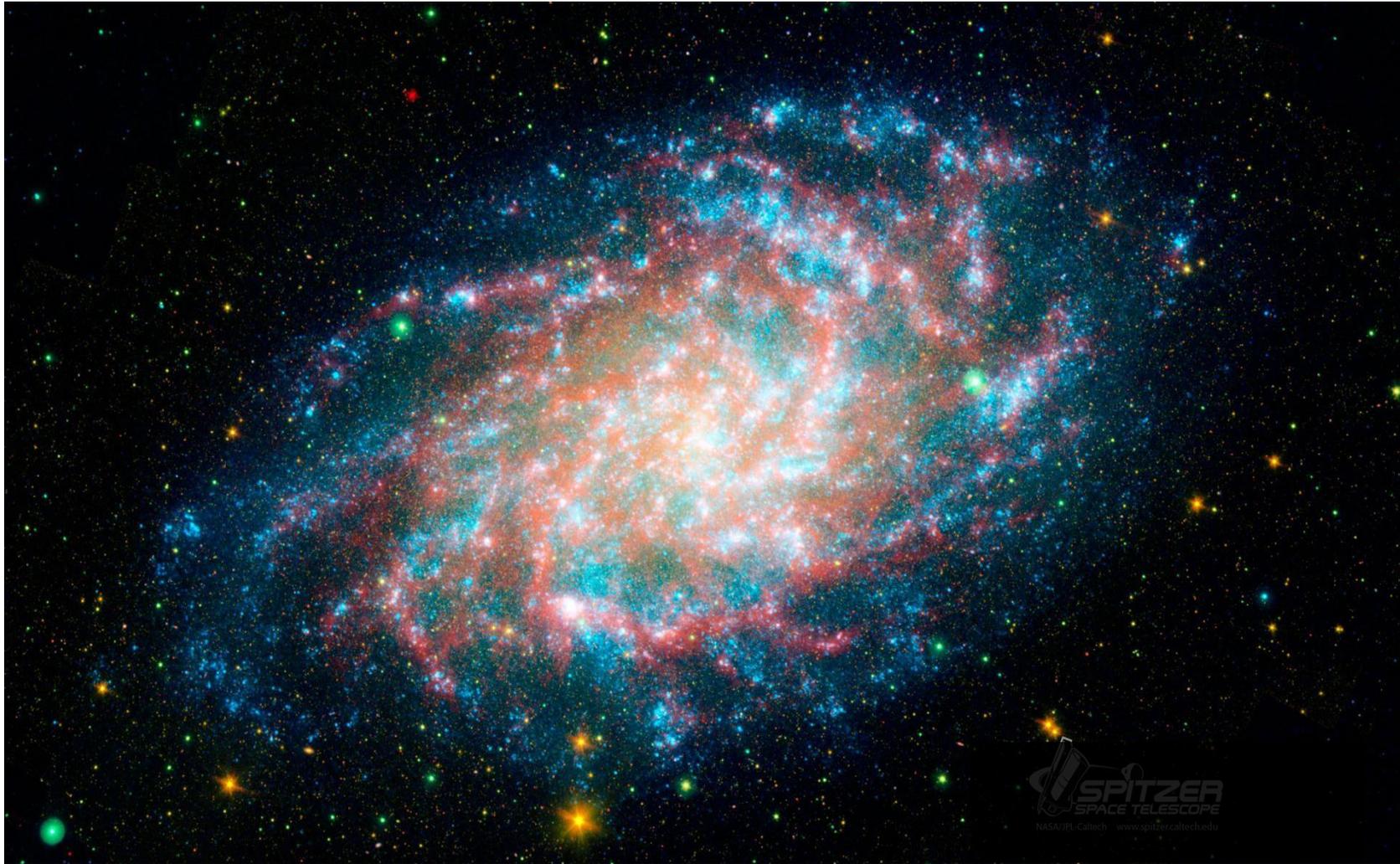
FIGURE 6.21
(a) Monochrome X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)



a b

FIGURE 6.20 (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)





灰度到彩色的变换

□ 从灰度到彩色的变换（映射）

- 对任何输入像素的灰度执行3个独立的变换
- 将3个变换结果分别送入彩色监视器的红、绿、蓝通道
- 产生一幅合成图像，图像的彩色内容有变换函数的特性调制
- 该方法是灰度分层方法的推广

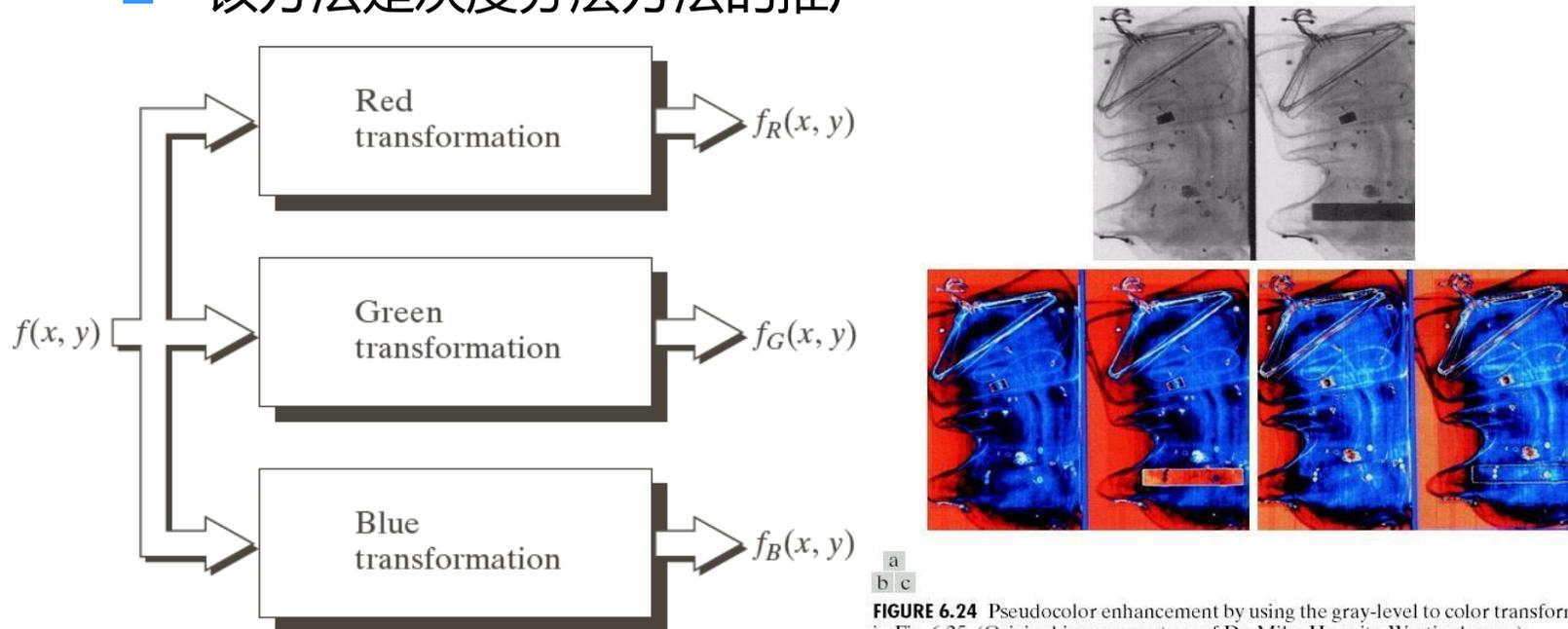


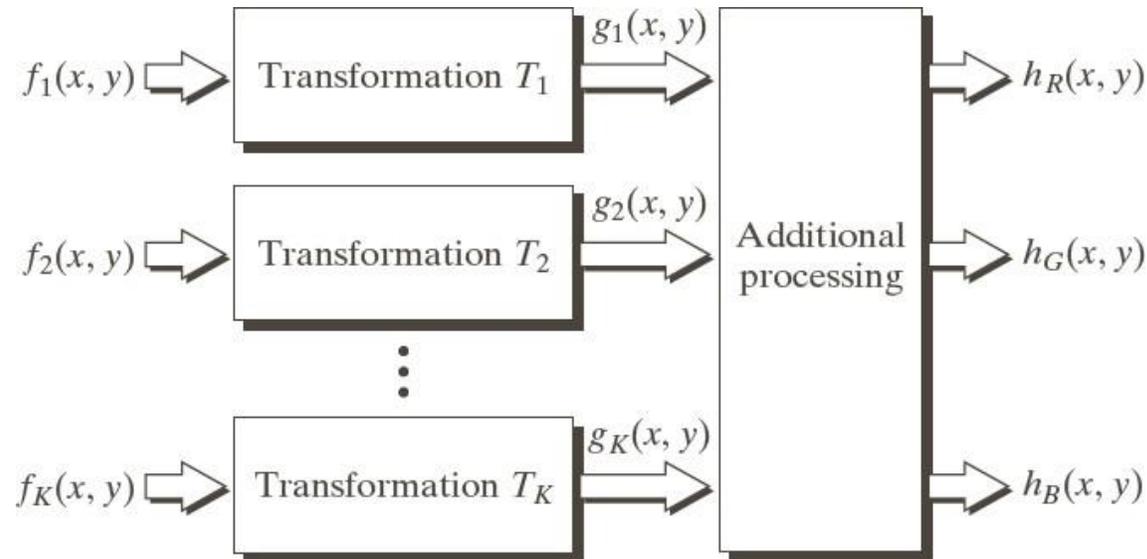
FIGURE 6.24 Pseudocolor enhancement by using the gray-level to color transformations in Fig. 6.25. (Original image courtesy of Dr. Mike Hurwitz, Westinghouse.)



灰度到彩色的变换

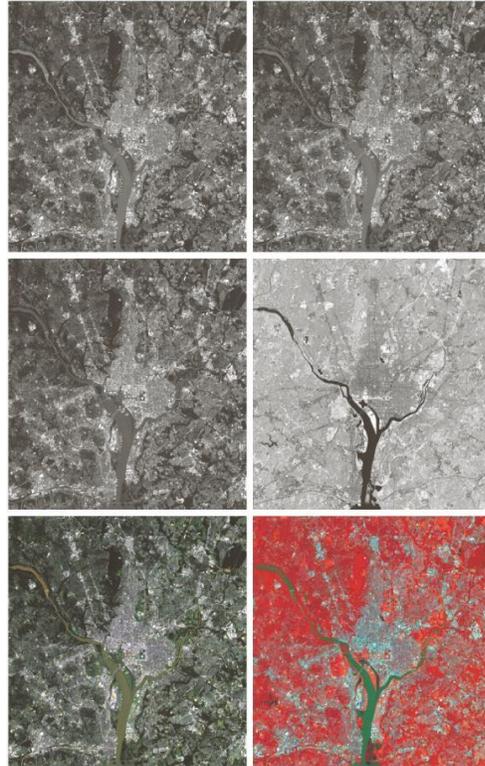
□ 多光谱图像处理

- 将多幅单色图像组合为一幅彩色图像合成图像是很重要的
- 不同的传感器在不同的谱段产生独立的单色图像
- 附加处理可以是彩色平衡、混合图像，以及以产生图像的传感器响应特性的知识为基础来选择三幅用于显示的图像等技术



灰度到彩色的变换

□ 多光谱图像的彩色编码



华盛顿的光谱卫星图像，包括波托马克河部分

a)-c)可见光的红绿蓝图像

d)近红外图像

e)前三幅合成的彩色图像

f)近红外代替红分量后图

近红外对场景中生物分量有很强的反应，所以与场景中人造目标有十分明显的差别。

FIGURE 6.27 (a)–(d) Images in bands 1–4 in Fig. 1.10 (see Table 1.1). (e) Color composite image obtained by treating (a), (b), and (c) as the red, green, blue components of an RGB image. (f) Image obtained in the same manner, but using in the red channel the near-infrared image in (d). (Original multispectral images courtesy of NASA.)

a	b
c	d
e	f

灰度到彩色的变换

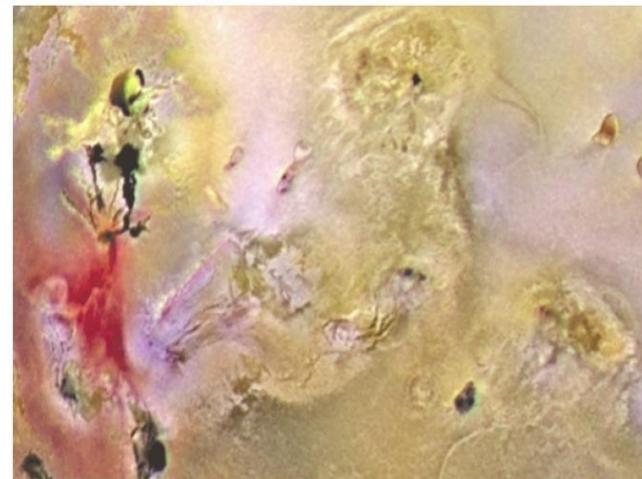
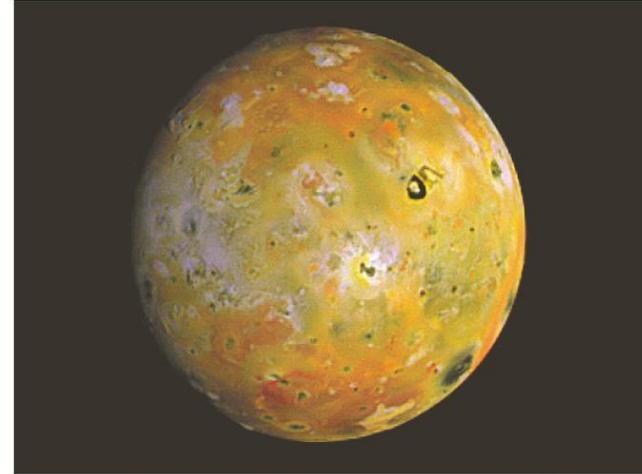
木星的伪彩色图像

亮红色为活火山喷出的物质
黄色为陈旧的硫沉积物

Galileo宇宙飞船几幅不同传感器图像合并伪彩方式显示的图像，其中一些传感器是人眼不可见的谱范围。

a
b

FIGURE 6.28
(a) Pseudocolor rendition of Jupiter Moon Io.
(b) A close-up.
(Courtesy of NASA.)





(a) Based on color strokes



(b) Based on reference color image





彩色图像处理

1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-

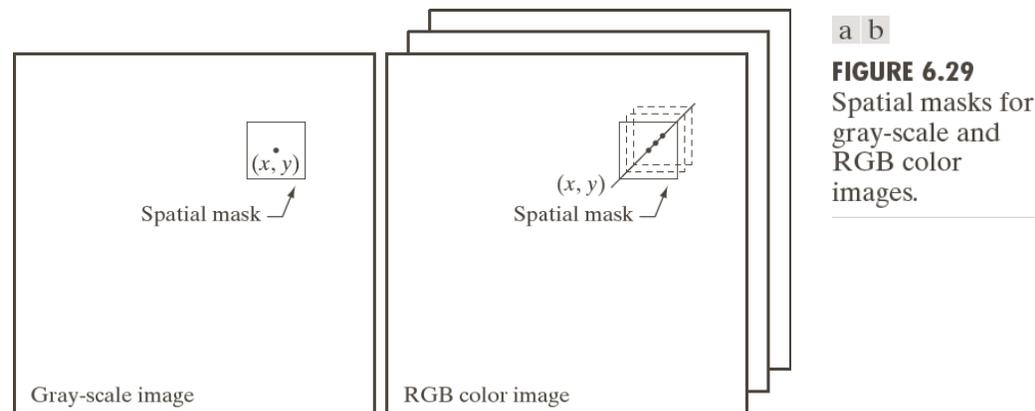
真彩色图像处理

- 真（全）彩色图像处理方法分为两大类
 - 将一幅彩色图象看作三幅分量图象的组合体，先分别单独处理，再将结果合成
 - 将一幅彩色图象中的每个象素看作具有三个属性值，即属性现在为一个矢量，利用对矢量的表达方法进行处理



真彩色图像处理

- 每种彩色分量的处理等同于基于向量的处理，两个条件
 - 第一，处理必须对向量和标量都可用；
 - 第二，对向量的每个分量的操作对于其他分量必须是独立的
 - 这两个条件满足，两种处理结果是相同的



- 图显示了灰度图像和全彩色图像的邻域空间处理，邻域平均
- (a)中平均是把邻域内的所有像素的灰度相加除以像素总数
- (b)中是把邻域内所有向量相加，并用邻域向量总数去除每个分量



彩色图像处理

1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-

彩色变换

□ 变换公式

- 如灰度变换技术那样，用下式表达的彩色变换作为模型

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

- 其中 $f(x, y)$ 是输入图像， $g(x, y)$ 是变换后处理过的彩色输出图像， T 是在 (x, y) 的空间领域上对 f 操作的一个算子
- 类似于灰度变换的方法，本节只关注如下的彩色变换

$$s_i = T_i(r_1, r_2, \dots, r_n), \quad i = 1, 2, \dots, n$$

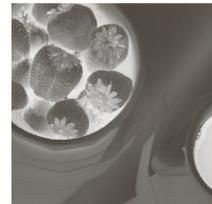
- ✓ 为标记简单起见， r_i 和 s_i 是 $f(x, y)$ 和 $g(x, y)$ 在任何点处彩色分量的变量， n 是彩色分量数。
- ✓ 注意， n 个变换合并可执行单一变换函数 T 。用于描述 f 和 g 的像素的所选彩色空间决定 n 的值。
- ✓ 例如，如果选择 RGB 彩色空间，则 $n=3$ ，且 r_1, r_2, r_3 分别表示输入图像的红、绿、蓝分量。

彩色变换

□ 全彩色图像及其各种彩色空间分量



Full color



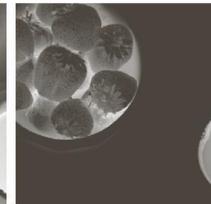
Cyan



Magenta



Yellow

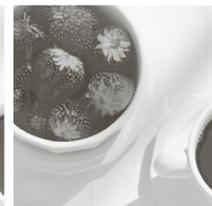


Black

图示第一行为一全彩色图像；
第二行为CMYK彩色分量图；
第三行为RGB彩色分量图；
第四行为HSI彩色分量图。



Red



Green



Blue



Hue



Saturation



Intensity

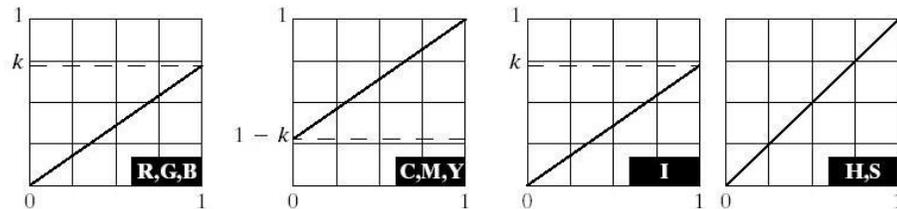
彩色变换

- 理论上，任何变换都可在任何彩色模型中执行。但实际上，某些操作对特定的模型比较适用(简洁)
 - 例如，要降低彩色图像的亮度，在RGB、CMY和HSI模型中的变换函数是不同的

a b
c d e

FIGURE 6.31

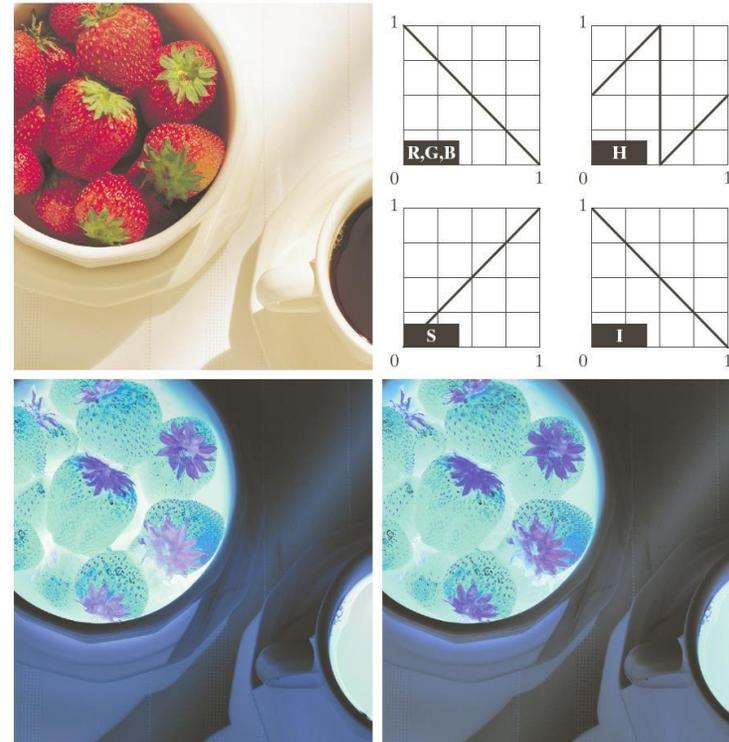
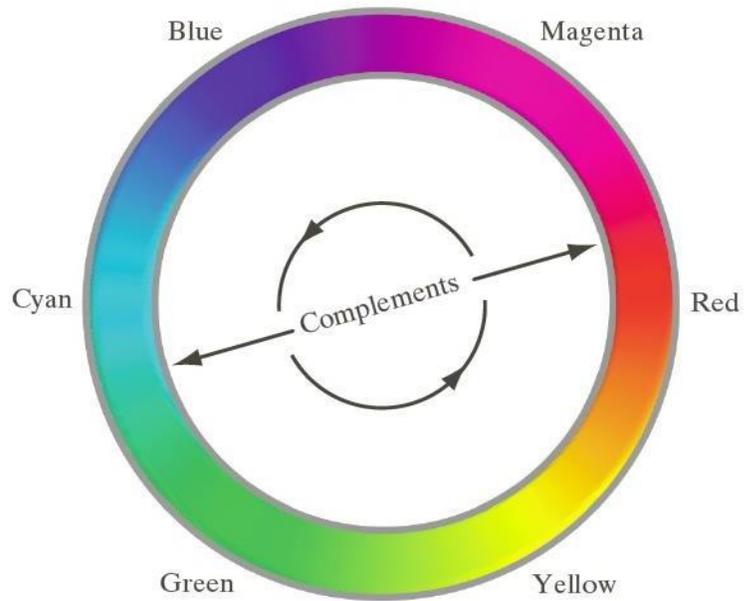
Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)



彩色变换

□ 补色

- 彩色环上与色调直接相对的另一端被称为补色
- 补色对于增强嵌在彩色图像暗区的细节很有用





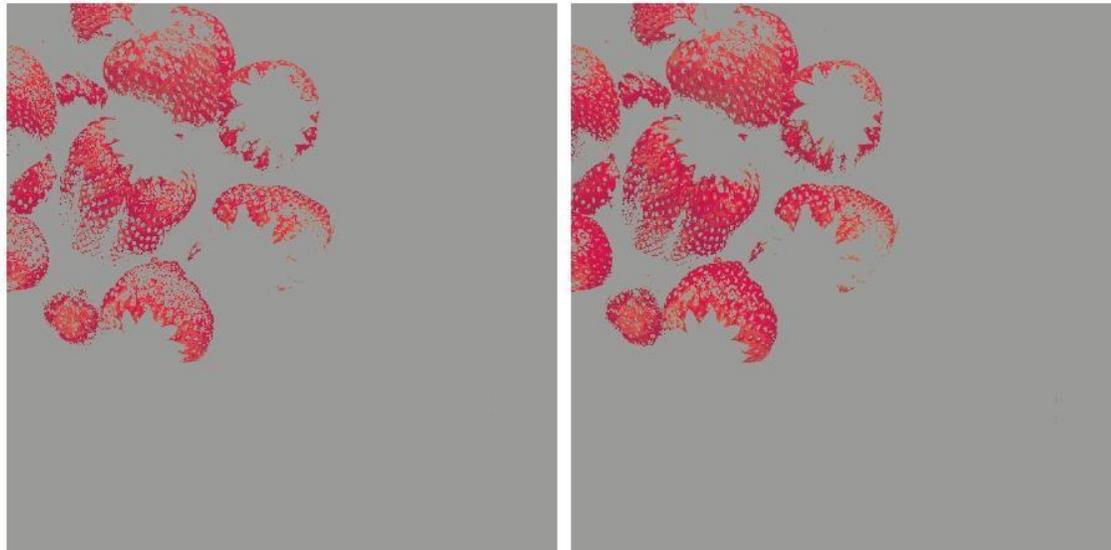
彩色变换

□ 彩色分层

- 突出图像中某个特定彩色区域，对从周围分离出目标物体很有用
- 变换函数涉及 n 维向量，一般很复杂
 - ✓ 其中较为简单的方法之一是，把某些感兴趣区域之外的彩色映射为不突出的无确定性质的颜色

$$s_i = \begin{cases} 0.5, & \left[|r_j - a_j| > \frac{W}{2} \right]_{1 \leq j \leq n} \\ r_i, & \text{其他} \end{cases}$$

$$s_i = \begin{cases} 0.5, & \sum_{j=1}^n (r_j - a_j)^2 > R_0^2 \\ r_i, & \text{其他} \end{cases}$$



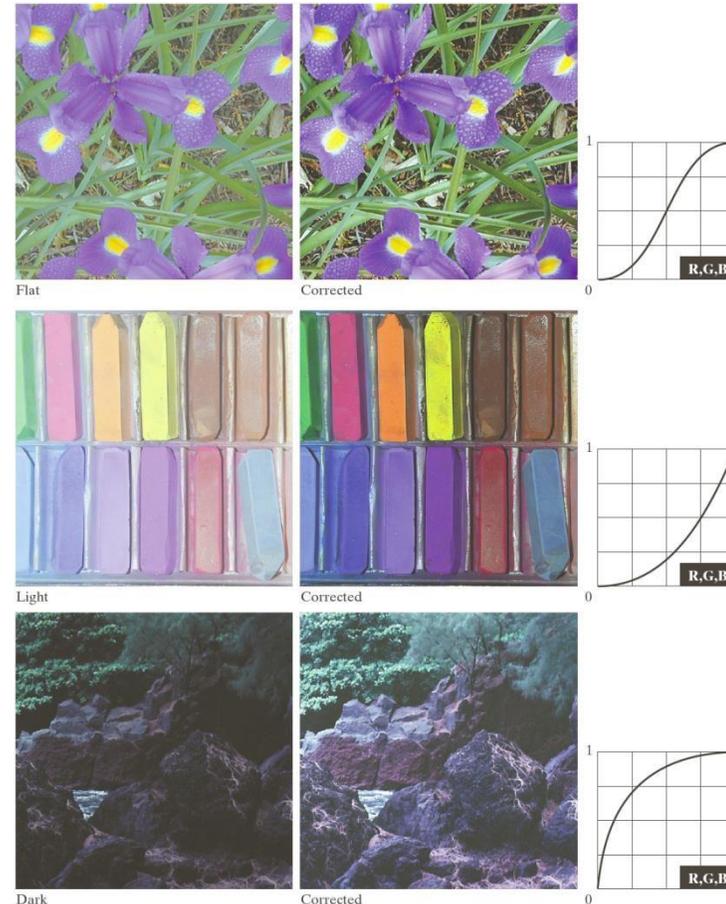
彩色变换

□ 色调和彩色校正

- 照片增强和颜色再现

□ 色调变换

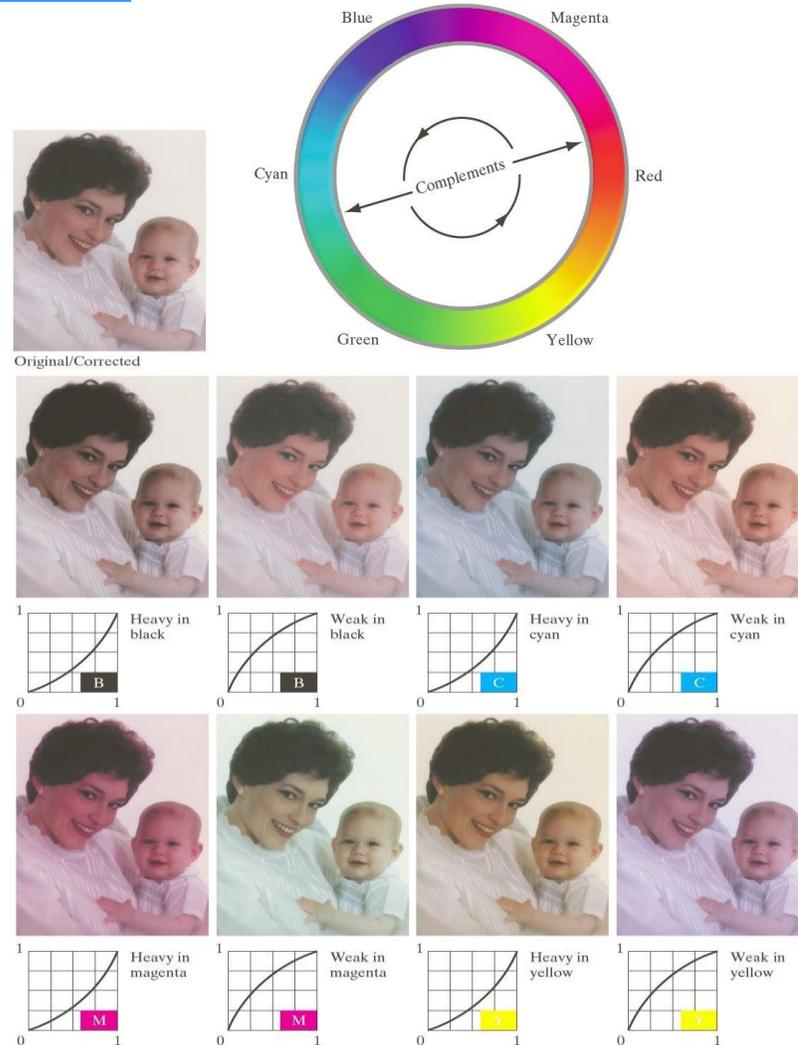
- 色调变换通常需要交互选择，**试验性调整图像亮度和对比度**，在合适的范围提供更多的细节，彩色本身不改变。
- 在RGB和CMY空间中，这意味着使用相同的变换函数映射所有的3个彩色分量
- 在HSI彩色空间中，则改进了亮度分量
- 图为校正3种常见色调不平衡的变换即平淡、较亮、较暗



彩色变换

□ 彩色平衡

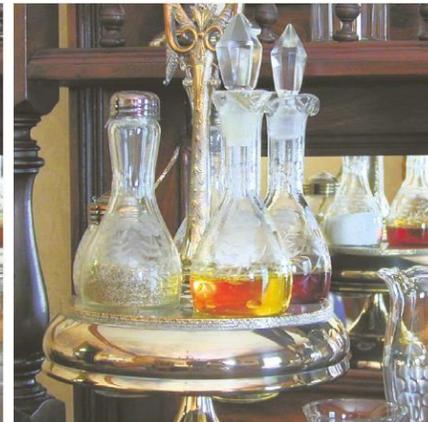
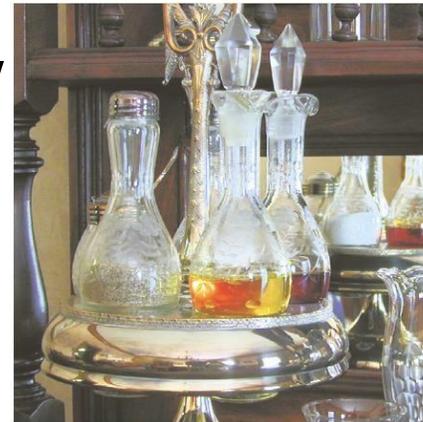
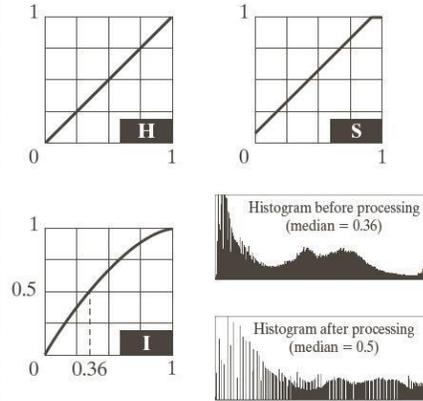
- 在调整一幅图像的彩色分量时，要意识到**每个操作都会影响到图像的全部彩色平衡**
- 彩色环可用于预测一个彩色分量如何影响其他彩色分量
- 基于彩色环，任何颜色的比例都可通过减小图像中相对色（或补色）数量来增大
- 右图显示了用于校正简单CMYK输出不平衡的变换



彩色变换

□ 直方图处理

- 直方图均衡会自动确定一种变换，产生具有均匀灰度值的直方图的图像
- 单独对彩色图像的分量进行直方图均衡是不明智的
- 应该均匀展开这种彩色灰度，而保持彩色本身（色调）不变
- 图为HSI彩色空间的直方图均衡





彩色图像处理

1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-

平滑和锐化

□ 彩色图像平滑

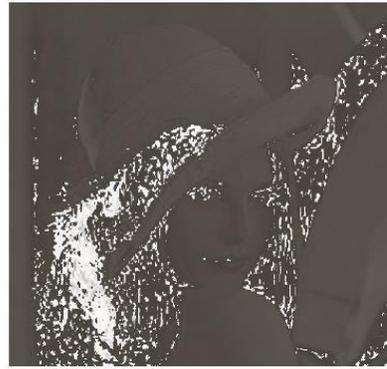
- 灰度图像平滑是一种空间滤波操作，每个像素被由模板定义的邻域中的像素的平均值代替。可推广到彩色图像处理上
- 邻域平均平滑可以在每个彩色平面的基础上执行，其结果与使用RGB彩色向量执行平均是相同的
- 下图为一副图像的RGB分量图像



平滑和锐化

- 第一行为该图的HSI分量图像，下图为平滑这2种彩色模型的区别
 - 仅平滑I分量图像，不会改变图像的颜色

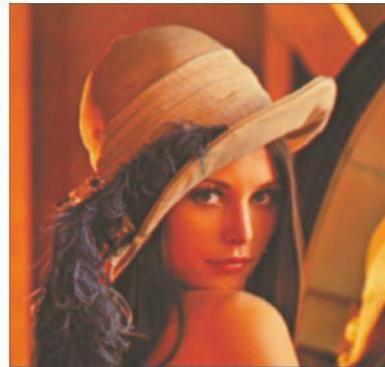
色调



饱和度



亮度

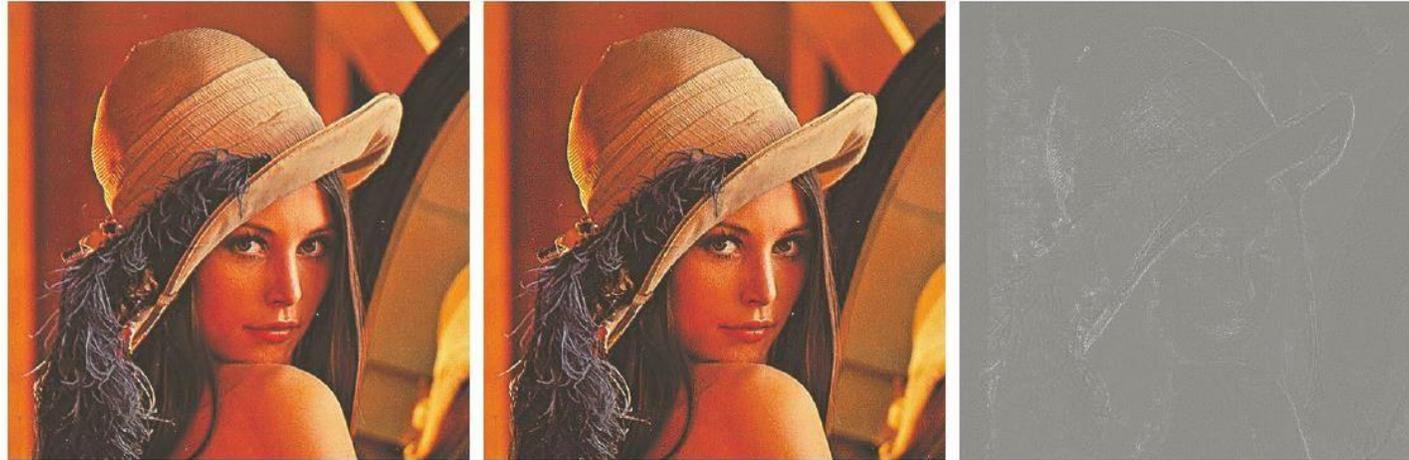


用5×5模板平滑图像：处理每个RGB分量结果（左）；处理HSI的I分量（中）；差别（右）

平滑和锐化

□ 彩色图像锐化

- 采用拉普拉斯变换方法的图像锐化处理
- 可通过分别计算每幅分量图像的拉普拉斯变换来计算彩色图像的拉普拉斯变换



拉普拉斯锐化图像：处理每个RGB分量结果（左）；处理HIS的I分量（中）；差别（右）



彩色图像处理

1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-



彩色图像中的噪声

- 之前讨论的噪声模型可用于彩色图像
- 通常，彩色图像的噪声内容在每个彩色通道中具有相同的特性，但噪声对不同彩色通道所造成的影响不同
- 不同的噪声水平像是由每个彩色通道的相对照射强度的差异造成的
 - 例如，在CCD摄像机中，红色滤镜的使用将减少用于红色传感器的照射强度。CCD传感器在低照明水平下就是噪声源，因此，这种情况下得到的RGB图像红色分量图像与其他两幅分量图像相比往往是噪声源

彩色图像中的噪声

- 与在单色图像中相比，细粒噪声在彩色图像中不太引人注意



(a)~(c)由均值为0、方差为800的高斯噪声污染的红、绿和蓝分量图像；
(d)最终的RGB图像

彩色图像中的噪声

- 噪声从一个彩色模型转换为另一个模型
 - 噪声图像的色调与饱和度分量明显降质了，这分别是由转换函数中求余弦与取最小值操作的非线性造成的
 - 图c中的强度分量比3个带噪声的RGB分量图像中的任何一个都要平滑一些，这是由亮度图像是RGB图像的平均这一事实造成的



上图(d)中带噪声的彩色图像的HSI分量：(a)色调图像；(b)饱和度图像；(c)亮度图像

彩色图像中的噪声

- 在仅有一个RGB通道受噪声影响的情况下，到HSI的转换才将噪声扩散到所有HSI分量图像



计算HSI分量
要用到RGB
所有的分量

- (a) 绿平面被椒盐噪声污染的RGB图像；
- (b) HSI图像的色调分量；
- (c) 饱和度分量；
- (d) 亮度分量



彩色图像处理

1. 彩色基础
 2. 彩色模型
 3. 伪彩色图像处理
 4. 真彩色图像处理
 5. 彩色变换
 6. 平滑和锐化
 7. 彩色图像中的噪声
 8. 彩色图像压缩
-



彩色图像压缩

- 因为描述彩色所要求的比特数比描述灰度级所要求的比特数大3~4倍，所以数据压缩在存储和传输彩色图像中起着核心的作用。
- 对于前几节描述的RGB, CMY(K)和HSI图像，任何压缩对象的数据是每个彩色像素的分量，它们是传达彩色信息的方式。
- 压缩是减少或消除冗余和不相干数据的处理。

彩色图像压缩

□ 实例

- 一幅24比特RGB全彩色图像
- 采用JPEG 2000压缩算法
- 原图每230比特数据，压缩后仅包含1个数据比特
- 假定压缩后图像在Internet上能于1分钟内传输完成，那么原图像的传输至少需要4小时







数字图像处理 (Digital Image Processing)

王正

Email: zheng_wang@seu.edu.cn
办公室: 无线谷A5415

**Image
Processing
Project Ideas**
Beginners



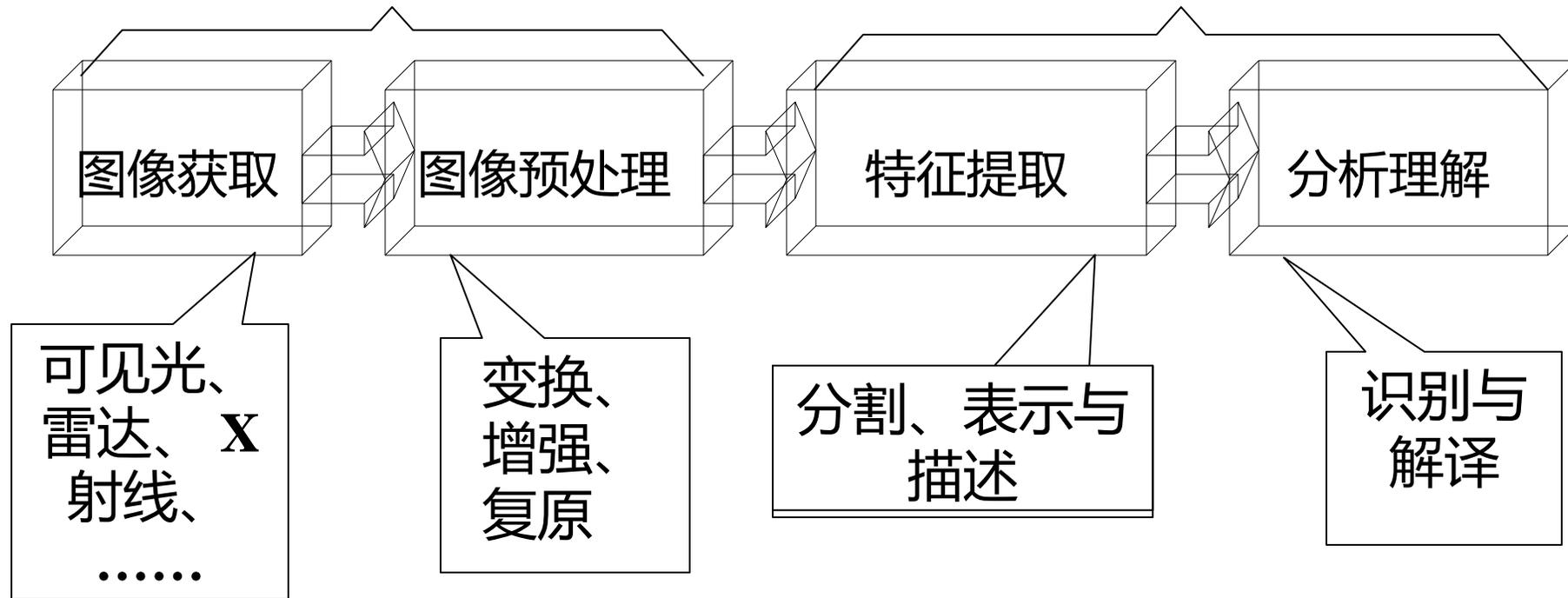
图像分割

- 图像分割：将图像细分为构成它的子区域或物体。
 - 细分的程度取决于要解决的问题。
 - 在应用中，当感兴趣的物体或区域已经被检测出来时，就停止分割
 - 异常图像的分割是图像处理中最困难的任务之一
- ☞ 把图像分割成不同的区域； ← 图像分割
- ☞ 找出分开的各区域的特征； ← 特征提取
- ☞ 对于不同区域进行描述；或相关区域连接起来组成有意义的结构； ← 分析、描述和解释
- ☞ 识别图像中要找的目标，或对图像中不同的特征进行分类； ← 识别、分类

■ 数字图像处理与分析基本技术框架

经典数字图像处理

数字图像分析



- 本章中的多数分割算法均基于灰度值的两个基本性质之一：**不连续性**和**相似性**。
 - 对于不连续性的灰度，方法是以灰度突变为基础分割一幅图像，比如图像的边缘。
 - 对于相似的灰度，主要方法是根据一组预定义的准则把一幅图像分割为相似的区域。阈值处理、区域生长和区域聚合都是这类方法的例子



技术

- ◇ **基于点相关**：依据各个像素灰度的不连续性进行分割
 - **门限方法** - 选择合适的门限将目标从背景中分割出来
 - **边缘检测** - 先确定目标边缘轮廓像素再把它连接在一起以构成所需的边界
 - **跟踪方法** - 利用前期处理过的点的信息，判断图像当前处理点是否为目标点，再进行跟踪运算
- ◇ **基于区域相关**：依据同一区域具有相似的灰度特征或组织特征，寻求不同区域的边界



知乎 @stone



知乎 @stone

The interface includes a central canvas with a landscape painting of a waterfall. On the left, there are icons for eraser, brush, and pencil. On the right, there is a dice icon and a vertical stack of image thumbnails. At the bottom, there is a color palette with the following labels:

sky	tree	cloud	mountain	grass	sea	river	rock	plant	sand
snow	water	hill	dirt	road	flower	stone	bush	wood	gravel

- 图像分割的基本思路
 - 从简到难，逐级分割
 - 控制背景环境，降低分割难度
 - 把焦点放在增强感兴趣对象，缩小不相干图像成分的干扰上





图像分割

1. 基础知识
 2. 点、线和边缘检测
 3. 阈值处理
 4. 基于区域的分割
-



基础知识

- 令集合 R 代表整个图像区域，对 R 的分割可看做将 R 分成若干个满足下述条件的非空的子集（子区域）

R_1, R_2, \dots, R_n :

- 1、 $\bigcup_{i=1}^n R_i = R$;
- 2、对 $i \neq j$ ，有 $R_i \cap R_j = \emptyset$;
- 3、每个子区域 R_1, R_2, \dots, R_n 是连通的;
- 4、对于各个子区域，有均匀性测度度量 P 为真

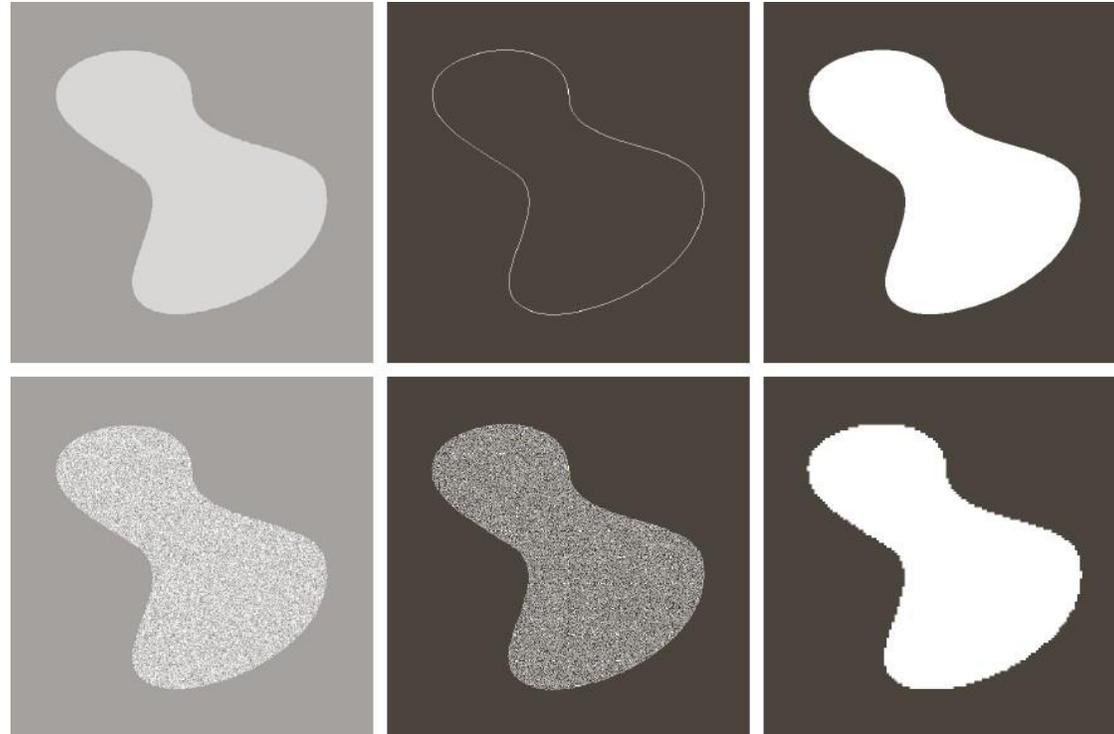
$$P(R_i) = \text{TRUE};$$

- 5、对其中任意两个和两个以上相邻子区域之并，其均匀性测度度量 P 为假，即：

$$P(R_i \cup R_j) = \text{FALSE}, (i \neq j)$$

基础知识

□ 下面三幅图像说明了基于区域的分割



a	b	c
d	e	f

FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.



图像分割

1. 基础知识
2. 点、线和边缘检测
3. 阈值处理
4. 基于区域的分割

点、线和边缘检测

- 本节介绍以灰度局部剧烈变化的检测为基础的分割方法。
- 我们感兴趣的三种图像特征是孤立点、线和边缘
- 就像局部平均平滑一幅图像那样，假设平均处理类似于积分，对于灰度的突变，局部变化可以用微分来检测。
- 由于变化非常短促，因此一阶微分和二阶微分特别适合
- 数字函数的导数可用差分定义
 - 一维函数 $f(x)$ 在点 x 处的导数的近似：数字差分

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

- 关于点 x 处的二阶导数的近似：

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

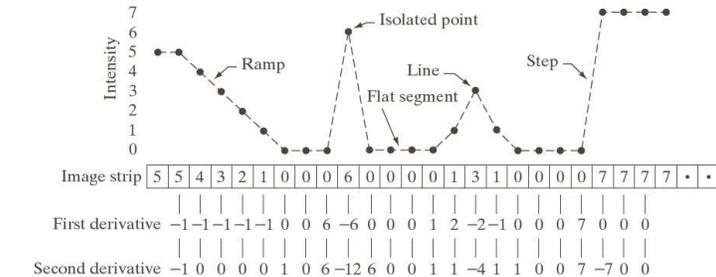
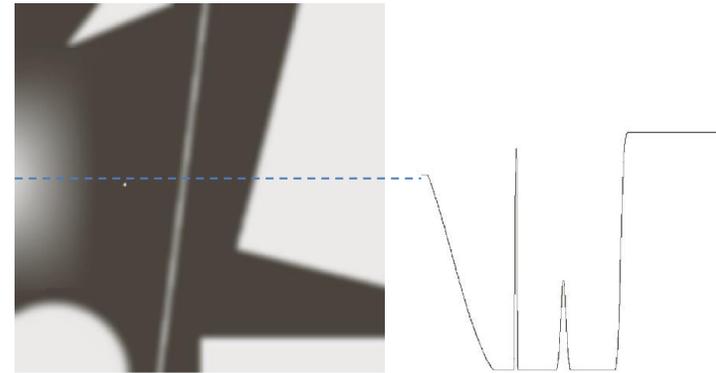
- 计算图像中每个像素位置处的一阶导数和二阶导数的另一种方法是使用空间滤波器：

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

FIGURE 10.3
A general 3×3
spatial filter mask.

点、线和边缘检测

- 基于一阶导数和二阶导数检测图像包含有不同的实心物体、一条线和单个噪声点（右图）
- 结论：
 - (1) 一阶导数通常在图像中产生较粗的边缘；
 - (2) 二阶导数对精细细节，如细线、孤立点和噪声有较强的响应；
 - (3) 二阶导数在灰度斜坡和灰度台阶过渡处会产生双边缘响应；
 - (4) 二阶导数的符号可用于确定边缘的过渡是从亮到暗还是从暗到亮



a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

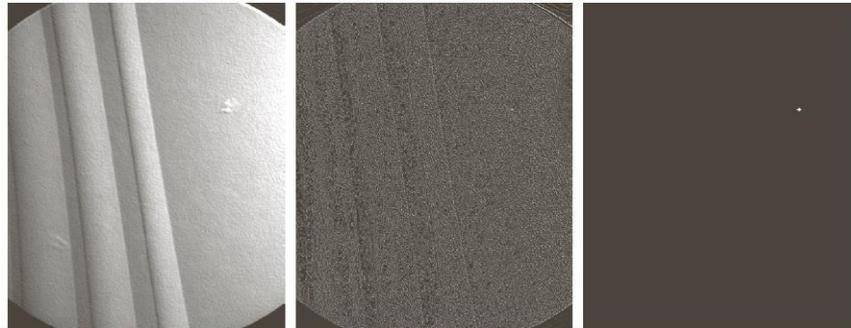
孤立点的检测

- 点的检测应以二阶导数为基础，即使用拉普拉斯算子：

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

- 同样，该算子可用模板来实现
- 如果在某个点处，该模板的响应的绝对值超过了指定的阈值，则在 (x, y) 处的点已被检测到。在输出图像中，该点标为1，否则为0

1	1	1
1	-8	1
1	1	1



a
b c d

FIGURE 10.4

(a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

线检测

- 点检测的拉普拉斯算子是各向同性的，因此其响应与方向无关（相对于该 3×3 拉普拉斯目标的4个方向：垂直方向、水平方向和两个对角方向）。
- 通常，我们的兴趣在于检测特定方向的线。
 - 第一个模板对于水平线有最佳响应；
 - 第二个模板对于45度方向的线有最佳响应；
 - 第三个模板对于垂直线有最佳响应；
 - 第四个模板对于-45度方向的线有最佳响应

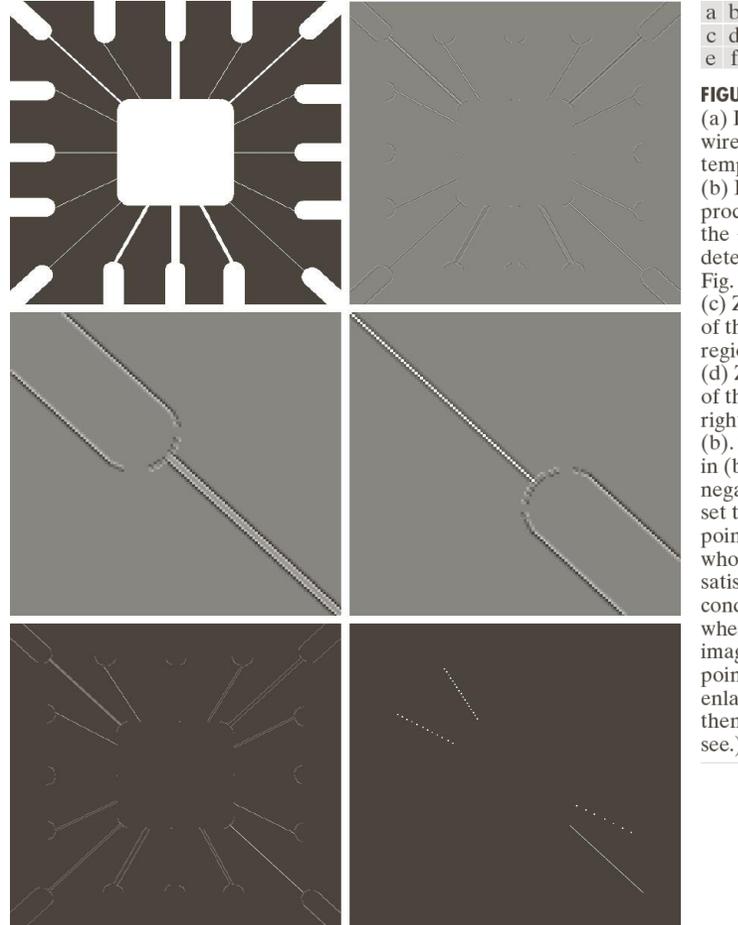
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
Horizontal			+45°			Vertical			-45°		

FIGURE 10.6 Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

- 每个模板的系数之和为0，这表明恒定灰度区域中的响应为0

线检测

□ 特定方向线的检测

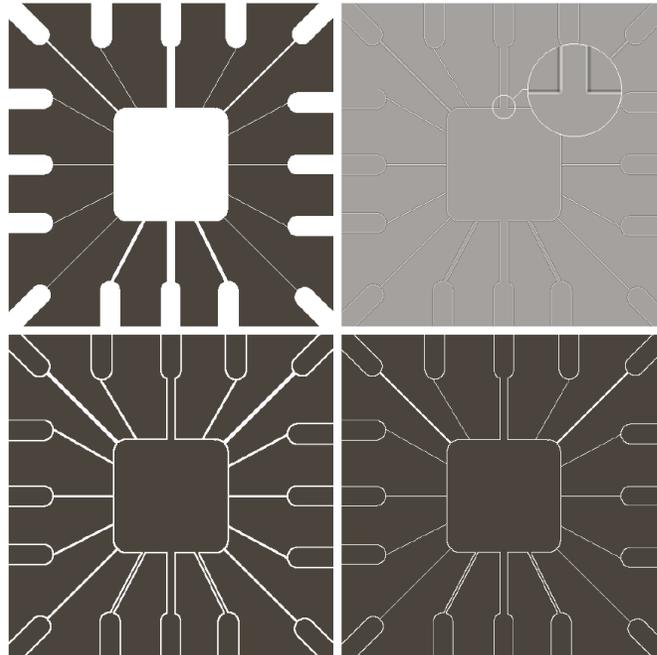


a b
c d
e f

FIGURE 10.7
 (a) Image of a wire-bond template.
 (b) Result of processing with the $+45^\circ$ line detector mask in Fig. 10.6.
 (c) Zoomed view of the top left region of (b).
 (d) Zoomed view of the bottom right region of (b).
 (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

线检测

- 对于线检测，二阶导数将导致更强的响应，产生比一阶导数更细的线。我们可以使用拉普拉斯模板，但要处理双线效应
- 例：



a b
c d

FIGURE 10.5

(a) Original image.

(b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian.

(c) Absolute value of the Laplacian.

(d) Positive values of the Laplacian.

边缘模型

- 边缘检测是基于灰度突变来分割图像的常用方法
- 边缘模型根据它们的灰度剖面来分类。有台阶边缘、斜坡边缘和“屋顶”边缘等
- 下图显示了这些边缘模型和相应的灰度剖面

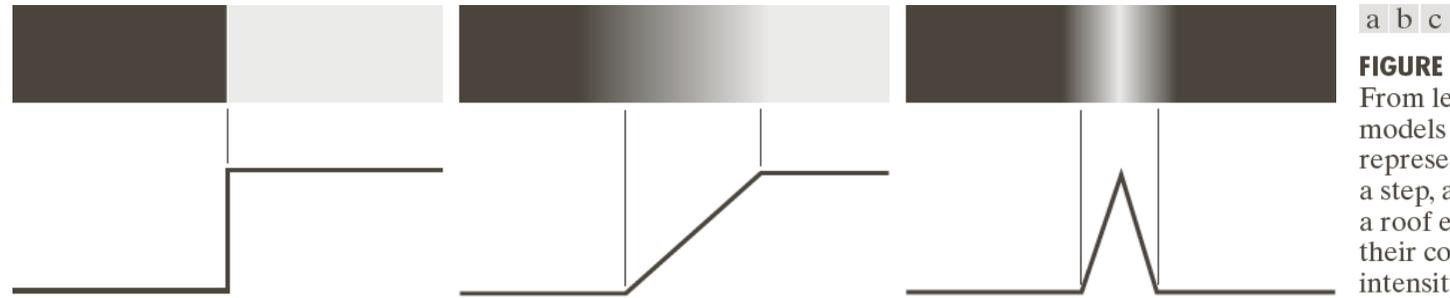
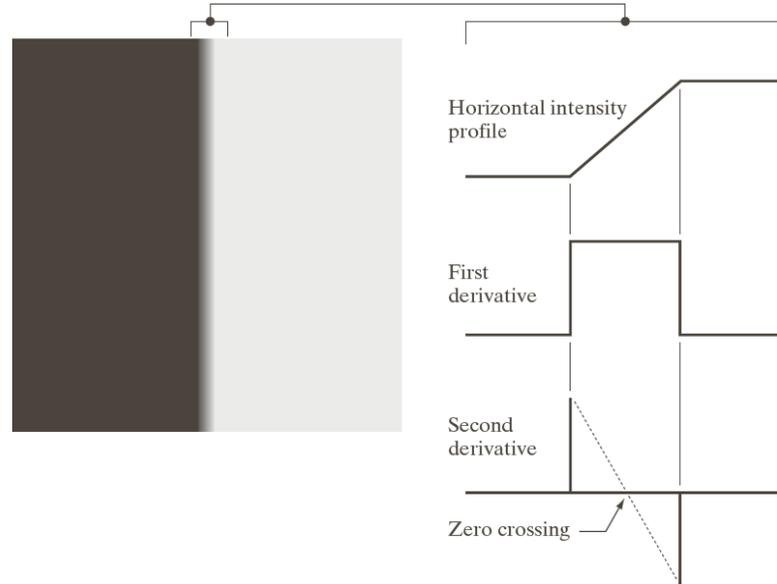


FIGURE 10.8
From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

边缘模型



a b

FIGURE 10.10
(a) Two regions of constant intensity separated by an ideal vertical ramp edge.
(b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

□ 上图结论

- 一阶导数的幅度可用于检测图像中的某个点处是否存在一个边缘
- 二阶导数的符号用于确定一个边缘像素是位于该边缘的暗侧还是位于该边缘的亮侧

□ 围绕一条边缘的二阶导数的两个附加性质：

- 对图像中的每条边缘，二阶导数生成两个值（一个不希望的特点）
- 二阶导数的零交叉点可用于定位粗边缘的中心

边缘模型

- 微弱的可见噪声严重影响检测边缘所用的一阶导数和二阶导数
- 因此执行边缘检测的三个基本步骤是：
 - 为降噪图像进行平滑处理
 - 边缘点的检测。这是一个局部操作，从一幅图像中提取所有边缘点的潜在候选者
 - 边缘定位。这一步的目的是从候选边缘点中选择组成边缘点集合的真实成员

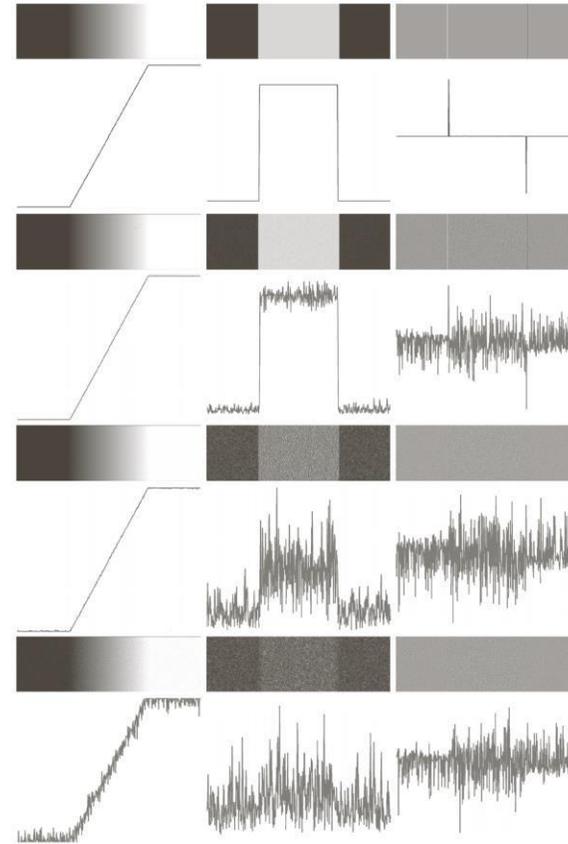


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

基本边缘检测

□ 图像梯度及其性质

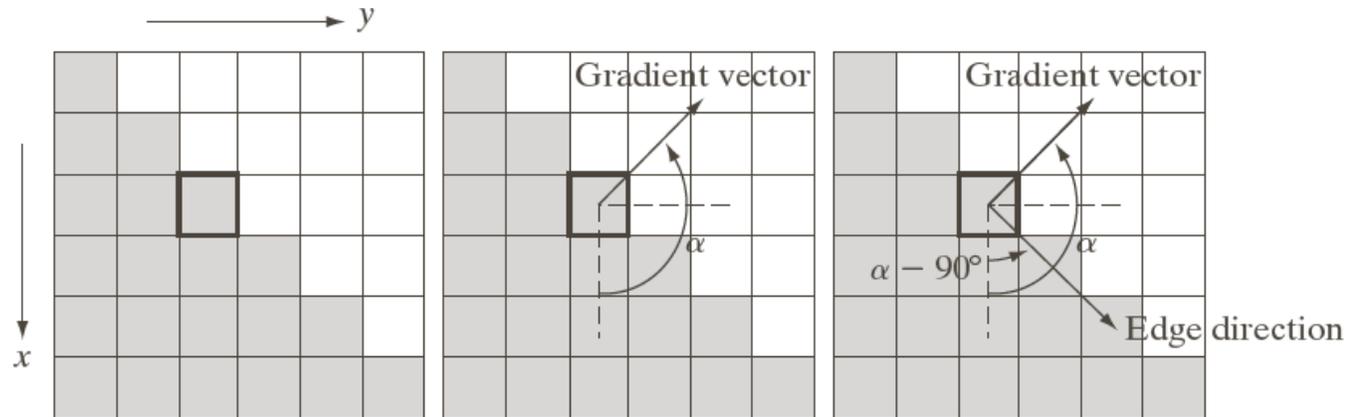
- 梯度向量、幅度和方向角

$$\nabla f(x, y) = [G_x \ G_y]^T = \left[\frac{\partial f}{\partial x} \ \frac{\partial f}{\partial y} \right]^T$$

$$\text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2}$$

$$\phi(x, y) = \arctan(G_y / G_x)$$

- 图像边缘方向与边缘上点的梯度向量正交



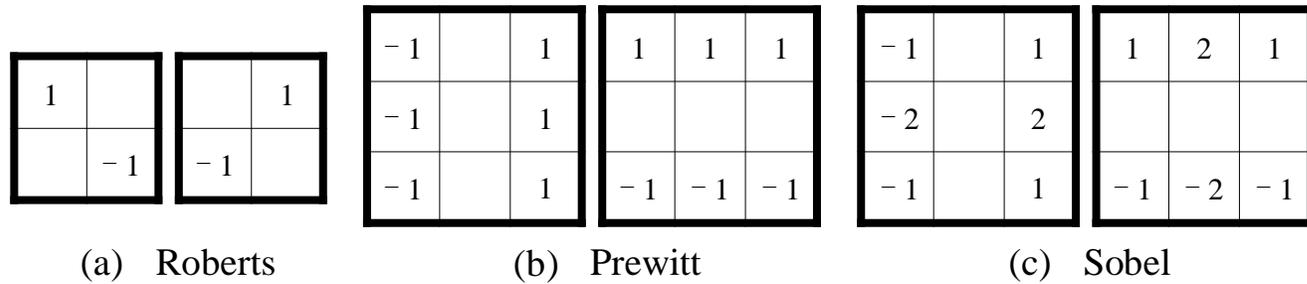
a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

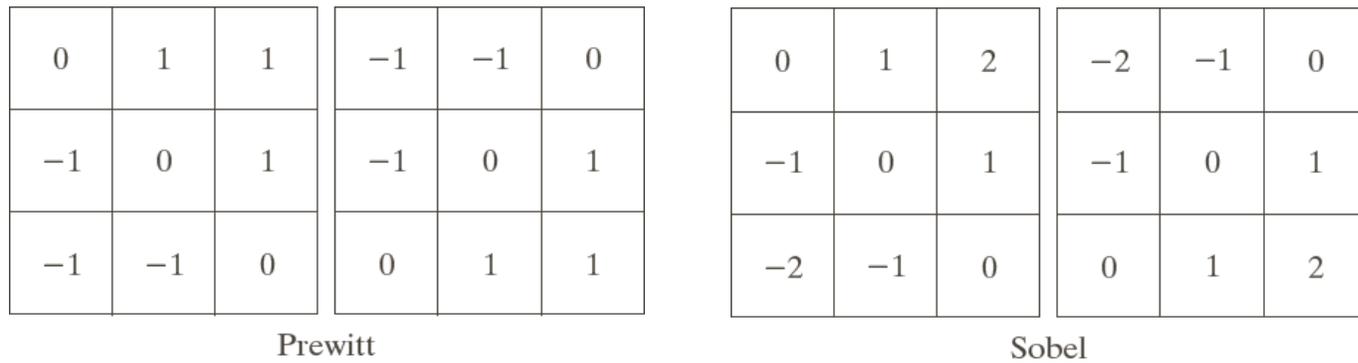
基本边缘检测

□ 梯度算子

- 导数用模板对图像进行滤波来实现



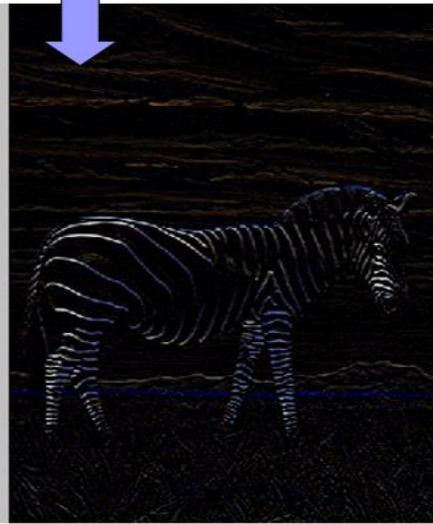
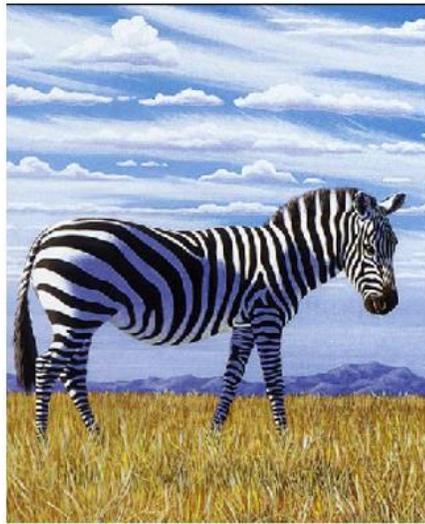
- 检测对角边缘的模板



Prewitt 算子

$$\frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

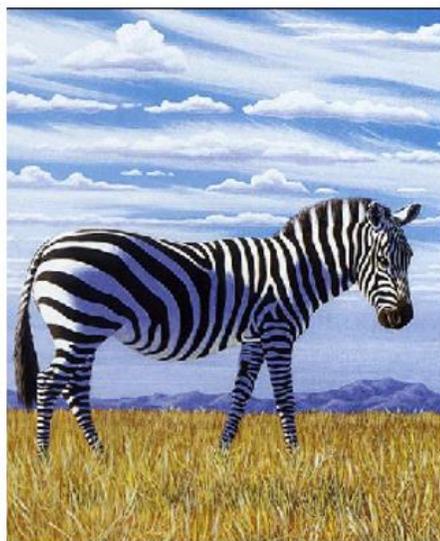


Sobel算子

能够更好地抑制噪声

$$\frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



Sobel梯度算子是先做加权平均，然后再微分

基本边缘检测

□ 梯度算子



a b
c d

FIGURE 10.16

(a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$.
(b) $|g_x|$, the component of the gradient in the x -direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image.
(c) $|g_y|$, obtained using the mask in Fig. 10.14(g).
(d) The gradient image, $|g_x| + |g_y|$.

基本边缘检测

□ 梯度算子

- 精细的细节在边缘检测中通常是不符合需要的，因为它往往表现为噪声，使主要边缘的检测变得复杂。
- 减少精细细节的一个方法是对图像进行平滑处理，如下图



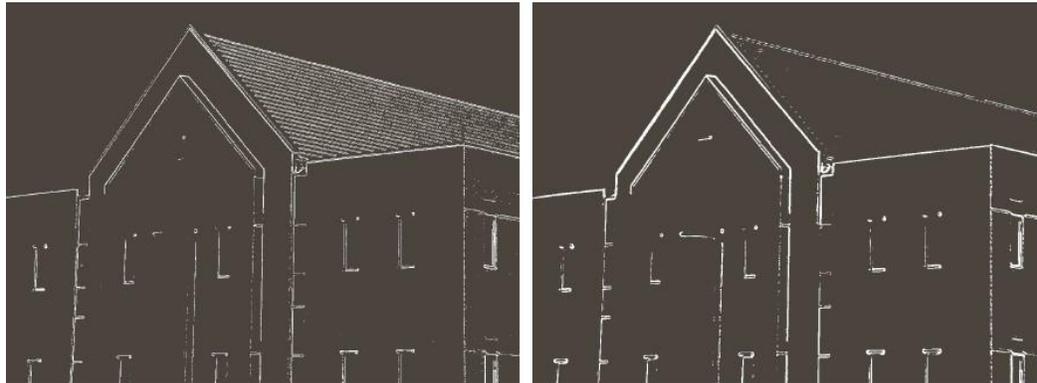
a b
c d

FIGURE 10.18
Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.

基本边缘检测

□ 与阈值处理相结合的梯度

- 在计算梯度前对图像进行平滑处理，边缘检测可做更多的选择。
- 实现相同基本目标的另一种方法是，对**梯度图像进行阈值处理**。
 - ✓ 阈值处理后的图像边缘更少，并且图像中的边缘要清晰得多。
 - ✓ 另一方面，许多边缘被断开了
- 实践中通常既做平滑处理也做阈值处理
 - ✓ 突出主要边缘，并尽可能保持连接



a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.



更先进的边缘检测

- 用于边缘检测的算子应有两个特点
 - 能计算图像中每一个点处的一阶导数或二阶导数的数字近似的微分算子(灰度突变在一阶导数中给出一个峰值或谷值, 或等效地在二阶导数中给出过零点)
 - 能被“调整”以便在任何期望的尺寸上起作用, 因此大算子也可用于检测模糊边缘, 小算子可用于检测锐度集中的精细细节(灰度变化与图像尺度是相关的, 检测需要不同大小的算子)

□ Marr-Hildreth边缘检测器（**马尔算子**）

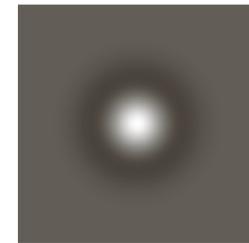
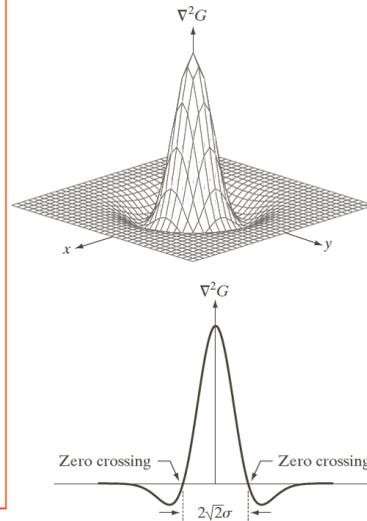
- 满足上述条件的算子是 $\nabla^2 h$ ：高斯拉普拉斯（LoG）
- 下图显示了一个LoG的负函数的三维图、图像和剖面

$$h(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad g(x, y) = h(x, y) \otimes f(x, y)$$

$$\nabla_g^2 = \nabla^2[h(x, y) \otimes f(x, y)] = \nabla^2 h(x, y) \otimes f(x, y)$$

$$= \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \otimes f(x, y)$$

$$\nabla^2 h = h''(r) = \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (LOG)$$



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

a b
c d

FIGURE 10.21
(a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

墨西哥草帽算子



更先进的边缘检测

□ Marr-Hildreth边缘检测器（马尔算子）

- 算子的高斯部分会模糊图像，但不存在振铃效应
- 关于 ∇^2 ：尽管一阶导数可用于检测灰度突变，但它们是有方向的算子。另一方面，拉普拉斯有**各向同性**的优点，符合人的视觉系统特性

□ Marr-Hildreth边缘检测算法可小结如下：

- 用一个2-D的高斯平滑模板与源图象卷积
- 计算卷积后图象的拉普拉斯值
- 检测拉普拉斯图象中的**过零点**作为边缘点

The Marr-Hildreth algorithm consists of convolving the LoG kernel with an input image,

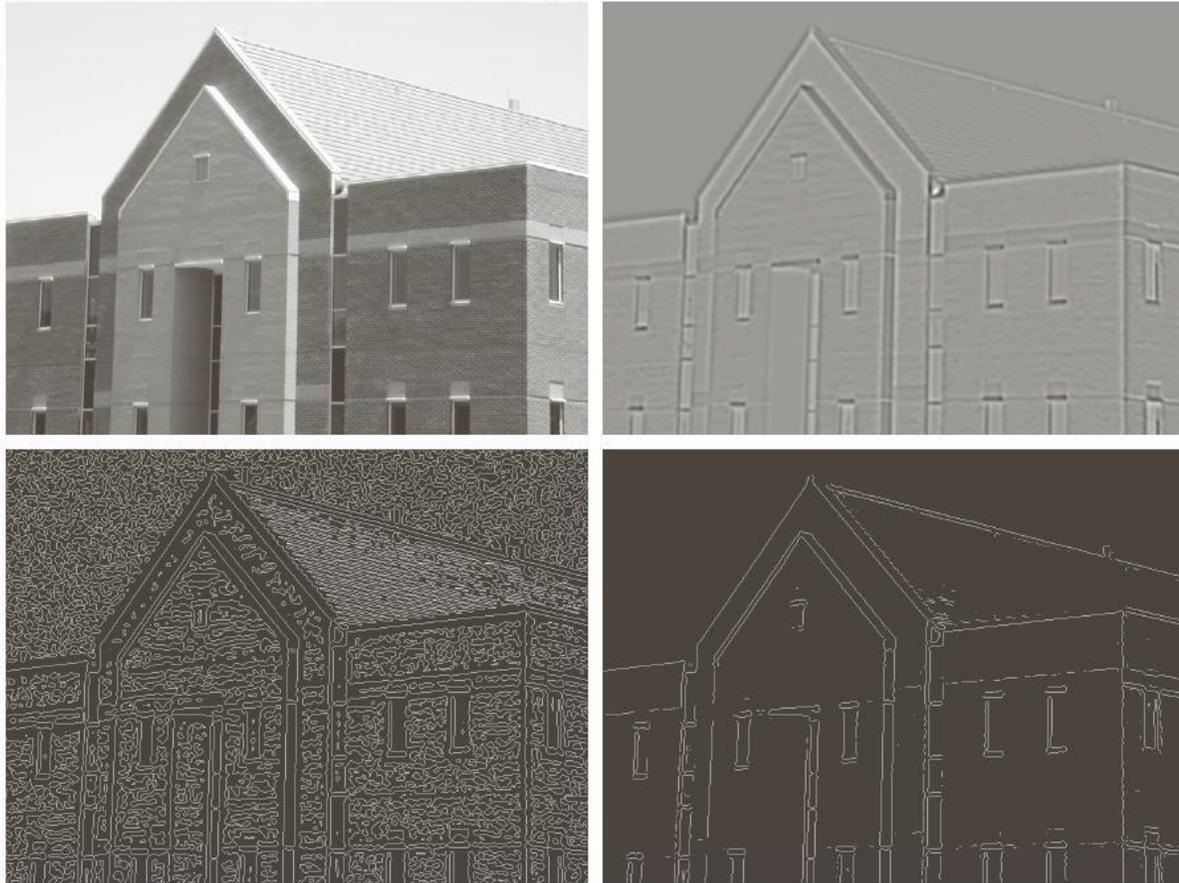
$$g(x, y) = [\nabla^2 G(x, y)] \star f(x, y) \quad (10-30)$$

and then finding the zero crossings of $g(x, y)$ to determine the locations of edges in $f(x, y)$. Because the Laplacian and convolution are linear processes, we can write Eq. (10-30) as

$$g(x, y) = \nabla^2 [G(x, y) \star f(x, y)] \quad (10-31)$$

更先进的边缘检测

□ Marr-Hildreth边缘检测算法说明



a b
c d

FIGURE 10.22

(a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.



边缘连接和边界检测

- 理想情况下，边缘检测应该仅产生位于边缘上的像素集合。
- 实际上，由于噪声、不均匀照明引起的**边缘间断**，以及其他引入灰度值虚假的不连续的影响，这些像素并不能完全描述边缘特性。
- 因此，一般是在边缘检测后紧跟**连接算法**，将边缘像素组合成有意义的边缘或区域边界。

边缘连接和边界检测

□ 使用局部处理的边缘连接



a b c
d e f

FIGURE 10.27 (a) A 534×566 image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning. (Original image courtesy of Perceptics Corporation.)



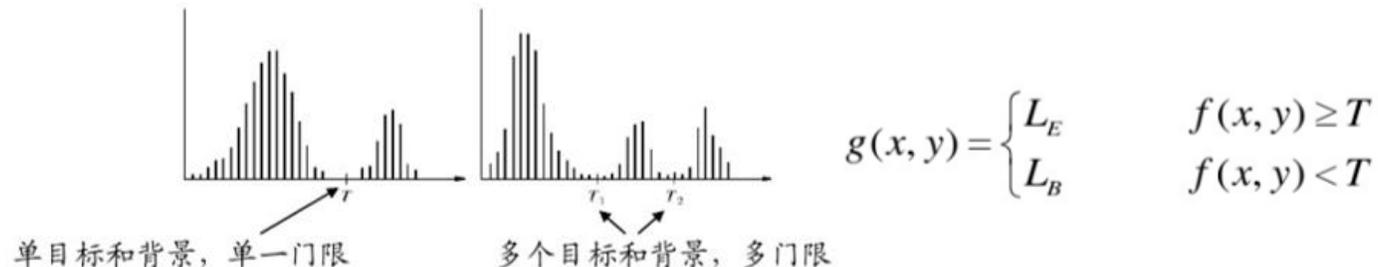
图像分割

1. 基础知识
2. 点、线和边缘检测
3. 阈值处理
4. 基于区域的分割

阈值处理

- 由于阈值处理直观、实现简单且计算速度快，因此图像阈值处理在图像分割应用中处于核心地位。
- 在前面的章节中，我们采用首先寻找边缘线段，然后将这些线段连接为边界的方法来识别区域。本节讨论基于灰度值或灰度值的特性来将图像直接划分为区域的技术

- 若想从亮的背景中分离出暗的物体，利用一阈值（阈值） T 将像素分为“亮”的和“暗”的两类。
- 利用对一些差分算子的响应值进行门限化，将像素分为“边缘”上的点和“非边缘”上的点两类。（类似边缘检测）



➤ 阈值分割法的特点：

- ✓ 简单，处理方便
- ✓ 适用于物体与背景有较强对比的情况，重要的是背景或物体的灰度比较单一，**确定合适的阈值是关键**
- ✓ 这种方法总可以得到封闭且连通区域的边界
- ✓ 噪声、前景或背景灰度变化范围大时将难以应用

- ❖ **全局阈值**—对全图采用单一阈值

$$T = T(f(x, y))$$

- ❖ **局部阈值**—不同局部采用不同阈值

$$T = T(p(x, y), f(x, y))$$

- ❖ **动态或自适应阈值**—每像素点阈值随像素特性而变化

$$T = T(x, y, p(x, y), f(x, y))$$

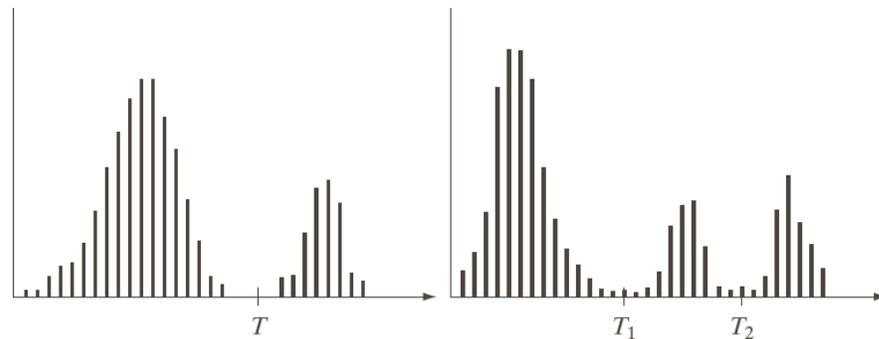
基础知识

□ 灰度阈值处理基础

- 左图中的灰度直方图对应于图像 $f(x, y)$ ，该图像由暗色背景上的较亮物体组成。
- 从背景中提取物体：选择一个将这些模式分开的阈值 T ；然后， $f(x, y) > T$ 的任何点 (x, y) 称为一个对象点；否则该点称为背景点。
- 换句话说，分割后图像 $g(x, y)$ 由下式给出：

$$g(x, y) = \begin{cases} 1 & \text{如 } f(x, y) > T \\ 0 & \text{如 } f(x, y) \leq T \end{cases}$$

- 当 T 是一个适用于整个图像的常数时，称为全局阈值处理。当 T 在一幅图像上改变时，使用可变阈值处理这一术语



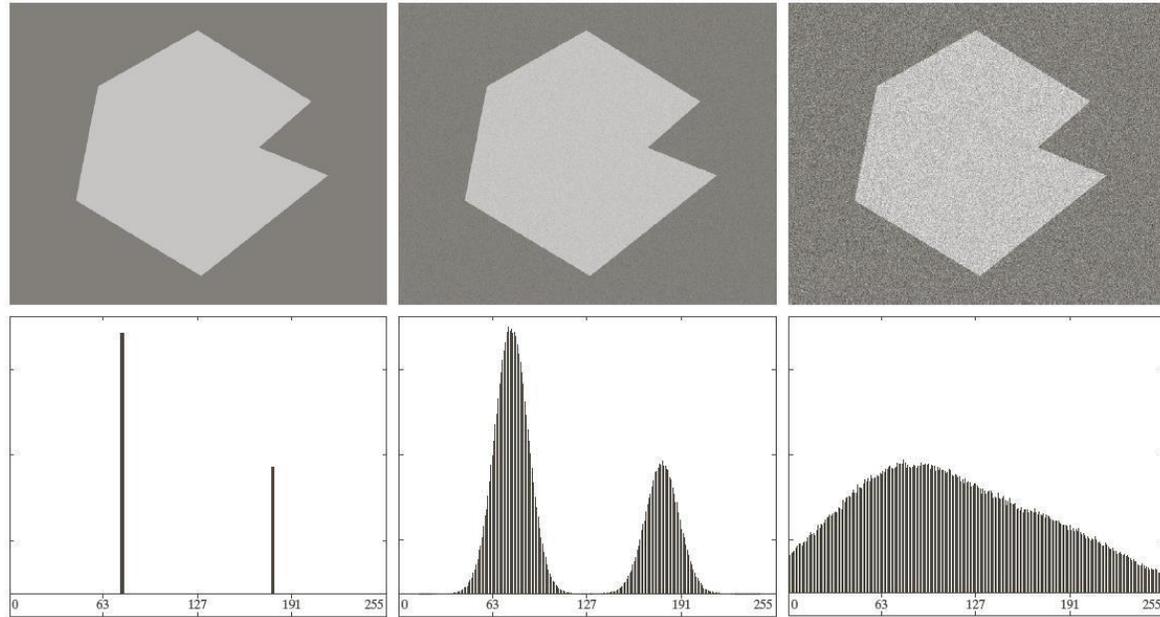
a b

FIGURE 10.35
Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

- 右图显示了一个更为困难的阈值处理问题，它包含有三个支配模式的直方图。我们可以采用双阈值处理的方法

基础知识

□ 图像阈值处理中噪声的作用



a b c
d e f

FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

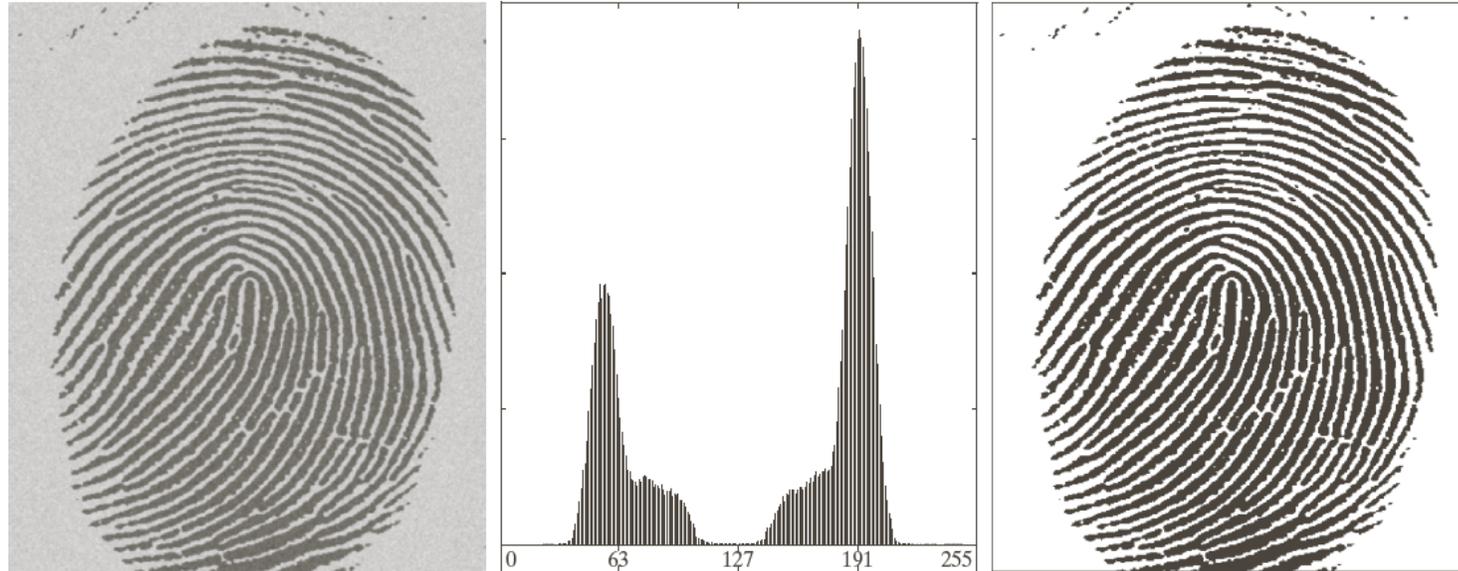


基本的全局阈值处理

- 当物体和背景像素的灰度分布十分明显时，可以用适用于整个图像的单个（全局）阈值。
- 在大多数应用中，通常图像之间有较大变化，即使全局阈值是一种合适的方法，也需要有能对每幅图像自动估计阈值的算法。
- 下面的迭代算法可用于这一目的（本质是k-means聚类，k=2）
 - （1）为全局阈值T选择一个初始估计值
 - （2）用单个阈值T分割该图像。这将产生两组像素： G_1 由灰度值大于T的所有像素组成， G_2 由所有小于等于T的像素组成
 - （3）对 G_1 和 G_2 的像素分别计算平均灰度值（均值） m_1 和 m_2
 - （4）计算一个新的阈值：
$$T = \frac{1}{2}(m_1 + m_2)$$
 - （5）重复步骤（2）到步骤（4），直到连续迭代中的T值间的差小于一个预定义参数 ΔT 为止
- 参数 ΔT 用于控制迭代的次数。通常， ΔT 越大，则算法执行的迭代次数少。图像的平均灰度对于T来说是较好的初始选择

基本的全局阈值处理

□ 全局阈值处理



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

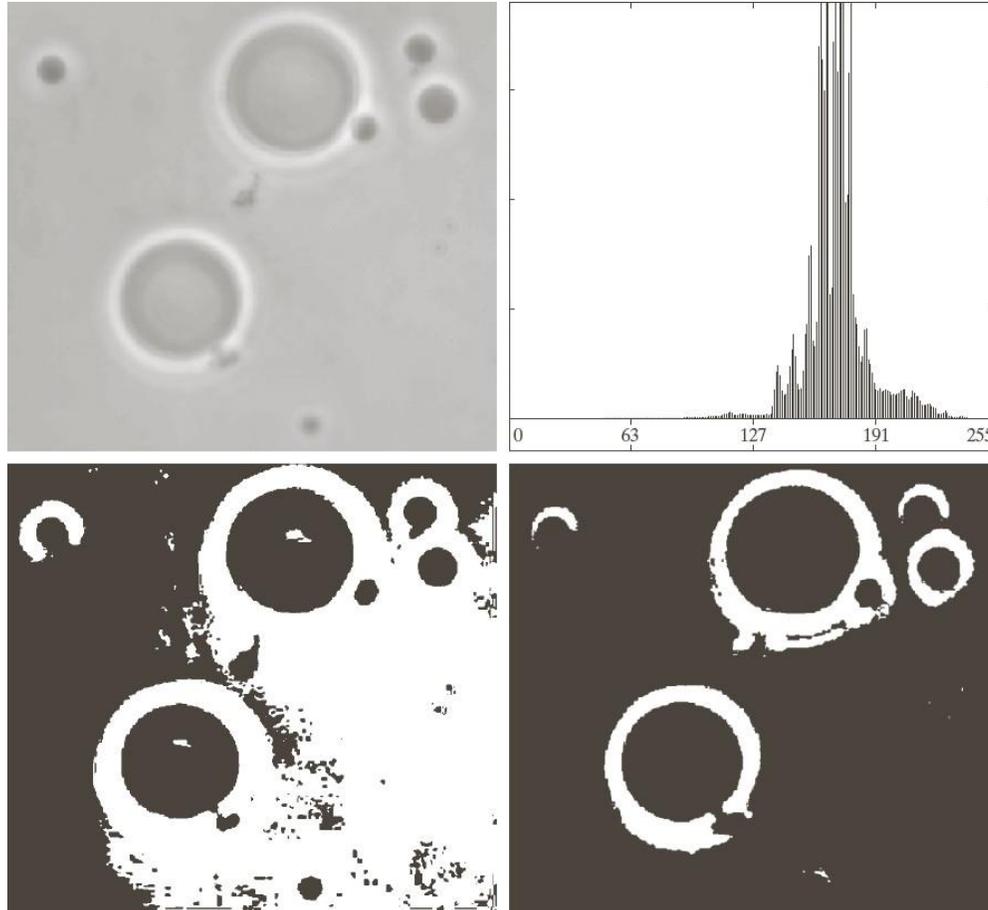


Otsu方法：最佳全局阈值处理

- 该方法在类间方差最大的情况下是最佳的；完全以在一幅图像的直方图上执行计算为基础。
- Otsu算法小节如下：
 - 计算输入图像的归一化直方图。使用 $p_i, i = 0, 1, 2, \dots, L - 1$ 表示该直方图的各个分量
 - 计算累积和： $P_1(k) = \sum_{i=0}^k p_i, (k = 0, 1, 2, \dots, L - 1), P_2(k) = 1 - P_1(k)$
 - 计算累积均值： $m(k) = \sum_{i=0}^k ip_i, 对于k = 0, 1, 2, \dots, L - 1,$
 - 计算全局灰度均值： $m_G = \sum_{i=0}^{L-1} ip_i$
 - 用 $\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$ ，对于 $k = 0, 1, 2, \dots, L - 1$ ，计算类间方差 $\sigma_B^2(k)$
 - 得到Otsu阈值 k^* ，即使得 $\sigma_B^2(k)$ 最大的 k 值。如果最大值不唯一，用相应检测到的各个最大值的 k 的平均得到 k^*
 - 在 $k = k^*$ 处计算 $\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$ ，得到可分性测度 η^*

用Otsu方法的最佳全局阈值处理

- 使用Otsu方法的最佳全局阈值处理



a b
c d

FIGURE 10.39

(a) Original image.
(b) Histogram (high peaks were clipped to highlight details in the lower values).
(c) Segmentation result using the basic global algorithm from Section 10.3.2.
(d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

用图像平滑改善全局阈值处理

- 噪声会将简单的阈值处理问题变为不可解决的问题
 - 解决方法：在阈值处理之前平滑图像

□ 例：

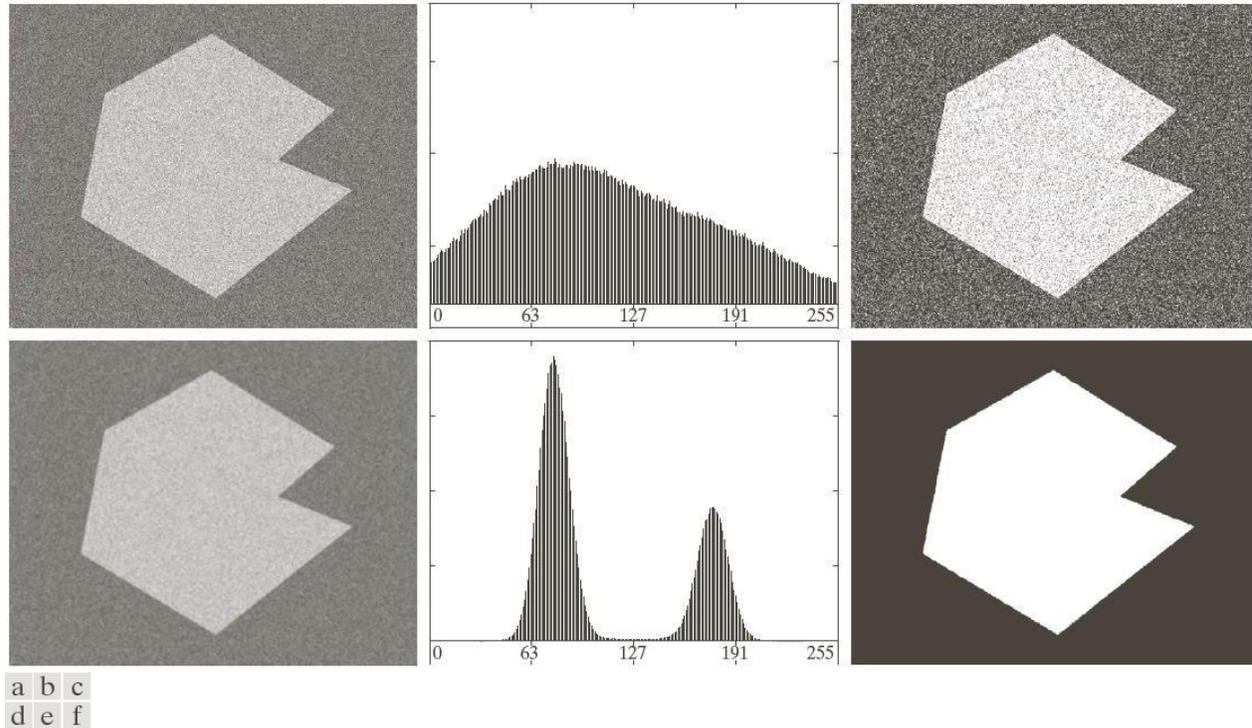


FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

用图像平滑改善全局阈值处理

- 考虑降低图中相对于背景的区域大小所带来的影响
 - 失败的原因可归于这样一个事实，即区域太小，以至于该区域对直方图的贡献与由噪声引起的灰度扩散相比无足轻重。

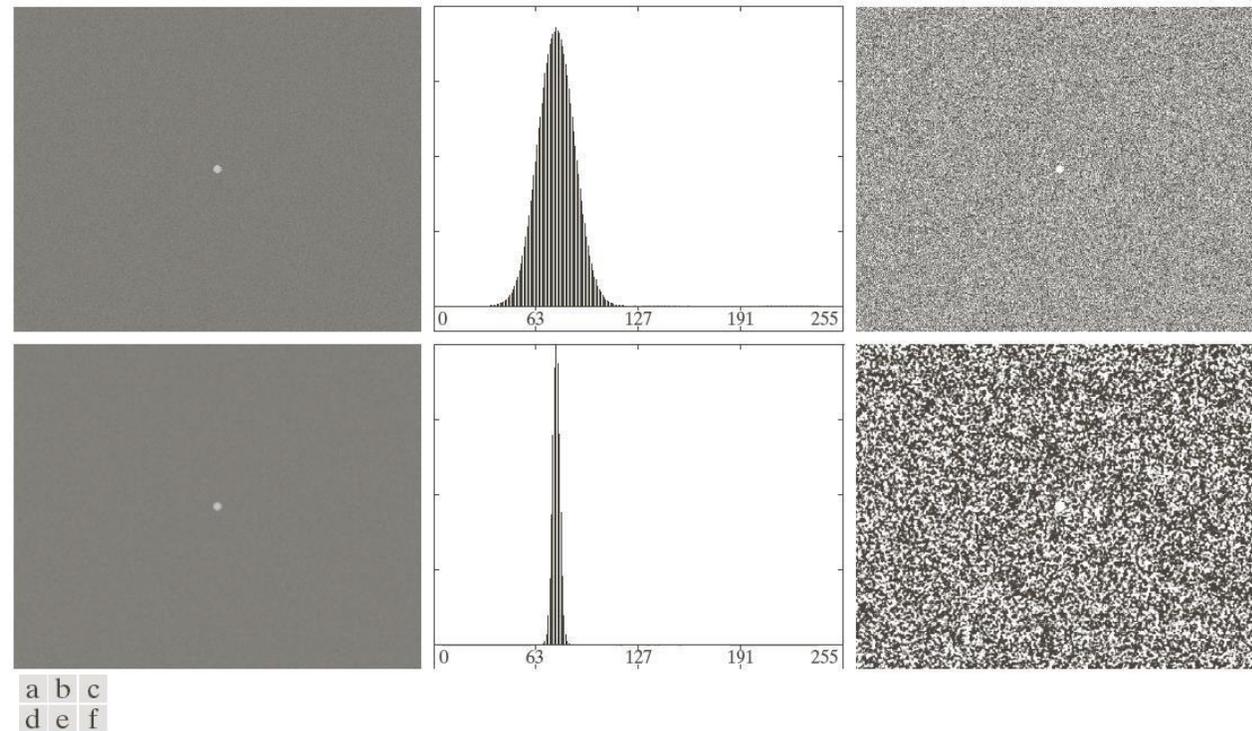


FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

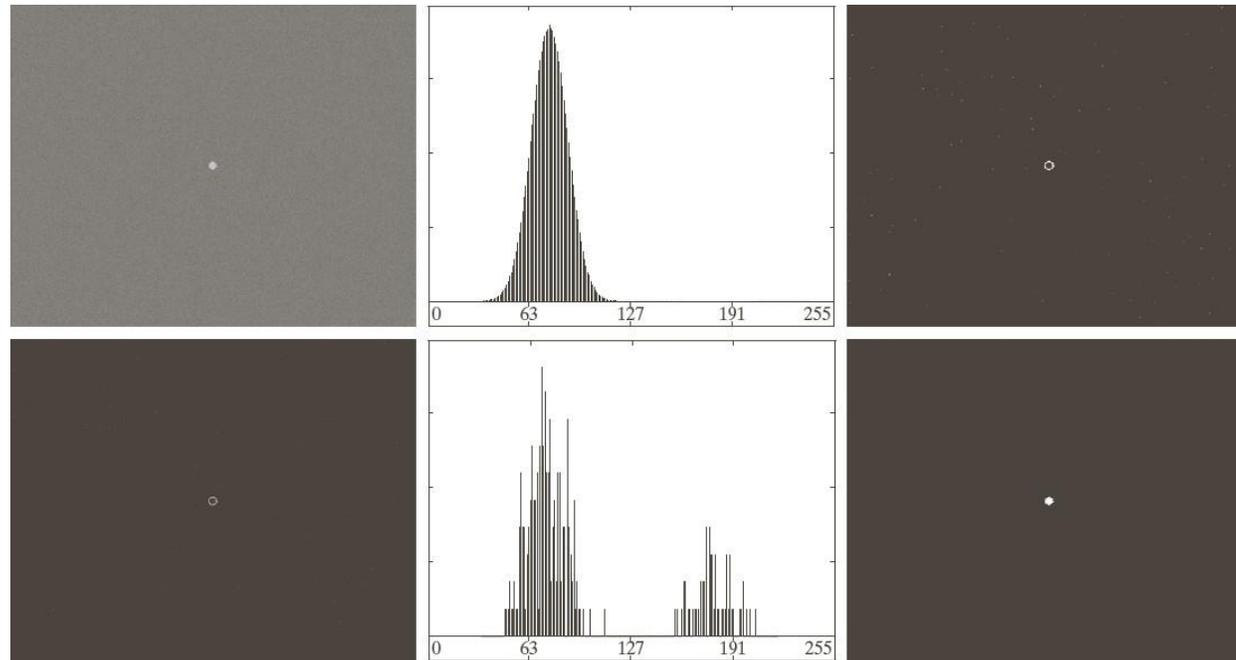


利用边缘改进全局阈值处理

- 如果直方图的波峰是高、窄、对称的，且被深波谷分开，则选取一个“较好”阈值的机会相当大。改进直方图形状的一种方法是，仅考虑那些位于或靠近物体和背景间的边缘的像素。该算法可小结如下，其中 $f(x, y)$ 是输入图像：
 - 1、计算一幅边缘图像，无论是 $f(x, y)$ 梯度的幅度还是拉普拉斯的绝对值均可
 - 2、指定一个阈值 T
 - 3、用步骤2中的阈值对步骤1中的图像进行阈值处理，产生一幅二值图像 $g_T(x, y)$ 。在从 $f(x, y)$ 中选取对应于“强”边缘像素的下一步中，该图像用做一幅模板图像
 - 4、仅用 $f(x, y)$ 中对应于 $g_T(x, y)$ 中像素值为1的位置的像素计算直方图
 - 5、用步骤4中直方图全局地分割 $f(x, y)$ ，例如使用Otsu方法

利用边缘改进全局阈值处理

- 用以梯度为基础的边缘信息改进全局阈值处理

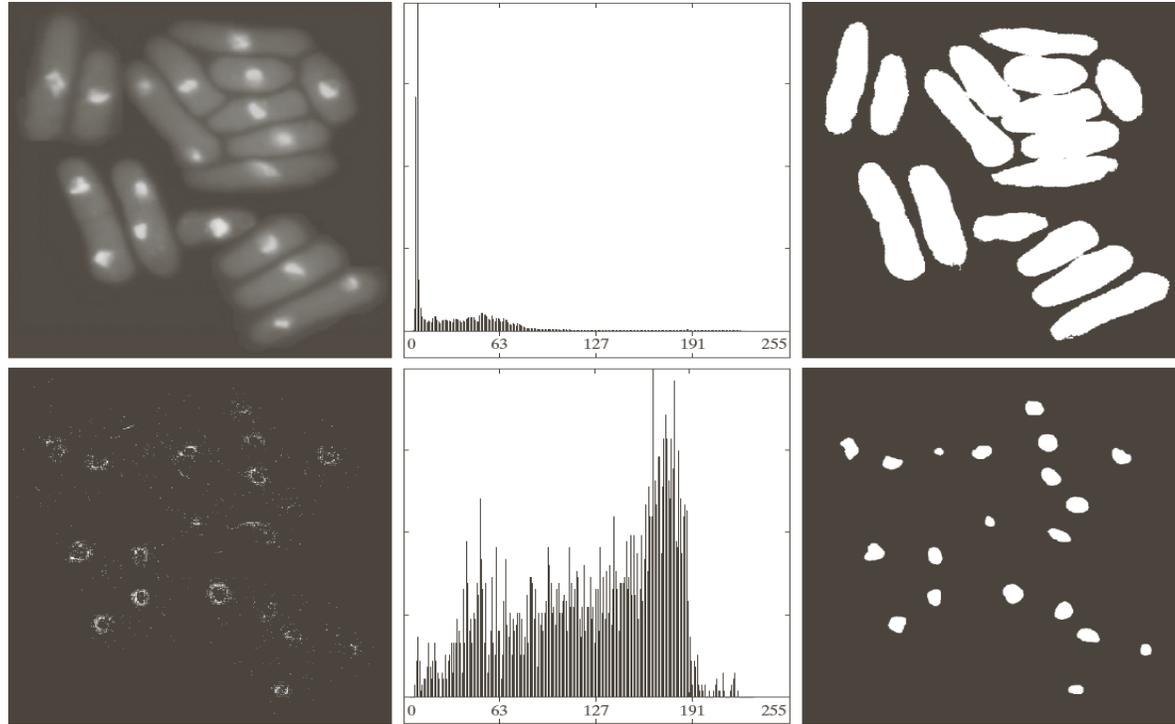


a b c
d e f

FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.

利用边缘改进全局阈值处理

- 用拉普拉斯为边缘信息的改进全局阈值处理



a b c
d e f

FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

多阈值处理

- 在K个类 C_1, C_2, \dots, C_K 的情况下，类间方差可归纳为下面的表达式：

$$\sigma_B^2 = \sum_{k=1}^K P_k (m_k - m_G)^2$$

式中， $P_k = \sum_{i \in C_k} P_i$ ， $m_k = \frac{1}{P_k} \sum_{i \in C_k} iP_i$ ， $m_G = \sum_{i=0}^{L-1} ip_i$

K类有K-1个阈值来分离，这些值 $k_1^*, k_2^*, \dots, k_{K-1}^*$ 是上式的最大值：

$$\sigma_B^2(k_1^*, k_2^*, \dots, k_{K-1}^*) = \max_{0 < k_1 < k_2 < \dots < k_{K-1} < L-1} \sigma_B^2(k_1, k_2, \dots, k_{K-1})$$

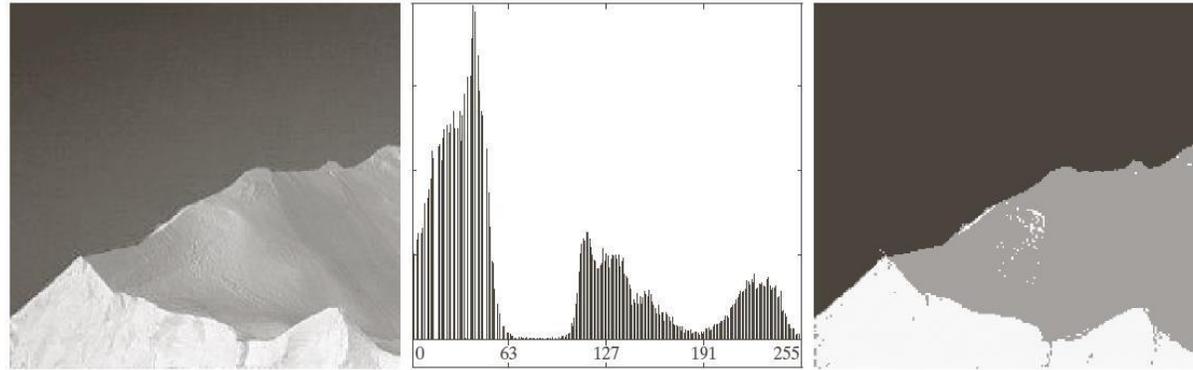
- 对于由三个灰度间隔组成的三个类（两个阈值），最佳阈值是：

$$\sigma_B^2(k_1^*, k_2^*) = \max_{0 < k_1 < k_2 < L-1} \sigma_B^2(k_1, k_2)$$

该过程由选择第一个 k_1 值开始（为1）。接着， k_2 的所有值在大于 k_1 和小于L-1的范围内增加。然后，将 k_1 增大到其下一个值， k_2 的所有值再次在大于 k_1 的所有值范围内增加。重复该过程，直到 $k_1 = L - 3$ 为止。该结果是一个二维阵列 $\sigma_B^2(k_1, k_2)$ ，最后一步是在阵列中寻找最大值。对应于最大值的 k_1 值和 k_2 值就是最佳阈值 k_1^* 和 k_2^* 。

多阈值处理

□ 多个全局阈值处理

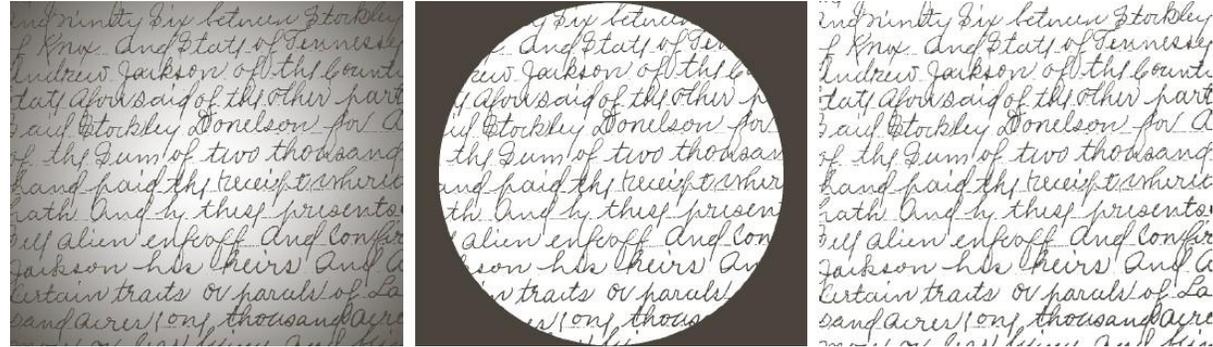


a b c

FIGURE 10.45 (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

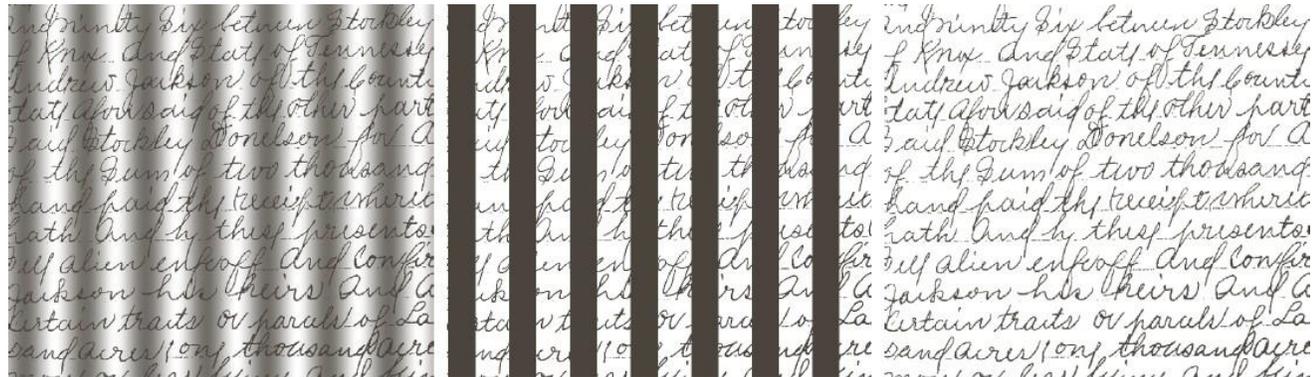
可变阈值处理

□ 用移动平均的文档阈值处理



a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.



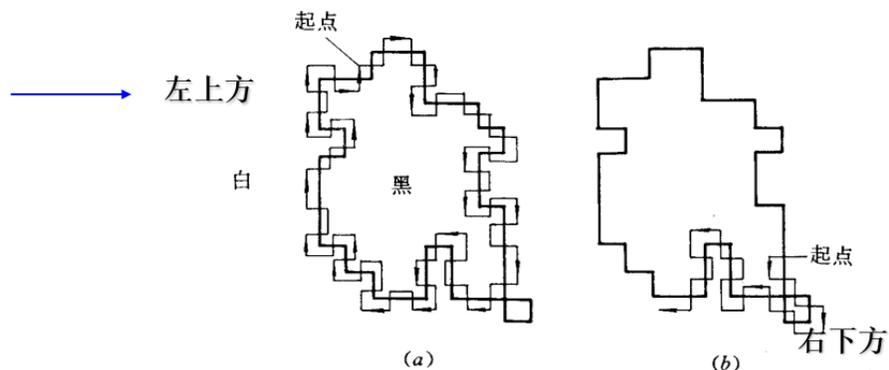
a b c

FIGURE 10.50 (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

➤ 跟踪方法 - 利用前期处理过的点的信息，判断图像当前处理点是否为目标点，再进行跟踪运算

轮廓跟踪 - 靠近边缘任取一起始点 A

- ① 每次只前进一步，步距为一个像素
- ② 当由白区进入黑区，各步向左转直到走出黑区
- ③ 当由黑区进入白区，各步向右转直到走出白区
- ④ 重复②③步，将会围绕目标一周后，回到起始点，完成轮廓跟踪





图像分割

1. 基础知识
2. 点、线和边缘检测
3. 阈值处理
4. 基于区域的分割

基于区域的分割

- 分割的目的是将一幅图像划分为多个区域。我们首先**基于灰度级的不连续性**尝试寻找区域间的边界来解决这一问题；接下来，分割是通过**以像素特性分布为基础**的阈值处理来完成的，如灰度值或彩色。

✎ **基于区域相关**：依据同一区域具有相似的灰度特征或组织特征，寻求不同区域的边界

方法

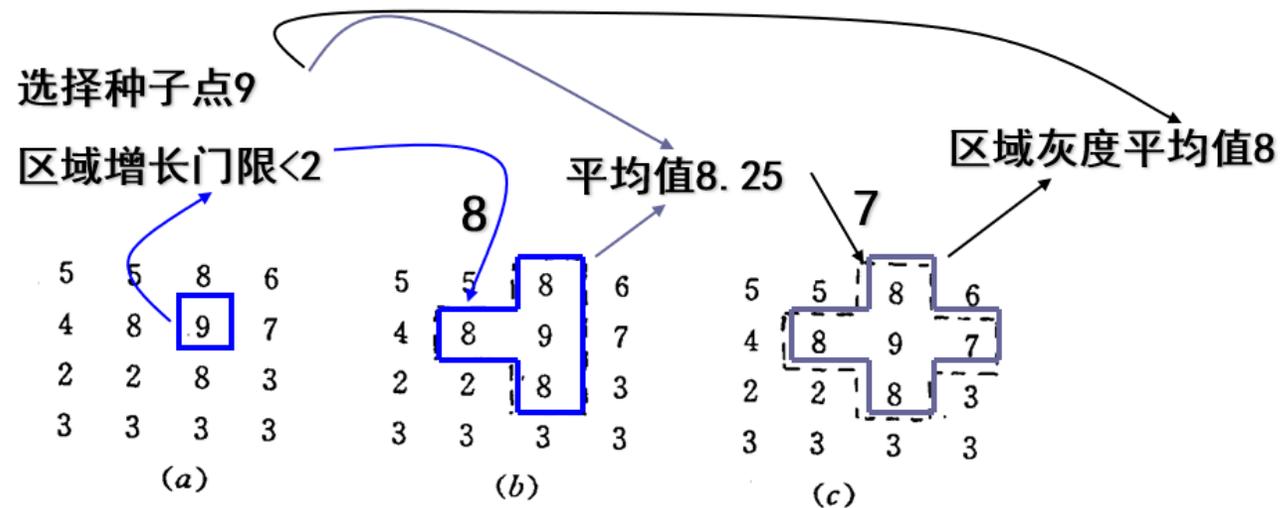
➡ 区域增长

➡

区域增长

举例

- 以9所在像素为种子点
- 增长条件：每一步所接受的邻近点的灰度级与先前位置的平均灰度级相差小于2（增长门限）。



区域增长

从一组“种子”点开始，将于种子预先定义的性质相似的那些邻域像素添加到每个种子上来，形成这些生长区域。

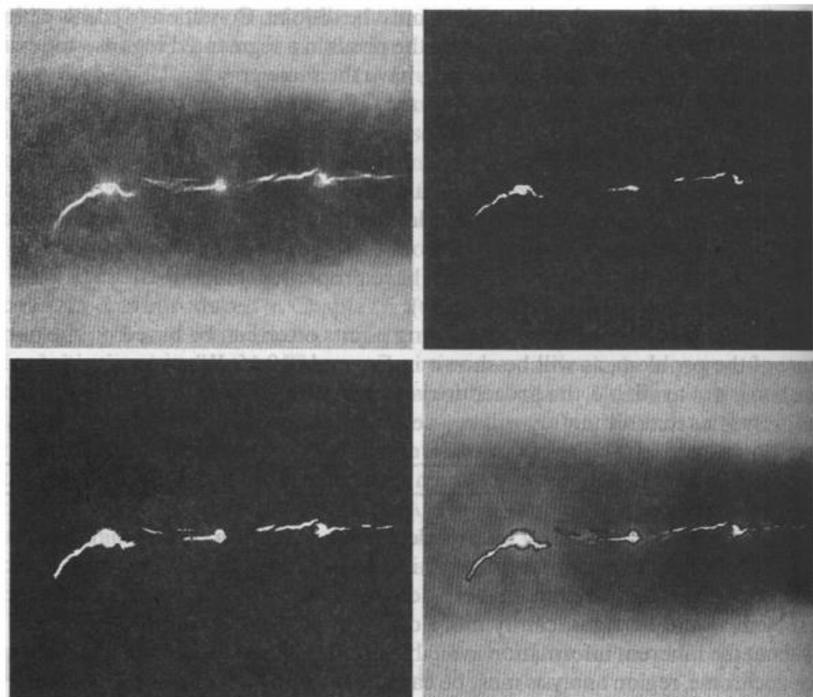
把图像分成许多小区域，这些初始的区域可能是单个像素。在每个区域中，对经过适当定义的能反映一个物体隶属度的性质进行计算，在各个方向上生长出区域。

区域增长的算法实现：

- ① 根据图像的不同应用选择一个或一组种子，它或者是最亮或最暗的点，或者是位于点簇中心的点
- ② 选择一种描述符（像素增长的约束条件）
- ③ 从该种子开始向外扩张，首先把种子像素加入结果集合，然后不断将与集合中各个像素连通、且满足描述符的像素加入集合
- ④ 上一过程进行到不再有满足条件的新结点加入集合为止

区域增长举例

有缺陷
的焊缝



种子点(初
始区域)

区域生
长结果

分割后
得到的
边界





数字图像处理 (Digital Image Processing)

王正

Email: zheng_wang@seu.edu.cn
办公室: 无线谷A5415

**Image
Processing
Project Ideas**
Beginners



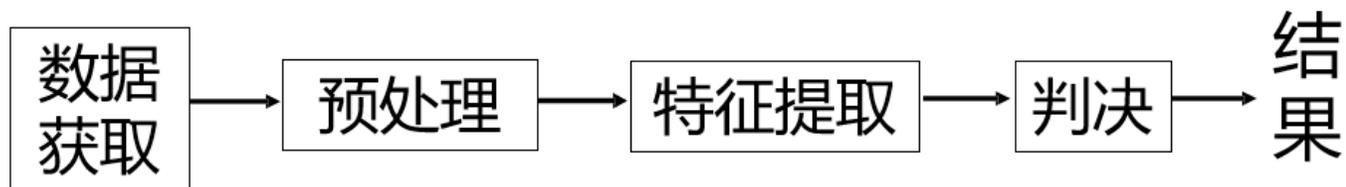
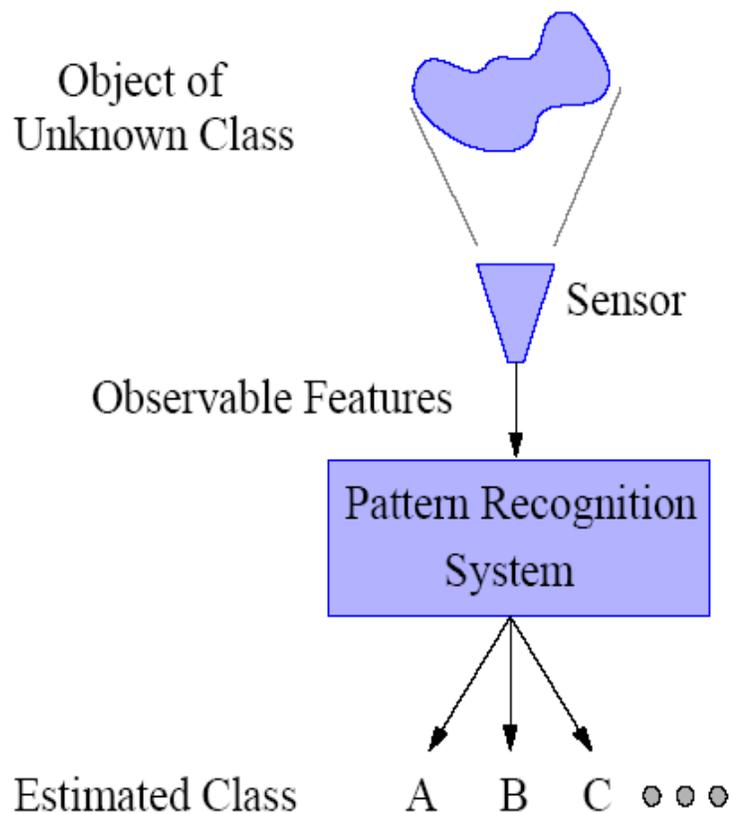


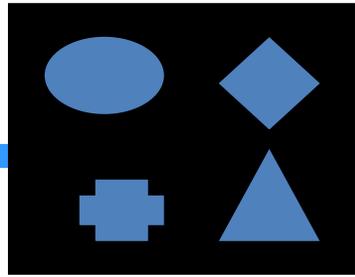
■ 模式识别的理论与方法

- 模式识别概述
- 基本概念—模式与模式类
- 统计模式识别——决策论法
- 句法模式识别——结构法
- 模式识别的应用实例

- 图像分析的重要目的
 - ✓ 图像分割得到目标
 - ✓ 纹理分析得到表示纹理的特征
 - ✓ 图形描述得到特定目标图像的抽象化描述
 - ✓ 图像的模式识别进一步明确目标的性质与特征
- 图像识别（模式识别）：对待研究的对象，根据其某些特征，进行识别并分类
- 文字识别是迄今为止人类最熟悉、应用最成功的图像识别领域
- 目前在指纹鉴别、手语识别、面容识别、表情识别等方面正在取得进展
- 进一步的发展结合人工智能，向更高级的识别水平发展

模式识别系统



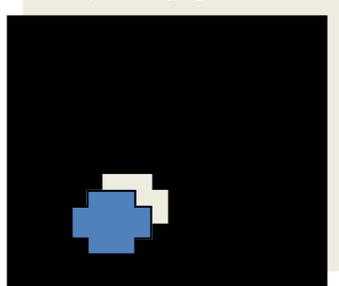


输入图像

模式识别过程与实现

图像分割

物体
图像



特征提取

特征
向量

x_1
 x_2
 \cdot
 x_N

分类

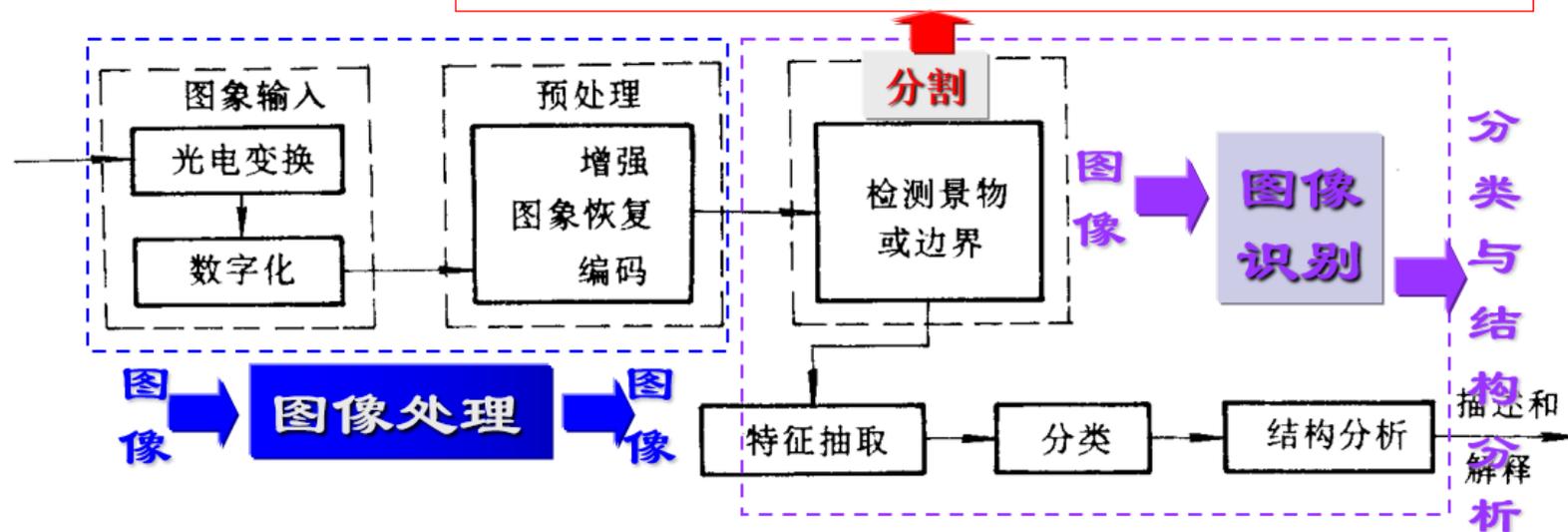
物体
类型

- ① 物体检测器设计
- ② 特征提取
- ③ 分类器设计
- ④ 分类器训练
- ⑤ 性能评估

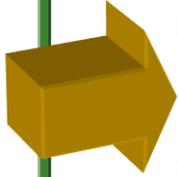
连通集中的任意两个像素之间，存在一条完全由该集合的元素构成的连通路径

像素连
通集

数字图像划分成互不相交区域的过程



特征提取



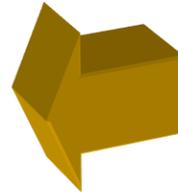
- ① 可区别性
- ② 可靠性
- ③ 独立性
- ④ 数量少



分类器

- ① 特征方差
- ② 特征相关系数
- ③ 类间距离

... ..



特征选择

有哪些可提取的特征!!!

幅度特征

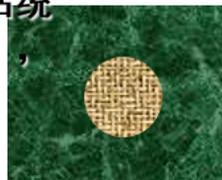
像素灰度值
三色值
频谱值
邻域平均灰度幅度

统计特征

直方图
均值、方差、熵
自相关、协方差、矩、能量

纹理特征

反映一个区域中像素灰度级的空间分布的属性。包括统计纹理特征（共生矩阵）、频谱特性和结构特征等。



特征提取

边界特征

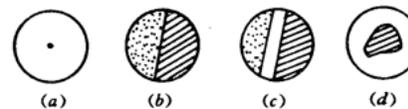
亮度突变、断续
灰度跃变高度、变化倾角
彩色边界特征

拓扑特征

孔、洞 H
连接特征 C
欧拉数 $E=C-H$

点线边缘

模板匹配





- 模式识别 (pattern recognition, 亦可译为模式辨认、图像识别、图形识别、型式识别) 是近30年来得到迅速发展的一门新兴边缘学科。关于什么是模式或者机器所能辨认的模式, 迄今还没有一个确切而严格的定义。卡纳尔 (Kanal . L.) 曾经说过这样一段话:
 - ✓ “关于什么是模式识别和机器所能辨认的模式, 至今还没有人能象香农(Shannon)对‘信息’一词做出定义那样, 给出一个确切的定义。如果一旦出现了这样一个定义并被证实能够推动理论的发展, 那将标志着人类智力的一大进展。虽然如此, 目前的局面并不影响模式识别在各领域中的广泛应用。”
- 我们可以设计出自动图像识别系统, 但仍然**缺乏理论依据**。有待人类视觉认识理论的进一步研究。
- 我们可以做出**在某一应用上超过人的能力**的系统, 但缺乏扩展性。**过分依赖具体应用**。



■ 图像模式识别的基础理论与方法

- 决策理论是图像识别的理论基础
- 特征和分类是图像识别的关键
- 图像识别的基本方式
 - ✓ 从待分类的图像中提取能够代表或排除某些类别的特征
 - ✓ 根据特征将图像目标分到相应的类别中
- 分类器的两个基本过程
 - ✓ 学习过程：首先分离具有典型性的特征，根据这些特征对每个类别建立独立的描述，建立特征相应的特征空间
 - ✓ 测试过程：根据输入样本和特征空间的划分，对全体图像进行分类

■ 图像模式识别的基本方法

- 识别的**统计分类方法**：对待分类图像进行大量统计分析的基础上，找出规律性，抽取反映目标特征的统计向量进行识别。
- 识别的**句法结构方法**：通过对图像结构的分析，一幅图像可模仿语言构造，用一些语句来表达。通过符号匹配、语法分析等，实现图像中目标的识别
- 宏观比较：统计识别方法很少利用图像本身的结构关系；句法结构方法未考虑图像获取过程中的噪声干扰

■ 现代方法研究

- 模糊模式识别、神经网络模式识别、……

课程介绍——拥抱人工智能第一站



◆ **课程性质：** 本科生大三专业选修课32课时；以需求为牵引，重概念！重理解！

◆ **课程目标：** 为信息领域提供具有机器学习与人工智能基础知识与背景的初级人才，为下一阶段的学习奠定基础

◆ **参考书籍：**

- 《机器学习》，周志华
- 《统计学习方法》，李航
- 《神经网络与深度学习》，邱锡鹏
- 《机器学习与应用》，雷明
- Machine Learning, T. Mitchell, McGraw Hill, 1997;
- Pattern Recognition & Machine Learning, C. M. Bishop, Springer, 2006



机器学习与人工智能Reading Group

1. 决策树
2. 贝叶斯分类器
3. 支持向量机
4. 降维
5. 聚类
6. 集成学习
7. EM算法和高斯混合模型
8. 隐马尔科夫模型
9. 条件随机场
10. 近似推断
11. 核学习
12. 神经网络
13. MCMC
14. 强化学习
15. 循环神经网络
16. GAN



基本概念

- 基本概念—模式与模式类
 - 模式的定义
 - 模式类的定义
 - 模式识别的定义
 - 常用的模式序列
 - ✓ 模式向量
 - ✓ 模式串
 - ✓ 模式树
-

■ 模式的定义

- 样本(sample, object): 一类事物的一个具体体现, 对具体的个别事物进行观测所得到的某种形式的信号或表达
- 模式(pattern): 图像中的一个对象或某些感兴趣区域的特征或结构的描述—物体抽象特征的描述
 - 模式是由一个或多个描述子来组成
 - 模式是一个描述子的序列 (名词“特征”经常被用来代指描述子)
 - 模式是一组特征或一组描述子
- 样本是具体的事物, 而模式是对同一类事物概念性的概括



■ 模式类的定义

- 模式所属的类别或同一类中模式的总体称为模式类，其中个别具体的模式常常称为样本
- 模式类是具有某些公共特征的模式系列
- 模式类与模式联合使用时，模式表示具体的事物，而模式类则是对这一类事物的概念性描述

■ 模式识别的定义

- 根据图像中对象的特征组成的模式，确定对象是属于那一个模式类，即为模式识别
- **模式识别是从样本到类别的映射**

■ 模式与模式类举例

- 已知汽车的长、宽、高 (x_1, x_2, x_3) ，希望识别出：
大客车、小轿车、卡车 (w_1, w_2, w_3)

- ✓ 汽车的长、宽、高 (L, W, H) ——模式

大客车: $(L, W, H)_{\text{大}}$; 小轿车: $(L, W, H)_{\text{小}}$; 卡车: $(L, W, H)_{\text{卡}}$

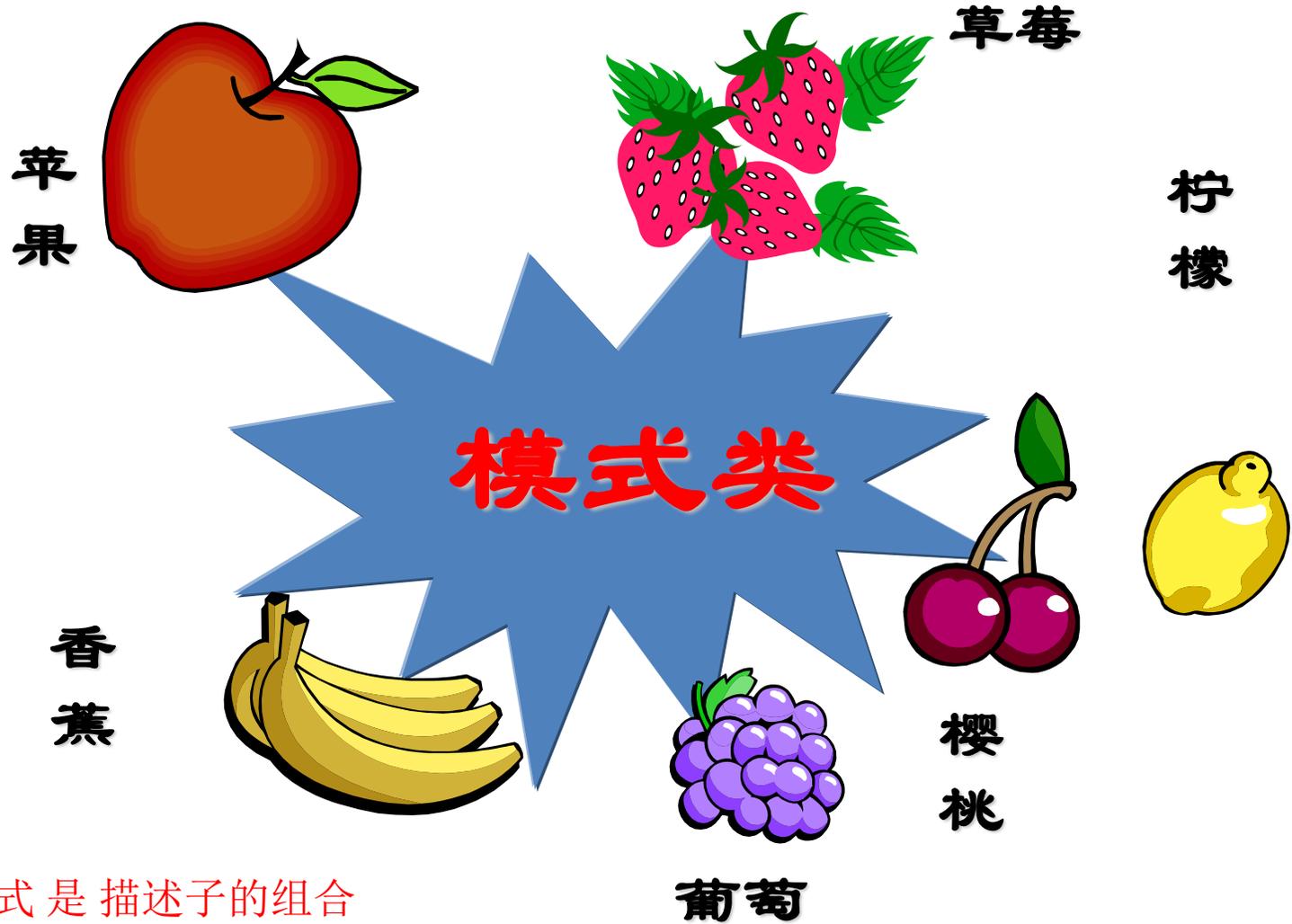
从而有模式类 $(w_{\text{大}}, w_{\text{小}}, w_{\text{卡}})$

- 定义了不同汽车的载重量，希望识别出大客车、小轿车、卡车 (w_1, w_2, w_3)

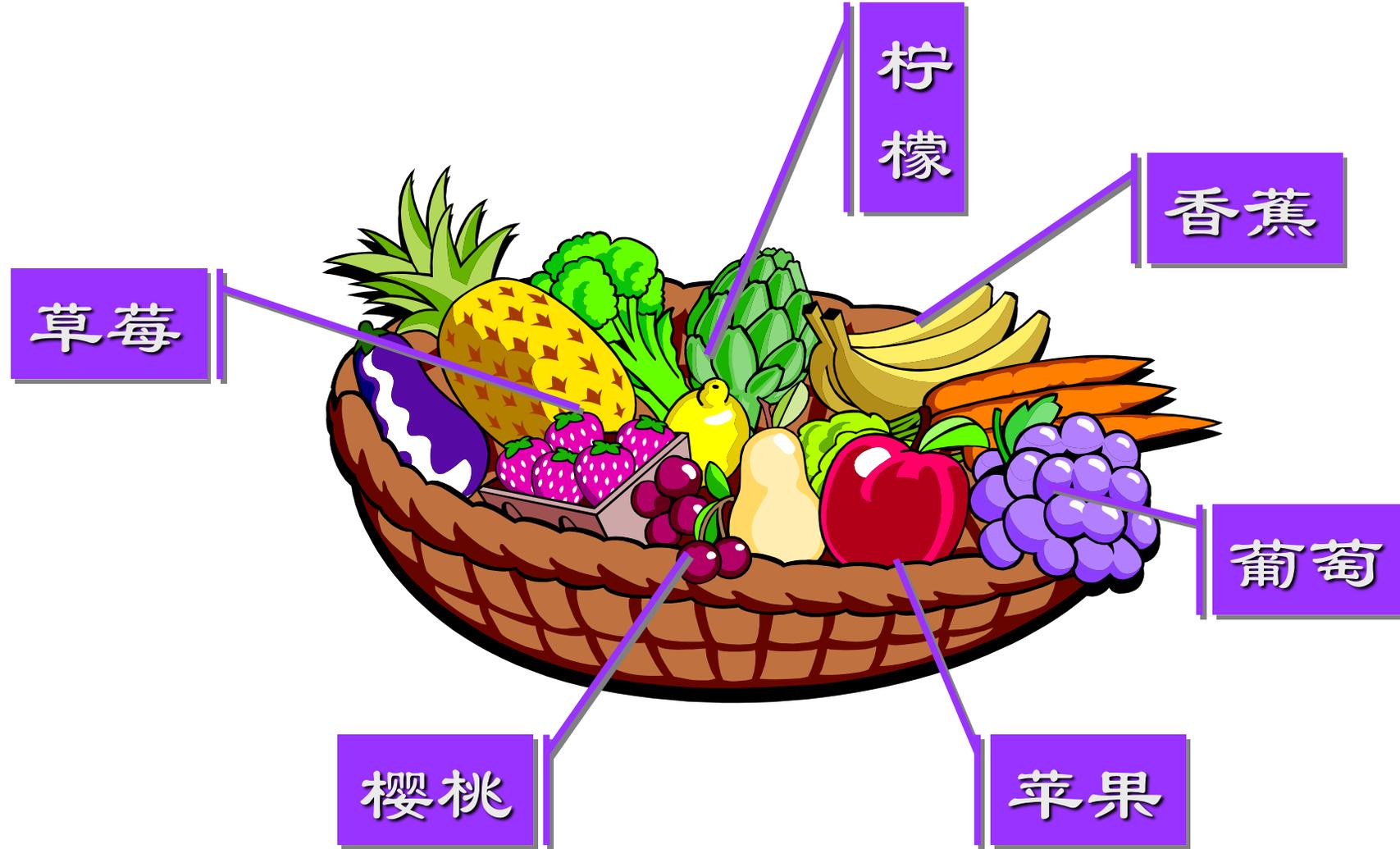
- ✓ 汽车的载重量 M ——模式

大客车: $M_{\text{大}}$; 小轿车: $M_{\text{小}}$; 卡 车: $M_{\text{卡}}$

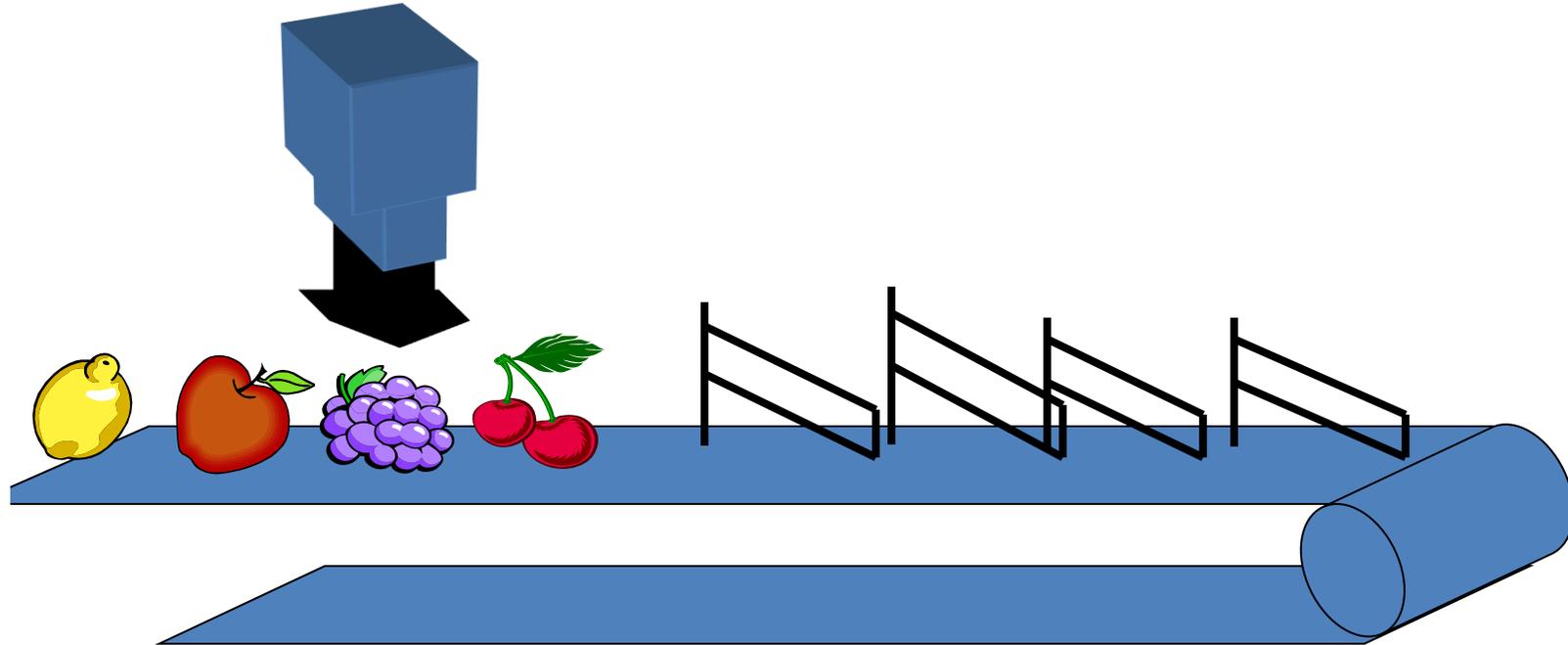
从而有模式类 $(w_{\text{大}}, w_{\text{小}}, w_{\text{卡}})$



模式是描述子的组合



CCD/TV



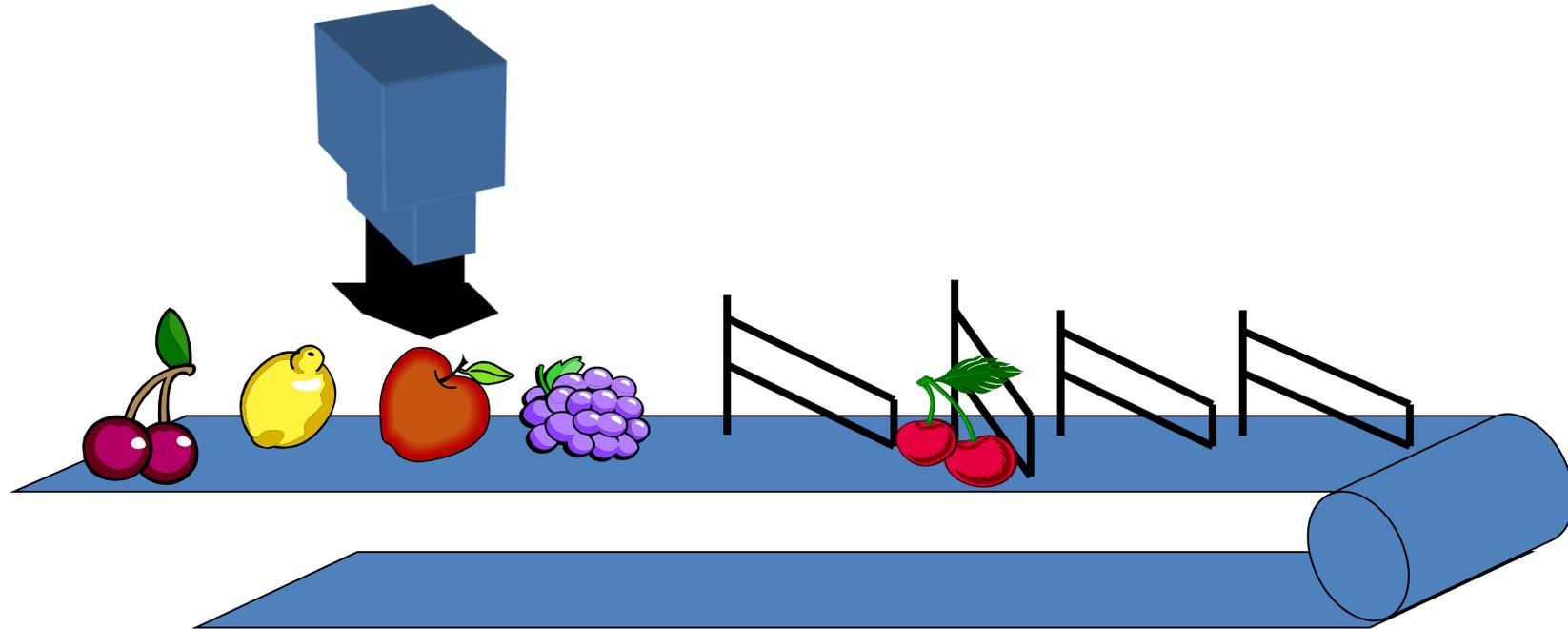
苹果

樱桃

柠檬

葡萄

CCD/TV



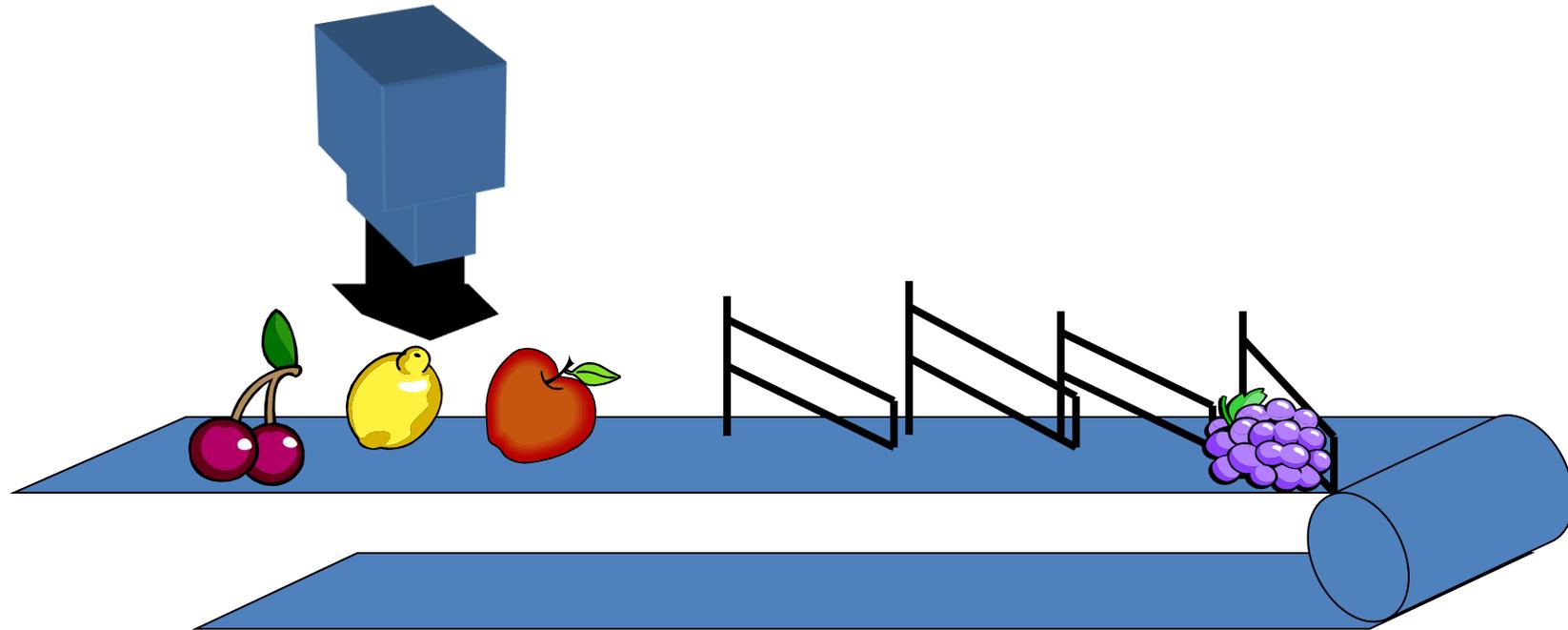
苹果

樱桃

柠檬

葡萄

CCD/TV



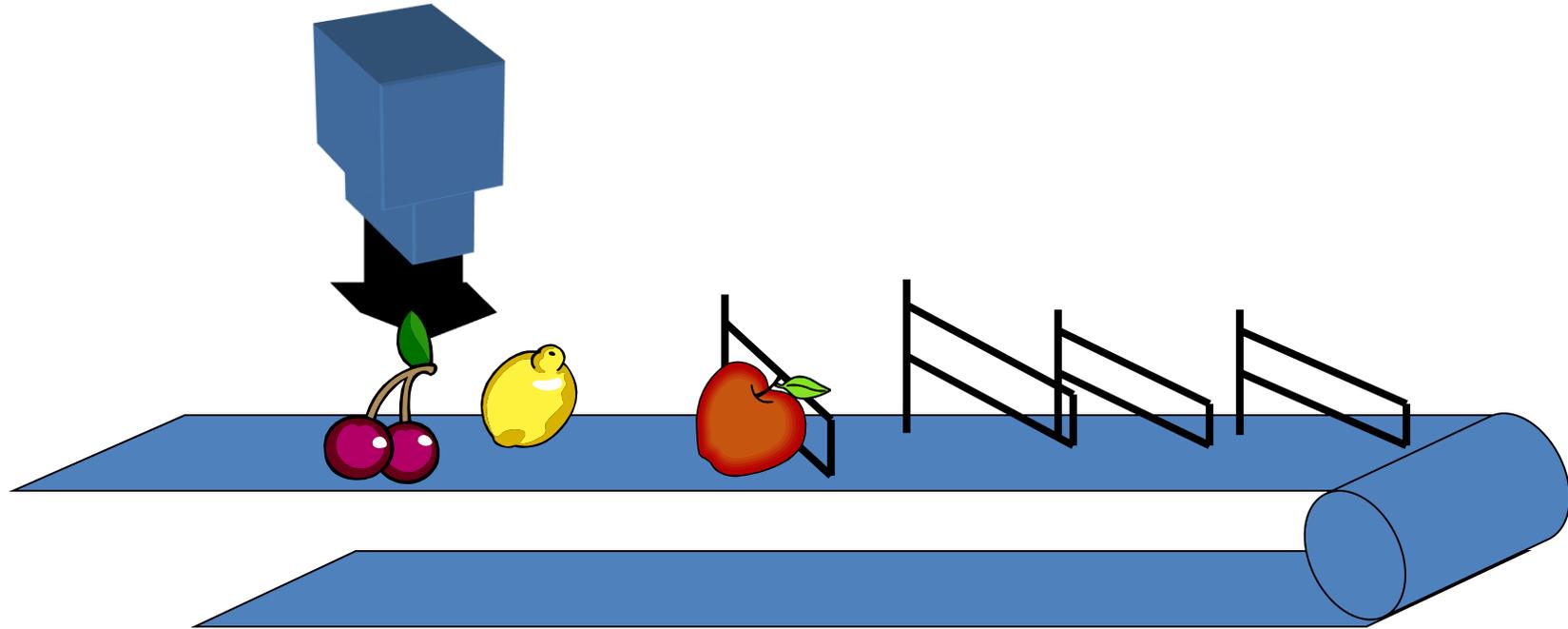
苹果

樱桃

柠檬

葡萄

CCD/TV



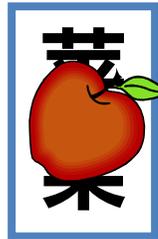
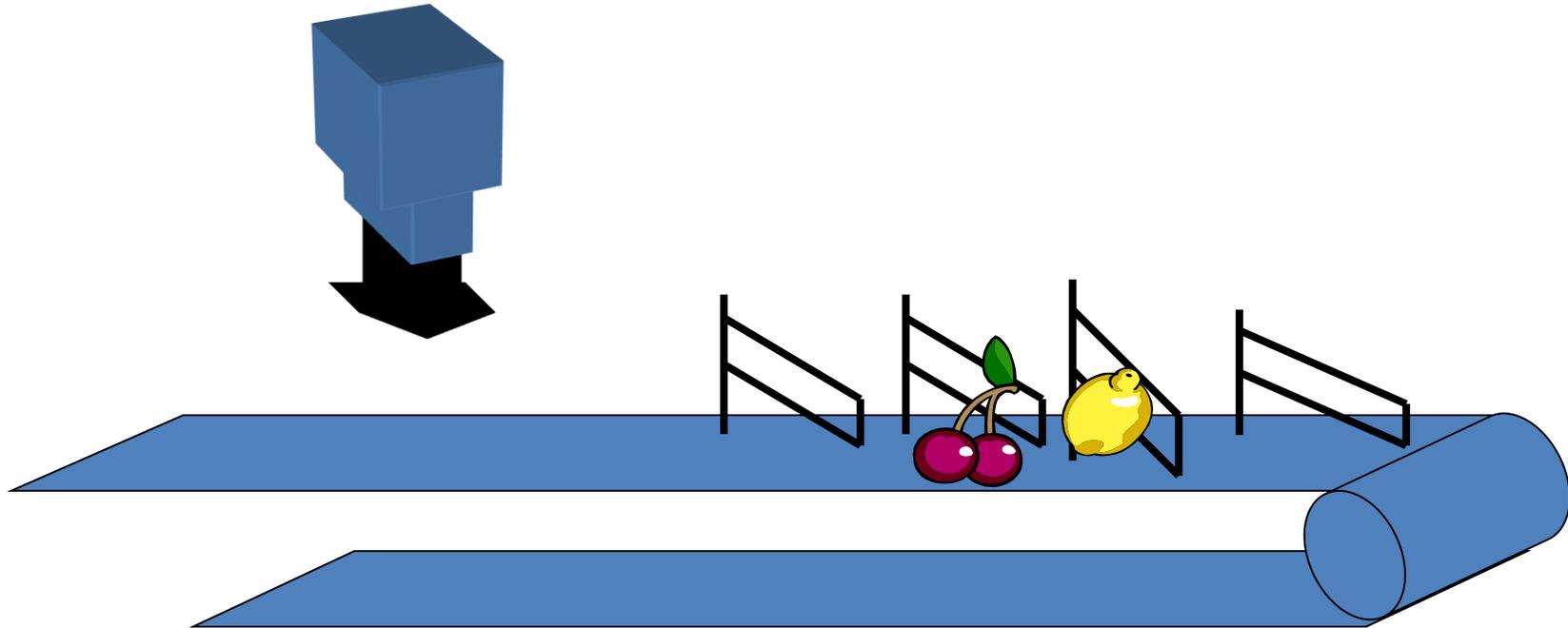
苹果

樱桃

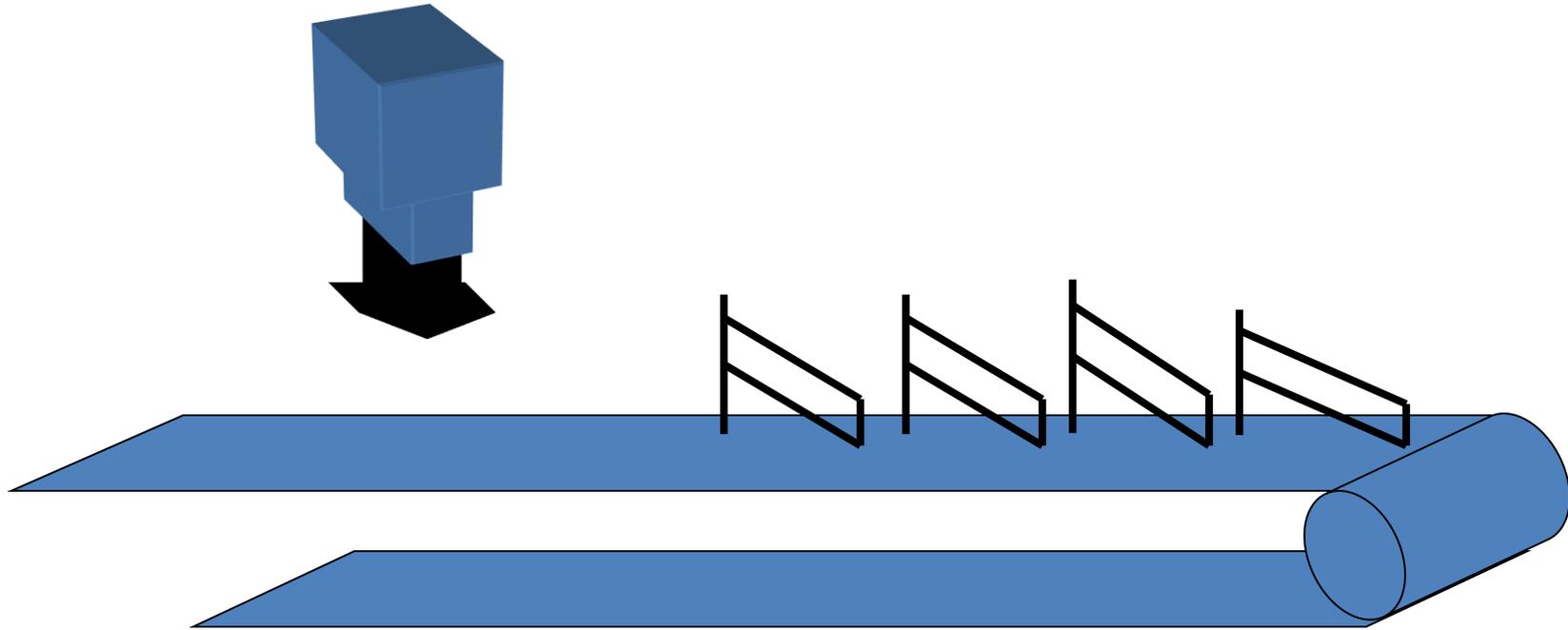
柠檬

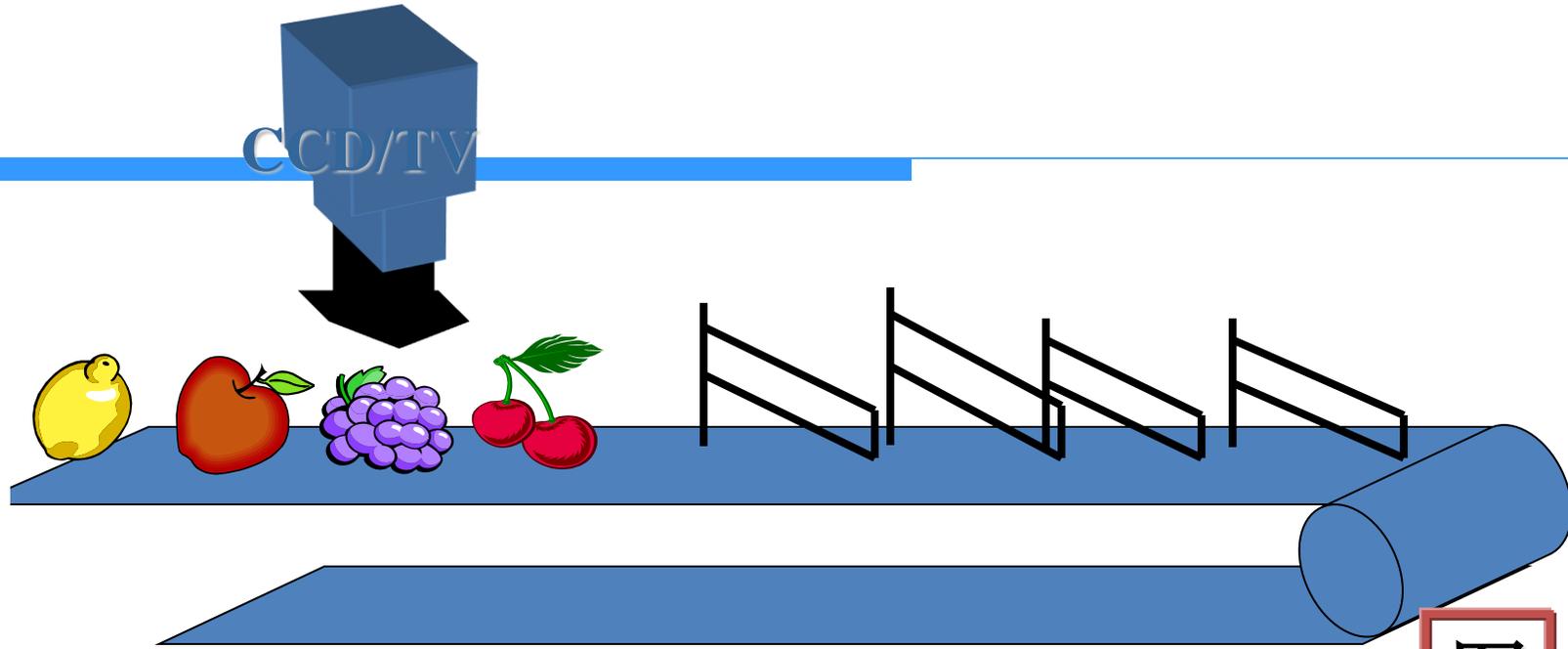
葡萄

CCD/TV



CCD/TV





苹果 樱桃 柠檬 葡萄

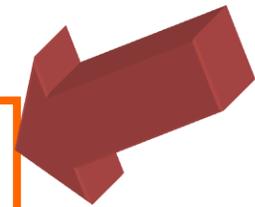
图像处理系统

特征

- ① 红色程度
- ② 直径



分类系统



■ 常用的模式序列—模式向量、模式串、模式树

➤ 模式向量

✓ 模式向量用粗体的小写字母表示，如 \mathbf{x} , \mathbf{y} 形式如下：

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{pmatrix}$$

其中每一个 x_i 代表第 i 个描述子， n 是这种描述子的数量

➤ 模式向量举例

假设我们想描述三种蝴蝶花（多毛的、维吉尼亚、多色的）通过测量它们花瓣的宽度和长度。这里涉及一个两维的模式向量：

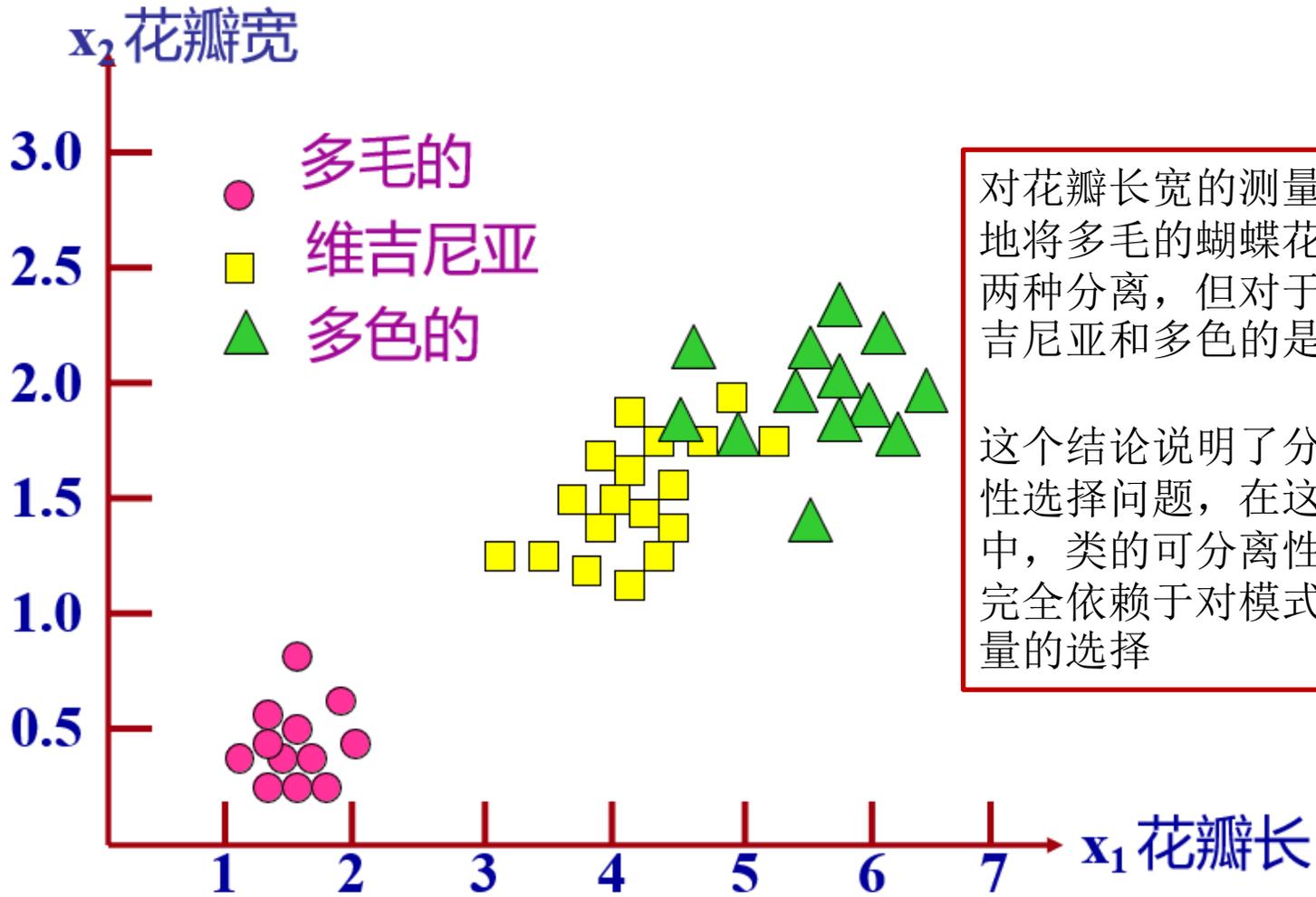
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

其中 x_1 、 x_2 分别对应花瓣的长和宽

三种模式类用 w_1 、 w_2 、 w_3 表示

由于所有的花瓣在宽和长上都有某种程度的变化，所以描述这些花瓣的模式向量也将有变化，不仅在不同的类之间，而且也在类的内部

在这种情况下每一种花变成二维欧几里德空间的一个点



对花瓣长宽的测量，成功地将多毛的蝴蝶花与其它两种分离，但对于分离维吉尼亚和多色的是失败的

这个结论说明了分类的特性选择问题，在这个问题中，类的可分离性的程度，完全依赖于对模式尺寸测量的选择



统计模式识别

- 统计模式识别——决策论法
 - 统计模式识别基本概念与过程
 - 决策论法的基本概念
 - 最小距离分类器
 - 相关匹配
 - 贝叶斯分析
 - 特征的抽取与选择

■ 统计模式识别基本概念

- 假定我们所处理的模式每一个样本都表示为N维特征矢量

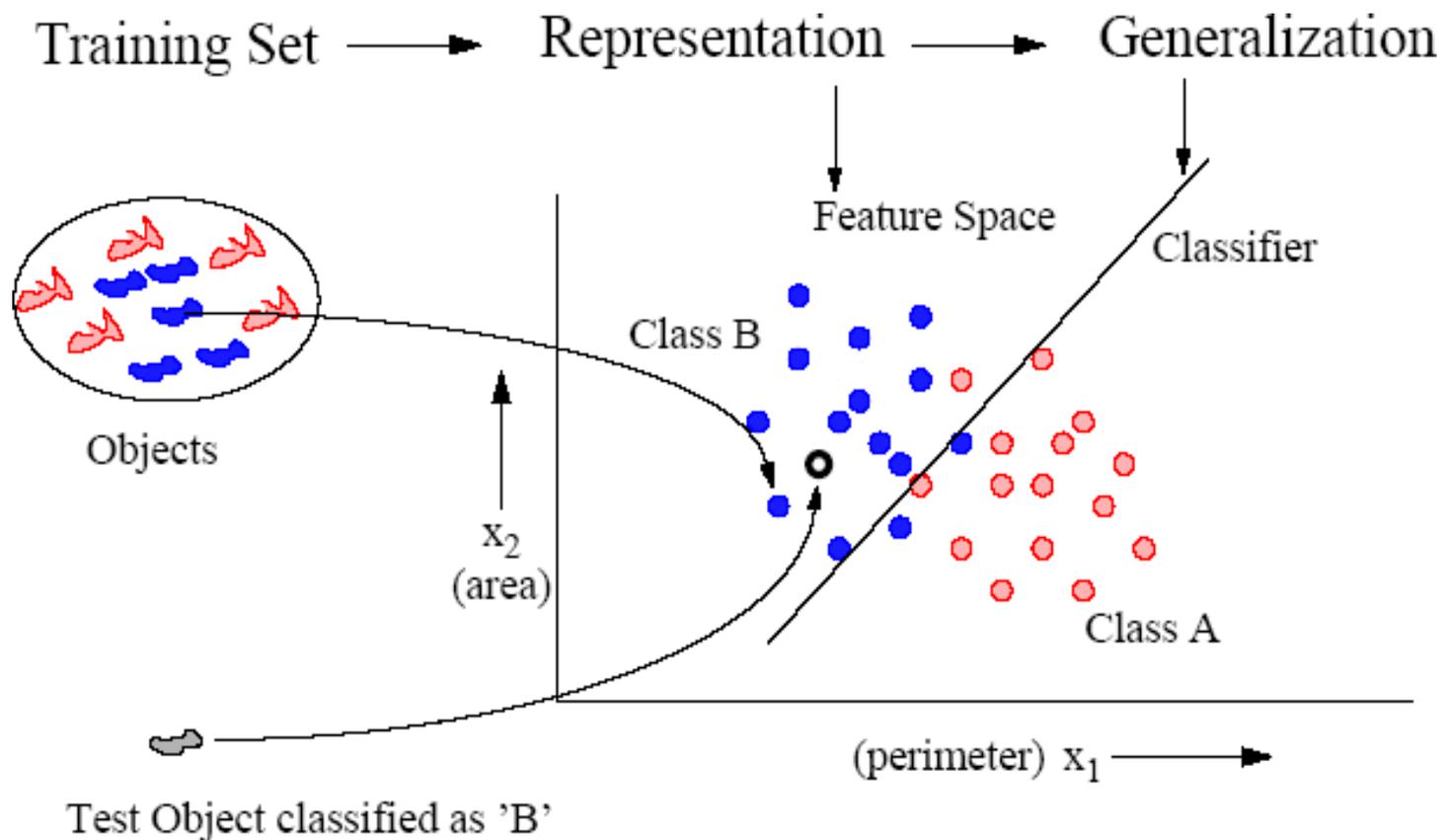
$$\vec{x} = (x_1, x_2, \dots, x_N)'$$

- 将样本表示为N维特征矢量空间中的一个点，在特征空间中研究不同样本归属不同类别的问题
- 假设我们要把一个样本集合分成M类：

$$\{\vec{x}_1, \vec{x}_2, \dots\} \quad \omega_1, \omega_2, \dots, \omega_M$$

- 在N维空间如何找到一个正确的子空间划分
- 对样本 x 进行分类 - 统计决策理论

■ 统计模式识别基本过程



■ 决策论法的基本概念

➤ 决策论识别法的定义

设：模式向量： $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$,

对于： M 个模式类 w_1, w_2, \dots, w_M ,

寻找 M 个决策函数 $d_1(\mathbf{x}), d_2(\mathbf{x}), \dots, d_M(\mathbf{x})$, 具有这样的特性：如果模式 \mathbf{x} 属于模式类 w_i , 那么：

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, \dots, M; j \neq i$$

换句话说, 如果一个未知模式 \mathbf{x} 属于第 i 个模式类, 把 \mathbf{x} 代入所有的决策函数, $d_i(\mathbf{x})$ 的数值最大。

➤ 决策边界的定义

- ✓ 对于模式 \mathbf{x} ，如果决策函数值有：

$$\mathbf{d}_i(\mathbf{x}) - \mathbf{d}_j(\mathbf{x}) = 0$$

此 \mathbf{x} 值，被称为 w_i 与 w_j 的决策边界。

- ✓ 通常用一个单一的函数标识两个类之间的决策边界，定义为：

$$\mathbf{d}_{ij}(\mathbf{x}) = \mathbf{d}_i(\mathbf{x}) - \mathbf{d}_j(\mathbf{x}) = 0$$

如果 $\mathbf{d}_{ij}(\mathbf{x}) > 0$ \mathbf{x} 属于类 w_i

如果 $\mathbf{d}_{ij}(\mathbf{x}) < 0$ \mathbf{x} 属于类 w_j

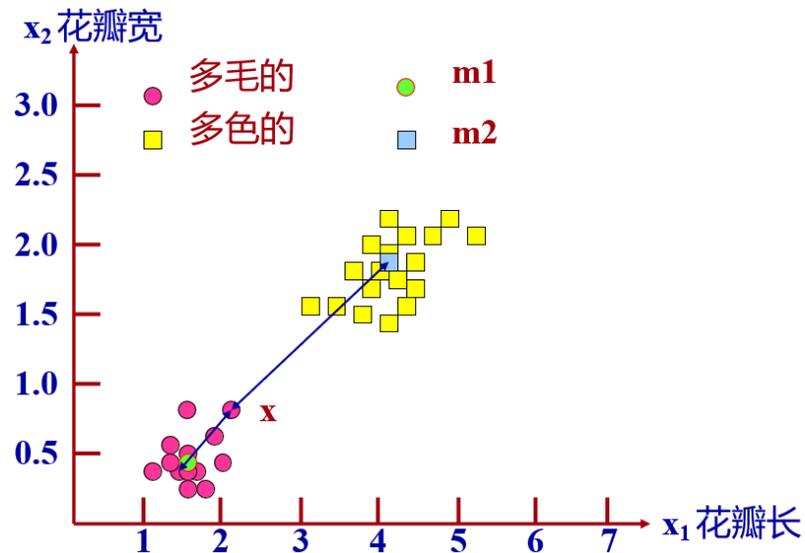
■ 最小距离分类器

1) 算法思想:

- ✓ 对于M个模式类 w_j $j = 1, 2, \dots, M$
为每一个模式类确定一个原形模式向量 m_j
- ✓ 对于一个未知模式向量 x , 如果 x 与 m_j 的距离最小, 就称 x 属于 w_j 。

■ 最小距离分类器(以蝴蝶花的例子为例)

- 〈1〉 为多色 (w_1) 和多毛 (w_2) 的两种蝴蝶花, 确定两个原形 (或称模板) m_1 和 m_2
- 〈2〉 对于一个未知模式向量 x , 判断 x 与 m_1 和 m_2 的距离, 如果与 m_1 的距离小于与 m_2 的距离, 则 x 属于 w_1 , 否则属于 w_2 。





■ 最小距离分类器

2) 最小距离分类器定义:

〈1〉 计算模式类 w_j 的原形向量:

$$\mathbf{m}_j = 1/N_j \sum_{\mathbf{x} \in w_j} \mathbf{x} \quad j = 1, 2, \dots, M$$

其中 N_j 是属于模式类 w_j 的模式向量的个数。
通过计算已知属于 w_j 的模式向量的各分量的均值,
得到原形模式向量 \mathbf{m}_j

〈2〉 计算 \mathbf{x} 与 \mathbf{m}_j 的距离

$$d_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\| \quad j = 1, 2, \dots, M$$

其中 $\|\mathbf{a}\| = (\mathbf{a}^T \mathbf{a})^{1/2}$ 是欧几里德范式 (平方和开方)

〈3〉 决策

如果, $d_i(\mathbf{x}) = \min(d_j(\mathbf{x})) \quad j = 1, 2, \dots, M$

就说: \mathbf{x} 属于 w_i

改写成求最大的标准形式, 决策函数为:

$$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - 1/2 \mathbf{m}_j^T \mathbf{m}_j \quad j = 1, 2, \dots, M$$

如果, $d_i(\mathbf{x}) = \max(d_j(\mathbf{x})) \quad j = 1, 2, \dots, M$

就说: \mathbf{x} 属于 w_i

〈4〉 类 w_i 和 w_j 之间的决策边界是:

$$\begin{aligned} d_{ij}(\mathbf{x}) &= d_i(\mathbf{x}) - d_j(\mathbf{x}) \\ &= \mathbf{x}^T (\mathbf{m}_i - \mathbf{m}_j) - 1/2 (\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m}_j) = 0 \end{aligned}$$

举例:

多色的和多毛的蝴蝶花, 用 w_1 和 w_2 分别表示, 有简单的原形向量

$$\mathbf{m}_1 = (4.4, 1.3)^T$$

$$\mathbf{m}_2 = (1.5, 0.3)^T$$

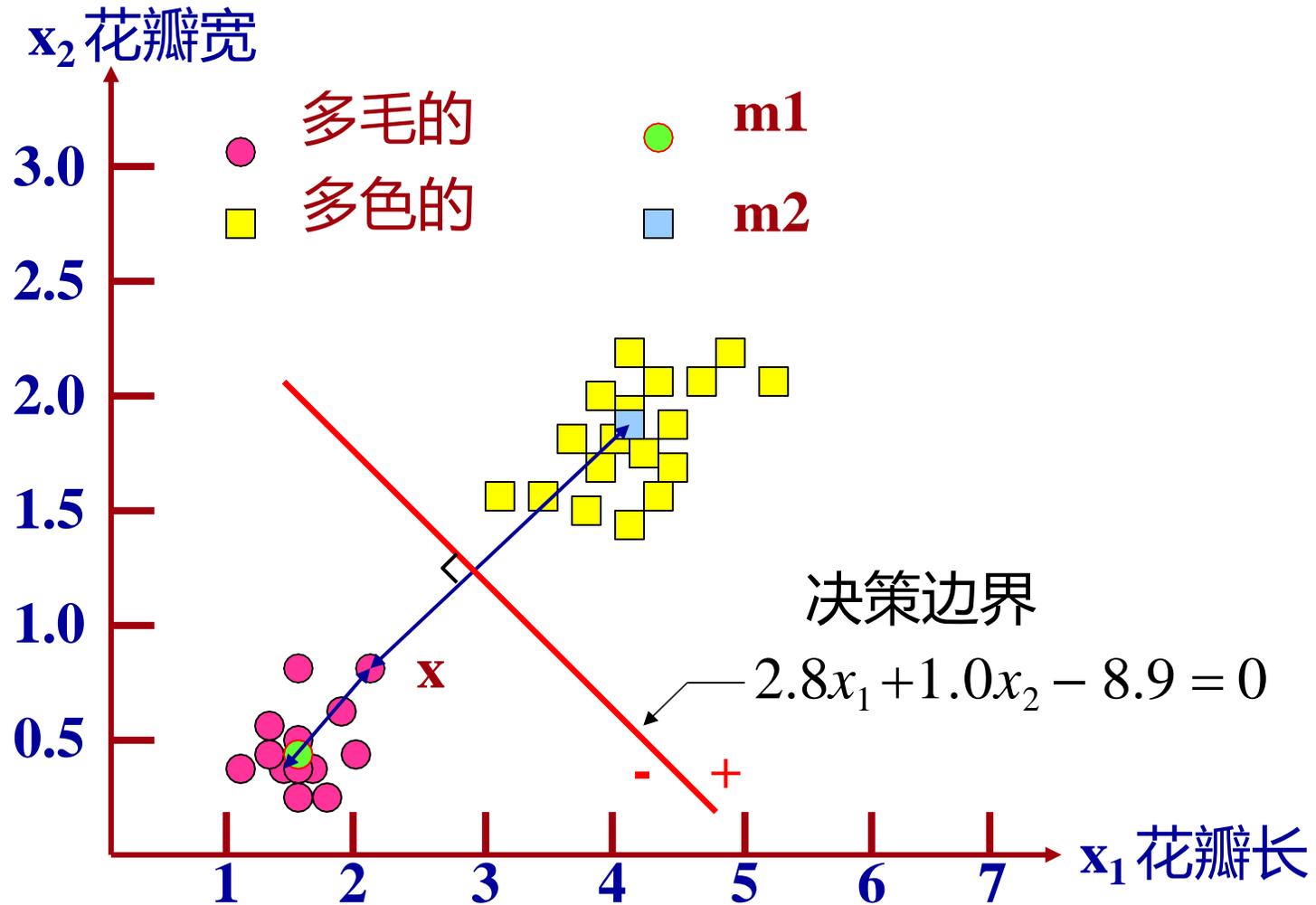
决策函数是:

$$\mathbf{d}_1(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_1 - 1/2 \mathbf{m}_1^T \mathbf{m}_1 = 4.3x_1 + 1.3x_2 - 10.1$$

$$\mathbf{d}_2(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_2 - 1/2 \mathbf{m}_2^T \mathbf{m}_2 = 1.5x_1 + 0.3x_2 - 1.17$$

决策边界的等式:

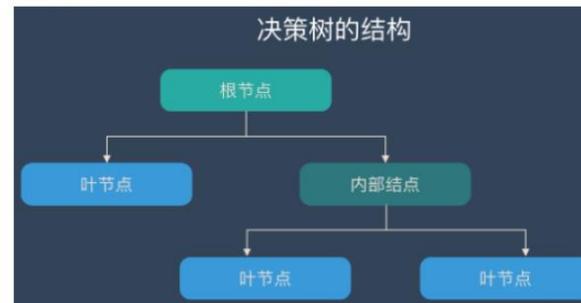
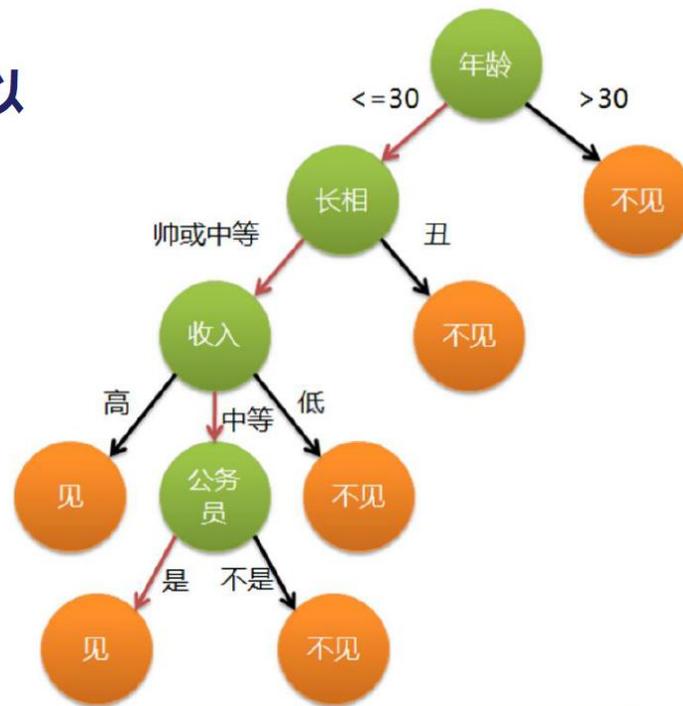
$$\mathbf{d}_{12}(\mathbf{x}) = \mathbf{d}_1(\mathbf{x}) - \mathbf{d}_2(\mathbf{x}) = 2.8x_1 - 1.0x_2 - 8.9 = 0$$



决策树(Decision Tree) 是一种数据结构，可以用来分类和回归，决策树是数据结构，但构建决策树有一系列的算法，决策树的核心之一就是利用算法构建最佳的决策树，以达到在训练数据和测试数据都表现优秀的效果

决策树的构建和人类的思维过程非常的类似

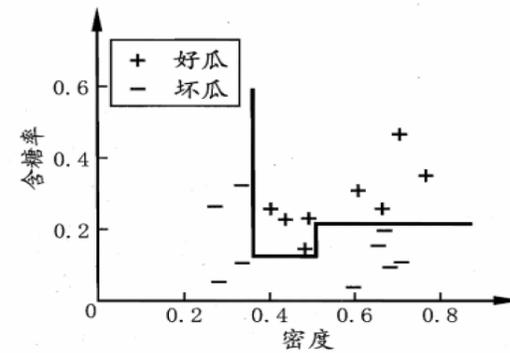
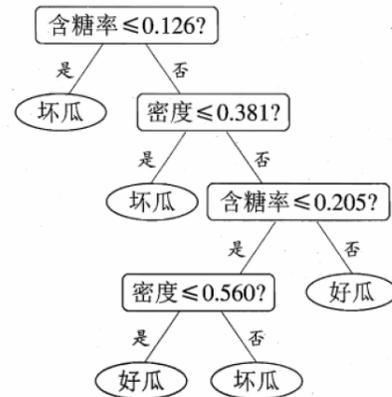
- 1 母亲：给你介绍个对象。
- 2 女儿：年纪多大了？
- 3 母亲：26。
- 4 女儿：长的帅不帅？
- 5 母亲：挺帅的。
- 6 女儿：收入高不？
- 7 母亲：不算很高，中等情况。
- 8 女儿：是公务员不？
- 9 母亲：是，在税务局上班呢。
- 10 女儿：那好，我去见见。



决策树的轴平行特性

- 若我们把每个属性视为坐标空间中的一个坐标轴，则 d 个属性描述的样本就对应了 d 维空间中的一个数据点，对样本分类则意味着在这个坐标空间中寻找不同类样本之间的分类边界。
- 决策树所形成的分类边界有一个明显的特点：**轴平行** (axis-parallel)，即它的分类边界由若干个与坐标轴平行的分段组成。

编号	密度	含糖率	好瓜
1	0.697	0.460	是
2	0.774	0.376	是
3	0.634	0.264	是
4	0.608	0.318	是
5	0.556	0.215	是
6	0.403	0.237	是
7	0.481	0.149	是
8	0.437	0.211	是
9	0.666	0.091	否
10	0.243	0.267	否
11	0.245	0.057	否
12	0.343	0.099	否
13	0.639	0.161	否
14	0.657	0.198	否
15	0.360	0.370	否
16	0.593	0.042	否
17	0.719	0.103	否



对应的决策树和分类边界

支持向量机 (SVM)

在样本空间中，划分超平面可通过如下线性方程来描述：

$$\mathbf{w}^T \mathbf{x} + b = 0$$

其中 \mathbf{w} 为法向量； b 为位移项。

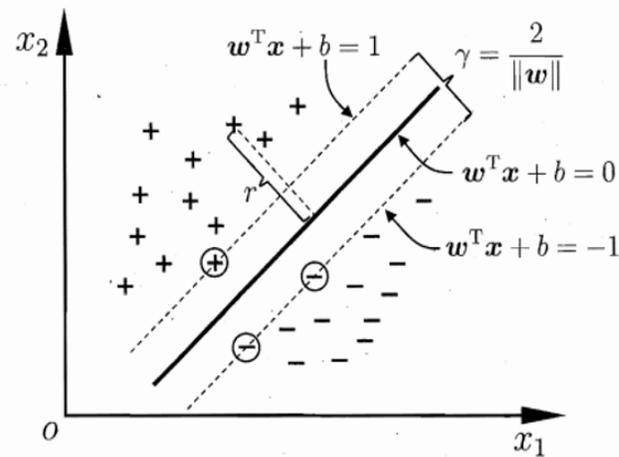
样本空间中任意点 \mathbf{x} 到超平面 (\mathbf{w}, b) 的距离可写为：

$$r = \frac{|\mathbf{w}^T \mathbf{x} + b|}{\|\mathbf{w}\|}$$

假设超平面 (\mathbf{w}, b) 能将训练样本正确分类，

即对于 $(x_i, y_i) \in D$ ，令

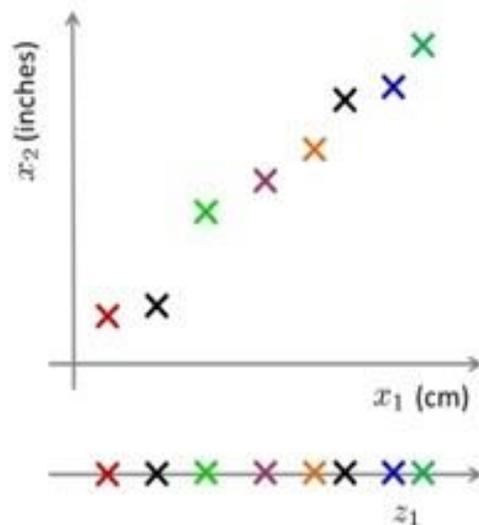
$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq +1, & y_i = +1 ; \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1, & y_i = -1 . \end{cases}$$



降维

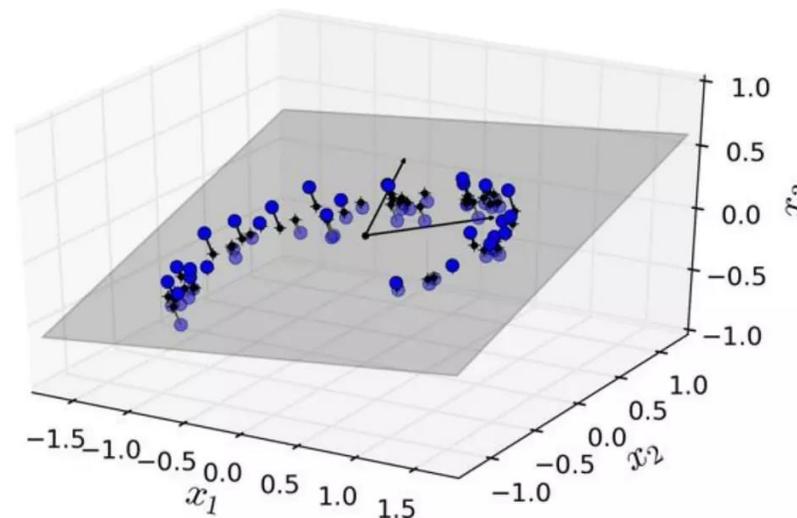
- **是什么**: 高维数据到低维数据的**映射**
- **为什么**: 提高效率, 节省空间, 去除噪声, 实现数据可视化
- **怎么做**: 根据不同的**数据样本**和**约束条件**, 选择合适的降维算法

Data Compression



Reduce data from
2D to 1D

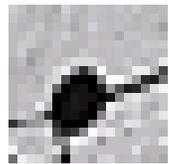
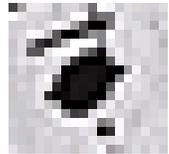
$$\begin{aligned}x^{(1)} &\rightarrow z^{(1)} \\x^{(2)} &\rightarrow z^{(2)} \\&\vdots \\x^{(m)} &\rightarrow z^{(m)}\end{aligned}$$



■ 相关匹配

(1) 相关匹配的基本思想:

- 用模板子图像直接作为模式（不是用描述子）
- 通过子图像与原图像直接进行**相关**计算，把相关计算结果作为决策函数。
- 相关计算获得最大值的位置，就被认为匹配成功。

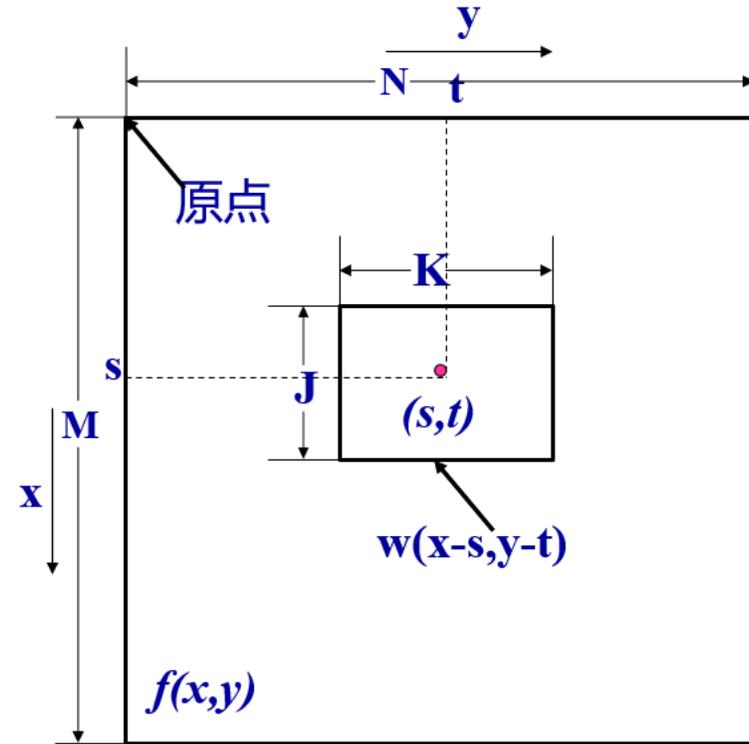


(2) 算法描述

决策函数是相关函数

$$c(s,t) = \sum_x \sum_y f(x,y)w(x-s,y-t)$$

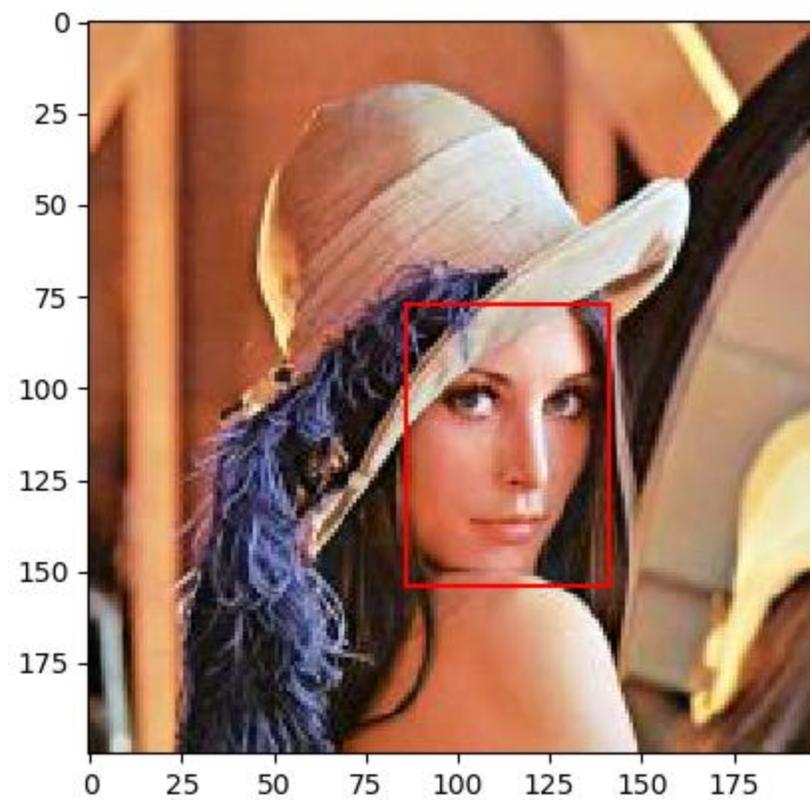
对图像的每一个点进行相关计算，只计算重叠部分



(3) 问题:

- 在边界处将失去准确性，其误差与子图像的尺寸成正比
- 相关函数对振幅的变化太敏感， $f(x,y)$ 加倍， $c(s,t)$ 也加倍。

模板lena:



搜索图:



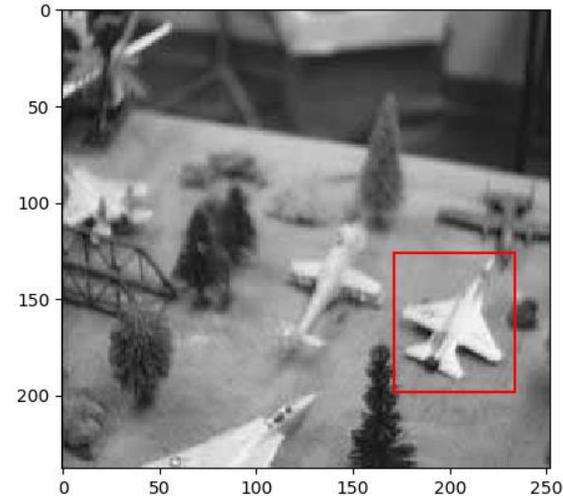
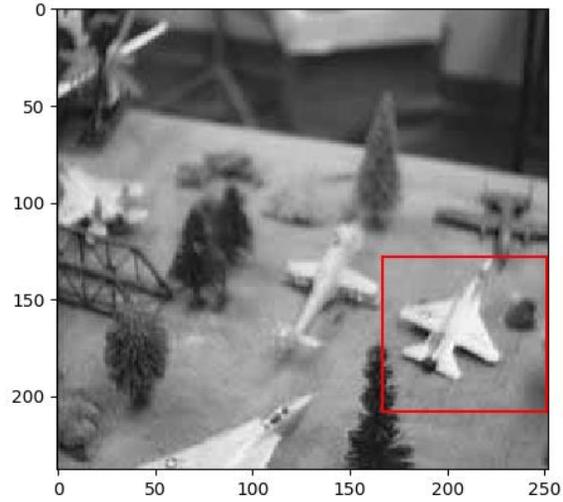
模板一:



模板2:



匹配结果如下:





■ 贝叶斯 (Bayes) 分析

- 当特征抽取时存在噪声干扰，则抽取的特征可能代表不了相应的模式，具有某种随机性
- 利用训练样本产生的不同类别的模式，每个模式难于用一个向量来表示，只能得到每一类模式的一个概率分布
- 为对此类特征进行分类，需采用统计意义上最佳的分类方法
- 统计分类方法的最基本方法之一：贝叶斯分析
- 贝叶斯分析包括：贝叶斯决策方法、贝叶斯分类器、贝叶斯估计理论、贝叶斯距离等

■ 贝叶斯 (Bayes) 分析

➤ 贝叶斯公式

设有 N 个互不相容的事件： B_1, B_2, \dots, B_N ，每个事件发生的概率为 $P(B_i)$ ；

已知仅当其中某些事件发生时，事件 A 才发生。若其中某一事件 B_i 发生， A 发生的可能性以概率 $P(A/B_i)$ 描述。

反过来，若事件 A 确实已发生，则对于 A 事件发生条件下事件 B_i 发生的概率应给与重新估计，可有贝叶斯公式：

$$P(B_i/A) = \frac{P(B_i)P(A/B_i)}{\sum_{j=1}^N P(B_j)P(A/B_j)}$$

■ 贝叶斯分析

➤ 贝叶斯分类法

设有M类图像： $\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_M$

若已知 \mathbf{B}_i 的先验概率： $\mathbf{P}(\mathbf{B}_i)$

和某观察特征值 X 的条件概率： $\mathbf{P}(X / \mathbf{B}_i)$

则由特征 X 所决定的 \mathbf{B}_i 的概率由贝叶斯公式决定：

$$P(B_i / X) = \frac{P(B_i)P(X / B_i)}{\sum_{j=1}^N P(B_j)P(X / B_j)}$$

➤ 贝叶斯分类实质：

从统计角度，在已知分类模式概率分布的条件下，判断新出现的模式所应归属的类别

Bayes 定律

在 ω_i 出现下事件 X
概率（条件）

$$P(\omega_i / X) = \frac{P(X / \omega_i)P(\omega_i)}{\sum_{j=1}^m P(X / \omega_j)P(\omega_j)}$$

事件 X 出现时 ω_i 出现
的概率（后验）

事件 ω_j 出
现的概率
（先验）

Bayes 决策法

- ω_i - 要识别的 m 类别中第 i 个模式
- X - 特征向量 $[x_1, x_2, \dots, x_n]^T$

第 i 个模式出现的
先验概率

$$P(\omega_i / X) = \frac{P(X / \omega_i)P(\omega_i)}{\sum_{i=1}^m P(X / \omega_i)P(\omega_i)}$$

特征向量 X 出现时归属
模式类别 ω_i 的概率

属于 ω_i 模式的 X 状
态的条件概率



$$\text{当 } P(\omega_i / X) = \max_j [P(\omega_j / X)]$$

Bayes 决策准则 X 归属 ω_i 类可能性最大, 即该特征向量 X 归属于 ω_i 类

■ 贝叶斯分析

➤ 贝叶斯分类法 - 两个类别的分类

如果某观察特征值 X 使 $P(B_1/X) > P(B_2/X)$, 则表明在 X 特征值出现的情况下, 类别 B_1 的可能性较高

贝叶斯分类法则:

若 $P(B_1/X) > P(B_2/X)$, 则决策 B_1

反之, 若 $P(B_1/X) < P(B_2/X)$, 则决策 B_2

考察贝叶斯公式, 不同模式间重要的区别是其分子项, 可有另一等价决策法则为:

若 $P(X/B_1)P(B_1) > P(X/B_2)P(B_2)$, 则决策 B_1

反之, 若 $P(X/B_1)P(B_1) < P(X/B_2)P(B_2)$, 则决策 B_2

■ 贝叶斯分析

➤ 贝叶斯分类法 - 两个类别的分类 (例)

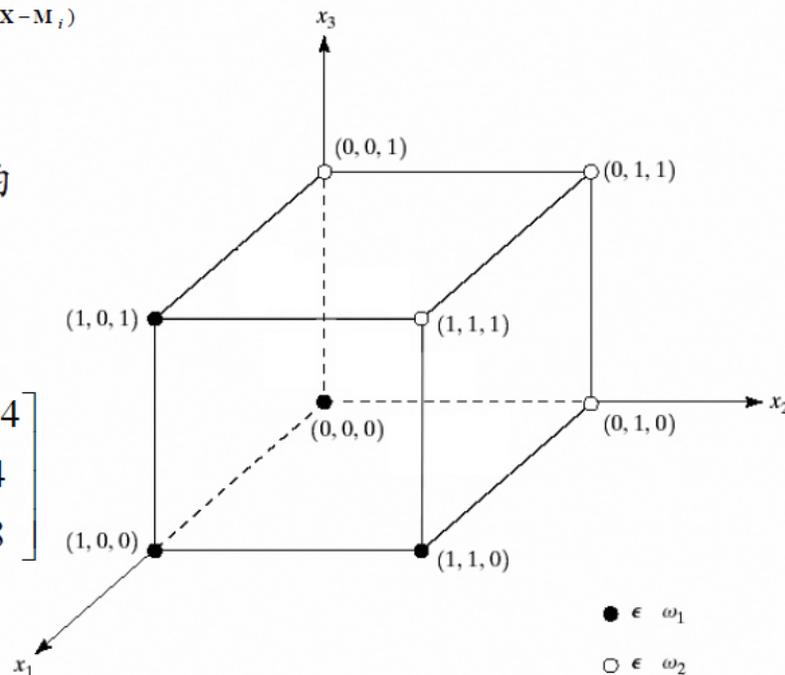
在三维空间分布的两个模式类 ω_1 和 ω_2 , 假定每个类中的模式均服从高斯分布, 即有

$$P(\mathbf{X} / \omega_i) = \frac{1}{(2\pi)^{n/2} |\mathbf{C}_i|^{1/2}} e^{-\frac{1}{2}(\mathbf{X} - \mathbf{M}_i)^T \mathbf{C}_i^{-1} (\mathbf{X} - \mathbf{M}_i)}$$

即每一个模式类中变量的密度函数均由其均值和协方差矩阵唯一确定

$$\mathbf{M}_i = E_i\{\mathbf{X}\} \quad \mathbf{C}_i = E_i\{(\mathbf{X} - \mathbf{M}_i)(\mathbf{X} - \mathbf{M}_i)^T\}$$

$$\mathbf{M}_1 = \frac{1}{4} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{M}_2 = \frac{1}{4} \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix} \quad \mathbf{C}_1 = \mathbf{C}_2 = \begin{bmatrix} 8 & -4 & -4 \\ -4 & 8 & 4 \\ -4 & 4 & 8 \end{bmatrix}$$



■ 贝叶斯分析

由贝叶斯公式，可得出决策函数为：

$$P(\omega_i / \mathbf{X}) = P(\mathbf{X} / \omega_i)P(\omega_i) \quad i=1,2$$

为计算方便，采用对数形式：

$$d_i(x) = \ln[P(\omega_i / \mathbf{X})] = \ln P(\mathbf{X} / \omega_i) + \ln P(\omega_i) \quad i=1,2$$

进一步假定 $P(\omega_1) = P(\omega_2) = 1/2$ ，可有

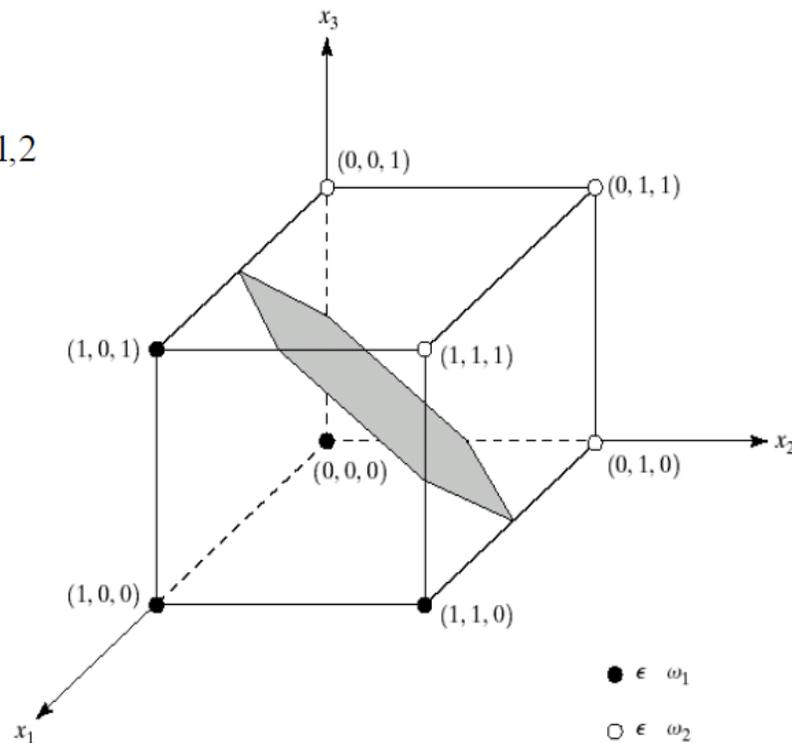
$$\vdots$$

$$d_i(x) = X^T C^{-1} M_i - \frac{1}{2} M_i^T C^{-1} M_i \quad i=1,2$$

$$d_1(x) = 4x_1 - 1.5 \quad d_2(x) = -4x_1 + 8x_2 + 8x_3 - 5.5$$

进一步得出分开两个类别的决策面为：

$$d_1(x) - d_2(x) = 8x_1 - 8x_2 - 8x_3 + 4 = 0$$



■ 贝叶斯分析

➤ 贝叶斯分类法——对多类别问题，可有：

若 $P(B_i/X) > P(B_j/X)$ 对所有 $j \neq i$ 成立，则决策 B_i

➤ 贝叶斯分类器

利用上述函数构成贝叶斯分类器如右图

实例

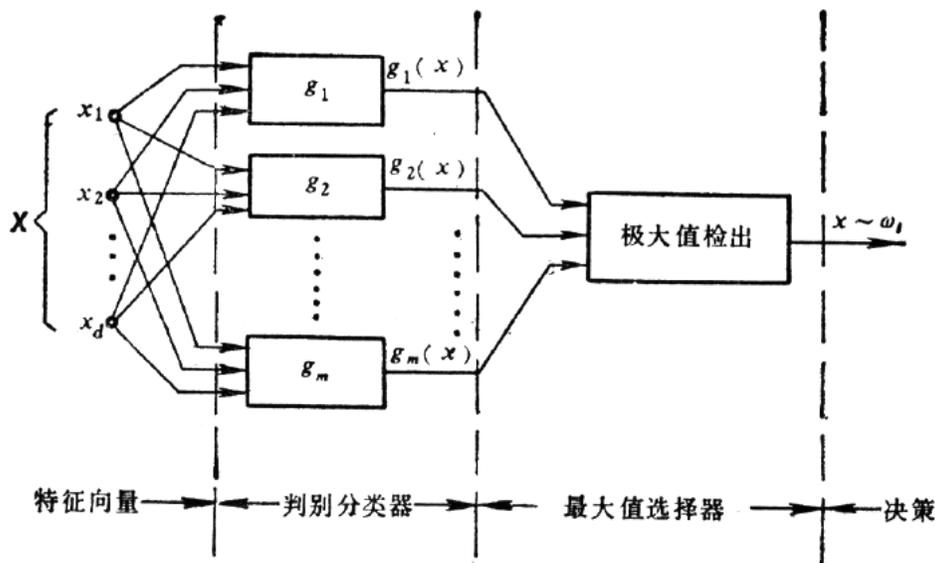


图2-13 图像分类器

朴素贝叶斯分类器训练的例子

表 4.3 西瓜数据集 3.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
测 1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	?

任务:

- 用西瓜数据集训练一个朴素贝叶斯分类器
- 对测试样本“测 1”进行分类

1. 估计类先验概率 $P(c)$: $P(\text{好瓜} = \text{是}) = \frac{8}{17} \approx 0.471$ $P(\text{好瓜} = \text{否}) = \frac{9}{17} \approx 0.529$

表 4.3 西瓜数据集 3.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是

2. 为每个属性估计条件概率 $P(x_i | c)$

➤ 离散属性:

$$P_{\text{青绿}|\text{是}} = P(\text{色泽} = \text{青绿} | \text{好瓜} = \text{是}) = \frac{3}{8} = 0.375$$

➤ 连续属性:

$$P_{\text{密度: 0.697}|\text{是}} = p(\text{密度} = 0.697 | \text{好瓜} = \text{是})$$

$$= \frac{1}{\sqrt{2\pi} \cdot 0.129} \exp\left(-\frac{(0.697 - 0.574)^2}{2 \cdot 0.129^2}\right) \approx 1.959$$

朴素贝叶斯分类器训练的例子

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
测 1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	?

$$P_{\text{青绿}|\text{是}} = P(\text{色泽} = \text{青绿} | \text{好瓜} = \text{是}) = \frac{3}{8} = 0.375$$

$$P_{\text{青绿}|\text{否}} = P(\text{色泽} = \text{青绿} | \text{好瓜} = \text{否}) = \frac{3}{9} \approx 0.333$$

$$P_{\text{蜷缩}|\text{是}} = P(\text{根蒂} = \text{蜷缩} | \text{好瓜} = \text{是}) = \frac{5}{8} = 0.375$$

$$P_{\text{蜷缩}|\text{否}} = P(\text{根蒂} = \text{蜷缩} | \text{好瓜} = \text{否}) = \frac{3}{9} \approx 0.333$$

$$P_{\text{浊响}|\text{是}} = P(\text{敲声} = \text{浊响} | \text{好瓜} = \text{是}) = \frac{6}{8} = 0.750$$

$$P_{\text{浊响}|\text{否}} = P(\text{敲声} = \text{浊响} | \text{好瓜} = \text{否}) = \frac{4}{9} \approx 0.444$$

$$P_{\text{清晰}|\text{是}} = P(\text{纹理} = \text{清晰} | \text{好瓜} = \text{是}) = \frac{7}{8} = 0.875$$

$$P_{\text{清晰}|\text{否}} = P(\text{纹理} = \text{清晰} | \text{好瓜} = \text{否}) = \frac{2}{9} \approx 0.222$$

$$P_{\text{凹陷}|\text{是}} = P(\text{脐部} = \text{凹陷} | \text{好瓜} = \text{是}) = \frac{6}{8} = 0.750$$

$$P_{\text{凹陷}|\text{否}} = P(\text{脐部} = \text{凹陷} | \text{好瓜} = \text{否}) = \frac{2}{9} \approx 0.222$$

$$P_{\text{硬滑}|\text{是}} = P(\text{触感} = \text{硬滑} | \text{好瓜} = \text{是}) = \frac{6}{8} = 0.750$$

$$P_{\text{硬滑}|\text{否}} = P(\text{触感} = \text{硬滑} | \text{好瓜} = \text{否}) = \frac{6}{9} \approx 0.667$$

$$P_{\text{密度: 0.697}|\text{是}} = p(\text{密度} = 0.697 | \text{好瓜} = \text{是})$$

$$= \frac{1}{\sqrt{2\pi} \cdot 0.129} \exp\left(-\frac{(0.697 - 0.574)^2}{2 \cdot 0.129^2}\right) \approx 1.959$$

$$P_{\text{密度: 0.697}|\text{否}} = p(\text{密度} = 0.697 | \text{好瓜} = \text{否})$$

$$= \frac{1}{\sqrt{2\pi} \cdot 0.195} \exp\left(-\frac{(0.697 - 0.496)^2}{2 \cdot 0.195^2}\right) \approx 1.203$$

$$P_{\text{含糖: 0.460}|\text{是}} = p(\text{含糖率} = 0.460 | \text{好瓜} = \text{是})$$

$$= \frac{1}{\sqrt{2\pi} \cdot 0.101} \exp\left(-\frac{(0.460 - 0.279)^2}{2 \cdot 0.101^2}\right) \approx 0.788$$

$$P_{\text{含糖: 0.460}|\text{否}} = p(\text{含糖率} = 0.460 | \text{好瓜} = \text{否})$$

$$= \frac{1}{\sqrt{2\pi} \cdot 0.108} \exp\left(-\frac{(0.460 - 0.154)^2}{2 \cdot 0.108^2}\right) \approx 0.066$$

3. 计算贝叶斯判定准则 $P(c) \prod P(x_i | c)$

$$P(\text{好瓜} = \text{是}) \times P_{\text{青绿}|\text{是}} \times P_{\text{蜷缩}|\text{是}} \times P_{\text{浊响}|\text{是}} \times P_{\text{清晰}|\text{是}} \times P_{\text{凹陷}|\text{是}} \times P_{\text{硬滑}|\text{是}} \times P_{\text{密度: 0.697}|\text{是}} \times P_{\text{含糖: 0.460}|\text{是}} \approx 0.038$$

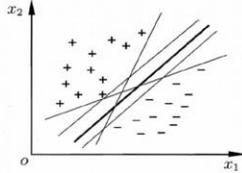
$$P(\text{好瓜} = \text{否}) \times P_{\text{青绿}|\text{否}} \times P_{\text{蜷缩}|\text{否}} \times P_{\text{浊响}|\text{否}} \times P_{\text{清晰}|\text{否}} \times P_{\text{凹陷}|\text{否}} \times P_{\text{硬滑}|\text{否}} \times P_{\text{密度: 0.697}|\text{否}} \times P_{\text{含糖: 0.460}|\text{否}} \approx 6.80 \times 10^{-5}$$

4. 判定: $0.038 > 6.80 \times 10^{-5} \rightarrow$ "好瓜"

支持向量机—(1)间隔与支持向量

支持向量机：特征空间上间隔最大的线性分类器

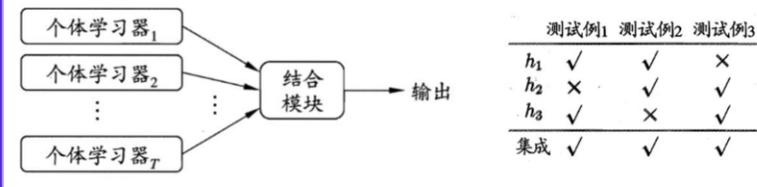
- 基本思想：给定训练样本集D，在样本空间中找到一个划分超平面，将不同类别的样本分开。



- 学习模型由简至繁有：
线性可分支持向量机、线性支持向量机、非线性支持向量机

集成学习——概论

集成学习：通过构建并结合多个学习器来完成学习任务



核心问题：如何构建并结合好而不同的个体学习器

- 个体学习器生成方式：序列化方法 (Boosting)
并行化方法 (Bagging、随机森林)
- 个体学习器好而不同：有一定的准确性要有多样性 (多样性的度量/增强)
- 多个学习器结合策略：平均法/投票法/学习法

聚类 (1) 什么是聚类

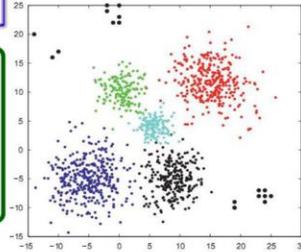
监督学习和无监督学习

- 监督学习：训练集中包括输入和输出，即输入数据已被标记。
- 无监督学习：输入数据没有被标记，也没有确定的结果。



聚类

- 是一种无监督学习，只有数据x，没有标签y
- 试图通过数据间的关系发现一定的模式
- 可以作为监督学习中稀疏特征的预处理



和分类的关系：相当于无监督的分类问题

EM算法解决的问题及算法步骤

什么是EM算法及要解决的问题

- EM算法：全称期望极大算法(expectation maximization algorithm)，是1977年由Dempster等人总结出的一种迭代算法
- 解决问题：EM算法就是含有隐变量的概率模型参数的极大似然估计法，或极大后验概率估计法

EM算法步骤

输入：观测变量数据Y，隐变量数据Z，联合分布 $P(Y, Z|\theta)$ ；
输出：模型参数 θ 。

- 选择参数初值 $\theta^{(0)}$ ，开始迭代；
- E步：记 $\theta^{(i)}$ 为第i次参数 θ 的估计值，在第i+1次迭代的E步，计算

$$Q(\theta, \theta^{(i)}) = E_Z[\log P(Y, Z|\theta) | Y, \theta^{(i)}] \\ = \sum_Z P(Z|Y, \theta^{(i)}) \log P(Y, Z|\theta)$$

- M步：求使 $Q(\theta, \theta^{(i)})$ 极大化的 θ ，确定第i+1次迭代的参数的估计值

$$\theta^{(i+1)} = \arg \max_{\theta} Q(\theta, \theta^{(i)})$$

- 重复第(2)步和第(3)步，直到收敛。





数字图像处理 (Digital Image Processing)

王正

Email: zheng_wang@seu.edu.cn

办公室: 无线谷A5415

**Image
Processing
Project Ideas**
Beginners



人工智能起源

人工智能的起源

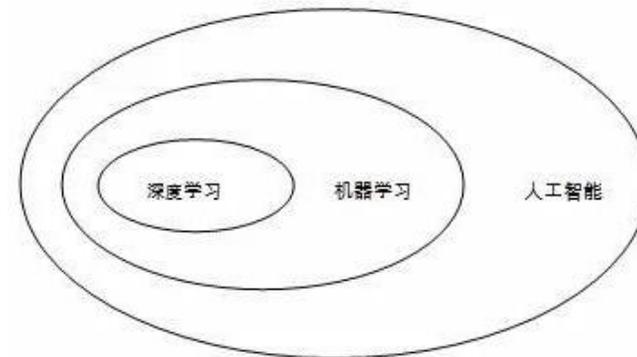
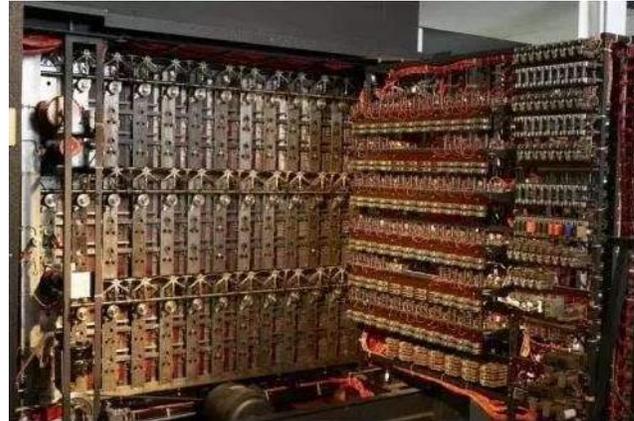
➤ 现代的人工智能

✓ 阿兰·图灵的图灵机

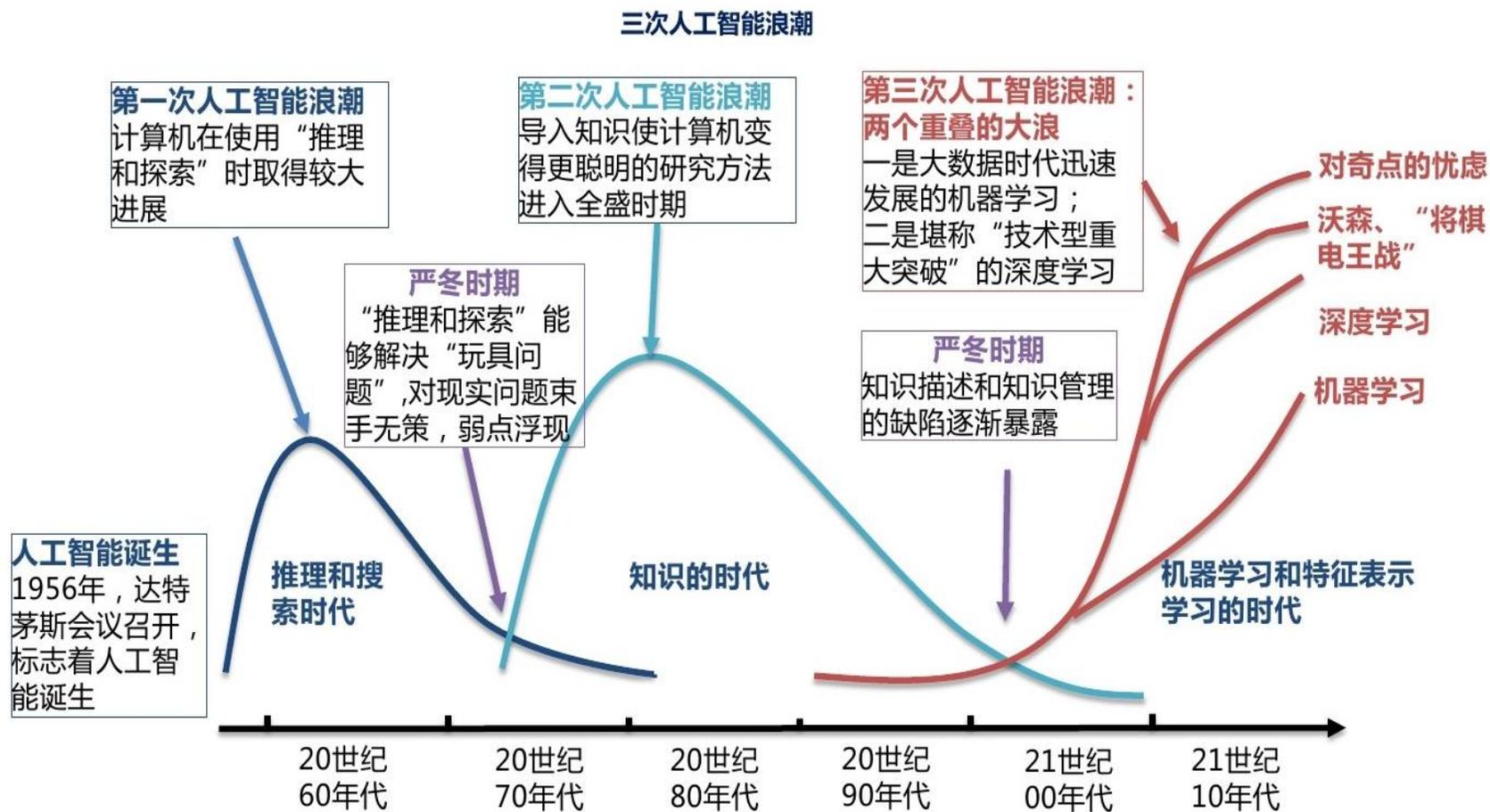
“一台机器如果能够同人类进行交互，且不被辨识出其机器身份，便可认为该机器是智能的”

✓ 人工智能会议

1956年Marvin Minsky, 约翰·麦卡锡和另两位资深科学家Claude Shannon以及Nathan Rochester组织召开了达特茅斯会议



■ 发展历史：热潮和严冬交替出现



基础知识

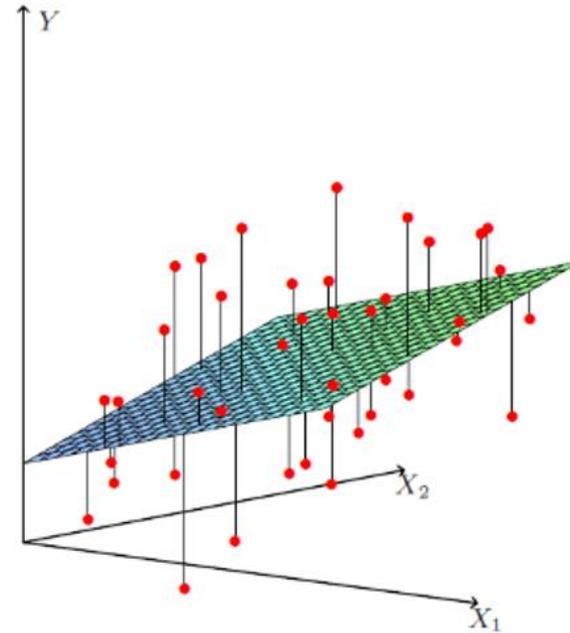
评价指标cost function $\hat{y} = w^T x$

MSE (Mean Squared Error) 叫做均方误差

$$\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

RMSE (Root Mean Squared Error) 均方根误差。

$$\sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}$$



目标：通过调整参数（权重） w 的值，使得测试数据

MSE_{test} 最小。

现实：由于预测数据的 y 未知，只能在训练集 $(X^{(train)}, y^{(train)})$

上，最小化 MSE_{train} ：



- ◆ 挑战：必须对**新的、未知的**输入目标，利用训练数据获取的模型进行决策。
- ◆ **泛化（Generalization）：在未知数据上的性能。**

在**未知数据**集上如何进行优化，最小化泛化误差？

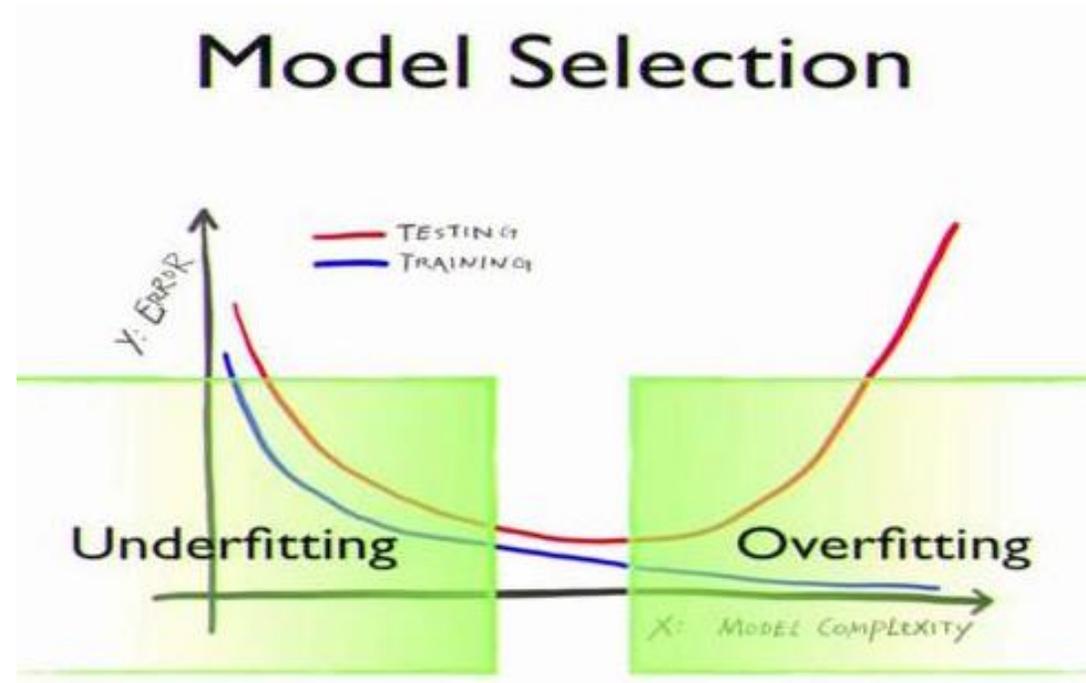
如果，训练集与数据集分布独立，我们**无事可做**。

如果，训练集与数据集相关性（或相似性）较大，我们**大有作为**。

数据泛化过程

- 训练数据集与测试数据中，样本相互独立；
- 训练与测试数据集是同分布；
- 则，训练误差与测试误差是相同的。

泛化的U-shape曲线

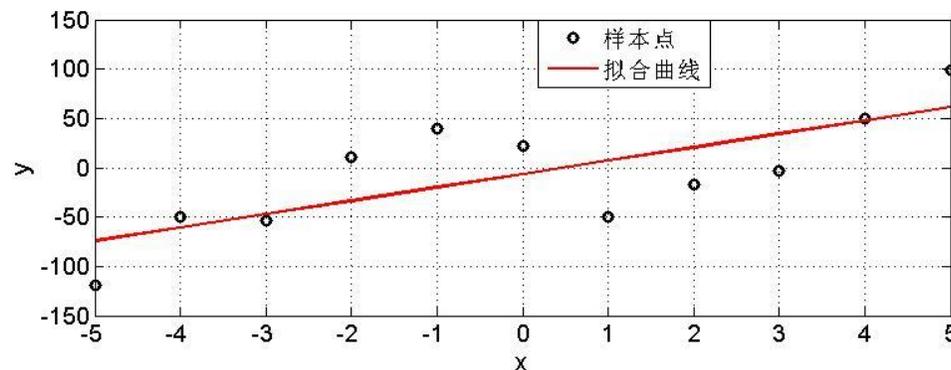


我们只能，使得训练误差尽量小（否则欠拟合），并使得训练误差与测试误差之间的鸿沟尽量小（否则过拟合）。

欠拟合

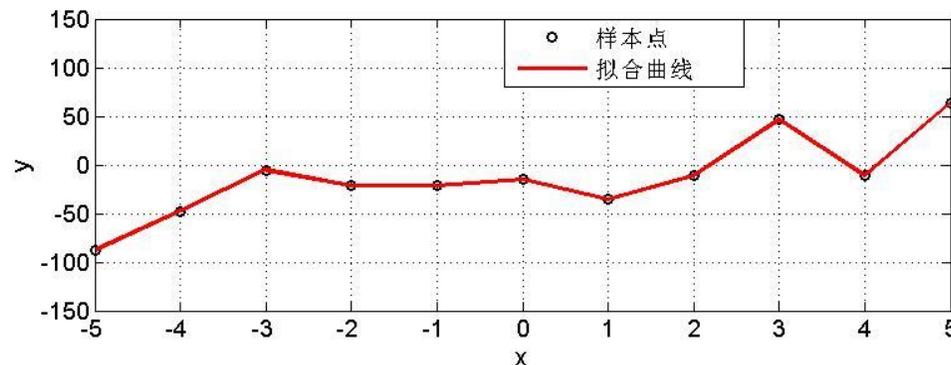
在训练集上，误差较大；

避免欠拟合：优化问题+选择合适的模型。



过拟合

在训练集上，误差小，过于符合训练数据，使得在测试集上误差较大。

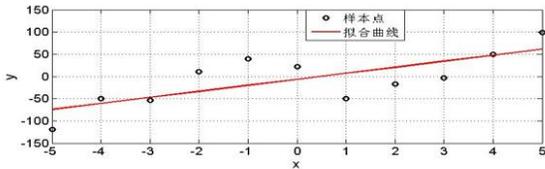


过拟合

避免过拟合方法1：增加训练数据（模型不变），若训练数据 \gg 测试数据，可减少过拟合。

避免过拟合方法2：引入正则项。

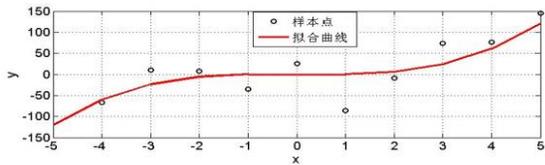
$$J(\mathbf{w}) = \text{MSE}_{\text{train}} + \lambda \mathbf{w}^T \mathbf{w}$$



$$\text{MSE}_{\text{train}} \uparrow \quad \lambda \mathbf{w}^T \mathbf{w} \downarrow$$

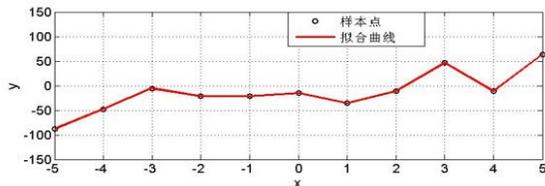
超参数

➤通过控制某些超参数，来控制一个学习系统的性能



$$\text{MSE}_{\text{train}} \quad \lambda \mathbf{w}^T \mathbf{w}$$

➤在学习系统中，参数是用来**学习的**，超参数是用来**调控的**



$$\text{MSE}_{\text{train}} \downarrow \quad \lambda \mathbf{w}^T \mathbf{w} \uparrow$$

验证集

- 与训练集、测试集并列，用来寻找学习系统最优的超参数设置；
- 测试集不能用来调控学习系统超参数（否则就有作弊行为）；
- 通常，测试集与验证集比例为**4:1**。
- 例如，在ILSVRC12 challenge中:选取ImageNet的**1,281,167**幅图像作为训练集,验证集为**50,000**幅图像,而测试集包括**100,000**幅图像。这些图像没有交叠。



类别	验证集	测试集
是否被训练到	否	否
作用	用于调超参数，监控模型是否发生过拟合（以决定是否停止训练）	为了评估最终模型泛化能力
使用次数	多次使用，以不断调参	仅仅一次使用
缺陷	模型在一次次重新手动调参并继续训练后所逼近的验证集，可能只代表一部分非训练集，导致最终训练好的模型泛化性能不够	测试集为了具有泛化代表性，往往数据量比较大，测试一轮要很久，所以往往只取测试集的其中一小部分作为训练过程中的验证集

训练集-----学生的课本；学生 根据课本里的内容来掌握知识。

验证集-----作业，通过作业可以知道 不同学生学习情况、进步的速度快慢。

测试集-----考试，考的题是平常都没有见过，考察学生举一反三的能力。

神经网络——概论

什么是神经网络

- **人工神经网络**: 由具有**适应性的简单单元**组成的广泛并行互连的**网络**, 它的组织能够**模拟**生物神经系统对真实世界物体所做出的的交互反应。
- **神经网络学习**: 机器学习与神经网络的交叉。

神经元模型

神经元作用机制: 当一个神经元兴奋时, 会向相连的神经元发送化学物质, 从而改变这些神经元内的电位, 如果某个神经元的电位超过一个阈值, 那么它就会被激活, 向其他神经元发送化学物质。

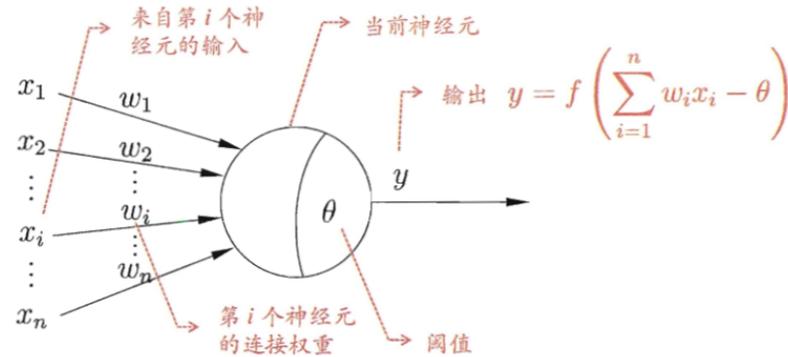
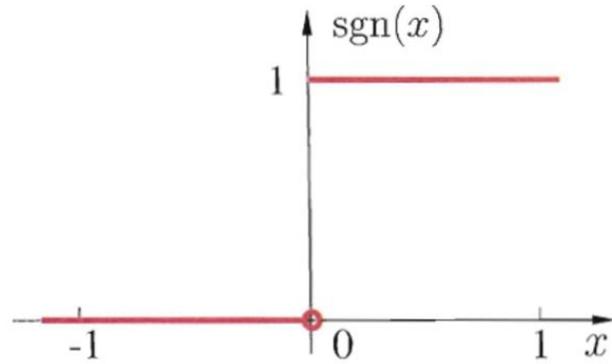


图 5.1 M-P 神经元模型

激活函数

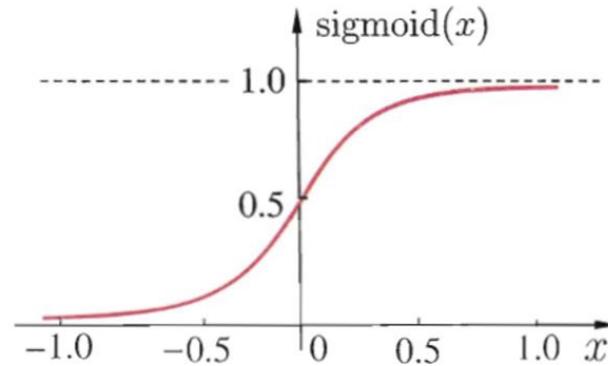
激活函数

$$\text{当 } x \neq 0 \text{ 时, } \lim_{T \rightarrow 0} \frac{1}{1+e^{-\frac{x}{T}}} = \text{sgn}(x)$$



$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0; \\ 0, & x < 0. \end{cases}$$

(a) 阶跃函数



$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

(b) Sigmoid 函数

不连续 不光滑

$$\frac{\partial \text{sigmoid}(x)}{\partial x} = \text{sigmoid}(x) (1 - \text{sigmoid}(x))$$



输出单元 (output units)

- 有时输出为**多分类**问题。
- 输出为Multinoulli分布下的Softmax输出单元

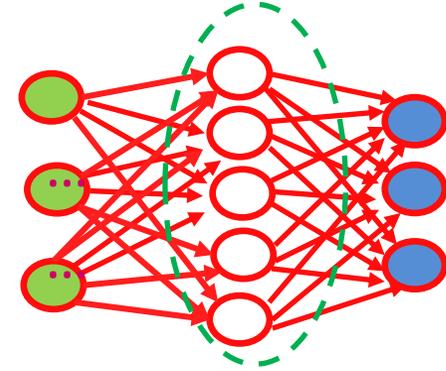
Multinoulli分布是伯努力 (Bernoulli) 分布的扩展, 随机变量输出状态为 k ($k > 2$) 个

延续Sigmoid函数的假设: $\log \tilde{p}(y = i|x) = z_i$, 将Softmax定义为

$$\text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

ReLU

➤ 隐藏层通常选用修正线性单元（Rectified Linear Units），激活函数为 $g(z) = \max\{0, z\}$ 。



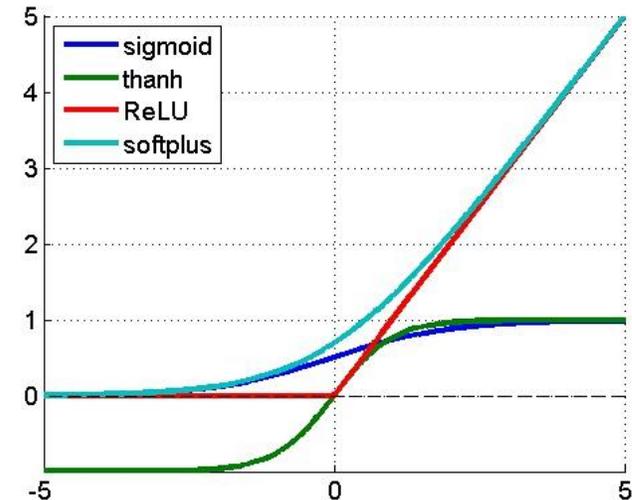
➤ ReLU

$$h = g(W^T x + b)$$

通常将 \mathbf{b} 设为**很小的正值**，保证**初始化后**函数处于激活状态。

➤ 由于与线性单元近似，所以易于优化

➤ 缺点：梯度算法在激活函数为零的时候，无法学习



感知机

- **感知机**：由两层神经元组成，输入层接收外接输入信号后传递给输出层，输出层为神经元。
- 感知机可以实现逻辑与、或、非运算。

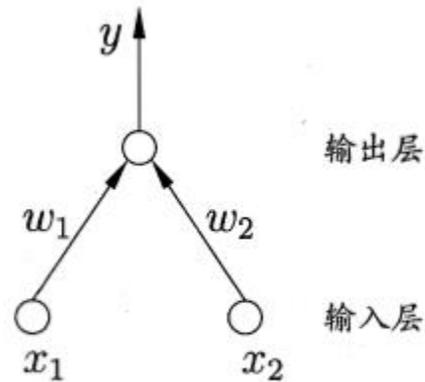


图 5.3 两个输入神经元的感知机网络结构示意图

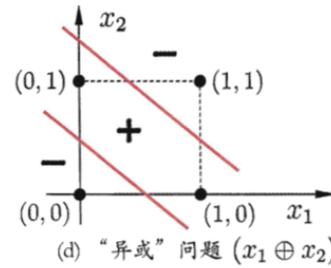
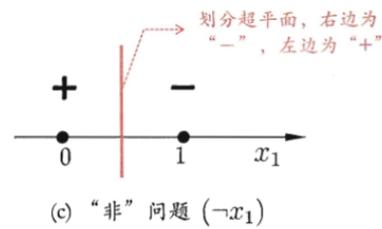
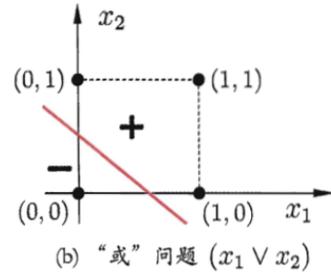
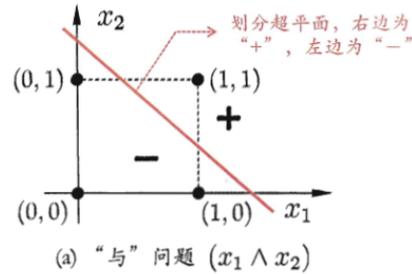
- 给定训练数据集，权重 w_i 以及阈值可以通过学习得到
- 对于训练样例 (x,y) ，若当前感知机的输出为 \hat{y} ，则进行更新

$$w_i \leftarrow w_i + \Delta w_i ,$$

$$\Delta w_i = \eta(y - \hat{y})x_i ,$$

- η 被称为学习率
- 若感知机对训练样例预测正确，则感知机不会发生变化，否则将根据错误的程度进行权重调整
- 学习能力有限！ 仅能处理线性可分问题

感知机：两层神经元构成的神经网络，无隐层



$$w_i \leftarrow w_i + \Delta w_i ,$$

$$\Delta w_i = \eta(y - \hat{y})x_i ,$$

线性可分收敛，线性不可分难以收敛。

threshold logic unit

$$x_1 \wedge x_2$$

$$x_1 \vee x_2$$

$$\neg x_1$$

$$y = f(\sum_i w_i x_i - \theta)$$

$$y = \text{sgn}(1 \cdot x_1 + 1 \cdot x_2 - 2)$$

$$y = \text{sgn}(1 \cdot x_1 + 1 \cdot x_2 - 0.5)$$

$$y = \text{sgn}(-0.6 \cdot x_1 + 0 \cdot x_2 + 0.5)$$

多层前馈神经网络

多层前馈神经网络：每层神经元与下层神经元全互连，神经元之间不存在同层连接和跨层连接

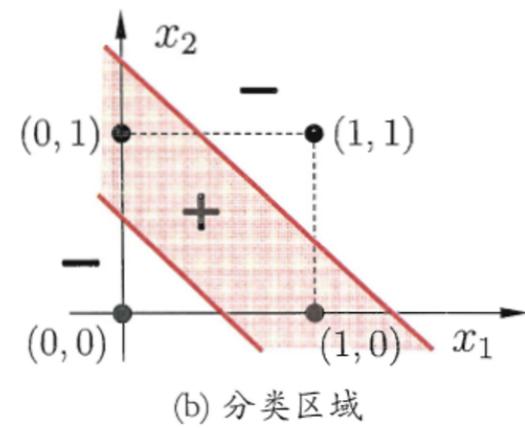
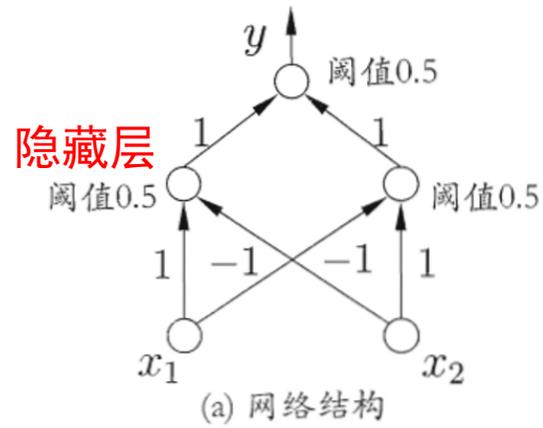
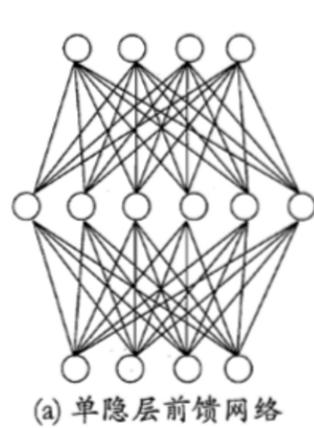


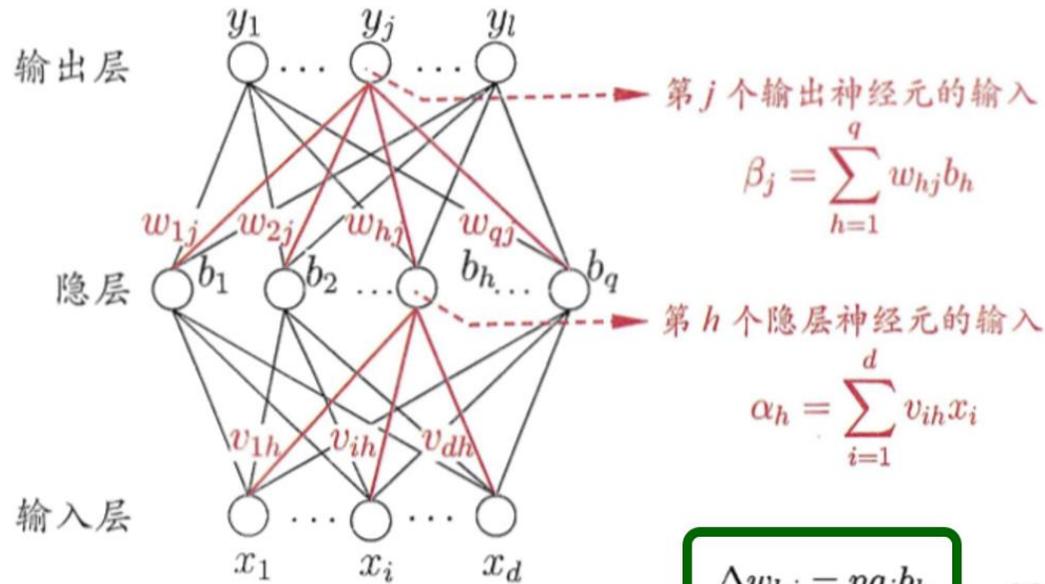
图 5.6 多层前馈神经

图 5.5 能解决异或问题的两层感知机

➤ 解决非线性可分问题，需要使用多层感知机

➤ 神经网络“学”到的东西即为权值和阈值

单隐层前馈神经网络训练：误差逆传播算法（BP）



梯度下降法求解权值、阈值

$$v \leftarrow v + \Delta v$$

$$\Delta w_{hj} = \eta g_j b_h$$

$$\Delta \theta_j = -\eta g_j$$

$$\Delta v_{ih} = \eta e_h x_i$$

$$\Delta \gamma_h = -\eta e_h$$

$$\beta_j = \sum_{h=1}^q w_{hj} b_h$$

$$\alpha_h = \sum_{i=1}^d v_{ih} x_i$$

$$E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2$$

$$\hat{y}_j^k = f(\beta_j - \theta_j)$$

$$b_h = f(\alpha_h - \gamma_h)$$

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}}$$

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}}$$

$$\frac{\partial \beta_j}{\partial w_{hj}} = b_h \quad g_j = -\frac{\partial E_k}{\partial \hat{y}_j^k} \frac{\partial \hat{y}_j^k}{\partial \beta_j} = (y_j^k - \hat{y}_j^k) \cdot \hat{y}_j^k (1 - \hat{y}_j^k)$$

$$e_h = -\frac{\partial E_k}{\partial b_h} \frac{\partial b_h}{\partial \alpha_h} = -\sum_j \frac{\partial E_k}{\partial \beta_j} \frac{\partial \beta_j}{\partial b_h} \frac{\partial b_h}{\partial \alpha_h} = \sum_j g_j \cdot \omega_{hj} \cdot b_h (1 - b_h)$$

梯度下降法

$$x_\alpha = x - \alpha \nabla f(x), \quad \forall \alpha \geq 0.$$

$$f(x_\alpha) = f(x) - \alpha \|\nabla f(x)\|^2 + o(\alpha)$$

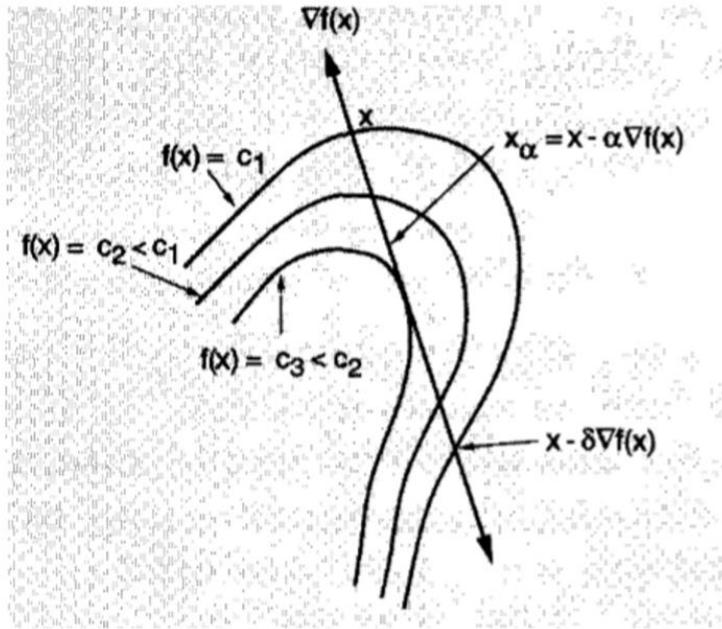


Figure 1.2.2. If $\nabla f(x) \neq 0$, there is an interval $(0, \delta)$ of stepsizes such that $f(x - \alpha \nabla f(x)) < f(x)$ for all $\alpha \in (0, \delta)$.

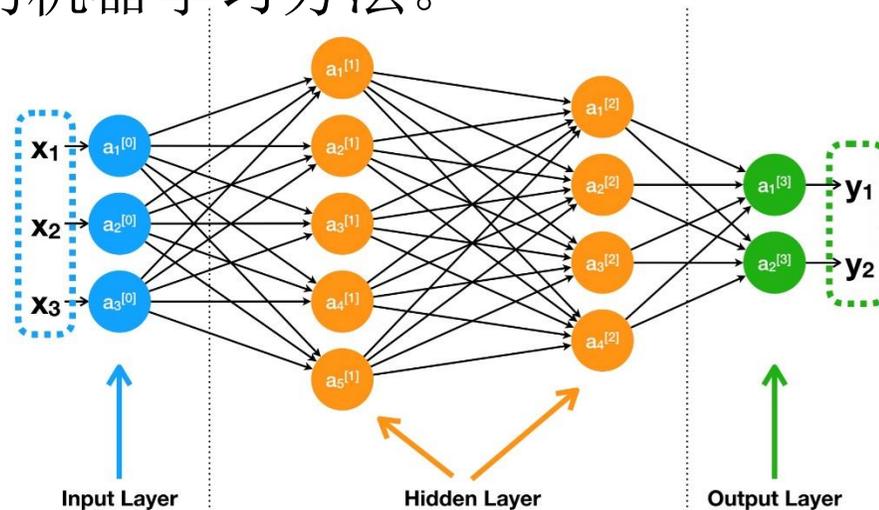
反向传播

什么是反向传播？

➤ **Back-Propagation**，一种基于梯度下降的优化方法，是多
层神经网络中**极其重要**的参数训练方法。

➤ 反向传播方法要求有对每个输入值想得到的已知输出，对
网络中所有权重计算损失函数的梯度。这个梯度会反馈给优
化算法，用来更新权值以最小化损失函数。

➤ 一种监督的机器学习方法。



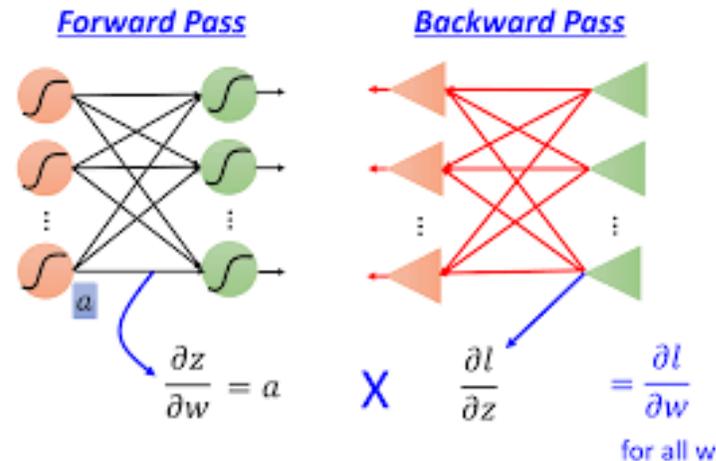
反向传播的基本思想

➤按照损失函数的梯度，按照**链式法则**，进行递归分解。

➤分为两个阶段：

1. **正向过程（激励传播）**：输入信息从输入层经隐藏层逐层计算各单元输出值。

2. **反向过程（权值更新）**：由输出误差逐层向前，计算各隐藏层单元的误差，并用此误差修正前层的权值。



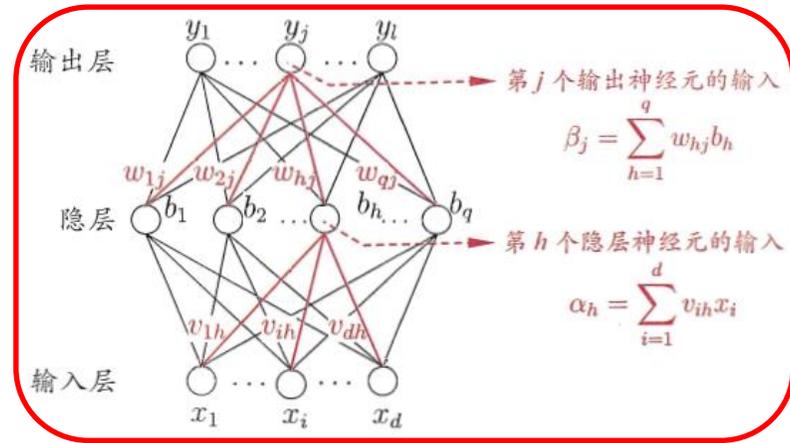
➤ 若图例中共有 $(d+l+1)q+l$ 个参数需要确定

对训练例 $(\mathbf{x}_k, \mathbf{y}_k)$, 假定神经网络的输出为 $\hat{\mathbf{y}}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$, 即

$$\hat{y}_j^k = f(\beta_j - \theta_j),$$

则网络在 $(\mathbf{x}_k, \mathbf{y}_k)$ 上的均方误差为

$$E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2.$$



BP 算法基于梯度下降 (gradient descent) 策略, 以目标的负梯度方向对参数进行调整. 对式(5.4)的误差 E_k , 给定学习率 η , 有

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}}.$$

注意到 w_{hj} 先影响到第 j 个输出层神经元的输入值 β_j , 再影响到其输出值 \hat{y}_j^k , 然后影响到 E_k , 有

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \quad (5.7)$$

根据 β_j 的定义, 显然有

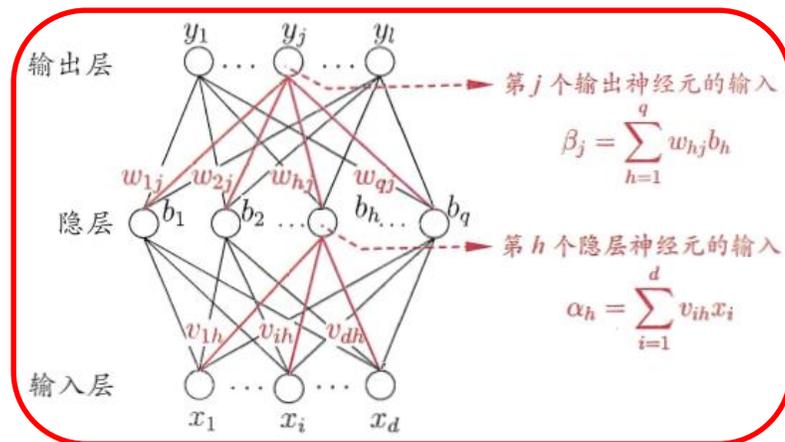
$$\frac{\partial \beta_j}{\partial w_{hj}} = b_h .$$

图 5.2 中的 Sigmoid 函数有一个很好的性质:

$$f'(x) = f(x)(1 - f(x)) ,$$

于是根据式(5.4)和(5.3), 有

$$\begin{aligned} \hat{y}_j^k &= f(\beta_j - \theta_j) \\ E_k &= \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2 \\ g_j &= -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \\ &= -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta_j) \\ &= \hat{y}_j^k (1 - \hat{y}_j^k) (y_j^k - \hat{y}_j^k) . \end{aligned} \quad (5.10)$$



将式(5.10)和(5.8)代入式(5.7), 再代入式(5.6), 就得到了BP 算法中关于 w_{hj} 的更新公式

$$\Delta w_{hj} = \eta g_j b_h . \quad (5.11)$$

类似可得

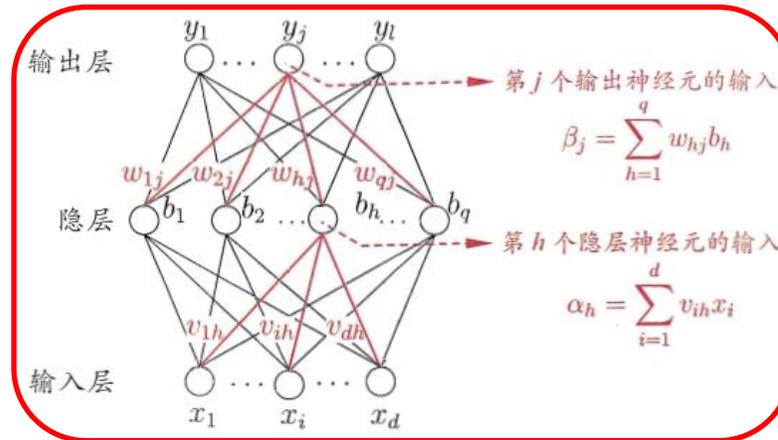
$$\Delta \theta_j = -\eta g_j ,$$

$$\Delta v_{ih} = \eta e_h x_i ,$$

$$\Delta \gamma_h = -\eta e_h ,$$

式(5.13)和(5.14)中

$$\begin{aligned} e_h &= -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} \\ &= -\sum_{j=1}^l \frac{\partial E_k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} f'(\alpha_h - \gamma_h) \end{aligned}$$



- 学习率控制着算法每一轮迭代更新中的步长, 太大容易震荡, 太小收敛速度又会过慢

单隐层前馈神经网络算法

输入: 训练集 $D = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^m$;
学习率 η .

过程:

- 1: 在(0, 1)范围内随机初始化网络中所有连接权和阈值
- 2: **repeat**
- 3: **for all** $(\mathbf{x}_k, \mathbf{y}_k) \in D$ **do**
- 4: 根据当前参数和式(5.3) 计算当前样本的输出 $\hat{\mathbf{y}}_k$;
- 5: 根据式(5.10) 计算输出层神经元的梯度项 g_j ;
- 6: 根据式(5.15) 计算隐层神经元的梯度项 e_h ;
- 7: 根据式(5.11)-(5.14) 更新连接权 w_{hj} , v_{ih} 与阈值 θ_j, γ_h
- 8: **end for**
- 9: **until** 达到停止条件

输出: 连接权与阈值确定的多层前馈神经网络

$$g_j = (y_j^k - \hat{y}_j^k) \hat{y}_j^k (1 - \hat{y}_j^k)$$

$$e_h = b_h (1 - b_h) \sum_j g_j \omega_{hj}$$

$$v \leftarrow v + \Delta v$$

$$\Delta w_{hj} = \eta g_j b_h$$

$$\Delta \theta_j = -\eta g_j$$

$$\Delta v_{ih} = \eta e_h x_i$$

$$\Delta \gamma_h = -\eta e_h$$

标准BP算法

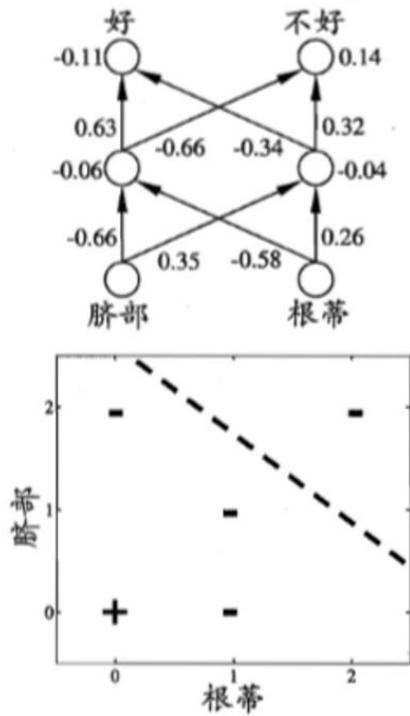
累积BP算法

BP算法目标是最小化累计误差

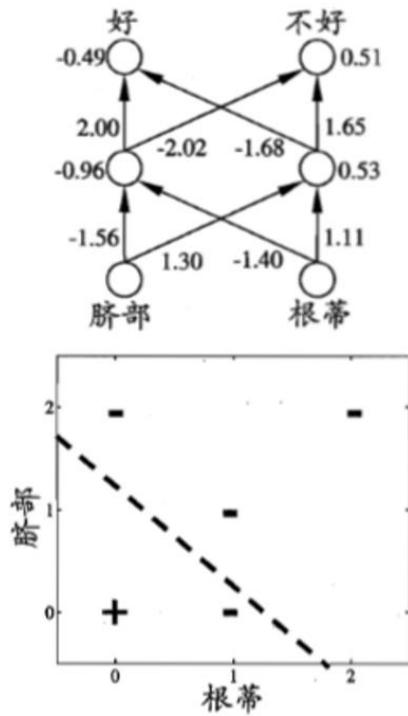
$$E = \frac{1}{m} \sum_{k=1}^m E_k$$

目标函数

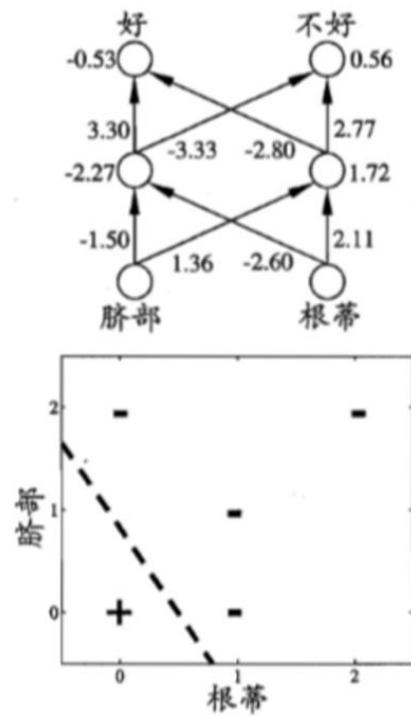
示例



(a) 第 25 轮



(b) 第 50 轮



(c) 第 100 轮

单隐层前馈神经网络算法

- **标准BP算法**: 更新频繁, 不同样本训练效果可能抵消, 迭代次数多。
- **累积BP算法**: 更新频率低, 累积误差下降到一定程度, 进一步下降缓慢。

重要结论: 只需一个包含足够多神经元的隐层, 多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数。**问题**: 神经元个数的设置

抗过拟合策略

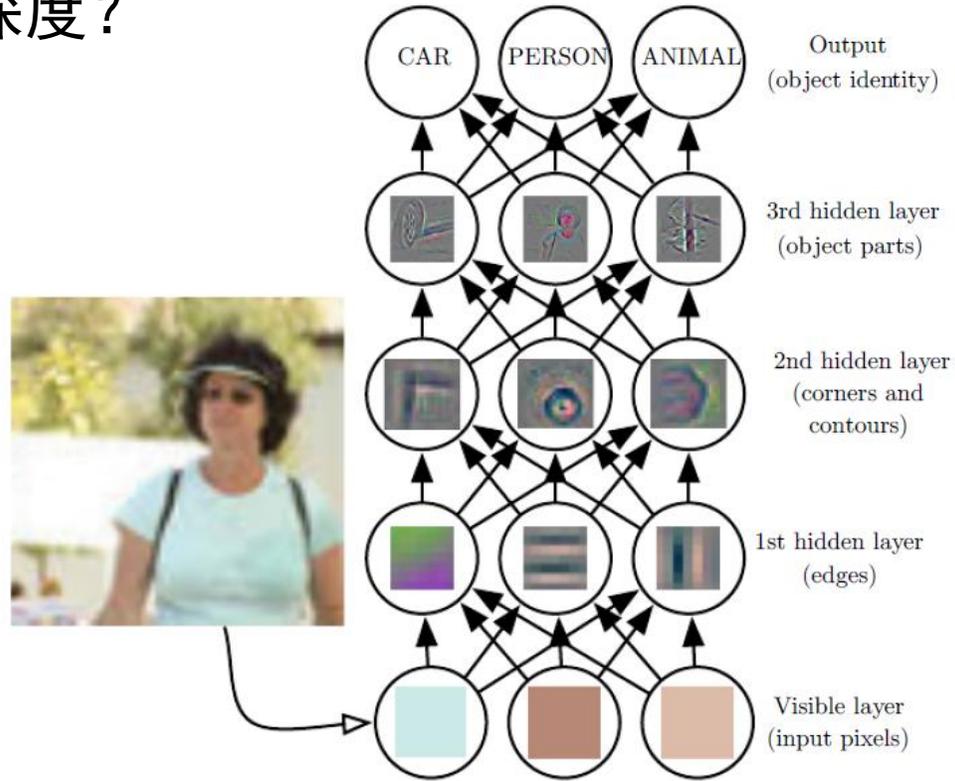
- **早停**: 将数据集分为训练集和验证集, 验证集只估计误差, 若训练集误差下降而验证集误差上升则停止训练。
- **正则化**: 在误差目标函数中增加一个描述网络复杂度的部分。

$$E = \lambda \frac{1}{m} \sum_{k=1}^m E_k + (1 - \lambda) \sum_i w_i^2$$

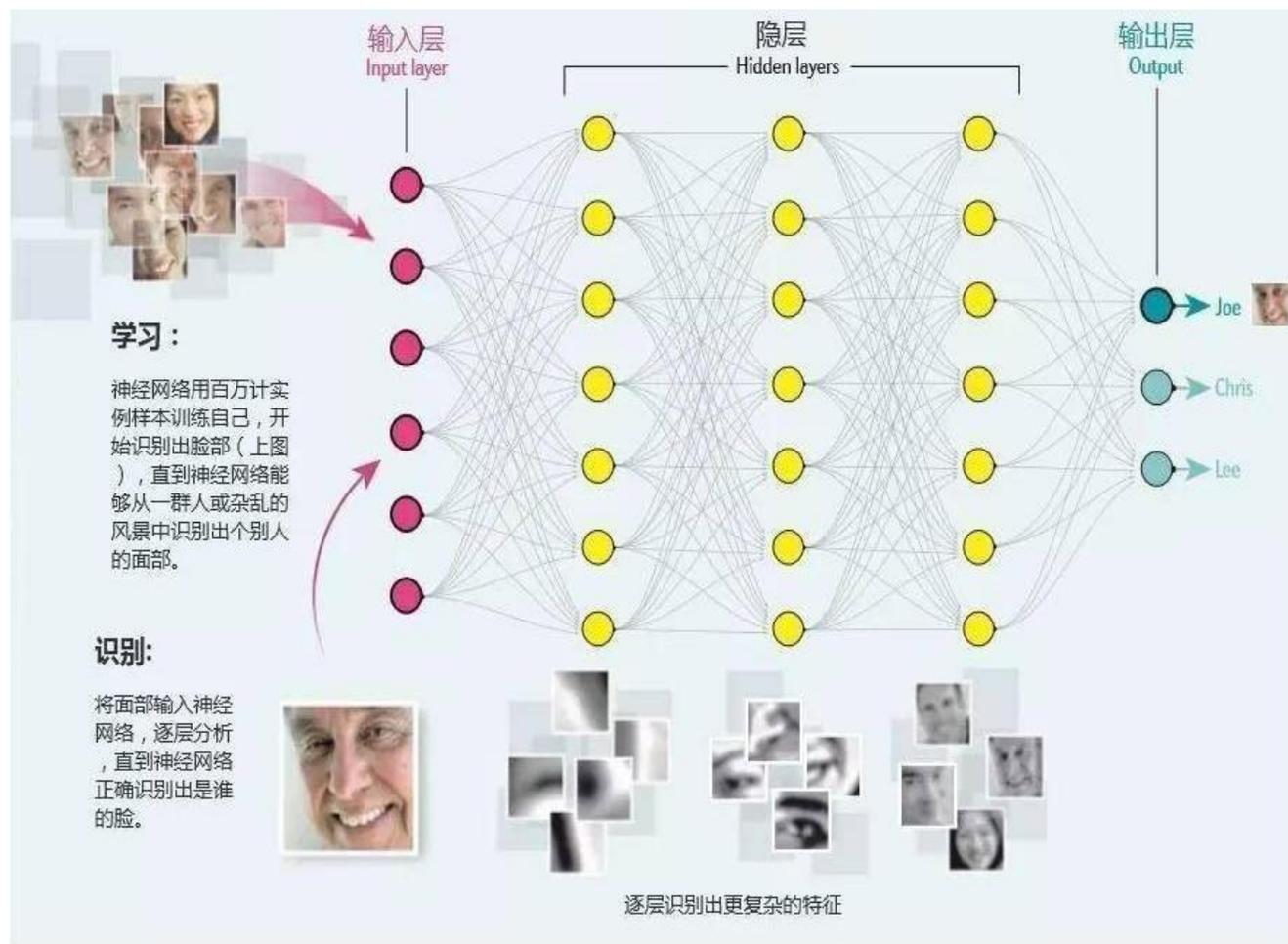
网络输出更光滑

深度学习

- ◆ 为什么要有深度？



知识的表征是层级式的（Hierarchical model）；
提高非线性的程度，增加网络的稀疏度。



- 模型复杂度越高、容量越大意味着能完成的学习任务更复杂。
但**训练效率低，容易过拟合。**
- 计算能力大幅提高可缓解训练低效性。
训练数据大幅增加可降低过拟合风险

为什么要深

从模型复杂度的角度看，增加隐层数目比增加隐层神经元的数目更有效，因为增加隐层数目不仅增加了拥有激活函数的神经元数目，还增加了激活函数的嵌套层数。**但也存在梯度消失的问题**

深度学习网络训练方法

- 无监督逐层训练
- **局部(稀疏)连接**
- 权共享 (卷积神经网络)

CNN示例

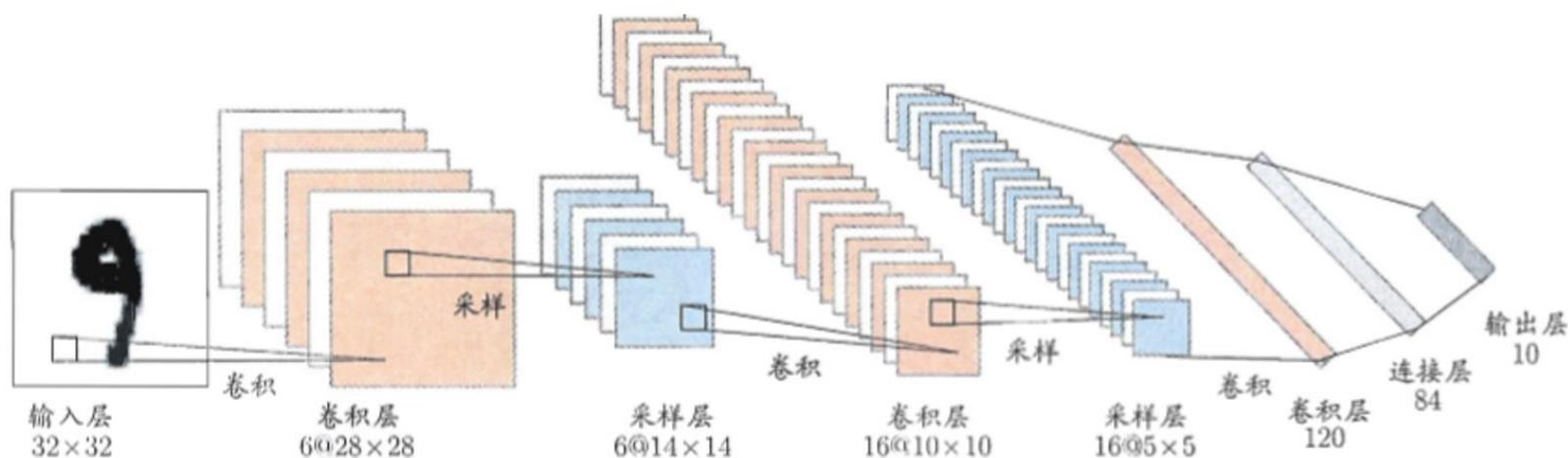


图 5.15 卷积神经网络用于手写数字识别 [LeCun et al., 1998]

➤ 用卷积来代替全连接

CNN: 卷积

参数共享

稀疏表示

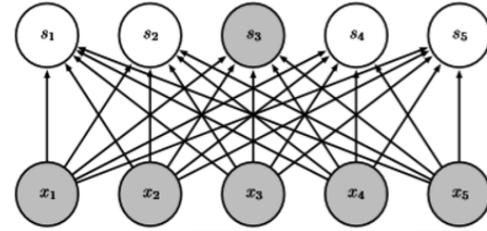
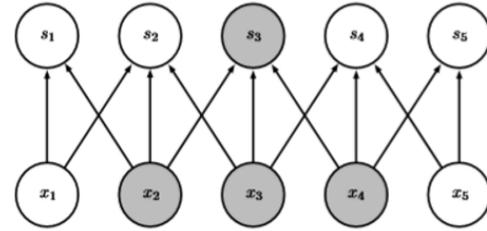
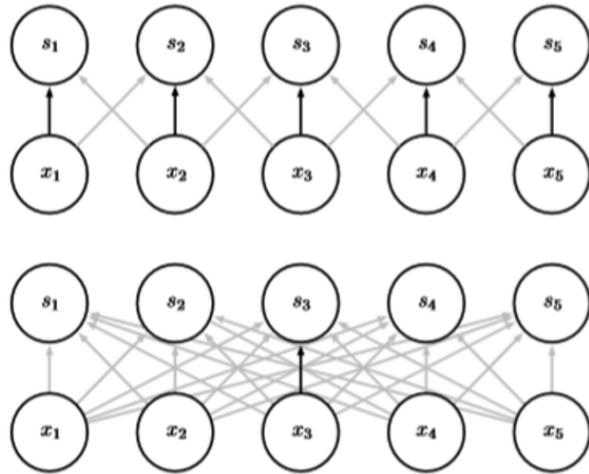


Figure 9.3: Tmp

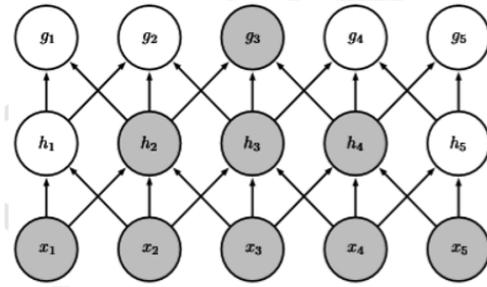
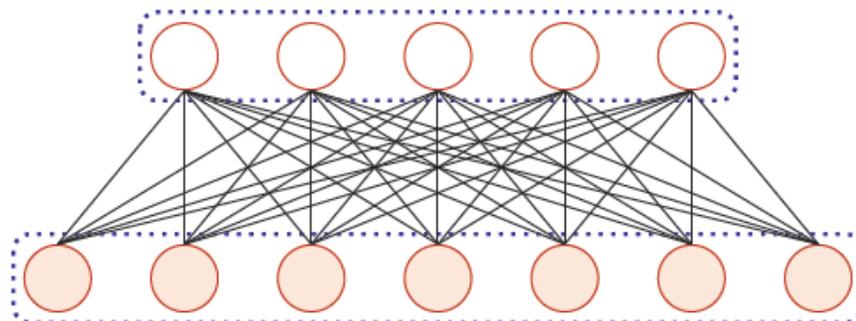
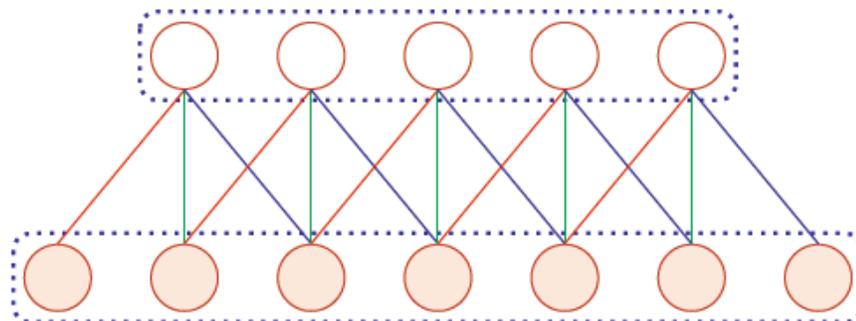


Figure 9.4: Tmp



(a) 全连接层



(b) 卷积层

- **卷积层的作用是提取一个局部区域的特征，不同的卷积核相当于不同的特征提取器**

CNN: 卷积

- ▶ 让过滤器在图像上逐步滑动

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

1	0	-1
1	0	-1
1	0	-1

filter/kernel
3*3

=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

- ▶ $1*1+6*1+2*1+7*0+2*0+3*0+8*(-1)+8*(-1)+9*(-1)=-16$

CNN: 卷积

- 我们假设卷积步长为2。原始图片大小7*7，过滤器3*3

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

7*7矩阵

步长stride=2

3	4	4
-1	0	2
-1	0	3

filter/kernel
3*3



91	100	83

CNN: 卷积

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

8*8矩阵

*

1	0	-1
1	0	-1
1	0	-1

filter/kernel
3*3

=

6*6矩阵
和原始图片大小相同

1	0	-1
1	0	-1
1	0	-1

垂直过滤器

1	1	1
0	0	0
-1	-1	-1

水平过滤器

1	0	-1
2	0	-2
1	0	-1

Sobel过滤器

3	0	-3
10	0	-10
3	0	-3

Scharr过滤器

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

1	0	-1
1	0	-1
1	0	-1

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



在输入层,特征映射就是图像本身.如果是灰度图像,就是有一个特征映射,输入层的深度 $D = 1$;如果是彩色图像,分别有 RGB 三个颜色通道的特征映射,输入层的深度 $D = 3$.

不失一般性,假设一个卷积层的结构如下:

(1) 输入特征映射组: $\mathcal{X} \in \mathbb{R}^{M \times N \times D}$ 为三维张量 (Tensor), 其中每个切片 (Slice) 矩阵 $\mathbf{X}^d \in \mathbb{R}^{M \times N}$ 为一个输入特征映射, $1 \leq d \leq D$;

(2) 输出特征映射组: $\mathcal{Y} \in \mathbb{R}^{M' \times N' \times P}$ 为三维张量, 其中每个切片矩阵 $\mathbf{Y}^p \in \mathbb{R}^{M' \times N'}$ 为一个输出特征映射, $1 \leq p \leq P$;

(3) 卷积核: $\mathcal{W} \in \mathbb{R}^{U \times V \times P \times D}$ 为四维张量, 其中每个切片矩阵 $\mathbf{W}^{p,d} \in \mathbb{R}^{U \times V}$ 为一个二维卷积核, $1 \leq p \leq P, 1 \leq d \leq D$.

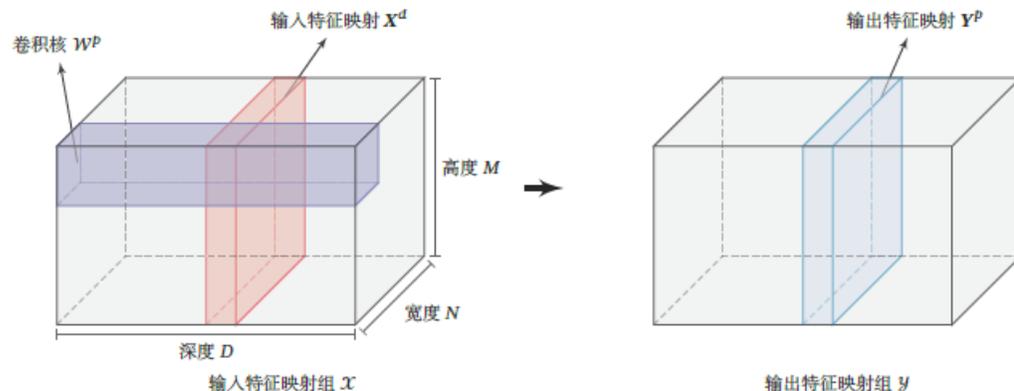


图 5.6 卷积层的三维结构表示

整个计算过程如图5.7所示. 如果希望卷积层输出 P 个特征映射, 可以将上述计算过程重复 P 次, 得到 P 个输出特征映射 Y^1, Y^2, \dots, Y^P .

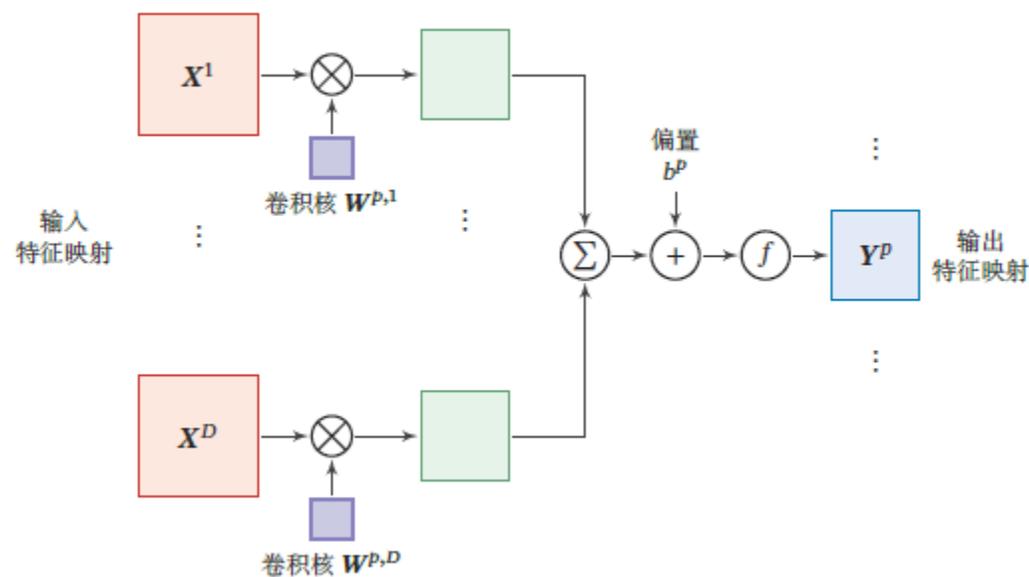
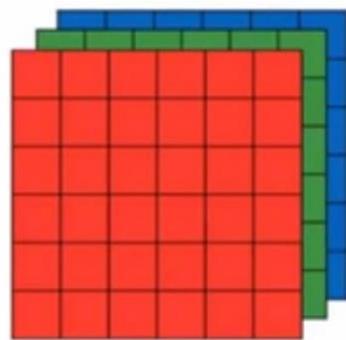
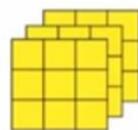


图 5.7 卷积层中从输入特征映射组 X 到输出特征映射 Y^P 的计算示例

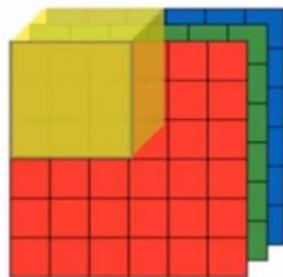


原始图片：大小为6*6，但是由于有R/G/B三种颜色通道color channels，所以整个是一个6*6*3大小的矩阵。



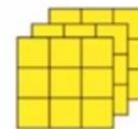
过滤器：针对每一个颜色通道，对应地有一层过滤器，所以过滤器大小为3*3*3，其中最后一个3代表通道数channels，也叫深度depth。

过滤器channel的个数和原始图片channel的个数必须相等，一个对应一个进行卷积操作。



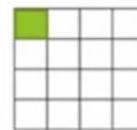
6*6*3

*



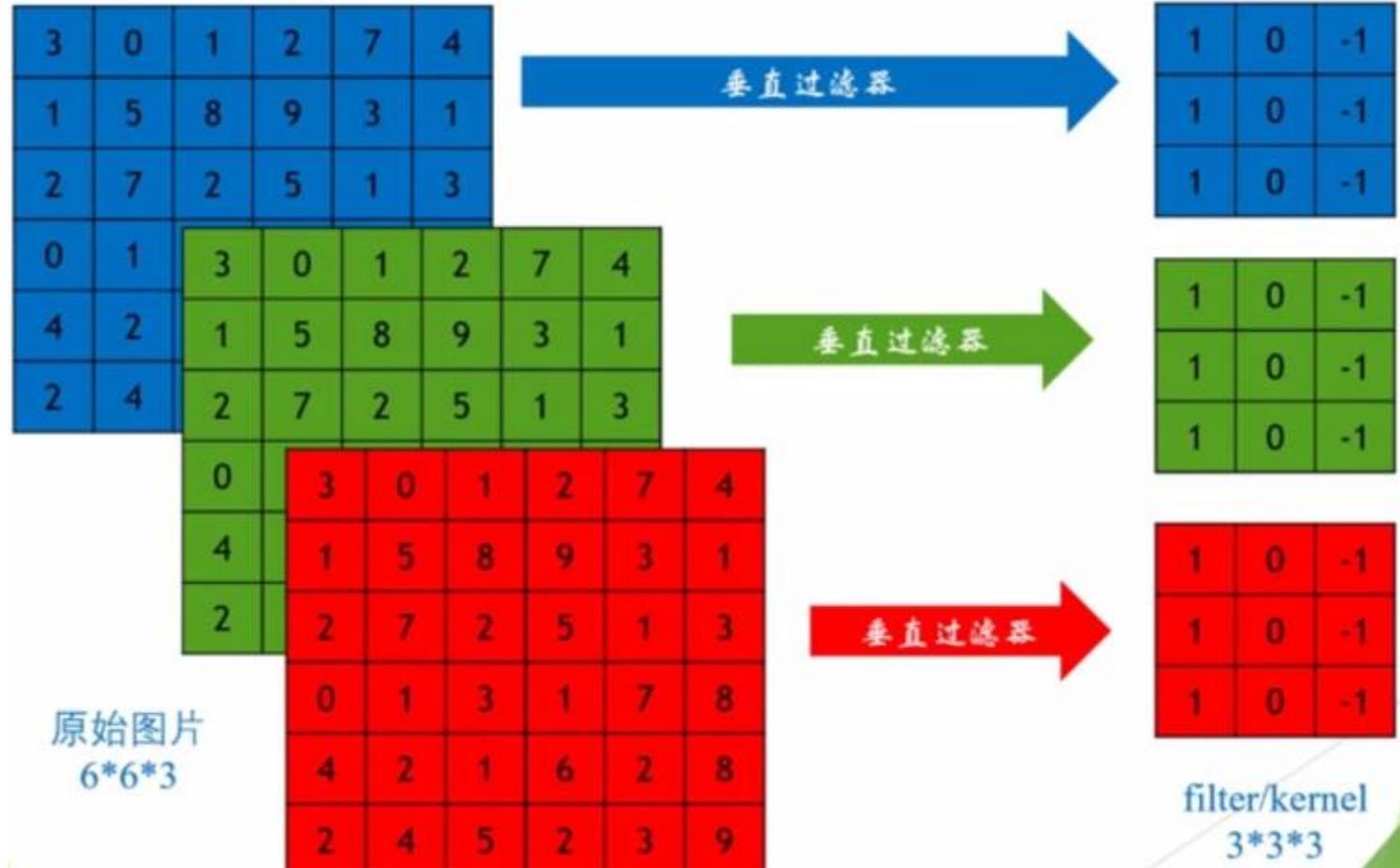
filter/kernel
3*3*3

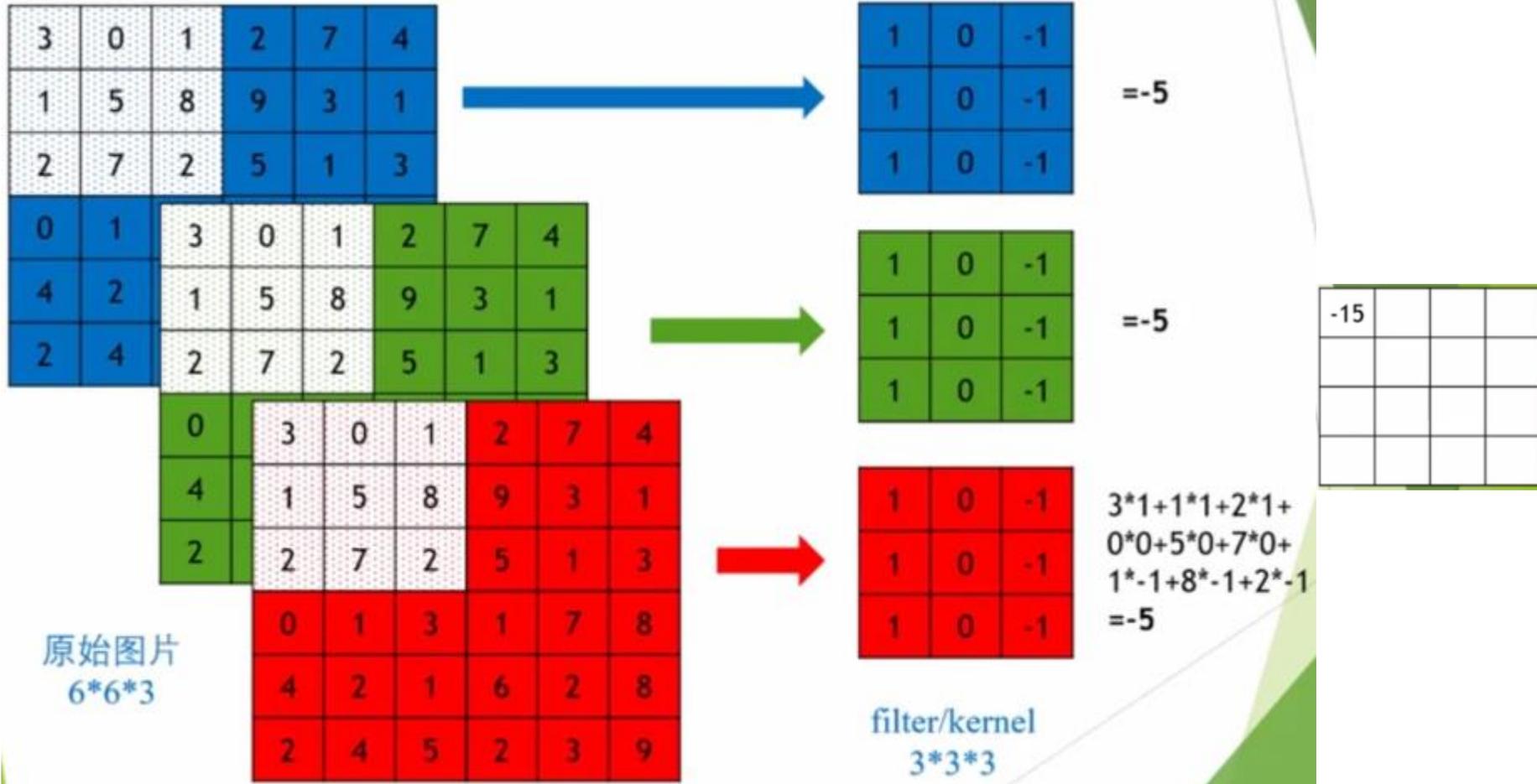
=



4*4*1

原始图片的某个channel和过滤器的对应channel按照灰度图片的计算方法计算，然后把各个channel的结果相加

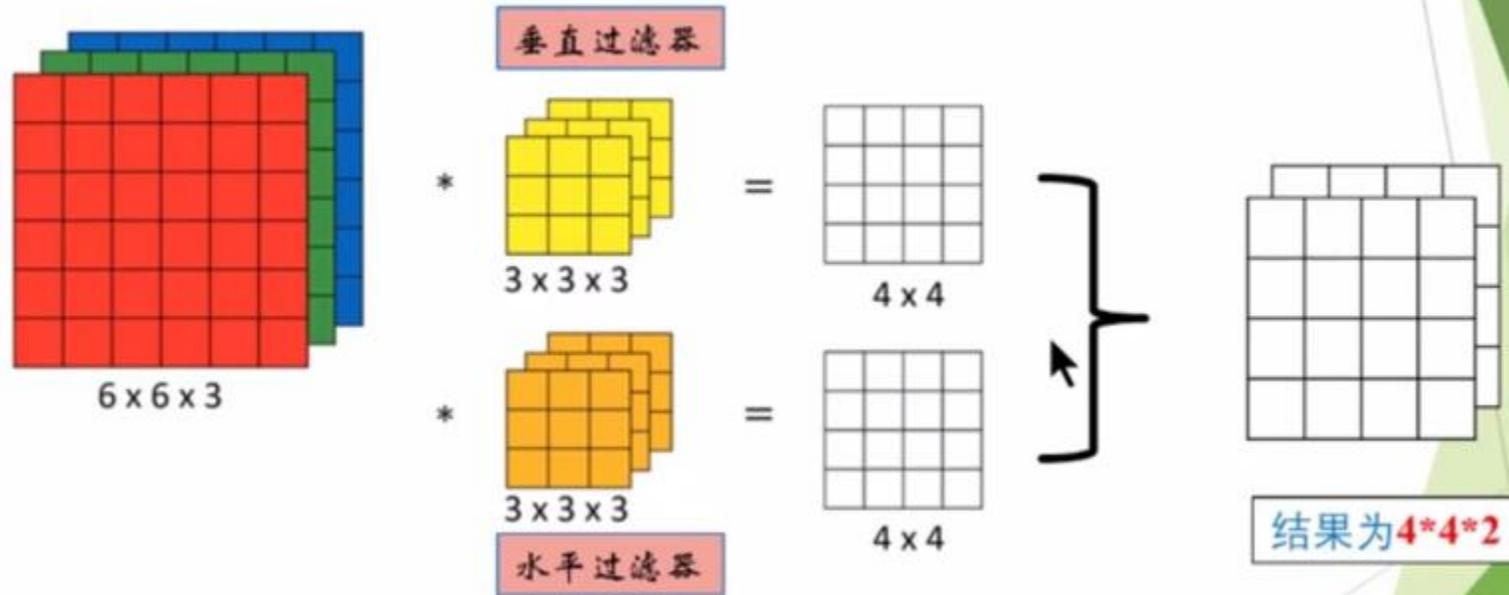




多层过滤器

如果我们在不仅仅在图像总检测一种类型的特征，而是要同时检测垂直边缘、水平边缘、45度边缘等等，就需要用到多个过滤器。

多层过滤器



如果有5个过滤器那么输出图像就是4*4*5的立方体。

池化

- 卷积层虽然可以显著减少网络中连接的数量，但特征映射组中的神经元个数并没有显著减少，可以在卷积层之后加上一个汇聚层，从而降低特征维数，避免过拟合
- **汇聚**（Pooling）是指对每个区域进行下采样（Down Sampling）得到一个值，作为这个区域的概括



常用的汇聚函数有两种：

(1) **最大汇聚** (Maximum Pooling 或 Max Pooling) : 对于一个区域 $R_{m,n}^d$, 选择这个区域内所有神经元的最大活性值作为这个区域的表示, 即

$$y_{m,n}^d = \max_{i \in R_{m,n}^d} x_i, \quad (5.25)$$

其中 x_i 为区域 R_k^d 内每个神经元的活性值.

(2) **平均汇聚** (Mean Pooling) : 一般是取区域内所有神经元活性值的平均值, 即

$$y_{m,n}^d = \frac{1}{|R_{m,n}^d|} \sum_{i \in R_{m,n}^d} x_i. \quad (5.26)$$

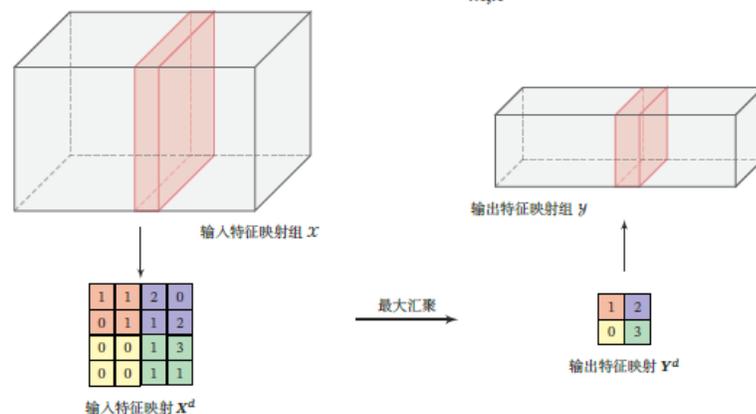
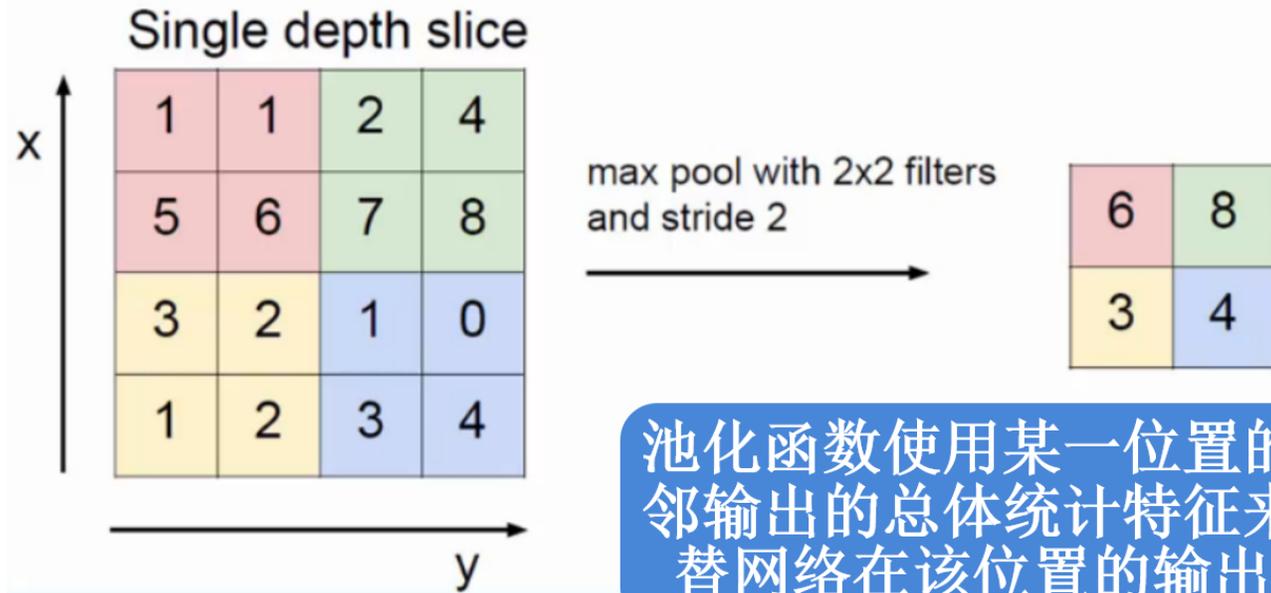


图 5.8 汇聚层中最大汇聚过程示例

CNN: 池化



池化函数使用某一位置的相邻输出的总体统计特征来代替网络在该位置的输出。

相邻矩形区域内的最大值

相邻矩形区域内的平均值

CNN: 池化

局部平移
不变性

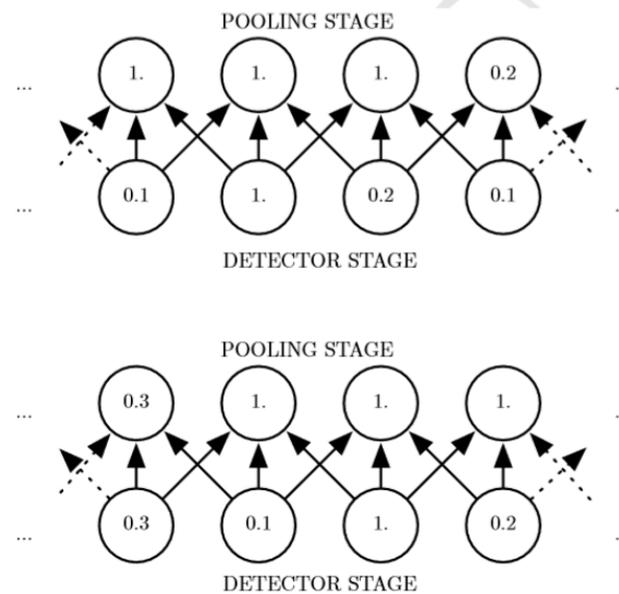


图 9.8: 最大池化引入不变性。(上) 卷积层中间输出的视图。下面一行显示非线性的输出。上面一行显示最大池化的输出, 每个池的宽度为三个像素并且池化区域的步幅为一个像素。(下) 相同网络的视图, 不过对输入右移了一个像素。下面一行的所有值都发生了改变, 但上面一行只有一半的值发生了改变, 这是因为最大池化单元只对周围的最大值比较敏感, 而不是对精确的位置。

CNN: 池化

旋转不变性

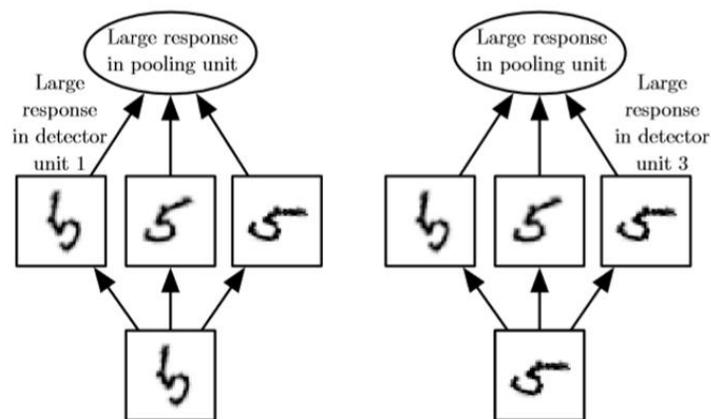


图 9.9: 学习不变性的示例。使用分离的参数学得多个特征，再使用池化单元进行池化，可以学得对输入的某些变换的不变性。这里我们展示了用三个学得的过滤器和一个最大池化单元可以学得对旋转变换的不变性。这三个过滤器都旨在检测手写的数字 5。每个过滤器尝试匹配稍微不同方向的 5。当输入中出现 5 时，相应的过滤器会匹配它并且在探测单元中引起大的激活。然后，无论哪个探测单元被激活，最大池化单元都具有大的激活。我们在这里展示网络如何处理两个不同的输入，导致两个不同的探测单元被激活。然而对池化单元的影响大致相同。这个原则在 maxout 网络 (Goodfellow *et al.*, 2013b) 和其他卷积网络中使用。空间位置上的最大池化对于平移是天然不变的；这种多通道方法只在学习其他变换时是必要的。

CNN: 池化

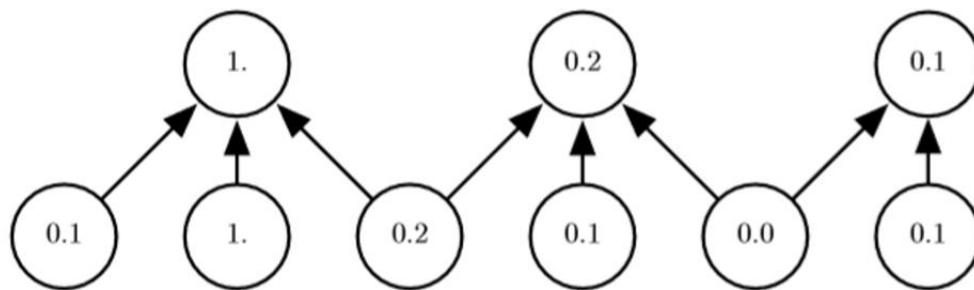
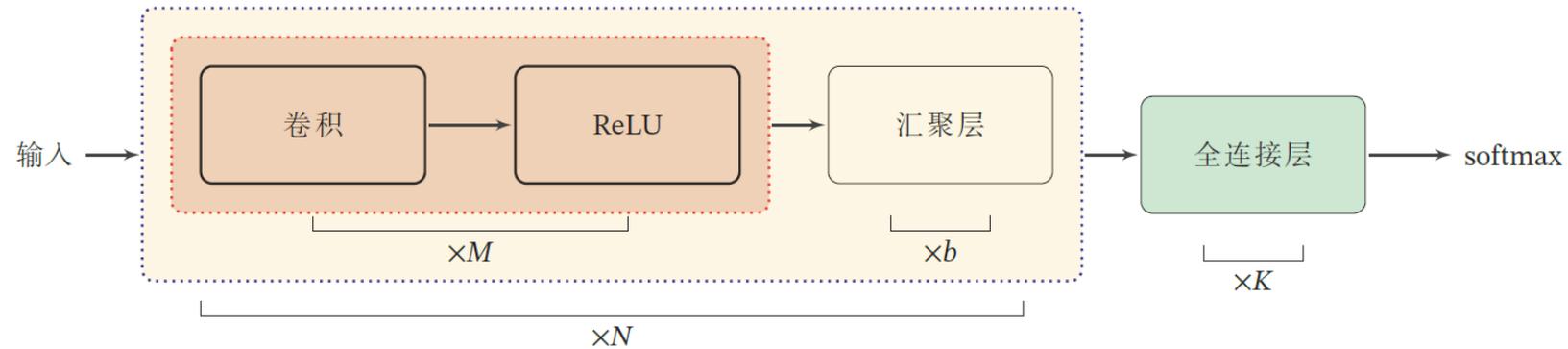


图 9.10: 带有下采样的池化。这里我们使用最大池化, 池的宽度为三并且池之间的步幅为二。这使得表示的大小减少了一半, 减轻了下一层的计算和统计负担。注意到最右边的池化区域尺寸较小, 但如果我们不想忽略一些探测单元的话就必须包含这个区域。

减少输入规模提高统计效率并且减少对于参数的 存储需求。

卷积神经网络

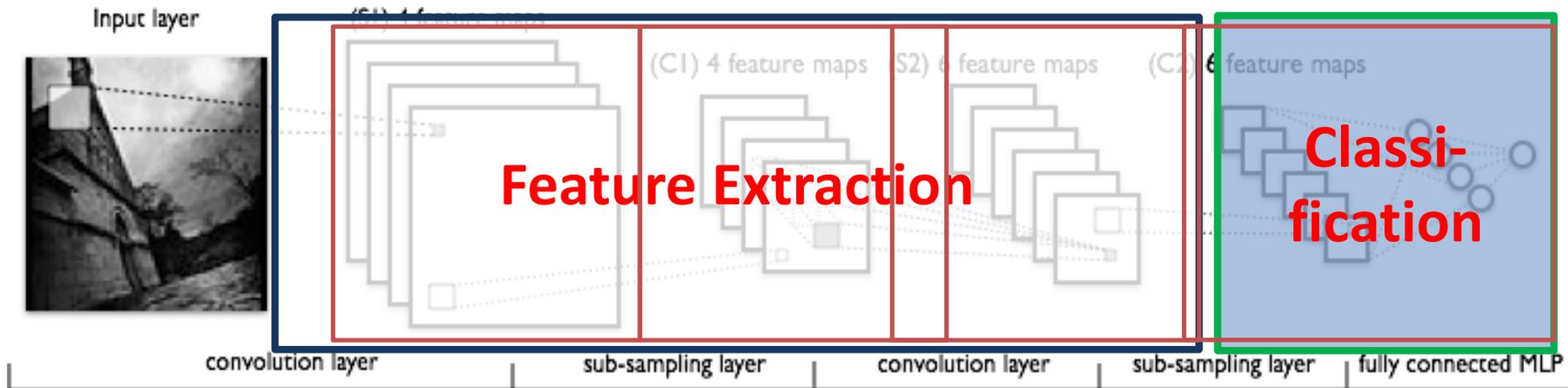
- 一个典型的卷积网络是由卷积层、汇聚层、全连接层交叉堆叠而成
- 例如一个卷积块为连续 M 个卷积层和 b 个汇聚层（ M 通常设置为2~5， b 为0或1）。一个卷积网络中可以堆叠 N 个连续的卷积块，然后在后面接着 K 个全连接层（ N 的取值区间比较大，比如1~100或者更大， K 一般为0~2）



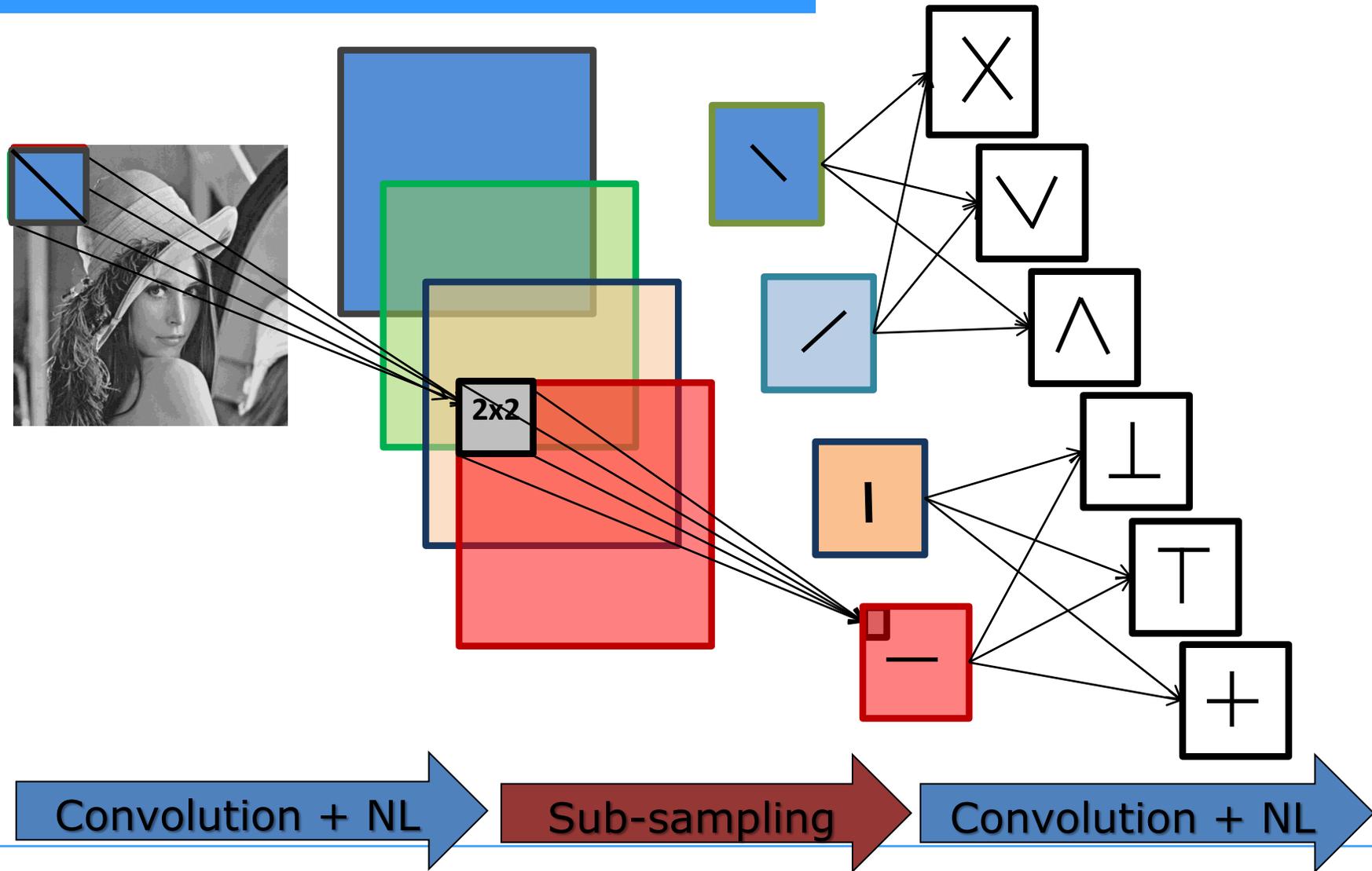
- 卷积网络的整体结构趋向于使用更小的卷积核（比如 1×1 和 3×3 ）以及更深的结构（比如层数大于50）

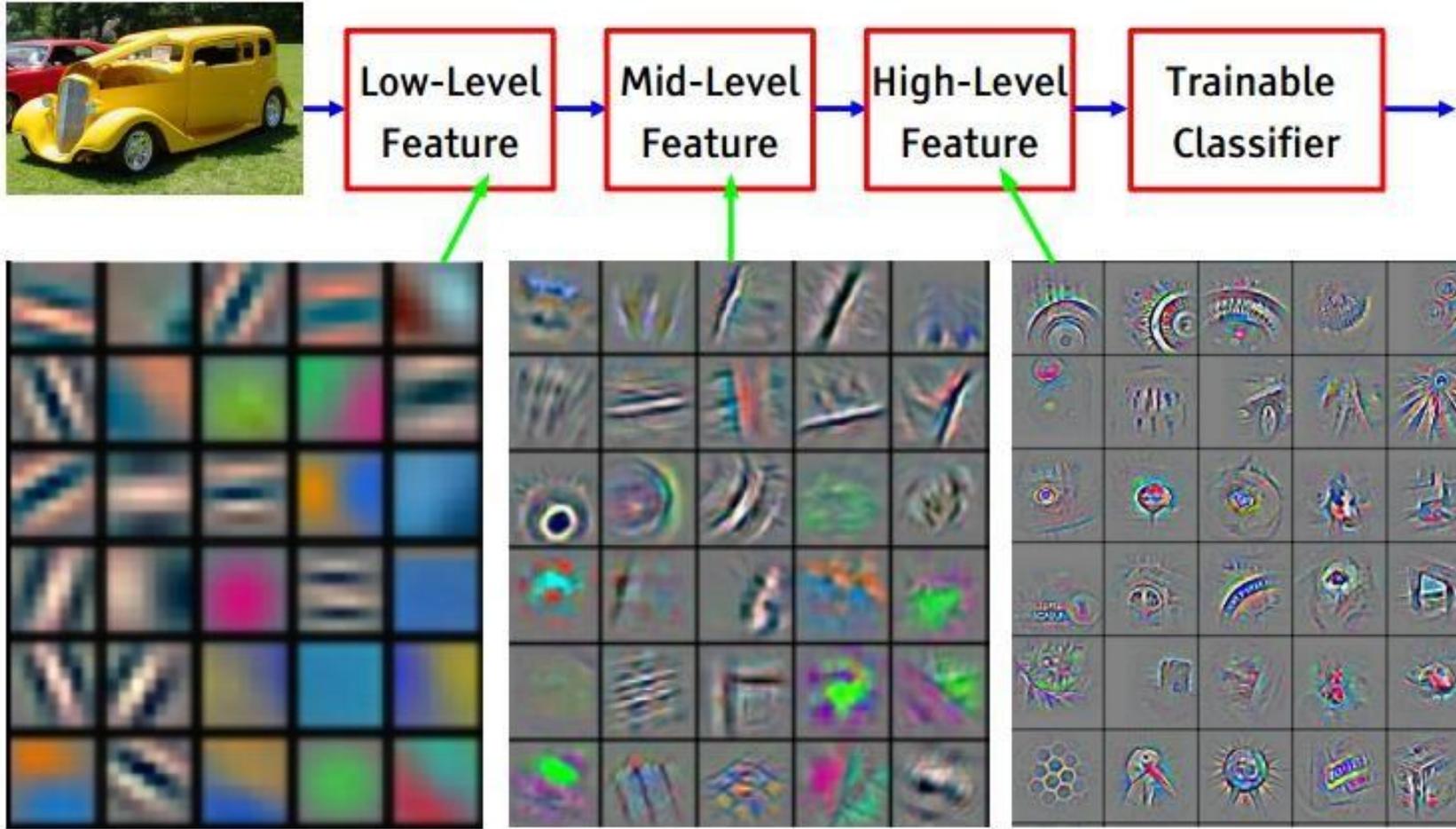
➤ CNN结构：多层神经网络架构

- 卷积+非线性层（ReLU）。
- 降采样（Max Pooling）。
- 卷积+非线性层（ReLU）。
- 降采样（Max Pooling）。
- 全连接



➤ CNN结构：多层神经网络架构





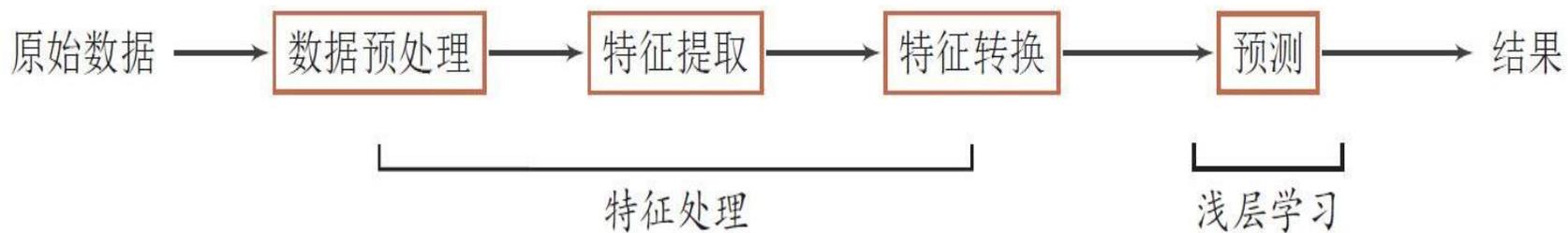


图 1.3 传统机器学习的数据处理流程

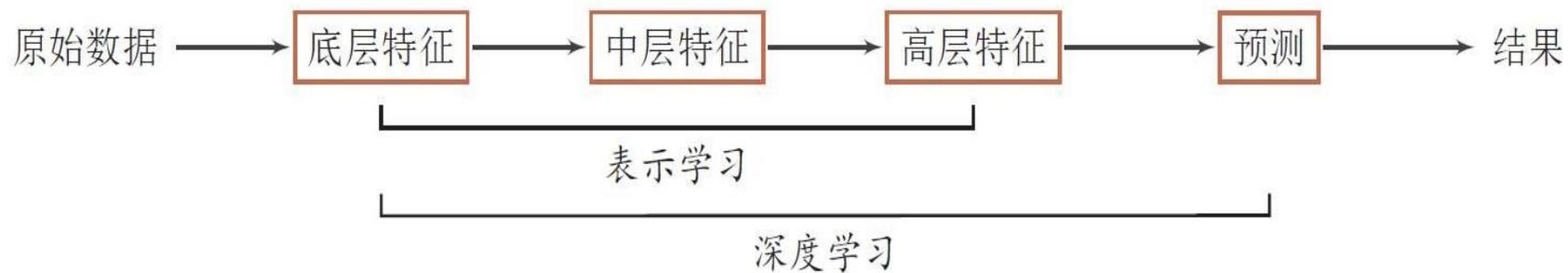


图 1.5 深度学习的数据处理流程

深度神经网络举例

LeNet-5

LeNet-5[LeCun et al., 1998] 虽然提出的时间比较早,但它是一个非常成功的神经网络模型. 基于 LeNet-5 的手写数字识别系统在 20 世纪 90 年代被美国很多银行使用,用来识别支票上面的手写数字. LeNet-5 的网络结构如图 5.10 所示.

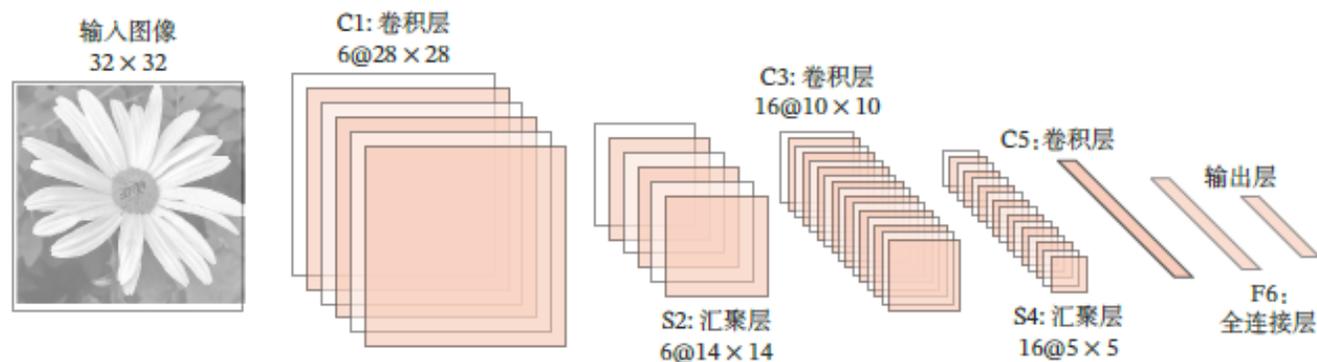


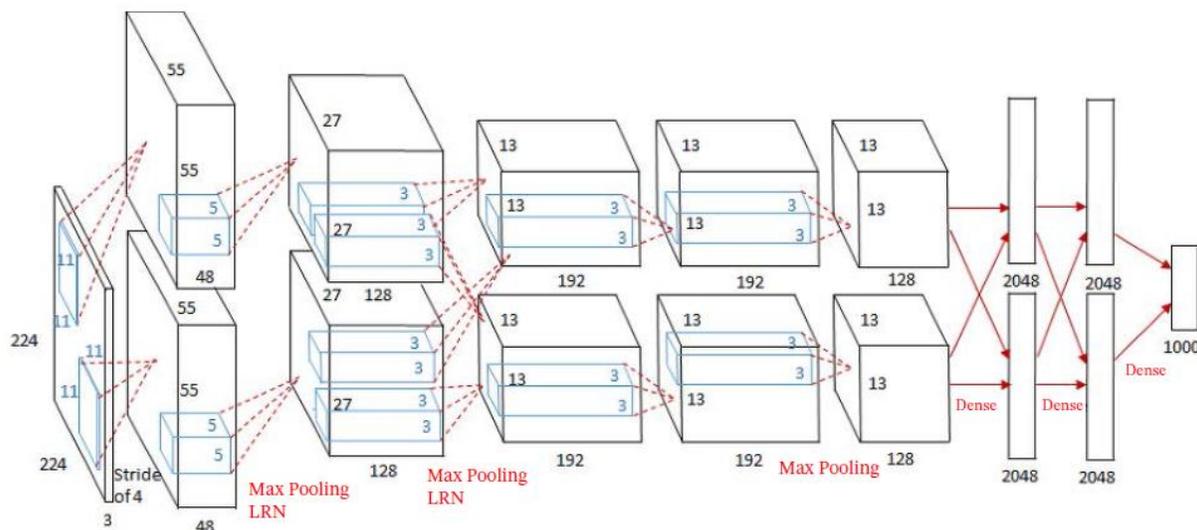
图 5.10 LeNet-5 网络结构 (图片根据 [LeCun et al., 1998] 绘制)

LeNet-5 共有 7 层, 接受输入图像大小为 $32 \times 32 = 1024$, 输出对应 10 个类别的得分.

AlexNet

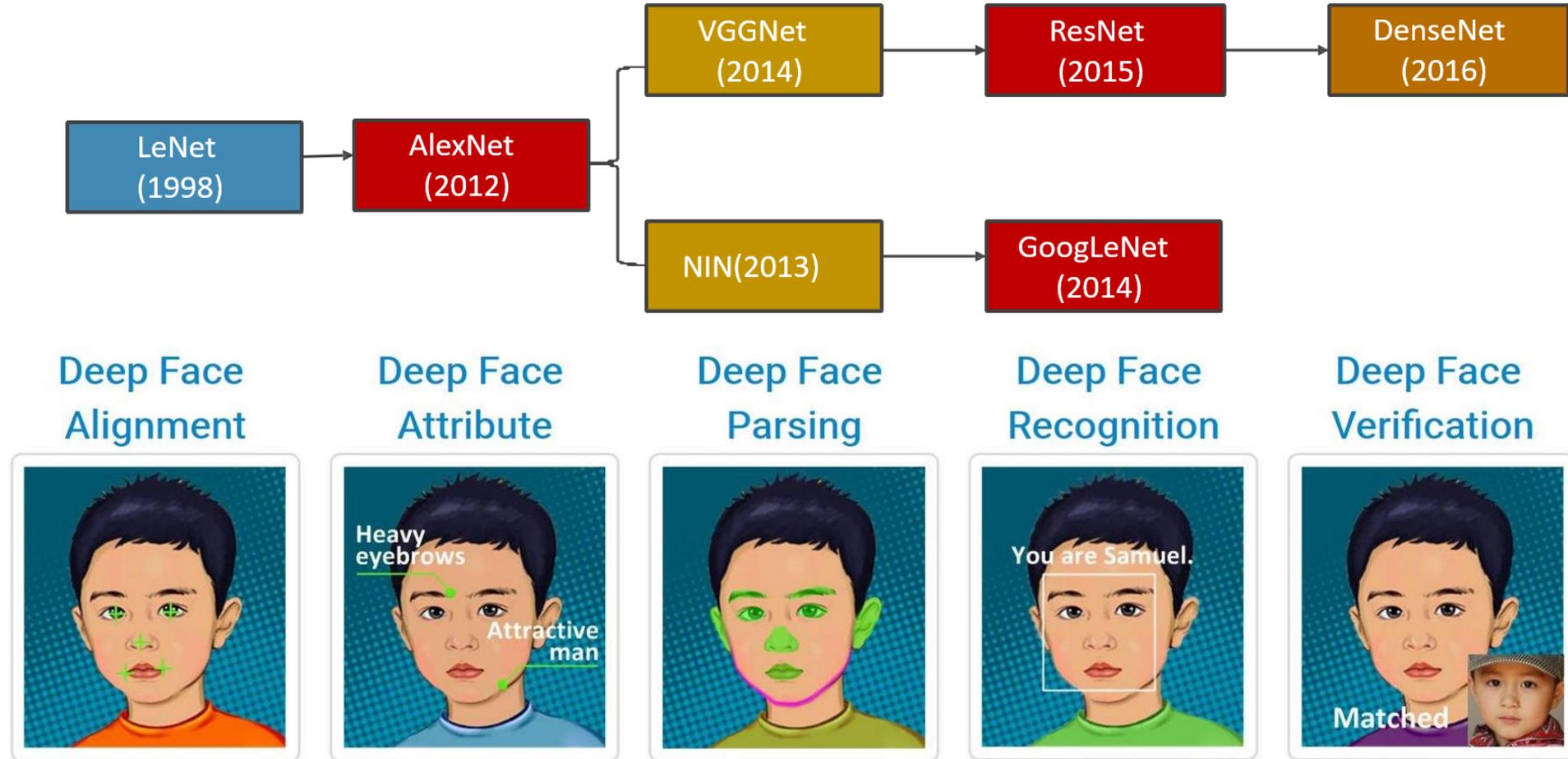
AlexNet[Krizhevsky et al., 2012] 是第一个现代深度卷积网络模型, 其首次使用了很多现代深度卷积网络的技术方法, 比如使用 GPU 进行并行训练, 采用了 ReLU 作为非线性激活函数, 使用 Dropout 防止过拟合, 使用数据增强来提高模型准确率等. AlexNet 赢得了 2012 年 ImageNet 图像分类竞赛的冠军.

AlexNet 的结构如图 5.12 所示, 包括 5 个卷积层、3 个汇聚层和 3 个全连接层 (其中最后一层是使用 Softmax 函数的输出层). 因为网络规模超出了当时的单个 GPU 的内存限制, AlexNet 将网络拆为两半, 分别放在两个 GPU 上, GPU 间只在某些层 (比如第 3 层) 进行通信.



AlexNet 的输入为 $224 \times 224 \times 3$ 的图像, 输出为 1 000 个类别的条件概率

• CNN模型结构变迁



生成对抗网络

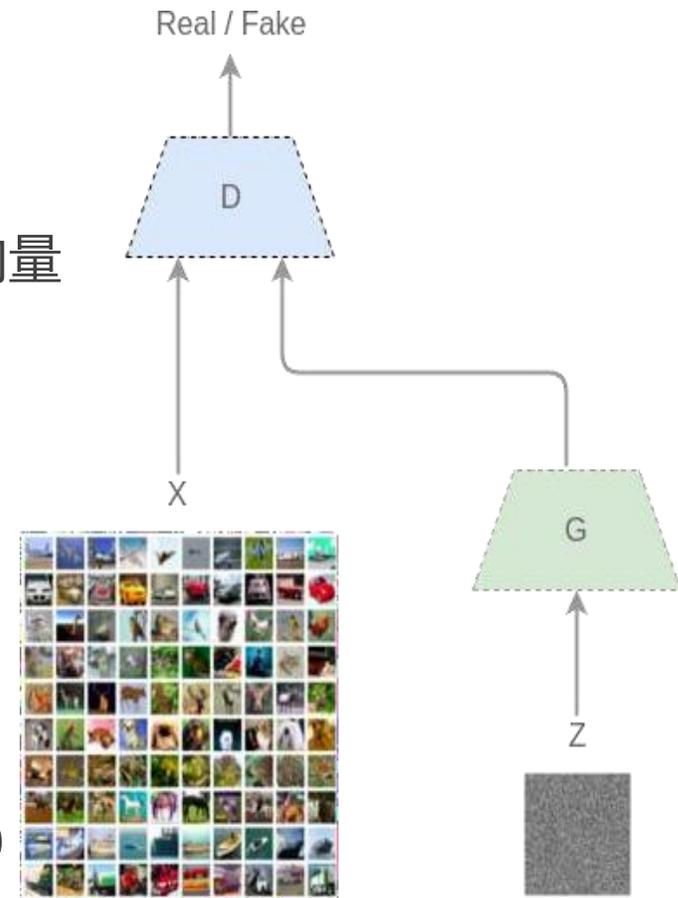
- 生成对抗网络 (Generative Adversarial Network, GAN)

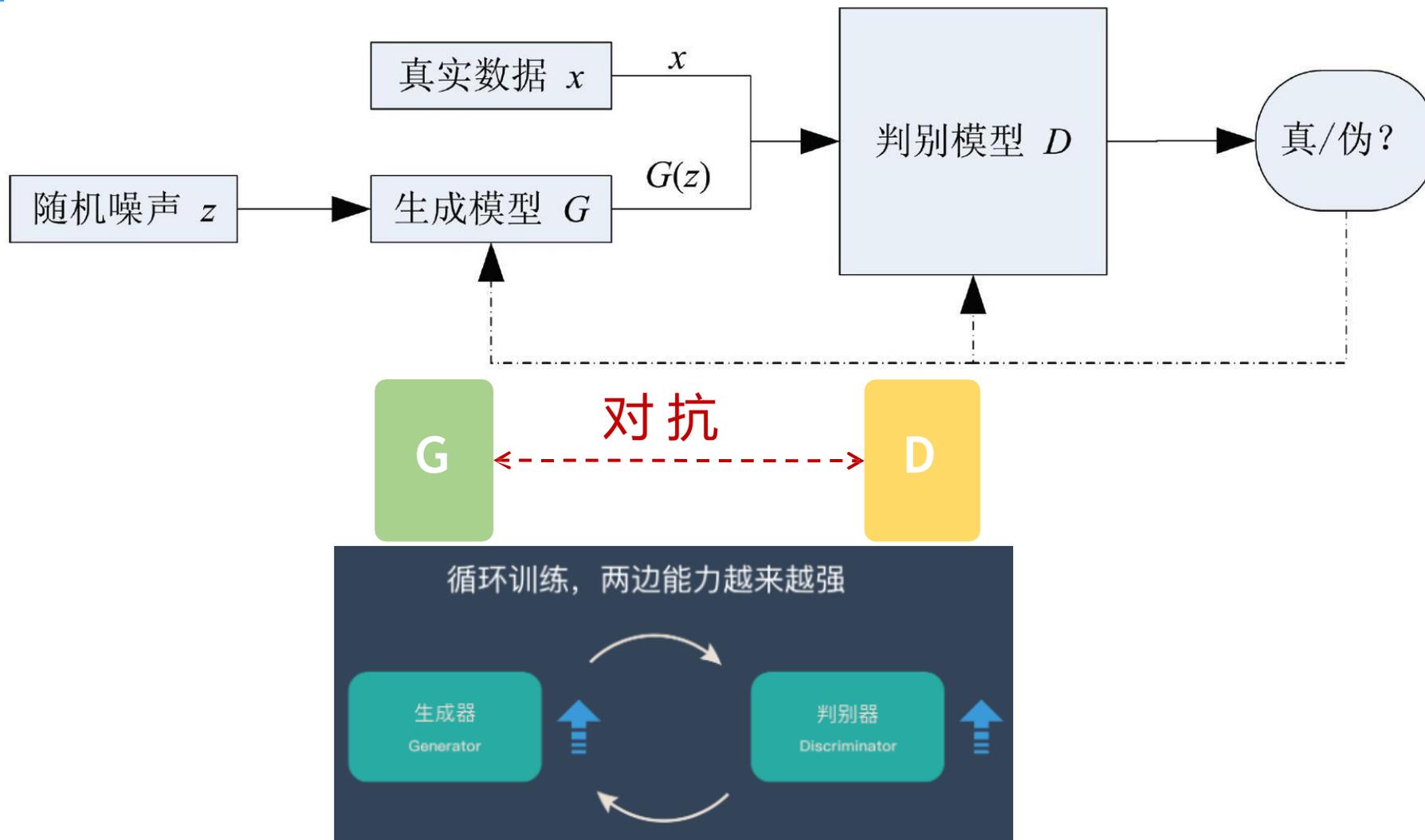
- 生成器 G

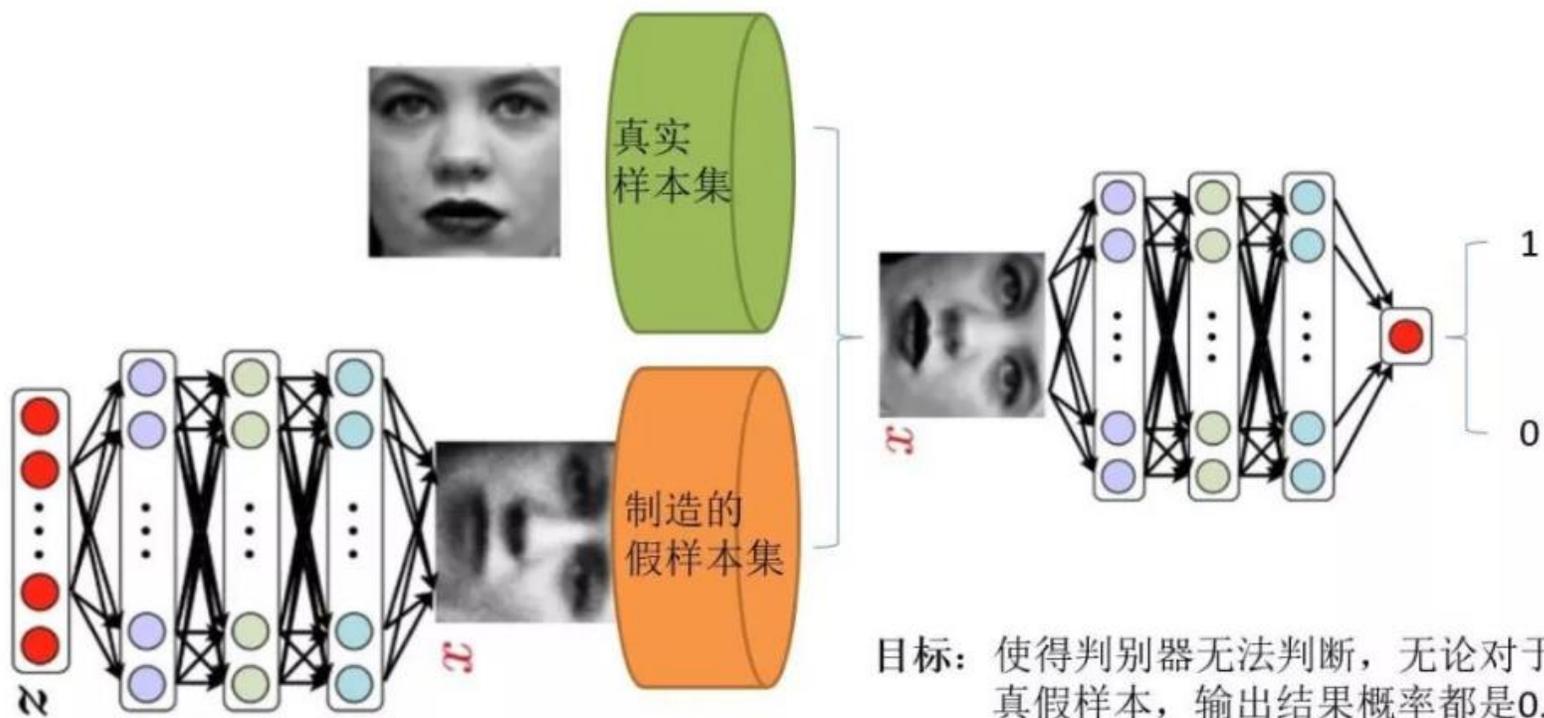
- 生成让 D 无法判断真假的样本
- 输入 z : 从一个先验概率分布取样的随机向量
- 输出 $G(z)$: 生成的样本

- 判别器 D

- 判断样本是真实的还是生成的
- 输入: 真实样本/G 生成的样本
- 输出: 对真实样本输出 1, 生成样本输出 0









ZAO - 逢脸造戏

仅需一张照片，出演天下好戏

★★★★☆ 410

获取

