

# 图像处理与分析

## 第十七讲：图像压缩



胡晰远

中国科学院自动化研究所

[xiyuan.hu@ia.ac.cn](mailto:xiyuan.hu@ia.ac.cn)

# 内容大纲

- 基本概念
- 信息论基础
- 常见的无损压缩方法
- 图像压缩方法
  - 预测编码
  - 变换编码
- 图像视频压缩标准介绍

# 基本概念

- 数据压缩定义及分类
- 图像的数据冗余
  - 编码冗余
  - 像素间冗余
  - 心理视觉冗余
- 图像保真度和质量
- 图像压缩一般框架



## 图像压缩的必要性

- 图像和视频的数据量通常很大，对存储、处理和传输带来许多问题
- 图像和视频的应用越来越广泛
  - ✓ 互联网，移动互联网的应用
  - ✓ 遥感影像
  - ✓ 视频会议，超高清电视
  - ✓ 数字图书馆
  - ✓ .....



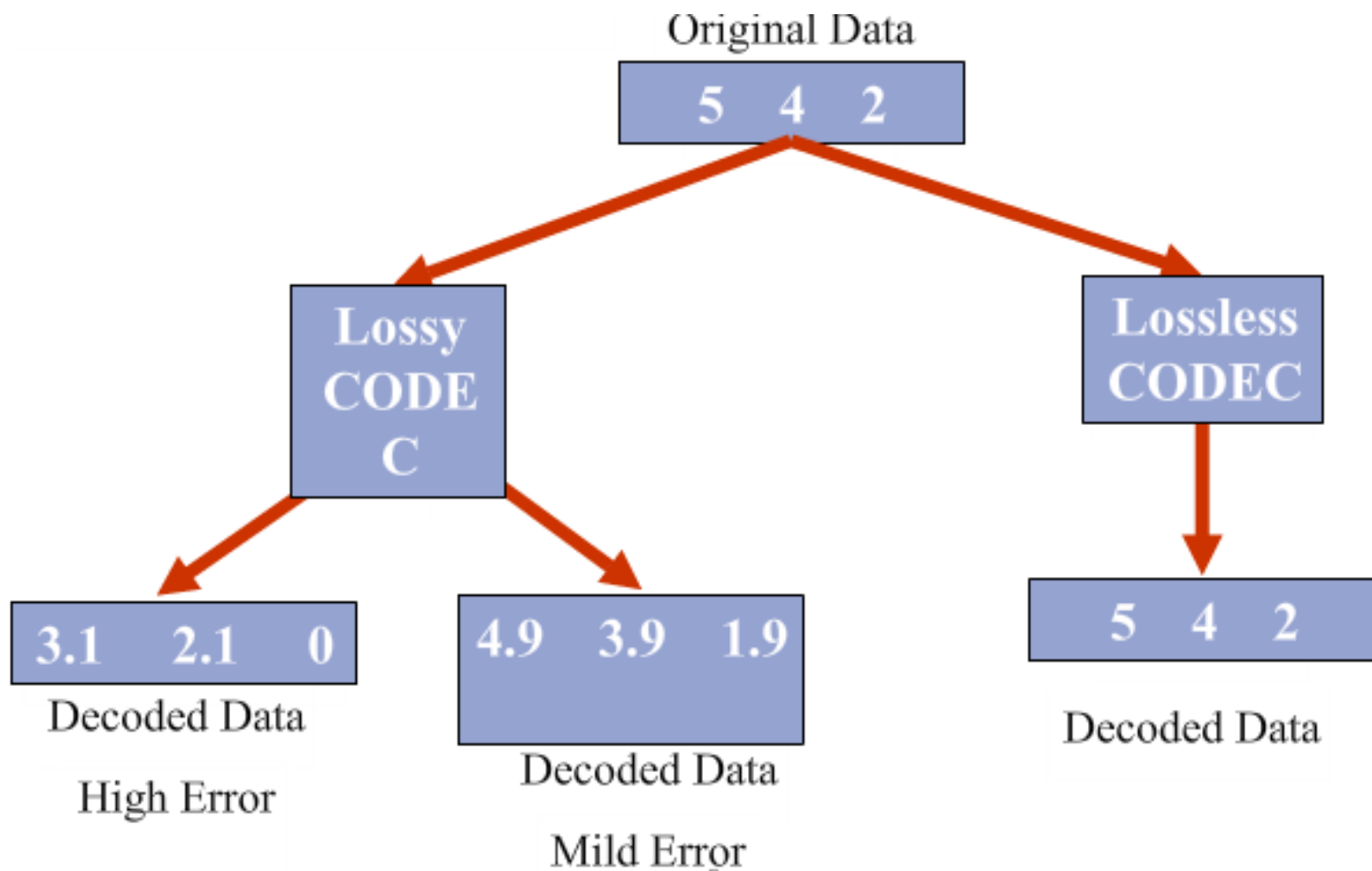
## 图像压缩的必要性

- Uncompressed Image & Video
  - 512×512 gray scale image → 256k
  - 512×512 color image → 768k
  - 90 minute 512×512 color movie (25@fps)  
→  $90 \times 60 \times 25 \times 768k = 98G$
- Given a 1T hard disk, can only store about 10 of 1.5 hour high quality videos
- MPEG-4/H.264 compressed video often less than 1G for a high quality video

## 数据压缩的定义与分类

- 数据压缩 (Data Compression) :
  - 减少表示给定信息量所需数据量的处理。
  - refers to the process of reducing the amount of data required to represent a given quality of information
- 压缩方法的分类:
  - 无损压缩 (lossless compression)
  - 有损压缩 (lossy compression)
  - 准无损压缩 (可根据需要支持无损/有损压缩)

## ■ 数据压缩的定义与分类





## ■ 图像压缩

- 无损压缩: png, tif, .....
- 有损压缩: jpg, jp2, mp4, .....



.png (438k)



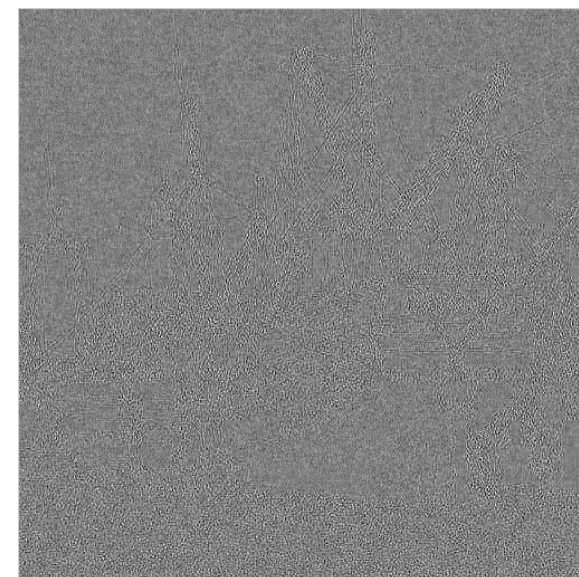
Diff



.bmp (768k)



.jpg (39k)



Diff





## 数据冗余的概念

- 数据是用来表示信息的。如果不同的方法为表示给定量的信息使用了不同的数据量，那么使用较多数据量的方法中，有些数据必然是代表了无用的信息，或者是重复地表示了其它数据已表示的信息，这就是数据冗余的概念。
- 数据冗余： $R = 1 - \frac{1}{C}$ ，其中 $C$ 定义为压缩率：

$$C = \frac{n_1}{n_2}。$$



## 图像中的数据冗余

- 和一般数据相比，自然图像中存在哪些冗余信息？
  - 编码冗余（信源符号出现的概率不同）
  - 像素间冗余（图像数据是高度相关的）
  - 心理视觉冗余（人眼对所有视觉信息感受的灵敏度不同）
- 如果能减少或消除上述三种冗余的1种或多种冗余，就能取得数据压缩的效果。

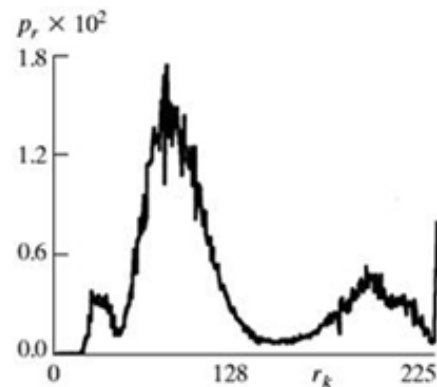
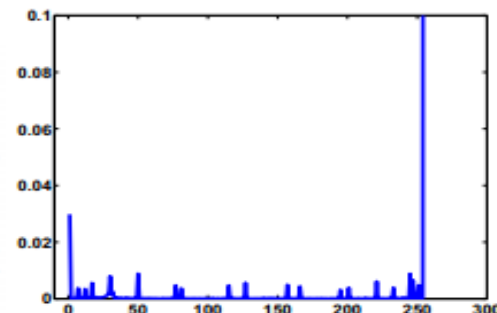


# 图像中的数据冗余

- 编码冗余（信源符号出现的概率不同）
  - 如果一个图像的灰度级编码，使用了多于实际需要的编码符号，就称该图像包含了编码冗余
  - 例如：文本图像和自然图像

CVPR  
2014

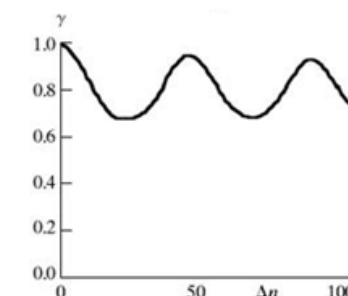
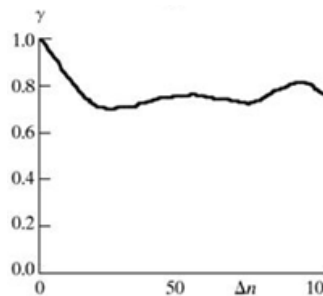
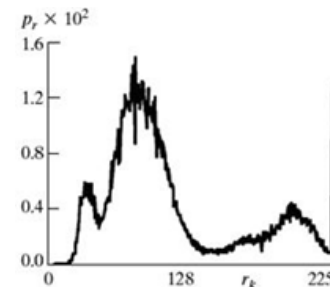
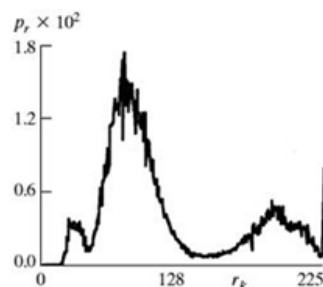
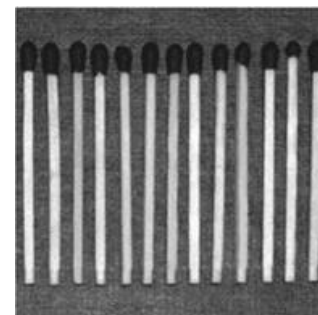
CVPR is the premier annual Computer Vision event comprising the main CVPR conference and several co-located workshops and short courses. With its high quality and low cost, it provides an exceptional value for students, academics and industry researchers.



# ■ 图像中的数据冗余

- 像素间冗余

- 反映图像中像素之间的相互关系
- 图像数据是高度相关的
- 单个像素的值可以通过与它相邻的像素值为基础进行预测



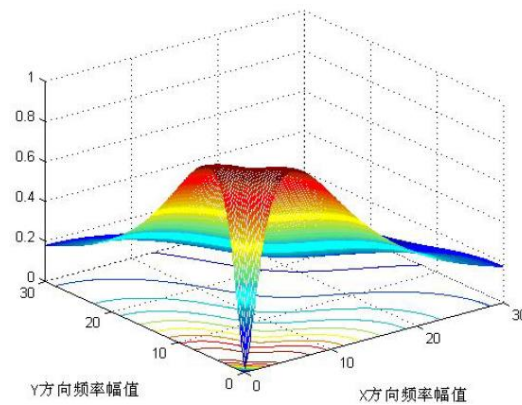
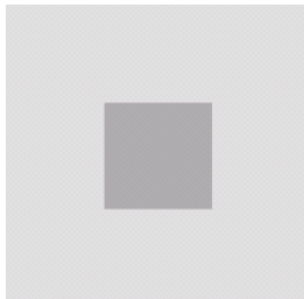
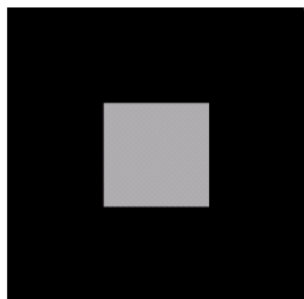
## Example

Original: 234 223 231 238 235 ...

Compressed: 234 -11 8 7 -3 ...

# ■ 图像中的数据冗余

- 心理视觉冗余
  - 人眼对所有视觉信息感受的灵敏度不同；
  - 有些信息在通常的视觉过程中与另外一些信息相比并不那么重要；
  - 上述这些信息被认为是心理视觉冗余的





## 图像保真度

- 需要评价信息损失的测度以描述解码图像相对于原始图像的偏离程度，这些测度称为保真度准则；
- 常用保真度分为两类：
  - 客观保真度准则
  - 主观保真度准则



## ■ 常用的图像客观保真度指标

- 原始图像：  $x(m, n)$ ，解码后图像  $\hat{x}(m, n)$
- 均方（根）误差（MSE/RMSE）

$$MSE = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |x(m, n) - \hat{x}(m, n)|^2$$

- 峰值信噪比（PSNR）

$$PSNR = 10 \log_{10} \frac{(\text{peak signal value})^2}{MSE} \text{ (db)}$$

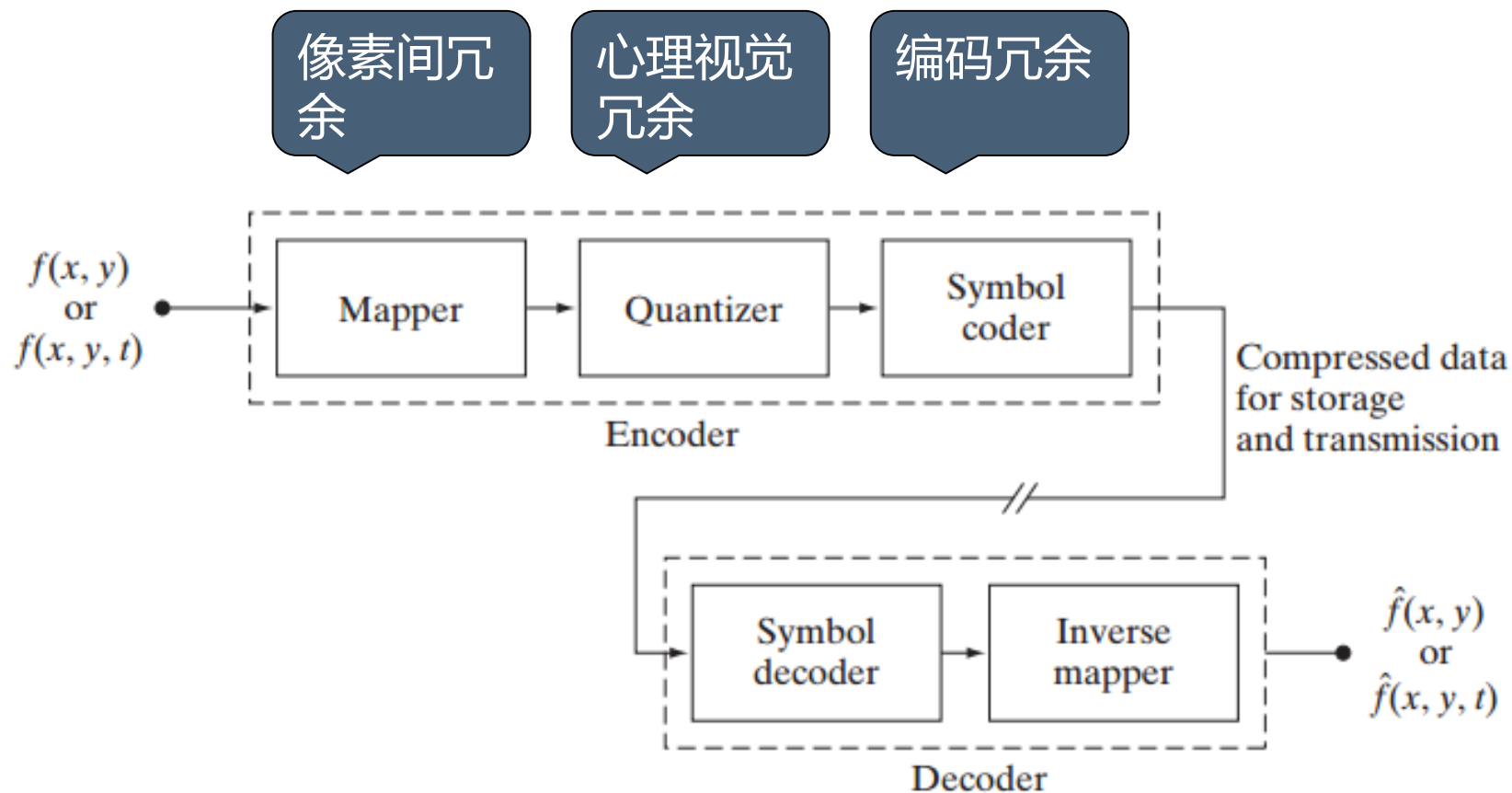
$$= 10 \log_{10} \frac{255^2}{MSE} \text{ (db)}$$

- 结构化相似度（Structural Similarity Index, SSIM）

## ■ 图像的主观保真度指标

评分	评价	说明
1	优秀	图像质量非常好，如同人想象出的最好质量
2	良好	图像质量高，观看舒服，有干扰但不影响观看
3	可用	图像质量可接受，有干扰但不太影响观看
4	刚可看	图像质量差，干扰有些妨碍观看，希望改进
5	差	图像质量很差，妨碍观看的干扰始终存在，几乎无法观看
6	不能用	图像质量极差，不能使用

## ■ 图像压缩一般框架



# 内容大纲

- 基本概念
- 信息论基础
- 常见的无损压缩方法
- 图像压缩方法
  - 预测编码
  - 变换编码
- 图像视频压缩标准介绍



## 信息论基础

- 显示一幅图像需要多大的数据量?
- 有没有描述一幅图像且没有信息丢失的最小数据量?
- 信息度量: 对一个随机事件 $E$ , 如果它的出现概率是 $P(E)$ , 那么它包含的信息:

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

其中 $I(E)$ 称为 $E$ 的自信息。

## ■ 信息论基础

- 香农信息熵:

- 信源字母表  $A = \{a_1, a_2, \dots, a_J\}$
- 信源产生符号  $a_j$  的概率为  $P(a_j)$ , 并且  $\sum_{j=1}^J P(a_j) = 1$
- 该信源的平均信息 (熵)  $H = -\sum_{j=1}^J P(a_j) \log P(a_j)$

- 每像素平均比特数 (bit per pixel, bpp)

$$L_{\text{avg}} = -\sum_{j=0}^{255} l(a_j) P(a_j)$$

$$\text{其中 } P(a_j) = \frac{N_{a_j}}{MN}, j = 0, 1, 2, \dots, 255.$$



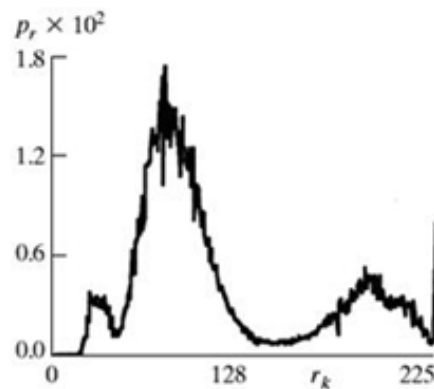
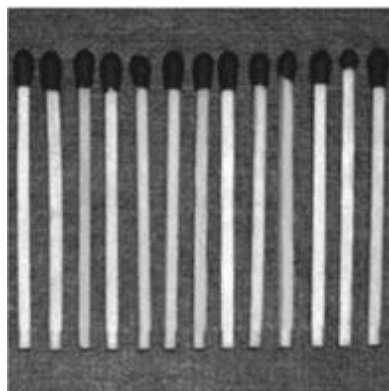
## ■ 信息论基础

- 香农第一定理（无噪声，独立信源）

$$\frac{H}{\log r} + \frac{1}{N} > \frac{L_{avg,N}}{N} \geq \frac{H}{\log r}, \quad \lim_{N \rightarrow \infty} \left[ \frac{L_{avg,N}}{N} \right] = H$$

- 对2为底的数字图像而言

$$L_{avg} \geq \tilde{H} = - \sum_{k=0}^{255} P_r(r_k) \log_2 P_r(r_k)$$



# 内容大纲

- 基本概念
- 信息论基础
- 常见的无损压缩方法
- 图像压缩方法
  - 预测编码
  - 变换编码
- 图像视频压缩标准介绍

## 常见的无损压缩方法

- 变长编码
  - 根据信息熵，把短码字赋予出现概率最大的灰度级；
  - 霍夫曼(Huffman)编码
  - 算术编码
  - 其它变长编码
- LZW 编码，行程编码
- 位平面编码
- 无损预测编码



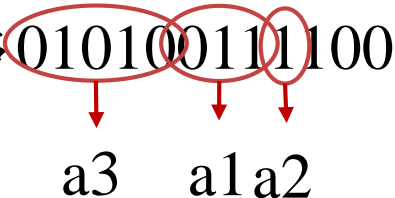
# 霍夫曼编码

- I. 符号概率排序，并将最低概率的符号联结为一个单一符号；
- II. 对每个化简后的信源进行编码，从最小的信源开始，一直编码到原始的信源；

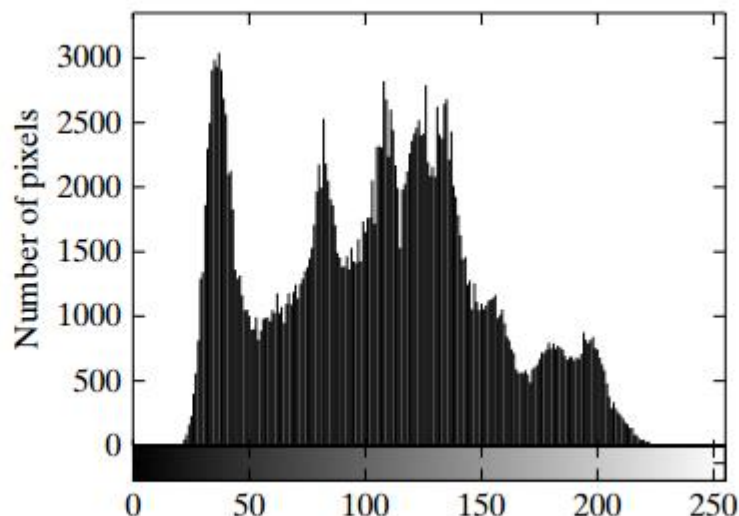
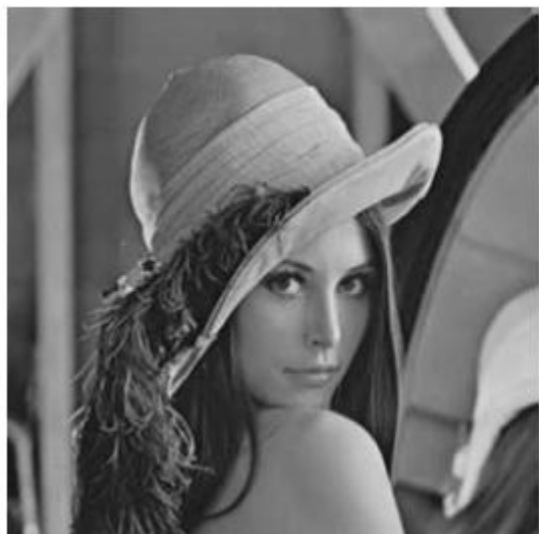
Original source		Source reduction				
Symbol	Probability	1	2	3	4	
$a_2$	0.4	0.4	0.4	0.4	0.6 0.4	
$a_6$	0.3	0.3	0.3	0.3		
$a_1$	0.1	0.1	0.2	0.3		
$a_4$	0.1	0.1				
$a_3$	0.06	0.1	0.1			
$a_5$	0.04					

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
$a_2$	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
$a_6$	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
$a_1$	0.1	011	0.1	011	0.2	010	0.3	01		
$a_4$	0.1	0100	0.1	0100	0.1	011				
$a_3$	0.06	01010	0.1	0101						
$a_5$	0.04	01011								

## 霍夫曼解码

- 解码通过查询表的方式完成
- 例：编码串 010100111100  


Symbol	Code
$a_2$	1
$a_6$	00
$a_1$	011
$a_4$	0100
$a_3$	01010
$a_5$	01011



a b

**FIGURE 8.9** (a) A  $512 \times 512$  8-bit image, and (b) its histogram.

$$L_{avg} = 7.428, \tilde{H} = 7.3838, C = \frac{8}{L_{avg}} = 1.077, R = 1 - \frac{1}{C} = 7.15\%$$

## 算术编码

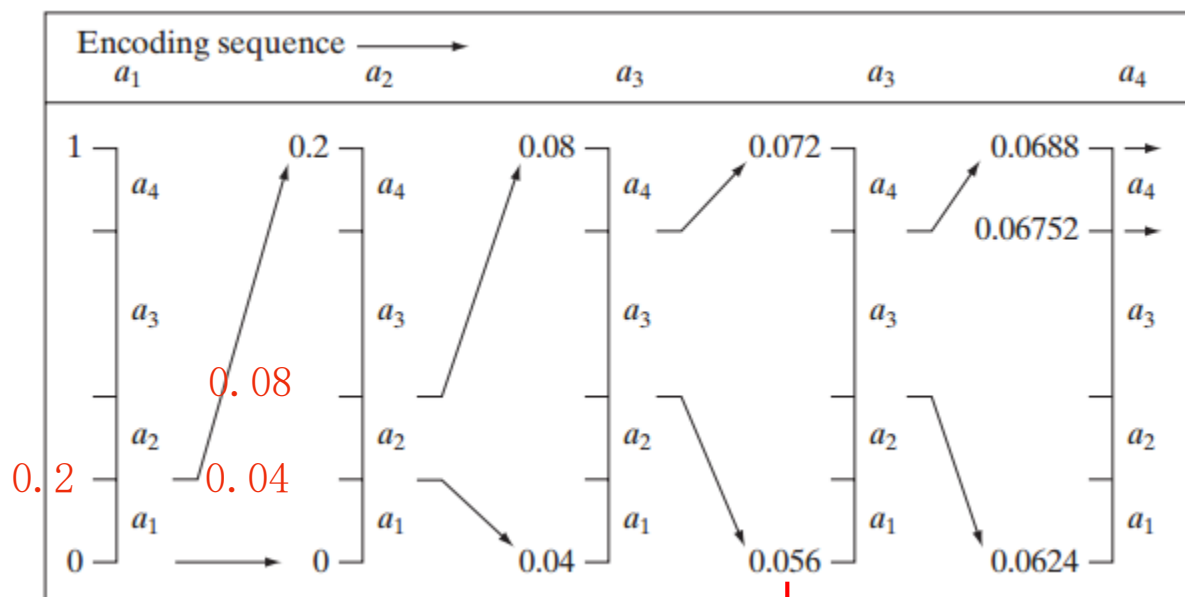
- 从整个符号序列出发，采用递推形式连续编码；
- 在算术编码中，源符号和码字间的一一对应关系并不存在。1个算术码字要赋给整个信源符号序列，而码字本身确定0和1之间的1个实数区间；
- 随着符号序列中的符号数量增加，用来代表它的区间减小而表达区间的信息单位数量变大。



# ■ ■ ■ 算术编码

Source Symbol	Probability	Initial Subinterval
$a_1$	0.2	$[0.0, 0.2)$
$a_2$	0.2	$[0.2, 0.4)$
$a_3$	0.4	$[0.4, 0.8)$
$a_4$	0.2	$[0.8, 1.0)$

**TABLE 8.6**  
Arithmetic coding  
example.



**FIGURE 8.12**  
Arithmetic coding  
procedure.

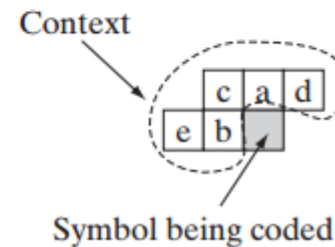
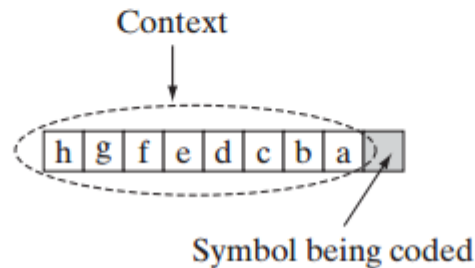
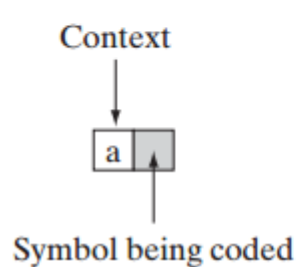
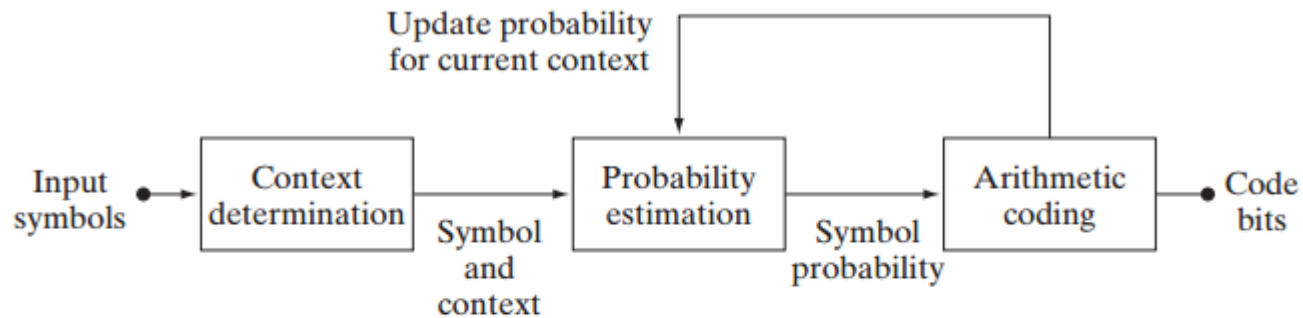
$$0.04 = 0 + (0.2 - 0) / 5 \times 1$$

$$0.08 = 0 + (0.2 - 0) / 5 \times 2$$

$$0.056 = 0.04 + (0.08 - 0.04) / 5 \times 2$$

$$0.072 = 0.04 + (0.08 - 0.04) / 5 \times 4 = 0.04 + 0.032$$

# Adaptive context dependent probability estimates



• 思考：解码过程如何实现？

## ■ LZW 编码，行程编码

- LZW 编码：消除像素间冗余

- 由Lemple和Ziv最早提出，然后由Welch充实的有专利保护的算法；
- 将原始数据中的重复字符串建立一个字符串表，然后用该重复字符串在字符串表中的索引替代原始数据达到压缩目的；
- 一个由8位组成的文件可以被编成12位的代码。在  $2^{12}$ (4096) 个可能的代码中， $2^8$ (256)个代表所有可能的单个字符（8位），剩下的 3840个代码分配给压缩过程中出现的字符串。

## LZW 编码

- LZW 编码：消除像素间冗余

- 每当表中没有的字符串第一次出现时，它就被原样保存，同时将分配给它的代码也保存；
- 之后，当这个串再次出现时，只将它的代码保存，这就去掉了文件冗余信息；
- 不但字符串表是在压缩过程中动态生成，而且字符串表也不必保存在压缩文件里，因为解压缩算法可以由压缩文件中的信息重构。
- 使用LZW的文件格式包括GIF，TIFF和PDF等。

# LZW 编码

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

$$C = \frac{128}{90} = 1.42$$



## 位平面编码

- 位平面编码：消除像素间冗余
  - 将一幅图像分解为一系列二值图像并通过二值图像压缩方法对每幅二值图像进行压缩

- 二值图像位平面

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

- 灰度编码位平面

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

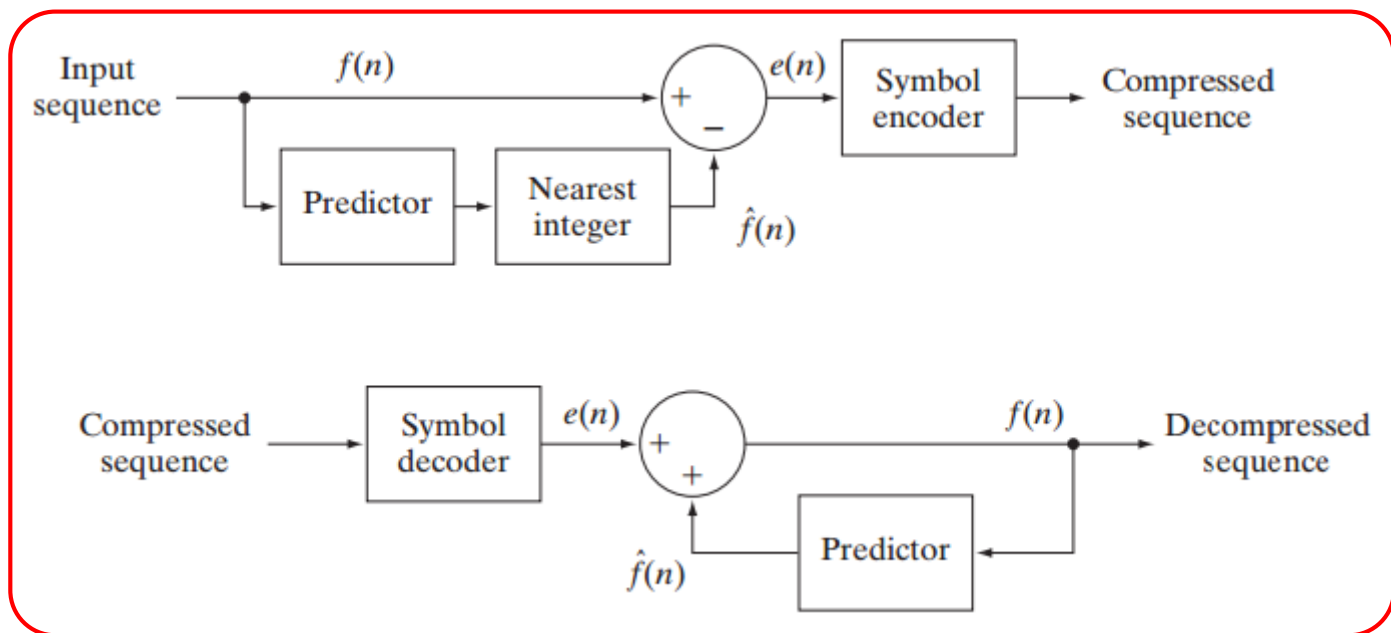
$$g_{m-1} = a_{m-1}$$

- 思考：两种编码方式的差异？



## ■ 无损预测编码

- 通过仅提取每个像素中的新信息并对它们编码来消除像素间的冗余；
- 1个像素的新信息定义为该像素的当前值与预测值的差；
- 正是由于像素间有相关性，所以才使预测成为可能。



## ■ 无损预测编码

- 当输入图像的像素序列 $f_n$ 逐个进入编码器，预测器根据过去的输入产生当前输入像素的估计值。预测器的输出舍入成最近的整数 $\hat{f}_n$ 并被用来计算预测误差：

$$e_n = f_n - \hat{f}_n$$

- 该误差用符号编码器借助变长码进行编码以产生压缩数据流的下一个元素。然后解码器根据接收到的变长码字重建 $e_n$ ，并执行下列操作：

$$f_n = e_n + \hat{f}_n$$

## ■ 无损预测编码

- 一般情况下，可将 $m$ 个先前的像素进行线性组合以得到预测：

$$\hat{f}_n = [\sum_{i=1}^m a_i f_{n-i}]$$

- 其中， $m$ 是线性预测器的阶， $[\cdot]$ 是舍入函数， $a_i$ 是预测系数。
- 最简单的一维线性预测编码是一阶( $m = 1$ )，此时

$$\hat{f}_n = [a_1 f_{n-1}]$$

# 内容大纲

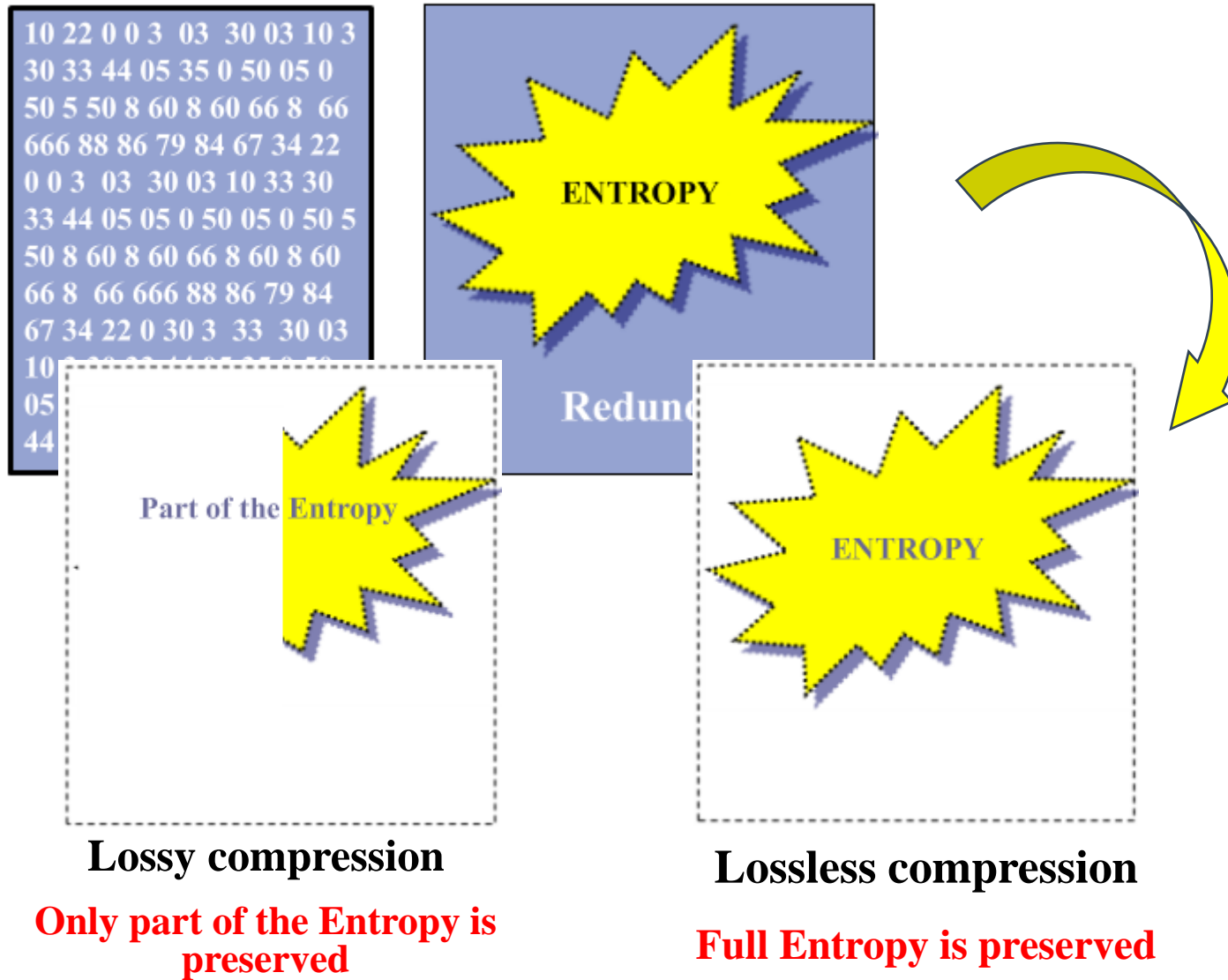
- 基本概念
- 信息论基础
- 常见的无损压缩方法
- 图像压缩方法
  - 预测编码
  - 变换编码
- 图像视频压缩标准介绍

## 图像压缩

- 大部分是有损压缩；
- 牺牲图像复原的准确度以换取压缩能力的增加；
- 如果产生的失真可以容忍，则压缩能力的增加是有效的；

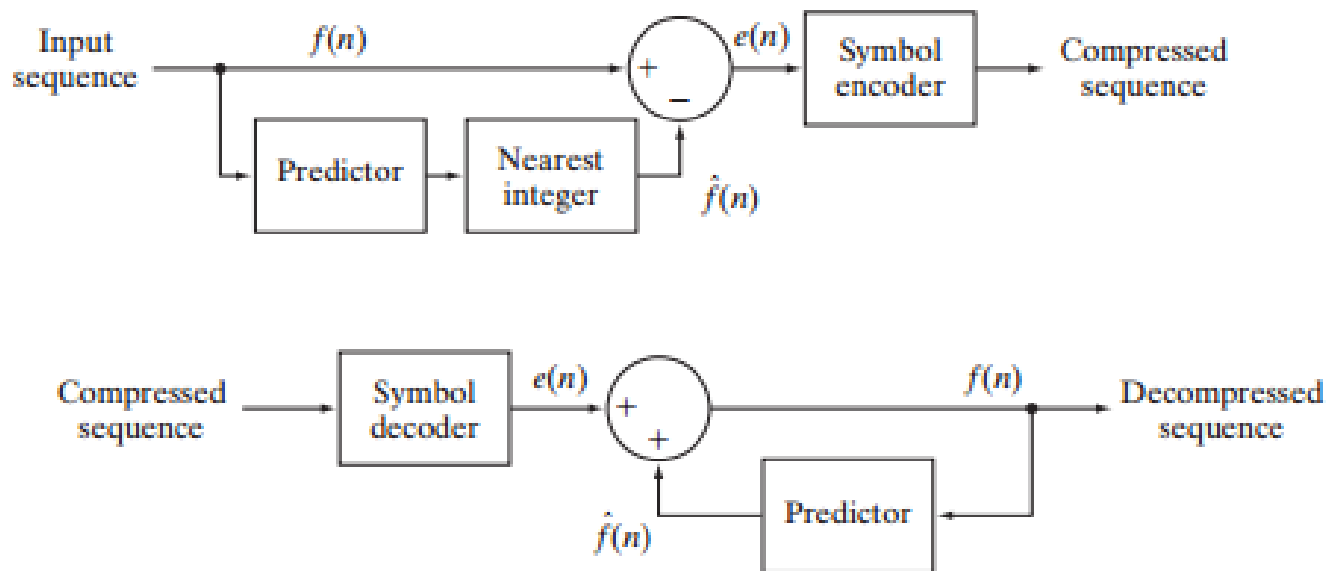
# 有损和无损压缩

From the view of Entropy theory



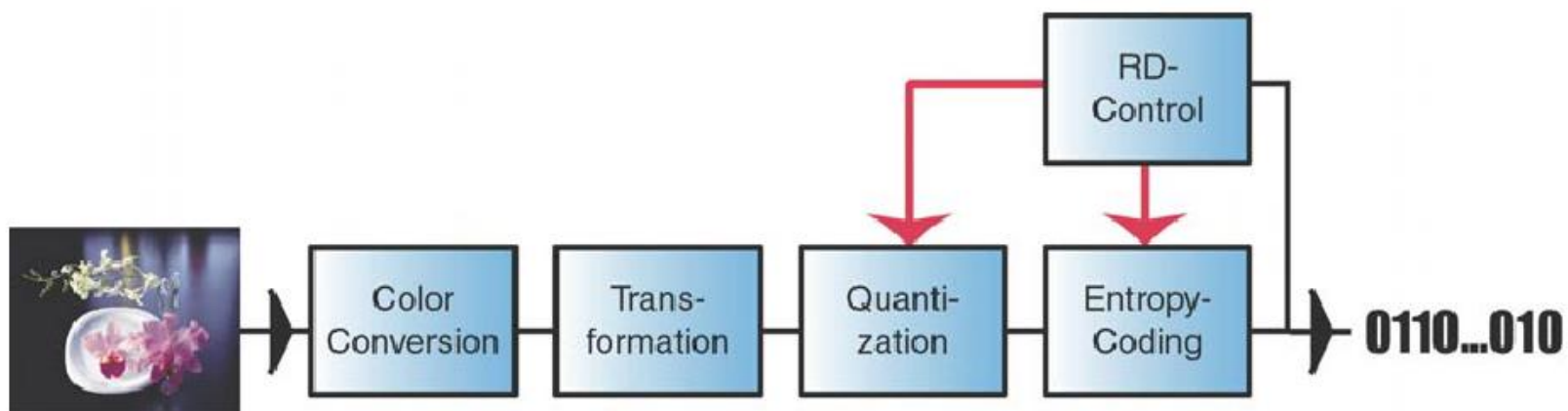
## ■ 图像压缩两大框架介绍

- 预测编码



## ■ 图像压缩两大框架介绍

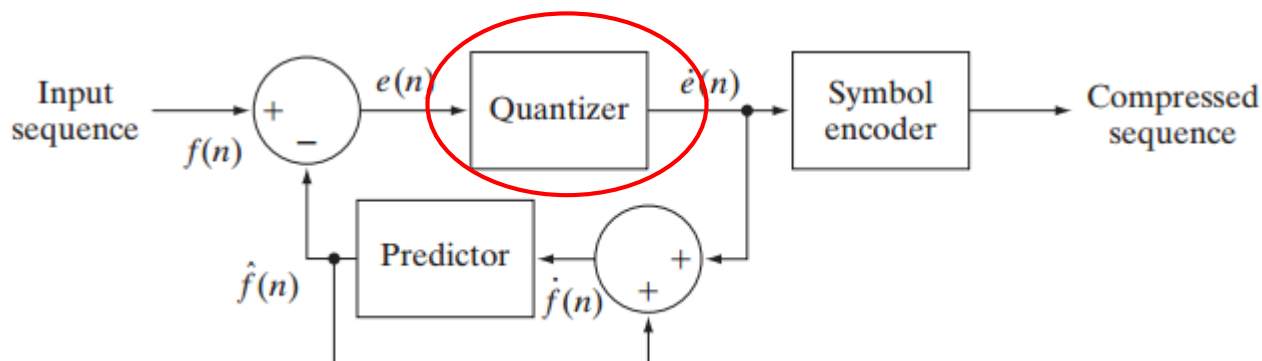
- 变换编码





## ■ 有损预测编码

- 量化器插在符号编码器和预测误差产生处之间，把原来无损编码器中的整数舍入模块吸收进来；
- 量化器将预测误差映射进输出  $\dot{e}_n$  中， $\dot{e}_n$  确定了有损预测编码中的压缩量和失真量；
- 反馈环的输入是过去预测和与其对应的量化误差的函数： $\dot{f}_n = \dot{e}_n + \hat{f}_n$



## ■ 有损预测编码

- 最优预测器：
  - 在绝大多数预测编码中用到的最优预测器在满足限制条件：

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f_n$$

$$\hat{f}_n = \sum_{i=1}^m a_i f_{n-i}$$

的情况下能最小化编码器的均方预测误差

$$E\{e_n^2\} = E\{\|f_n - \hat{f}_n\|^2\}$$

## ■ 有损预测编码

- 最优预测器：
  - 最优准则是最小化均方预测误差，设量化误差可以忽略（ $\dot{e}_n \approx e_n$ ），并用线性组合进行预测；
  - 最优预测器设计的问题简化为比较直观地选择  $m$  个预测系数以最小化下式的问题：

$$\operatorname{argmin}_{\hat{a}_i} E\{e_n^2\} = \|f_n - \sum_{i=1}^m a_i f_{n-i}\|^2$$

- 基于上述条件的预测编码方法称为差值脉冲码调制法（DPCM）

## ■ 有损预测编码

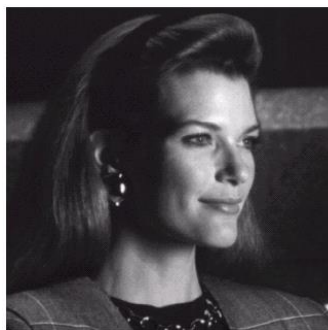
### • Example 8.24

$$\hat{f}(x, y) = 0.97f(x, y - 1)$$

$$\hat{f}(x, y) = 0.5f(x, y - 1) + 0.5f(x - 1, y)$$

$$\hat{f}(x, y) = 0.75f(x, y - 1) + 0.75f(x - 1, y) - 0.5f(x - 1, y - 1)$$

$$\hat{f}(x, y) = \begin{cases} 0.97f(x, y - 1) & \text{if } \Delta h \leq \Delta v \\ 0.97f(x - 1, y) & \text{otherwise} \end{cases}$$



1阶



2阶



3阶

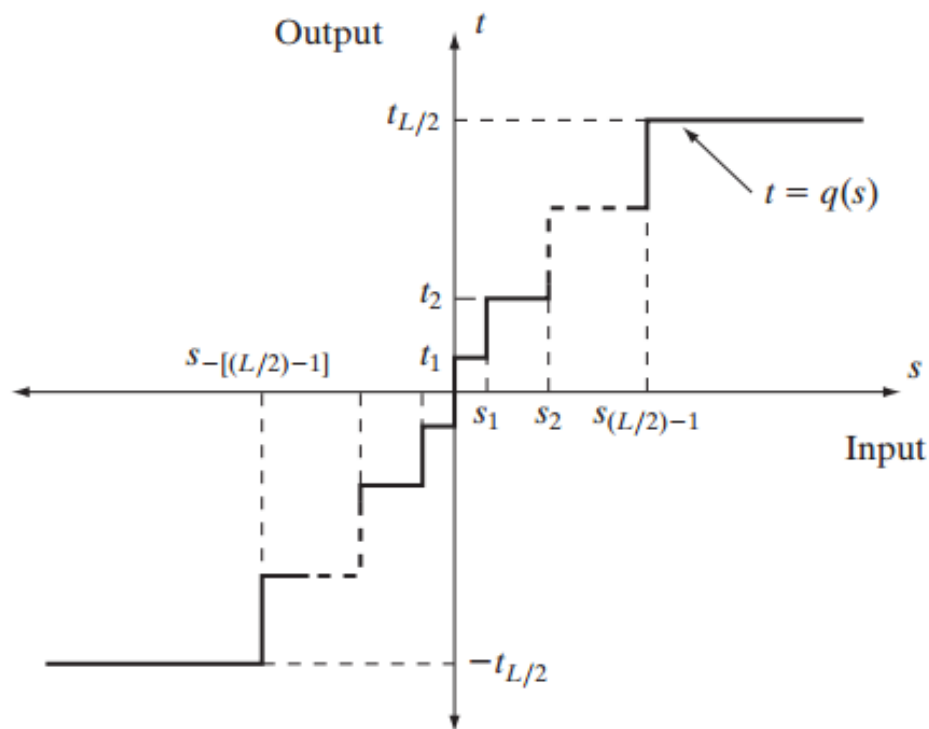


4阶



## ■ 有损预测编码

- 最优量化（心理视觉冗余）
- $t = q(s)$  是  $s$  的奇函数
- 断点定义了函数的不连续性，被称为量化器的判决和重构级



## ■ 有损预测编码

- 量化器的设计就是要在给定优化准则和 输入概率密度函数 $p(s)$ 的条件下选择最优的 $s_i$ 和 $t_i$ ，优化准则可以是统计的或心理视觉的准则；
- 用最小均方误差准则下 $E\{(s_i - t_i)^2\}$ :

$$\int_{s_{i-1}}^{s_i} (s - t_i) p(s) ds = 0, i = 1, 2, \dots, \frac{L}{2}$$

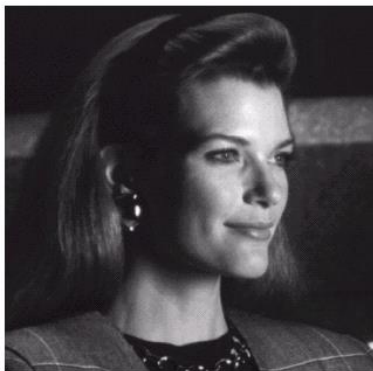
$$s_{-i} = -s_i, t_{-i} = -t_i$$

$$s_i = \begin{cases} 0 & i = 0 \\ \frac{t_i + t_{i+1}}{2} & i = 1, 2, \dots, \frac{L}{2} - 1 \\ \infty & i = \frac{L}{2} \end{cases}$$

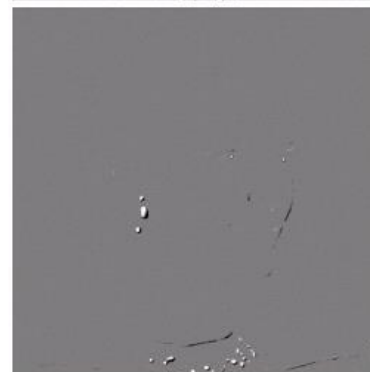
判定层在重构层的一半处

Lloyd-  
Max  
量化器

## ■ Example: 量化和复原

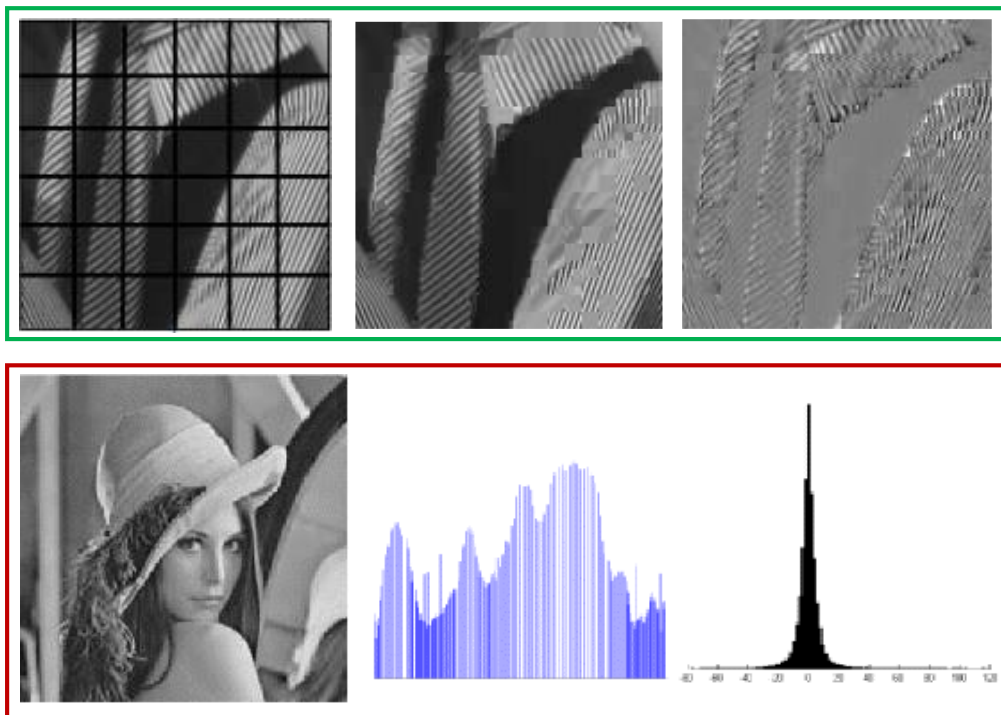


三阶预测  
器 + 2,4,8  
层Lloyd-  
Max量化  
器



## ■ 图像的变换编码

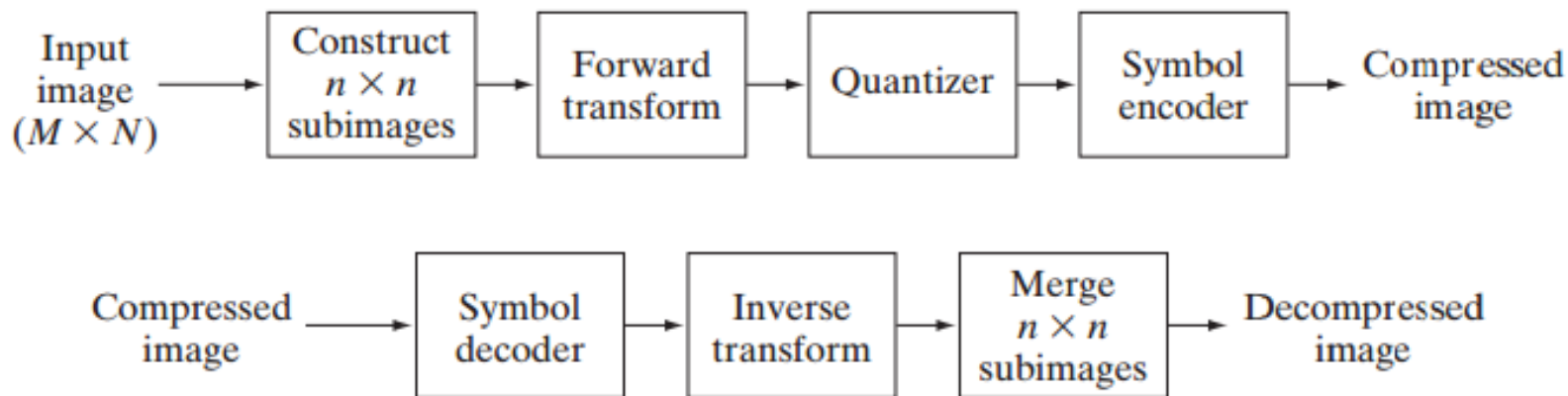
- 在图像的变换域进行的编码方法
- 通过正交变换降低像素间的相关性（像素间冗余）
- 可逆的线性变换



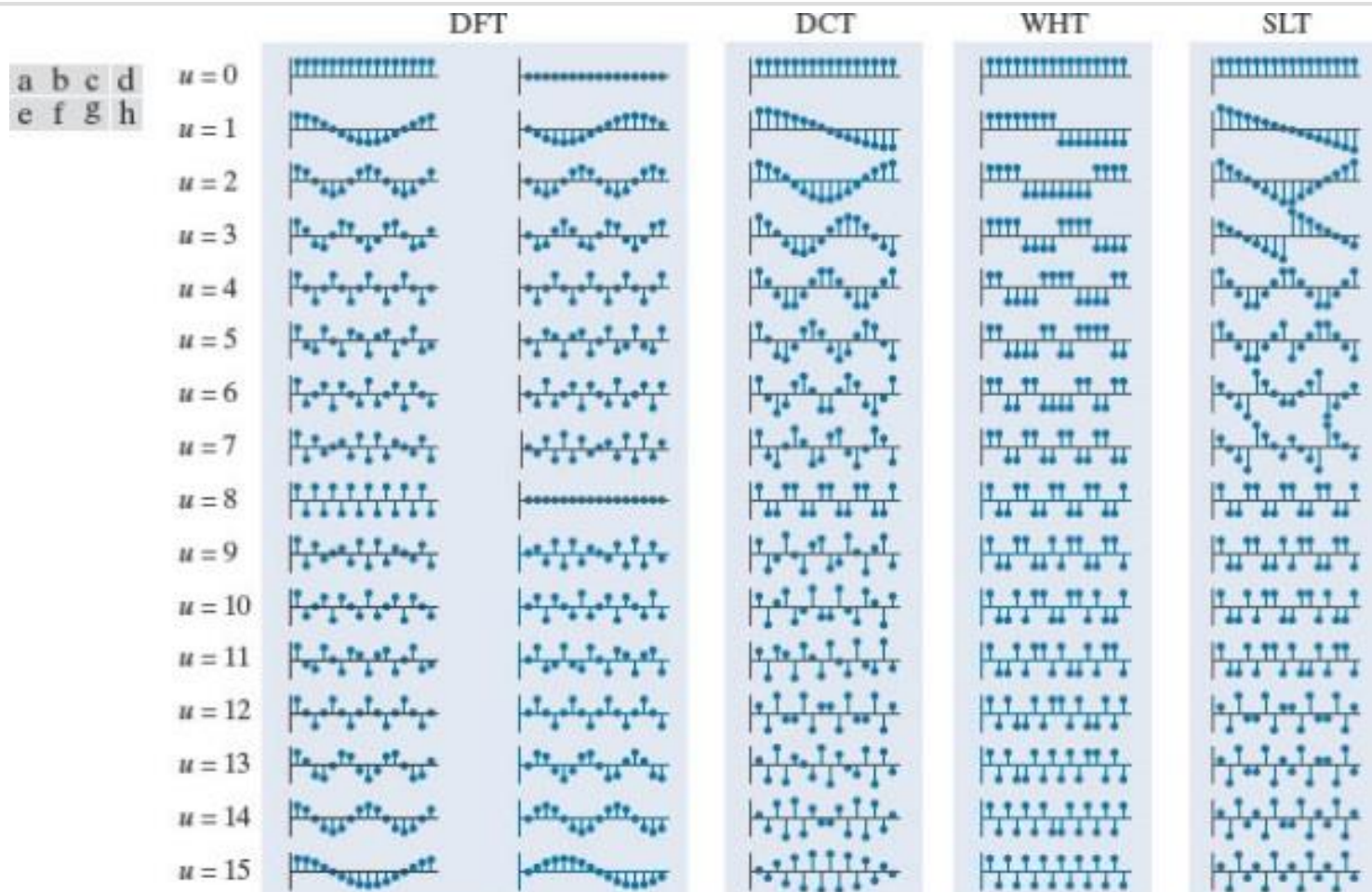


## ■ 图像的变换编码

- 变换系数值进行量化和编码
- 一般都是有损压缩
- 一般针对子图 (sub-image) 操作



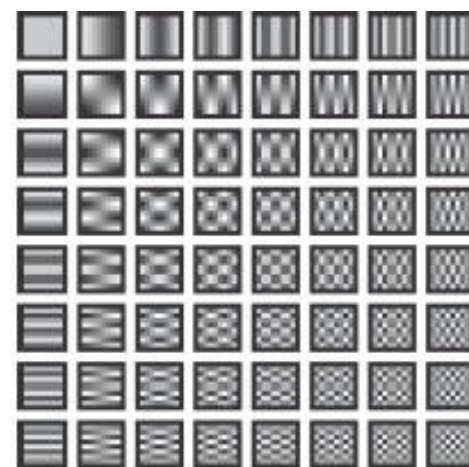
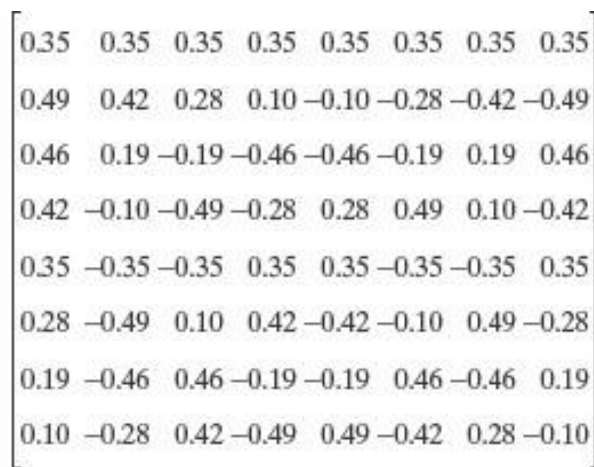
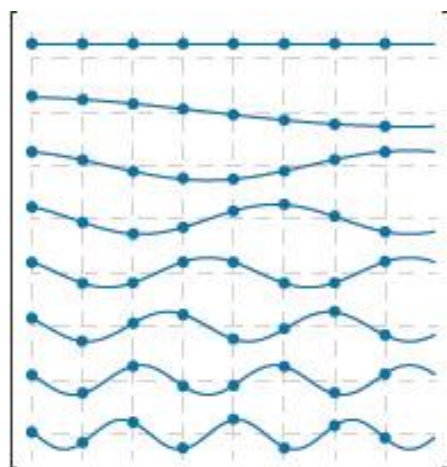
# 各种变换基总览



Basis vectors (for  $N = 16$ ) of some commonly encountered transforms: (a) Fourier basis (real and imaginary parts), (b) discrete Cosine basis, (c) Walsh-Hadamard basis, (d) Slant basis, (e) Haar basis, (f) Daubechies basis, (g) Biorthogonal B-spline basis and its dual, and (h) the standard basis, which is included for reference only (i.e., not used as the basis of a transform).

# 离散余弦变换

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



a b c

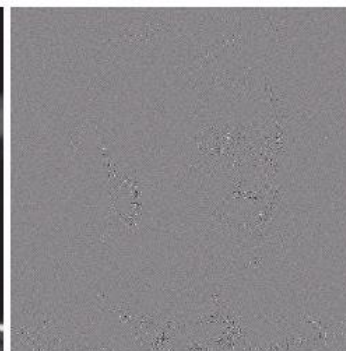


## 变换编码比较实验

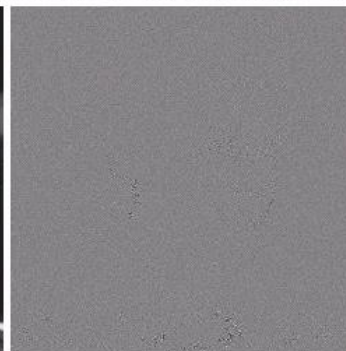
选择合适的变换非常重要

- Size:  $512 \times 512$
- Block size:  $8 \times 8$
- Transform:  
DFT, WHT, DCT
- Retain: 50%  
coefficients
- RMSE:

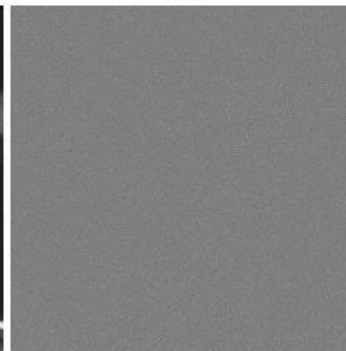
$$R_{DFT} > R_{WHT} > R_{DCT}$$



$$R_{DFT} = 1.28$$



$$R_{WHT} = 0.86$$



$$R_{DCT} = 0.68$$

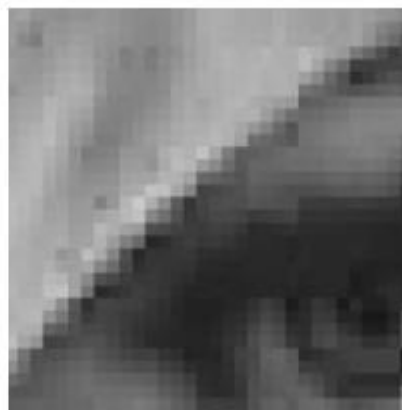
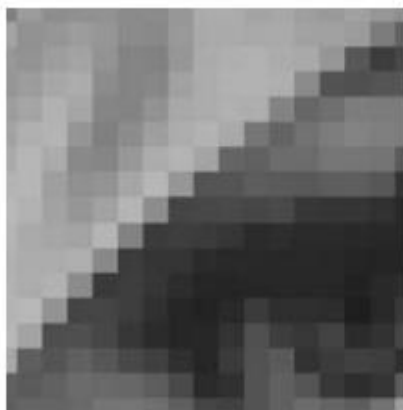
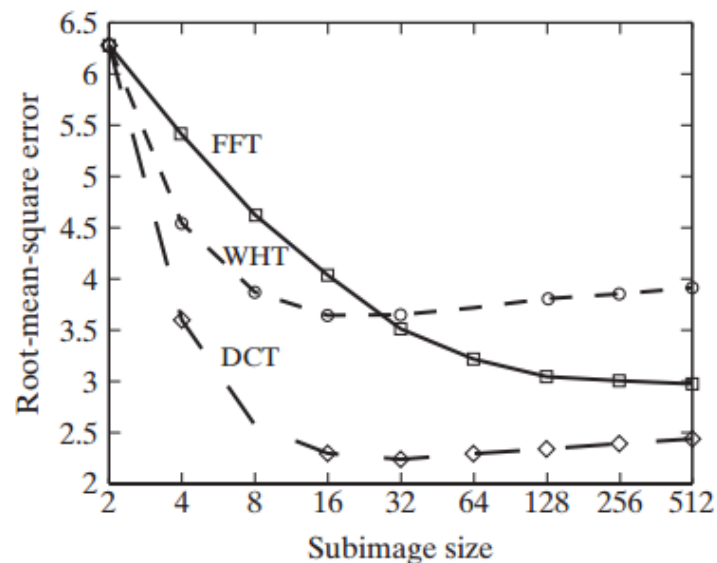


## 子图尺寸的选择

- 子图像尺寸是影响变换编码误差和计算复杂度的一个重要因素；
- 一般满足下列2个条件
  - 相邻子图像之间的相关（冗余）减少到可接受
  - 子图像的长和宽都是2的整数次幂。这主要是为了简化对子图像变换的计算
- 最常用的子图像尺寸是 $8 \times 8$ 和 $16 \times 16$

## 子图大小比较实验

- Block size: 2, 4, 8, 16, 32
- Transform:  
DFT, WHT, DCT
- Retain: 25% coefficients





## 比特分配

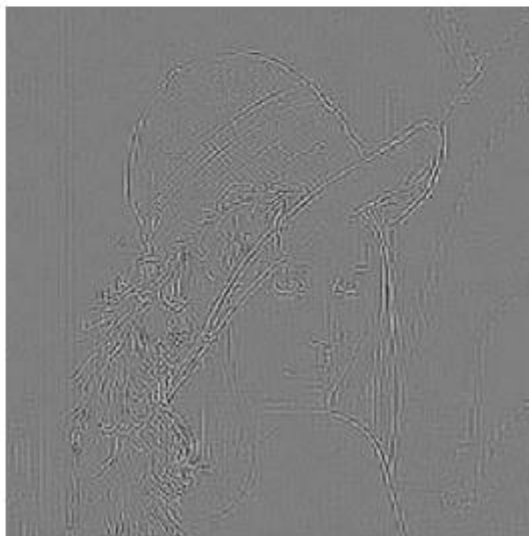
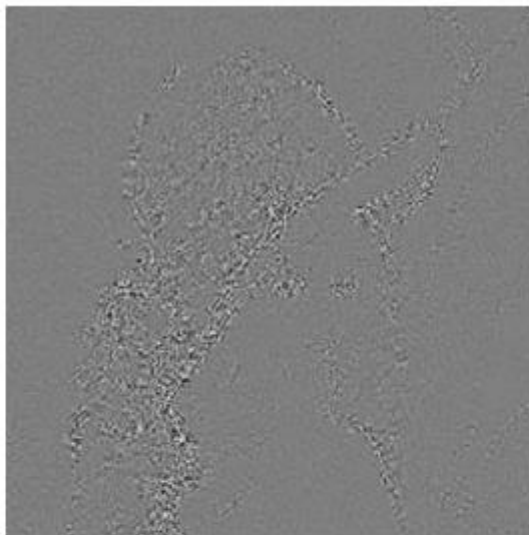
- 截断误差和2个因素有关：
  - 截除的变换系数的数量和相对重要性
  - 用来表示所保留系数的精度
- 保留系数选择的2个常用准则：
  - 最大方差准则，称为分区编码
  - 最大幅度准则，称为阈值编码
- 整个对变换子图像的系数截断、量化和编码的全过程称为比特分配





## 比特分配

DCT: 阈值编码  
优于分区编码



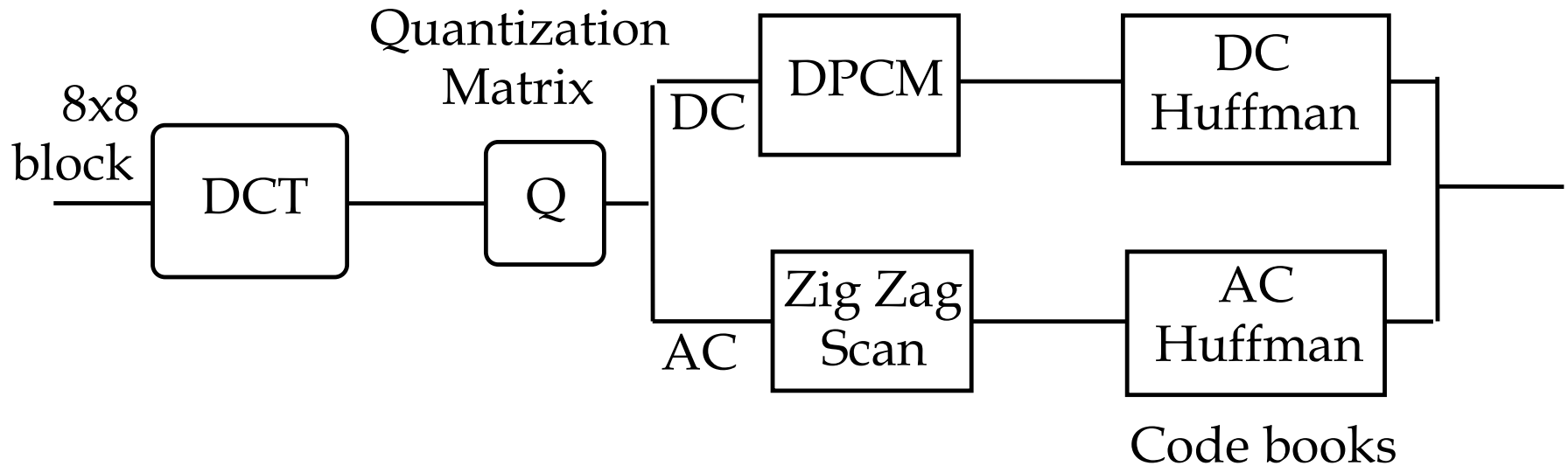
a b  
c d

**FIGURE 8.28**

Approximations of Fig. 8.9(a) using 12.5% of the  $8 \times 8$  DCT coefficients: (a)—(b) threshold coding results; (c)—(d) zonal coding results. The difference images are scaled by 4.

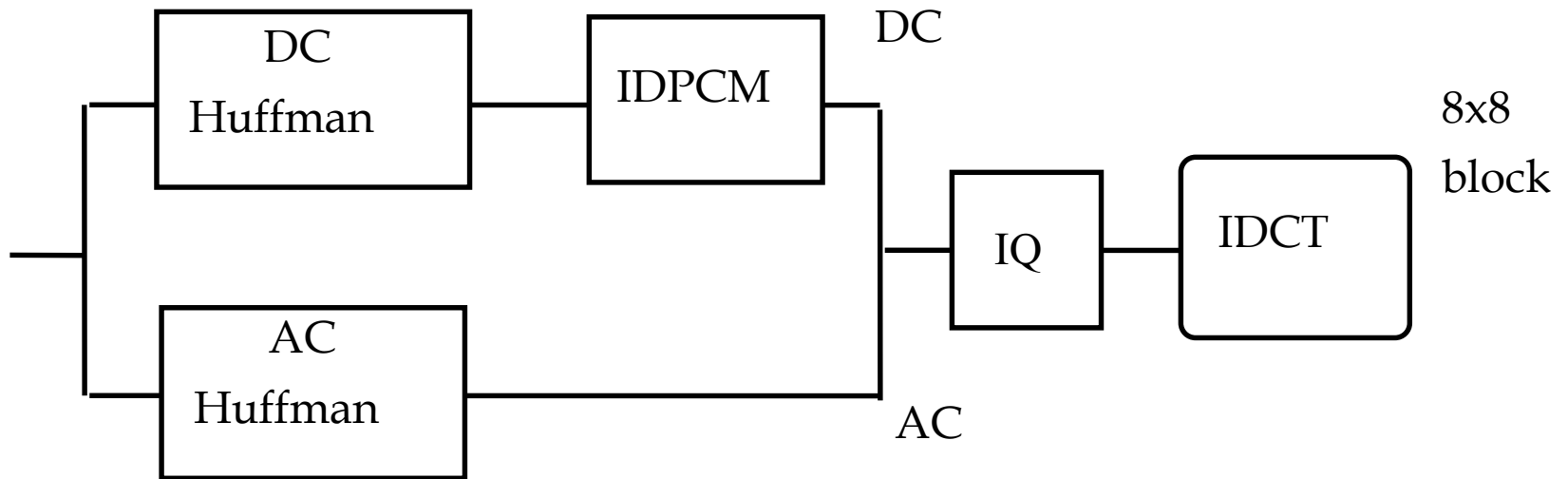


# ■ JPEG Image Coding Algorithms



JPEG Encoding Process

# ■ ■ JPEG Decoding



JPEG Decoding Process

## An example of JPEG encoder

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	-76	-73	-67	-62	-58
67	61	68	-65	-69	-62	-38	-19
79	65	60	-66	-69	-60	-15	16
85	71	64	-65	-70	-57	-6	26
87	79	69	-61	-67	-60	-24	-2
			-49	-63	-60	-40	-60
			-43	-57	-60	-24	-58
			-41	-49	-55	-40	-58
			-415	-29	-62	25	55
			7	-21	-62	9	11
			-46	8	77	-25	-30
			-50	13	35	-15	-9
			11	-8	-13	-2	-1
			-10	1	3	-3	-1
			-4	-1	2	-1	2
			-1	-1	-1	-2	-1

Block DCT

# ■ An example of JPEG encoder

-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

$$\hat{T}(0,0) = \left\lceil \frac{T(0,0)}{Z(0,0)} \right\rceil$$

$$= \left\lceil \frac{-415}{16} \right\rceil = -26$$

Quantize

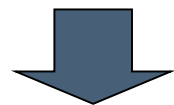
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

-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

# ■ ■ An example of JPEG encoder

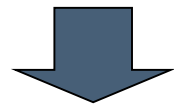
-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63



Coefficients Serialize

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



Huffman Encoding (DC/AC)

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



## 小波与图像压缩

- 什么是小波变换?
- 为什么要用小波变换来做图像压缩?
  - 不需要对图像进行分块，避免产生块失真；
  - 对传输中的容错性较好；
  - 更易于实现图像的渐进传输；
  - 能够更加符合人类视觉系统（HVS）模型；



## 可分级图像编码 (Scalable Image Coding)

- 在可分级图像压缩中，通过对比特流的逐步解码从而提供具有不同品质的图像，例如：空间分辨率的分级，图像质量分级，或者图像分辨率及质量分级。
- In scalable compression, the bitstream can be progressively decoded to provide different versions of the image in terms of *spatial resolutions (spatial scalability)*, *quality levels [ signal-to-noise ratio (SNR) scalability ]*, or *combinations of the spatial and the quality scalabilities*.

## 感兴趣区域 (ROI) 随机访问

- 用户可以随意指定感兴趣区域进行访问和编码。
- 在这些区域，用户可以在压缩时指定特定的压缩质量，或在恢复时指定特定的解压缩要求。
- 在有些情况下，图像中只有一小块区域对用户是有用的，对这些区域，采用低压缩比，而感兴趣区域之外采用高压缩比，在保证不丢失重要信息的同时，又能有效地压缩数据量。



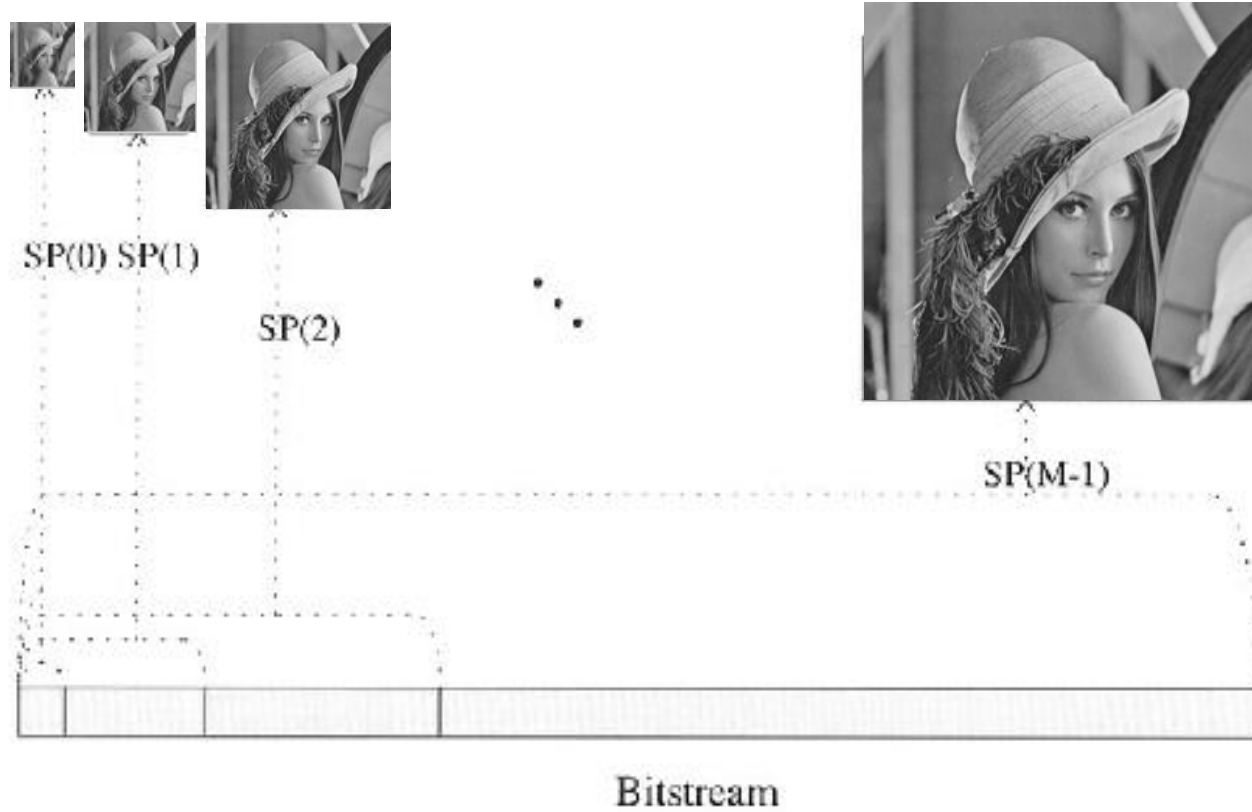


Figure 1. *N Layers of Spatial Scalability*

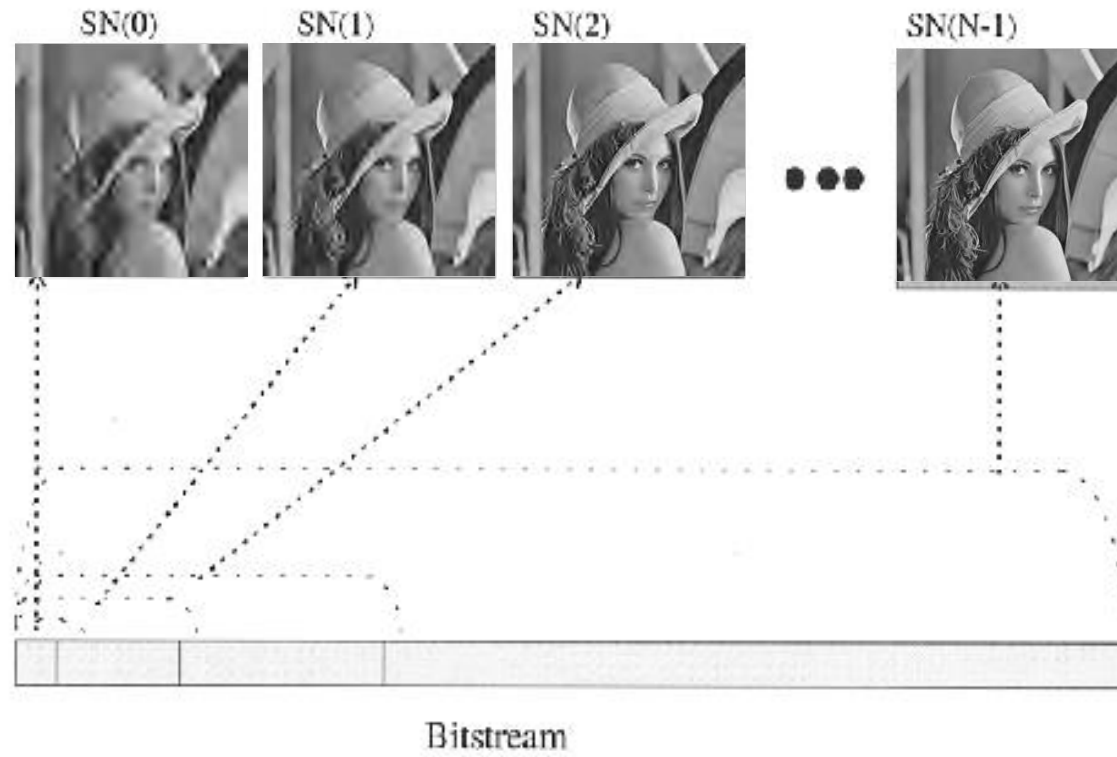


Figure 2.  $N$  layers of SNR(quality) Scalability

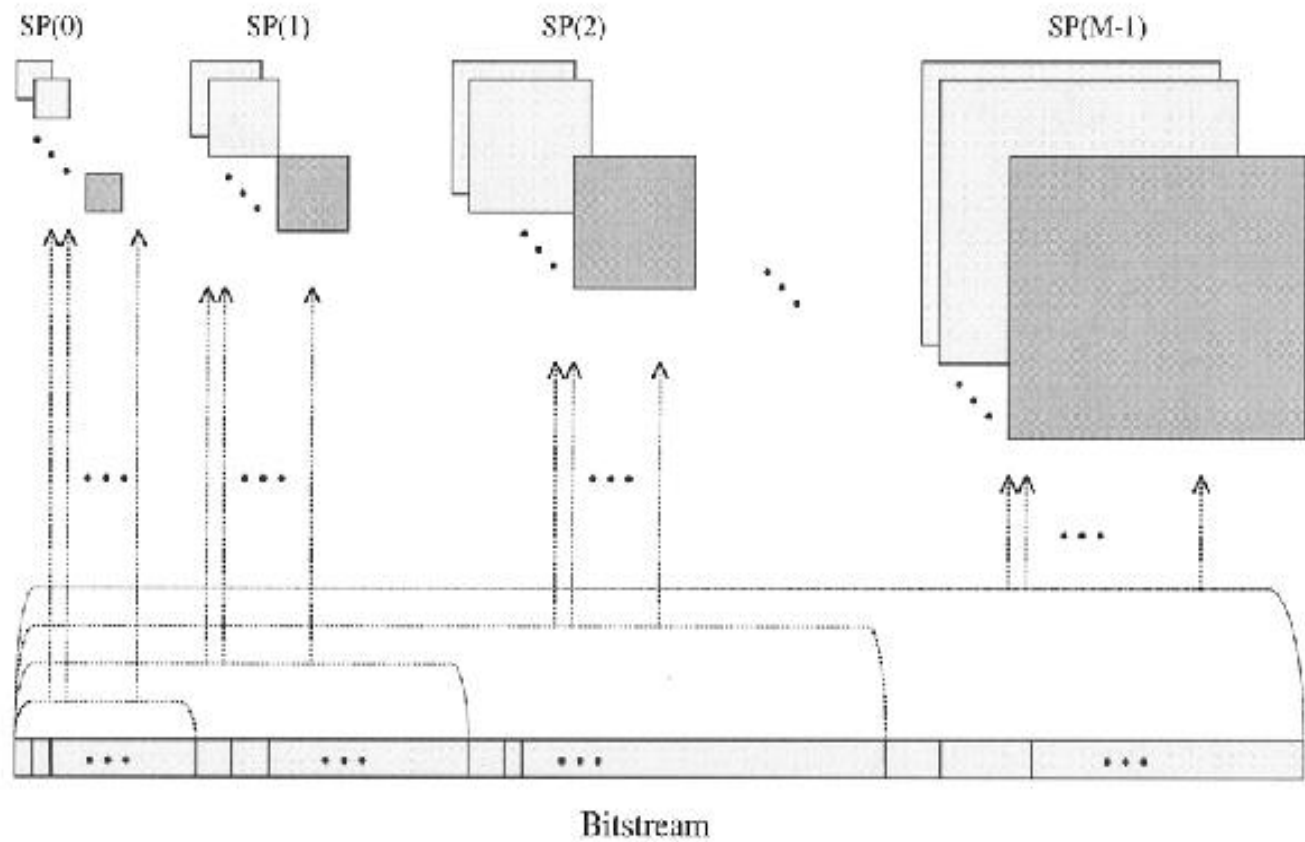
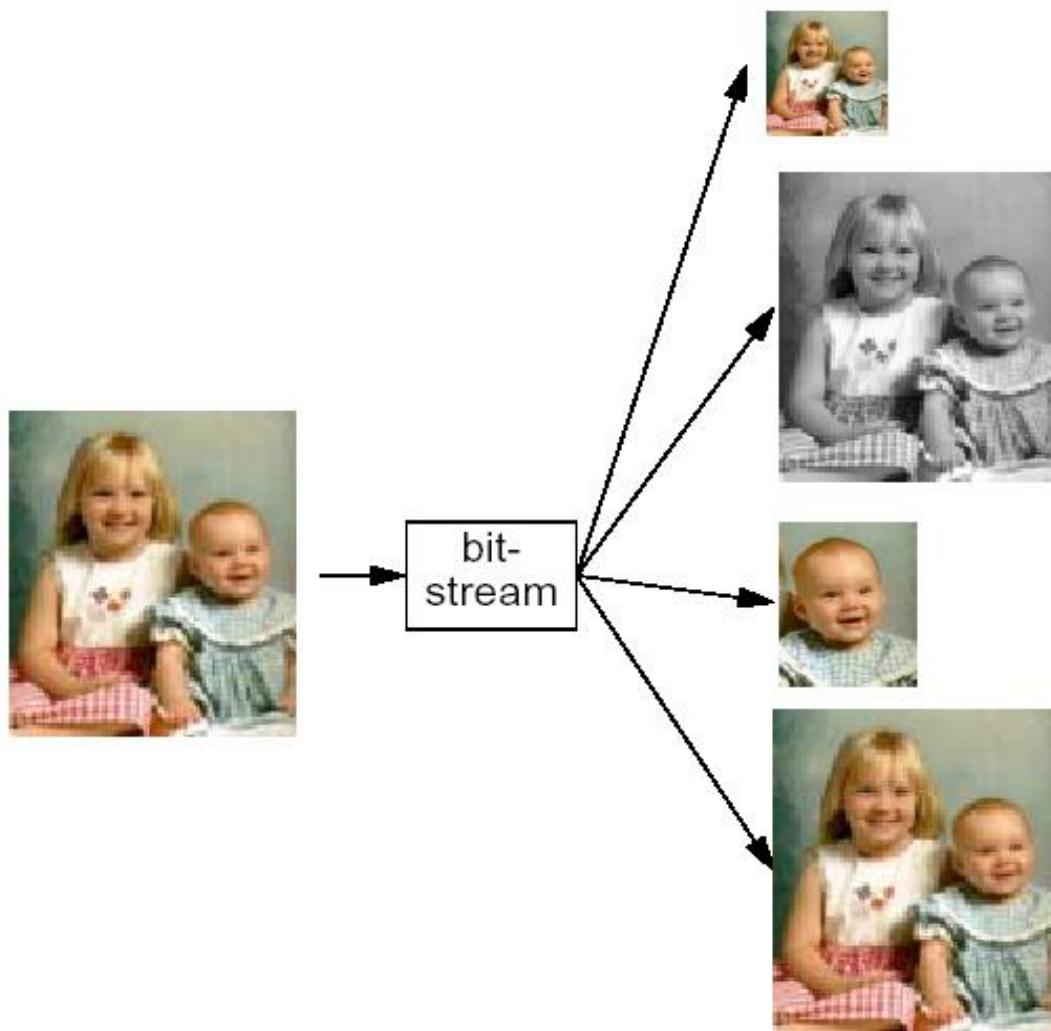


Figure 3.  $N \times M$  Layers of Spatial/Quality Scalability



## ■ 嵌入式零树小波编码 (EZW)

- 在倍频小波分解中，每个高频子带中的系数，在比它更高频的子带中都有4个与它空间位置相对应的系数，称之为小波变换的一种树形结构；
- Lewis和Knowles于1992年，首次介绍了小波变换中的这种树形数据结构表示；
- Shapiro, J.M.于1993年将这种树形数据结构称为“零树” (zerotree)，并开发了嵌入式零树小波 (Embedded Zerotree Wavelet, EZW) 的图像编码算法。

# The Zerotree

## 1.2.1 Formation of Wavelet Trees

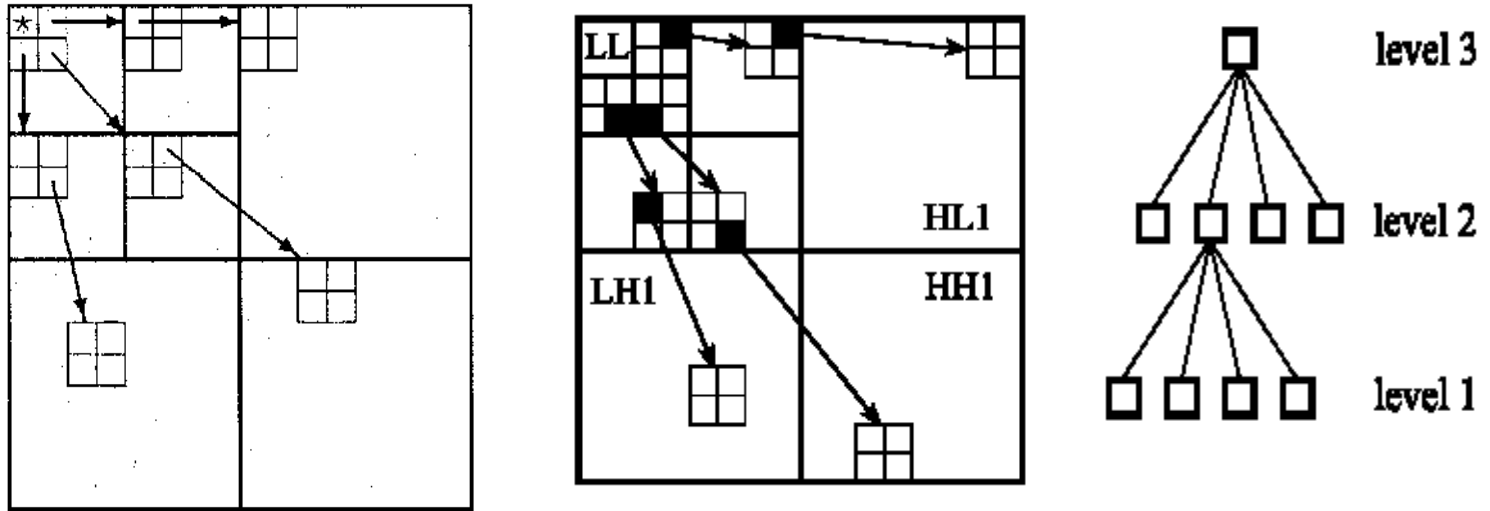


Figure 6. The parent-child relationship of wavelet coefficients  
( Defined in EZW and SPIHT algorithm )

## The Definition of Zerotree

- A wavelet coefficient  $x$  is said to be in significant with respect to a given threshold  $T$  if  $|x| > T$ ;
- The **zerotree** is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold  $T$ , then all wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to  $T$ .
- *Empirical evidence suggests that this hypothesis is often true!*



# Why Zerotree ?

## *Three important observations:*

- I. Natural images in general have a low pass spectrum.
- II. Large wavelet coefficients are more important than smaller wavelet coefficients.
- III. Wavelet coefficients decrease with scale.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

63							
			-13				
		3				3	9
						3	2
				4	6		
				3	-2		





# The Two styles to scan the coefficients

(1) In EZW and SPIHT:

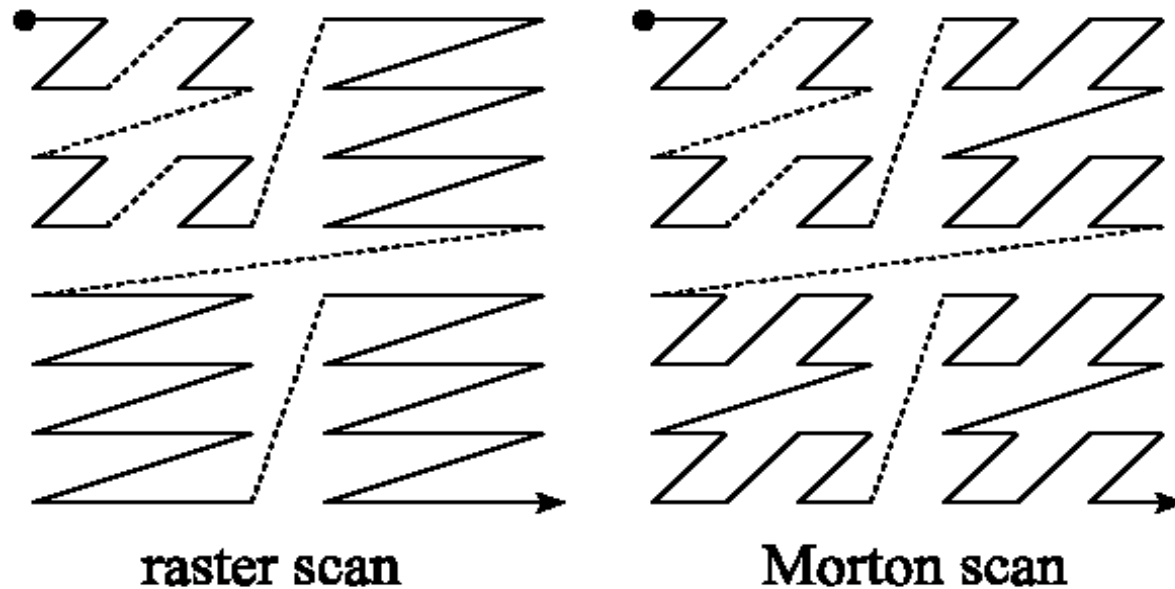
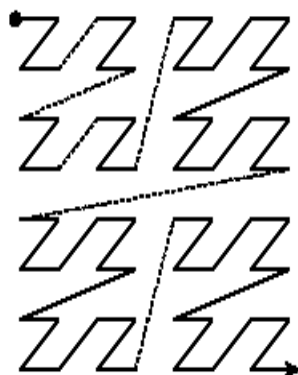


Figure 9: *The scan order defined in EZW and SPIHT*

# 1.5 An Example for EZW Coding

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4



EZW Coding

EZW Decoding

Symbol Notation:

P: Positive

N: Negative

Z: Isolated Zero

T: Zerotree

D1: PNZT PTTT TZTT TTTT TPTT

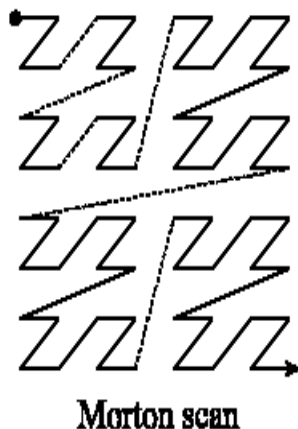
S1: 1010

设定阈值 $T_0=32$ 开始, 当符号T出现时, 它的所有子孙节点将不再扫描。

[32, 48)  $\rightarrow$  0, [48, 63)  $\rightarrow$  1, So 63 $\rightarrow$ 1(56), 34 $\rightarrow$ 0(40), 49 $\rightarrow$ 1(56), 47 $\rightarrow$ 0(40). 并重新排序为: 63, 49, 34, 47.

## 1.5 An Example for EZW Coding

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4



EZW Coding

EZW Decoding

Symbol Notation:

P: Positive

N: Negative

Z: Isolated Zero

T: Zerotree

D2: ZTNP TTTT TTTT

S2: 100110

阈值减半，即新阈值为16。在这次的扫描过程中，只会扫描那些在前一次扫描过程中没有被标记为显著系数的位置。

现在，在子序列中包含的系数幅值按顺序排列为：(63(60)，49(52)，34(36)，47(44)，31(28)，23(20))

# 1.5 An Example for EZW Decoding

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

56	-40	56	0	0	0	×	×
0	0	0	0	0	0	×	×
0	0	×	×	×	×	×	×
0	0	×	×	×	×	×	×
×	×	0	40	×	×	×	×
×	×	0	0	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

EZW Coding

EZW Decoding

Symbol Notation:

P: Positive

N: Negative

Z: Isolated Zero

T: Zerotree

D1:            PNZT    PTTT    TZTT    TTTT    TPTT

S1:            10            1                            0

解码器阈值从32开始, P1->[48, 63]->56, N0->[32, 47]->-40, Z->0, T->0, P1->56 .....

## 结果比较



• 0.36bpp (75倍)

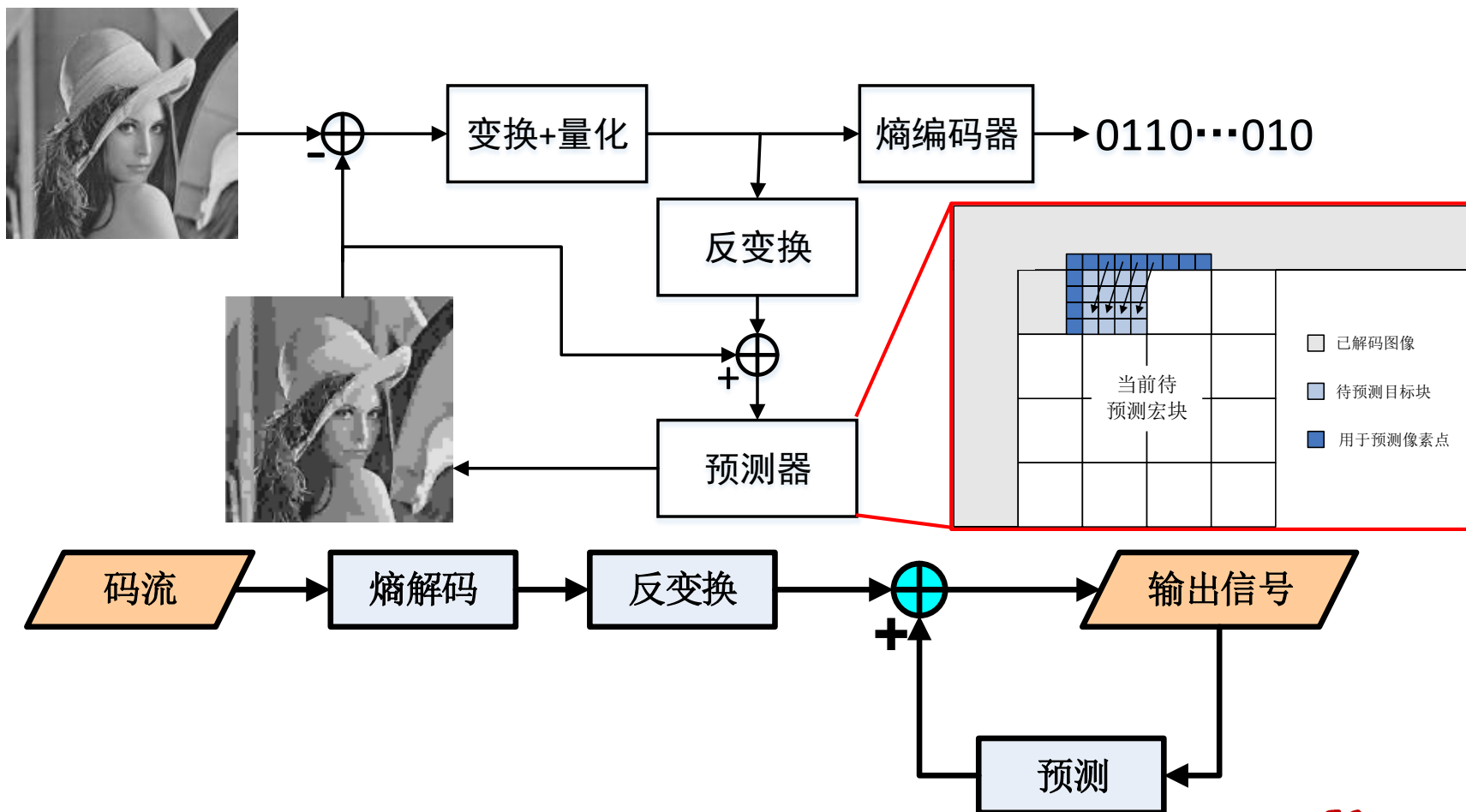
## 结果比较








# ■ 预测编码与变换编码相结合

## • H.264(帧内压缩)



## 预测编码与变换编码相结合

- 预测编码存在的问题
  - 描述图像中性质不同的区域?  
小波, 匹配追踪, .....   
**图像分解**
  - 如何提高预测的准确性?  
增加多种预测模式, .....   
**多分量分别预测**
  - 提高压缩后处理效果?  
简单去方块   
**3D 变换域联合滤波**



## ■ ■ 多分量预测编码

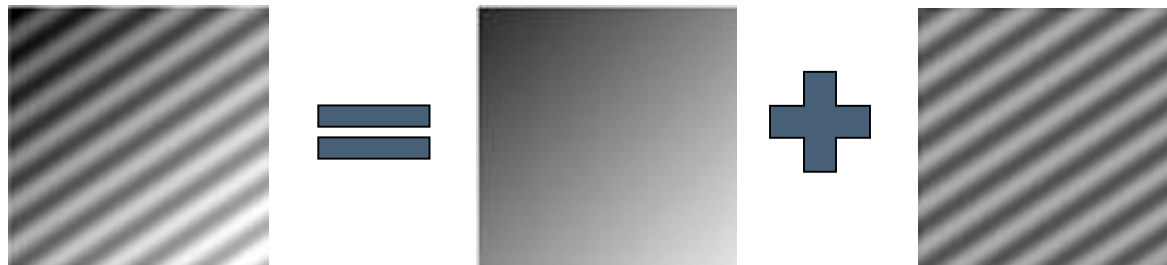
- Basic idea:

Different prediction methods for different mono-component.

- Model:

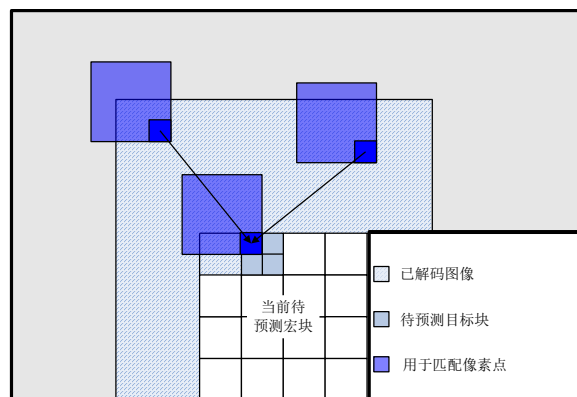
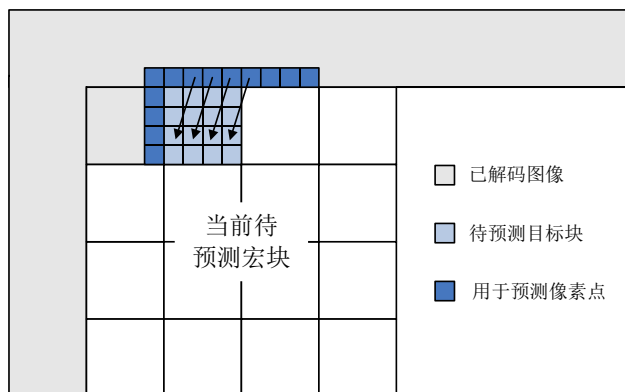
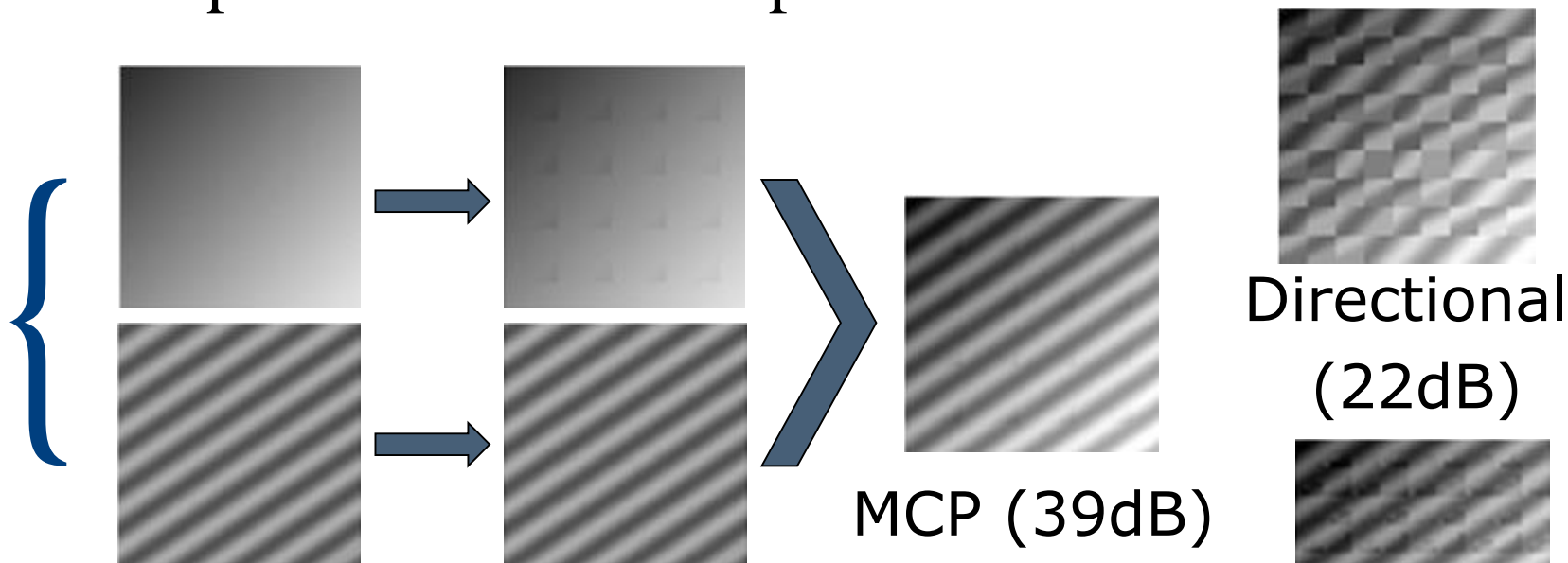
Smooth trend plus several oscillatory textures.

- A simple toy



## ■ 多分量预测编码 (MCP)

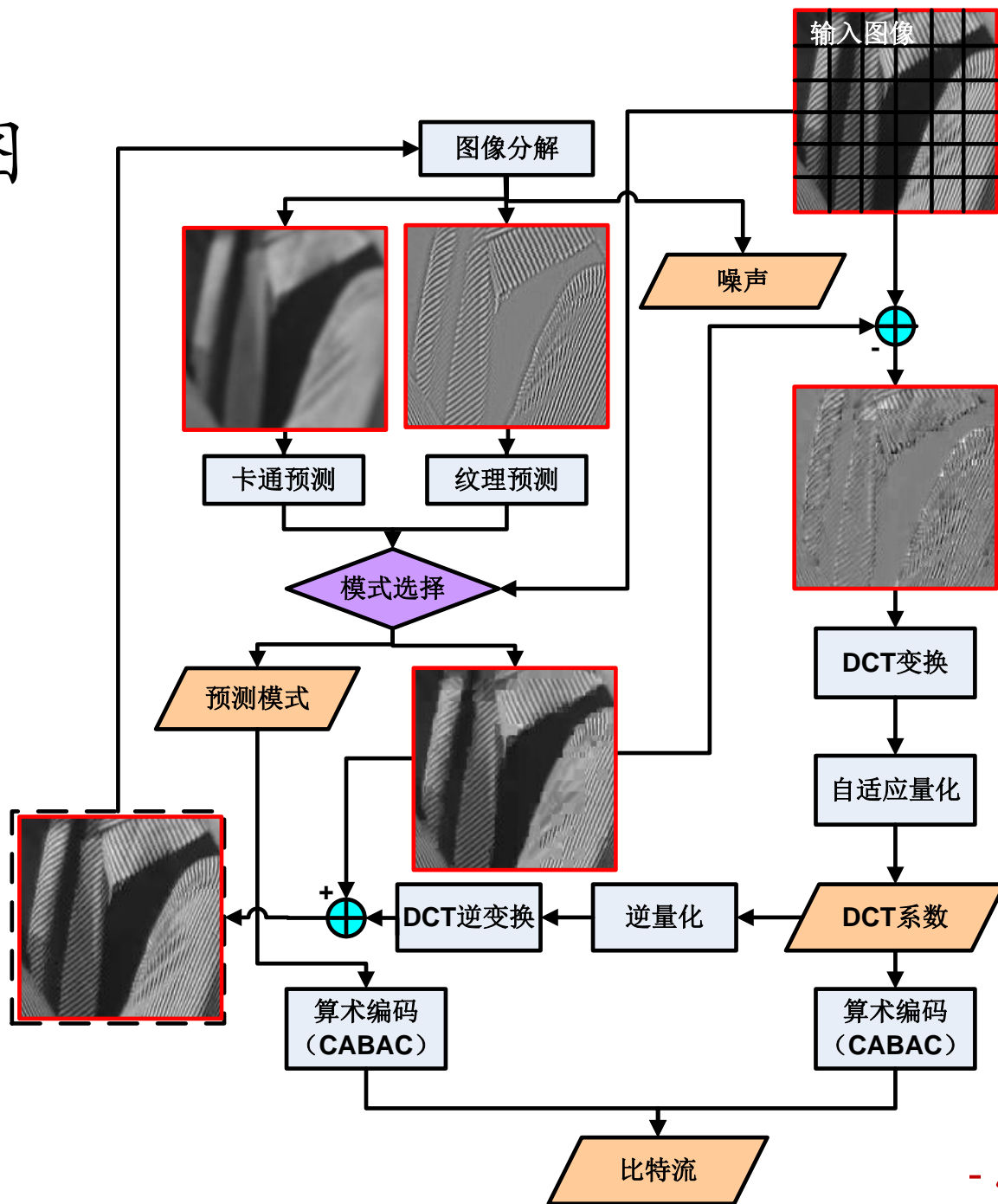
- Comparison of different prediction methods.



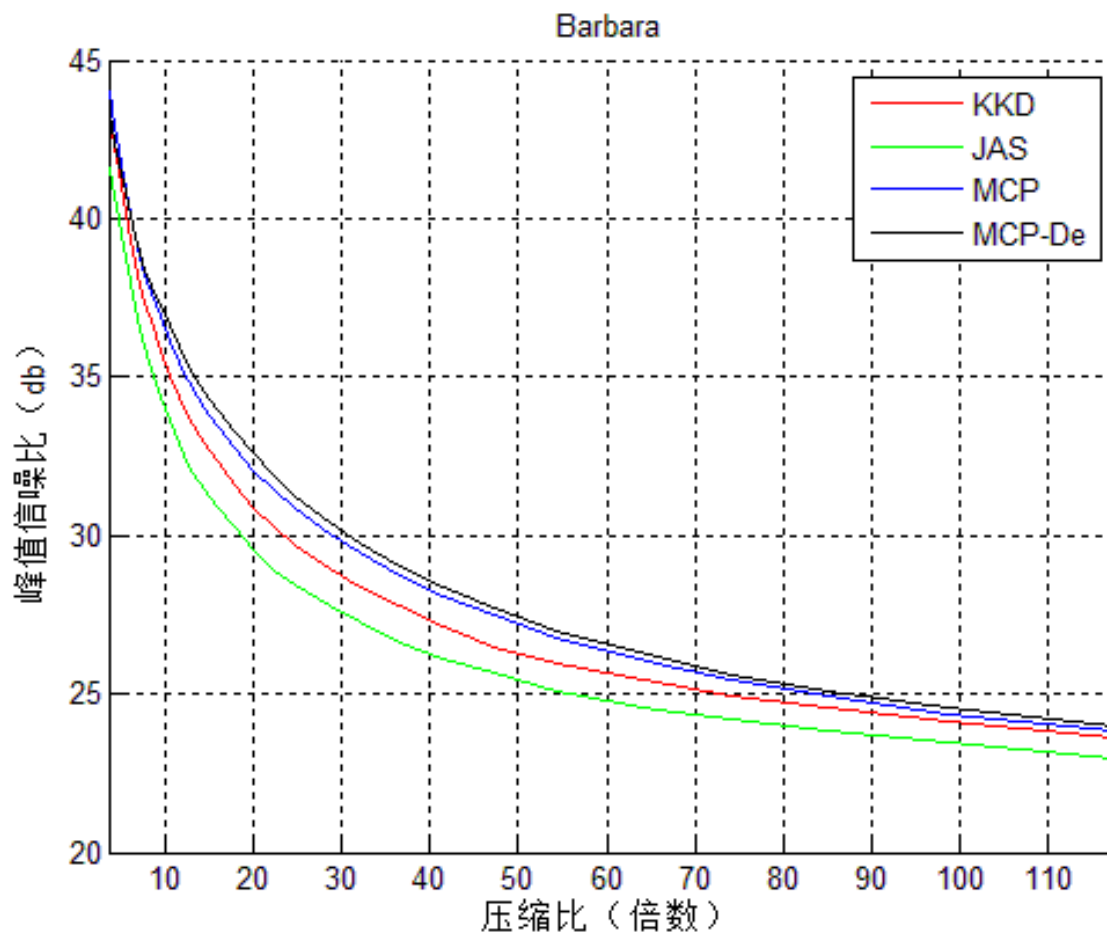
Markov  
(26dB)



## 编码流程图



## 实验结果（不同压缩比）





# 细节部分比较 (手臂)

**KKD**

**MCP**

**MCP-De**



**0.5bpp**



**0.25bpp**





# 内容大纲

- 基本概念
- 信息论基础
- 常见的无损压缩方法
- 图像压缩方法
  - 预测编码
  - 变换编码
- 图像视频压缩标准介绍



# 图像压缩标准

- 相关的国际组织

- ISO (International Standardization Organization, 国际标准化组织)
- ITU (International Telecommunication Union, 国际电信联盟), 前身是CCITT (国际电话电报咨询委员会)

- 相关工作

- 覆盖了从二值到灰度 (彩色) 值的静止和运动图像
- 采用的大部分基本技术前面已经介绍, 主要包括预测和变换编码技术



## 图像压缩标准

- 可分成下面几个系列
  - 用于压缩二值图像
  - 用于压缩静止图像，包括灰度和彩色图像
  - 用于压缩运动图像（视频），包括灰度和彩色图像
- 上述2个组织还在制定一些新的标准，其中一些已经超出纯图像编码的范围





# 图像压缩标准

- JPEG(Joint Picture Expert Group)
  - 由ISO和CCITT联合成立的专家组负责制定静态图像（彩色与灰度图像）的压缩算法
  - 基于DCT的有损编码基本系统，可用于绝大多数压缩应用场合
  - 用于高压缩比、高精确度或渐进重建应用的扩展编码系统
  - 用于无失真应用场合的无损系统
  - JPEG2000



# 视频压缩标准

- 基本概念
- 电视会议标准
- H.263、H.264、H.265, .....
- MPEG标准
- MPEG-1、MPEG-2、MPEG-4、MPEG-7 和  
MPEG-21, .....

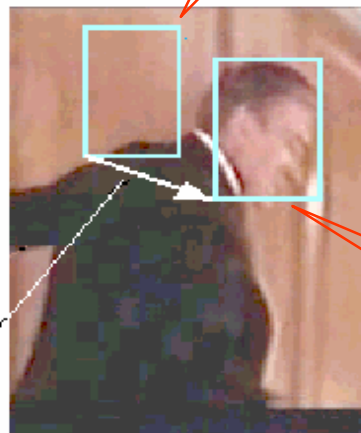
## ■ 视频压缩标准（示例）

由于画面内容有连贯性,因此当前画面的内容可以看作是前面画面内容的位移(运动)。



时刻1

《泰坦尼克》电影上的镜头



时刻2

画面各处相应的位移量(幅度、方向)各有不同,因此画面应细分成块(宏块),以宏块为单位,找出两帧画面中相应宏块之间的位移量---运动矢量



*Any Questions?*