

图像处理和分析技术



第四章 图像编码技术

主讲：李子印
中国计量大学



第四章 图像编码技术

图像编码的核心问题：利用最少时间和最小空间，传输和保存图像数据。



4. 1

数据冗余和压缩

4.1.1 研究背景

1. 信息传输方式发生了很大的改变

- 通信内容的改变
文字+语音→图像+文字+语音
- 通信对象的改变
人与人→人与机器，机器与机器



4.1 数据冗余和压缩

4.1.1 研究背景

2. 图像传输与存储需要的信息量空间：

➤ 彩色视频信息

对于电视画面的分辨率640*480的彩色图像，每秒30帧，则一秒钟的数据量为：

$$640*480*24*30=221.12\text{M}$$

所以播放时，需要221Mbps的通信回路。

参考数据：宽带网为10-200M，电话线为56K。

存储时，1张CD可存640M，则仅可以存放2.89秒的数据。



4.1 数据冗余和压缩

4.1.1 研究背景

2. 图像传输与存储需要的信息量空间：

➤ 传真数据

如果只传送2值图像，以200dpi的分辨率传输，一张A4稿纸的内容的数据量为：

$$1654 * 2337 * 1 = 3888768 \text{ bit}$$

按目前14.4K的电话线传输速率，需要传送的时间是：270秒（4.5分）



4. 1 数据冗余和压缩

4.1.1 研究背景

由于通信方式和通信对象的改变带来的最大问题是：

传输带宽、速度、存储器容量的限制。

给我们带来的一个难题，也给了我们一个机会：

如何用软件的手段来解决硬件上的物理极限。

4.1 数据冗余和压缩

4.1.2 图像编解码过程

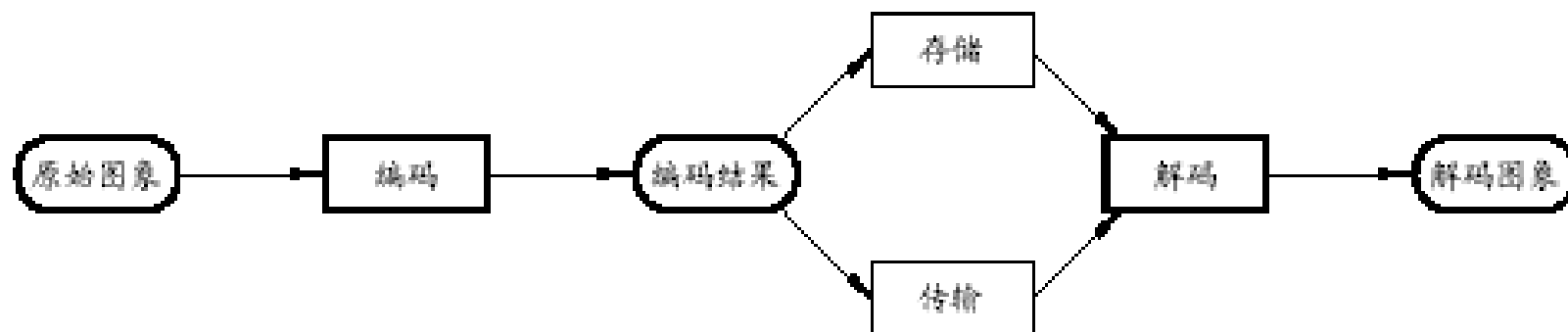


图 5.1.1 图像编解码过程

4.1 数据冗余和压缩

4.1.2 图像编解码过程



图 5.1.2 图像编解码示意



4. 1

数据冗余和压缩

4.1.2 图像编解码过程

图象压缩方法的分类：

➤无失真（无损）压缩：

在压缩和解压缩过程中没有信息损失

压缩率一般在2 ~ 10之间

➤有失真（有损）压缩：

常能取得较高的压缩率（几十~几百）

压缩后并不能经解压缩恢复原状



4. 1

数据冗余和压缩

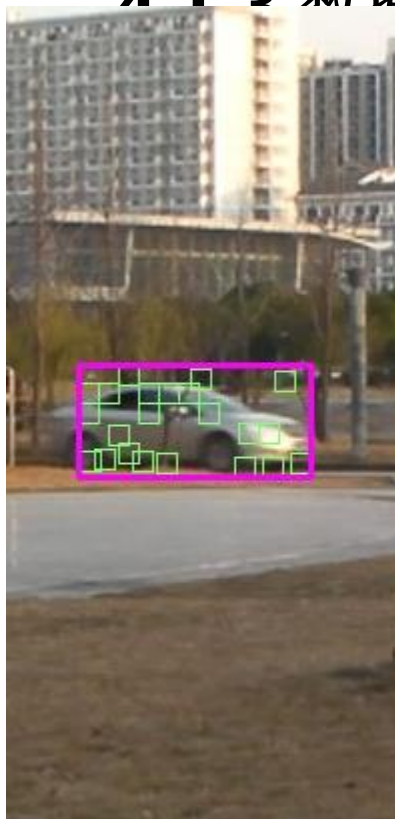
4.1.3 数据冗余

数据和信息不是等同的概念

- 数据是信息的载体；
- 对给定量的信息可用不同的数据量来表示；
- 对给定量的信息，设法减少表达这些信息的数据量称为数据压缩。

4.1 数据冗余和压缩

4.1.2 数据冗余



居
或者

或是已



4.1 数据冗余和压缩

4.1.3 数据冗余

设 n_1 和 n_2 分别代表用来表达相同信息的2个数据集合中的数据量。

➤ 压缩率（压缩比）：——描述压缩算法性能

$$CR = n_1 / n_2$$

其中， n_1 是压缩前的数据量， n_2 是压缩后的数据量。

➤ 相对数据冗余：

$$RD = 1 - 1/CR$$

例： $CR=20$ ； $RD = 19/20$

n_1 相对于 n_2	C_R	R_D	对应的情况
$n_1 = n_2$	1	0	第 1 种表达相对第 2 种表达不含冗余数据
$n_1 \gg n_2$	$\rightarrow \infty$	$\rightarrow 1$	第 1 个数据集合含相当多的冗余数据
$n_1 \ll n_2$	$\rightarrow 0$	$\rightarrow \infty$	第 2 个数据集合包括比原始表达多得多的数据



4. 1

数据冗余和压缩

4.1.3 数据冗余

你的妻子，Helen，将于明天晚上6点零5分在上海的虹桥机场接你。

($23*2+10=56$ 个半角字符)

你的妻子将于明天晚上 6点零5分在虹桥机场接你。

($20*2+3=43$ 个半角字符)

Helen将于明晚6点在虹桥接你。

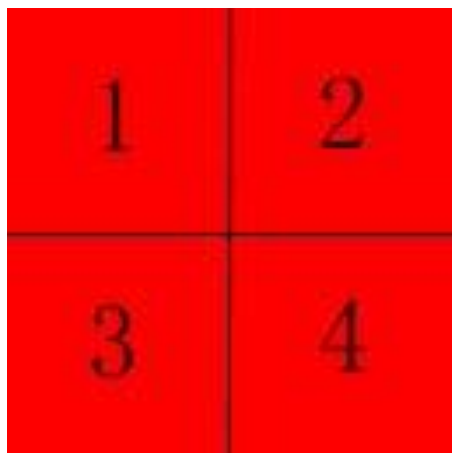
($10*2+7=27$ 个半角字符)

结论：只要接收端不会产生误解，就可以减少承载信息的数据量。

4. 1

数据冗余和压缩

4.1.3 数据冗余



- ◆ 由此我们知道，整理图像的描述方法可以达到压缩的目的。

- 描述语言

1) “这是一幅 2×2 的图像，图像的第一个像素是红的，第二个像素是红的，第三个像素是红的，第四个像素是红的”。

2) “这是一幅 2×2 的图像，整幅图都是红色的”。



4. 1

数据冗余和压缩

4.1.3 数据冗余

数据冗余类别

(1) 编码冗余

与灰度分布的概率特性有关

(2) 象素相关冗余

空间冗余或几何冗余，时间冗余

(3) 心理视觉冗余

与主观感觉有关

减少/消除其中的一种/多种冗余，就能取得数据压缩的效果



4. 1

数据冗余和压缩

4.1.3 数据冗余

1. 编码冗余

编码：需建立码本来表达数据

码本：用来表达一定量的信息或一组事件所需的一系列符号（如字母、数字等）

码字：对每个信息或事件所赋的码符号序列

码字的长度（字长）：

每个码字里的符号个数

使用不同编码方法，得到的码字及其长度都不同。



4.1 数据冗余和压缩

4.1.3 数据冗余

1. 编码（统计）冗余

图象中灰度出现的概率

$$p_s(s_k) = n_k / n \quad k = 0, 1, \dots, L-1$$

不同灰度出现的概率不同

平均比特数

$$L_{\text{avg}} = \sum_{k=0}^{L-1} l(s_k) p_s(s_k)$$

用较少的比特数表示出现概率较大的灰度级

用较多的比特数表示出现概率较小的灰度级



4. 1

数据冗余和压缩

4.1.3 数据冗余

1. 编码（统计）冗余

自然码：对每个信息或事件所赋的码是从 2^m 个 m bit的二元码中选出来一个；

变长码：对不同概率的信息或事件赋予不同长度的码长。



4.1 数据冗余和压缩

s_k	$P_s(s_k)$	自然码	自然码 $l(s_k)$	变长码	变长码 $l(s_k)$
0	0.02	000	3	00111	5
1	0.05	001	3	00110	5
2	0.09	010	3	0010	4
3	0.12	011	3	011	3
4	0.14	100	3	010	3
5	0.20	101	3	11	2
6	0.22	110	3	10	2
7	0.16	111	3	000	3

2.81 1.068

4.1 数据冗余和压缩

4.1.3 数据冗余

2. 像素间冗余（空间冗余或几何冗余，时间冗余）

➤在一幅图像中规则的物体和规则的背景具有很强的相关性；



4.1 数据冗余和压缩

4.1.3 数据冗余

2. 像素间冗余（空间冗余或几何冗余，时间冗余）

➤ 在视频中相邻帧图像具有很强的相关性；





4. 1

数据冗余和压缩

4.1.3 数据冗余

2. 像素间冗余

- 为了减少图像中的像素相关冗余，需要将常用的2-D像素矩阵表达形式转换为某种更有效（但可能不直观）的表达形式
- 这种转换常称为映射；
- 可以是可反转的，也可以是不可反转的。

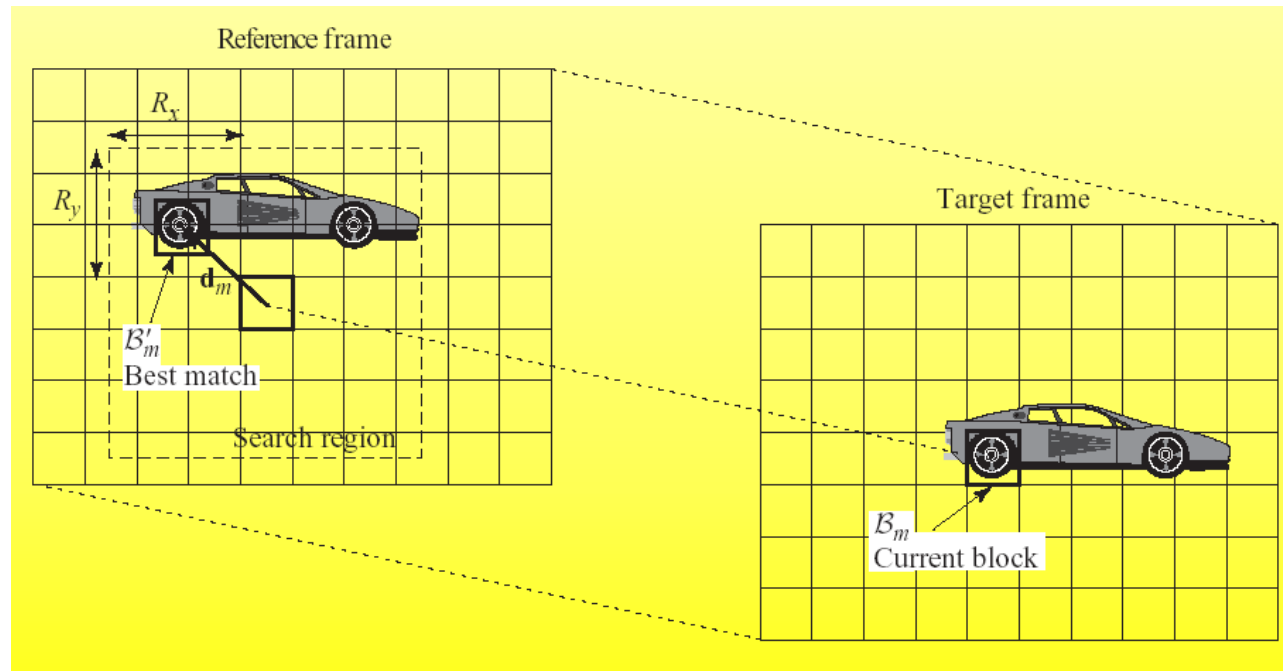
4. 1

数据冗余和压缩

4.1.3 数据冗余

2. 像素间冗余

	C	B		
	A	X		





4. 1

数据冗余和压缩

4.1.3 数据冗余

3. 心理视觉冗余

- 什么是心理视觉冗余？
- 其存在与人观察图象的方式有关

眼睛对某些视觉信息更敏感

人对某些视觉信息更关心

- 与实在的视觉信息相联系
- 量化（损失不可逆转）



4. 1

数据冗余和压缩

4.1.3 数据冗余

$$C(u, v) = \text{Round} \left(\frac{F(u, v)}{Q(u, v)} \right),$$

$C(u, v)$: 量化器输出

$F(u, v)$: 量化器输入

$Q(u, v)$: 量化器步长

$$F'(u, v) = C(u, v) \times Q(u, v)$$

$C(u, v)$: 逆量化器输入

$Q(u, v)$: 量化器步长

4.1

数据冗余和压缩

4.1.4 编解码器结构

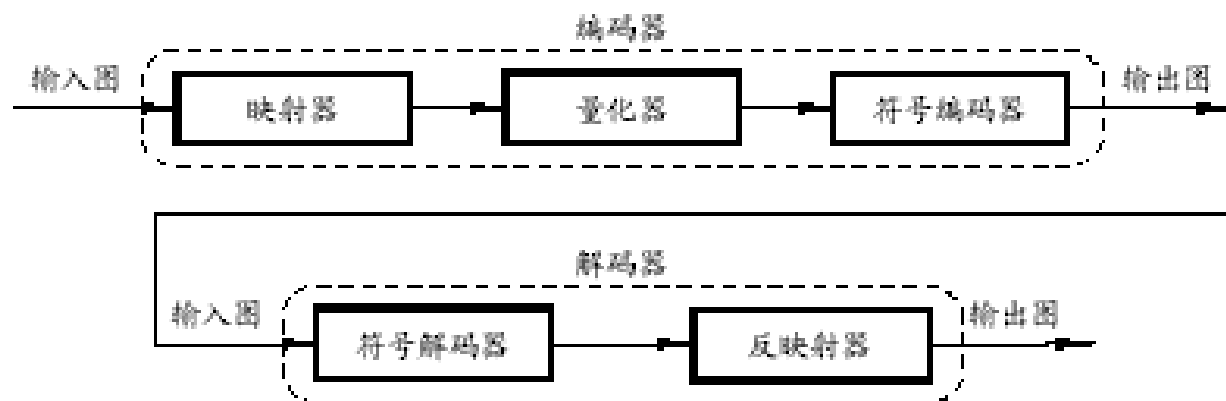


图 5.1.3 编解码器模型

- 映射器通过将输入数据变换以减少像素相关冗余；
- 量化器通过减小映射器输出的精确度来减少心理视觉冗余；
- 符号编码器通过将最短的码赋给最频繁出现的量化器输出值以减少编码冗余。



4.2 图像保真度

4.2.1 基本概念

图像保真度

信息保存型/信息损失型

描述解码图象相对于原始图象的偏离程度

对信息损失的测度

主观保真度准则

主观测量图象的质量，因人而异，应用不方便

客观保真度准则

用编码输入图与解码输出图的某个确定函数表示损失的信息量，便于计算或测量



4.2 图像保真度

4.2.2 客观保真度准则

点误差

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

图误差

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |\hat{f}(x, y) - f(x, y)|$$

均方根误差

$$e_{\text{rms}} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{1/2}$$

均方信噪比

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2}$$



4.2 图像保真度

4.2.2 客观保真度准则

(归一化) 信噪比: 令

单位: 分贝 (dB)

$$\bar{f} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

$$SNR = 10 \lg \left[\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \bar{f}]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2} \right]$$

峰值信噪比

$$PSNR = 10 \lg \left[MN \times f_{\max}^2 / \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]$$



4.2 图像保真度

4.2.3 主观保真度准则

观察者对图象综合评价的平均

电视图象质量评价尺度

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



4.3 无失真编码定理

4.3.1 信息论简介

1、信息量

概率为 $P(E)$ 的随机事件 E 的信息量

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

$I(E)$ 称为 E 的自信息（随概率增加而减少）

特例： $P(E) = 1$ （即事件总发生），那么 $I(E) = 0$

信息的单位：比特（ \log 以2为底）

1个比特：即2个相等可能性的事件之一发生



4.3 无失真编码定理

4.3.1 信息论简介

2、信息系统

一副图像可看做一个具有随机离散输出的信源，这个信源能从一个有限的符号集合中产生一个随机符号序列，这样信源符号集为 $B = \{b_1, b_2, \dots, b_J\}$ ，其中每个元素 b_j 称为信源符号。信源产生符号 b_j 的概率是 $P(b_j)$ ，且有

$$\sum_{j=1}^J P(b_j) = 1$$

4.3 无失真编码定理

4.3.1 信息论简介

2、信息系统

信源通过信道与信宿（即信息用户）连通以传递自信息。



信源符号集: $A = \{a_1, a_2, \dots, a_J\}$

概率矢量: $\mathbf{u} = [P(a_1) \ P(a_2) \ \dots \ P(a_J)]^T$

$$\sum_{j=1}^J P(a_j) = 1$$

用 (A, \mathbf{u}) 可以完全描述信源



4.3 无失真编码定理

4.3.1 信息论简介

平均信息

产生单个信源符号的自信息： $I(a_j) = -\log P(a_j)$

产生 k 个信源符号，符号 a_j 平均来说将产生 $kP(a_j)$ 次

信源平均信息（熵，不确定性）

$$H(u) = -\sum_{j=1}^J P(a_j) \log P(a_j)$$

定义了观察到单个信源符号输出时所获得的平均信息量，信源各符号的出现概率相等时，熵最大。

4.3 无失真编码定理

4.3.1 信息论简介

平均信息

二元信息的熵

信源符号集: $A = \{b_1, b_2\}$

概率矢量: $\mathbf{u} = [p_{bs}, 1 - p_{bs}]^T$

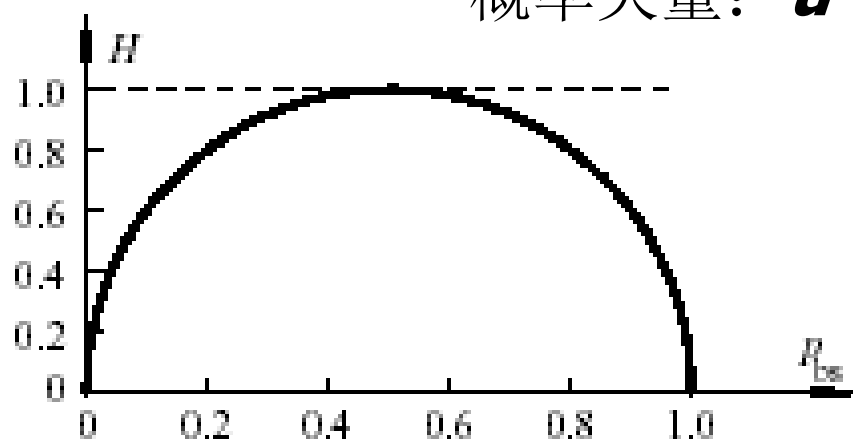


图 5.3.1 二元信源的熵



4.3 无失真编码定理

4.3.2 无失真编码定理(香农第一定理)

单符号输出：

信源符号集： $A = \{b_1, b_2, \dots, b_J\}$

概率矢量： $\mathbf{u} = [P(b_1) \ P(b_2) \ \dots \ P(b_J)]^T$

$$H(\mathbf{u}) = -\sum_{j=1}^J P(a_j) \log P(a_j)$$



4.3 无失真编码定理

4.3.2 无失真编码定理(香农第一定理)

块符号输出:

信源符号集: $B' = \{\beta_1, \beta_2, \dots, \beta_{J^n}\}$

$$P(\beta_j) = P(\mathbf{b}_{j1}) P(\mathbf{b}_{j2}) \dots P(\mathbf{b}_{jn})$$

概率矢量: $\mathbf{u}' = [P(\beta_1) \ P(\beta_2) \ \dots \ P(\beta_{J^n})]^T$

$$H(\mathbf{u}') = -\sum_{i=1}^{J^n} P(\beta_i) \log P(\beta_i) = nH(\mathbf{u})$$

产生块随机变量的（零记忆）信源的熵是对应单符号信源的 n 倍。



4.3 无失真编码定理

4.3.2 无失真编码定理

扩展信源平均码长 $L'_{\text{avg}} = \sum_{i=1}^{J^n} P(\beta_i) l(\beta_i)$

$$-\log P(\beta_i) \leq l(\beta_i) < -\log P(\beta_i) + 1$$

$$-P(\beta_i) \log P(\beta_i) \leq P(\beta_i) l(\beta_i) < -P(\beta_i) \log P(\beta_i) + P(\beta_i)$$

$$H(\mathbf{u}') \leq L'_{\text{avg}} = \sum_{i=1}^{J^n} P(\beta_i) l(\beta_i) < H(\mathbf{u}') + 1$$



4.3 无失真编码定理

4.3.2 无失真编码定理

$$H(\mathbf{u}') \leq L'_{\text{avg}} = \sum_{i=1}^{J^n} P(\beta_i) l(\beta_i) < H(\mathbf{u}') + 1$$

第一定理：

$$\lim_{n \rightarrow \infty} \left[\frac{L'_{\text{avg}}}{n} \right] = H(\mathbf{u})$$

通过对信源的无穷长扩展的编码，可以使 $\frac{L'_{\text{avg}}}{n}$ 任意接近 $H(\mathbf{u})$ 。

编码效率：

$$\eta = n \frac{H(\mathbf{u})}{L'_{\text{avg}}}$$

4.3 无失真编码定理

一阶和二阶编码

	块符号	源符号	$P(\beta_i)$	$I(\beta_i)$	$l(\beta_i)$	码字	码长
一阶 扩展	β_1	b_1	$2/3$	0.58	1	0	1
	β_2	b_2	$1/3$	1.58	2	1	1
二阶 扩展	β_1	b_1b_1	$4/9$	1.17	2	0	1
	β_2	b_1b_2	$2/9$	2.17	3	10	2
	β_3	b_2b_1	$2/9$	2.17	3	110	3
	β_4	b_2b_2	$1/9$	3.17	4	111	3

$$\begin{cases} L_{\text{avg}} = 1 \\ H(\mathbf{u}) = 0.918 \\ \eta = 0.918 \end{cases}$$

$$\begin{cases} L'_{\text{avg}} = 1.89 \\ H'(\mathbf{u}) = 1.836 \\ \eta = 0.97 \end{cases}$$



4.3 无失真编码定理

4.3.3 图像编码技术简介

按原始数据是否有损失:

- (1) 有失真编码 (不可逆编码)
- (2) 无失真编码 (可逆编码)

按编码原理分类:

- (1) 预测编码
- (2) 变换编码
- (3) 统计编码
- (4) 混合编码等



4.3 无失真编码定理

4.3.3 图像编码技术简介

按照产生的年代

- **第一代压缩编码**

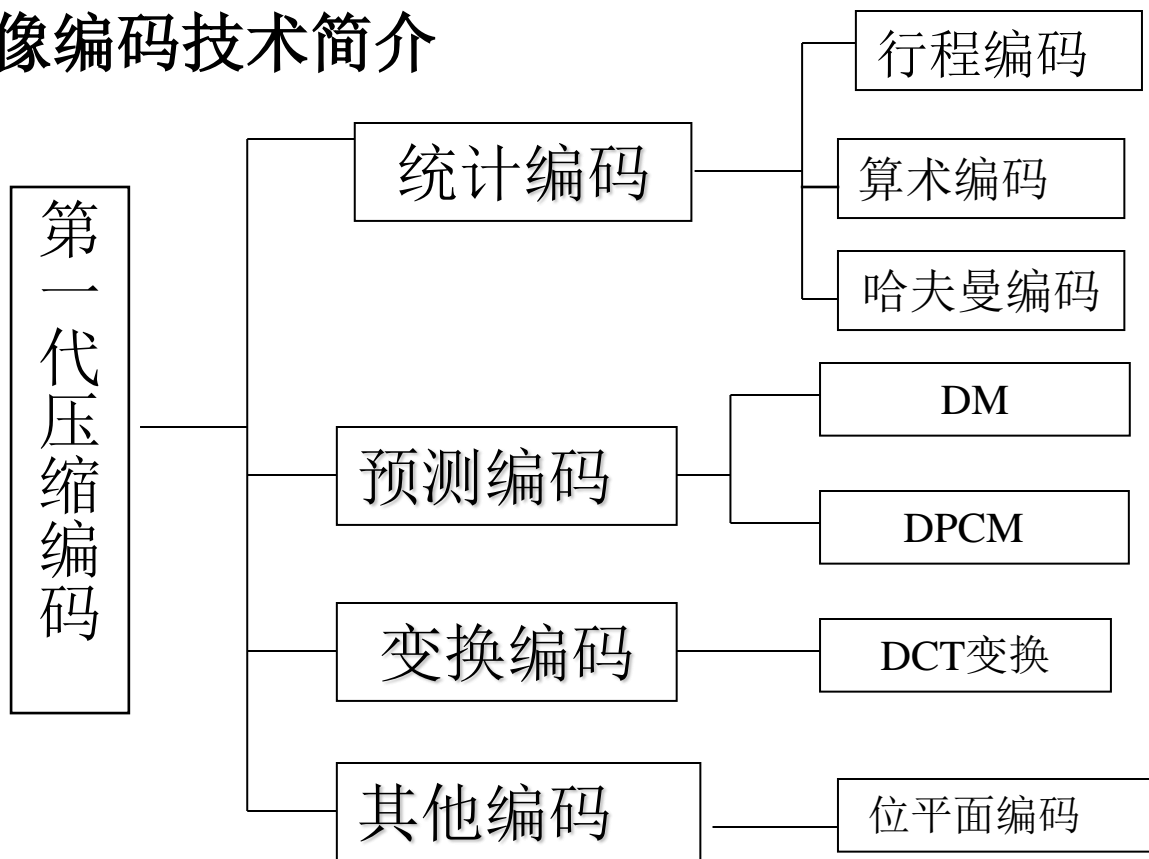
八十年代以前，主要是根据传统的信源编码方法。

- **第二代压缩编码**

八十年代以后，突破信源编码理论，结合分形、模型基、神经网络、小波变换等数学工具，充分利用视觉系统生理心理特性和图像信源的各种特性。

4.3 无失真编码定理

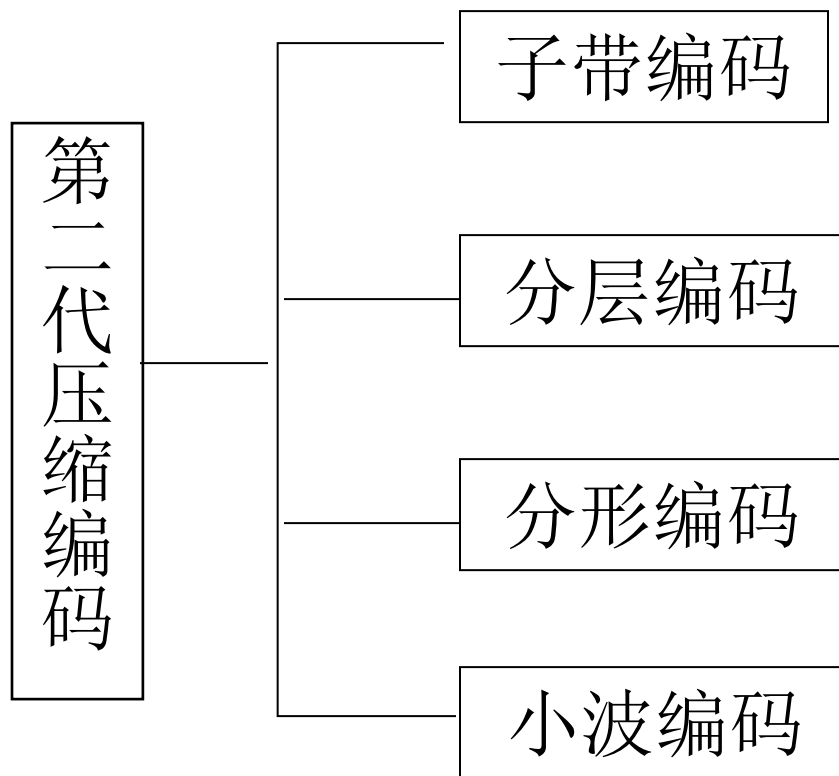
4.3.3 图像编码技术简介





4.3 无失真编码定理

4.3.3 图像编码技术简介





4.3 无失真编码定理

4.3.4 行程编码（RLE编码）

- 基本原理:

通过改变图像的描述方式，来实现压缩。将一行中颜色值相同的相邻像素用一个计数值和该颜色值来代替。

- 举例说明:

aaaa bbb cc d eeeeee ffffff (共 $22 \times 8 = 176$ bits)

→ 4a3b2c1d5e7f (共 $12 \times 8 = 96$ bits)



4.3 无失真编码定理

4.3.4 行程编码（RLE编码）

- 在传真中的应用：

传真件中一般都是白色比较多，而黑色相对比较少。所以可能常常会出现如下的情况：

600w 3b 100w 12b 4w 3b 200w

上面的行程编码所需用的字节数为：

因为： $512 < 600 < 1024$

所以：计数值必须用10bit来表示

$10 * 7 = 70\text{bit}$ $70 + 7 = 77\text{bit}$



4.3 无失真编码定理

4.3.4 行程编码（RLE编码）

- 在传真中的应用：

现在我们就希望对其有一个改善。

白色：10bit，黑色：4bit

600w 3b 100w 12b 4w 3b 200w

所需字节数为：

$$4*10+3*4=52\text{bit} \quad 52+7=59\text{bit}$$

比原来的方式 $11*7=77\text{bit}$ 减少了18bit



4. 3 无失真编码定理

4.3.4 行程编码（**RLE**编码）

行程编码方法特点：

- 1、对传输差错敏感；
- 2、适用于二值图像（如传真图像）的编码。



4.4 哈夫曼编码

4.4.1 基本原理

- 将在图像中出现频度大的像素值给一个比较短的编码，将出现频度小的像素值给一个比较长的编码；
- 哈夫曼编码是一种最优编码。

4.4 哈夫曼编码

4.4.1 基本原理

➤ 举例说明:

aaaa bbb cc d eeeeee fffffff (共 $22 \times 8 = 176$ bits)
4 3 2 1 5 7

编码: f=0 e=10 a=110 b=1111 c=11100 d=11101

11011011011011111111111111110011100111011010
1010100000000

(共 $7 \times 1 + 5 \times 2 + 4 \times 3 + 3 \times 4 + 2 \times 5 + 1 \times 5 = 56$ bits)

4.4 哈夫曼编码

4.4.2 哈夫曼编码

哈夫曼编码步骤

(1) 缩减信源符号数量

将信源符号按出现概率从大到小排列，然后结合

初始信源		信源的消减步骤			
符号	概率	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

The diagram illustrates the step-by-step reduction of the source symbols. It shows the initial probabilities and how they are combined in each step to form new nodes in the tree structure.

- Initial probabilities: a_2 (0.4), a_6 (0.3), a_1 (0.1), a_4 (0.1), a_3 (0.06), a_5 (0.04).
- Step 1: a_3 and a_5 are combined into a new node with probability 0.1.
- Step 2: a_4 and the new node from Step 1 are combined into a new node with probability 0.2.
- Step 3: a_1 and the new node from Step 2 are combined into a new node with probability 0.3.
- Step 4: a_6 and the new node from Step 3 are combined into a new node with probability 0.4.
- Step 5: a_2 and the new node from Step 4 are combined into the final root node with probability 0.6.

4.4 哈夫曼编码

4.4.2 哈夫曼编码

哈夫曼编码步骤

(2) 对每个信源符号赋值

从（消减到）最小的信源开始，逐步回到初始信源

初始信源			对消减信源的赋值							
符号	概率	码字	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								



4.4 哈夫曼编码

4.4.2 哈夫曼编码

哈夫曼编码结果

平均长度

$$L_{avg} = \sum_{k=0}^{L-1} l(s_k) p_s(s_k) = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times (3 + 4 + 5) = 2.2$$

信源熵

$$H(\mathbf{u}) = - \sum_{j=1}^J P(a_j) \log P(a_j) = 2.14$$

编码效率

$$\eta = n \frac{H(\mathbf{u})}{L_{avg}} = 1 \times \frac{2.14}{2.2} = 0.973$$



4.4 哈夫曼编码

4.4.2 哈夫曼编码

哈夫曼编码结果(00 101 1001 1000 01 11)

aaaa bbb cc d eeeeee fffffff (共 $22 \times 8 = 176$ bits)

4 3 2 1 5 7

经过Huffman编码之后的数据为:

0000000010110110110011001100001010101011
11111111111111

(共 $7 \times 2 + 5 \times 2 + 4 \times 2 + 3 \times 3 + 2 \times 4 + 1 \times 4 = 53$ bits)

比前面我们给出的编码得到的56bits的数据量还小



4.4 哈夫曼编码

4.4.2 哈夫曼编码

哈夫曼编码方法：

- (1) 哈夫曼方法构造出的码并不是唯一的；
- (2) 哈夫曼编码对于不同的信息源概率分布，其编码效率不同。



4.4 哈夫曼编码

4.4.2 哈夫曼编码

哈夫曼编码的特点：

- 是一种块（组）码，各个信源符号都被映射成1组固定次序的码符号；
- 是一种即时码。即时性是指对任意一个有限长的码符号串，可以对每个码字分别解码；
- 解码的唯一性

可用简单的查表方式从左到右进行解码

4.4 哈夫曼编码

4.4.2 哈夫曼编码

初始信源			对消减信源的赋值							
符号	概率	码字	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								



4.4 哈夫曼编码

4.4.2 哈夫曼编码

解码的唯一性和即时性的关系：

- 即时码一定是唯一可解码；
- 唯一可解码不一定是即时码。

符号	第1种码	第2种码	第3种码	第4种码
b_1	0	0	0	1
b_2	10	11	01	00
b_3	11	10	00	01

即时唯一的哈夫曼码



4.4 哈夫曼编码

4.4.2 哈夫曼编码

哈夫曼编码

计算量

信源： N 个符号

信源消减次数： $N - 2$

码赋值次数： $N - 2$

- ▶ 当需要对大量符号编码时，构造最优哈夫曼编码的计算量会很大；
- ▶ 此时，可采用一些亚最优的变长编码方法，通过牺牲编码效率来换取编码计算的简便。



4. 4 哈夫曼编码

4.4.3 亚最优哈夫曼编码

- 截断哈夫曼码
- 平移哈夫曼码



4. 4 哈夫曼编码

4.4.3 亚最优哈夫曼编码

截断哈夫曼码

➤ 只对最可能出现的M个符号进行哈夫曼编码，而对其他的码都用在1个合适的定长码前加1个前缀码来表示；

4.4 哈夫曼编码

4.4.3 亚最优哈夫曼编码

块号	信源符号	概率	截断哈夫曼码		平移哈夫曼码		哈夫曼码
第1块	b_1	0.25	01	01	01	01	01
	b_2	0.21	10	10	10	10	10
	b_3	0.19	000	11	000	11	11
	b_4	0.16	001	001	001	001	001
第2块	b_5	0.08	11 00	000 00	11 01	000 01	0001
	b_6	0.06	11 01	000 01	11 10	000 00	00000
	b_7	0.03	11 10	000 10	11 000	000 11	00010
	b_8	0.02	11 11	000 11	11 001	000 001	000011
熵		2.65					
平均长度			2.73		2.78	2.75	2.7



4.4 哈夫曼编码

4.4.3 亚最优哈夫曼编码

平移哈夫曼码步骤

- 重新排列信源符号使它们的概率单减；
- 将符号总数分成相同大小的符号块；
- 对所有块中的各个元素采用同样方法编码；
- 对每个块加上专门的平移符号以区别它们。

4.4 哈夫曼编码

4.4.3 亚最优哈夫曼编码

块号	信源符号	概率	截断哈夫曼码		平移哈夫曼码		哈夫曼码
第1块	b_1	0.25	01	01	01	01	01
	b_2	0.21	10	10	10	10	10
	b_3	0.19	000	11	000	11	11
	b_4	0.16	001	001	001	001	001
第2块	b_5	0.08	11 00	000 00	11 01	000 01	0001
	b_6	0.06	11 01	000 01	11 10	000 00	00000
	b_7	0.03	11 10	000 10	11 000	000 11	00010
	b_8	0.02	11 11	000 11	11 001	000 001	000011
熵		2.65					
平均长度			2.73		2.78	2.75	2.7



4.4 哈夫曼编码

4.4.3 亚最优哈夫曼编码

- ▶ 截断哈夫曼码和平移哈夫曼码的码长都比哈夫曼码要长；
- ▶ 但它们的编码计算都比哈夫曼码要简单。



4.4 哈夫曼编码

4.4.4 哈夫曼编码在图像压缩中的应用

- Huffman编码在图像压缩中的实现

我们知道，对一幅图像进行编码时，如果图像的大小大于**256**时，这幅图像的不同的码字就有可能是非常大，例如极限为**256**个不同的码字。

对整幅图直接进行Huffman编码时，小分布的灰度值，就有可能具有很长的编码。

如：**100**位以上，这样不但达不到压缩的效果反而会使数据量加大，应该如何处理？



4.4 哈夫曼编码

4.4.4 哈夫曼编码在图像压缩中的应用

- 常用的且有效的方法是：

将图像分割成若干的小块，对每块进行独立的Huffman编码。例如：分成 8×8 的子块，就可以大大降低不同灰度值的个数（最多是64而不是256）。



4.5 算术编码

4.5.1 基本原理

- 从整个符号序列出发;
- 采用递推形式连续编码;
- 只需用到加法和移位运算;
- 将被编码的信息表示成实数0和1之间的一个间隔;
- 即时码;
- 唯一可解码 .

4.5 算术编码

4.5.1 基本原理

信源符号集: $A = \{b_1, b_2, b_3, b_4\}$;
概率矢量 $\mathbf{u} = [0.1 \ 0.38 \ 0.22 \ 0.3]^T$
符号串: $b_1 b_2 b_3 b_4$

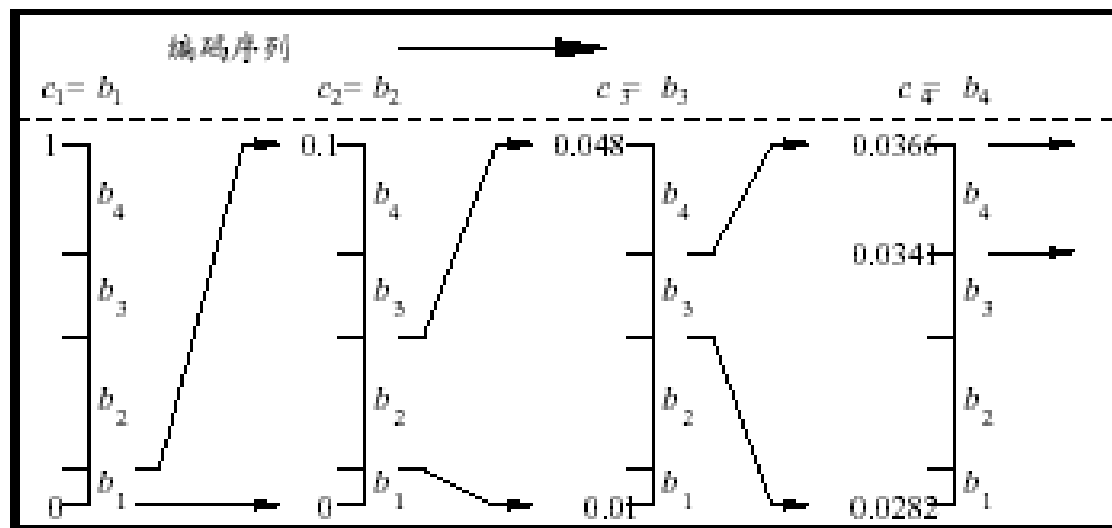


图 5.5.1 算术编码过程图解

[0.0341, 0.0366]

0.00001001

0.03515625



4.5 算术编码

4.5.1 基本原理

字符	概率	范围
a	0.2	[0,0.2)
e	0.3	[0.2,0.5)
i	0.1	[0.5,0.6)
o	0.2	[0.6,0.8)
u	0.1	[0.8,0.9)
!	0.1	[0.9,1.0)

$range = high - low$, 初始 $high = 1, low = 0$

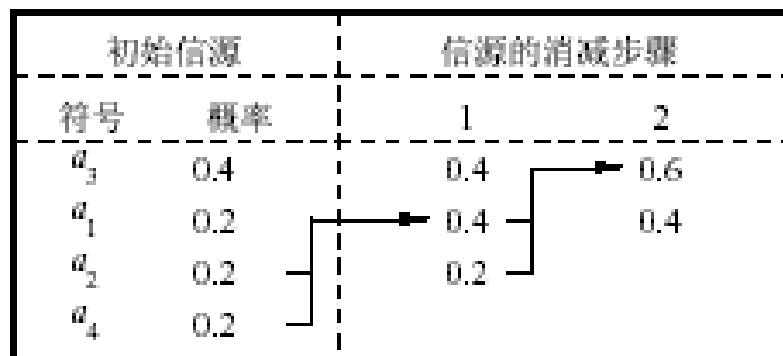
$$\left\{ \begin{array}{l} low' = low + range \times rangelow \\ high' = low + range \times rangehigh \end{array} \right\}$$

$$position = \frac{(code - low)}{range}$$

将最后范围传送给解码器，根据表中各个符号概率及其所分配的范围按上述公式计算，可依次解出各传输字符。

4.5 算术编码

4.5.2 哈夫曼编码和算术编码对比



$a_1 a_2 a_3 a_3 a_4$



1110000101

2^{10}

图 5.5.2 哈夫曼编码中的信源消减

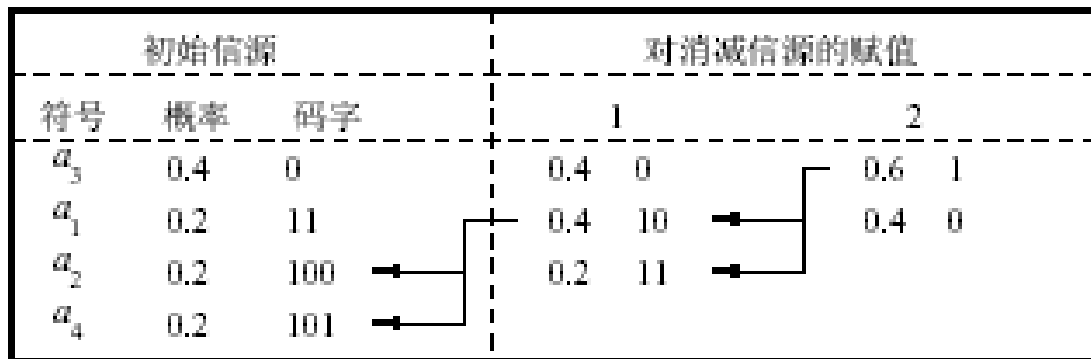


图 5.5.3 哈夫曼编码的信源赋值

4.5 算术编码

4.5.2 哈夫曼编码和算术编码对比

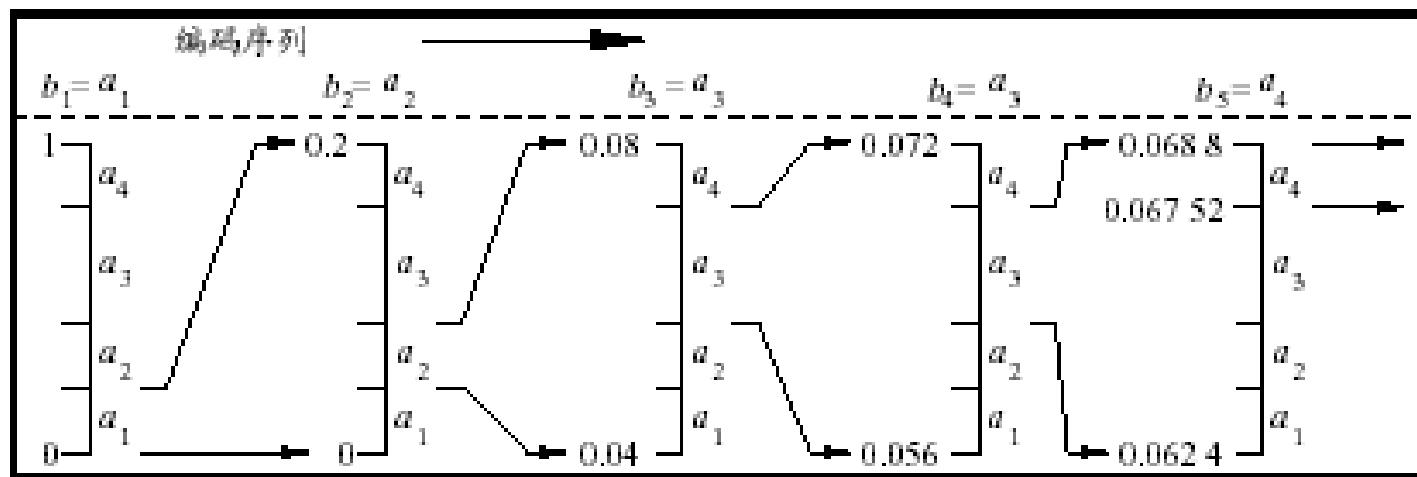


图 5.5.4 算术编码过程

$a_1 a_2 a_3 a_3 a_4$

10^3



$[0.06752, 0.0688]$

0.068



4.5 算术编码

4.5.2 哈夫曼编码和算术编码对比

- 哈夫曼码的数据量比算术编码的数据量要大，或者说哈夫曼编码的效率比算术编码的效率低。

4.5 算术编码

4.5.3 算术解码

信源符号集: $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$;
 概率矢量 $\mathbf{u} = [0.2 \ 0.3 \ 0.1 \ 0.2 \ 0.1 \ 0.1]^T$

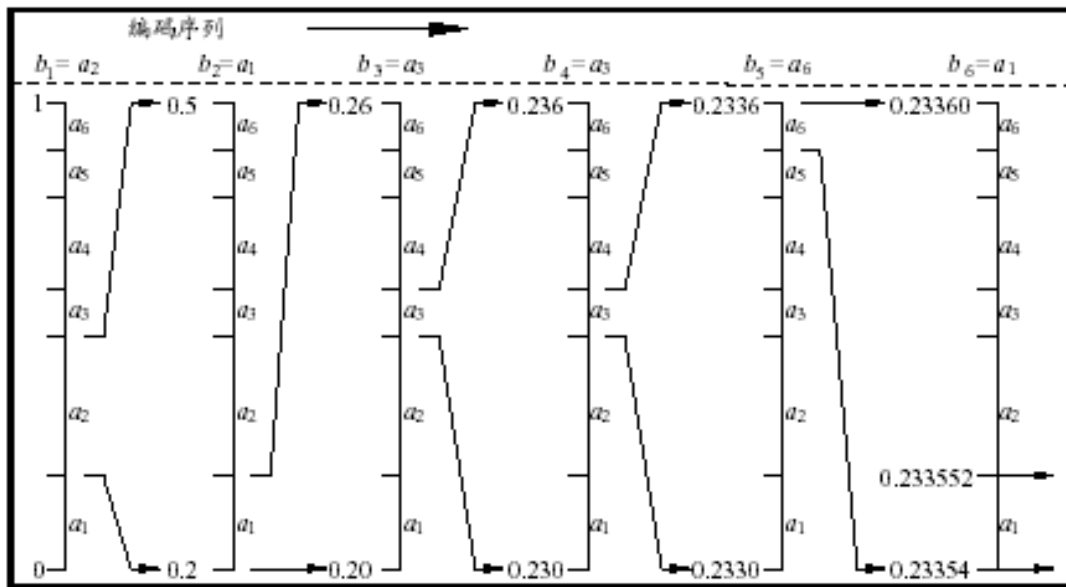


图 5.5.5 编算术码

$range = high - low$, 初始 $high = 1, low = 0$

$$\begin{cases} low' = low + range \times rangelow \\ high' = low + range \times rangehigh \end{cases}$$

$$position = \frac{(code - low)}{range}$$



4.6 例题

4.6.1

设信源符号集 $B = \{b_1, b_2, b_3, b_4, b_5\}$, 其对应的概率矢量为 $u = \{0.5, 0.25, 0.125, 0.0625, 0.0625\}$, 请:

(1)进行哈夫曼编码; (2)写出符号串 $b_2b_1b_1b_5b_4b_3b_3$ 对应的码串; (3)求取该信源符号集的熵; (4)求取哈夫曼编码的平均码长。

4.6 例题

解：(1) 哈夫曼编码：

① 首先，缩减信源符号数量。

+

初始信源		信源的消减步骤		
符号	概率	1	2	3
b_1	0.5	0.5	0.5	0.5
b_2	0.25	0.25	0.25	0.5
b_3	0.125	0.125	0.25	
b_4	0.0625	0.125	0.25	0.5
b_5	0.0625			

4.6 例题

②其次，对信源进行赋值。

初始信源			信源的消减步骤					
符号	概率		1		2		3	
b_1	0.5	0	0.5	0	0.5	0	0.5	0
b_2	0.25	10	0.25	10	0.25	10	0.5	1
b_3	0.125	110	0.125	110	0.25	11		
b_4	0.0625	1110	0.125	111				
b_5	0.0625	1111						

所以，符号 b_1, b_2, b_3, b_4, b_5 对应的哈夫曼码字分别为 0, 10, 110, 1110 和 1111。

4.6 例题

(2) 符号串 $b_2b_1b_1b_5b_4b_3b_3$ 对应的码串为 100011111110110110。↵

(3) 该信源符号集的熵为：↵

$$\begin{aligned} H(u) &= -\sum_{j=1}^J P(b_j) \log P(b_j) \\ &= -0.5 \times \log 0.5 - 0.25 \times \log 0.25 - 0.125 \times \log 0.125 - 2 \times 0.0625 \times \log 0.0625 \text{↵} \\ &= 0.5 + 0.5 + 0.375 + 0.5 = 1.875 \end{aligned}$$

该信源符号集的熵为 1.875。↵

(4) 哈夫曼编码的平均码长为：↵

$$\begin{aligned} L_{avg} &= \sum_{j=1}^J l(b_j) P(b_j) \\ &= 0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 2 \times 0.0625 \times 4 \text{↵} \\ &= 0.5 + 0.5 + 0.375 + 0.5 = 1.875 \end{aligned}$$

哈夫曼编码的平均码长为 1.875。↵



4.7 无损预测编码

4.7.1 预测编码原理

- 空域方法，消除像素间的冗余
- 像素间的相关性使得预测成为可能
- 仅提取每个像素中的新信息并对它们编码

无损预测编码

信息保存型

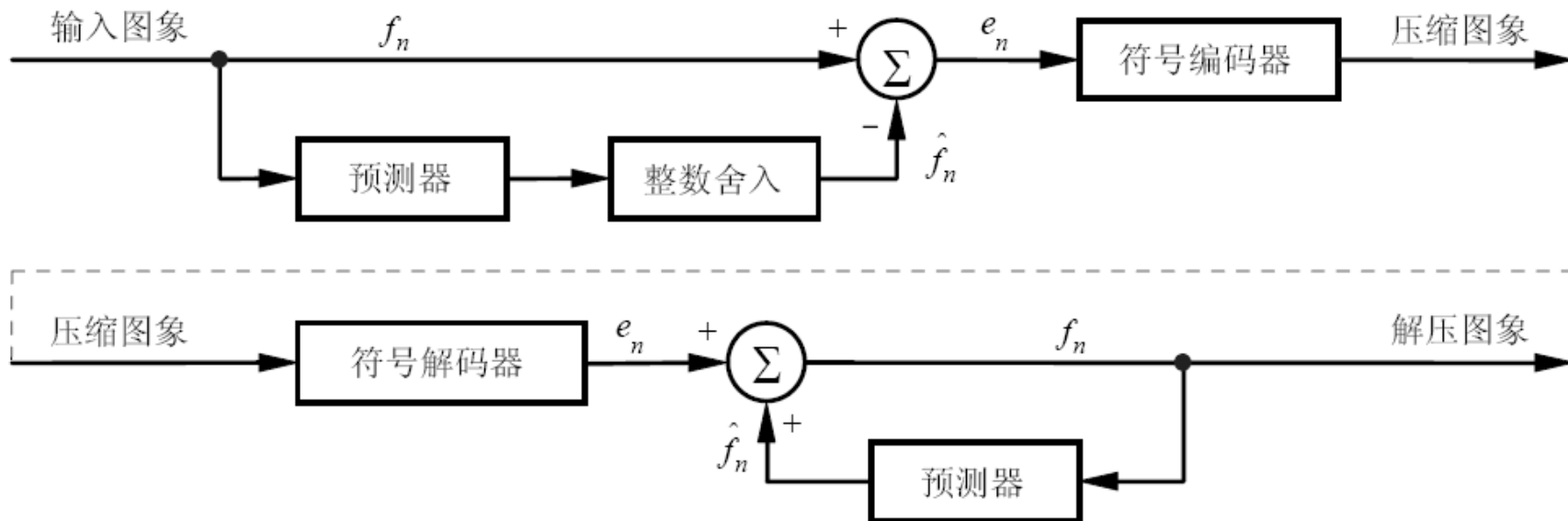
有损预测编码

信息损失型

4.7 无损预测编码

4.7.2 无损预测编码原理

编码器 + 解码器（有相同的预测器）





4.7 无损预测编码

4.7.2 无损预测编码原理

输入序列: $f_n (n = 1, 2, \dots)$

预测输出: \hat{f}_n (舍入成整数)

预测误差: $e_n = f_n - \hat{f}_n$

误差编码: 在符号编码器中用变长码编误差

解压序列: $f_n = e_n + \hat{f}_n$

哪里取得了压缩?

(消除了像素间冗余)



4.7 无损预测编码

4.7.2 无损预测编码原理

m 阶线性预测:

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

1-D线性预测:

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

一阶1-D线性预测:

$$\hat{f}_n(x, y) = \text{round} [a f(x-1, y)]$$



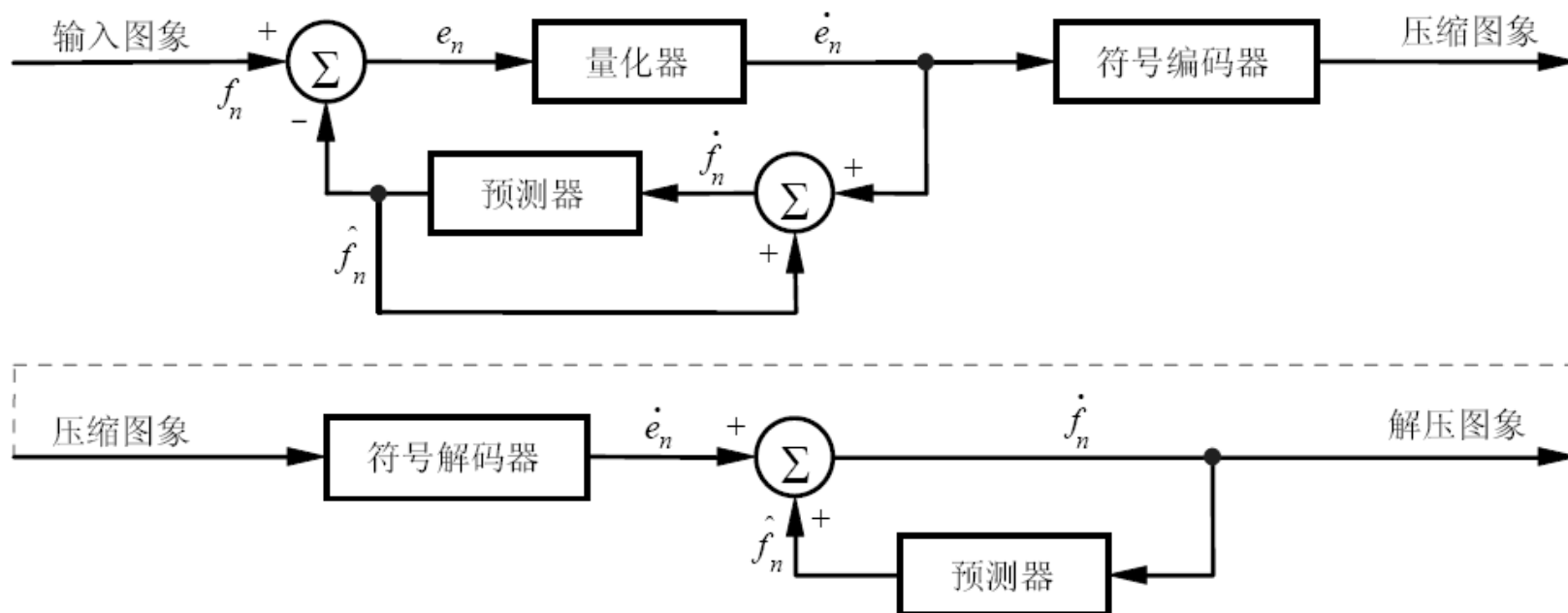
4.7 无损预测编码

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
f_n	10	10	12	15	19	24	30	37	45	54	64	74	83	91	98	104
f_{n-1}	\sim	10	10	12	15	19	24	30	37	45	54	64	74	83	91	98
Λf_n	\sim	10	10	12	15	19	24	30	37	45	54	64	74	83	91	98
e_n	\sim	0	2	3	4	5	6	7	8	9	10	10	9	8	7	6

4.8 有损预测编码

4.8.1 有损预测编码原理

增加了1个量化器，预测器放在1个反馈环中





4.8 有损预测编码

4.8.1 有损预测编码原理

输入序列: $f_n (n = 1, 2, \dots)$

量化输出: $\dot{e}_n = q(e_n)$

预测输入: $\dot{f}_n = \dot{e}_n + \hat{f}_n$

解压序列: $\dot{f}_n = \dot{e}_n + \hat{f}_n$

编码误差: $f_n - \dot{f}_n$

哪里又取得了压缩?

(量化, 减少了
心理视觉冗余)



4.8 有损预测编码

4.8.2 有损预测编码举例

德尔塔调制 (DM)

预测器 $\hat{f}_n = a \dot{f}_{n-1}$

量化器
$$\dot{e}_n = \begin{cases} +c & \text{对 } e_n > 0 \\ -c & \text{其它} \end{cases}$$

预测系数 $a \leq 1$, 常数 $c > 0$

DM方法得到的码率是1比特/象素

4.8 有损预测编码

输入		编码器				解码器		误差
n	f	\hat{f}	e	\dot{e}	\dot{f}	\hat{f}	\dot{f}	$[f - \dot{f}]$
0	12.0	-	-	-	12.0	-	12.0	0.0
1	16.0	12.0	4.0	5.0	17.0	12.0	17.0	-1.0
2	14.0	17.0	-3.0	-5.0	12.0	17.0	12.0	2.0
3	18.0	12.0	6.0	5.0	17.0	12.0	17.0	1.0
4	22.0	17.0	5.0	5.0	22.0	17.0	22.0	0.0
5	32.0	22.0	10.0	5.0	27.0	22.0	27.0	5.0
6	46.0	27.0	19.0	5.0	32.0	27.0	32.0	14.0
7	52.0	32.0	20.0	5.0	37.0	32.0	37.0	15.0
8	50.0	37.0	8.0	5.0	42.0	37.0	42.0	8.0
9	51.0	42.0	9.0	5.0	47.0	42.0	47.0	4.0
10	50.0	47.0	3.0	5.0	52.0	47.0	52.0	-2.0

4.8 有损预测编码

4.8.2 有损预测编码举例

DM编码中的失真示例

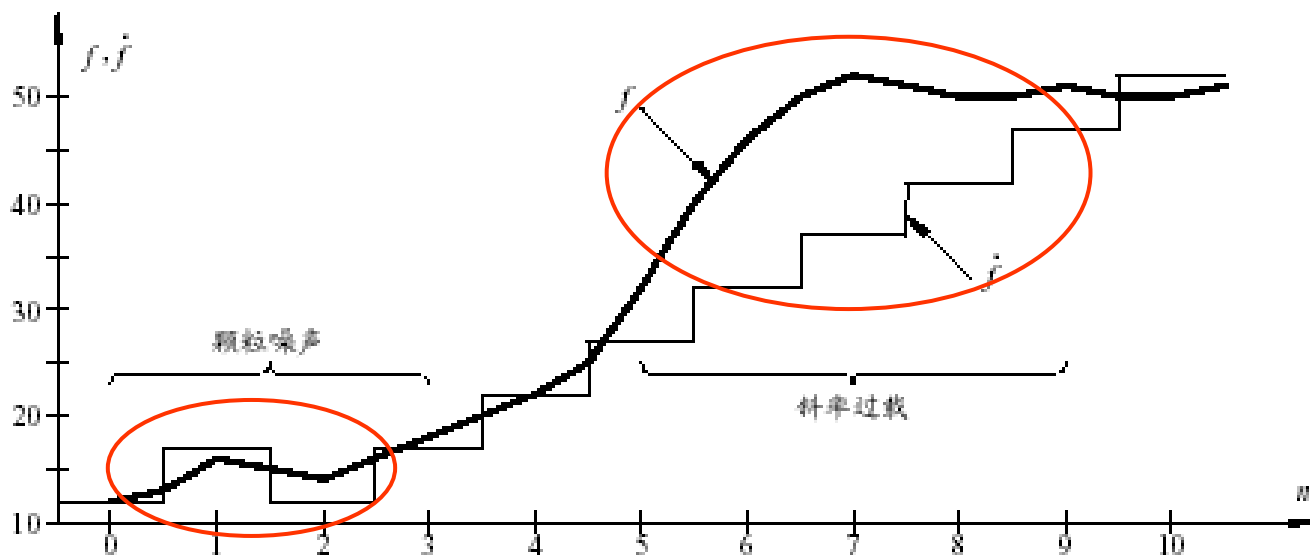


图 5.8.2 DM 编码中的失真示例

4.8 有损预测编码

4.8.3 DPCM

4阶线性预测器

2 1 4

3 C

$$\hat{f}(x, y) = a_1 f(x, y-1) + a_2 f(x-1, y-1) + a_3 f(x-1, y) + a_4 f(x+1, y-1)$$

$$\hat{f}_1(x, y) = 0.97 f(x, y-1)$$

$$\hat{f}_2(x, y) = 0.5 f(x, y-1) + 0.5 f(x-1, y)$$

$$\hat{f}_3(x, y) = 0.75 f(x, y-1) + 0.75 f(x-1, y) - 0.5 f(x-1, y-1)$$

$$\hat{f}_4(x, y) = \begin{cases} 0.97 f(x, y-1) & \text{如 } |f(x-1, y) - f(x-1, y-1)| \leq |f(x, y-1) - f(x-1, y-1)| \\ 0.97 f(x-1, y) & \text{其它} \end{cases}$$

4.8 有损预测编码

4.8.3 DPCM

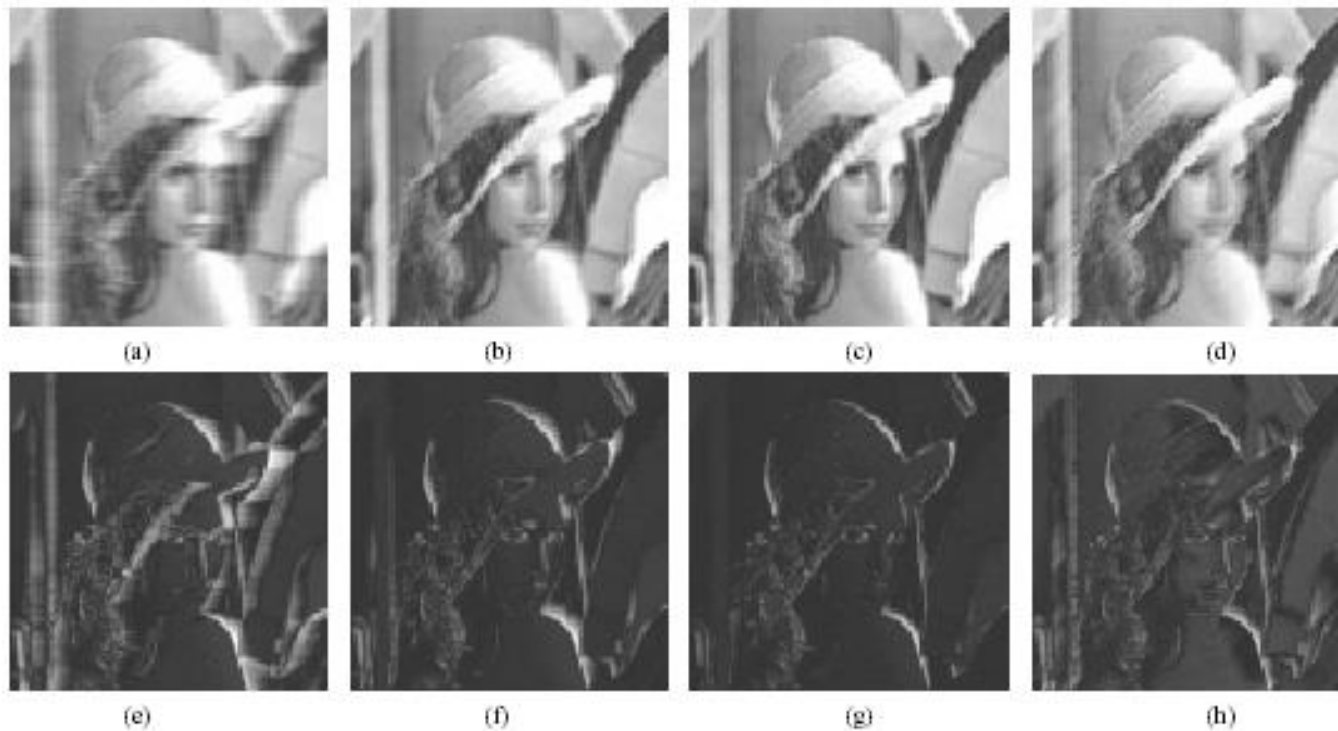


图 5.8.3 预测器效果比较图

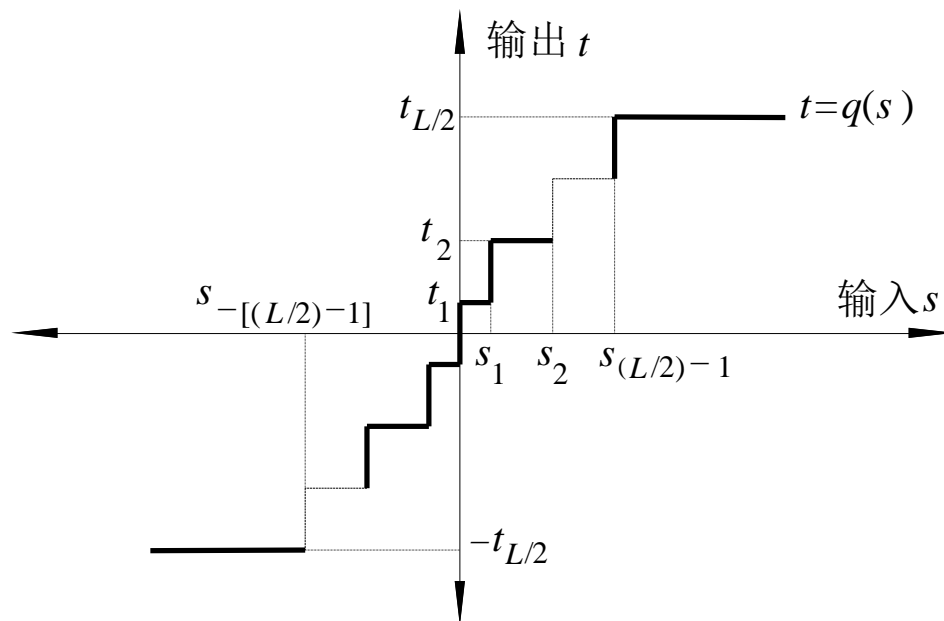
4.8 有损预测编码

4.8.4 量化

重建

$$t = q(s)$$

判别



4.8 有损预测编码

4.8.4 量化



图 5.8.5 DPCM 编码中不同量化器的效果比较



4.9 变换编码

- 预测编码属于空域方法;
- 变换编码属于频域方法;
- 频域图像的特点。



4.9

变换编码

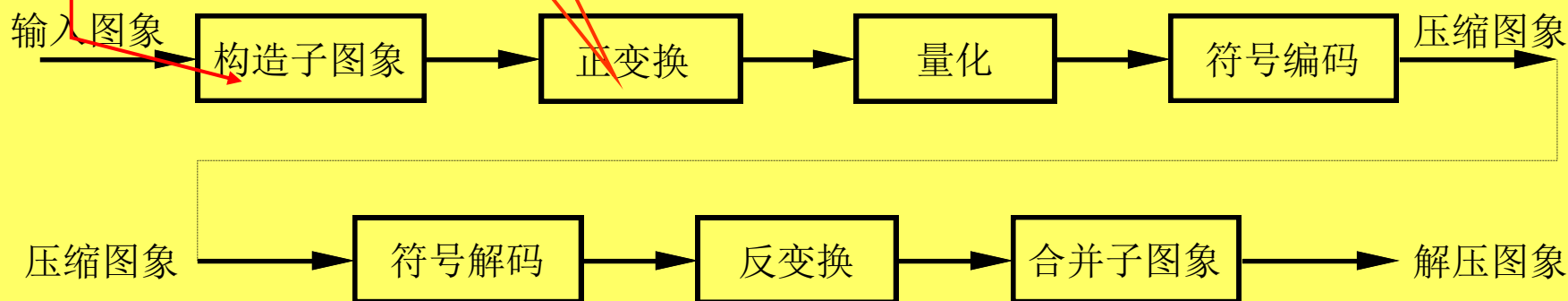
频域方法，非信息保持型

- 变换编码系统
- 子图象尺寸选择
- 变换选择
- 比特分配

4.9 变换编码

4.9.1 变换编码系统

- 图象分解：减少变换的计算复杂度；减少内存使用
- 图象变换：解除每个子图象内部像素之间的相关性，或者说将尽可能多的信息集中到尽可能少的变换系数上





4.9 变换编码

4.9.2 子图像尺寸选择

影响变换编码误差、计算复杂度和内存的使用
(压缩量和计算复杂度都随子图像尺寸的增加而增加)

两个条件:

- ① 相邻子图像之间的相关性(冗余)减少到某个可接受的水平;
- ② 子图像的长和宽都是2的整数次幂

最常用的子图像尺寸: 8×8 和 16×16

4.9 变换编码

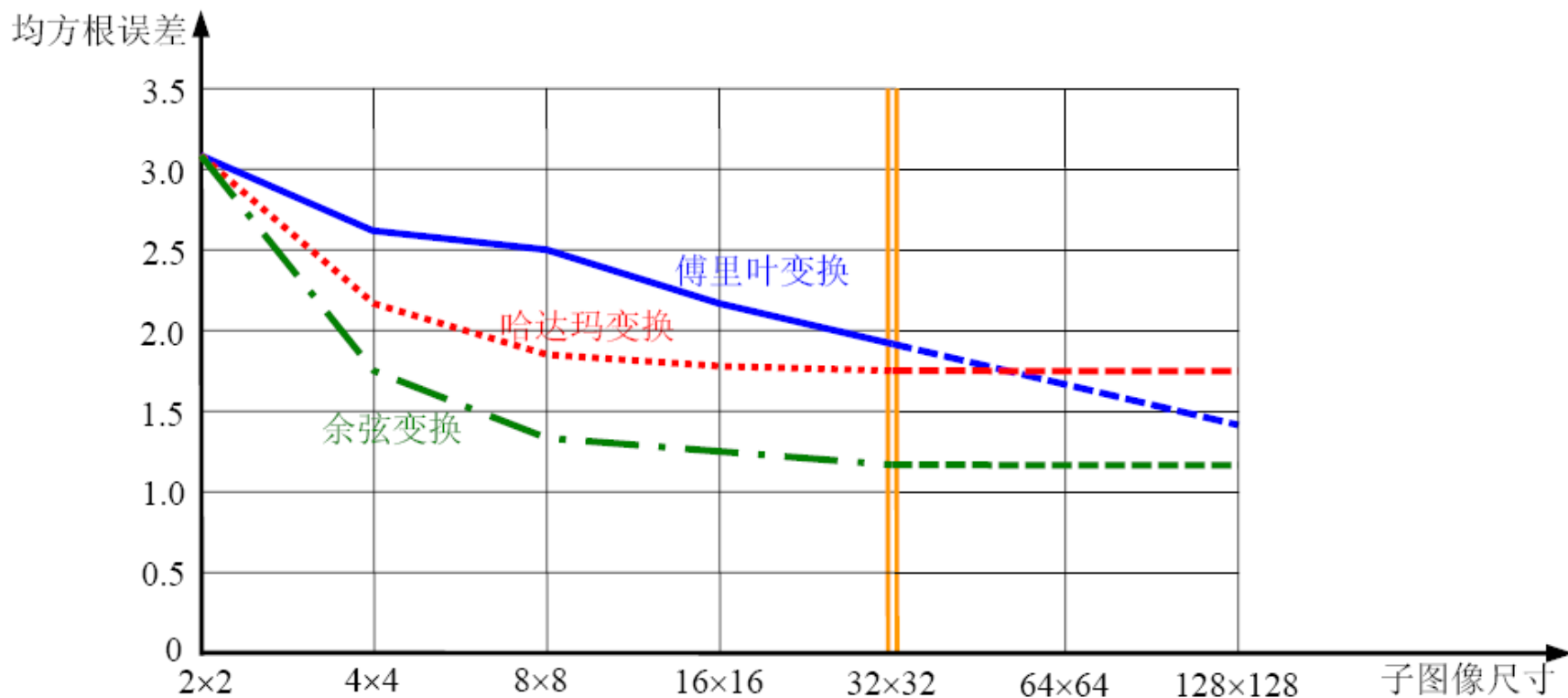
4.9.2 子图像尺寸选择



4.9 变换编码

4.9.2 子图像尺寸选择

变换编码重建误差与子图像尺寸的关系





4.9

变换编码

4.9.3 变换选择

- ▶ 一个能把最多的信息集中到最少的系数上去的变换所产生的重建误差最小
- ▶ 不同变换的信息集中能力不同
 - KLT最优，但计算量非常大
 - 正弦类变换（如DFT和DCT）较优
 - 非正弦类变换（如WHT）实现简单



4.9 变换编码

4.9.3 变换选择

信息集中能力：

$\text{KLT} > \text{DCT} > \text{DFT} > \text{WHT}$

所需计算量：

$\text{KLT} \gg \text{DCT} > \text{DFT} > \text{WHT}$

DCT是较好的（综合）选择



4.9

变换编码

4.9.4 比特分配

比特分配：对变换子图象的系数截断、量化和编码的全过程

截断误差

- ① 截除的变换系数的数量和相对重要性
- ② 用来表示所保留系数的精度（量化）

保留系数的2个准则

- ① 最大方差准则，称为分区编码
- ② 最大幅度准则，称为阈值编码

4.9 变换编码

4.9.4 比特分配

1、分区编码

具有最大方差的变换系数带有最多的图象信息
事先确定模板，保留一定的系数

1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	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

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

4.9 变换编码

4.9.4 比特分配

2、阈值编码

根据子图象特性自适应选择保留系数
将系数排队，与阈值比较确定去舍

1	1	0	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	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



4.9 变换编码

4.9.4 比特分配

2、阈值编码

随子图象不同而保留不同位置的变换系数
常用三种对变换子图象取阈值的方法：

- (1) 对所有子图象用一个全局阈值
压缩的程度随（不同）图象而异
- (2) 对各个子图象分别用不同的阈值
舍去同数量系数，码率是个常数
- (3) 根据子图象中系数的位置选取阈值。

4.9

变换编码

4.9.5 DCT变换编码

例:

原图像为: $F = \begin{bmatrix} 59 & 60 & 58 & 57 \\ 61 & 59 & 59 & 57 \\ 62 & 59 & 60 & 58 \\ 59 & 61 & 60 & 56 \end{bmatrix}$

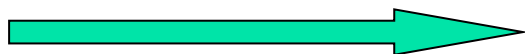
DCT变换



$$D_1 = \begin{bmatrix} 120.5 & 119.5 & 118.5 & 114.0 \\ -0.27 & -0.65 & -1.58 & 0.38 \\ -2.50 & 1.50 & -0.50 & -1.00 \\ 0.65 & -0.27 & 0.11 & 0.92 \end{bmatrix}$$

Huffman:42bits

除以量化矩阵, 取整



$$C = \begin{bmatrix} 16 & 11 & 11 & 16 \\ 12 & 12 & 14 & 19 \\ 14 & 13 & 16 & 24 \\ 14 & 17 & 22 & 29 \end{bmatrix}$$

$$D = \begin{bmatrix} 8 & 11 & 11 & 7 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Huffman:28bits



4.10 混和编码

4.10.1 原理

- 混合编码实现的可能性及有效性：

1. 行程编码：擅长于重复数字的压缩。
2. Huffman编码：擅长于信源符号出现概率有较大差异情况下的编码。

4.10 混和编码

4.10.2 举例

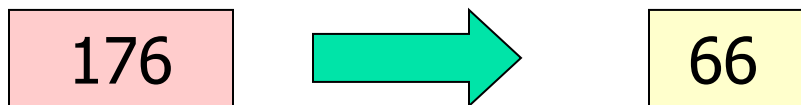
■ 例：

aaaa bbb cc d eeeeee ffffff (共 $22 \times 8 = \mathbf{176 \text{ bits}}$)

4 3 2 1 5 7

行程编码：4a3b2c1d5e7f

(共 $6 \times (8+3) = \mathbf{66 \text{ Bits}}$)



4.10 混和编码

4.10.2 举例

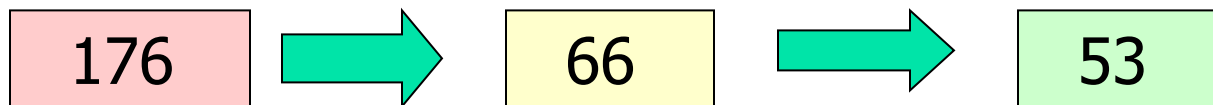
aaaa bbb cc d eeeeee ffffff (共 $22*8=176$ bits)
4 3 2 1 5 7

Huffman编码:

f=01 e=11 a=10 b=001 c=0001 d=0000

101010101000100100100010001000011111111110101
0101010101

(共 $7*2+5*2+4*2+3*3+2*4+1*4=53$ bits)



4.10 混和编码

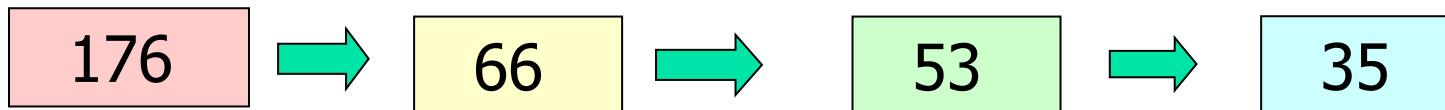
4.10.2 举例

aaaa bbb cc d eeeeee fffffff (共 $22 \times 8 = 176$ bits)
4 3 2 1 5 7

Huffman与行程编码混合:

41030012000110000511701

(共: $3+2+3+3+3+4+3+4+3+2+3+2=35$ bits)



4.11

图像压缩标准

4.11.1 标准和标准化

国际标准组织 (ISO/IEC)

↓
MPEG

国际电报电话咨询委员会 (CCITT)

↑
ITU



联合专家组JPEG



- (1) 大范围调节图像压缩率及其相应的图像保真度，同时编码器是参数可调的，满足用户选择期望的压缩/质量比。
- (2) 能应用于任何连续色调数字源图像，不限制图像的景像内容。
- (3) 计算复杂性易控制，使软件实现可在CPU完成，高性能要求的应用，可用硬件实现。



4.11

图像压缩标准

4.11.2 二值图像压缩标准

1. G3和G4

- 由CCITT的两个小组（Group 3和Group 4）负责制定
- 最初是为传真应用而设计
- G3采用了非自适应、1-D游程编码技术
- G4是G3的一种简化版本
- G3的压缩率约为15:1，G4的压缩率要高1倍



4.11

图像压缩标准

4.11.2 二值图像压缩标准

2. JBIG

- 二值图联合组（joint bilevel imaging group, JBIG）于1991年制定
- 采用了自适应技术，提高了压缩比
- 对打印字符的扫描图象，可提高1.1 ~ 1.5倍
- 对计算机生成的打印字符图象：可提高约5倍



4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG

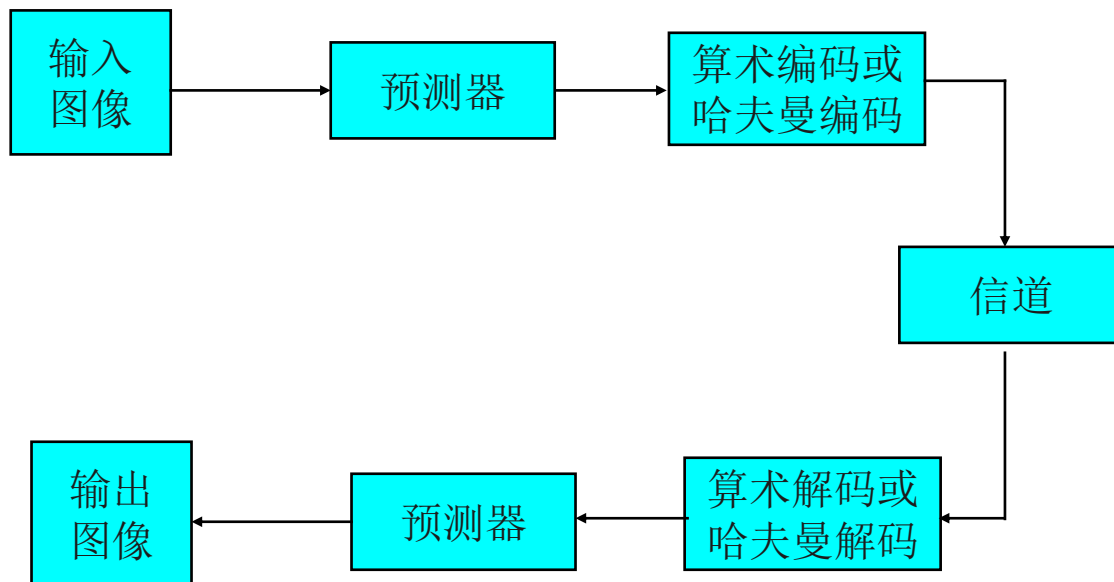
- 联合图象专家组 (joint picture expert group, JPEG) 于1991年制定
- 定义了3种编码系统:
 - (1) 基于DCT的有损编码基本系统, 可用于绝大多数压缩应用场合
 - (2) 用于高压缩比、高精确度或渐进重建应用的扩展编码系统
 - (3) 用于无失真应用场合的无损系统
- 对录象机质量的静止图象的压缩率一般可达到25:1

4.11

图像压缩标准

4.11.3 静止图像压缩国际标准

JPEG的无损预测编码算法



4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的无损预测编码算法

	C	B		
	A	X		

选择值	预测值
0	非预测值
1	A
2	B
3	C
4	$A+B-C$
5	$A+[(B-C)/2]$
6	$B+[(A-C)/2]$
7	$(A+B)/2$

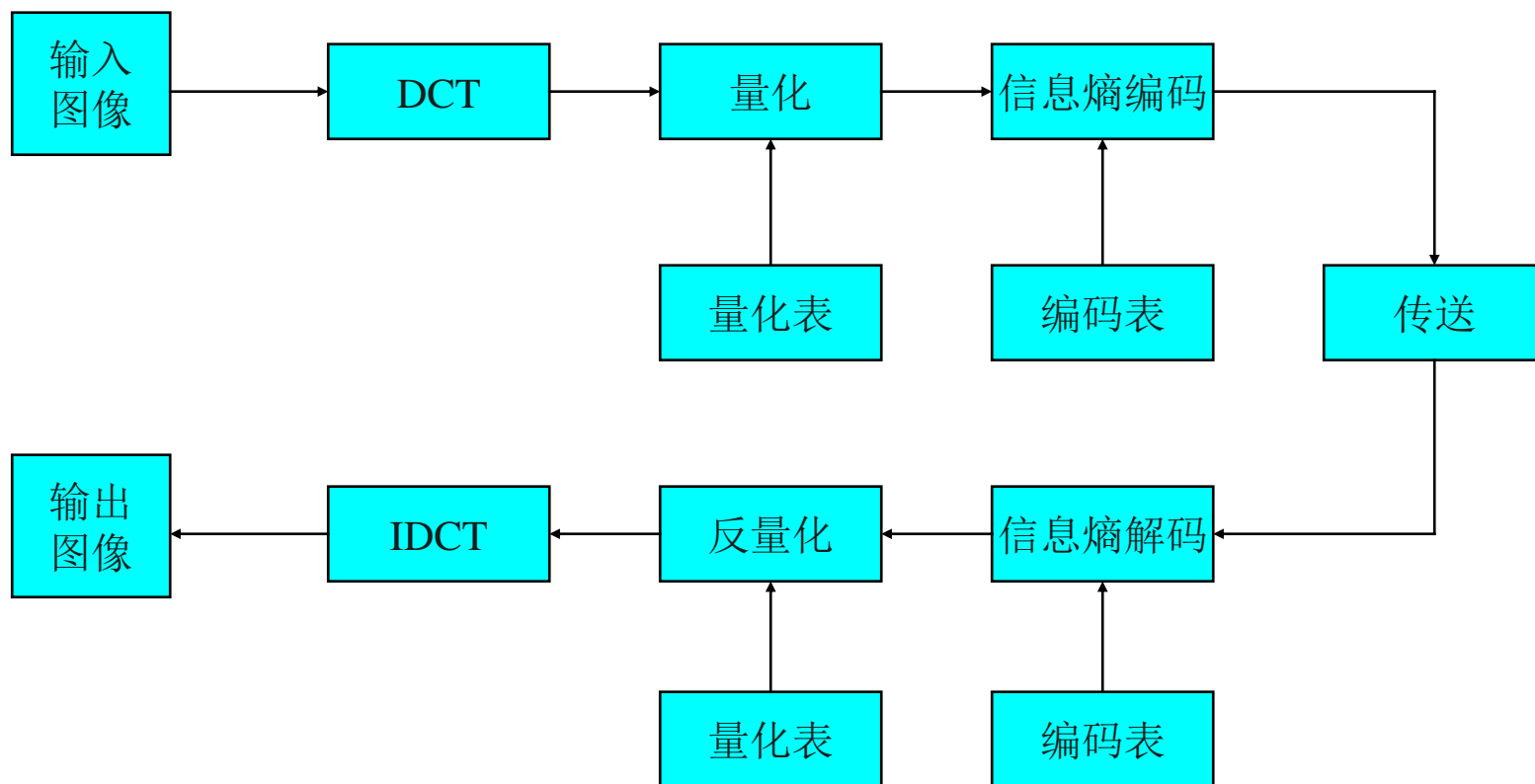
结论：执行时间短，对图像质量要求高的系统有较好的应用价值。

4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法





4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

1、数据块的分块考虑：

图像尺寸 $M \times M$ ，考虑选择小于 M 的整数 N ，令 $M/N=k$ ，保证 M ， N ， k 都是2的幂。则可将 $M \times M$ 的块分割成 k^2 个 $N \times N$ 块。

存储器的空间需求节省 k^2 倍。

减少DCT变换的计算量。



4.11 图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

2、 DCT变换

$$f(i, j) = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) \cdot C(v) F(u, v) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N}$$

$$F(u, v) = \frac{C(u) \cdot C(v)}{4} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N}$$

$$C(\omega) = \begin{cases} \frac{1}{\sqrt{2}}, & \omega = 0 \\ 1, & \omega \neq 0 \end{cases}$$



4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

3、 量化

DCT后，图像时域信号 \longrightarrow 频域信号

频域数据量化的优点：

- 1、增加频域数据里的0个数
- 2、降低了非零数据的表示范围

目的： 提高压缩比



4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

$$C(u, v) = \text{Integer} \left(\text{Round} \left(\frac{F(u, v)}{Q(u, v)} \right) \right),$$

$C(u, v)$: 量化器输出

$F(u, v)$: 量化器输入

$Q(u, v)$: 量化器步长



4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

$$F'(u, v) = C(u, v) \times Q(u, v)$$

$C(u, v)$: 逆量化器输入

$Q(u, v)$: 量化器步长



4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

量化表

彩色像素：灰度 (Y)
色度 (U 和 V)

亮度量化表

色度量化表



4. 11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

亮度量化表

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



4. 11

图像压缩标准

4.11.3 静止图象压缩国际标准

JPEG的基于DCT的变换编码算法

色度量化表

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

4.11

图像压缩标准

4、系数的编码

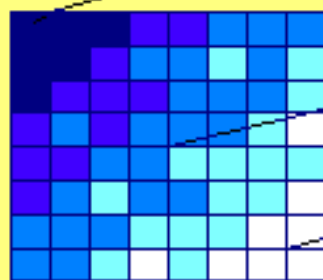
分组原则：

8*8块的DCT编码和量化

64个系数： $u=v=0$ 坐标，DC系数（直流系数）

63个系数， AC系数（交流系数）

DC 分量 $F(0,0)$ 图像平均灰度值， AC 分量 63 个系数



高频部分，图像细节

255

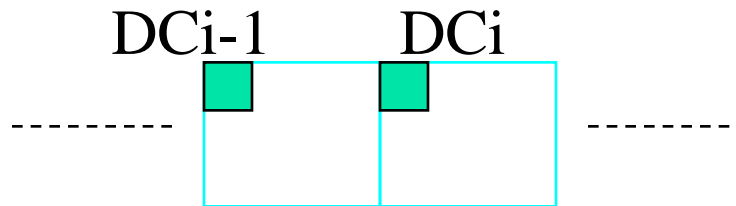
0

4.11

图像压缩标准

4.11.3 静止图象压缩国际标准 JPEG的基于DCT的变换编码算法

差分编码: $\Delta DC_i = DC_i - DC_{i-1}$

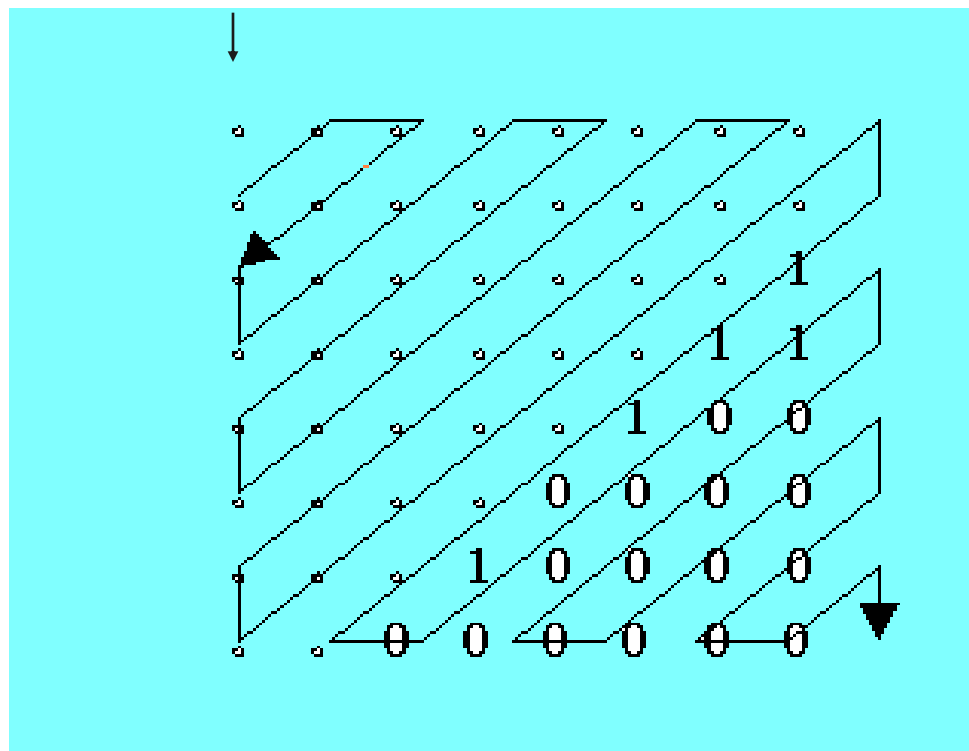


4.11

图像压缩标准

4.11.3 静止图象压缩国际标准 JPEG的基于DCT的变换编码算法

Z形扫描





4.11

图像压缩标准

4.11.3 静止图象压缩国际标准

2. JPEG2000

- 对JPEG标准进行更新换代，2000年问世
- 提高低码率时图象的压缩质量
- 增加了许多功能：

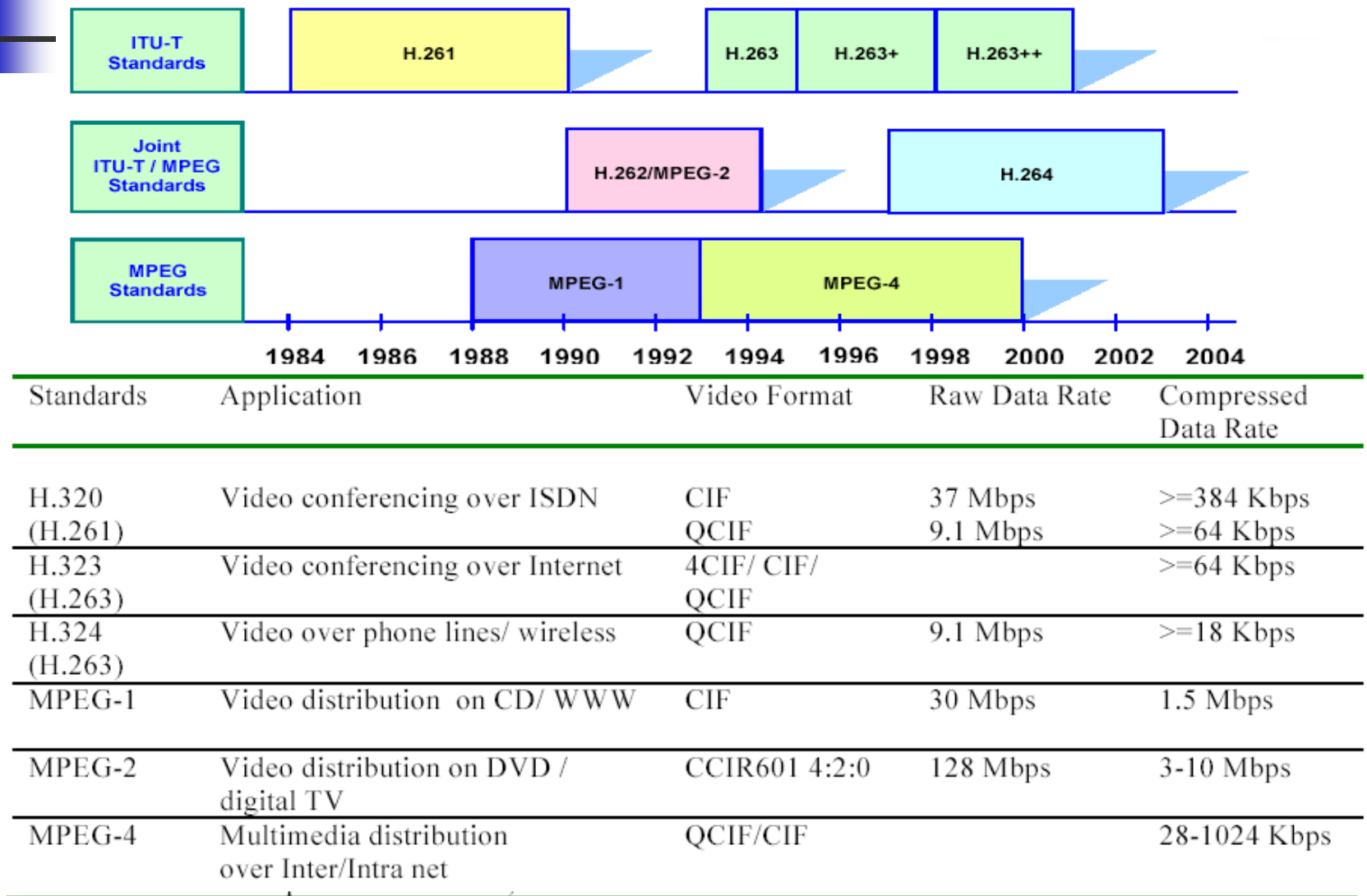
根据图象质量、视觉感受和分辨率进行渐进传输，可对码流随机存取和处理

- 编码变换采用了小波变换

消除了DCT的块效应

4.11 图像压缩标准

4.11.4 运动图像压缩国际标准压缩





4.11 图像压缩标准

4.11.4 运动图象压缩国际标准压缩

MPEG-1视频压缩算法（ISO/IEC 11172）

➔ Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s

主要技术：

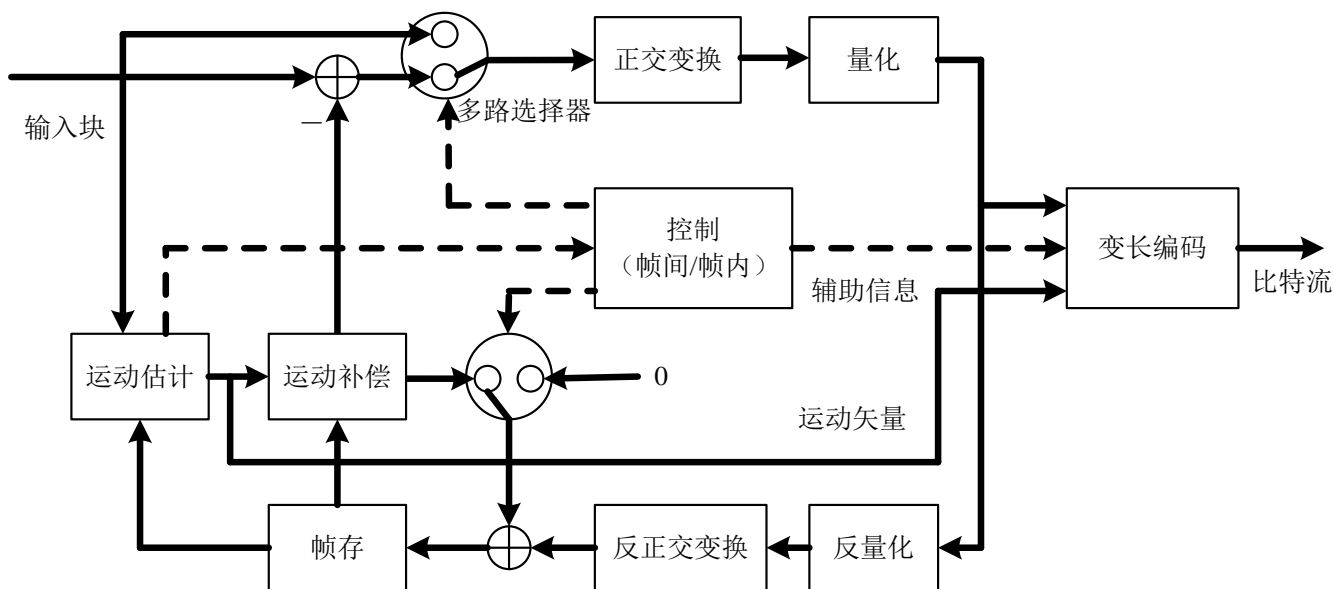
- （1）、减少时间冗余的基于块的帧间运动补偿技术
- （2）、基于DCT变换的减少空间冗余的技术。

4.11 图像压缩标准

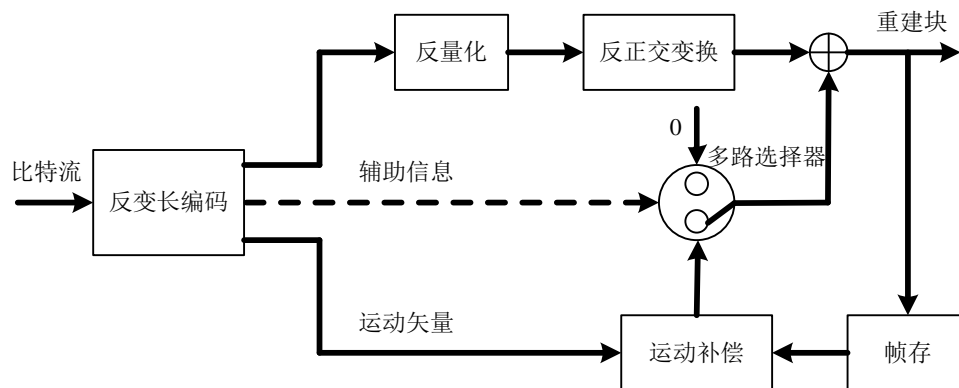
4.11.4 运动图象压缩国际标准压缩

MPEG-1视频 压缩算法

编码器

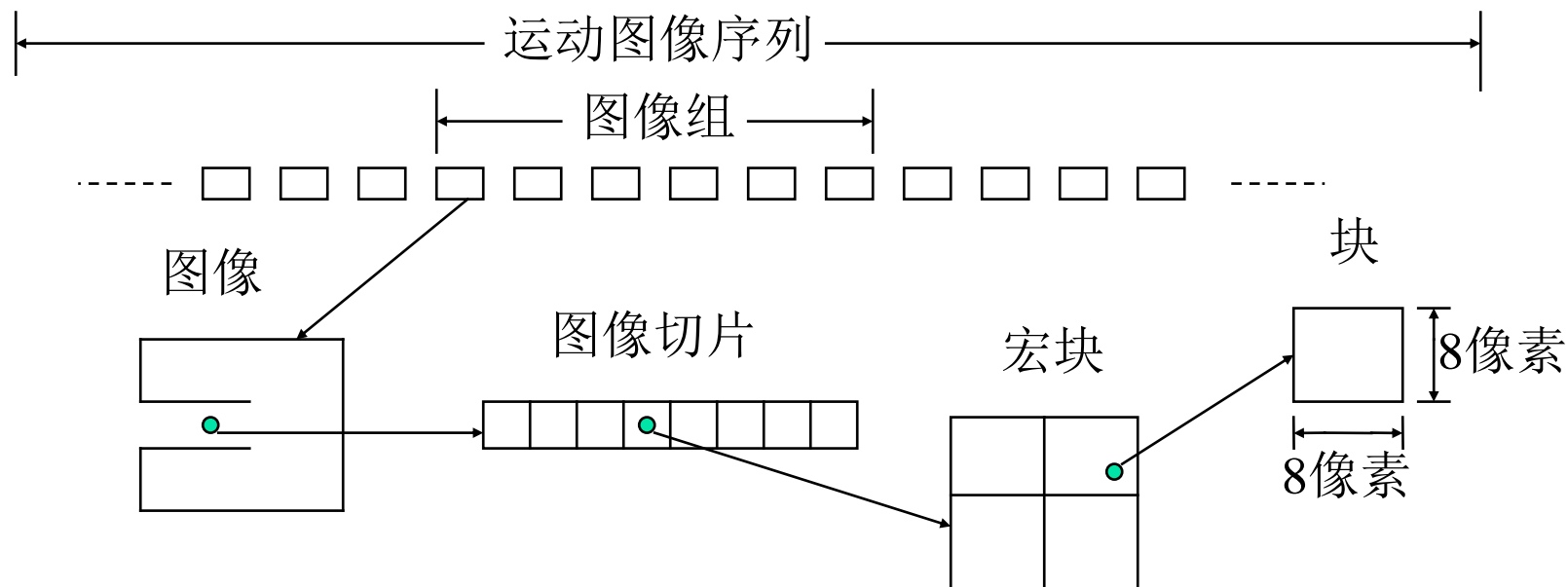


解码器



4.11 图像压缩标准

4.11.4 运动图像压缩国际标准压缩



运动序列：一个运动图像的数据流（包括表头、结束标志码）

图像组：一系列图像组成，可从运动序列中随机抽取

图像：最原始的编码单位（包括一个亮度信号Y、两个色度信号U、V。）

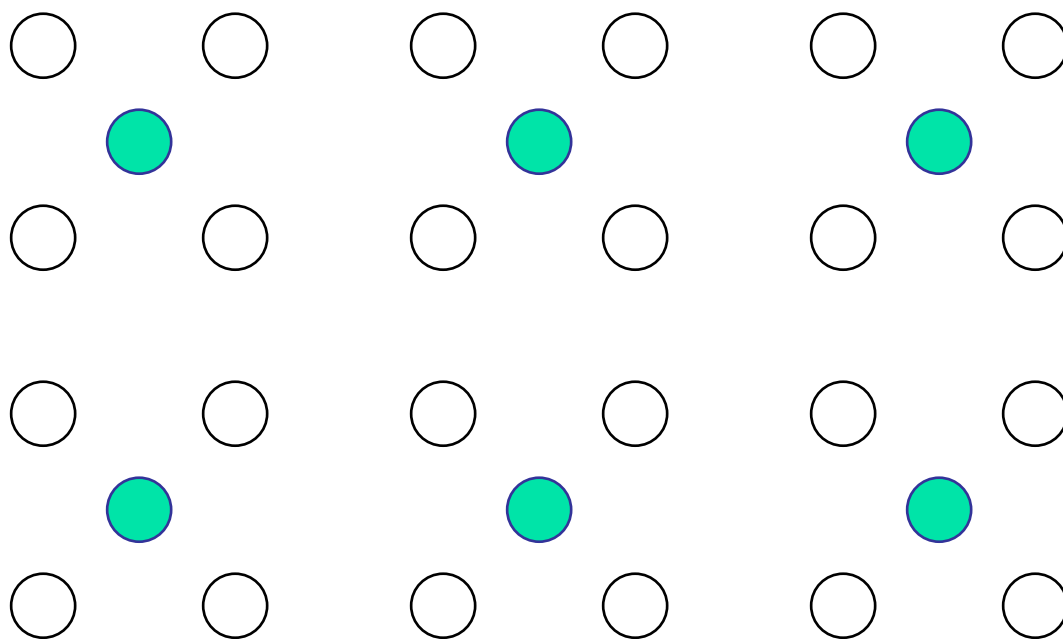
图像切片：由一个或多个连续的宏块组成

宏块：一个宏块由一个16*16的亮度信息和两个8*8的色度信息组成
（包括4个亮度块，两个色度块（一个U块，一个V块））

4.11 图像压缩标准

4.11.4 运动图象压缩国际标准压缩

MPEG的数据流结构:



○ Y (亮度)

● U、V (色差)

4.11 图像压缩标准

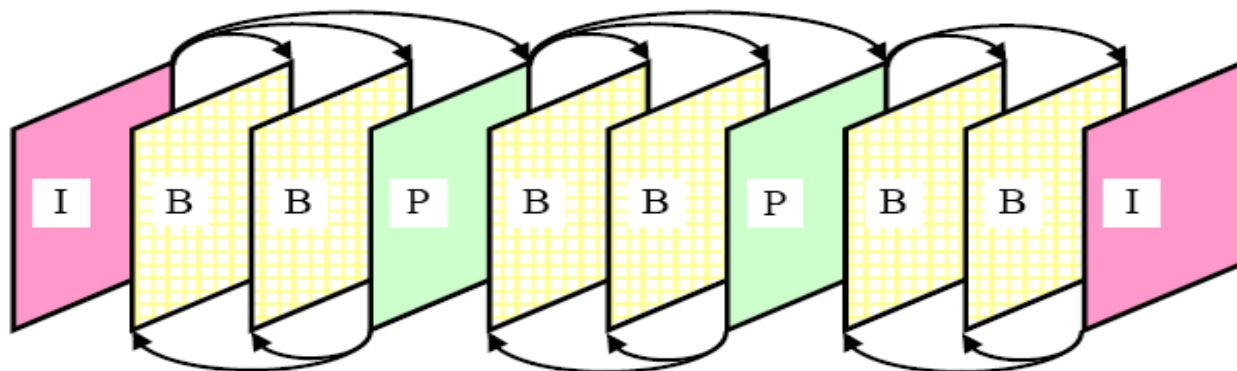
4.11.4 运动图像压缩国际标准压缩

图像类型：

帧内图（I）：压缩比较小，随机访问点

预测图（P）：以参照图（前一个I或P）为基础进行编码，压缩比中等

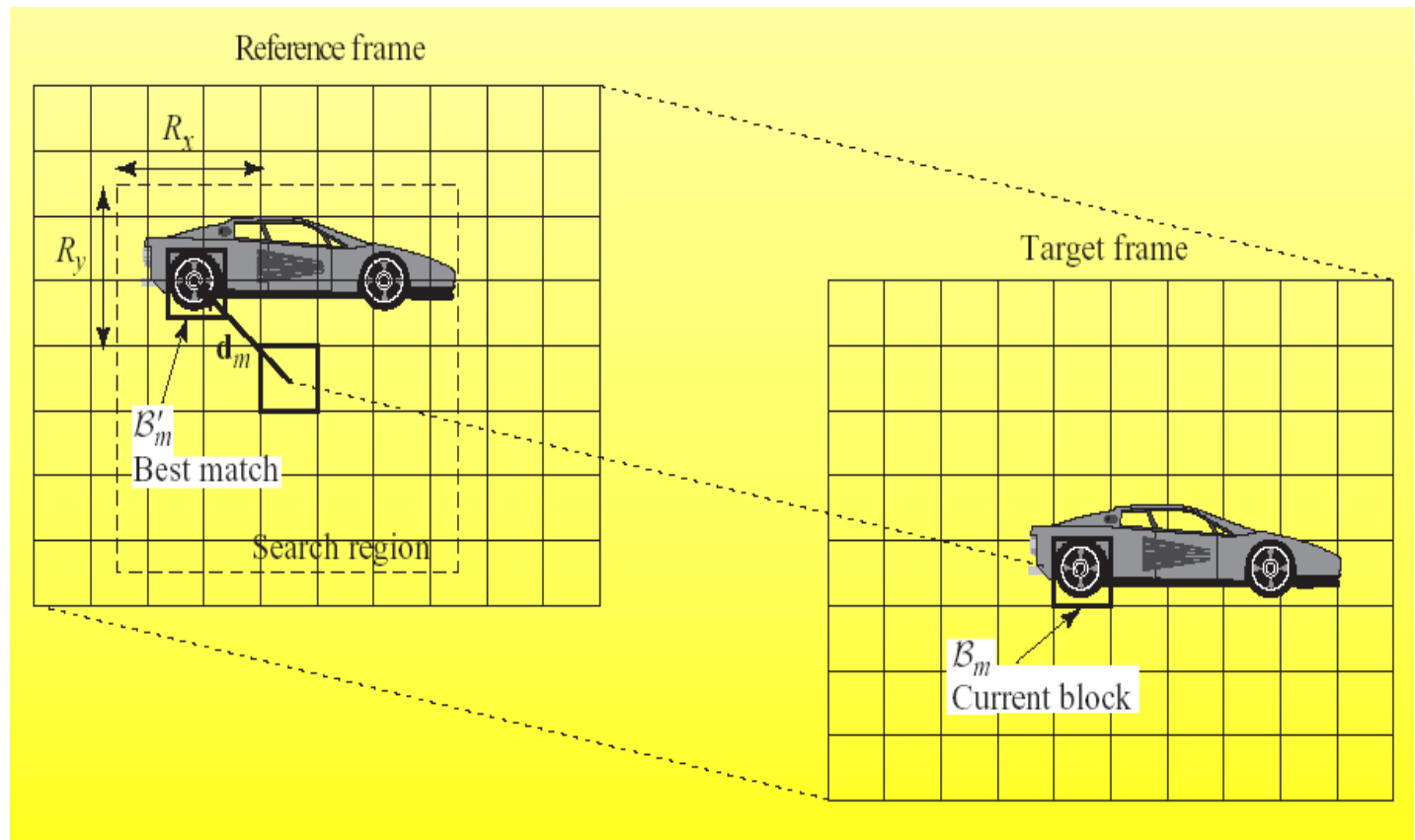
内插图（B）：最高压缩比



4.11

图像压缩标准

4.11.4 运动图像压缩国际标准压缩





4.11

图像压缩标准

4.11.4 运动图像压缩国际标准压缩

运动估计准则

$$NCCF(i, j) = \min \sum_{m=1}^M \sum_{n=1}^N [f_K(m, n) - f_L(m, n)]^2, (i, j) \in SR$$

$$\text{或 } NCCF(i, j) = \min \sum_{m=1}^M \sum_{n=1}^N |f_K(m, n) - f_L(m, n)|, (i, j) \in SR$$

4.11 图像压缩标准

4.11.4 运动图象压缩国际标准压缩



4.11

图像压缩标准

4.11.4 运动图像压缩国际标准压缩

宏块类型	预测器表达式	预测误差
前向预测	$\hat{I}_1(X) = I_0(X + mv_{01})$	$I_1(X) - \hat{I}_1(X)$
后向预测	$\hat{I}_1(X) = I_2(X + mv_{21})$	$I_1(X) - \hat{I}_1(X)$
双向预测	$\hat{I}_1(X) = \frac{1}{2}[I_0(X + mv_{01}) + I_2(X + mv_{21})]$	$I_1(X) - \hat{I}_1(X)$

I_0 : 前一参考帧

I_1 : 当前帧

I_2 : 后一参考帧

\hat{I}_1 : 预测值

X : 当前帧被预测点的坐标

mv_{01} : 该点相对于 I_0 的运动向量（位移向量）

mv_{21} : 该点相对于 I_2 的运动向量



4. 11

图像压缩标准

4.11.4 运动图像压缩国际标准压缩

帧内编码技术:

- 主要步骤;
- (1) DCT (计算变换系数、变换系数量化、Z行扫描重组数据)
 - (2) 量化 (视觉敏感部位对应块精确编码)
 - (3) 行程编码
 - (4) 哈夫曼编码

4.11

图像压缩标准

4.11.4 运动图像压缩国际标准压缩

- ◆ 恒定码率（1.5Mbps），良好的视觉质量
- ◆ 视频格式：CIF（ 352×288 ），SIF（ 352×240 ）
- ◆ 半像素精度的运动估计，MV范围[-64, 64]
- ◆ B帧编码模式：采用双向预测

◆ 内部编码模式中，DCT系数根据HVS制定的权值矩阵自适应量化，解码图像的PSNR尽管降低了，却增强了视觉效果

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83



4.11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩

H.261

➤ 主要为电视会议等应用而制定

➤ $P \times 64$ 标准 ($P = 1, 2, \dots, 30$)

➤ 视频格式: CIF (352×288 , $>128\text{Kbps}$)

QCIF (176×144 , $64-128\text{ Kbps}$)

➤ 最基本步骤有2个: I-frame和P-frame

(1) 对序列中的第一帧 (或某参考帧) 图用类似于JPEG中用的DCT压缩, 以减少帧内冗余度

(2) 估计目标的运动 (通过计算帧间的相关), 以确定如何压缩下一帧 以减少帧间冗余度

4.11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩 H.261

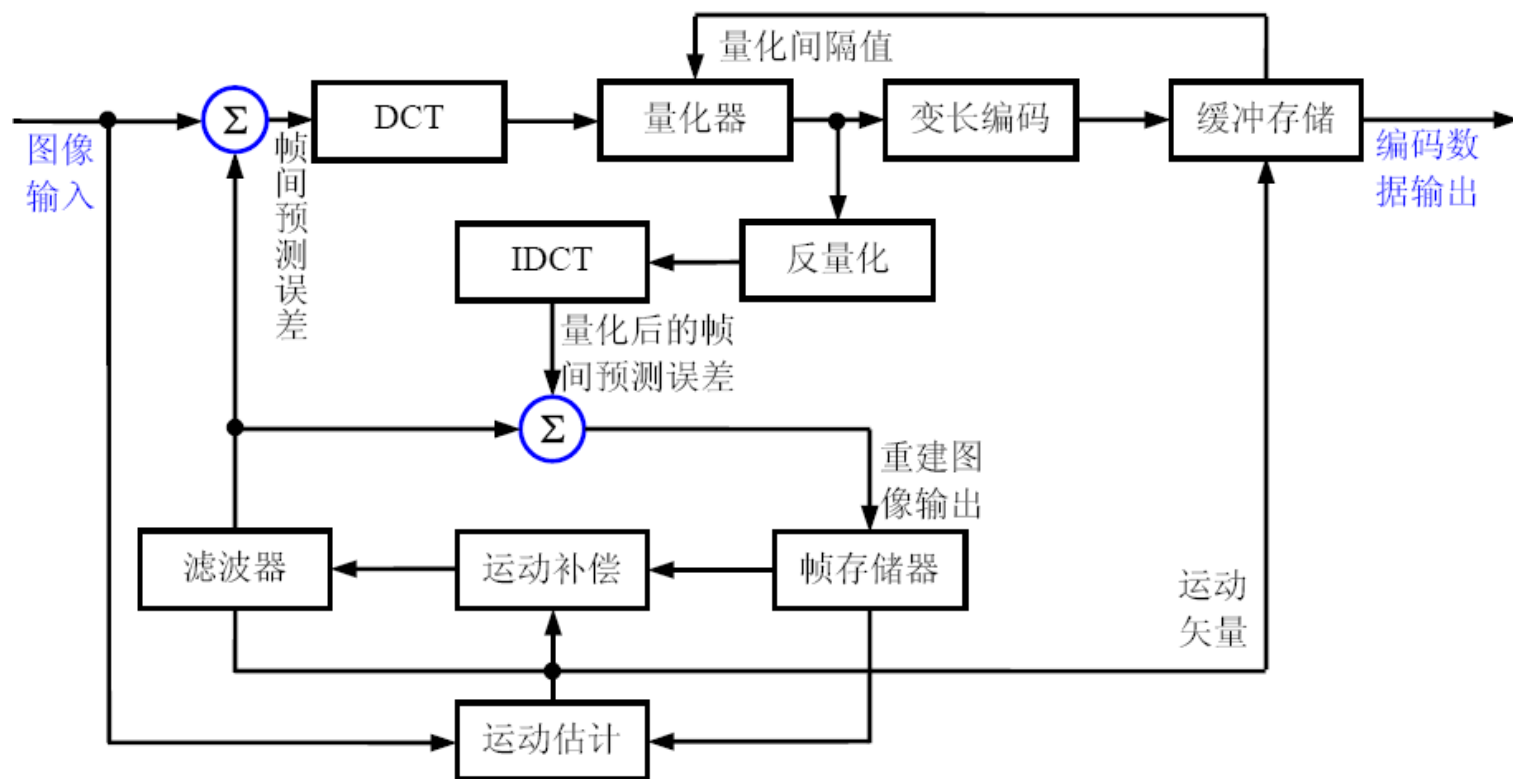


图 A.4.1 国际标准 H.261 编码器的基本框图

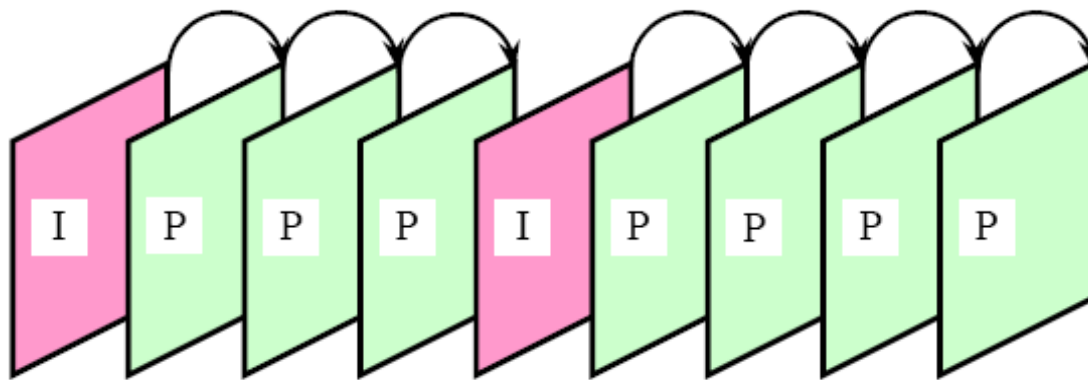
4.11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩

H.261

编（解）码序列的结构



在每个I-帧后面接续若干个P-帧，I-帧独立编码，而P-帧则参照上一帧编码



4.11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩 MPEG-2

DVD , 音视频广播 (**DTV**, **HDTV**, 卫星传输, 高速局域网、internet), DTV 4-8Mbps, HDTV 18-45Mbps

与MPEG1相比:

- 目标为高分辨率 (4CIF) 的隔行扫描序列
- 复杂高效的运动估计方法 (帧预测和场预测) 以提高逐行扫描序列的预测精度
- 应用于隔行序列的场DCT和DCT系数扫描方法
- 提供了质量SNR可分级、空间可分级、时间可分级和数据划分等分级模式
- 有多种框架和级别以适应不同应用



4.11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩 MPEG-4 (ISO/IEC 14496)

➤对各种AV进行统一编码和操作

① $< 64 \text{ k bit/s}$

② $64 \sim 384 \text{ k bit/s}$

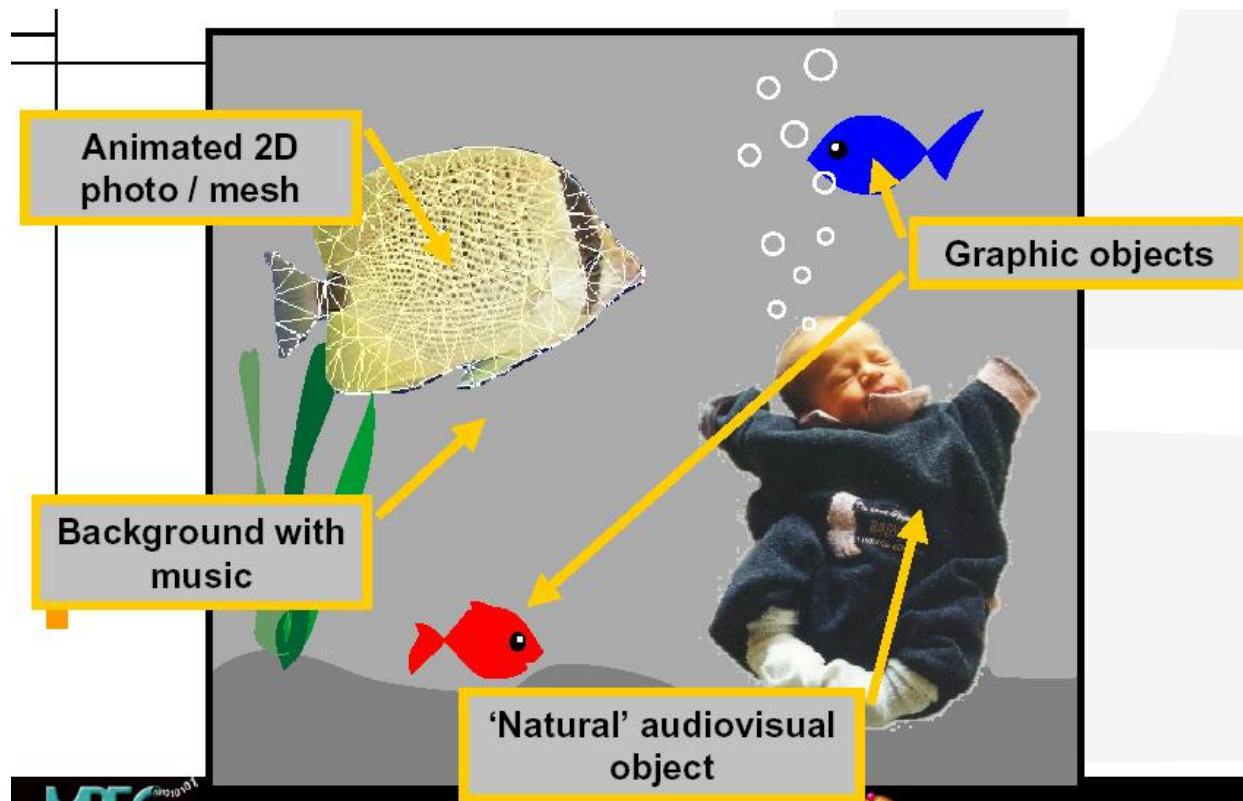
③ $384 \text{ k bit/s} \sim 4 \text{ M bit/s}$

➤基于对象的编码

4.11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩 MPEG-4 (ISO/IEC 14496)





4. 11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩

H.264 （ ISO/IEC 14496-AVC, ITU-T H.264 ）

- 在提高压缩效率的同时，提供网络友好的视频表达方式
- 既支持“会话式”（如可视电话）也支持“非会话式”（如广播或流媒体）视频应用



4.11

图像压缩标准

4.11.4 运动图象压缩国际标准压缩

H.264 （ ISO/IEC 14496-AVC, ITU-T H.264 ）

- 整数变换;
- 多帧多模式运动预测
- 熵编码
 - ① 一致性变长编码
 - ② 前后文自适应二值算术编码
- 自适应环内消块效应滤波器
- 比MPEG-4可节省40%到50% 的比特率



4. 11

图像压缩标准

4.11.5 研究方向

在MPEG-4或H.264标准的框架下

- 1、基于块匹配的运动估计快速算法
- 2、基于视频对象分割的运动估计和编码
- 3、设计易于DSP实现的视频分割和运动估计算法
- 4、基于视频的目标跟踪和识别算法



本章小节

4.1 数据冗余与压缩

4.2 图像保真度

4.3 无失真编码定理

4.4 哈夫曼编码

4.5 算术编码

4.7 无损预测编码

4.8 有损预测编码

4.9 变换编码



作业

6.1.3

6.2.1

6.3.3

6.4.2

6.5.1

6.5.3

6.8.3