

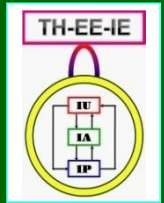
图象工程（上）

图 象 处 理

（第4版）

章毓晋

清华大学电子工程系 100084 北京

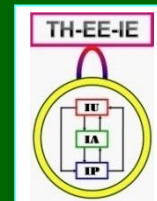


第3单元 图象编码

- 第9章 图象编码基础
- 第10章 图象变换编码
- 第11章 其他图象编码方法

图象编码的目的是在保证一定视觉质量的前提下减少数据量（从而也减少图象传输所需的时间），这也可看作使用较少的数据量来获得较好的视觉质量

图象编码以信息论为基础，以压缩数据量为主要目的，也常被称为图象压缩



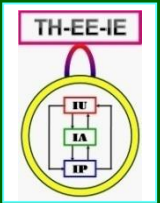
第9章 图象编码基础

9.1 图象压缩原理

9.2 编码定理

9.3 位平面编码

9.4 变长编码



9.1 图象压缩原理

动机/原因： 表达数字图象所需数据量通常很大

图象编码：

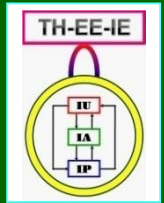
采用对图象的新的表达方法以减小所需的数据量

◆ **数据和信息：** 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**

◆ 图象压缩（编码）和图象解压缩（解码）



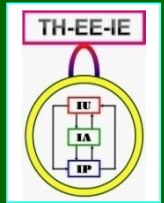
9.1 图象压缩原理

在保持图象中原有信息或让信息损失满足应用要求的基础上尽可能地减少数据量

9.1.1 数据冗余

9.1.2 图象编解码

9.1.3 图象保真度和质量



9.1.1 数据冗余

数据冗余的概念

数据是信息的载体

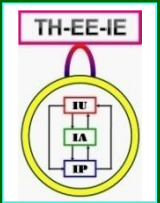
同量的数据可表达不同量的信息

同量的信息可用不同量的数据表达

冗余

数据表达了无用的信息

数据表达了已表达的信息



9.1.1 数据冗余

数据冗余类别

(1) 心理视觉冗余

与主观感觉有关

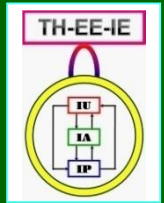
(2) 像素相关冗余

空间冗余，几何冗余

(3) 编码冗余

与灰度分布的概率特性有关

减少/消除其中的一种/多种冗余，就能取得数据压缩的效果



9.1.1 数据冗余

1. 心理视觉冗余

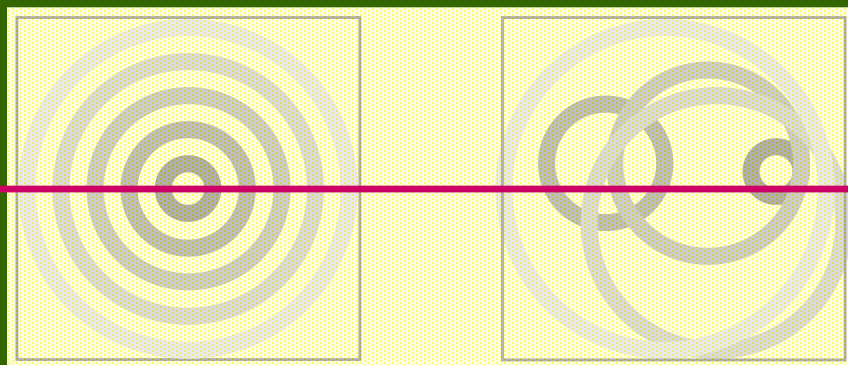
- 主观：因人而异，因应用要求而异
- 其存在与人观察图象的方式有关
 - 眼睛对某些视觉信息更敏感
 - 人对某些视觉信息更关心
- 心理视觉冗余与实在的视觉信息有联系
(如有损失则不可逆转，也称量化)

9.1.1 数据冗余

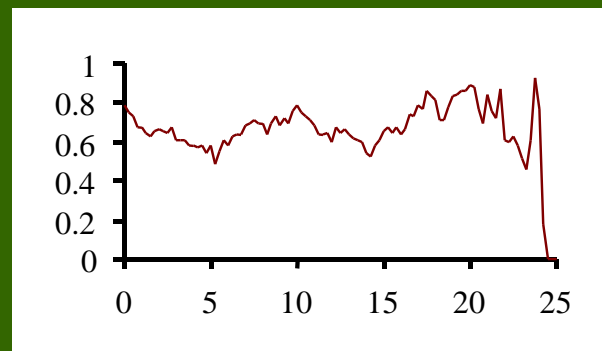
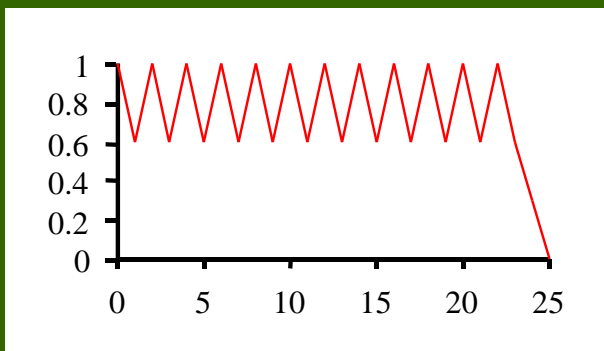
2. 像素间冗余

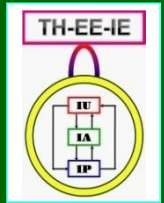
直接与像素之间的相关性联系

规则
冗余大



不规则
冗余小





9.1.1 数据冗余

3. 编码冗余

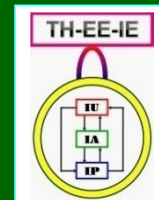
编码：建立码本来表达数据

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

码长（码字的长度，字长）：

每个码字里的符号个数



9.1.1 数据冗余

3. 编码冗余

图象中灰度（对灰度值编码）出现的概率

$$p_s(s_k) = n_k / N \quad k = 1, 2, \dots, L-1$$

不同灰度出现的概率不同，表达比特数可不同

平均比特数

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

为表示 s_k 所需的比特数

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

9.1.2 图象编解码

1、图象编解码过程

通过对原始图象的**编码**以达到减少数据量的目的（压缩过程），然后为了实际应用的需要对编码结果进行**解码**来恢复图象形式以使用解码图象（解压过程）

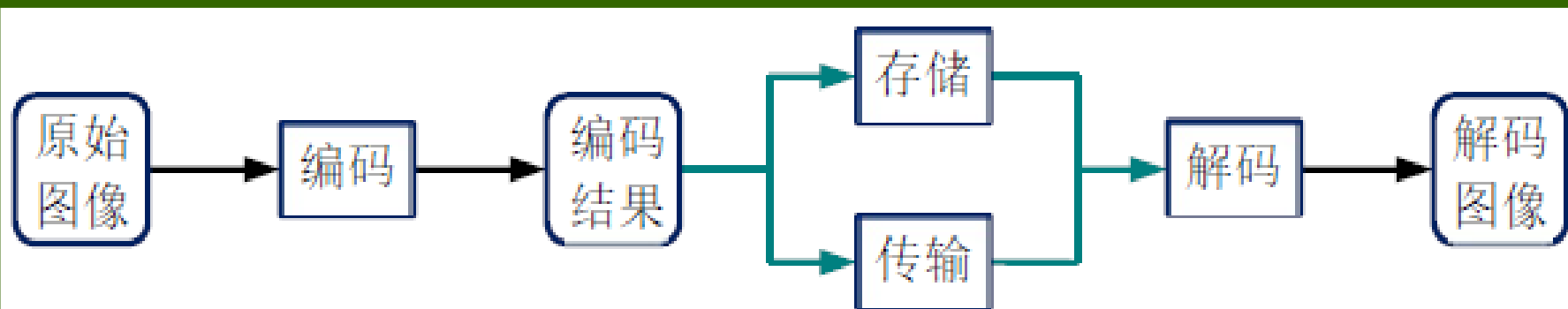


图 9.1.2 图像编解码过程

9.1.2 图象编解码

2、图象编解码系统

编码器包括顺序完成三个独立操作的模块，而对应的解码器仅包含反序完成两个独立操作的模块

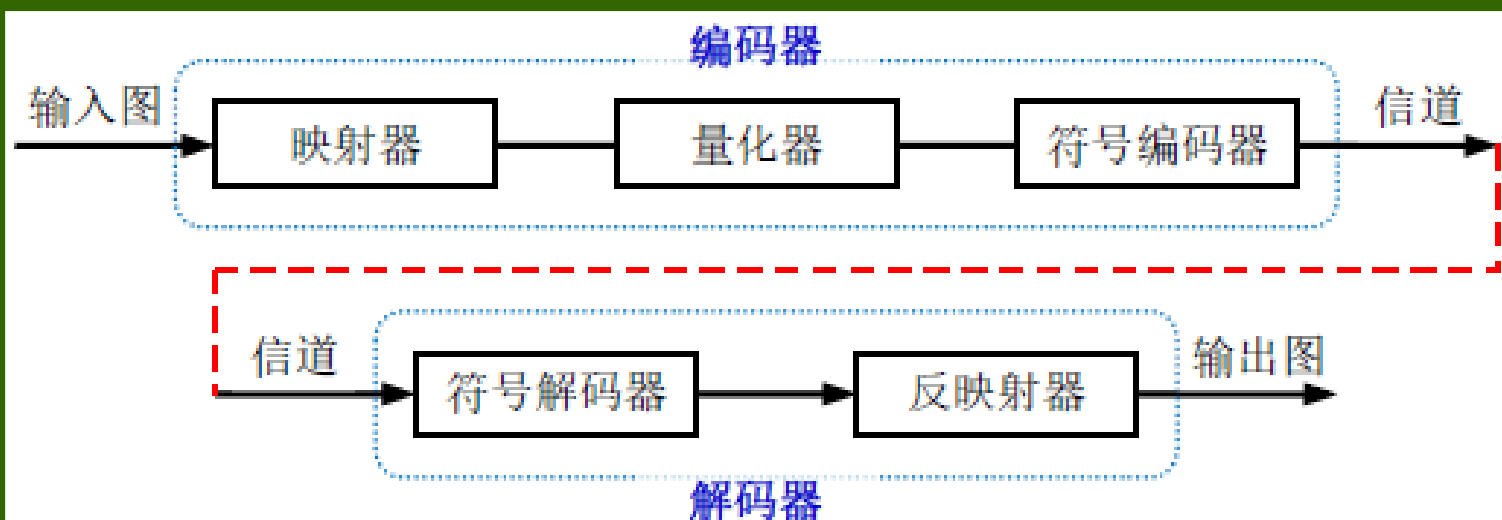
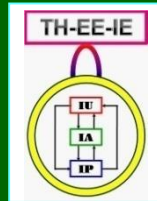


图 9.1.4 编码器和解码器及其模块



9.1.3 图象保真度和质量

图象压缩方法的分类：

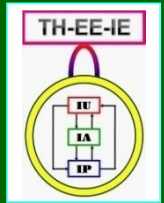
信息保存型：

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

信息损失型：

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

准无损（near-lossless）：{11.5节}



9.1.3 图象保真度和质量

图象保真度

图象编码方法：信息保存型/信息损失型

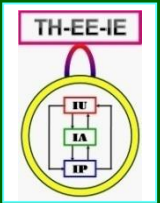
图象保真度描述解码图象相对于原始图象的偏离程度，是对信息损失的一种测度

➤ 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之间的某个确定函数表示损失的信息量，便于计算或测量



9.1.3 图象保真度和质量

1. 客观保真度

点误差

$$e(x, y) = g(x, y) - f(x, y)$$

图误差

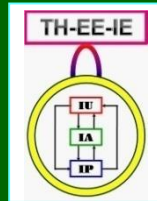
$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |g(x, y) - f(x, y)|$$

均方根误差

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

均方信噪比

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



9.1.3 图象保真度和质量

1. 客观保真度

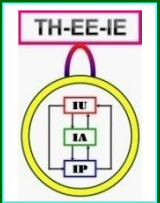
(归一化) 信噪比: 令
$$h = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

单位: 分贝 (dB)

峰值信噪比:

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

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



9.1.3 图象保真度和质量

2. 主观保真度

很多解压图象最终是供人看的

(1) 损伤检验 (impairment tests)

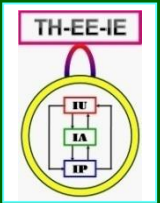
观察者对图象根据其损伤程度打分

(2) 质量检验 (quality tests)

观察者对图象根据其质量排序

(3) 对比测试 (comparison tests)

观察者对图象进行两两比较 (相对优劣)



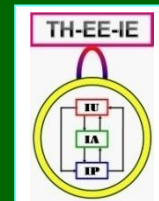
9.2 编码定理

{信息论是图象编码的基础}

9.2.1 信息单位和信源描述

9.2.2 无失真编码定理

9.2.3 率失真编码定理



9.2.1 信息单位和信源描述

信息量

概率为 $P(E)$ 的随机事件 E 的信息量

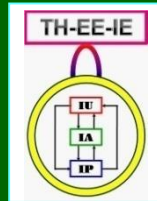
$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

$I(E)$ 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



9.2.1 信息单位和信源描述

信源（一幅图象可看作一个信源）

信源符号集： $S = \{s_1, s_2, \dots, s_J\}$

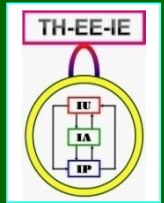
概率矢量： $\mathbf{u} = [P(s_1) \ P(s_2) \ \dots \ P(s_J)]^T$

✓ 用 (S, \mathbf{u}) 可以完全描述信源的输出

信源平均信息（熵，不确定性）

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

{例9.2.1}



9.2.1 信息单位和信源描述

编码输出

编码输出符号集: $T = \{t_1, t_2, \dots, t_K\}$

概率矢量: $\mathbf{v} = [P(t_1) \ P(t_2) \ \dots \ P(t_K)]^T$

✓ 用 (T, \mathbf{v}) 可以完全描述编码输出

编码输出的概率 $P(t_k)$ 和信源的概率分布联系

$$P(t_k) = \sum_{j=1}^J P(t_k | s_j) P(s_j)$$

9.2.1 信息单位和信源描述

编码输出

将条件概率放入一个 $K \times J$ 的传递矩阵 Q

$$q_{kj} = P(t_k | s_j)$$

$$Q = \begin{bmatrix} P(t_1 | s_1) & P(t_1 | s_2) & \cdots & P(t_1 | s_J) \\ P(t_2 | s_1) & \ddots & \cdots & P(t_2 | s_J) \\ \vdots & \vdots & \ddots & \vdots \\ P(t_K | s_1) & P(t_K | s_2) & \cdots & P(t_K | s_J) \end{bmatrix}$$

编码输出符号
集的概率分布

$$v = Qu$$

编码输入符号
集的概率分布

9.2.1 信息单位和信源描述

互信息

对应每个 t_k 有一个条件熵函数

输出 t_k 条件下 s_j 产生的概率

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

$H(u|t_k)$ 对所有 t_k 的期望值

$$H(u|v) = \sum_{k=1}^K H(u|t_k) P(t_k) = - \sum_{j=1}^J \sum_{k=1}^K P(s_j, t_k) \log P(s_j|t_k)$$

$$q_{kj} = \frac{P(t_k/s_j)}{P(t_k)}$$

$$I(u, v) = H(u) - H(u|v) = \sum_{j=1}^J \sum_{k=1}^K P(s_j, t_k) \log \frac{P(s_j, t_k)}{P(s_j)P(t_k)} = \sum_{j=1}^J \sum_{k=1}^K P(s_j) q_{kj} \log \frac{q_{kj}}{\sum_{i=1}^J P(s_i) q_{ki}}$$

9.2.2 无失真编码定理

在没有失真的条件下（无损压缩），编码表达每个信源符号时可达到的最小平均码字长度
信源输出 n 个一块（组）的随机变量

$$P(\sigma_i) = P(s_{j1})P(s_{j2}) \cdots P(s_{jn})$$

信源的熵

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

产生块随机变量的信源的熵是对应单符号信源的 n 倍。它也可看作是单符号信源的 n 阶扩展

9.2.2 无失真编码定理

用长度为 $l(\sigma_i)$ 的整数码字来对 σ_i 编码

自信息

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

乘以 $P(\sigma_i)$

对所有 i 求和

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

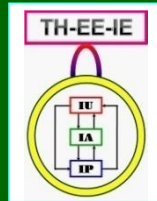
取极限

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

$H(u)$ 是 L'_{avg}/n 的下限，所以效率 $\eta (\leq 1)$ ：

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

无损信源压缩的极限



9.2.2 无失真编码定理

压缩率（另一种定义方式）

一幅有 G 个灰度级的图象 $f(x, y)$ ，如果灰度级 k 的出现概率为 $P(k)$ ，该图象的熵为

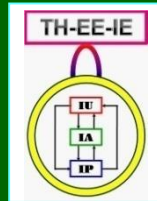
$$H = - \sum_{k=0}^{G-1} P(k) \log P(k)$$

如果表示该图象所需的最小数据量或比特数为 B ，则（绝对的）冗余 R_a 是 $R_a = B - H$

用图象的灰度直方图来估计图象的熵

$$H_e = - \sum_{k=0}^{2^B-1} h(k) \log h(k)$$

冗余 $R_a \approx B - H_e$ ，绝对的压缩率 $C_a \approx B / H_e$



9.2.3 率失真编码定理

将对固定字长编码方案的失真（重建误差） D 与编码所用的数据率（如每像素比特数） R 联系在一起。它给出由于压缩而产生的平均误差被限制在某个最大允许水平 D 时的最小的 R

用重建的均方误差作为失真度

$$D = E\{[\hat{f}(x, y) - f(x, y)]^2\}$$

可以证明，重建误差的熵有如下的上限

$$H[f(x, y) - \hat{f}(x, y)] \leq \frac{1}{2} \log [2\pi e D_{\max}]$$

9.2.3 率失真编码定理

率失真函数

$$R(D) = \min_{Q \in Q_D} [I(u, v)]$$

失真平均值

$$Q_D = \{q_{kj} \mid d(Q) \leq D\}$$

$R(D)$ 给出在平均失真小于或等于 D 时，信源可以传送给编码输出的最小平均信息量

计算 $R(D)$ 的3个约束条件

$$q_{kj} \geq 0$$

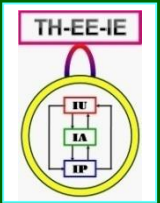
Q 的元素必须是正的

$$\sum_{k=1}^K q_{kj} = 1$$

Q 的任一系列之和为1

$$d(Q) = D$$

如果允许最大可能的失真，就可获得最小的数据率



9.3 位平面编码

不仅能消除或减少编码冗余也能消除或减少
图象中的像素间冗余

9.3.1 位平面的分解

9.3.2 位平面的编码

9.3.1 位平面的分解

1. 二值分解

多项式

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

每个比特表示一个位面

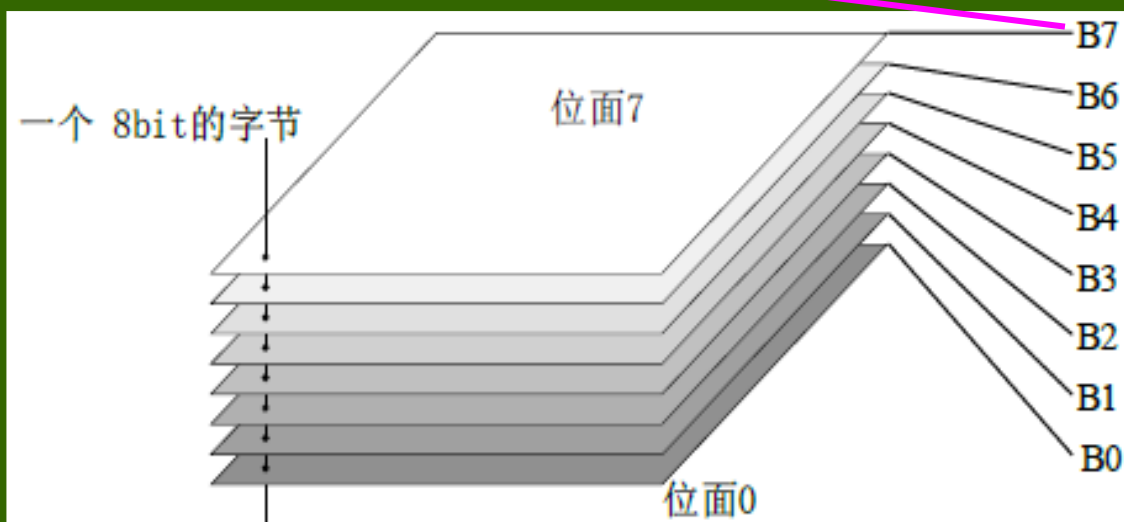


图 9.3.1 图像的位面表示

9.3.1 位平面的分解

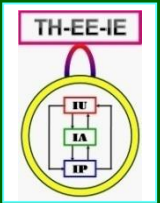
2. 灰度码分解

减少小灰度值变化影响的位面分解法

异或

$$g_i = \begin{cases} a_i \oplus a_{i+1} & 0 \leq i \leq m-2 \\ a_i & i = m-1 \end{cases}$$

	二值码	灰度码
127	01111111 ₂	01000000 ₂
128	10000000 ₂	11000000 ₂



9.3.2 位平面的编码

1. 常数块编码 (CAC)

- 用专门的码字表达全是0或1的连通区域
- 将图象分成全黑，全白或混合的 $m \times n$ 尺寸块
- 出现频率最高的类赋予1 bit 码字0

其它两类分别赋予2 bit 码字10和11 {P.210}

压缩： 原需用 mn 比特表示的常数块现在只用
1 bit 或2 bit 码字表示

9.3.2 位平面的编码

2. 1-D游程编码 (RLC)

设每行均由0游程开始

对第2位平面 (最高位):

第 2 位平面

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

4 2 2, 3 3 2, 3 4 1, 4 2 2

对第1位平面 (中间位):

8, 3 1 4, 1 1 1 2 2 1, 0 6 2

对第0位平面 (最低位):

0 1 7, 0 1 2 1 4, 0 1 2 1 1 1 1 1, 8

不同位平面的复杂度

12

13

17

9.3.2 位平面的编码

3. 2-D游程编码

相对地址编码 (RAC)

跟踪各个0和1游程的起始和终结的过渡点，算出各对点之间的距离，确定RAC距离
取 cc' 和 ec 中小的

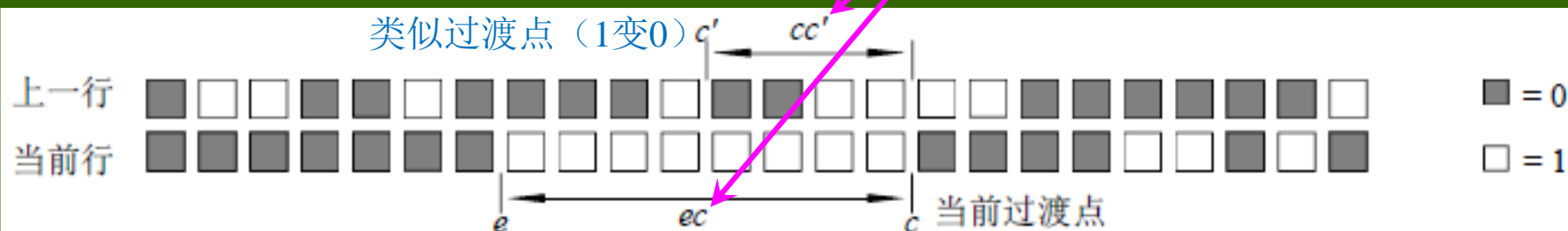


图 9.3.4 RAC 编码示意图

9.3.2 位平面的编码

3. 2-D游程编码

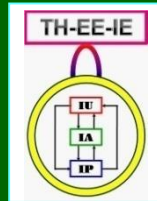
例9.3.3 (参见图9.3.4)

$cc' = 4 \Rightarrow 1100$, $h(4) \Rightarrow 011$

对RAC距离进行变长编码

$d-1=3$

测量距离	RAC 距离	第 1 前缀码	距离范围	第 2 前缀码 $h(d)$
cc'	0	0	1 ~ 4	0 xx
ec 或 cc' (左)	1	100	5 ~ 20	10 xxxx
cc' (右)	1	101	21 ~ 84	110 xxxxxx
ec	$d (d > 1)$	111 $h(d)$	85 ~ 340	1110 xxxxxxxx
cc' (c' 在左)	$d (d > 1)$	1100 $h(d)$	341 ~ 1364	11110 xxxxxxxxxx
cc' (c' 在右)	$d (d > 1)$	1101 $h(d)$	1365 ~ 5460	111110 xxxxxxxxxx



9.4 变长编码

{减少编码冗余，信息保存型}

9.4.1 哥伦布编码

9.4.2 哈夫曼编码

9.4.3 香农-法诺编码

9.4.4 算术编码

9.4.1 哥伦布编码

给定一个非负整数 n 和一个正整数除数 m ，将 n 相对于 m 的哥伦布码记为 $G_m(n)$ ，它是对商 $\lfloor n/m \rfloor$ 的一元码和对余数 $n \bmod m$ 的二值表达的组合。 $G_m(n)$ 可根据以下三个步骤计算 {对照例9.4.1, $G_4(9)$ }

- 110 (1) 构建商 $\lfloor n/m \rfloor$ 的一元码 (整数个1后跟个0)
- 01 (2) 令 $k = \lceil \log_2 m \rceil$, $c = 2^k - m$, $r = n \bmod m$, 计算截断的 r' :

$$r' = \begin{cases} r \text{ 截断到 } k-1 \text{ 比特} & 0 \leq r < c \\ r + c \text{ 截断到 } k \text{ 比特} & \text{其他} \end{cases}$$

- 11001 (3) 将上两步骤的结果拼接起来得到 $G_m(n)$

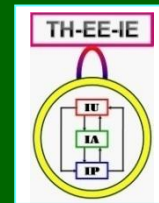
9.4.1 哥伦布编码

哥伦布-莱斯码

在 $m = 2^k$, $c = 0$ 时, $r' = r = n$ 都截断到 k 比特

表 9.4.1 若干哥伦布码示例

n	$G_1(n)$	$G_2(n)$	$G_4(n)$	$G_{\text{exp}}^0(n)$
0	0	00	000	0
1	10	01	001	100
2	110	100	010	101
3	1110	101	011	11000
4	11110	1100	1000	11001
5	111110	1101	1001	11010
6	1111110	11100	1010	11011
7	11111110	11101	1011	1110000
8	111111110	111100	11000	1110001
9	1111111110	111101	11001	1110010



9.4.1 哥伦布编码

阶为 k 的指数哥伦布码 $G_{\text{exp}}^k(n)$ {例9.4.2, $G_{\text{exp}}^0(8)$ }

(1) 确定满足下式的整数 $i \geq 0$

$$\sum_{j=0}^{i-1} 2^{j+k} \leq n < \sum_{j=0}^i 2^{j+k}$$

$$i = \lfloor \log_2 9 \rfloor = 3$$

并构建 i 的一元码

1110

(2) 计算下式的二值表达

$$n - \sum_{j=0}^{i-1} 2^{j+k}$$

0001

并将其截断到最低的 $k+i$ 比特

001

(3) 将上两个步骤的结果拼接起来 1110001

9.4.2 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，然后从小到大依次结合

初始信源		信源消减步骤			
符号	概率	1	2	3	4
s_2	0.4	0.4	0.4	0.4	0.6
s_6	0.3	0.3	0.3	0.3	0.4
s_1	0.1	0.1	0.2	0.3	直到剩下两个
s_4	0.1	0.1	0.1		
s_3	0.06	0.1			
s_5	0.04				

9.4.2 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			信源消减步骤			
符号	概率	码字	1	2	3	4
s_2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
s_6	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
s_1	0.1	011	0.1 011	0.2 010	0.3 01	
s_4	0.1	0100	0.1 0100	0.1 011		
s_3	0.06	01010	0.1 0101			
s_5	0.04	01011				

9.4.2 哈夫曼编码

哈夫曼编码结果

平均长度

$$0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5$$

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

信源熵

$$H(u) = - \sum_{j=1}^J P(a_j) \log_2 P(a_j) = 2.14$$

编码效率

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

9.4.2 哈夫曼编码

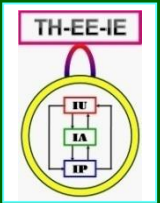
哈夫曼码特点

- 块码（组码）：每个信源符号映射成一组码符号
- 即时码；唯一可解码（图9.4.2）

0001001 0110101001011
 s_6 s_4 s_2 s_1 s_3 s_5

- 对同一个信源可编出不同的哈夫曼码

符号	第 1 种码	第 2 种码	第 3 种码	第 4 种码
s_1	0	0	1	1
s_2	10	11	01	00
s_3	11	10	00	01



9.4.3 香农-法诺编码

- 变长编码技术，其码字中的0和1是独立的，并且基本上等概率出现

主要步骤为 (图9.4.3, 图9.4.4)：

- (1) 将信源符号依其概率从大到小排列
- (2) 将信源符号分成概率之和相接近的两部分
- (3) 分别给两部分的信源符号组合进行赋值
- (4) 如果两部分均只有一个信源符号了，编码结束，否则返回(2)继续进行

9.4.4 算术编码

编码示例

编码来自1个4-符号信源 $\{s_1, s_2, s_3, s_4\}$ 的由5个符号组成的符号序列： $s_1 s_2 s_3 s_3 s_4$

0.068

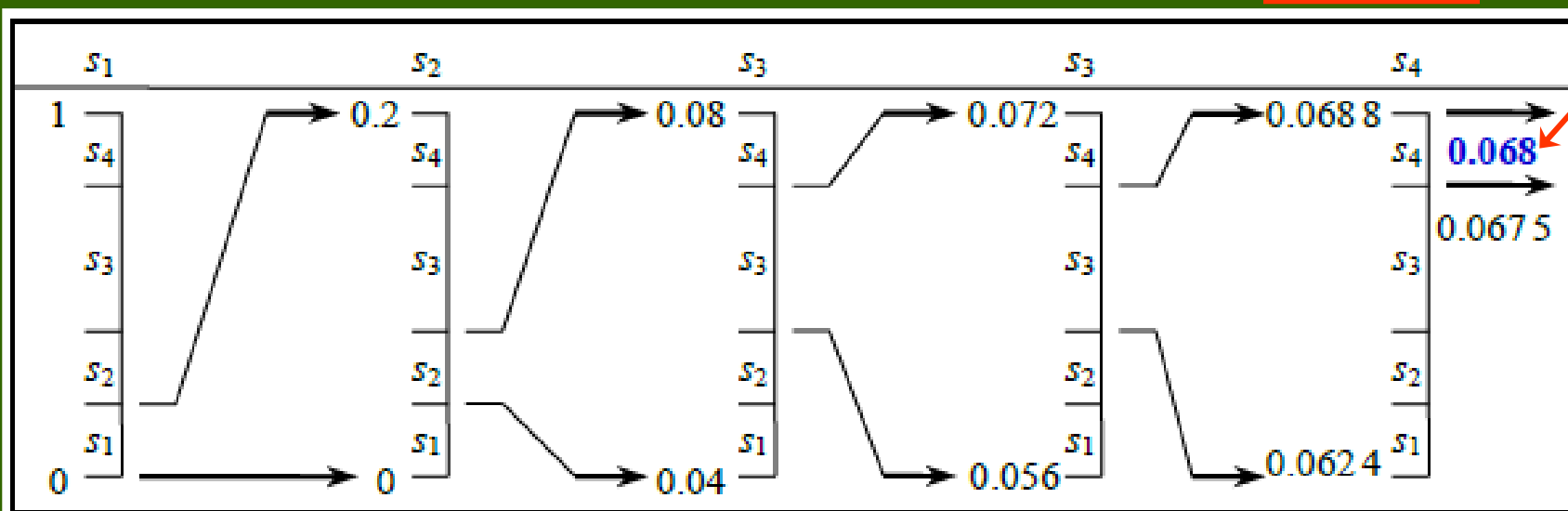


图 9.4.5 算术编码过程图解

9.4.4 算术编码

解码示例

信源符号集 $S = \{s_1, s_2, s_3, s_4\}$, $u = [0.2 \ 0.2 \ 0.4 \ 0.2]^T$ 。

原始符号序列长度为5, 编码得到码字为0.2235

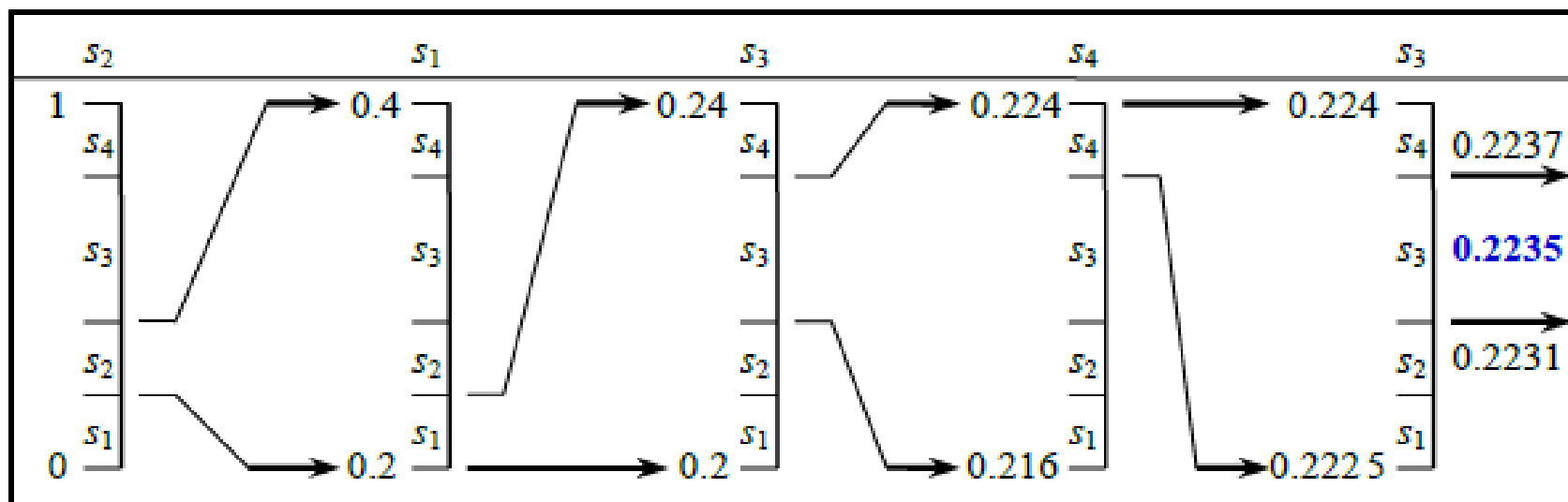


图 9.4.6 算术解码过程示例



联系信息

- ☞ 通信地址：北京清华大学电子工程系
- ☞ 邮政编码：100084
- ☞ 办公地址：清华大学，罗姆楼，6层305室
- ☞ 办公电话：(010) 62798540
- ☞ 传真号码：(010) 62770317
- ☞ 电子邮件：zhang-yj@tsinghua.edu.cn
- ☞ 个人主页：oa.ee.tsinghua.edu.cn/~zhangyujin/