

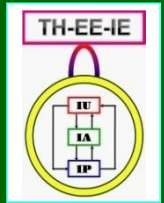
图象工程（上）

图 象 处 理

（第4版）

章毓晋

清华大学电子工程系 100084 北京

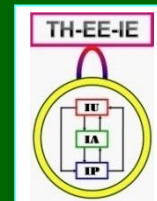


第3单元 图象编码

- 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11章 其他图象编码方法

11.1 基于符号的编码

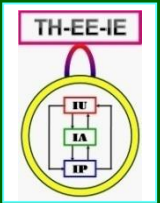
11.2 LZW编码

11.3 预测编码

11.4 矢量量化

11.5 准无损编码

11.6 比较和评述



11.1 基于符号的编码

基本思路（文本图象）

将每个文字看作一个基本符号或子图象，而将文本图象看作这些子图象的集合

需要建立一个符号字典，存储所有可能出现的符号（对每个符号赋一个码）

对图象的编码 \Rightarrow 确定每个符号的码字以及确定符号在图象中的空间位置

一幅图象可用一系列三元组来表示，即 $\{(x_1, y_1, l_1), (x_2, y_2, l_2), \dots, (\text{行}, \text{列}, \text{字典中标号})\}$

11.1 基于符号的编码

编码示例

设在需要编码的图象中有一个6字母的序列“ABABAB”。每个字母由一个 7×5 的像素矩阵来表示。设将像素矩阵用位图来表示，则每个字母对应一幅含35个像素的位图（1 bit / pixel）

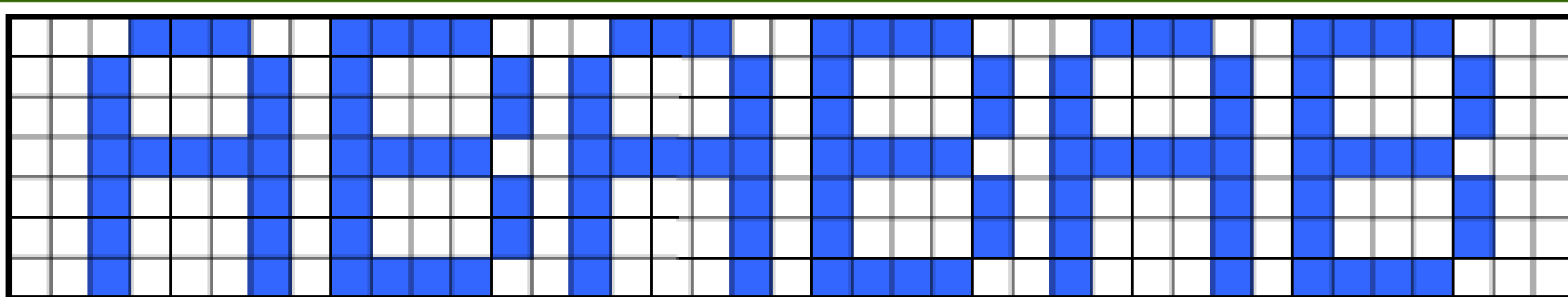
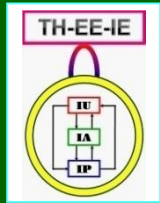


图 11.1.1 基于符号编码的示例图像

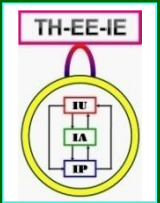


11.1 基于符号的编码

编码压缩率

原始图象共有7行39列，需 $7 \times 39 = 273$ 个比特
编码后，原始图象被表示成一个三元组序列
设对每个位置用一个字节（8个比特）来表示，每个三元组需24个比特，现有6个符号，所以需要144个比特。另外，字典需要70个比特，所以编码结果需214个比特。此时压缩率约为1.2757

将此字母序列的长度增加一倍，则压缩率会增加到 $525/358 \approx 1.4665$



11.2 LZW编码

一种信息保存型的编码方式，能消除或减少图象中的像素间冗余

LZW编码对信源所输出的不同长度的符号序列分配固定长度的码字，且不需要有关符号出现概率的知识（自适应，与哈夫曼码等不同）

在编码的开始阶段要构造一个对信源符号进行编码的（原始）码本（字典）

在编码器顺序地扫描排成串像素的灰度时，算法要确定字典中还没有出现过的灰度值序列的位置，并建立（增加）一个新的码字

11.2 LZW编码

LZW编码示例 (8 bits) 图象

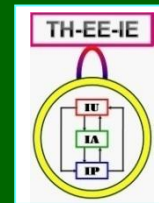
91	91	91	0
91	91	0	0
91	0	0	0
0	0	0	0

初始字典

设使用一个9比特可容纳512个字的字典。先将字典前256个位置(码字)对应分配给灰度值,后256个位置暂时空着。标256的位置用于下一个(目前尚未)出现的灰度值序列

表 11.2.1 初始字典

字典位置	0	1	...	91	...	255	256	257	...	511
字典条目	0	1	...	91	...	255	—	—	...	—



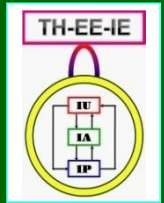
11.2 LZW编码

LZW编码示例

编码器从左向右、从上向下对每个输入象素扫描，作为**编码输入**，其灰度值如表中第2列所示。第3列的**识别序列**依次读取上一个编码输入（开始为空）。第4列的**拼接序列**是由第2列的编码输入与第3列同行中的识别序列相拼接（识别序列在前，称为前缀串；编码输入在后，称为扩展字符）而成，该序列开始时也为空。

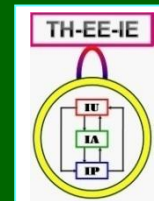
表 11.2.2 LZW 编码的步骤和结果

步骤	编码输入	识别序列	拼接序列	编码输出	字典位置	字典条目
1	91	—	—	—	—	—
2	91	91	91-91 (-)	91	256	91-91
3	91	91	91-91 (+)	—	—	—



11.2 LZW编码

表中最后两列给出了在上述过程中所有加到字典中的码字和它们所表示的序列。可见在该例中，一共增加了8个新码字（其中3个码字还没有得到使用）。在编码结束时，字典中共有264个码字，每个码字需用9个比特来表示。用这264个码字可对全图进行编码。原始图象共有16个像素，每个像素需用8个比特来表示，所以全图共需128个比特来表示。现第5列编码输出为9个码字，每个码字需用9个比特来表示，所以共需81个比特。这样，该例中压缩率为 $128/81 \approx 1.58$ ，或者说压缩比为1.58 : 1



11.2 LZW编码

LZW编码的重要特性——前缀性，即如果一个码字在字典中，那么它的前缀串也已在字典里

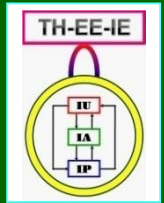
LZW编码方法是一种自适应的压缩方法，但它对输入数据的适应比较慢，因为每次字典中的条目只增加一个，而且这个条目只比原有的条目增加了一个字符

字典的尺寸是一个重要的编码器参数。如果太小，可用于匹配（放置）灰度值序列的位置会不够；如果太大，表达码字的比特数增加将影响对图象压缩的性能

11.2 LZW编码

在编码和解码的同时都建立（同样）一个码本

步骤	解码输入	拼接序列	解码输出	字典位置	字典条目
1	91	—	91	—	—
2	256 (-)	91-91	91-91	256	91-91
3	0 (+)	91-91-0	0	257	256-0
4	257 (+)	0-91	91-91-0	258	0-91
5	258 (+)	91-91-0-0	0-91	259	257-0
6	0 (+)	0-91-0	0	260	258-0
7	261 (-)	0-0	0-0	261	0-0
8	262 (-)	0-0-0	0-0-0	262	261-0
9	0 (+)	0-0-0-0	0	263	262-0



11.3 预测编码

- 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

11.3.1 无损预测编码

11.3.2 有损预测编码

11.3.1 无损预测编码

无损预测编码系统

编码器 + 解码器 (有相同的预测器)

3个基本步骤: 预测、误差映射、编码

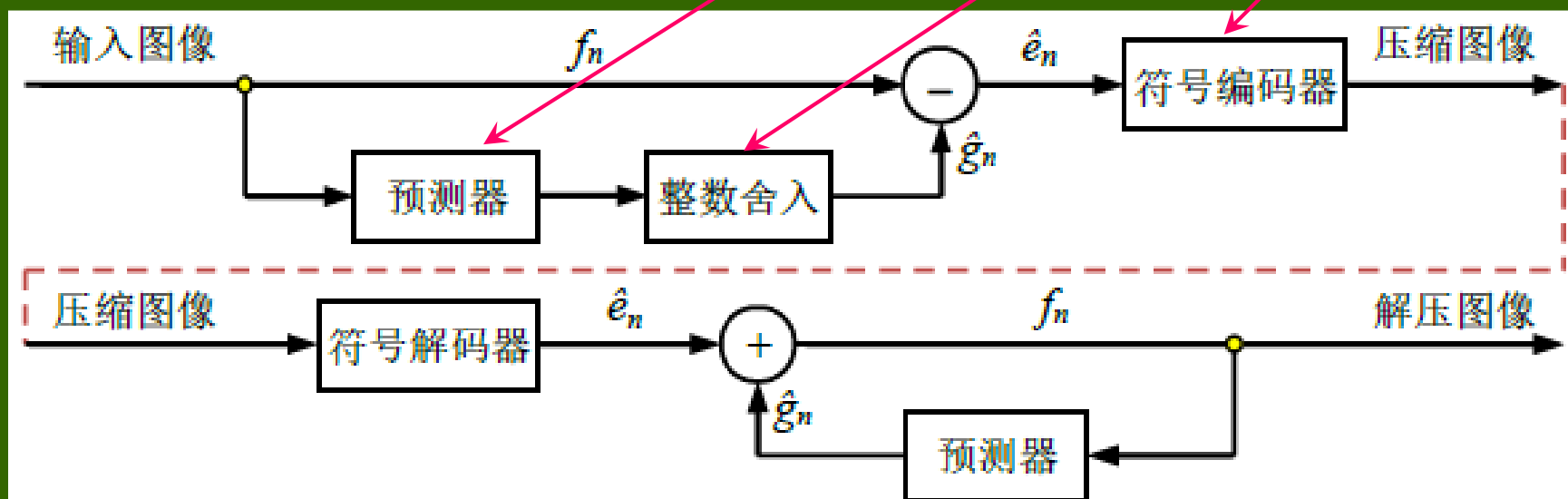
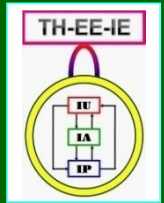


图 11.3.1 无损预测编码系统



11.3.1 无损预测编码

无损预测编码过程

输入序列: $f_n (n = 1, 2, \dots)$

预测输出: \hat{g}_n (舍入成整数)

预测误差: $\hat{e}_n = f_n - \hat{g}_n$

映射

误差编码: 在符号编码器中用变长码编误差

解压序列: $f_n = \hat{e}_n + \hat{g}_n$

哪里取得了压缩? (消除了像素间冗余)

11.3.1 无损预测编码

预测器

m 阶线性预测:
(前 m 个)

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

预测系数

1-D线性预测:
(同一行)

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

? : 每行最开始 m 个像素无法 (预测) 计算

一阶1-D线性预测: $\hat{g}_n(x, y) = \text{round} [a f(x, y-1)]$

在无损预测编码中所取得的压缩量与将输入图映射进预测误差序列所产生的熵减少量直接有关

11.3.2 有损预测编码

1. 有损预测编码系统

增加了1个量化器，预测器放在1个反馈环中

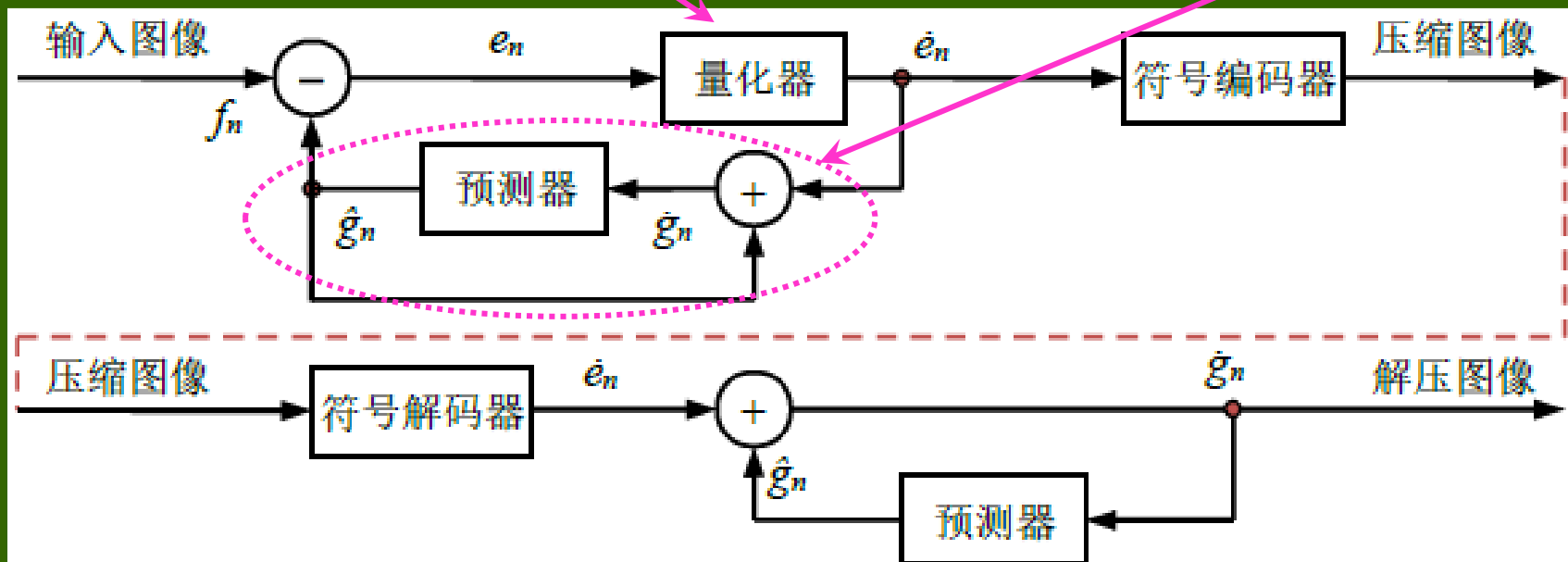
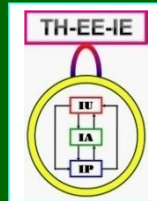


图 11.3.2 有损预测编码系统



11.3.2 有损预测编码

1. 有损预测编码系统

输入序列: $f_n (n = 1, 2, \dots)$

量化输出: $\dot{e}_n = q(e_n)$ q 为量化函数

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

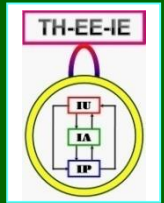
解压序列: $\dot{g}_n = \dot{e}_n + \hat{g}_n$

} 相同

编码误差: $|f_n - \dot{g}_n|$

哪里又取得了压缩?

(量化, 减少了
心理视觉冗余)



11.3.2 有损预测编码

1. 有损预测编码系统

德尔塔调制 (DM)

预测器

$$\hat{g}_n = a\dot{g}_{n-1}$$

量化器

$$\dot{e}_n = \begin{cases} +c & \text{如果 } e_n > 0 \\ -c & \text{其他} \end{cases}$$

预测系数 $a \leq 1$, 常数 $c > 0$

DM方法得到的码率是 1比特/像素

11.3.2 有损预测编码

表11.3.2 一个 DM 编码例子

($a = 1, c = 5$)

输入		编码器				解码器		误差
n	f	\hat{g}	e	\dot{e}	\dot{g}	\hat{g}	\dot{g}	$[f - \hat{g}]$
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

(11.3.7) (11.3.1) (11.3.8) (11.3.6)

11.3.2 有损预测编码

1. 有损预测编码系统

两种典型的失真（上一表格中？）

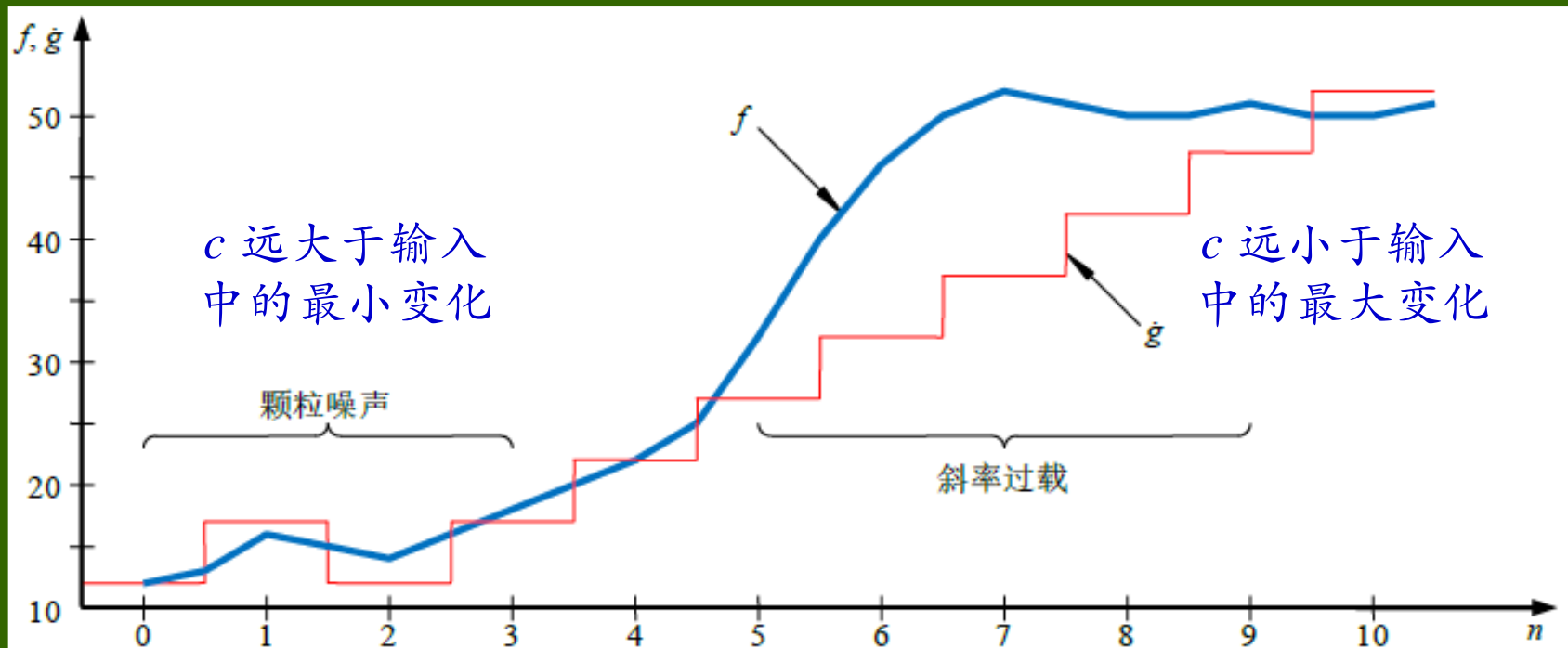


图 11.3.3 DM 编码中的失真示例

11.3.2 有损预测编码

2. 最优预测

最小化编码器的均方预测误差

预测系数

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

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

设量化误差
可以忽略

$$E\{e_n^2\} = E\{[f_n - \hat{g}_n]^2\}$$

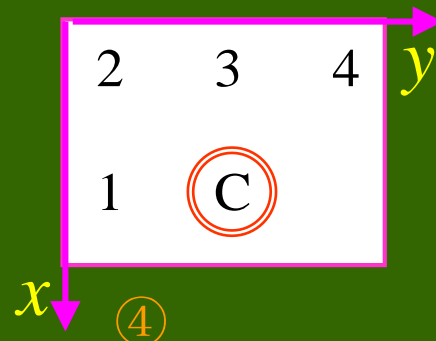
差值脉冲码调制法
(DPCM)

$$E\{e_n^2\} = E\left\{\left[f_n - \sum_{i=1}^m a_i f_{n-i}\right]^2\right\}$$

11.3.2 有损预测编码

2. 最优预测

4阶线性预测器 (2-D)



$$\hat{g}(x, y) = \overset{\textcircled{1}}{a_1} f(x, y-1) + \overset{\textcircled{2}}{a_2} f(x-1, y-1) + \overset{\textcircled{3}}{a_3} f(x-1, y) + \overset{\textcircled{4}}{a_4} f(x-1, y+1)$$

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

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

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

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

$\textcircled{3}$
 $\textcircled{2}$
 $\textcircled{1}$
 $\textcircled{2}$

11.3.2 有损预测编码

3. 最优量化

量化函数

判别电平

重建电平

$$t = q(s)$$

这个函数可完全由在第I象限的 $L/2$ 个 s_i 和 t_i 所描述。这些值给出的转折点确定了函数的不连续性并被称为量化器的判别电平和重建电平

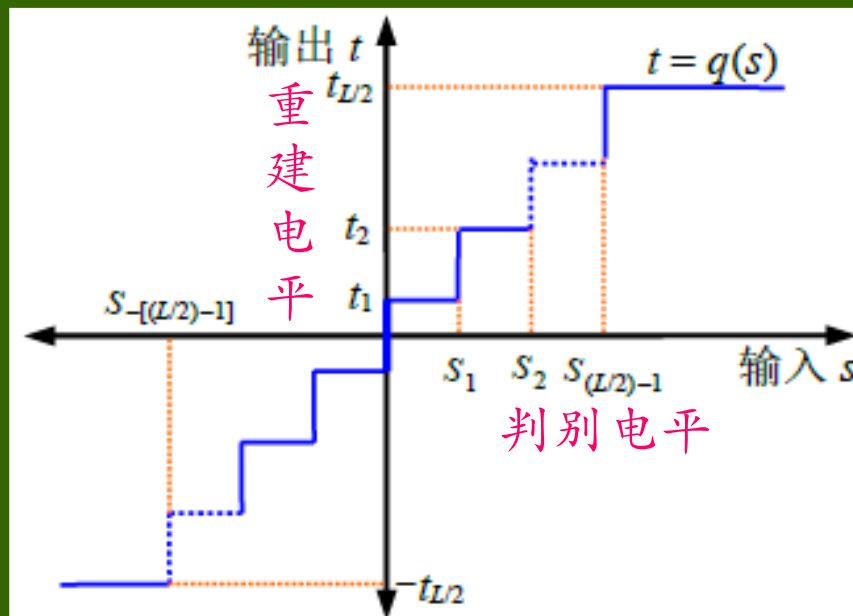


图 11.3.5 一个典型的量化函数

11.3.2 有损预测编码

3. 最优量化

最小均方量化误差

重建电平是 $p(s)$
曲线下面积的重心

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

输入概率密度
函数，偶函数

判别值为2个
重建值的中值

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

$q(s)$ 奇函数

$$s_i = -s_{-i} \quad t_i = -t_{-i}$$

{表11.3.3}

11.3.2 有损预测编码

3. 最优量化

不同量化器的效果比较

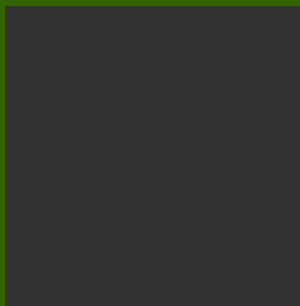
$L = 5$



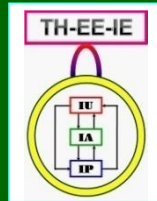
$L = 17$



$L = 9$



更多级的量化效果在视觉上并不能分辨



11.4 矢量量化

将一个有多个分量的矢量映射为一个只有较少分量的矢量

基本思路是把信源符号序列分组作为矢量看待而进行编码

矢量量化考虑了以下两个因素：

(1) 对符号串的压缩比对单符号的压缩更能取得好的效果（矢量编码比标量编码好）

(2) 对自然图象，空间上相邻的像素之间有较强的相关性

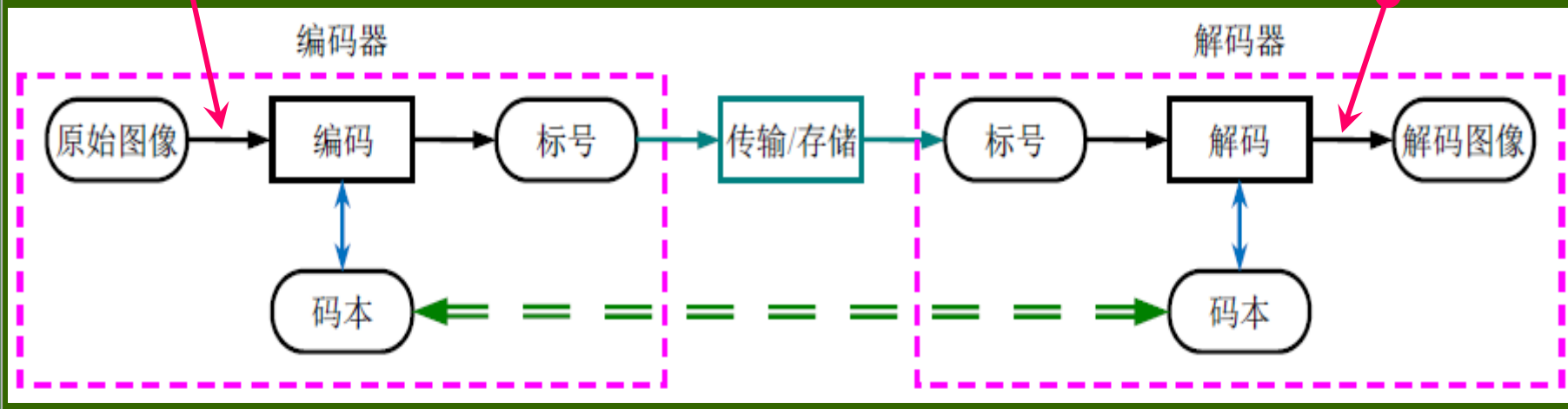
11.4 矢量量化

1. 矢量量化流程

在编码端，先将原始图象划分成小块，用矢量来表示这些小块并对矢量进行量化

构建一个码本，对图象块矢量编码

在解码端，根据矢量码字的标号获得码矢量



11.4 矢量量化

2. 矢量量化原理

(1) 将矢量空间分割为有限个子空间，它们覆盖整个矢量空间且互相不相交。常用沃罗诺伊 (Voronoi) 区域划分

(2) 对每个子空间选择一个代表矢量 (如质心)，称为码矢量，作为量化结果

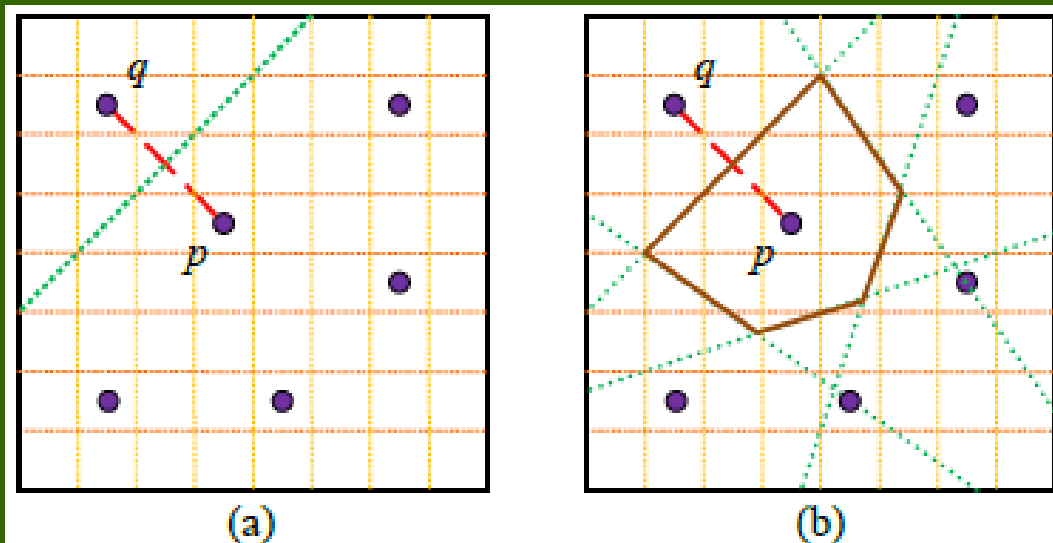
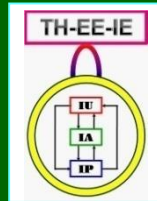


图 11.4.2 Voronoi 多边形



11.4 矢量量化

2. 矢量量化原理

由编码器 C 和解码器 D 两个映射联合构成

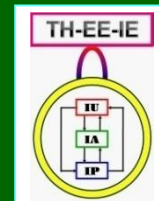
$$C: \mathbf{R}^L \rightarrow I \quad D: I \rightarrow Y$$

$I = \{i \mid i = 1, 2, \dots, N\}$ 是标号集

Y 是码本，包含 N 个码矢量

C 计算输入矢量 \mathbf{x} 与 Y 中各个码矢量 \mathbf{y}_i 间的失真（误差），然后输出一个由映射确定的 \mathbf{y}_i 的标号 i

D 根据接收到的标号 i 从与编码器相同的码本中找到 \mathbf{y}_i ，并用 \mathbf{y}_i 代替输入矢量 \mathbf{x} 作为输出矢量 \mathbf{y}



11.4 矢量量化

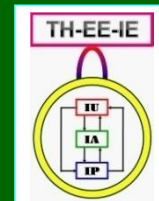
2. 矢量量化原理

码矢量标号 i 被编码成由二进制表示的码字

对定长码，为表示 N 个码矢量标号需要每个码有 $B = \log_2 N$ 个比特。对 L 维矢量，比特率（每像素的比特数）为

$$r = \frac{1}{L} \log_2 N$$

例：将图象划分成 4×4 的块，则 $L=16$ 。如果采用 $2^B=256=N$ 个码字的码本，则 $B=\log_2 256=8$ ，即每个矢量编成 8 比特，每个像素需 $8/16 = 0.5$ 比特



11.4 矢量量化

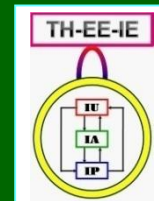
3. 最优码本设计

最优的矢量量化应设计出能将平均失真降为最小的包含 N 个码矢量的码本

(1) 给定需量化的矢量 \mathbf{x} ，最优量化选择的码矢量 \mathbf{y}_i 应能使 \mathbf{x} 和 \mathbf{y}_i 间的失真最小

(2) 最优量化选择的码矢量 \mathbf{y}_i 应能使对应子空间内的平均失真最小，即 \mathbf{y}_i 为子空间的质心

对给定的失真测度，确定码矢量和分割子区间是相关的



11.4 矢量量化

3. 最优码本设计

LBG算法，也称K-means算法

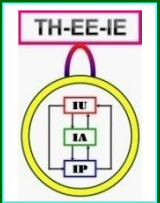
最小化训练集 \mathbf{X} 中的平均失真 T 以获得码矢量

$$T = \frac{1}{M} \sum_{i=1}^N \sum_{\mathbf{x}_m \in R_i} d(\mathbf{x}_m, \mathbf{y}_i)$$

P.257

算法
步骤

- (1) 初始化（选择初始码矢量）
- (2) 将训练矢量就近划到子区间
- (3) 码字更新（计算新质心）
- (4) 中断检查（失真减少率较小）



11.5 准无损编码

压缩率和保真度常是一对矛盾。提高压缩率常使解码图象的失真加大，而要求高保真度又常使压缩率受到限制

准无损编码可看作对无损编码和有损编码的一种折中，期望能在信息损失相对有损编码不太大的情况下能达到比无损编码更高的压缩性能

以 L_∞ 范数来限定准无损编码的压缩率，要使任意一个像素在压缩前后其灰度差的绝对值都不大于某一预先给定的容限值

11.5 准无损编码

1. 准无损压缩算法分类

(1) 基于预测编码的方法

对误差量化

误差容限

$$\hat{e} = \left\lfloor \frac{e + \delta}{2\delta + 1} \right\rfloor (2\delta + 1)$$

(2) 基于可逆变换的方法

变换部分是无损的，但预处理过程及其逆过程中允许出现误差（但不超过设定的容限值）

(3) 有损加准无损的方法

11.5 准无损编码

2. JPEG-LS

基于上下文模型的空域压缩算法

特点：对量化误差为0的像素采用游程编码，游程编码过程由游程检测及游程长度编码两步完成

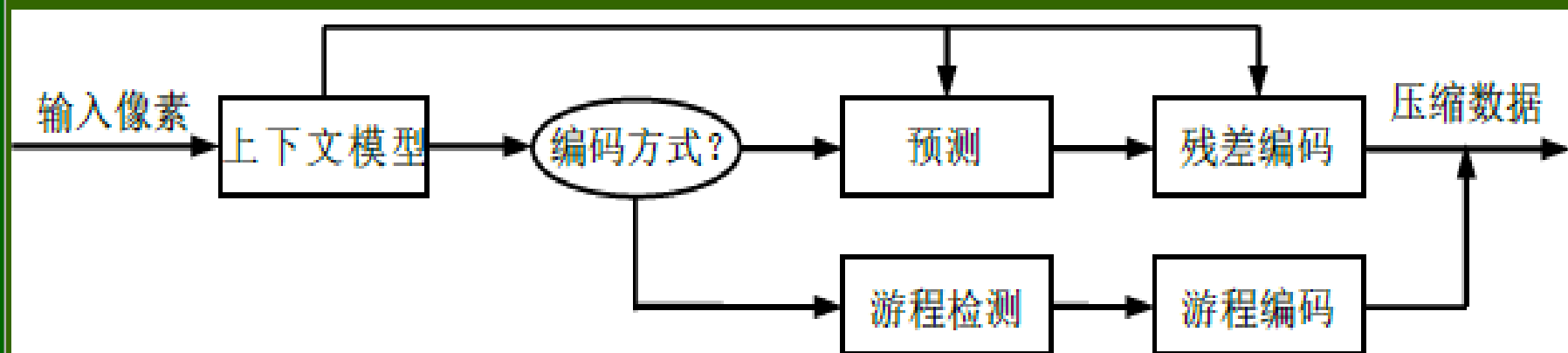


图 11.5.1 JPEG-LS 算法流程框图

11.5 准无损编码

2. JPEG-LS

上下文模型对当前象素进行分类，用以选择编码方式及控制编码各环节

进入游程编码
的上下文条件

当前编码象素的上下文位置关系

$$\begin{aligned} |\dot{x}_{i-1,j+1} - \dot{x}_{i-1,j}| &\leq E_{\max} \\ |\dot{x}_{i-1,j} - \dot{x}_{i-1,j-1}| &\leq E_{\max} \\ |\dot{x}_{i-1,j-1} - \dot{x}_{i,j-1}| &\leq E_{\max} \end{aligned}$$

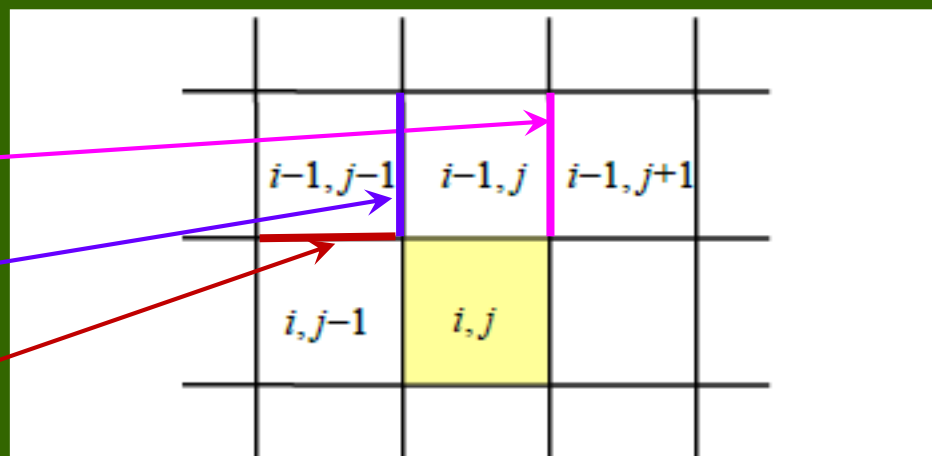


图 11.5.2 JPEG-LS 算法中的上下文位置关系

11.5 准无损编码

3. 准无损CALIC算法

预测上下文：水平方向的 d_h 和垂直方向的 d_v
误差修正上下文 w ，熵编码上下文 s

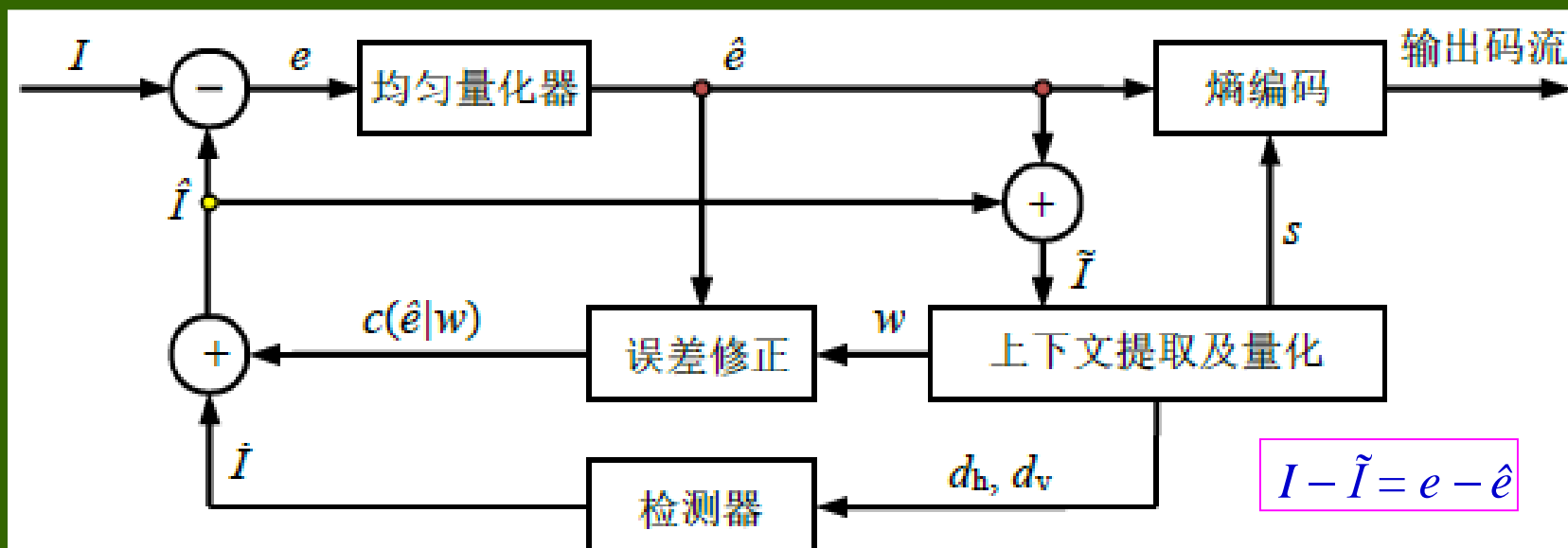
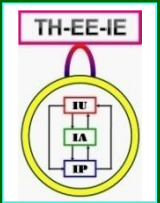


图 11.5.3 CALIC 基本算法流程框图

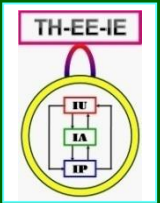


11.6 比较和评述

比较：已介绍的一些方法及它们的特性
评述：尚没有介绍过方法的概况

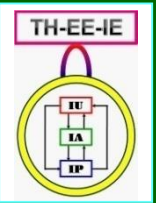
11.6.1 不同方法特性的比较

11.6.2 其他编码方法



11.6.1 不同方法特性的比较

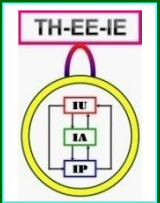
- 变换编码方法可以较好地保持图象的主观质量
- 预测编码方法的特点是用较小的计算代价就可取得较高的压缩率
- 矢量量化方法需要使用比较复杂的编码器
- 哈夫曼编码把固定数目的符号转变成可变长度的码字
- 算术编码把可变数目的符号转变成可变长度的码字
- LZW编码则把可变数目的符号转变成固定长度的码字



11.6.1 不同方法特性的比较

编码	符号	码字
哈夫曼编码	固定数目	可变长度
算术编码	可变数目	可变长度
LZW编码	可变数目	固定长度

		码字	
		固定长度	可变长度
符号	固定数目		哈夫曼编码
	可变数目	LZW编码	算术编码



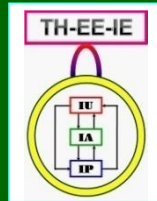
11.6.1 不同方法特性的比较

(1) 解码的即时性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

(2) 解码的唯一性也称单义性，指对任意一个有限长的码符号串，只有一种分解成其各个码符号的方法（只能以一种方式解）

即时码一定是唯一可解码，但唯一可解码不一定是即时码（如用算术编码得到的是唯一可解码但它并不是即时码）

不是唯一可解码肯定也不是即时码，但不是即时码并不能确定该码是否为唯一可解码



11.6.2 其他编码方法

1. 子带编码

将图象分解成一系列带限分量的集合，编码后重新组合起来可无失真重建原始图象。好处：

(1) 不同子带内的图象能量和统计特性不同，所以可以采取不同的变长码甚至不同的编码方法分别进行编码，从而提高编码效率

(2) 通过频率分解，可以减少或消除不同频率之间的相关性，有利于减少图象数据的冗余

(3) 将图象分解为子带后，量化等操作可在各子带内分别进行，避免了互相干扰和噪声扩散

11.6.2 其他编码方法

2. 分形编码



将图象分成子集（分形）并判断其自相似性
分形在不同尺度下保持相似几何形式的能力
可用一组仿射变换来描述。所以，对图象的分形
编码就是要确定一组描述图象的仿射变换，或者
说要建立一个迭代函数系统（IFS），从而得到一
组编码来表示图象

典型的压缩-解压缩不对称方法
分形编码独立于分辨率

11.6.2 其他编码方法

3. 基于内容的编码

将图象根据内容分成不同的区域

更关注图象中的景物概念和高层含义，也更关注观察者的视觉心理感受

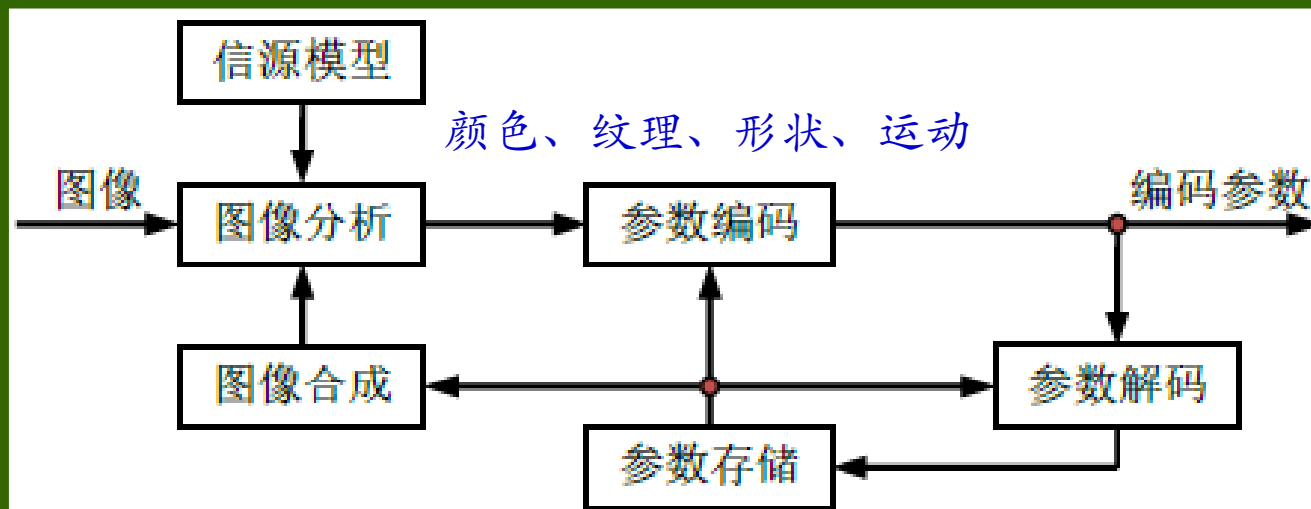
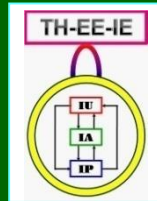


图 11.6.2 分析-合成/综合编码流程框图



11.6.2 其他编码方法

3. 基于内容的编码

模型基编码可分为3个层次：

① 最低层，**物体基编码**或**目标基编码**，要将图象中的目标提取出来，并实时构造未知景物的几何模型

② 中间层，**知识基编码**，利用对景物的先验知识，识别目标类型，构建景物严格的3-D模型

③ 最高层，**语义基编码**，借助复杂的学习推理，获取目标姿态和空间分布以及行为的知识，获取场景的语义

11.6.2 其他编码方法

4. 分层编码

一种编码框架，采用不同的信源模型



图 11.6.3 分层编码系统框图



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/