数字图象处理

第 8 章

图象压缩

本章主要内容

- 图像的编码基本概念
  - 信息论要素，图像熵
  - 基本编码定理（无噪声编码，噪声编码，信源编码）
- 无误差压缩的基本概念
  - 变长编码(重点Huffman编码，算术编码)
  - LZW编码；位平面编码（行程编码)；无损预测编码
- 有损压缩
  - 有损预测编码
  - 变换编码（ WHT， DFT， DCT，小波）

本章基本要求

- 基本要求
  - 基本编码定理
  - 几种Huffman编码等主要变长编码计算与实现；
  - 变换编码原理，重点DCT理论及实现
- 计划学时　　3-4 学时

图像编码引言

- 1.编码的动机
  - 数字图像数据量很大，需要缩减数据量，便于保存和交流
- 2.图像编码
  - 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）

图像编码引言

- 3.编码的可能性
  - 人眼对颜色的空间分辨和对亮度的分辨不一样
    - 彩色图像编码时可以对色度和亮度采用不同表示
  - 人眼对画面静止部分的空间分辨高于活动部分
    - 运动目标采用低分辨率表示
  - 人眼不能察觉亮度的细小变化，即存在视觉阈值，且该阈值与图像内容相关
    - 平坦区阈值低，对量化失真敏感，边缘和纹理区对量化失真不敏感，可以粗糙量化
  - 人眼对图像中心的失真敏感
    - 对图像周边可以粗糙量化节约码字

图像编码引言

- 4.图像压缩方法的分类
  - 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
  - 有损压缩
    - 常能取得较高的压缩率（几十）
    - 但图象经过压缩后并不能经解压缩恢复原状
- 5.本章内容
  - 基本概念-简单编码-预测编码-变换编码

§ 8.1 编码基本概念和理论基础

- 1.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
  - 相对数据冗余
    - n_1和n_2代表表达同一信息，2个数据集合中各自信息载体的个数
    - n_1相对n_2数据冗余的定量描述 $R_D=(n_1-n_2)/n_1= 1-1/C_R$ R_D在$(-\infty,1)$ 中取值

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
$n_1 \gg n_2$	$\to\infty$	$\to 1$	第 1 个数据集合含相当多的冗余数据
$n_1 \ll n_2$	$\to 0$	$\to\infty$	第 2 个数据集合包括比原始表达多得多的数据

§ 8.1 编码基本概念和理论基础

- 2.数据冗余的类别
  - 编码冗余
    - 与灰度分布的概率特性有关
  - 象素相关冗余
    - 空间冗余，时间冗余，几何冗余
  - 心理视觉冗余
    - 与主观感觉有关
  - 减少/消除其中的1种/多种冗余，就能取得数据压缩的效果

§ 8.1 编码基本概念和理论基础

- 2.1.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
  - 图像中每种信息（灰度）出现的概率：
    - $p_s(s_k) = n_k/n$　　$k = 0, 1, \ldots, L-1$
  - 不同灰度出现概率不同
    - 平均比特数

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

  - 减少编码冗余的办法-变长编码方法
    - 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

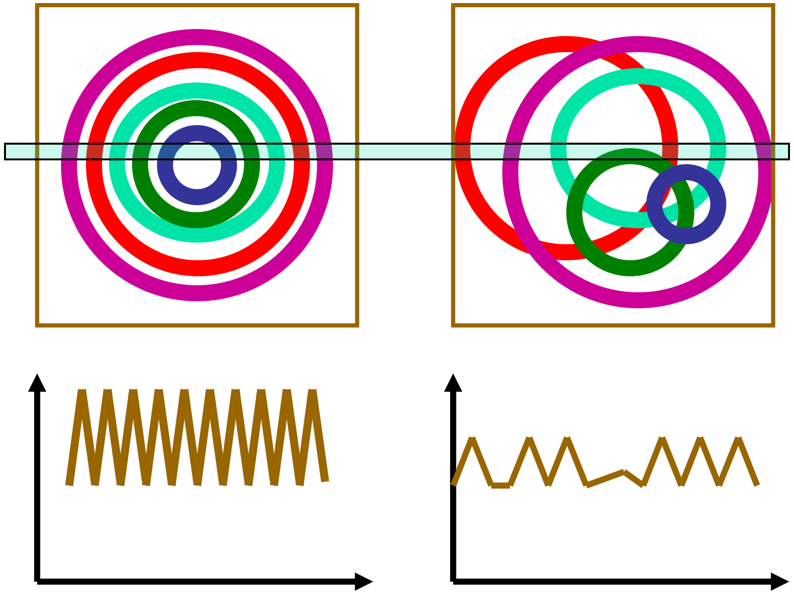
S_k 第k种信息的灰度值

$l(S_k)$ 第k种信息的比特数

§ 8.1 编码基本概念和理论基础

- 2.2.象素间冗余
  - 右边两幅图像，具有相同的目标
  - 直方图差别不大，编码冗余相近
  - 但两图自相关性相差很大
    - 对某一行计算自相关

$$A(n)=\frac{1}{N-n}\sum_{x=0}^{N-1-n} f(x,y)f(x+n,y)$$



  - 图像中有与象素间相关性直接联系的数据冗余，即象素间冗余。
    - 具有象素间冗余，图像中的各像素可以比较方便的用邻近像素的值进行预测，每个独立像素携带的信息量相对较少。
    - 减少像素冗余常采用一些映射变换的方式
  - 连续序列图像之间的帧间冗余与此有相同的物理概念。

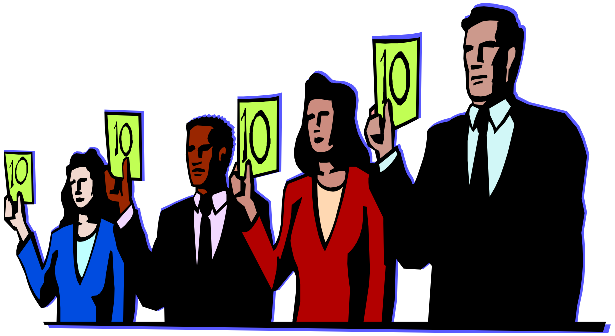
§ 8.1 编码基本概念和理论基础

➢2.3心理视觉冗余

- 人眼并非对所有的视觉信息都同样敏感，编码过程通过去除一些视觉不敏感的(空间、色彩)信息来进行压缩，就是基于心理视觉冗余理论的实践。也就是所谓的“第二代编码技术”。
- 这种编码方式是对信息的损失，只是这样的损失在人的心理上没有造成影响。
- 电视的隔行扫描就是一种基于心理视觉冗余的压缩应用。

§8.1 编码基本概念和理论基础

- 3.图像保真度和质量
  - 保真度的意义
    - 图像压缩有无损和有损两种方式
    - 有损压缩造成的损失如何度量？
      - 主观保真度准则：个人感觉的优、良、中、差。
      - 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来表示损失的信息量， 便于计算或测量



§ 8.1 编码基本概念和理论基础

➢ 客观保真度准则

点误差 $e(x,y)=\hat{f}(x,y)-f(x,y)$

图误差 $\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y)-f(x,y)\right]$

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

均方信噪比

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

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

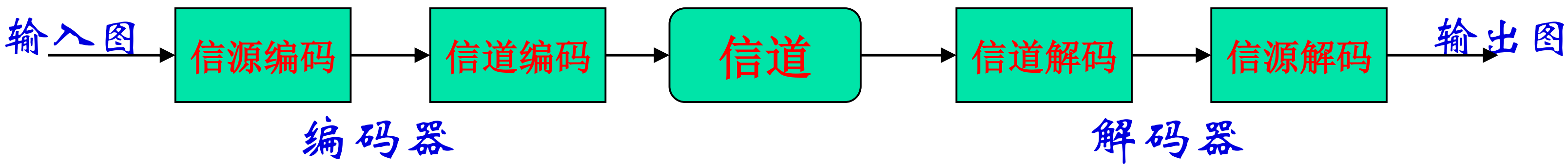
单位：分贝（dB）

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

峰值信噪比 $PSNR=10\lg\left[MNf_{\max}^2\Big/\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y)-f(x,y)\right]^2\right]$

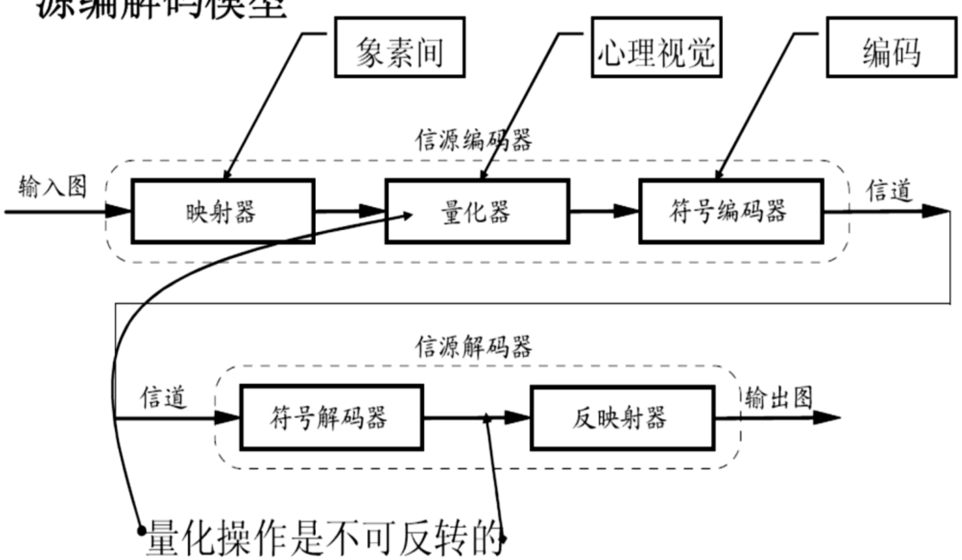
§ 8.1 编码基本概念和理论基础

- 4.图像编码模型



- 4.1信源编码与解码

源编解码模型



- 映射器
  - 数据变换，减少象素间冗余
- 量化器
  - 量化，减少心理冗余
- 符号编码
  - 缩短码字，减少编码冗余
- 编码过程量化是不可逆的有损处理
- 解码过程
  - 不包含无可逆的量化逆过程

§ 8.1 编码基本概念和理论基础

- 4.2信道编码与解码
  - 编码后的信源，冗余很少，抗干扰能力很低
  - 需要进行信道编码，增加可控制的数据冗余，便于在数据传输过程进行差错检测并进行自动纠错
  - 最常用的汉明码信道编解码过程
    - 编码对象：4bit二进制数，$b_3b_2b_1b_0$
    - 编码表示：$h_1h_2h_3h_4h_5h_6h_7$
    - 编码结果：

$$h_4 = b_3, h_5 = b_2, h_6 = b_1, h_7 = b_0$$
$$h_1 = b_3 \oplus b_2 \oplus b_0 = h_4 \oplus h_5 \oplus h_7$$
$$h_2 = b_3 \oplus b_1 \oplus b_0 = h_4 \oplus h_6 \oplus h_7$$
$$h_3 = b_2 \oplus b_1 \oplus b_0 = h_5 \oplus h_6 \oplus h_7$$

误差检测：

$$c_1 = h_3 \oplus h_5 \oplus h_6 \oplus h_7$$
$$c_2 = h_2 \oplus h_4 \oplus h_6 \oplus h_7$$
$$c_3 = h_1 \oplus h_4 \oplus h_5 \oplus h_7$$

§8.1 编码基本概念和理论基础

- 如果检测结果非0，解码器只需翻转码字中由奇偶校验字指出的比特位的位置，取出校正后的二进制码$h_4h_5h_6h_7$
- 校正位与误码位置对照

$C_1C_2C_3$	误码位置	$C_1C_2C_3$	误码位置
001	h_1	101	h_5
010	h_2	110	h_6
100	h_3	111	h_7
011	h_4	000	无错

- 例如：接受序列为0000011，可以计算$c_1c_2c_3$为011，查表可知h_4有错,正确的接受数据$b_3b_2b_1b_0$应该为1011

§ 8.1 编码基本概念和理论基础

5.信息论简介

5.1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

§ 8.1 编码基本概念和理论基础

- 5.2信息系统
  - 信源符号集：$A = \{a_1, a_2, \ldots, a_J\}$
  - 概率矢量：$u = [P(a_1)\ P(a_2)\ \ldots\ P(a_J)]^T$
  - 用(A, u)可以完全描述信源

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

- 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）

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

观察到单个信源符号输出时所获得的平均信息量

§8.1 编码基本概念和理论基础

- 6.无失真编码定理
  - 在无干扰条件下，存在一种无失真的编码方法，使得编码的平均长度L_{avg}与信源的熵$H(u)$任意的接近。

$$L_{avg} \ge H(u)$$

这就是无失真编码定理，也称香农第一定理

- 原始图像数据的冗余度R_D
  - 即原始图像平均码长与原始图像熵之比减1

$$R_D = \frac{L_{avg}}{H(u)} - 1$$

- 编码效率η
  - 冗余度接近0或编码效率接近1为高效码

$$\eta = \frac{H(u)}{L_{avg}} = \frac{1}{1+R_D}$$

- 压缩比C：原始数据平均比特率n与编码后平均比特率n_d之比
  - 无失真编码压缩比上限：$C_{\max} = n / H(u)$

§8.2 简单编码方法

1.变长编码

- 采用自然码会产生编码冗余，解决的办法是用短码表示出现多的灰度级，长码表示出现少的灰度级。
- 举例

s_k	$p_s(s_k)$	自然码	自然码 $l(s_k)$	变长码	变长码 $l(s_k)$
$r_0=0$	0.19	000	3	11	2
$r_1=1/7$	0.25	001	3	01	2
$r_2=2/7$	0.21	010	3	10	2
$r_3=3/7$	0.16	011	3	001	3
$r_4=4/7$	0.08	100	3	0001	4
$r_5=5/7$	0.06	101	3	00001	5
$r_6=6/7$	0.03	110	3	000001	6
$r_7=1$	0.02	111	3	000000	6

- 变长编码后，平均码长$L_{avg}=2.7$,压缩比$C_R=1.11$

§8.2 简单编码方法

- 2.哈夫曼编码
  - 2.1.简介
    - 哈夫曼编码是消除编码冗余的常用技术，对信源符号逐个编码时，它能给出最短码字
    - 基本步骤：
      - 缩减信源符号数量
        - 符号按照概率大小排列
        - 对概率最小两符号合并，计算合并后新符号概率并排序
        - 重复上述步骤，直至剩余符号不多于两个
      - 对各信源符号赋值
        - 赋值是合并的逆向过程
        - 每次将0、1赋予新增加的符号，直到回到最原始符号

§8.2 简单编码方法

➢2.2缩减信源符号数量

- ➢符号按照概率大小排列，逐次结合

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				

§8.2 简单编码方法

- 2.3对信源符号赋值
  - 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								

§8.2 简单编码方法

- 2.4编码与解码
  - 建立符号与码字对照表
  - 编码过程
    - 按对照表将符号变成码字
  - 解码过程
    - 按对照表将码字变成符号
  - 解码举例
    - 根据上面的编码表

      010100111100
    - 对应的符号为

      $a_3a_1a_2a_2a_6$

符号	概率	码字
a_2	0.4	1
a_6	0.3	00
a_1	0.1	011
a_4	0.1	0100
a_3	0.06	01010
a_5	0.04	01011

§8.2 简单编码方法

- 2.5编码评价
  - 编码的平均长度 $L_{avg}=\sum_{k=0}^{L-1}l(s_k)p_s(s_k)=0.4\times1+0.3\times2+3\times0.1\times4=2.2$
  - 信源的熵 $H(\boldsymbol{u})=-\sum_{j=1}^{J}P(a_j)\log P(a_j)=2.14$
  - 编码效率 $\eta=\dfrac{H(u)}{L_{avg}}=\dfrac{2.14}{2.2}=0.973$
  - 计算量：对于*N*个符号
    - 信源消减次数：*N* – 2
    - 码赋值次数：*N* – 2
    - 大量符号情况下，计算量大，亚最优的变长编码方法，通过牺牲编码效率来换取编码速度

§ 8.2 简单编码方法

➢2.6亚最优变长码

块号	信源符号	概率	截断哈夫曼码		平移哈夫曼码		哈夫曼码
第一块	B1	0.25	01	01	01	01	01
	B2	0.21	10	10	10	10	10
	B3	0.19	000	11	000	11	11
	B4	0.16	001	001	001	001	001
第二块	B5	0.08	11 00	000 00	11 01	000 01	0001
	B6	0.06	11 01	000 01	11 10	000 10	00000
	B7	0.03	11 10	000 10	11 000	000 11	000010
	B8	0.02	11 11	000 11	11 001	000 001	000011
熵		2.65					
平均码长			2.73		2.78	2.75	2.7

§ 8.2 简单编码方法

- 2.6亚最优变长码（续）
  - 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，
    - 其他符号用合适的定长编码加前缀进行综合编码
      - 合适长度L与待编码符号N的关系 $N<=2^L$
      - 前缀为其他符号总概率参与前M个符号编码得到的码字
  - 平移哈夫曼码
    - 将符号总数分成相同大小的符号块
    - 所有块中的各个元素采用同样方法编码
    - 每个块加上专门的平移符号进行区分

§ 8.2 简单编码方法

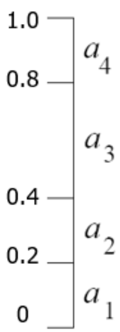
- 2.7哈夫曼编码的特点
  - 非等长码
  - 解码过程能自行确定码字的起止位置，解码唯一
  - 码表中任一一一个短码码字都不是码表中的长码码字的前缀，即前缀码
  - 码表中所有码字都在编码树的叶节点
  - 但是，一旦有误码，对解码的影响是巨大的
  - 哈夫曼编码的抗干扰能力不强

§8.2 简单编码方法

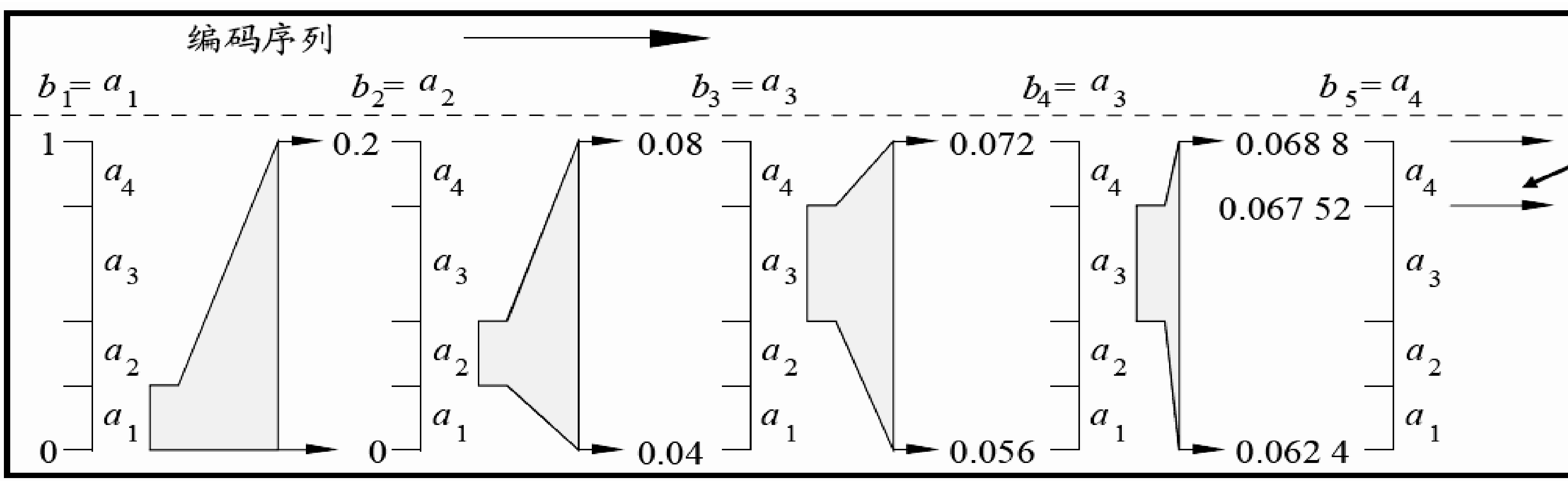
- 3.算术编码
  - 3.1简介
    - 从整个符号序列出发进行编码，不是对单个符号的编码，没有源符号与码字的一一对应关系
    - 需要每个符号的出现概率和整个符号的排列顺序
    - 编码过程按照符号序列顺序递推进行，随符号序列中符号数量增加
      - 用来代表它的区间变小
      - 用来表达区间所需的信息单位的数量变大
    - 算术编码是对符号窜整体的编码，理论上可达到无失真编码定理给出的编码极限

§ 8.2 简单编码方法

3.2编码过程示例



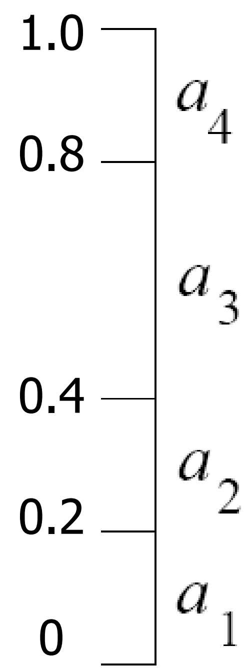
- 编码来自1个4-符号信源{a_1, a_2, a_3, a_4}的由5个符号组成的符号序列：$b_1b_2b_3b_4b_5 = a_1a_2a_3a_3a_4$
- 编码结果(0.06752-0.0688), 取值0.068
- 编码结果还需要有符号长度的描述



§ 8.2 简单编码方法

- 3.3解码过程示例
  - 对于一个序列进行解码
  - 根据码字范围确定第一个字符输出
  - 消除已译码字符影响
    - 码字取值减译码字符下限值
    - 运算结果再除以译码字符宽度值
  - 根据新的码字范围重复前述步骤
  - 根据信源符号长度完成全部译码
  - 举例：0.068,符号长度为5

码字	译码	减下限	变范围
0.068	**a_1**	**0.068**	**0.34**
0.34	**a_2**	**0.14**	**0.7**
0.7	**a_3**	**0.3**	**0.75**
0.75	**a_3**	**0.35**	**0.875**
0.875	**a_4**	**-**	**-**



Code:=(Code-LowRange(x))/Range

§8.2 简单编码方法

- 3.4编码结果分析
  - 只需用到加法和移位运算，所以称为算术编码
  - 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
  - 码字本身确定0和1之间的1个实数区间
  - 从整个符号序列出发采用递推形式连续编码
- 3.5解码的唯一性 举例而言
  - 信源符号集A$=\{a_1,a_2\}=\{0,1\}$
  - 符号产生概率$p(a_1)=0.25,p(a_2)=0.75$
  - 序列*11111100*的二进制算术编码码字为0.1101010_2
  - 编码面对8位序列，需将半开区间[0,1)分成256小区，任意一个可能的序列，只能在一个特定的区间，所以编码具有唯一性

§8.2 简单编码方法

- 4.LZW编码
  - 采用三个发明人姓氏首字母缩写（Lempel-Ziv-Welch）
  - 消除图像中的像素间冗余
  - 是Unix操作系统的文件压缩标准方法
  - 还应用于GIF、TIFF和PDF压缩文件中
- 4.1 特点
  - 码字长度固定
  - 无需信源符号的出现概率
  - 采用字典编码方式，编码中逐渐构成字典
  - 字典定义了符号出现的顺序

§8.2 简单编码方法

- 4.2 编码方式
  - 1.初始字典：字典的前n个位置分配给全部n个可能出现的灰度值
    - 一个8bit的灰度图像，若采用9bit构造字典，初始字典前256位置分配给灰度值0-255

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

  - 2.字典扩充：编码器顺序扫描像素，确定字典中还没有出现的灰度值序列，给定新的字典内容
  - 3.编码输出：每当有新的字典扩充，便输出当前识别序列在字典中的位置序号，作为编码输出

§ 8.2 简单编码方法

0	0	255	255
0	0	255	255
0	0	255	255
0	0	255	255

➢ 4.3编码举例

编号	当前识别序列	将被处理像素	编码输出	字典位置（码字）	字典条目
第1行	——	0	——		
第2行	0	0	0	256	0-0
第3行	0	255	0	257	0-255
第4行	255	255	255	258	255-255
第5行	255	0	255	259	255-0
第6行	0	0	——		
第7行	0-0	255	256	260	0-0-255
第8行	255	255	——		
第9行	255-255	0	258	261	255-255-0
第10行	0	0	——		
第11行	0-0	255	——		
第12行	0-0-255	255	260	262	0-0-255-255
第13行	255	0	——		
第14行	255-0	0	259	263	255-0-0
第15行	0	255	——		
第16行	0-255	255	257	264	0-255-255
第17行	255	——	255		

- 编码输出结果： *0，0，255，255，256，258，260，259，257，255*
- 压缩比： *8*16/9*10=1.42*

§ 8.2 简单编码方法

- 4.4解码过程
  - 按编码方式建立初始字典
  - 顺序解码编码串，根据已有字典对码字进行解码
  - 遇到新码串，扩充字典表

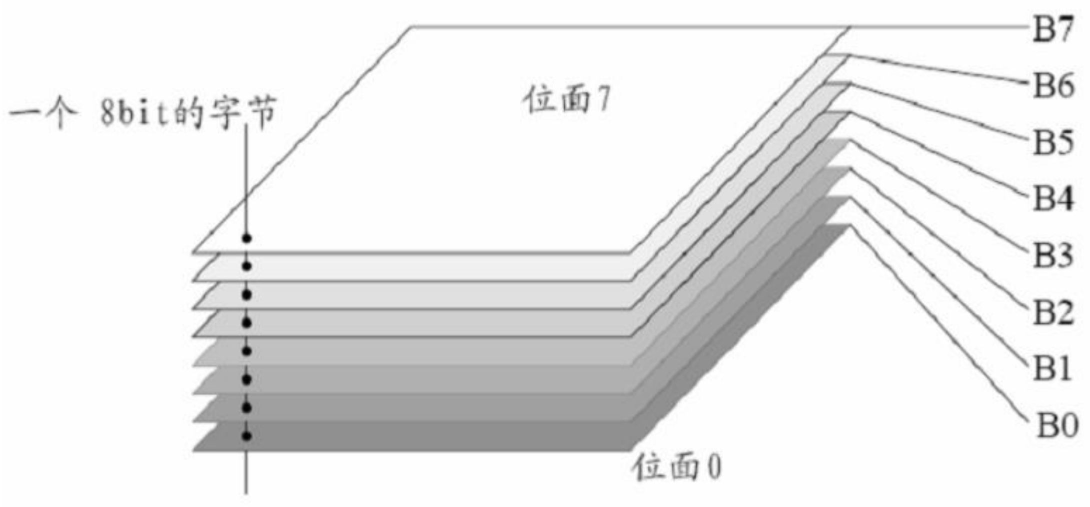
识别序列	编码值	像素（解码输出）	字典位置	字典条目
——	0	0	——	——
0	**0**	0	256	**0-0**
0	255	255	257	0-255
255	255	255	258	**255-255**
255	256	0-0	259	**255-0**
256	258	255-255	260	**0**-0-255
258	**260**	0-0-255	261	**255-255-0**
260	259	255-0	262	0-0-255-255
259	257	0-255	263	255-0-0
257	255	255	264	0-255-255

- 表中第2列为编码串，第3列为解码输出结果

§8.2 简单编码方法

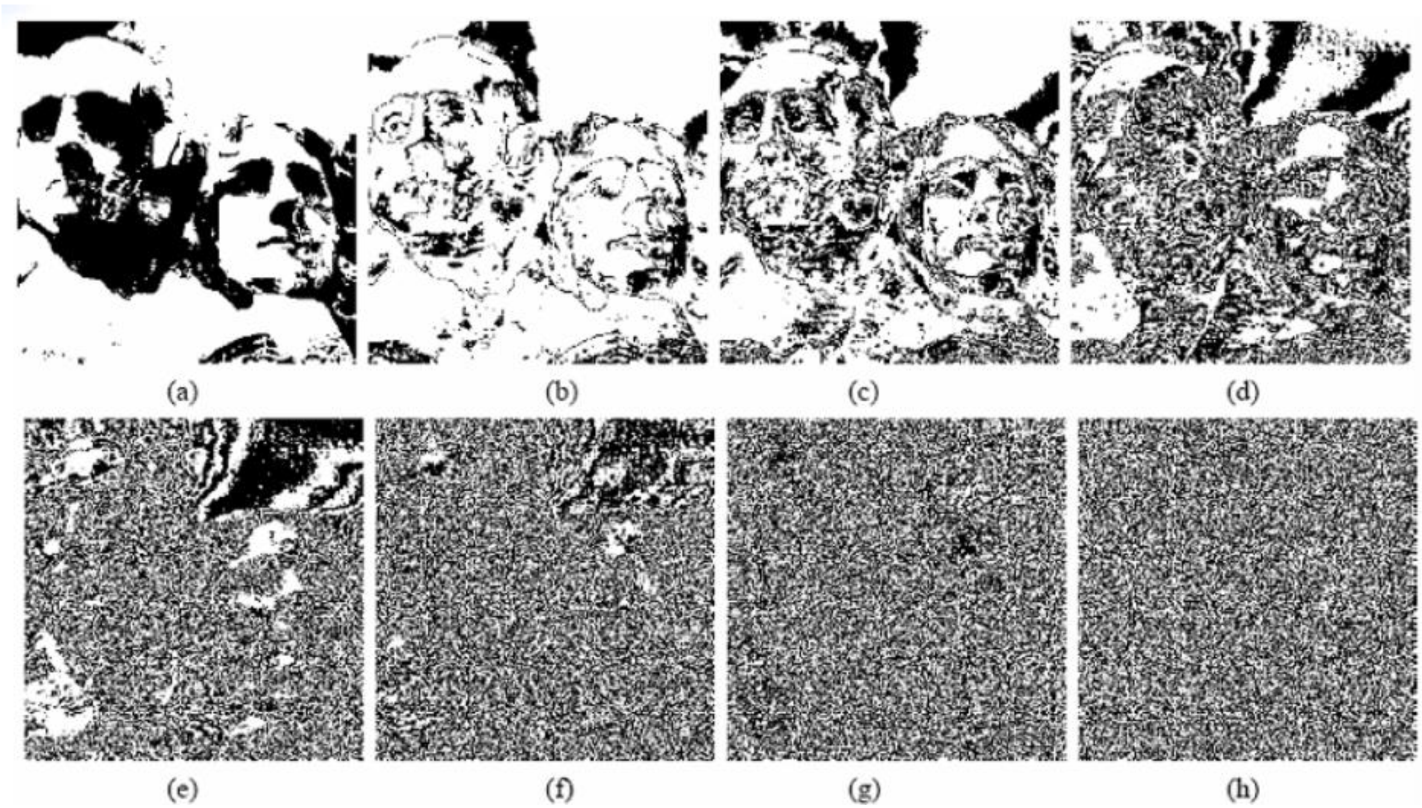
5.1 位平面编码

- 位平面编码将多灰度值的图像分解为一系列二值图像，然后对每一幅二值图像进行二元压缩，它既能够减少编码冗余，还能减少象素间冗余



§8.2 简单编码方法

二值位面图实例（a-h分别为第7-第0位面）



§8.2 简单编码方法

- 位平面分解

定义：位平面二值图像，每一Bit作为一个二值图

问题：像素点灰度的微小变化对位平面的复杂度影响很大

127（011111111）128（100000000）

- 改进为灰度码

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

- 127（11000000）128（01000000）

§8.2 简单编码方法

- 5.2常数块编码(Constant Area Coding)
  - 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2类分别赋予2 bit码字10和11
  - 对于混合块，初始码字为前缀，后接混合模式编码
  - 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

§8.2 简单编码方法

- 5.3 1-D游程编码(Run Length Coding)
  - 设每行均由白色（0）游程开始
  - 对第2位平面（最高位）：
    - 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 1，08

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

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

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

§8.3 预测编码方法

- 1.预测编码思路
  - 图像信息存在象素间冗余
  - 象素间的相关性使得预测成为可能
  - 仅提取每个象素中的新信息并对它们编码可以减少编码长度
  - 根据预测过程是否造成信息丢失，可分为
    - 无损预测编码
    - 有损预测编码

§ 8.3 预测编码方法

- 2.无损预测编码
  - 2.1编解码过程
    - 输入序列： $f_n\,(n = 1, 2, \ldots)$
    - 预测输出： $\hat{f}_n$
    - 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：$f_n = e_n + \hat{f}_n$
      - e_n为解码接收项，$\hat{f}_n$为根据已有结果可计算项
    - 哪里取得了压缩？（消除了象素间冗余）

§ 8.3 预测编码方法

2.2预测方法

- *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： $\hat{f}_n(x,y) = \text{round}\left[\sum_{i=1}^{m} a_i f(x-i,y)\right]$ $round[]$：舍入操作
- 一阶1-D线性预测： $\hat{f}_n(x,y) = \text{round}\left[af(x-1,y)\right]$

x-1,y-1	x-1,y	x-1,y+1
x,y-1	x,y	x, y+1
x+1,y-1	x+1,y	x+1,y+1

2.3系数的确定

- 使得均方误差最小的系数确定方法
  - 但一组系数只是在特定图像意义上面的解，适应性不强
- 一个简化的预测方法

$$f(x,y) = \frac{1}{2}f(x,y-1) + \frac{1}{4}f(x-1,y) + \frac{1}{8}f(x-1,y-1) + \frac{1}{8}f(x-1,y+1)$$

§ 8.3 预测编码方法

- 3.有损预测编码
  - 3.1编解码过程
    - 输入序列： $f_n\ (n = 1, 2, \ldots)$
    - 预测与误差： $e_n = f_n - \hat{f}_n$
    - 误差量化并编码： $e^*_n = q(e_n)$
      - 量化后的误差可以按照级别编码，码长很短
    - 解压序列： $f^*_n = e^*_n + \hat{f}_n$
      - e^*_n为解码接收项，$\hat{f}_n$为根据已有结果可计算项
    - 编码误差： $f_n - f^*_n$
    - 哪里取得了压缩？ （量化，减少了心理视觉冗余）

§ 8.3 预测编码方法

- 3.2有损预测编码举例（Delt调制-DM）
  - 预测器：$\hat{f}_n = a f^*_{n-1}$　量化器：$\dot{e}_n = \begin{cases} +c & 对 e_n>0 \\ -c & 其他 \end{cases}$
  - 预测系数$a \le 1$，常数$c > 0$
  - DM方法得到的码率是1比特/象素
  - 举例
    - 输入序列为：
      - {14，15，14，15，13，15，15，14，20，26，27，28，27，27，29，37，47，62，75，77，78，79，80，81，81，82，83}
    - 编码器参数：
      - a=1，　c=6.5
    - 编解码的初始条件：
      - $f_0 = f^*_0 = 14$
    - 编解码过程分别计算：

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

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

$$\dot{e}_n = \begin{cases} +6.5 & e_n > 0 \\ -6.5 & \text{other} \end{cases}$$

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

§ 8.3 预测编码方法

- DM编码举例

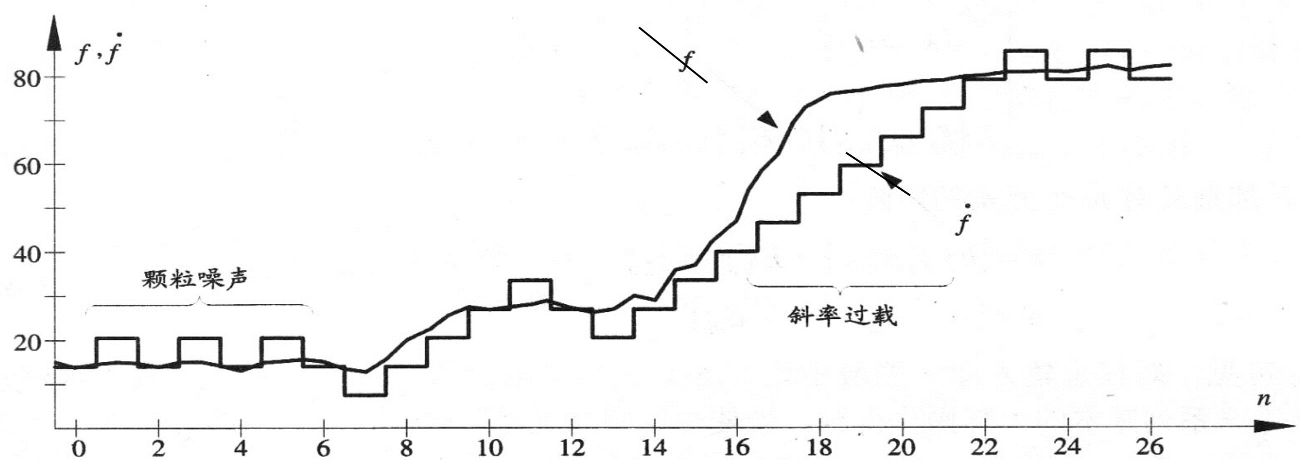
$$\hat{f}_n = \dot{f}_{n-1}$$

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

$$\dot{e}_n = \begin{cases} +6.5 & e_n > 0 \\ -6.5 & \text{other} \end{cases}$$

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

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	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	–5.5
2	14	20.5	–6.5	–6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	–5.5
…	…	…	…	…	…	…	…	…
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.5	53.0	59.5	17.5
…	…	…	…	…	…	…	…	…



§ 8.3 预测编码方法

- 4.最优预测器 $\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f$ $\hat{f}_n = \sum_{i=1}^{m} a_i f_{n-i}$
  - 4.1预测算法：
    - 最优的预测应该使得误差最小 $E\left\{e_n^{\ 2}\right\} = E\left\{\left[f_n - \hat{f}_n\right]^2\right\}$
      - 通过求极小值方式求解
    - 对于线性预测，系数的求解可以更加简单
      - 在误差最小时 $E\left\{e_n^{\ 2}\right\} = E\left\{\left[f_n - \sum_{i=1}^{m} a_i f_{n-i}\right]^2\right\}$
      - 求各个系数a的极小值
      - 可以构造预测器
    - 这种算法称为差值脉冲码调制法（DPCM）

§ 8.3 预测编码方法

- 4.2 DPCM算法
  - 4阶线性预测器

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

x-1,y-1 a_3	x-1,y a_2	x-1,y+1 a_4
x,y-1 a_1	x,y	x, y+1
x+1,y-1	x+1,y	x+1,y+1

  - 系数的意义
    - a1水平相关；a2垂直相关；a3、a4对角相关
  - 几种预测方式
    - 水平相关预测 $\hat{f}(x,y) = f(x,y-1)$
    - 十字相关预测 $\hat{f}(x,y) = 0.5f(x,y-1) + 0.5f(x-1,y)$
    - 半角相关预测 $\hat{f}(x,y) = 0.4f(x,y-1) + 0.4f(x-1,y) + 0.2f(x-1,y-1)$
    - 四邻相关预测

$$f(x,y) = \frac{1}{2}f(x,y-1) + \frac{1}{4}f(x-1,y) + \frac{1}{8}f(x-1,y-1) + \frac{1}{8}f(x-1,y+1)$$

§ 8.3 预测编码方法

- 自适应预测

$$\hat{f}(x,y)=\begin{cases} f(x,y-1) & |f(x-1,y)-f(x-1,y-1)|\le|f(x,y-1)-f(x-1,y-1)| \\ f(x-1,y) & \text{other} \end{cases}$$

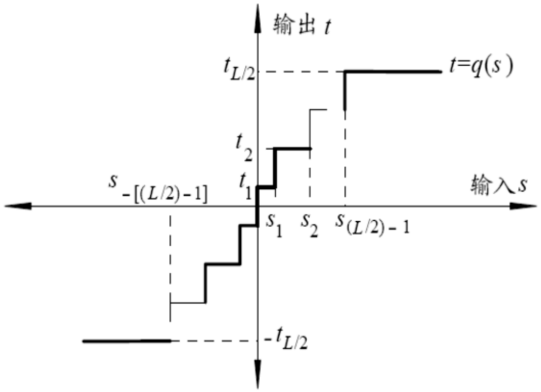
  - 根据目标的局部方向性，调整预测公式，以达到保边缘效果
  - 若水平方向灰度相似性大，有水平边界，宜用水平相关预测
- 关于系数的约束条件
  - 系数之和小于等于1 $\sum_{i=1}^{m} a_i \le 1$

§8.3 预测编码方法

➢ 5.最优量化器

➢ 5.1量化优化概念

- 对数据量化，可以减少编码复杂度，降低码长
- 预测编码，编码对象是预测误差
  - 量化层次少，码字短，编码效率高，误差大
  - 量化层次多，误差小，码字长，编码效率低
- 对预测误差量化，量化器应是奇函数
- 优化量化方式
  - 可结合数据统计特征
  - 可结合视觉特征



§ 8.3 预测编码方法

- 5.2最小均方量化误差
  - s_i、t_i分别为量化器的对应输入和输出
  - $P(s_i)$为s_i出现的概率，总的量化误差为：
  - 令上式等于0，求解优化结果
  - 即L级*(Lloyd_Max)*量化器表达式
  - 在$P(s_i)$符合单位方差的拉普拉斯分布时，量化器如下

$$\int_{s_{i-1}}^{s_i}(s-t_i)p(s)ds$$

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

量化级	1		3		7	
i	s_i	t_i	s_i	t_i	s_i	t_i
1	∞	0.707	1.102	0.395	0.504	0.222
2			∞	1.810	1.181	0.785
3					2.285	1.576
4					∞	2.994

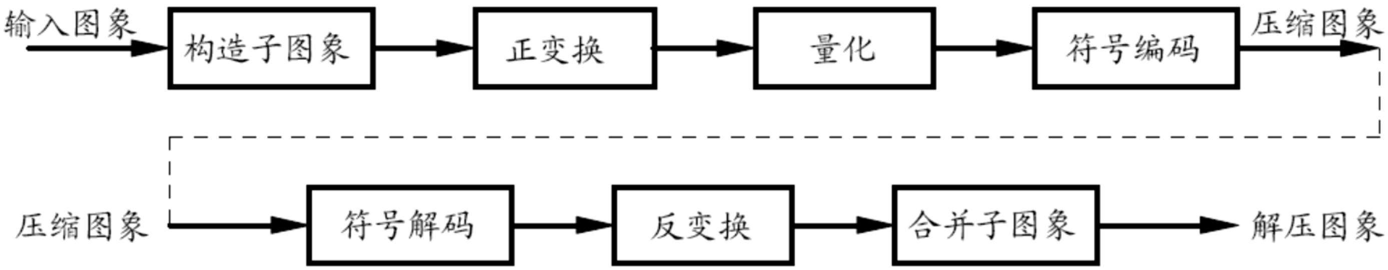
  - 在误差方差不为1，需要用上述数据乘以标准差

§8.4 变换编码方法

- 1.变换编码的概念
  - 变换编码是一种基于空间冗余压缩思想的方法
  - 编码中综合使用其他的信息冗余压缩方法
  - 解决变换编码问题有以下几个问题要解决
    - 变换的选择
    - 基于运算效率和数据相关性考虑的子图选择
    - 变换域中有用信息的选取原则
    - 信息的量化编码

§8.4 变换编码方法

2.变换编码系统



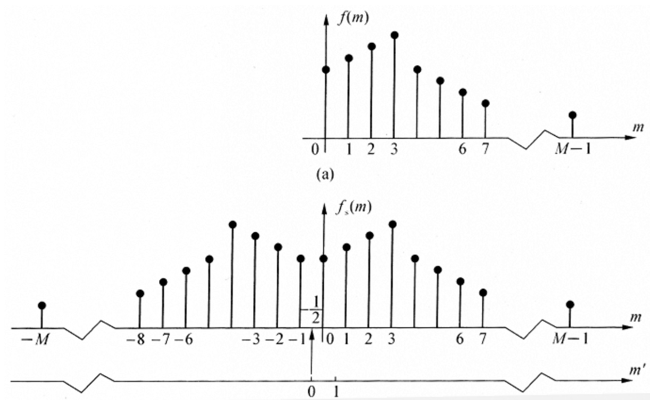
- 构造子图像-图像分解
  - 减少变换的计算复杂度，子图之间尽可能少的图像相关
- 图象变换：
  - 解除每个子图象内部象素之间的相关性，或者说将尽可能多的信息集中到尽可能少的变换系数上
- 量化
  - 压缩不是在变换中而是在量化变换系数时取得的

§8.4 变换编码方法

- 3.变换的选择
  - 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（依赖于图象）
    - 正弦类变换（如DFT和DCT）较优
    - 非正弦类变换（如WHT）实现简单

§ 8.4 变换编码方法-DCT

- 离散余弦变换DCT
  - 1-D离散余弦变换
    - 偶函数傅里叶变换的特点
      - 频域函数是实函数而且只有余弦项
    - 构造1-D离散余弦变换
      - 原序f(m),m=0,1,…M-1, 序列长度M
      - 以m=-1/2为对称拓展序列，序列长度2M
      - 把原点移到m=-1/2,形成偶函数序列$f_s(m)$
      - 新序列的傅里叶变换为：



$$F(S)=\frac{1}{\sqrt{2M}}\sum_{m=-M}^{M-1}f_s(m)\exp[-j\frac{\pi}{2M}(2m+1)s]=\frac{1}{\sqrt{2M}}\left(\sum_{m=-M}^{-1}+\sum_{m=0}^{M-1}\right)f_s(m)\exp[-j\frac{\pi}{2M}(2m+1)s]$$

$$=\frac{1}{\sqrt{2M}}\sum_{m=0}^{M}f(m)\exp[j\frac{\pi}{2M}(2m+1)s]+\frac{1}{\sqrt{2M}}\sum_{m=-M}^{M-1}f(m)\exp[-j\frac{\pi}{2M}(2m+1)s]$$

$$=\frac{2}{\sqrt{2M}}\sum_{m=0}^{M}f(m)\cos[\frac{\pi}{2M}(2m+1)s]$$

§ 8.4 变换编码方法-DCT

- 新序列的傅里叶变换计算上也只涉及到序列0-M
- 这一变换的形式也就是原序列的离散余弦变换
- 变换核为

$$g(s,m)=\sqrt{\frac{2}{M}}\cos[\frac{\pi}{2M}(2m+1)s]$$

- 其矩阵形式

$$g(s,m)=\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\cdots & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{2M} & \cos\frac{\pi}{2M}\cdots & \cos\frac{(2M-1)\pi}{2M} \\ \vdots & \vdots & \vdots \\ \cos\frac{(M-1)\pi}{2M} & \cos\frac{3(M-1)\pi}{2M}\cdots & \cos\frac{(2M-1)(M-1)\pi}{2M} \end{pmatrix}$$

- 反变换

$$f(m)=\sqrt{\frac{2}{M}}\sum_{m=0}^{M-1}c(s)F(s)\cos[\frac{\pi}{2M}(2m+1)s]$$

$$c(s)=\begin{cases} \frac{1}{\sqrt{2}}, & s=0 \\ 1, & s=1,2,\ldots,M-1 \end{cases}$$

§ 8.4 变换编码方法-DCT

- 2-D离散余弦变换
  - 根据1-D余弦变换的思路得到2-D变换表达式

$$F(u,v)=\frac{2}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)\cos[\frac{\pi}{2M}(2m+1)u]\cos[\frac{\pi}{2N}(2n+1)v]$$

  - 变换核为：

$$g(m,n,u,v)=\frac{2}{\sqrt{MN}}\cos[\frac{\pi}{2M}(2m+1)u]\cos[\frac{\pi}{2N}(2n+1)v]$$

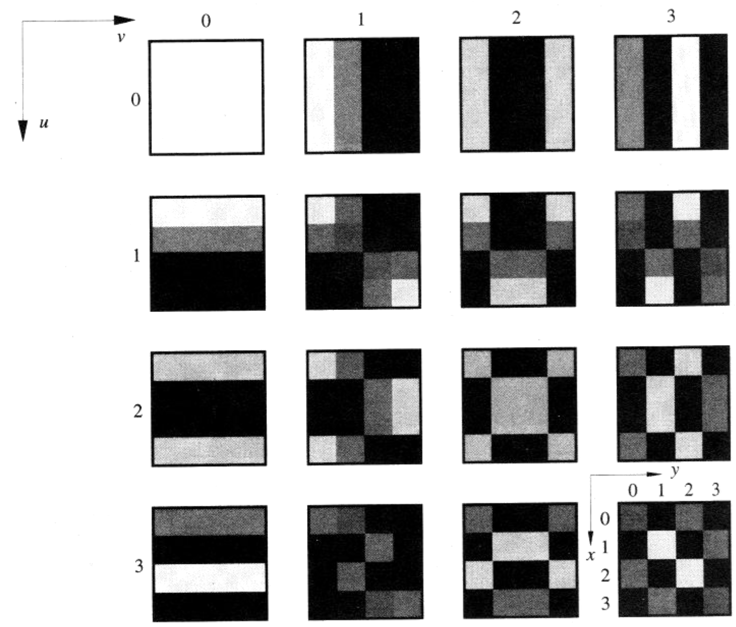
  - 反变换为：

$$f(m.n)=\frac{2}{\sqrt{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}c(u)c(v)F(u,v)\cos[\frac{\pi}{2M}(2m+1)u]\cos[\frac{\pi}{2N}(2n+1)v]$$

  - 可以看到变换是可分离的，也是对称的
  - 根据DCT的构建原理，其变换可以由FFT来实现

§8.4 变换编码方法-DCT

- N=4基函数
  - 共有4X4=16个基函数
  - 对应（u，v）不同取值使用
  - 每一个基函数与原图像运算
  - 得到对应的F（u，v）

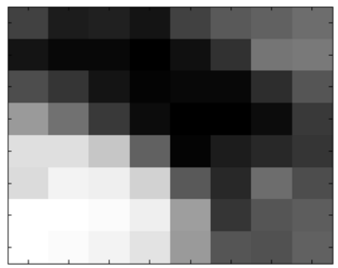


§ 8.4 变换编码方法-DCT

➢ 离散余弦变换的应用

- Gray-Scale Example:
- Value Range 0 (black) --- 255 (white)

63	33	36	28	63	81	86	98
27	18	17	11	22	48	104	108
72	52	28	15	17	16	47	77
132	100	56	19	10	9	21	55
187	186	166	88	13	34	43	51
184	203	199	177	82	44	97	73
211	214	208	198	134	52	78	83
211	210	203	191	133	79	74	86



- 2D-DCT of matrix

Numbers are coefficients of polynomial

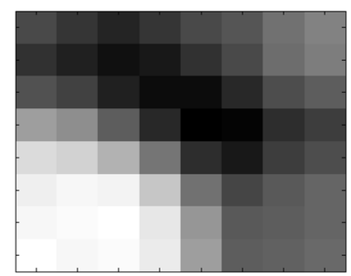
-304	210	104	-69	10	20	-12	7
-327	-260	67	70	-10	-15	21	8
93	-84	-66	16	24	-2	-5	9
89	33	-19	-20	-26	21	-3	0
-9	42	18	27	-7	-17	29	-7
-5	15	-10	17	32	-15	-4	7
10	3	-12	-1	2	3	-2	-3
12	30	0	-3	-3	-6	12	-1

- Cut the least significant components

-304	210	104	-69	10	20	-12	0
-327	-260	67	70	-10	-15	0	0
93	-84	-66	16	24	0	0	0
89	33	-19	-20	0	0	0	0
-9	42	18	0	0	0	0	0
-5	15	0	0	0	0	0	0
10	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- New Matrix and Compressed Image

55	41	27	39	56	69	92	106
35	22	7	16	35	59	88	101
65	49	21	5	6	28	62	73
130	114	75	28	-7	-1	33	46
180	175	148	95	33	16	45	59
200	206	203	165	92	55	71	82
205	207	214	193	121	70	75	83
214	205	209	196	129	75	78	85



§8.4 变换编码方法

- 4.子图象尺寸选择
  - 选择子图象进行变换编码的原因
    - 能减少变换编码误差和计算复杂度
      - 图像跨子图的相关性不多，相关性的集中利于减少误差
      - 压缩量和计算复杂度都随子图象尺寸的增加而增加
  - 选取子图象的条件
    - 相邻子图象之间的相关（冗余）减少到某个可接受的水平；
    - 子图象的长和宽都是2的整数次幂
      - 最常用的子图象尺寸：8 × 8和16 × 16

§8.4 变换编码方法

- 5.编码信息的选择-比特分配
  - 内容：
    - 对变换子图象的系数截断、量化和编码的全过程
  - 与截断误差有关的问题
    - 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（量化）
  - 保留系数的2个准则
    - 最大方差准则，依此准则构成分区编码
    - 最大幅度准则，依此准则构成阈值编码

§ 8.4 变换编码方法

- 6.分区编码
  - 具有最大方差的变换系数带有最多的图象信息
  - 本算法与数据无关，可采用事先确定模板，保留指定的系数

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

  - 数据的量化和编码：采用一致的量化或对每个系数量化方式不同

§8.4 变换编码方法

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

- 重新编排后的数据会有大量的连续0序列，可以采用游程编码进行压缩

§8.4 变换编码方法

- 7.1.阈值的选取
  - 随子图象不同而保留不同位置的变换系数，常用3种对变换子图象取阈值的方法：
    - 对所有子图象用1个全局阈值压缩
      - **压缩程度因图而异，取决于超过全图阈值的系数的数量**
    - 对各个子图象分别用不同的阈值
      - **可以通过要求每个子图舍去的系数数量相同，来动态规划阈值**
    - 根据子图象中系数的位置选取阈值
      - **可以将阈值的选取与量化结合起来**

本章小结

- 编码基本概念和理论基础
  - 数据冗余-编码冗余、像素冗余、心理冗余
  - 图象编码的模型-信源编码、信道编码
  - 信息编码的基本概念-平均码长的极限
- 简单编码方法
  - 哈夫曼编码、算术编码、LZW编码
  - 位平面编码-游程编码、边界跟踪编码
- 预测编码方法
  - 无损预测编码-预测器选择
  - 有损预测编码-量化器选择
- 变换编码方法
  - 变换方式选择、子图尺寸
  - 比特分配-分区编码、阈值编码