



武汉大学
Wuhan University

第八章 图像压缩(1)

涂卫平

武汉大学计算机学院

2018年秋季学期



主要内容

Main Content



数字图像压缩基础

统计编码

预测编码



引言

一部2小时的标清电影的数据量

每帧图像的数据量: 720×480 (像素) $\times 3$ (字节/像素) = 1,036,800 (字节)

帧率: 30帧/秒

每秒的数据量: 1,036,800 (字节/帧) $\times 30$ (帧) = 31,104,000 (字节)

2小时的数据量: 31,104,000 (字节/秒) $\times 60 \times 60 \times 2$ (秒) $\approx 2.24 \times 10^{11}$ (字节) $\approx 224\text{GB}$

存储: 27张8.5GB的双面DVD

实时传输: 需要的带宽约为31Mbps

数字图像的特点: 数据量大, 为其存储、传输带来困难。

图像压缩编码的必要性

现代通信中，图像传输已成为重要内容之一。采用编码压缩技术，减少传输数据量，是提高通信速度的重要手段。**没有图像编码与压缩技术的发展，大容量图像信息的存储与传输难以实现，多媒体、信息高速公路等新技术的实际应用会遇到很大困难。**





基础知识

数据压缩(Data Compression)

数据压缩是减少表示给定**信息量**所需**数据**比特数的技术，其目的是减少存储数据所需的空间和传输所用的时间。

数据是信息的载体。相同数量的信息可以用不同数量的数据表示。

- ◆ 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。（ $20*2+2=42$ 个半角字符）
- ◆ Helen将于明晚6点在虹桥接你。（ $10*2+6=26$ 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。

这种可以减少而不影响接收端正确理解信息的数据，就称为冗余。



基础知识

数据冗余的定义：

设 n_1 和 n_2 是原始图像和压缩后图像每个像素的**平均比特数**

■ 压缩率（压缩比）——用于描述图像压缩效果

- $CR = n_1 / n_2$

■ 相对数据冗余：

- $RD = 1 - 1/CR = (n_1 - n_2) / n_1$

数据冗余的分类：

- 编码冗余
- 空间和时间冗余
- 不相关的信息

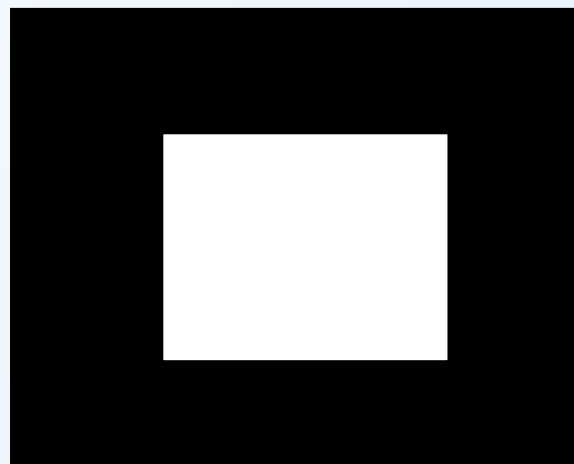


基础知识

编码冗余

- 编码（Code）：用于表示信息实体或者时间集合的符号系统，包含多个符号序列，每个符号序列叫一个“码字”（Code Word）。每个码字中的符号数量叫做码字的长度。
- 如果一个图像的灰度级编码，使用了多于实际需要的编码长度，就称该图像包含了编码冗余

例：如果用8位表示右边图像像素的灰度级，该图像就存在着编码冗余，因为该图像的像素只有两个灰度，用一位即可表示。





基础知识

编码冗余

设一幅 $M \times N$ 图像的灰度值取值区间为 $[0, L-1]$ ，每个灰度值用 r_k 表示，

即 $r_k \in [0, L-1], k = 0, 1, \dots, L-1$ 。

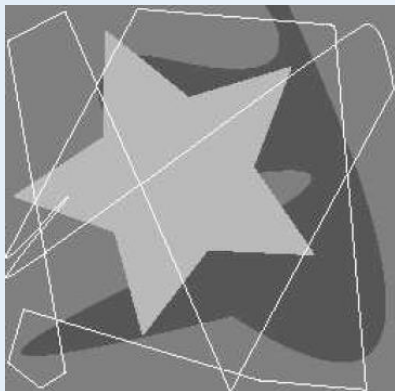
设 r_k 出现的概率为 $P_r(r_k) = \frac{\eta_k}{MN}$ ， $k = 0, 1, \dots, L-1$ ，其中 η_k 为灰度级 r_k 在图像中出现的次数。

设用于表示灰度级 r_k 所需的比特数为 $l(r_k)$ ，

则每像素所需平均比特为：

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

编码冗余



一幅有4个灰度级的图像

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Code1:

以普通二进制编码来表示灰度级，则编码每个灰度级的平均比特数为**8**

Code2:

按照“**出现概率较大的灰度级用较短的码字编码**”的原则来表示灰度级，则编码每个灰度级的平均比特数为**1.81**

$$L_{\text{avg}} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3) = 1.81 \text{ bits}$$

显然，Code1中存在大量编码冗余



基础知识

空间和时间冗余

空间冗余

一个像素通常与它周围的某些像素在**亮度**和**色度**上存在一定的关系。

时间冗余

连续播放的视频，每一帧就是一幅图像，相邻帧之间的时间间隔很小；同时实际生活中的运动物体速度有限，使得视频序列图像之间有很强的**相关性**。

空间冗余



256种灰度级条纹图

图中256种灰度级是等概率的（如灰度直方图所示）

每条横线的灰度是随机选择的，故像素在垂直方向上是不相关的

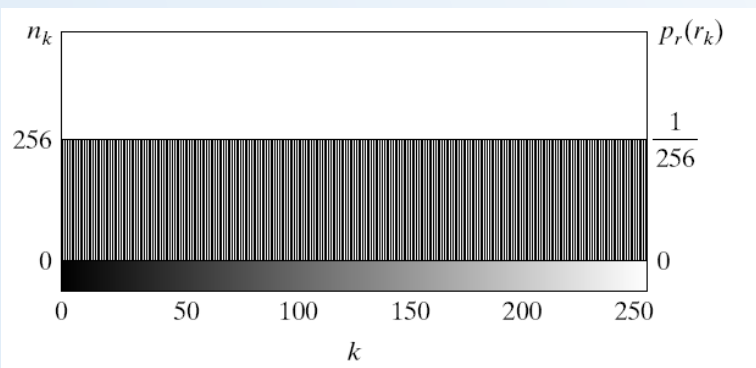
每条横线的灰度相同，故像素在水平方向上相关度最大（完全依赖）

对于图中的256条横线，可以用以下方式表示其灰度：

记录每条横线的灰度级和横线的长度（像素个数）
（行程编码的思想）

需要的比特数为： $(8 + 8) \times 256$

原始灰度图的数据量为： $256 \times 256 \times 8$



灰度级直方图



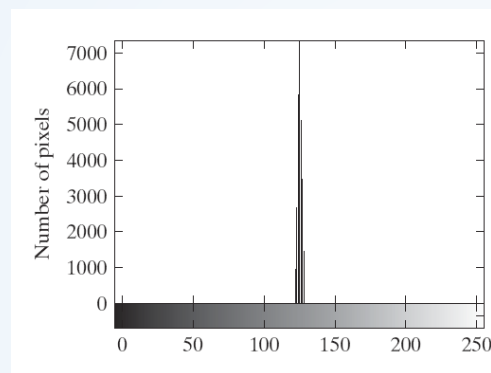
基础知识

不相关的信息

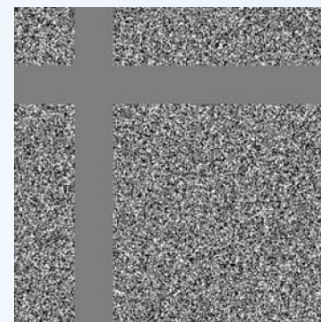
图像中包含的被人视觉系统忽略或与图像的用途无关的信息。



原始图像



灰度直方图



经直方图均衡处理后的图像

- 由直方图可知：图像中存在不同灰度级（125~131）的像素
- 人类**视觉分辨力的局限**，使得我们忽略了灰度的较小变化，而只能感知到灰度平均值
- 经直方图均衡化处理后的图像显示了两个恒定灰度区域（水平的和垂直的），以及它们周围的随机灰度变化区域（真实信息）



基础知识

不相关的信息——心理视觉冗余



175KB



1.46MB

受生理和心理上的影响，人眼对黑白和彩色信息的分辨率是有限的。光学上不一致的图像在视觉上可能是一样的。



基础知识

图像保真度准则

对图像的压缩可能导致信息的丢失，需要对信息的丢失进行衡量。

客观保真度标准：

信息的损失能够表示为原始图像与压缩并解压缩后图像的函数。

- 像素值的误差：
$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

- 两个图像之间的总误差：
$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (\hat{f}(x, y) - f(x, y))$$

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



基础知识

图像保真度准则

主观保真度标准：

- 解压缩后的图像最终还是要由人来观察的，故使用**人的主观评价**来衡量图像的质量更为适当
- 主观评价需要组织多人对解压图像进行评价打分，再取所有分值的**平均值**
- 主观评价分为“**绝对等级评分**”和“**比较评分**”两种



基础知识

图像保真度准则

主观保真度标准：绝对等级评分

表 8.2 电视配置研究组织的等级尺度 (Freundendall and Behrend)

值	等 级	描 述
1	优秀	一幅极高质量的图像，和您希望的一样好
2	好	一幅高质量的图像，提供赏心悦目的观看效果。干扰不讨厌
3	可通过的	一幅质量可接受的图像。干扰不讨厌
4	边缘的	一幅质量较差的图像，您希望能改进它，干扰有些讨厌
5	差	一幅质量非常差的图像，但还可以观看，明显存在讨厌的干扰
6	不能用	一幅坏到不能观看的图像

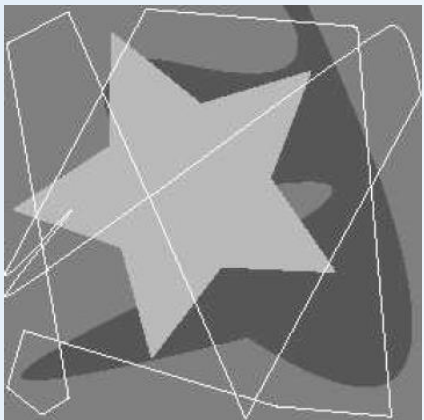
比较评价：

通过对原始图像和解压缩后图像的比较，用 $\{-3, -2, -1, 0, 1, 2, 3\}$ 这样的一个等级序列表示主观评估 {非常恶劣、恶劣、稍坏、普通、稍好、较好、非常好}

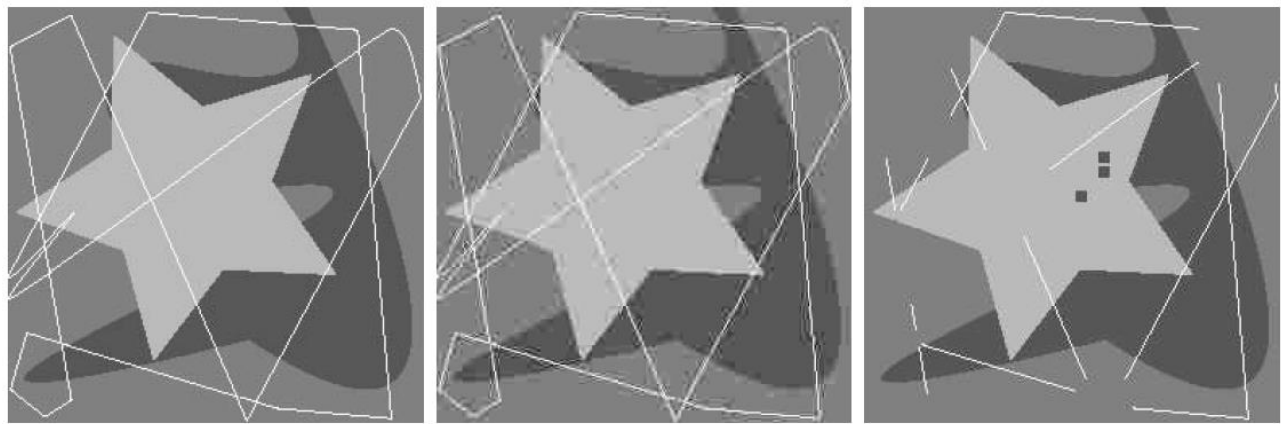


基础知识

图像保真度准则



原始图像



a b c

解压缩图像a, b和人工合成图像c

- ◆ (a)具有最低的均方根误差，(b)的均方根误差高一些，且在物体边界处存在明显的退化。对(a)和(b)的主、客观评价是一致的。
- ◆ (c)图像中失去了几条重要线条，且多了几个暗色小方块；但是其均方根误差却小于(b)。可见：客观测度与主观测度并非完全一致。



图像压缩模型



若 $f(x, y) = \hat{f}(x, y)$ ，则称**无损压缩**，否则称为**有损压缩**。



基础知识

图像压缩模型-压缩（编码）过程

◆ 映射

把图像由普通的基于像素灰度矩阵的表达形式**转换**成其它的**表达形式**，以**降低空间和时间冗余**。这一过程通常是**可逆**的，且可能会也可能不会直接减少表示图像所需的数据量。

◆ 量化

根据预设的保真度准则**降低映射器输出的精度**，目的是**排除**压缩表示的**无关信息**。这一操作**不可逆**，在无损压缩中要省去此步骤。

◆ 符号编码

为量化器的输出值设计码字。大多数情况下使用变长码字，**最短的码字**赋予出现**概率最高的量化器输出值**。



基础知识

图像压缩模型-解压缩（解码）过程

以相反的顺序执行编码器模块的逆操作。

◆ 符号解码

把码字转换成量化器的输出值。

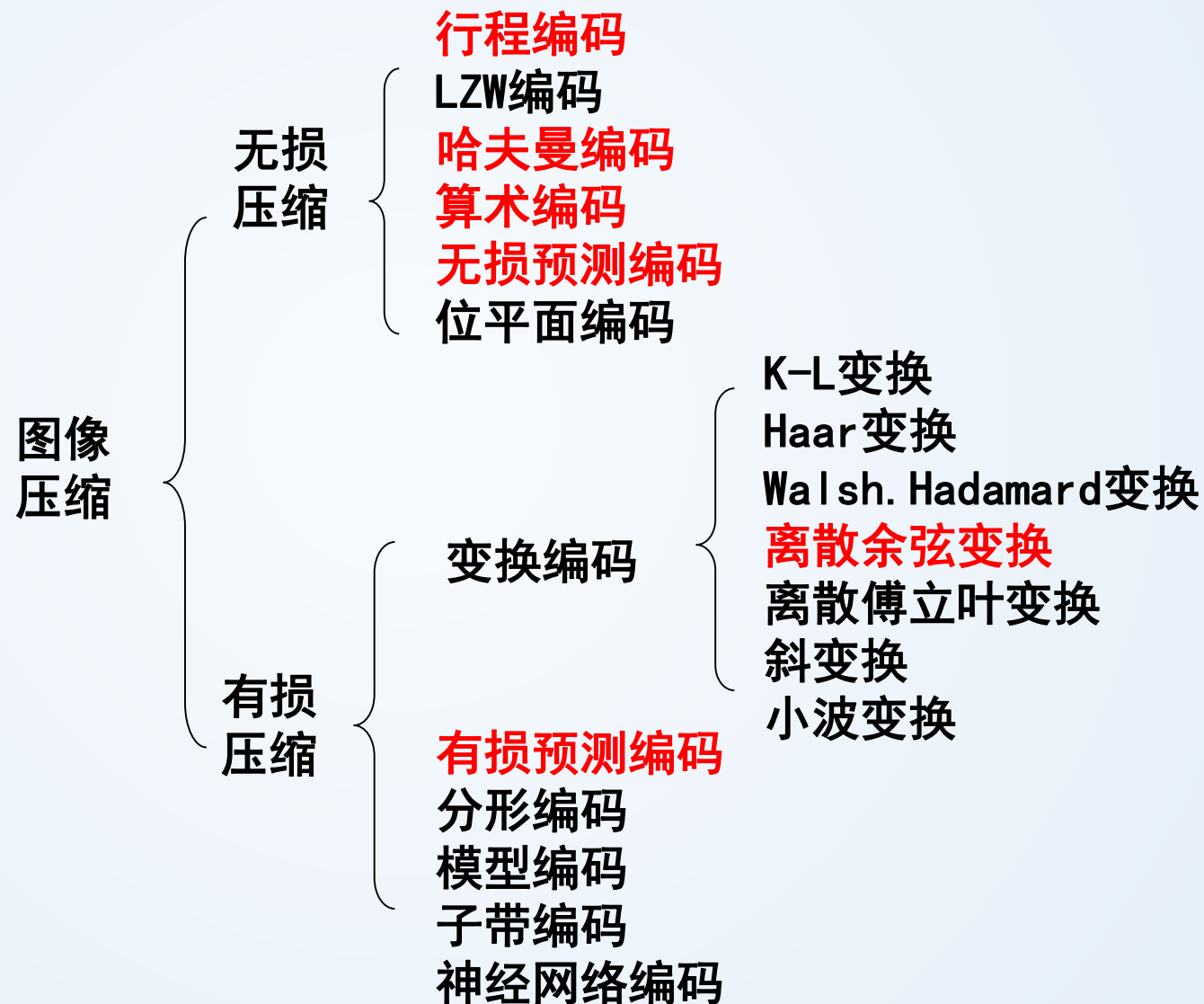
◆ 逆映射

映射操作的反操作，以得到像素矩阵表达形式的重建图像。



基础知识

图像压缩方法分类



主要内容

Main Content



数字图像压缩基础

统计编码

预测编码



统计编码

统计编码是指建立在图像**统计特性**基础之上的一类压缩编码方法，根据信源的**概率分布**特性，分配**不同长度**的码字，降低平均码字长度，以提高传输速度，节省存储空间。

等长编码与变长编码

对于一个无记忆离散信源中每一个符号，若采用**相同长度**的不同码字代表相应符号，就叫做**等长编码**；若对信源中的不同符号用**不同长度**的码字表示就叫做**不等长或变长编码**。

主要的统计编码方法：

- ◆ Huffman编码
- ◆ 行程编码
- ◆ 算术编码



统计编码

Huffman编码

- 思想：通过减少编码冗余来达到压缩的目的。
- 方法：统计符号的出现概率，建立一个概率统计表
 - 将最常出现(概率大)的符号用最短的码字编码
 - 最少出现的符号用最长的码字编码



统计编码

Huffman编码

Huffman码表的建立步骤:

1. 将信源符号按出现概率从大到小排成一行，然后把最末两个符号的概率相加，合成一个概率。
2. 把这个符号的概率与其余符号的概率按从大到小排列，然后再把最末两个符号的概率加起来，合成一个概率。
3. 重复上述做法，直到最后剩下两个概率为止。
4. 从最后一步剩下的两个概率开始逐步向前进行编码。每步只需对两个分支各赋予一个二进制码，如对概率大的赋予码0，对概率小的赋予码1。



统计编码

Huffman编码

输入	输入概率
S_2	0.4
S_6	0.3
S_1	0.1
S_4	0.1
S_3	0.06
S_5	0.04



统计编码

Huffman编码

输入	输入概率	第一步
S_2	0.4	0.4
S_6	0.3	0.3
S_1	0.1	0.1
S_4	0.1	0.1
S_3	0.06	0.1
S_5	0.04	



统计编码

Huffman编码

输入	输入概率	第一步	第二步
S_2	0.4	0.4	0.4
S_6	0.3	0.3	0.3
S_1	0.1	0.1	0.2
S_4	0.1	0.1	0.1
S_3	0.06	0.1	
S_5	0.04		



统计编码

Huffman编码

输入	输入概率	第一步	第二步	第三步
S_2	0.4	0.4	0.4	0.4
S_6	0.3	0.3	0.3	0.3
S_1	0.1	0.1	0.2	0.3
S_4	0.1	0.1	0.1	
S_3	0.06	0.1		
S_5	0.04			



统计编码

Huffman编码

输入	输入概率	第一步	第二步	第三步	第四步
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	
S_4	0.1	0.1	0.1		
S_3	0.06	0.1			
S_5	0.04				



统计编码

Huffman编码

输入	输入概率	第一步	第二步	第三步	第四步
S_2	0.4	0.4	0.4	0.4	0.6 0
S_6	0.3	0.3	0.3	0.3	0.4 1
S_1	0.1	0.1	0.2	0.3	
S_4	0.1	0.1	0.1		
S_3	0.06	0.1			
S_5	0.04				

The diagram illustrates the Huffman coding process for six symbols: S_2 , S_6 , S_1 , S_4 , S_3 , and S_5 . The process starts with the initial probabilities and proceeds through four steps of merging the two smallest probabilities at each stage. Red arrows and labels (0, 1) indicate the merging process and the resulting probabilities at each step.

- Initial Probabilities:** S_2 (0.4), S_6 (0.3), S_1 (0.1), S_4 (0.1), S_3 (0.06), S_5 (0.04).
- Step 1:** Merge S_3 (0.06) and S_5 (0.04) into a node with probability 0.1. The left branch is labeled 0 and the right branch is labeled 1.
- Step 2:** Merge S_4 (0.1) and the node from Step 1 (0.1) into a node with probability 0.2. The left branch is labeled 0 and the right branch is labeled 1.
- Step 3:** Merge S_1 (0.1) and the node from Step 2 (0.2) into a node with probability 0.3. The left branch is labeled 0 and the right branch is labeled 1.
- Step 4:** Merge S_6 (0.3) and the node from Step 3 (0.3) into a node with probability 0.6. The left branch is labeled 0 and the right branch is labeled 1.
- Final Step:** Merge S_2 (0.4) and the node from Step 4 (0.6) into the root node with probability 1.0. The left branch is labeled 0 and the right branch is labeled 1.



统计编码

Huffman编码

输入	输入概率	第一步	第二步	第三步	第四步
S_2	0.4	0.4	0.4	0.4	0.6 0
S_6	0.3	0.3	0.3	0.3	0.4 1
S_1	0.1	0.1	0.2	0.3	
S_4	0.1	0.1	0.1		
S_3	0.06	0.1			
S_5	0.04				

$S_2=1$



统计编码

Huffman编码

输入	输入概率	第一步	第二步	第三步	第四步
S_2	0.4	0.4	0.4	0.4	0.6 0
S_6	0.3	0.3	0.3	0.3	0.4 1
S_1	0.1	0.1	0.2	0.3	
S_4	0.1	0.1	0.1		
S_3	0.06	0.1			
S_5	0.04				

$S_6=00$



统计编码

Huffman编码

输入	输入概率	第一步	第二步	第三步	第四步
S_2	0.4	0.4	0.4	0.4	0.6 0
S_6	0.3	0.3	0.3	0.3	0.4 1
S_1	0.1	0.1	0.2	0.3	
S_4	0.1	0.1	0.1		
S_3	0.06	0.1			
S_5	0.04				

$S_1=011$



统计编码

Huffman编码

输入 输入概率 第一步 第二步 第三步 第四步

S_2	0.4	0.4	0.4	0.4	0.6	0
S_6	0.3	0.3	0.3	0.3	0.4	1
S_1	0.1	0.1	0.2	0.3		
S_4	0.1	0.1	0.1			
S_3	0.06	0.1				
S_5	0.04					

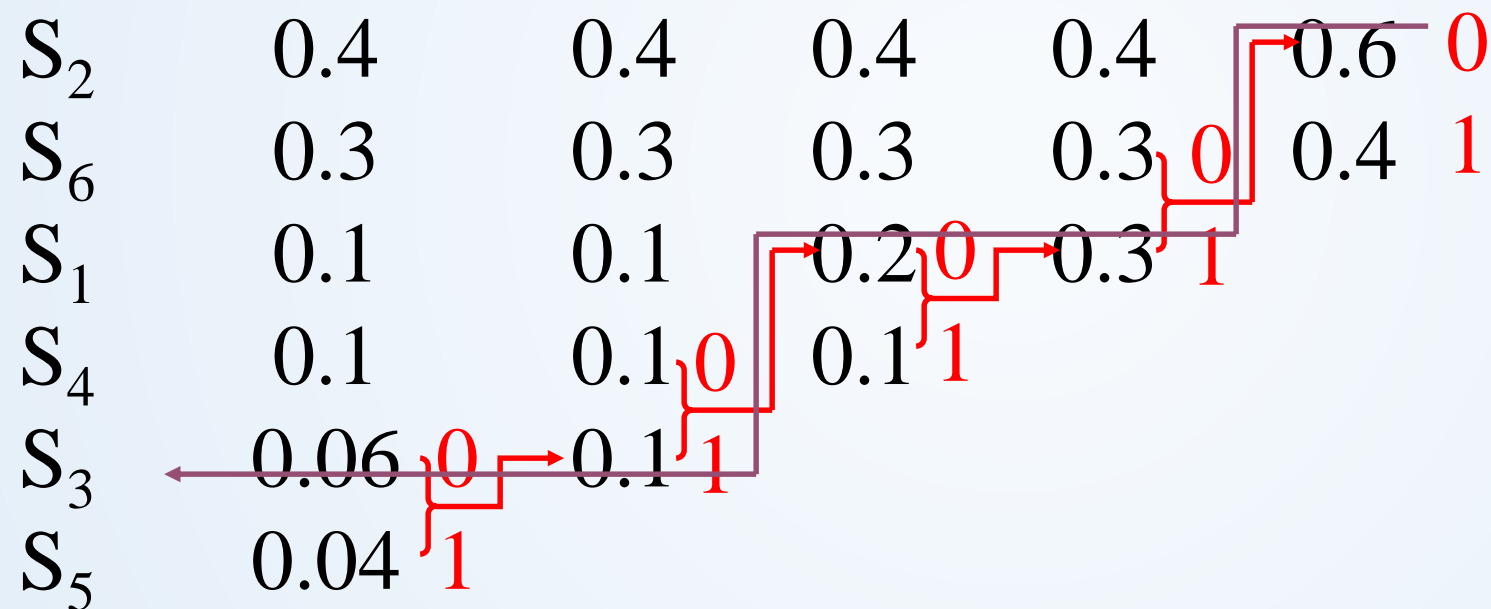
$S_4=0100$



统计编码

Huffman编码

输入 输入概率 第一步 第二步 第三步 第四步



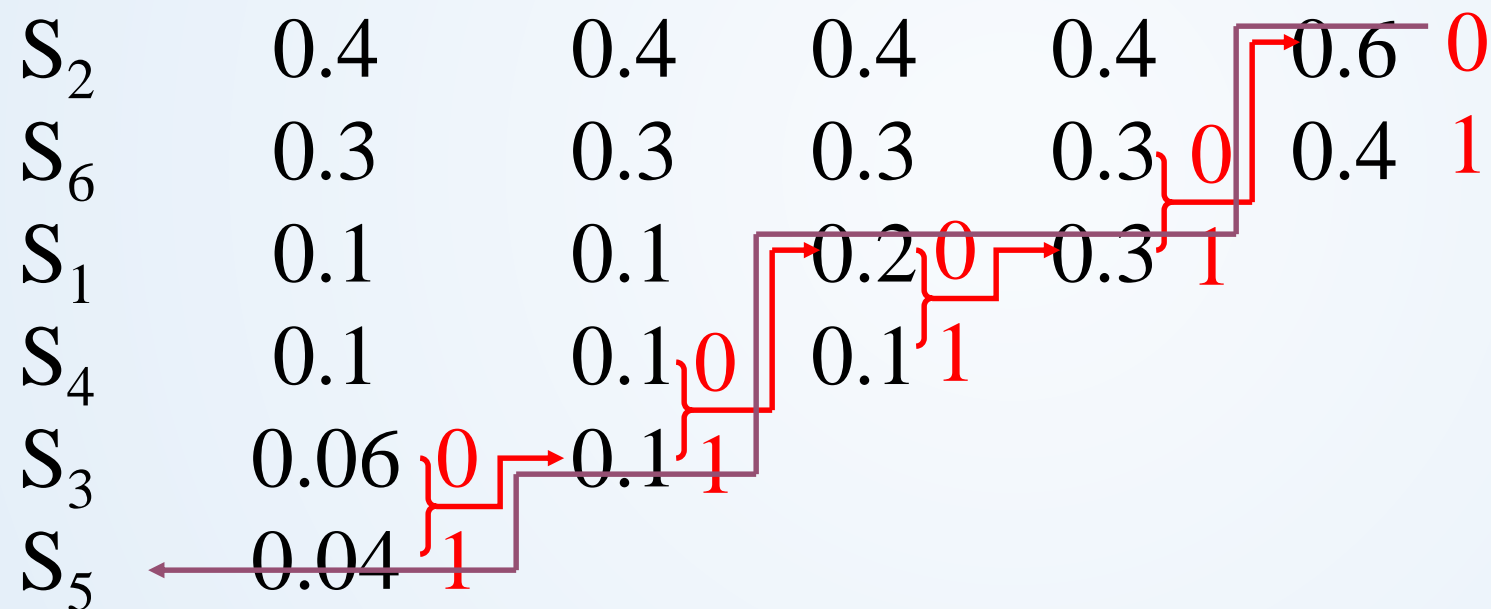
$$S_3 = 01010$$



统计编码

Huffman编码

输入 输入概率 第一步 第二步 第三步 第四步



$S_5=01011$



统计编码

Huffman编码

FIGURE 8.12
Huffman code
assignment
procedure.

Original source			Source reduction							
Sym.	Prob.	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								

Huffman编码平均字长:

$$L_{avg} = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5 = 2.2 \text{ bit / symbol}$$



统计编码

Huffman编码

利用Huffman编码过程建立Huffman码表，**编码/解码**就可以通过简单的**查表**来实现。

Symbol	Probability	Code
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 -

Huffman码表

Huffman编码的特点：

- 一种**块码**：各个信源符号都被映射成一组固定次序的码符号
- 一种**即时码**：编码符号中的每个码字都无需参考其后的符号就可以解码
- 一种**唯一可解的码**：任何编码符号都只能以唯一的方式进行解码（只有唯一解码结果）



统计编码

算术编码

- ◆ 算术编码并不是将单个信源符号映射成一个码字，而是把整个消息序列表示为实数线上0到1之间的一个区间，其长度等于该序列的概率
- ◆ 在该区间内选择一个代表性的小数，转换为二进制作为实际的编码输出
- ◆ 消息序列中的每个元素都要用来压缩这个区间
- ◆ 消息序列中元素越多，所得到的区间就越小；当区间变小时，就需要更多的位数来表示这个区间



统计编码

算术编码

例：对来自一个4符号信源 $\{a_1, a_2, a_3, a_4\}$ 的序列 $a_1a_2a_3a_4$ 进行编码。

1. 计算各个符号出现在序列中的概率，并按照各个符号的概率将 $[0, 1)$ 区间分成对应的子区间

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	$[0.8, 1.0)$

2. 根据当前编码的符号选择对应的子区间，按照各个符号的概率将这个子区间再次划分成下一级的子区间
3. 重复步骤2，直到所有的符号编码完，从最终的子区间中选择任意小数作为编码结果

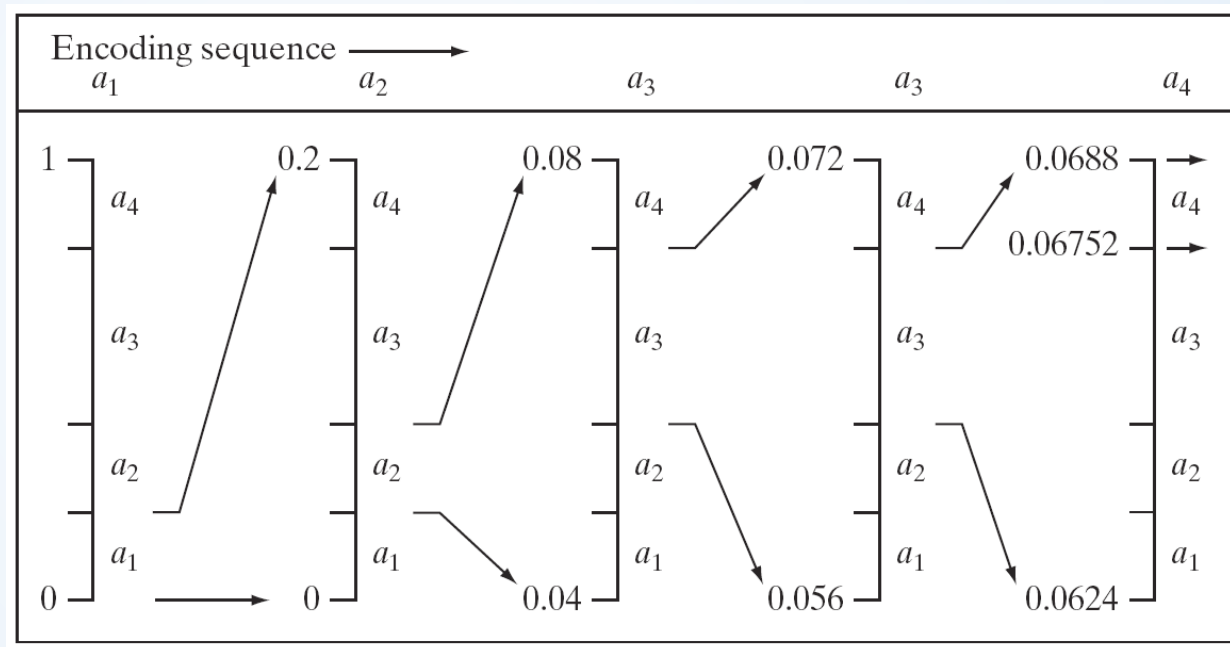


统计编码

算术编码

例：对来自一个4符号信源 $\{a_1, a_2, a_3, a_4\}$ 的序列 $a_1a_2a_3a_3a_4$ 进行编码。

Source Symbol	Probability
a_1	0.2
a_2	0.2
a_3	0.4
a_4	0.2



$$0.056 = 0.04 + (0.08 - 0.04) \times 0.4$$

$$0.072 = 0.04 + (0.08 - 0.04) \times 0.8$$

$$0.0624 = 0.056 + (0.072 - 0.056) \times 0.4$$

$$0.0688 = 0.056 + (0.072 - 0.056) \times 0.8$$

$$0.06752 = 0.0624 + (0.0688 - 0.0624) \times 0.8$$



统计编码

算术编码

编码过程

步骤	输入	编码间隔
1	a1	[0, 0.2)
2	a2	[0.04, 0.08)
3	a3	[0.056, 0.072)
4	a3	[0.0624, 0.0688)
5	a4	[0.06752, 0.0688)
6	从[0.06752, 0.0688)中选择一个数输出: 0.06752	



统计编码

算术编码

解码过程

1. 查看编码数值落在哪个信源符号对应的区间范围内。找到消息中的第一个信源符号，0.06752在 $[0, 0.2)$ 内，所以第一个符号为a1
2. 从编码数值中消去第一个符号a1的影响，即首先减去a1的所在区间的下界值，然后除以a1对应区间的宽度，即
$$(0.06752 - 0) / 0.2 = 0.3376$$
3. 查表找到该结果(0.3376)落入哪一个符号对应的区间范围，0.3376在 $[0.2, 0.4)$ 内，得到第二个符号a2
4. 重复步骤2和3，直至解出整个符号流



统计编码

算术编码

解码过程

步骤	被解码的数值	解码符号	该符号对应的范围
1	0.06752	a1	[0, 0.2)
2	0.3376	a2	[0.2, 0.4)
3	0.688	a3	[0.4, 0.8)
4	0.74	a3	[0.4, 0.8)
5	0.85	a4	[0.8, 1)
6	译码输出: a1a2a3a3a4		



统计编码

算术编码

算术编码的特点：

- ◆ 算术编码对**整个消息**只产生**一个码字**，这个码字是在间隔 $[0, 1)$ 中的一个实数，因此解码器在接收到这个实数的所有位之前不能进行解码
- ◆ 算术编码是一种**对错误很敏感**的编码方法，如果有一位发生错误就会导致整个消息解码错误
- ◆ 为了分开各个符号序列，需指定**序列结束标志**
- ◆ 算术操作**精度有限**



统计编码

行程编码

- ◆ 在一个逐行存储的图像中，具有**相同灰度值**的一些象素组成的**序列**称为一个**行程**。在编码时，对于每个行程只存储一个灰度值的码，再紧跟着存储这个行程的长度。
- ◆ 适用于有较多**灰度相同**对象的图像，例如海洋、湖泊的卫星图像，医学图像中的细胞，染色体，材料的显微图像等。

行程编码采用**整数对**进行编码，例如右侧图像可以编码为 (4, 4) (4, 1) (5, 3) (4, 1) (5, 1) (7, 2) (5, 4)

4	4	4	4
4	5	5	5
4	5	7	7
5	5	5	5



统计编码

行程编码

行程编码的效率

设图像的灰度级为 M ，一行的长度为 N ，则对每一行来说，行程数最少为1，最大为 N 。并设一行中的行程数为 m ，则描述一行像素需要的码字长度为：

$$m (\log_2 M + \log_2 N) \text{ bit}$$

而直接存储原图像一行所需的位数为： $N \log_2 M \text{ bit}$

显然，只有当 $m \ll N$ 时，行程编码才是可取的。

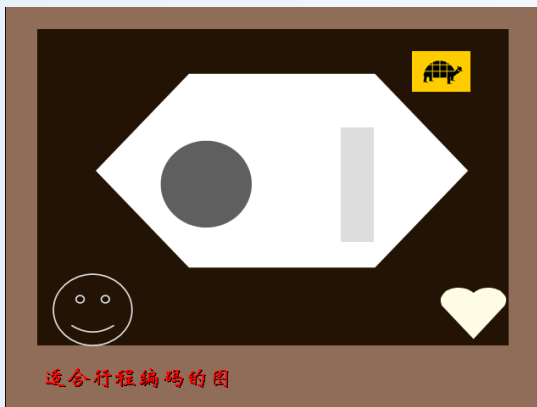


统计编码

行程编码

行程编码的特点

- 对于有大面积色块的图像，压缩效果很好
- 直观，经济，是一种无损压缩
- 对于纷杂的图像，压缩效果不好，最坏情况下，会使图像数据加倍



适合用行程编码进行压缩的图像

主要内容

Main Content



数字图像压缩基础

统计编码

预测编码



预测编码

预测编码

- ◆ 预测编码 (Predictive Coding)，就是根据“过去”时刻的像素值，运用一种模型来预测当前的像素值，然后对预测误差进行编码。当预测比较准确，误差较小时，即可达到压缩数据的目的。
- ◆ 预测编码通过消除近邻像素在空间和时间上的冗余来实现数据压缩，它仅对每个像素中的新信息进行提取和编码。一个像素的新信息定义为该像素的实际值与预测值之间的差。



预测编码

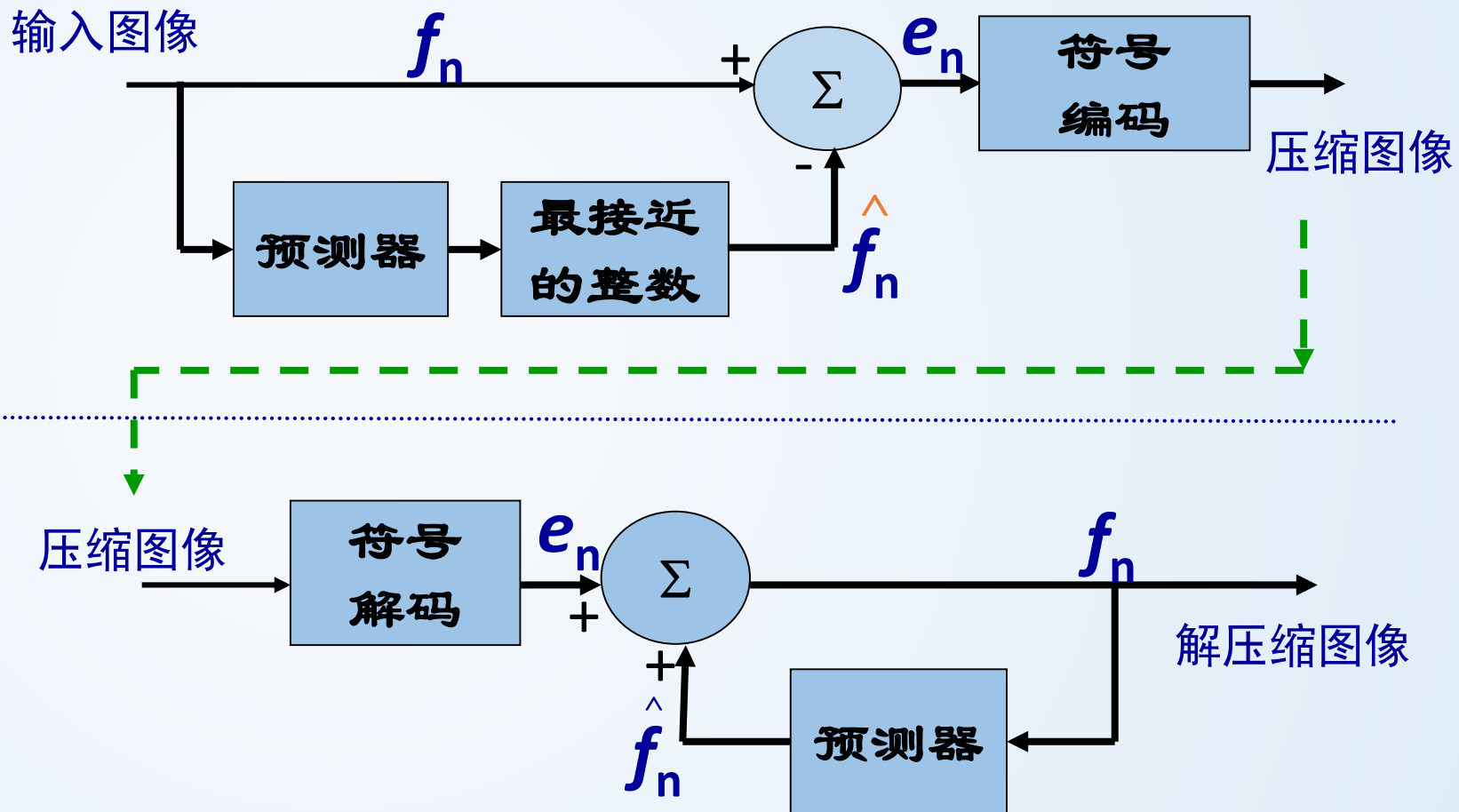
无损预测编码

- ◆ 无损预测直接对预测误差进行符号编码
- ◆ 预测值由前m个样值的线性组合生成

$$\hat{f}(n) = \text{round} \left[\sum_{i=1}^m \alpha_i f(n-i) \right]$$

$\alpha_i, i = 1, 2, \dots, m$ 称为预测器系数

m 称为预测器的阶数



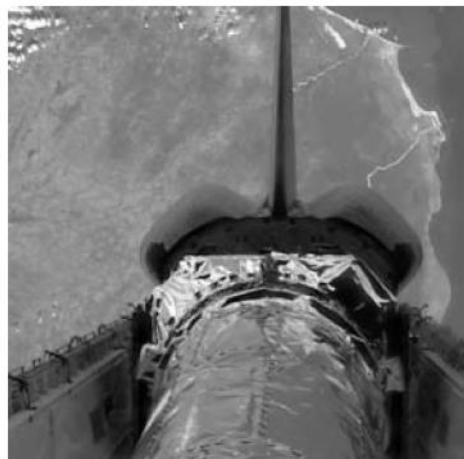


预测编码

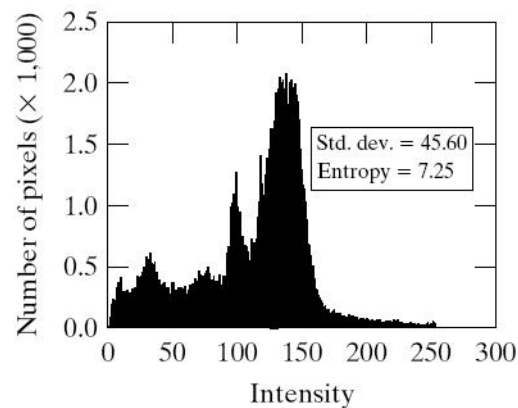
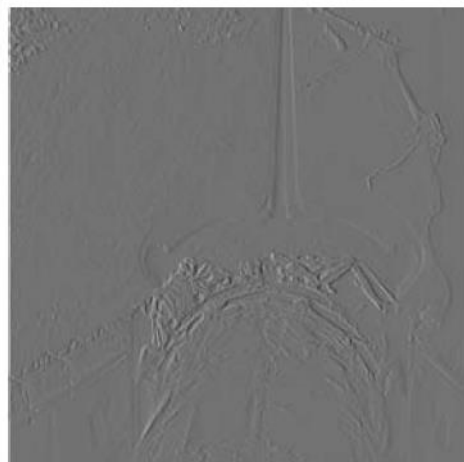
无损预测编码

- ◆ 预测误差图像的灰度分布范围远远小于原始图像灰度的分布范围
- ◆ 对误差进行编码需要的比特数显著降低

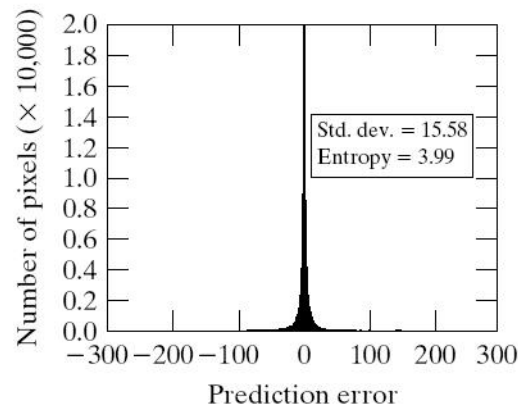
原始
图像



预测 误差 图像



原始图像的灰度直方图



预测误差图 像的灰度直 方图



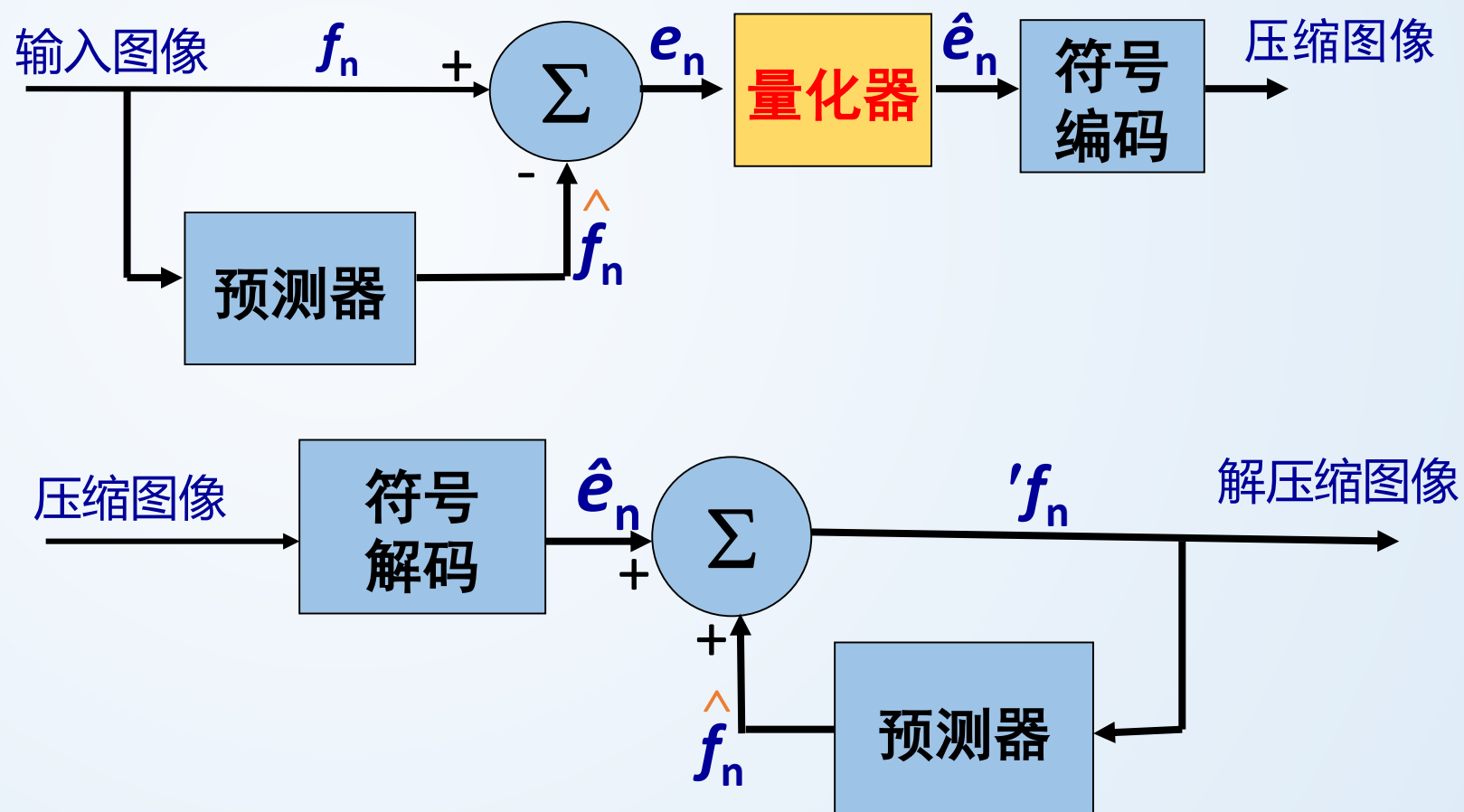
预测编码

有损预测编码

- ◆ 有损压缩是通过牺牲图像的准确度来达到增大压缩率的目的
- ◆ 有损压缩当压缩比大于30:1时，仍然能够重构图像；压缩比为10:1到20:1时，重构图像与原图几乎没有差别
- ◆ 无损压缩的压缩比很少有超过3:1的

有损预测编码与无损预测编码的区别在于增加了量化器

$$\hat{e}_n = Q(f_n - \hat{f}_n)$$



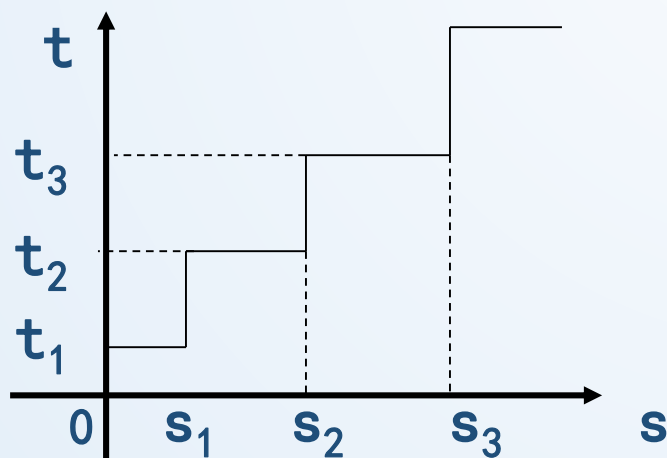


预测编码

有损预测编码

量化器基本思想：

- 减少数据量的最简单的办法是将图像**量化成较少的灰度级**，通过减少图像的灰度级来实现图像的压缩
- 量化是**不可逆**的，因而解码时图像有损失



主要内容

Main Content

数字图像压缩基础

统计编码

预测编码

变换编码



武汉大学
Wuhan University

谢谢!

2018.11.07.

