# 第19讲：数据压缩

- 什么使压缩成为可能?
- 压缩系统的要素
- 信息论概念
  - 信息与熵
  - 无损编码定理
  - 率失真曲线和信源编码定理
- 量化
  - 均匀量化
- 霍夫曼编码
- 算术编码
  - 二进制算术编码

# 为什么进行图像压缩?

- 数字图像需要**压缩**才能进行有效的**存储**和**传输**。
- **高分辨率彩色照片**（2400万像素摄像头）

  - 6000像素×4000行×24比特/像素= 576,000,000bit= 68.7 MB
- **标准摄影质量彩色照片**

  - 1920像素×1300行×24比特/像素= 59,904,000比特= 7.14MB
- **低分辨率彩色照片**

  - 640像素×480行×24比特/像素= 7,372,800比特= 900kB
- **文件和传真图片**

  - 以200 dpi扫描的8.5"×11"页面是1700×2200二进制图像= 467,500字节（456.54 kB）
- **医学图像**

# 什么使压缩成为可能?

- 图像中含有大量的:
  - 空间冗余(相关)
  - 视觉冗余

- 冗余度越高，可实现的压缩率越高。

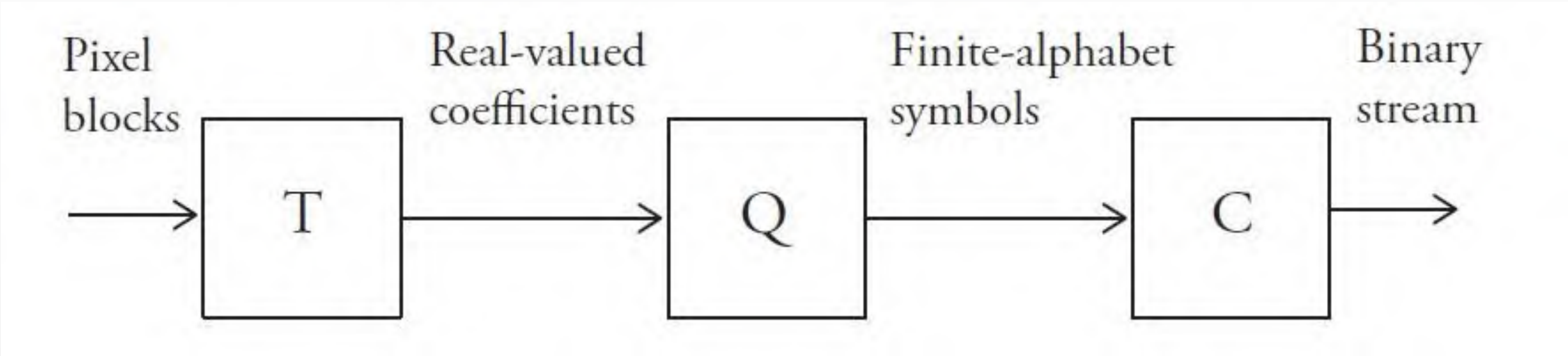
# 无损 vs.有损压缩

- 无损压缩
  – 在没有任何失真条件下对比特率最小化
- 近无损压缩
  – 在给定最大每像素失真(±NEAR)条件下对比特率最小化
- 有损压缩
  – 给定比特率（码率控制）下最大化保真度

  或者

  – 给定保真度度量（固定质量，码率可变）下最小化比特率

# 压缩系统的要素




- **变换/分解/表示**

  使用可有效压缩的形式表示图像数据。

- **量化**（在无损压缩中被省略）

  减少/限制符号的数量来表示数据。

- **符号编码**

  最小化用来表示符号的二进制码的平均长度

# 信息论概念

- 数据压缩的**基本原理:**

  ***将更短的码字分配给可能性更大的符号。***

- 取自有限字符集合A的信源X包含有限数字M个符号，
  即: A = $\{a_1, a_2, \ldots, a_M\}$， 各符号概率 $p(a_i) = p_i$, $i = 1, \ldots, M$.

- **离散无记忆源（DMS）**是具有有限字符表的信源，其中所有信源符号在统计上是独立的。

- 具有概率$p$的符号$a_i$的信息量由下式给出：

$$I(a_i) = \log_2(1/p(a_i))$$

  注意，如果p = 1，则I = 0，并且如$p \rightarrow 0$, $I \rightarrow \infty$。如果我们使用对数底为2，则信息单位是比特。

- 可以从**信源的直方图**或一些训练数据估计符号的概率。

- 在可变长编码中，符号的**最佳码字长度等于符号的信息量**（以比特为单位）。

# 熵

- **信源X的熵H**是每个符号信息量的期望值（概率加权平均）：

$$H(X) = \sum_{a_i \in A} p(a_i)\log_2\left(\frac{1}{p(a_i)}\right) = -\sum_{a_i \in A} p(a_i)\log_2(p(a_i))$$

- 信源熵表示每个符号的**平均码字长度**。
- 符号的概率分布函数（直方图）**越偏斜**，信源的**熵越小**。**熵最大化为平坦分布**，即当所有符号等概时。
- 示例 - **原始图像的熵**：令符号i为8比特/像素图像的灰度级。则原始图像的熵由下式给出

$$H(X) = -\sum_{i=0}^{255} p(i)\,\mathrm{Log}_2(p(i))$$

其中 p(i) 表示灰度级i的相对发生频率。

# 无损编码定理

- 因此，与具有较大熵的源相比，具有**较小熵的信源可以以更小的平均码长**进行编码。这可由无损编码定理推出。
- **无损编码定理** [Shannon，1948]

可变长编码方案的性能可以通过由下式给出的下限来度量:

$$\min R = H(\mathbf{X}) + \varepsilon \quad \text{bits/symbol}$$

其中**R是平均码率**，$H(\mathbf{X})$是信源熵，ε是任意接近零的正数。

**编码效率**：$\eta = \dfrac{H(\mathbf{X})}{R}$

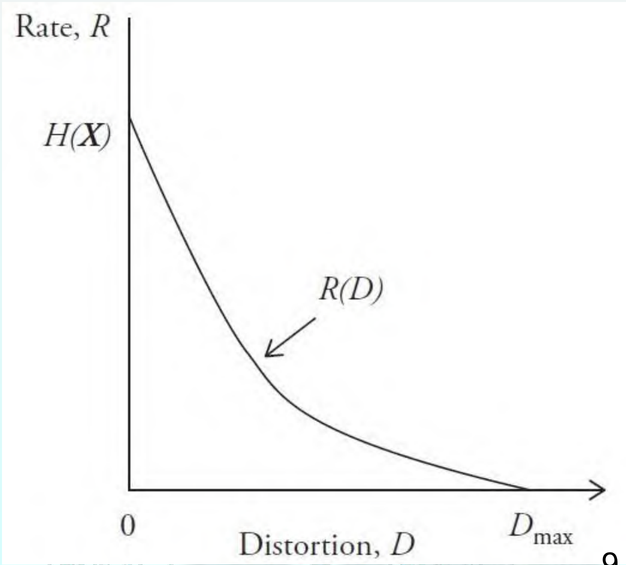
# 信源编码定理

- 在**有损编码**中，可实现的最小比特率是**关于允许失真的函数**。比特率和失真之间的关系由率失真函数给出，并可由信源编码定理推出。
- **信源编码定理** **[Ber 71]:** 存在从信源符号到码字的映射，使得对于给定的失真D，使用R( D)比特/符号就足以使信源重建后的平均失真任意接近D。
- **实际的码率 *R*** 应遵循：对于给定保真度

$$R \geq R(D)$$

- 函数 R( D)被称为**率失真函数**，

$R(0) = H(\mathbf{X})$ .



# 量化

- **标量量化器Q(·)**是由有限集合的判决电平$d_i$和重建电平$r_i$定义的函数。量化变量s由下式给出：

$$\dot{s} = Q(s) = r_i \quad \text{if } s \in (d_{i-1}, d_i], \quad i = 1, \dots, L$$

其中$L$是输出量化电平的数量。

量化器的性能取决于$d_i$和$r_i$的量化误差$e = s - \dot{s}$所导致的失真D。常见的失真度量是均方误差$D = E\{(s - \dot{s})^2\}$，其中$E\{\cdot\}$是求期望。

- 给定**失真测度D**和**信源pdf** $p_S(s)$，有**两种最优标量量化器**设计方法：
  - **Lloyd-Max量化器**：对于固定数量的L，找到$r_i$和$d_i$，$i = 0, \dots, L$，以便最小化均方差失真度量D.
  - **熵约束量化器**：对于固定输出信源熵C，找到L，$r_i$和$d_i$，$i = 0, \dots, L$，（L为未知），以最小化失真度量D。

# 均匀量化

- 如果**连续重建电平**之间的距离θ（称为步长）相等，则量化器称为**均匀量化器**，即：

$$r_{i+1} - r_i = \theta, \qquad 1 \le i \le L - 1.$$

- 均匀量化器可分为中平量化器和中升量化器。**中平量化器**为：

$$Q(s) = \operatorname{sgn}(s)\left\lfloor \frac{|s|}{\theta} + \frac{1}{2} \right\rfloor \theta = \operatorname{NINT}\left(\frac{s}{\theta}\right)\theta$$

  其中 $\lfloor x \rfloor$ 表示向下取整，具有零值重建电平。

  而**中升量化器**为：

$$Q(s) = \left(\left\lfloor \frac{s}{\theta} \right\rfloor + \frac{1}{2}\right)\theta$$

  具有**零值判决电平**。

- 当信源的pdf**关于s = 0对称**，并且对于更大的s值衰减时，可优先选择中平量化器。
- 在中平量化器中，s = 0周围的判定区称为**死区**。

# 示例：JPEG中的均匀量化

- 在JPEG有损图像压缩中，**信源（DCT系数）的pdf**由零均值**拉普拉斯分布**建模，即：

$$p_S(s) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|s|}$$

- 因此，采用**中平均匀量化**。
- 编码器**计算整数量化索引值**：

$$k = \mathrm{NINT}\left(\frac{s}{\theta}\right)$$

然后通过**熵编码**后用于传输/存储。

- **重建值**在步长为θ的情况下，由解码器计算：

$$\dot{s} = Q(s) = k\theta$$

# 示例: 量化噪声

- 假设有一个**无记忆零均值高斯信源S**，其方差为$\sigma^2$，失真度为均方误差，则如果均匀量化，为了获得40dB的SNR，最小的量化级别数量时多少？或等效地，以比特/像素为单位的速率R是多少？
- 我们可以将均方量化噪声表示为:

$$D = E\{(s - \dot{s})^2\}$$

- 以**dB为单位的信噪比**定义为:

$$SNR = 10\log_{10}\frac{\sigma^2}{D} = 40dB \quad \rightarrow \quad \frac{\sigma^2}{D} = 10{,}000$$

- 将其代入**无记忆高斯信源的率失真函数**,

$$R(D) = \frac{1}{2}\log_2\frac{\sigma^2}{D}$$

我们可以计算作为失真函数的速率R（D）≈7比特/样本。

- 类似地，8比特/像素的量化可获得大约48dB的SNR。

# 无损符号编码

- **固定长度编码**

    （将固定长码字分配给固定长的块）

- **可变长（熵）编码**
    - **霍夫曼编码**

        （将可变长码字分配给固定长度的块）

    - **算术编码**

        （将可变长码字分配给可变长块）

如果**某些符号比其他符号可能性更大**（实际上这就是为什么要有变换模块），则使用VLC。

# 固定长编码

| *Symbol* | *Code* |
|---|---|
| $a_1$ | 00 |
| $a_2$ | 01 |
| $a_3$ | 10 |
| $a_4$ | 11 |

**平均码长为 2；信源熵为 2。编码效率100%**。

- 如果满足如下条件，则**固定长度编码**是**最优**的（信源熵等于固定码字长度）：

1) 符号的**数量等于2的幂**，并且

2) 所有的**符号**都是**等概**的。

# 霍夫曼编码

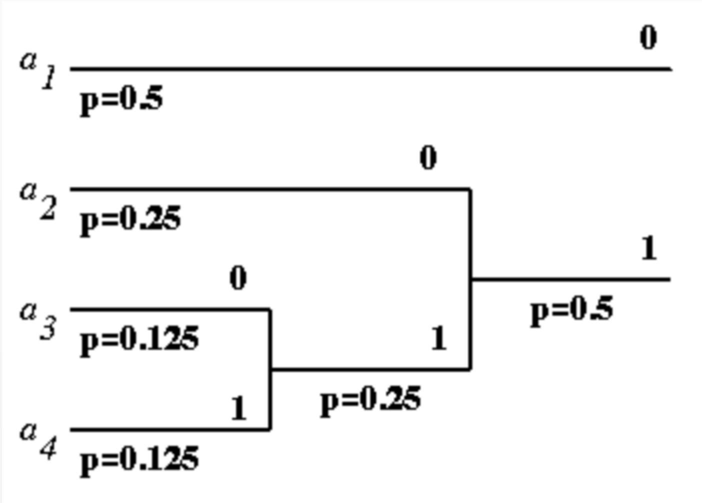
- **问题**：给定一个具有有限个符号的信源及其概率; 找到最优的具有（最小平均码字长度）整数长度的前缀码。
- **解**：令U表示信源，A为信源符号，p(a)为符号概率，$a \in \boldsymbol{A}$。
  - 如果A只有两个符号，则有：

$$c(a_1) = 0; \quad c(a_2) = 1$$

  - 如果A有两个以上的符号，我们合并两个最小概率的符号。这相当于产生了一个新的信源U’，其字符集A’的字符数量减少。

# 示例:概率是2的幂

| *Symbol* | *Probability* | *Information* |
|---|---|---|
| $a_1$ | 0.50 | 1 bit |
| $a_2$ | 0.25 | 2 bits |
| $a_3$ | 0.125 | 3 bits |
| $a_4$ | 0.125 | 3 bits |




**平均码长为:**

$$R = 0.5 \times 1 + 0.25 \times 2 + (0.125 \times 3) \times 2 = 1.75$$

**信源熵为:**

$$H = -0.5 \log_2 0.5 - 0.25 \log_2 0.25 - (0.125 \log_2 0.125) \times 2 = 1.75$$

$$\eta = H/R = 100\%$$

- 当符号**概率为2的幂**时，霍夫曼编码可达到信源熵。

# 示例: 概率不是 2的幂

| *Symbol* | *Probability* | *Information* |
| --- | --- | --- |
| $a_1$ | 0.40 | 1.32 bits |
| $a_2$ | 0.25 | 2 bits |
| $a_3$ | 0.15 | 2.73 bits |
| $a_4$ | 0.15 | 2.73 bits |
| $a_5$ | 0.05 | 4.32 bits |

平均码长为2.15；信源熵为2.07。

$\eta = 2.07/2.15 = 96.28\%$

| *Original Source* | | *Reduced Source 1* | | *Reduced Source 2* | | *Reduced Source 3* | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prob. | Code | Prob. | Code | Prob. | Code | Prob. | Code |
| 0.40 | 1 | 0.40 | 1 | 0.40 | 1 | 0.60 | **0** |
| 0.25 | 01 | 0.25 | 01 | 0.35 | 0**0** | 0.40 | **1** |
| 0.15 | 001 | 0.20 | 00**0** | 0.25 | 0**1** | | |
| 0.15 | 000**0** | 0.15 | 00**1** | | | | |
| 0.05 | 000**1** | | | | | | |

# 霍夫曼码的解码

- 霍夫曼码在适当的同步条件下是**唯一可解码**的，因为没有任何码字是另一个码字的**前缀**。
- **不容许比特错误**。检测到每个比特错误后，需要重新同步。
- 示例：接收到二进制码流：

001101101110000…

可被唯一解码为：

$a_3\, a_1\, a_2\, a_1\, a_2\, a_1\, a_1\, a_4 \ldots$

# 分组霍夫曼编码

- 霍夫曼码也可以**一次性将码字**分配给**具有L个符号**的分组。
- 这需要使用原始符号表中L个符号的**所有可能组合**构建新的分组符号表并计算其概率。
- 情况L ＝ 1是指将**单个码字**分配给原始符号表的每个符号，如上述两个例子所示。
- 已经表明，对于有记忆信源，**编码效率**随着**L变大而提高**，尽管此时霍夫曼码的设计将变得更加**复杂**。

# 算术编码

- 符号表**A**的符号和码字之间**没有一一对应的关系**。
- 符号序列**x**= { $x_1,\ldots,x_N$ } 与[0,1）上的**一个子区间相关联**，该子区间长度等于序列概率p(**x**)。
- 编码器**逐个处理输入符号**。在每个比特值确定后，以**从最高有效位开始向最低有效位的顺序**发送到信道。
- 最终传输的比特流是表示信源的唯一可解码码字，是指向与该序列**相关联的子区间的二进制数**。
- 由于整数长度码字不是分配给固定长度的符号分组，所以**当N →∞时算术编码可达到无损编码定理建立的下限**（对于iid信源）。

# 编解码过程

考虑一个**符号表** $\boldsymbol{A}=\{a_i, i=1,\ldots,M\}$，各**符号概率** $p(a_i)=p_i$.

- 1. 令**第一个符号** $x_1=a_i$, 则$I_1=[l_1,r_1)=[p_{i-1}, p_{i-1}+p_i)$，其中 $p_0=0$. 设 $n=1$, $L=l_1$, $R=r_1$, $d=r_1-l_1$.
- 2.计算**L和R的二进制扩展**为:

$$L=\sum_{k=1}^{\infty}u_k 2^{-k} \quad \text{and} \quad R=\sum_{k=1}^{\infty}v_k 2^{-k}$$

其中$u_k$和 $v_k$是 0 或 1.

  - **比较$u_1$和$v_1$**。如果$u_1\neq v_1$，转到步骤3。如果$u_1=v_1$，则发送$u_1$，并比较$u_2$和$v_2$。如果不一样，转到步骤3。
  - **如果$u_2=v_2$**，则也要发送二进制符号$u_2$，并比较$u_3$和$v_3$，依此类推，直到下一个两个对应的二进制符号不匹配，此时转到步骤3。

# 编解码过程 (cont’d)

- 3. 增加n，并读取下一个符号。如果第n个输入符号$x_n = a_i$，则将上一步的**间隔细分**为：

$$I_n = [\, l_n, r_n\,) = [\, l_{n-1}+p_{i-1}d,\ l_{n-1}+(p_{i-1}+p_i)d\,).$$

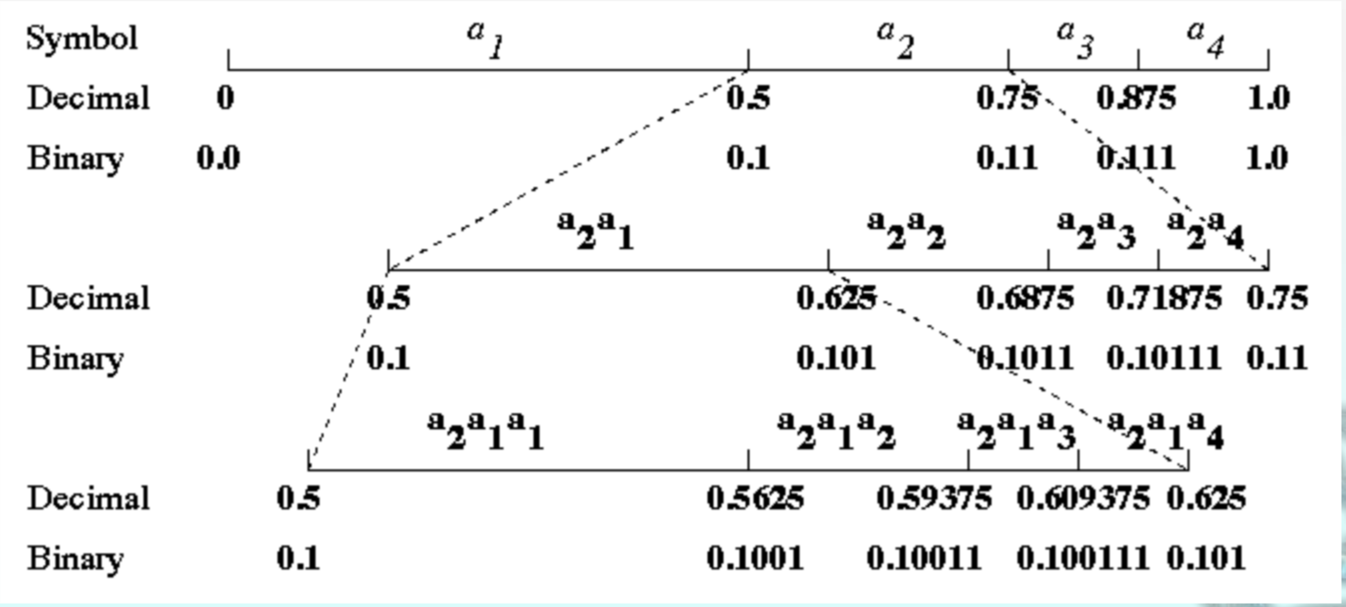
设$L = l_n$，$R = r_n$，$d = r_n - l_n$，**转到步骤2**。

- 解码器在解码一个或多个信源符号之前**可能需要几个二进制符号**。
- 当**符号概率不是2的幂**时，可实现**比霍夫曼编码更好**的性能。

# 示例

| Symbol | Probability | Information |
|---|---|---|
| $a_1$ | 0.50 | 1 bit |
| $a_2$ | 0.25 | 2 bits |
| $a_3$ | 0.125 | 3 bits |
| $a_4$ | 0.125 | 3 bits |

- 确定一个**算术编码来表示**符号序列 $a_2\ a_1\ a_3 \ldots$



# 示例 (cont'd): 编码器操作

- 因为在符号表中有四个符号，所以从**0到1的间隔最初被细分为4个**，其中子间隔的长度分别等于0.5,0.25,0.125和0.125。
- **第一个符号**定义$I_1 = [0.5,0.75]$，其中$L = 2^{-1} = 0.1$，$R = 2^{-1} + 2^{-2} = 0.11$。根据步骤2，$u_1 = v_1 = 1$; 因此，1被发送到信道。注意到$u_2 = 0$和$v_2 = 1$，读取第二个符号$a_1$。
- **步骤3**表示$I_2 = [0.5,0.625]$，$L = 0.10$，$R = 0.101$。由于$u_2 = v_2 = 0$，因此发送0到信道。但此时$u_3 = 0$和$v_3 = 1$，所以读取第三个符号$a_3$。
- $I_3 = [0.59375,0.609375]$，$L = 0.10011$，$R = 0.100111$。注意$u_3 = v_3 = 0$，$u_4 = v_4 = 1$，$u_5 = v_5 = 1$，但$u_6 = 0$和$v_6 = 1$。在这个阶段，向信道发送011，并读取下一个符号。

# 示例 (cont'd): 解码器操作

| *Received Bit* | *Interval* | *Symbol* |
|---|---|---|
| 1 | [0.5,1) | |
| 0 | [0.5,0.75) | a2 |
| 0 | [0.5,0.625) | a1 |
| 1 | [0.5625,0.625) | |
| 1 | [0.59375,0.625) | |
| . | . | . |

**第一个比特**将区间限制为[0.5,1）。但是，在这个范围内有三个符号; 因此，第一个比特没有包含足够的信息。

在接收到**第二个比特**之后，有"10"指向区间[0.5,0.75]。指向此范围的两个符号的所有可能组合从$a_2$开始。因此，此时可以将第一个符号解码为$a_2$。

# 上下文自适应二进制算术编码

- **统计建模+算术编码**
  - 模型越好，算术编码的性能越好
- **二进制算术编码**
  - 信源是二进制的，也就是说，字母表只包含两个符号。
- 可以基于通常由基于**先前编码的像素的8-10个状态**组成的模板（上下文）来对符号概率进行局部更新，使得在编码器和解码器处可以使用相同的上下文。
- **最简单的有限上下文模型是一个0阶模型**，这意味着每个符号的概率与任何先前的符号无关，并且可以由包含每个符号频率计数的单个表进行表示。
- **性能随着模型阶数的增加而增加**，其代价是复杂性和存储要求。