图像压缩

Image Compression

中南大学信息科学与工程学院　陈白帆

chenbaifan@csu.edu.cn

faculty.csu.edu.cn/chenbaifan

图像压缩的基本概念

为什么需要图像压缩

- 图像的数据量通常很大，对存储、处理和传 输带来许多问题（对比视频）
- 不断扩大的图像应用
- Internet上的大量图像
- 数字图书馆
- 遥感图像
- 视频，如电视会议、数字电视、IPTV
- ……

图像压缩的方法

- 消除冗余数据，从数学角度看，将原始图像转化为从统计角度看尽可能不相关的数据集
- 一般分为两类：
  - 无损压缩：在压缩和解压缩过程中没有信息损失。
  - 有损压缩：能取得较高的压缩率，但压缩后不能通过解压缩恢复原状。

图像压缩的方法

- 图像压缩的理论基础
  - 信息论
  - 图像处理的概念和技术
- 压缩方法
  - 预测编码方法（对应空域方法）
  - 变换编码方法（对应频域方法）

数据冗余

- 数据冗余
  - 如果不同的方法为表示给定量的信息使用了不同的数据量，那么使用较多数据量的方法中，有些数据必然是代表了无用的或重复的信息。
- 相对数据冗余
  - 如果n_1和n_2代表两个表示相同信息的数据集合中所携载信息单元的数量，则n_1表示的数据集合的相对数据冗余R_D定义为：

$$R_D = 1 - \frac{1}{C_R}$$

  - C_R称为压缩率，定义为

$$C_R = \frac{n_1}{n_2}$$

数据冗余

- 三种基本的数据冗余
  - 编码冗余
  - 像素间冗余
  - 心理视觉冗余
- 如果能减少或消除上述三种冗余的1种或多种冗余，就能取得数据压缩的效果。

编码冗余

- 如果一个图像的灰度级编码，使用了多于实际需要的编码符号，就称该图像包含了编码冗余。
  - 二值图像
    - 如果用8位表示该图像的像素？
  - 灰度图
    - 图像直方图 $p_r(r_k)=\frac{n_k}{n}$ $k=0,1,2,\ldots,L-1$
    - 每个像素所需的平均比特数为：灰度级所用的比特数和灰度级出现的概率相乘

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

像素间冗余

- 反映图像中像素之间的相互关系。
- 因为任何给定像素的值可以根据与这个像素相邻的像素进行预测，所以单个像素携带的信息相对较少。
- 对于一幅图像，很多单个像素对视觉的贡献是冗余的。它的值可以通过与它相邻的像素值为基础进行预测。

例：原图像数据：	234	223	231	238	235
压缩后数据：	234	-11	8	7	-3

心理视觉冗余

- 由于眼睛对所有视觉信息感受的灵敏度不同，在正常视觉处理过程中各种信息的相对重要程度不同。
- 有些信息在通常的视觉过程中与另外一些信息相比并不那么重要，这些信息被认为是心理视觉冗余的，去除这些信息并不会明显降低图像质量。

心理视觉冗余

- 由于消除心理视觉冗余数据会导致一定量信息的丢失，所以这一过程通常称为**量化**。
- 心理视觉冗余压缩是不可恢复的，量化的结果导致了数据有损压缩。

保真度准则

- 图像压缩可能会导致信息损失，如去除心理视觉冗余数据。
- 需要评价信息损失的测度以描述解码图像相对于原始图像的偏离程度，这些测度称为保真度准则。
- 常用保真度准则分为两大类：
  - 客观保真度准则
  - 主观保真度准则

保真度准则

- 客观保真度
  - 输入图 $f(x, y)$为原图
  - 输出图$\hat{f}(x, y)$为对原图先压缩后解压得到的图
  - 计算输入图和输出图之间的均方根误差

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

保真度准则

- 客观保真度
  - 如果将$\hat{f}(x, y)$看作原始图 $f(x, y)$ 和噪声的和，那么输出图的均方差信噪比SNR_{ms}和均分根信噪比SNR_{rms}为

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

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

保真度准则

- 主观保真度

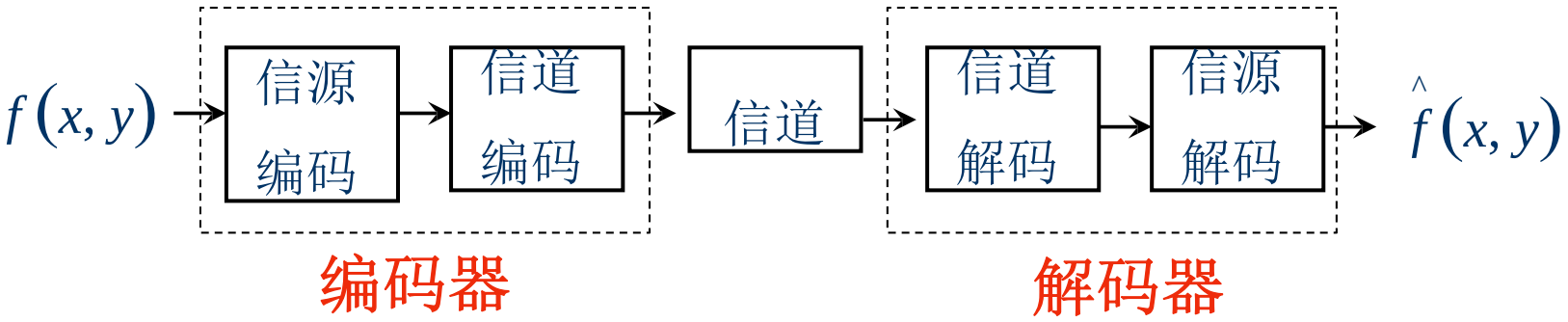
评分	评价	说明
1	优秀	图像质量非常好，如同想象出的最好质量
2	良好	图像质量高，观看舒服，有干扰但不影响观看
3	可用	图像质量可接受，有干扰但不太影响观看
4	刚可看	图像质量差，干扰有些妨碍观看，希望改进
5	差	图像质量很差，妨碍观看的干扰始终存在，几乎无法观看
6	不能用	图像质量极差，不能使用

图像压缩的常用方法

常用的压缩编码方法

- 图像压缩技术
  - 无损压缩
    - 哈夫曼编码
    - 程长编码RLE
    - 算术编码
  - 有损压缩
    - 有损预测编码
    - 变换编码
    - 其他编码

图像压缩模型



- 信源编码：完成原数据的压缩。
- 信道编码：为了抗干扰，增加一些容错、校验位，实际上是增加冗余。
- 信道： 如Internet、广播、通讯、可移动介质

图像压缩模型

- 信源编码器

  - 映射器 ：减少像素冗余，如使用RLE编码或进行图像变换。
  - 量化器 ：减少视觉心理冗余，仅用于有损压缩。
  - 符号编码器：减少编码冗余，如使用哈夫曼编码。

无损压缩

- 哈夫曼编码
  - 利用信息符号概率分布特性的变字长的编码方法。
  - 对于出现概率大的信息符号编以短字长的码，对于出现概率小的信息符号编以长字长的码。

无损压缩

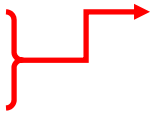
- 哈夫曼编码方法
  - 将信源符号按出现概率从大到小排成一列，然后把最末两个符号的概率相加，合成一个概率。
  - 把新生成的概率与其余的概率按从大到小排列，再把最末两个符号的概率加起来，合成一个概率。
  - 重复上述做法，直到最后剩下两个概率为止。
  - 从最后一步剩下的两个概率开始逐步向前进行编码。每步只需对两个分支各赋予一个二进制码，如对概率大的赋予码0，对概率小的赋予码1。

Huffman编码

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

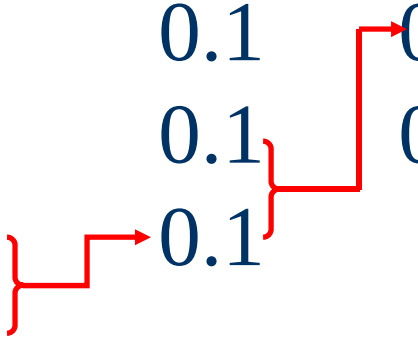
Huffman编码

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



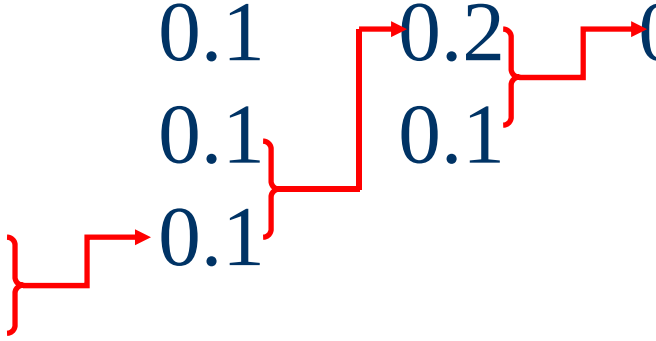
Huffman编码

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



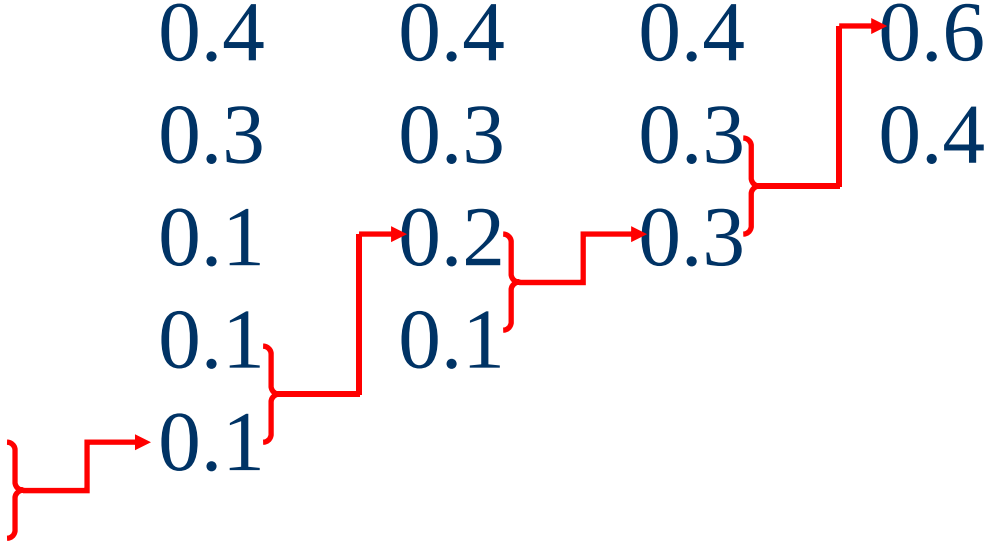
Huffman编码

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

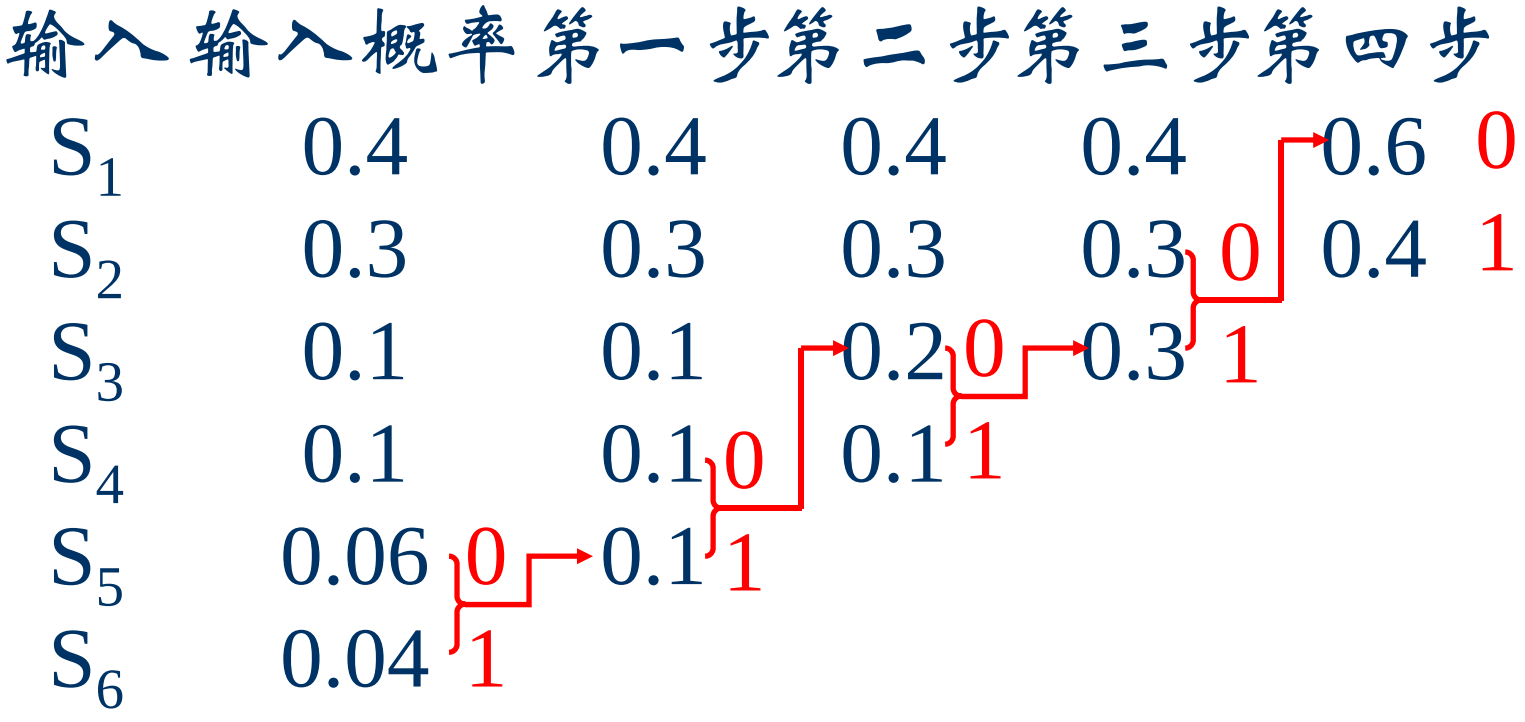


Huffman编码

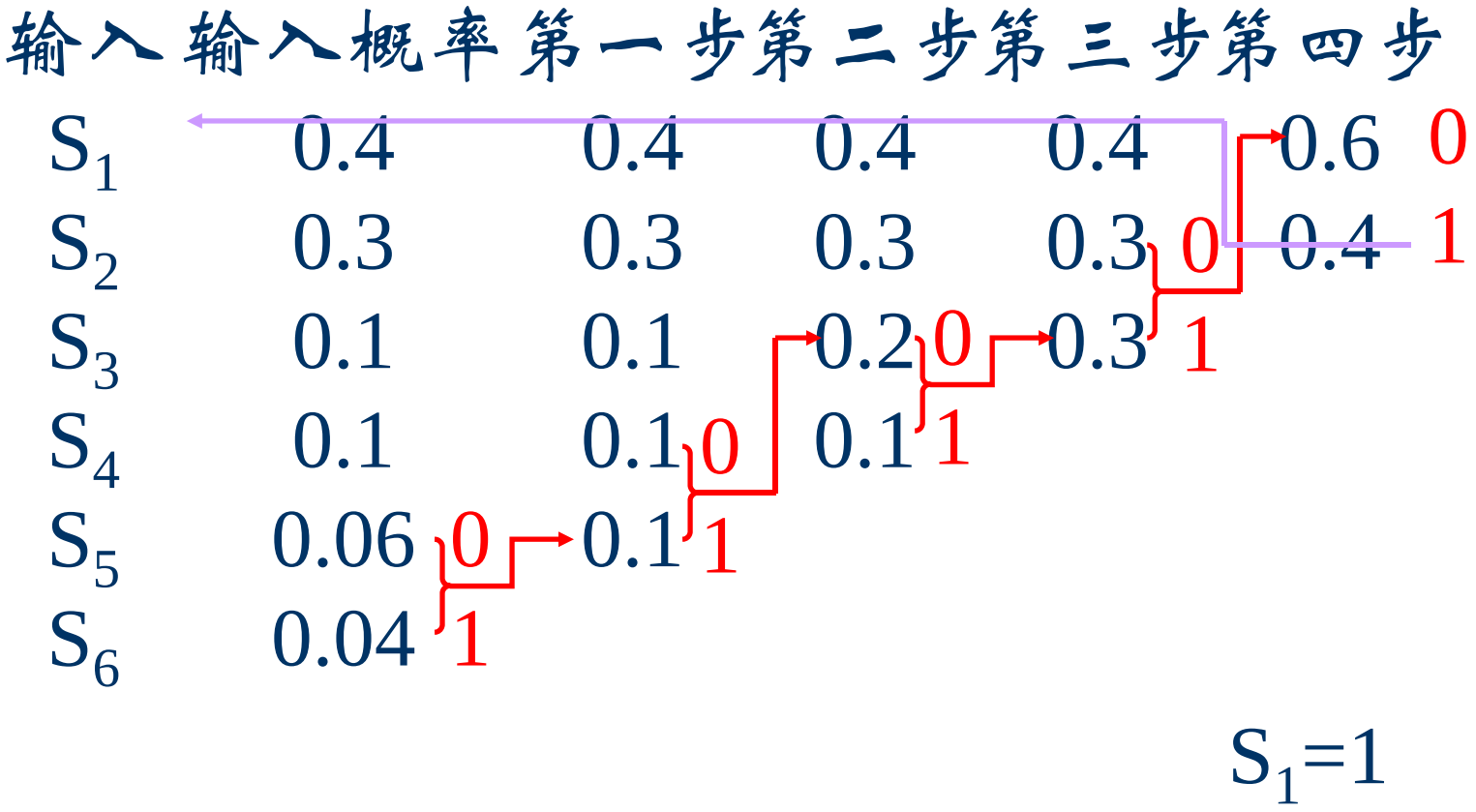
输入	输入概率	第一步	第二步	第三步	第四步
S_1	0.4	0.4	0.4	0.4	0.6
S_2	0.3	0.3	0.3	0.3	0.4
S_3	0.1	0.1	0.2	0.3	
S_4	0.1	0.1	0.1		
S_5	0.06	0.1			
S_6	0.04				



Huffman编码

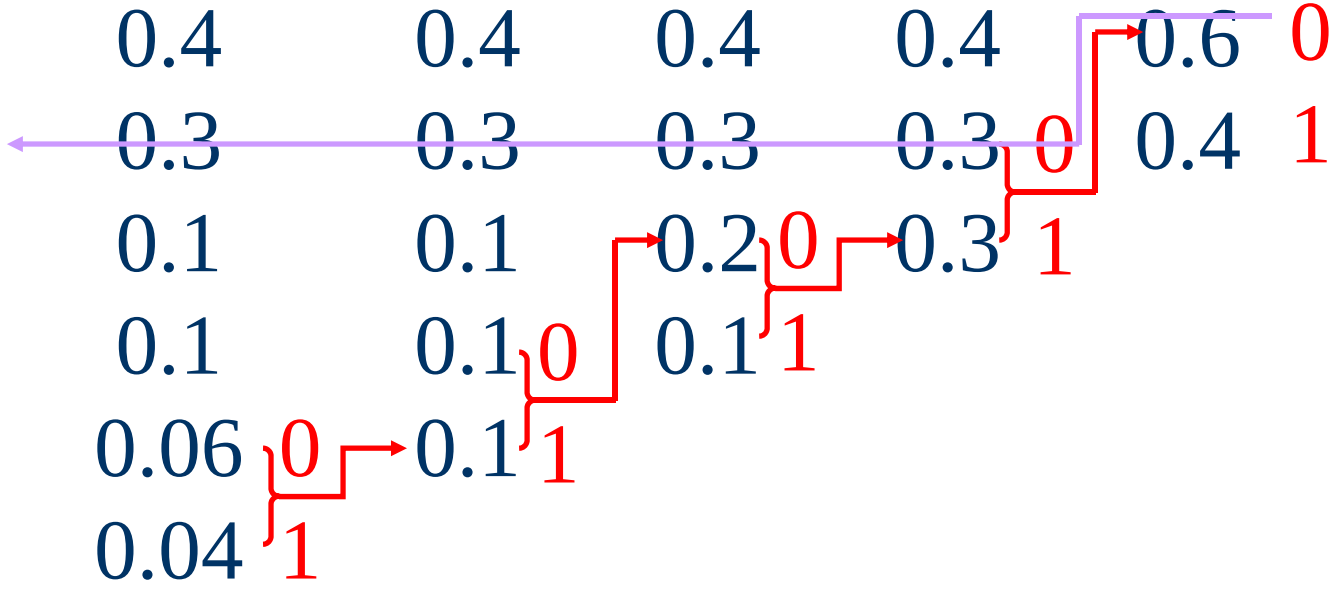

输入 输入概率 第一步 第二步 第三步 第四步
S1 0.4 0.4 0.4 0.4 0.6 0
S2 0.3 0.3 0.3 0.3 0 0.4 1
S3 0.1 0.1 0.2 0 0.3 1
S4 0.1 0.1 0 0.1 1
S5 0.06 0 0.1 1
S6 0.04 1

Huffman编码


输入 输入概率 第一步 第二步 第三步 第四步
S1 0.4 0.4 0.4 0.4 0.6 0
S2 0.3 0.3 0.3 0.3 0 0.4 1
S3 0.1 0.1 0.2 0 0.3 1
S4 0.1 0.1 0 0.1 1
S5 0.06 0 0.1 1
S6 0.04 1
S1=1

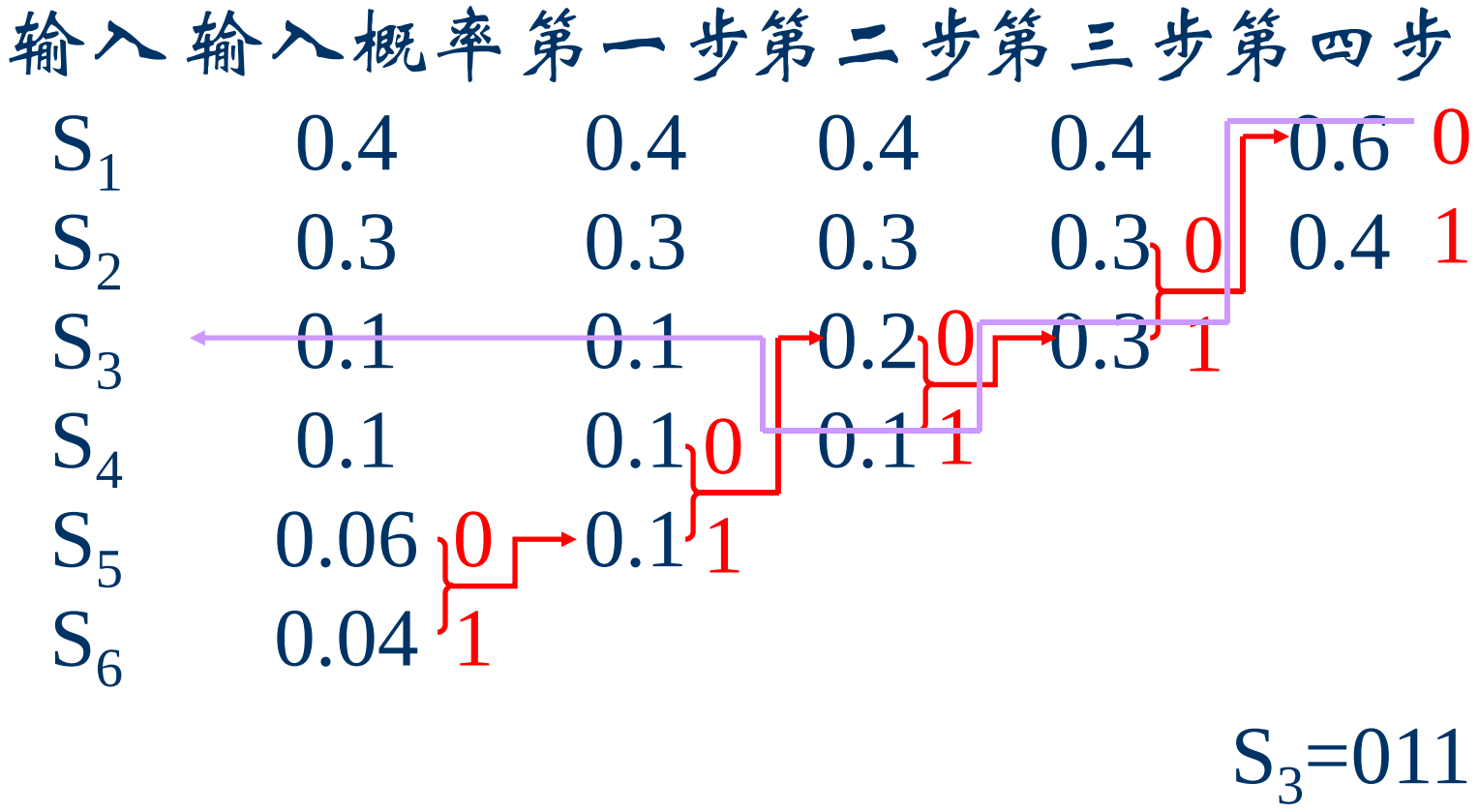
Huffman编码

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



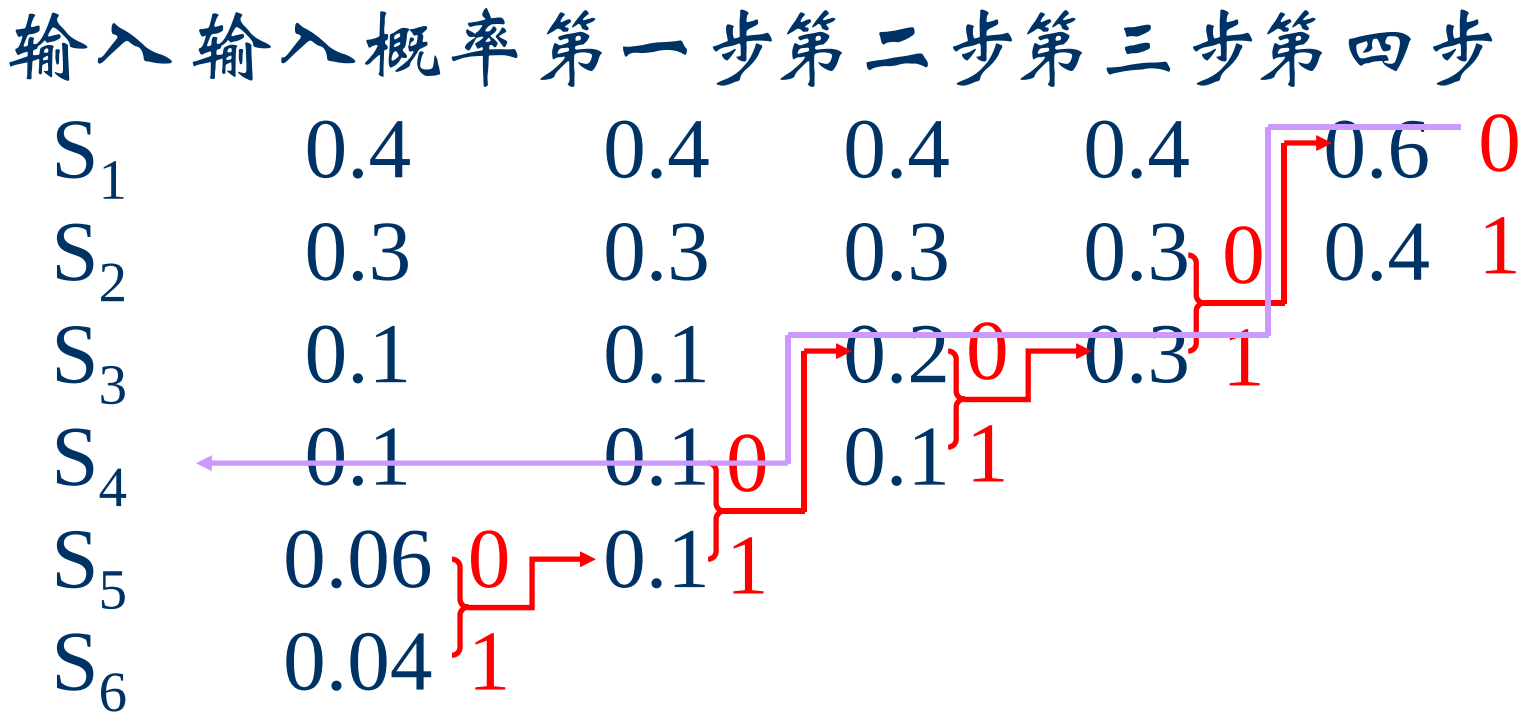
S_2=00

Huffman编码



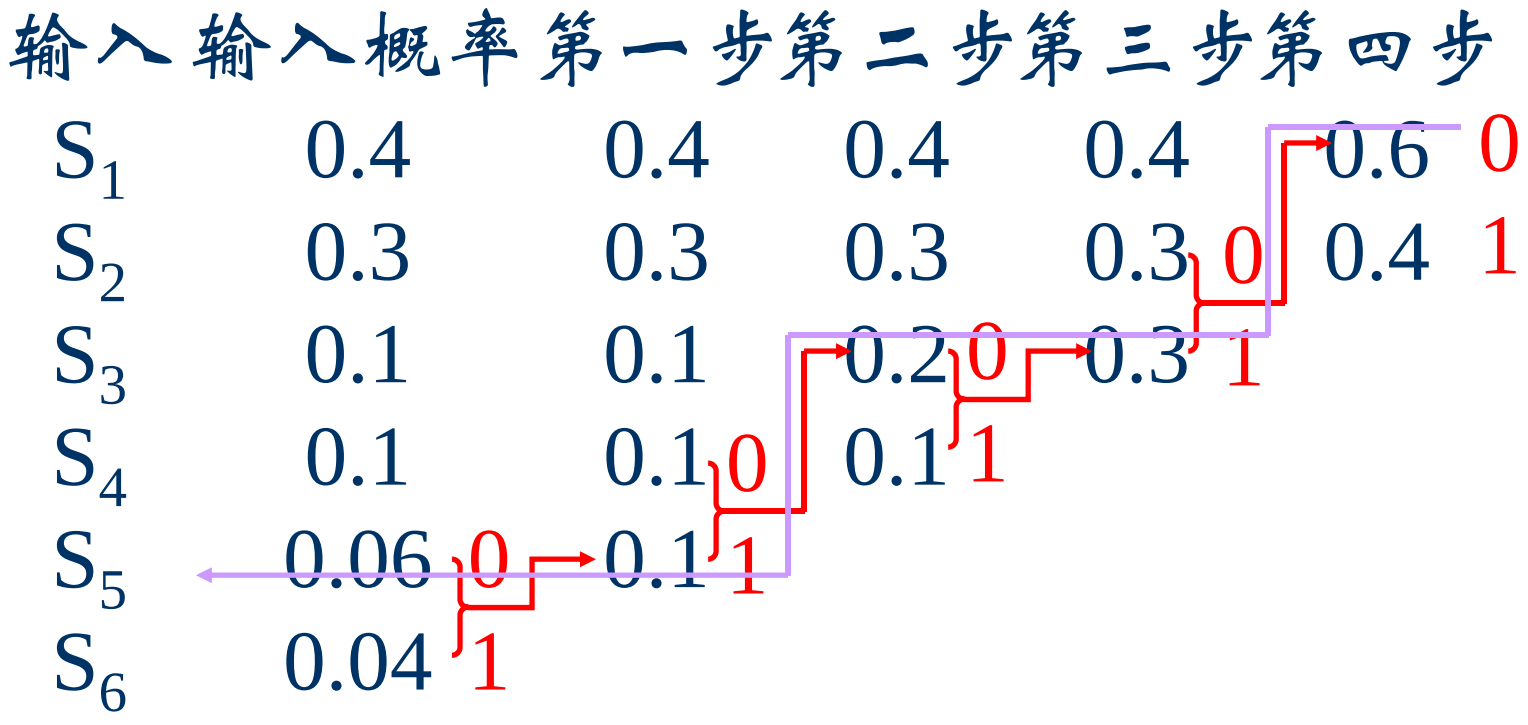
S_3=011

Huffman编码



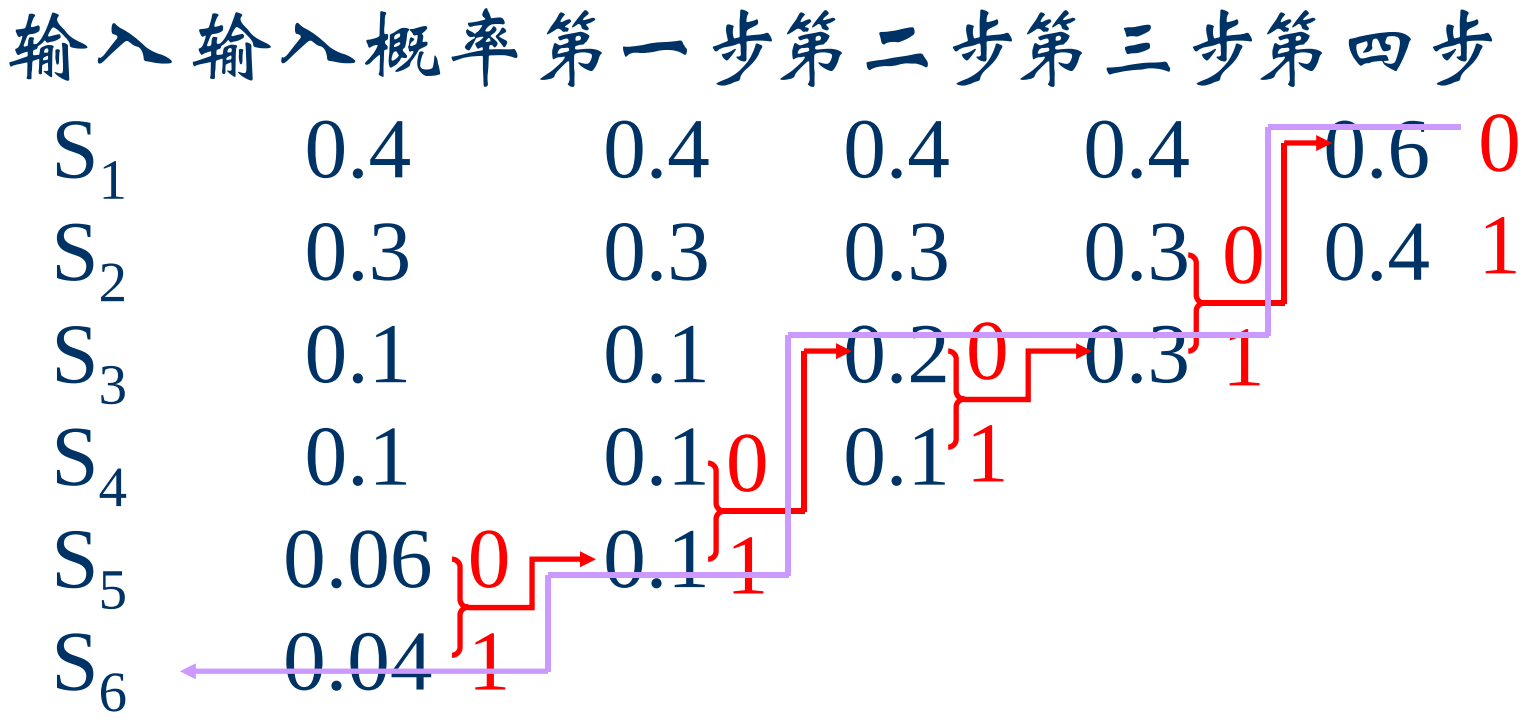
S_4=0100

Huffman编码



S_5=01010

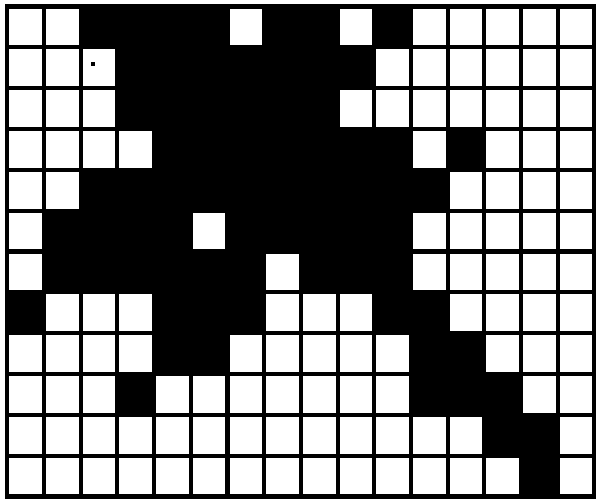
Huffman编码



S_6=01011

程长编码

- RLE : run length encoding
- 用一长度序列表示图像或位平面的每一行，这些长度描绘了对黑色和白色像素的连续行程
- 是传真编码的标准压缩方法

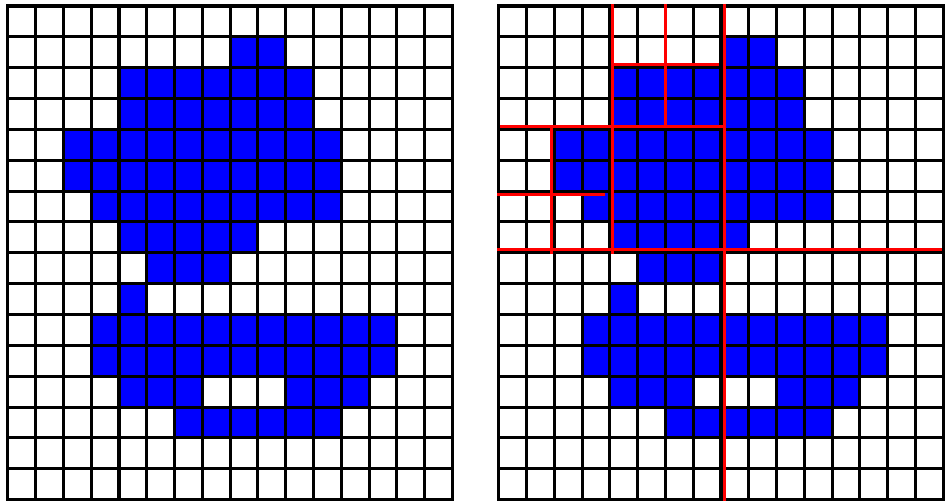


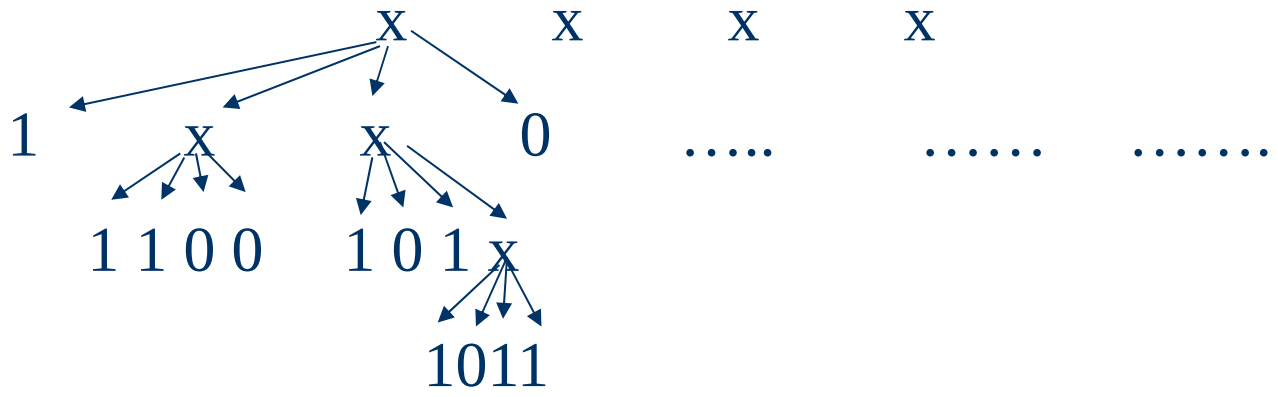
1,2, 0,4, 1,1, 0,2, 1,1, 0,1, 1,5

1,3,0,7,1,6

….

4-branch tree





(1(1100)(101(1011))0) (…)(…)(…)

预测编码

下一个像点值为前n个像点值x_k的组合,a_k是预测系数

$$\hat{x}_{i+1} = \sum_{k=i-n}^{i} a_k x_k$$

设 e_{i+1}是x_{i+1}与$\hat{x}_{i+1}$ 之间的预测误差

$$e_{i+1} = x_{i+1} - \hat{x}_{i+1}$$

- 对误差编码,而不是对像素点值编码,编码法有
  - DM (Delta Modulation)
  - 差分脉冲编码调制DPCM (Differential Puls Code Modulation)-JPEG

预测编码

- DPCM编码
  - 简单，图像质量影响小，但压缩比小（2:1）
  - 例，当前灰度值x的预测方法有8种

C	B
A	X

方法	预测值 $\hat{x}$	方法	预测值 $\hat{x}$
0	非预测	4	A+B−C
1	A	5	A+(B−C)/2
2	B	6	B+(A−C)/2
3	C	7	(A+B)/2

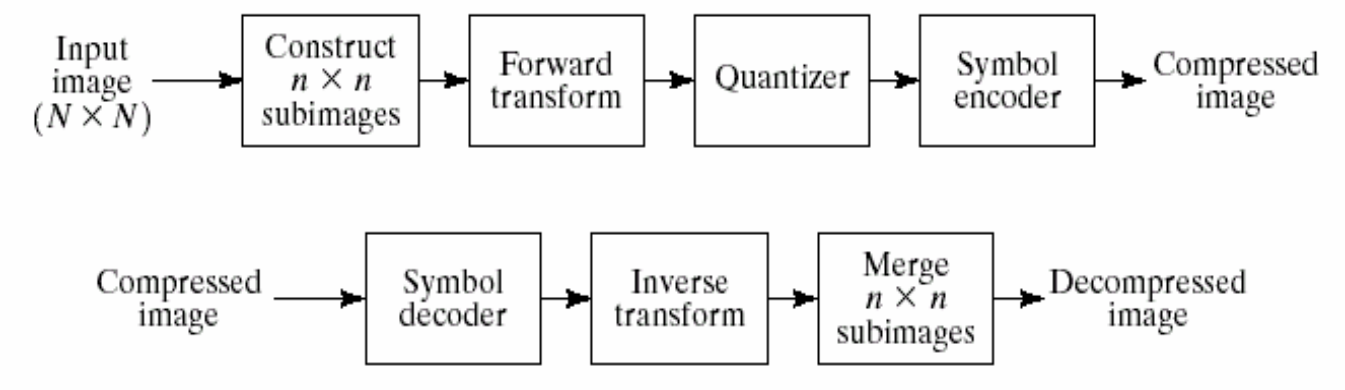
有损压缩

- 牺牲图像复原的准确度以换取压缩能力的增加
- 如果产生的失真可以容忍，则压缩能力的增加是有效的
- 类型
  - 有损预测编码：直接对像素在图像空间进行操作，称为空域方法
  - 变换编码：基于图像变换的编码方法，称为频域方法

变换编码

- 基于变换编码系统

子图分解　　变换　　量化　　编码



变换编码

- 考虑大小为N×N的图像$f(x,y)$，该图像的正向离散变换$T(u,v)$表示为

$$T(u,v)=\sum_{x=0}\sum_{y=0} f(x,y)g(x,y,u,v)$$

$u, v=0,1,2,\ldots,N-1$。

- 给定$T(u,v)$，$f(x,y)$可用离散反变换得到

$$f(x,y)=\sum_{u=0}\sum_{v=0} T(u,v)h(x,y,u,v)$$

$x, y=0,1,2,\ldots,N-1$

- $g(x,y,u,v)$和$h(x,y,u,v)$分别称为正向和逆向变换核函数。

变换编码

- DFT变换（离散傅里叶变换的简化版本（M=N））

$$g(x,y,u,v) = \frac{1}{N^2} e^{-j2\pi(ux+vy)/N}$$

$$h(x,y,u,v) = e^{j2\pi(ux+vy)/N}$$

- WHT变换（Walsh-Hadamard变换）

$$g(x,y,u,v) = h(x,y,u,v) = \frac{1}{N}(-1)^{\sum_{i=0}^{m-1}\lfloor b_i(x)\,p_i(u) + b_i(y)\,p_i(v)\rfloor}$$

变换编码

- 离散余弦变换DCT

For N*N block in image

$$S(k_1,k_2)=\sqrt{\frac{4}{N^2}}C(k_1)C(k_2)\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}s(n_1,n_2)\cos(\frac{\pi(2n_1-1)k_1)}{2N})\cos(\frac{\pi(2n_2-1)k_2)}{2N})$$

$k_1,k_2,n_1,n_2,=0,1,...N-1$ And $C(k)=\begin{cases}1/\sqrt{2};k=0\\1;\text{其它}\end{cases}$

It is fast implementation algorithm

DCT

- $N=2^3$ or 2^4 *(8X8 or 16X16)*
- DCT偶对称特性变换步骤可分离,块边沿连续
- DCT仅需进行实部计算

$$s(n_1,n_2)=\begin{bmatrix} 183 & 160 & 94 & 153 & 194 & 163 & 132 & 165 \\ 183 & 153 & 116 & 176 & 187 & 166 & 130 & 169 \\ 179 & 168 & 171 & 182 & 179 & 170 & 131 & 167 \\ 177 & 177 & 179 & 177 & 179 & 165 & 131 & 167 \\ 178 & 178 & 179 & 176 & 182 & 164 & 130 & 171 \\ 169 & 180 & 180 & 179 & 183 & 169 & 132 & 169 \\ 179 & 179 & 180 & 182 & 183 & 170 & 129 & 173 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix}$$

8*8 block in a image

从每个元素减去128 后,其DCT系数(就近取整)为

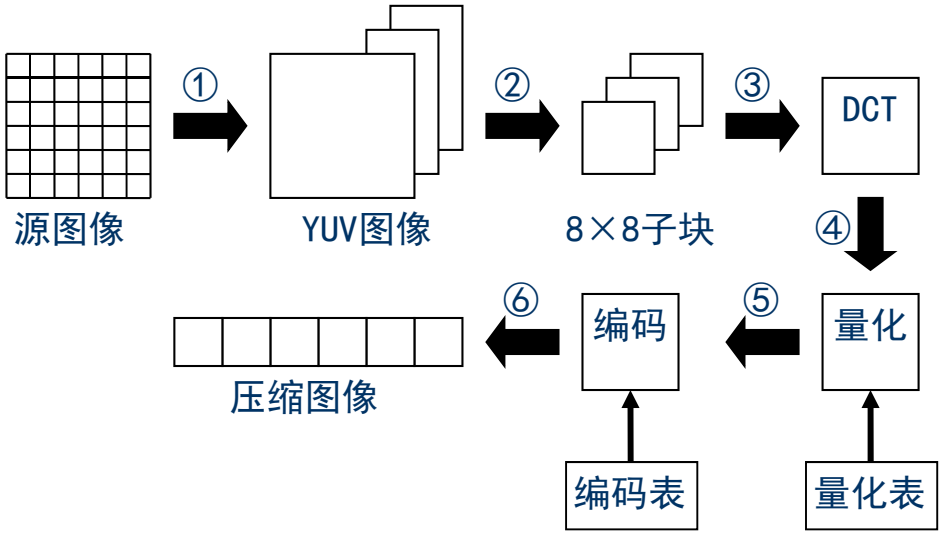
bigger

$$S(k_1,k_2)=\begin{bmatrix} 313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\ -38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\ -20 & -17 & 10 & 33 & 21 & -1 & -16 & -9 \\ -10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\ -6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\ 2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\ 4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\ 3 & 1 & 0 & -4 & -2 & -1 & 3 & 1 \end{bmatrix}$$

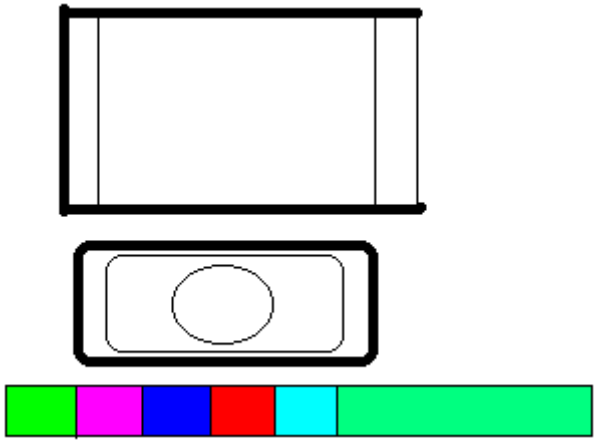
smaller

JPEG

- JPEG : Joint Photographic Experts Group, ISO/IEC 联合图片专家组

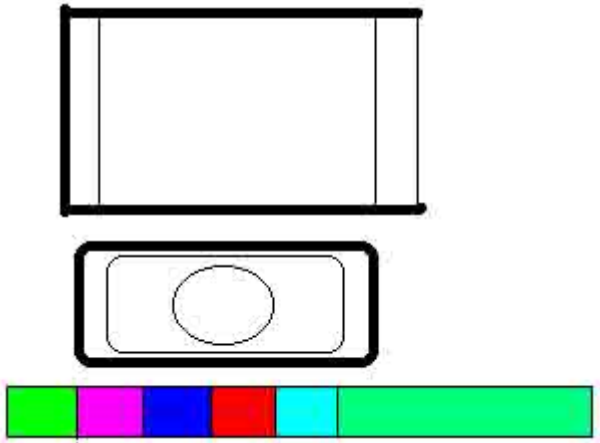


bmp

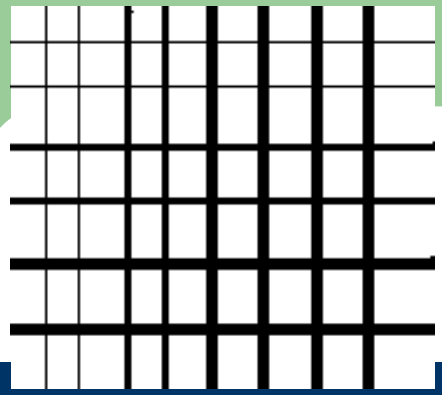


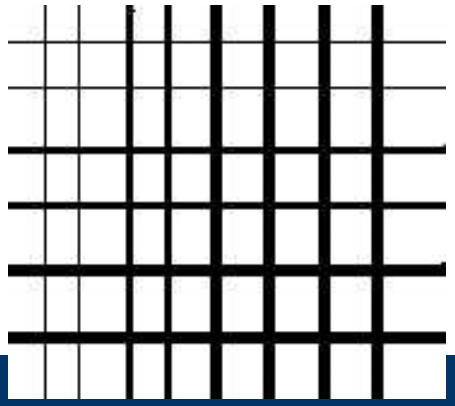
270kb

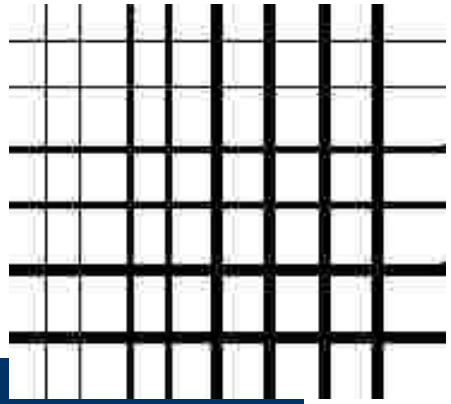
jpg



11kb



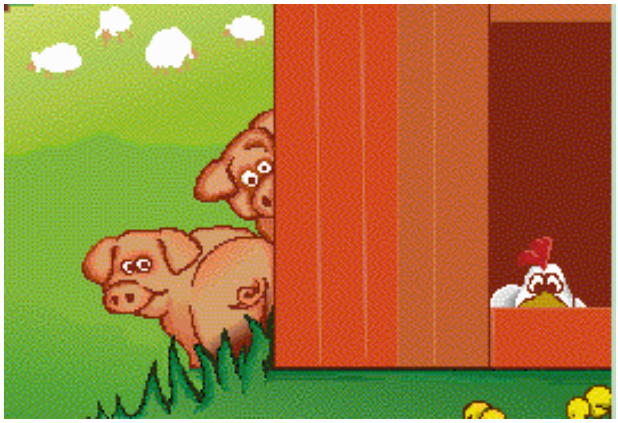


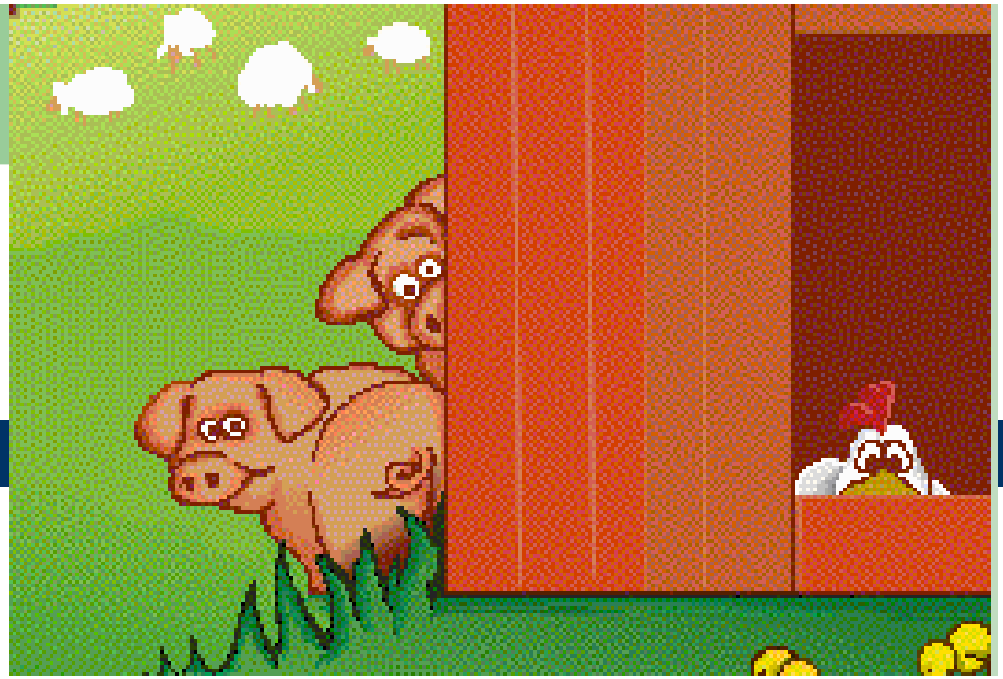


100% 14k

50% 7k

10% 4k





Bmp 49k

Jpg 7k

Jpg 3k