图像压缩简介

**冗余**

**空间冗余**

1920\*1080\*255 的图像



[105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 ……. 105]

第一行像素点的灰度表示 存储为 1920 个 105 占用1920单位的空间

压缩为： [105 1920] 占用2-3单位的空间

**时间冗余**

主要针对视频：

视频中相邻帧的背景大多相同，

相邻帧记录了相邻时刻的同一场景画面称为时间冗余。

**视觉冗余**

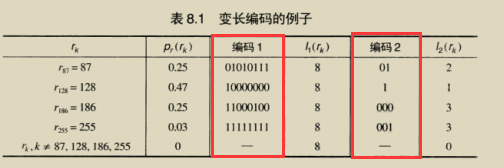
人类的视觉系统由于受生理特性的限制，

1. 对于图像场的注意是**非均匀**的，

2. 人对**细微的颜色差异感觉不明显**。

人类视觉的一般分辨能力为**26灰度等级**，而**一般的图像的量化采用的是28灰度等级**，即存在视觉冗余

**编码冗余**



**类似于 编码字符集**

**utf-8**

一种变长的编码方案，使用 1~6 个字节来存储；

—— 冗余量最小

**utf-16**

一种**固定长度**的编码方案，不管字符编号大小，始终使用 4 个字节来存储

—— 冗余量很大

**utf-32**

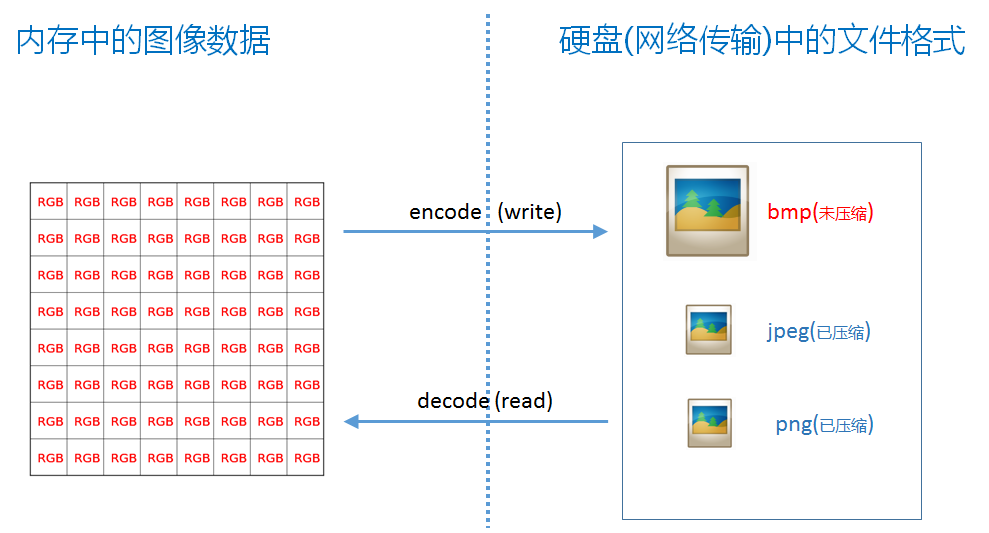
介于 UTF-8 和 UTF-32 之间，使用 2 个或者 4 个字节来存储，长度**既固定又可变**

—— 冗余量一般

**过程**

**编解码**

编码器（压缩） => 传输 => 解码器（显示）



**压缩分类**

**有损压缩**

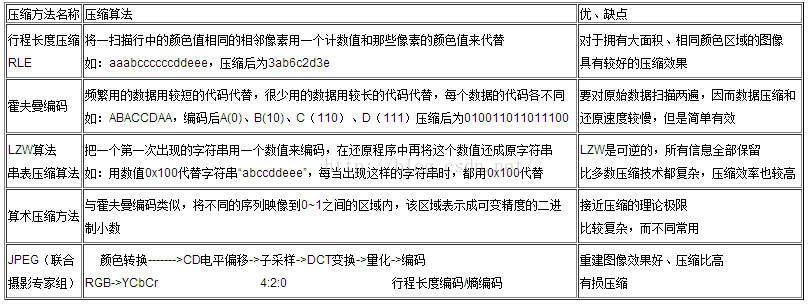
利用人眼视觉特性去除冗余信息和**对视觉不敏感的细节**，以尽可能少的byte来表示原图。

压缩后的数据重构为图像，和原图对比会存在一定的误差（失真）。

**无损压缩**

完全利用图像的冗余进行压缩，

可以**完全恢复图像**，不引起失真。

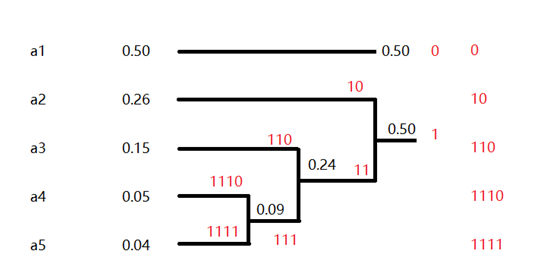


**哈夫曼（Huffman 霍夫曼）编码原理：**

概率高的符号用短码，概率低的符号用长码。

需要考虑符号出现的概率

原始符号（像素信息）—— 出现概率 —— 树（码表） —— 2叉 —— 编码结果



**LZW编码：**

不需要考虑符号出现的概率

需要初始码表

压缩生成扩展码表

传输不需要传出扩展码表

255,2,34,255,2,34,2,2,34

标号集

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | prev | current | entry(p+c) | has | output | tag |
| 1 |  | 255 | ( ,255) |  |  |  |
| 2 | 255 | 2 | (255,2) | n | 255 | 256 |
| 3 | 2 | 34 | (2,34) | n | 2 | 257 |
| 4 | 34 | 255 | (34,255) | n | 34 | 258 |
| 5 | 255 | 2 | (255,2) | y |  |  |
| 6 | 256 | 34 | (256,34) | n | 256 | 259 |
| 7 | 34 | 2 | (34,2) | n | 34 | 260 |
| 8 | 2 | 2 | (2,2) | n | 2 | 261 |
| 9 | 2 | 34 | (2,34) | y |  |  |

标号集为（0~255，256-(255,2)，257-(2,34)，258-(34,255)，259-(256,34)，260-(34,2)，261-(2,2)

**算术压缩：**

需要字符出现的概率

例如： CADACDB

过程：

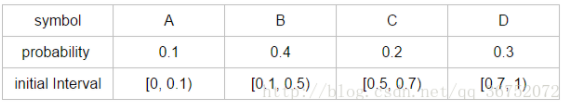
那么我们首先读入信号：C——因为C在最初始的间隔中是[0.5, 0.7)，所以读入C之后我们的编码间隔就变成[0.5, 0.7)了；

    紧接着，我们读入的是A，A在初始区间内是占整个区间的前10%，因此对应这个上来也是需要占这个编码间隔的前10%，因此编码区间变为：[0.5, 0.52)了；

    再然后是D，因为D占整个区间的70% ~ 100%，所以也是占用这个编码区间的70% ~ 100%，操作后的编码区间为[0.514, 0.52)

    ……

    直到最后将信号量全部读出。



编码：



编码结果为：0.5143876

解码：



**JPEG压缩：**

YUV 中 Y 表示明亮度（Luma），U 和 V 表示色度（Chrominance）和浓度（Chroma），UV 分量同时表示色差

**协议**

不同的编码解码协议 => 不同的格式

**常见格式：**

BMP格式 （BitMap picture）

特点：通常未压缩

24位/32位： RGB-24位 RGBA-32位

**数据结构对齐**：

CPU 每次从内存中读取并处理数据块（chunk），且通常为 32 比特（4 字节）或32的倍数。**因此，为了提升读取效率，位图每行的数据（字节）都需要是 4 的倍数——**指图像的行。如 24位对应为3个字节表示一个像素，如某图像一行有10个像素，则这一行需要 30个字节，不是4的倍数，所以会在行尾用0补位，补到32个字节。

JPEG格式

图像金字塔简介

图像水印的原理