# 第二章 多媒体数据编码基础

## 数字音频编码

## 数字图像编码

## 数字视频编码

## 常用的数据压缩技术

## 多媒体数据转换

# 2.1 数字音频编码

# **音频的基本特性**

## 声音是由振动的声波所组成（模拟音频），在任一时刻*t*，声波可分解为一系列正弦波线性叠加：

## *f*(*t*)= Σ*An*sin(*nωt*+*φn*)

## 其中, ω称为基频或基音，它决定声音的尖锐程度；*nω*称为*ω*的*n*次谐波分量或称为泛音，与声音的音色（是否有噪音）有关; *An*是振幅,表示声音的强弱（高低）；*φn*是*n*次谐波的初相位。

# 音频数字化

## 以数字方式表示声音。计算机是用声卡(包含ADC和DAC)来录制与播出数字化声音的。计算机对模拟的声音主要通过采样、量化、编码等步骤产生一系列数字化的声音数据。

## 事实上, 声波按频率可分为4类, 其中多媒体系统仅处理人类的听力所接受的频率范围的声音，我们称之为音频, 这个频率范围的声波称之为声音信号。

# 声音的频率分类



## 通过规则时间间隔测出音波振动幅度从而产生一系列声音数据。这种测出数据方法称之为**采样**,一秒内采样次数叫**采样率**。采样率需要满足奈奎斯特采样定理：采样频率应是信号频率的2倍以上。

## 采样的离散音频数据要转换成计算机能够表示的数据范围,这个过程称之为**量化 。**

## **量化后数字音频存储量计算公式**

## **音频数据存储量(字节)=采样率(Hz)×量化位数(位)×声道数×音频长度(秒)/8**

# MIDI （音乐设备数字接口）

## 1980年制定一项工业标准,目的是让音乐可以在不同的音乐设备间传输。

## MIDI提供了计算机外部的（数字）电子乐器与计算机之间的连接器接口。这种连接接口定义物理连接及电子乐器之间沟通的协议。

## MIDI也定义音频的形态与存储的方法。MIDI音频是以消息的方式而非波形的方式组成。

## MIDI有三种连接器(In、Out、Thru)。

## In为输入, Out为输出,而Thru是用来扩充MIDI与其它设备连接的。

## 计算机可以连接电子琴、电吉他、电萨克斯等。

## 音序器：计算机中记录、编辑、播放MIDI的程序。

## 音序器可以将音乐等声音以一种序列来储存。所谓序列便是一连串的音符加上系统事件的命令。

## MIDI适配器是用来改变频道、路径与按键的。当电子琴的键盘与一般的MIDI规格不一致时可以经由适配器来修正使两者一致。

# MIDI音频文件

## MIDI音频文件是一串时序命令,它记录乐器的行为。

## 命令消息分为频道消息(频道声音消息、频道模式消息)和系统消息(系统实时消息、系统通用消息与系统专用消息)。它是以某种乐器的发声为其数据记录的基础。

## 它的文件占用很少存储器空间,且可以做细部的修改，如修改节拍等。其声音效果不会因改变节拍而变调。

## MIDI不适合编制口语旁白的音频。

## 数字音频可从麦克风、录音带、CD、电视及其它来源获取。它把声音转换成存储器中的数字信息。

## 数字音频较为稳定,容易保持一致性,音频品质也较易获得保证。

## 缺点是记录非常详尽,数据量极大,数字音频文件较MIDI音频文件大出200倍以上。要修改数字音频细节非常困难,大大地增加了CPU的负担。

## 它可以适合任何一种音源,包括人的口语在内,故大多数节目仍采用这种音频。

# 3D音频

## 3D音效可分为以下几类：

## （1）扩展式立体声（使用声音延迟技术对传统的立体声（2D）进行处理，产生3D效果）。

## （2）环绕立体声（对多声道数据进行编码，如：AC3,DTS）。

## （3）交互式音效(音效随听者的位置移动发生变化)。

## 支持3D音频API种类:

## DirectX DirectSound 3D; Aureal 3D; EAX; Sensaura; Qsound; 杜比AC-3; 数字化影院系统DTS

# 2.2 数字图像编码

# 数字图像的种类：分辨率，1024x768

1、黑白图像（2）

2、灰度图像（256）

3、彩色图像（16位

24位，32位）

# 2.2.1 彩色空间（彩色图像）

## 常见模型**：**

### RGB彩色空间

### HSI彩色空间

### YUV彩色空间

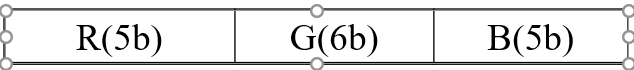
### YIQ彩色空间

# RGB彩色空间

## R、G、B是最基本的彩色表示模型,也是计算机系统中所使用的彩色模型。

### RGB5:6:5方式（16位增强色）

### 用2个字节表示一个象素，具体位分配。



## 因为人眼对绿色更敏感，因此，绿色用6个比特来存储

## RGB5:6:5方式

### RGB8:8:8方式（24位真彩色） ，32位真彩色

### R、G、B三个分量各占一个字节。

# HSI彩色空间

## 这种模型中, 用H(Hue,色调)、S(Saturation,饱和度)、I(Intensity,光强度)3个分量来表示一种颜色, 这种表示更适合人的视觉特性。

# YUV彩色空间 （广播电视领域广泛采用）

## Y为亮度信号,U、V是色差信号（基色信号减亮度信号即为色差信号）(B-Y,R-Y)，Cb，Cr。

## PAL制式彩色空间即为YUV。优点是亮度和色差信号分离,容易使彩色电视系统与黑白电视信号兼容。

## 国际无线电咨询委员会根据实验认为采用双倍度采样4:2:2（YUV各分量的采样率之比）方案效果较好, 提出CCIR601标准。

## 变换公式（YUV<--->RGB）

## Y = 0.299\*R + 0.587\*G+ 0.114\*B;

## U =-0.169\*R - 0.332\*G+ 0.500\*B;

## V = 0.500\*R + 0.419\*G - 0.081\*B

# YIQ彩色空间

## 广播电视系统另一种常用的亮度与色差分离的模型。NTSC制式彩色空间即为YIQ。这里Y是亮度, I和Q共同描述图象的色调和饱和度。

## 变换公式(YIQ<--->RGB)

## Y =0.299\*R+ 0.587\*G+ 0.114\*B;

## I =0.211\*R - 0.523\*G+ 0.312\*B;

## Q =0.596\*R - 0.275\*G - 0.322\*B

# 2.2.2 数字图象文件格式

## TIF

## PCX

## GIF、TGA、BMP、DVI、JPEG等

# 2.2.2 数字图象文件格式

## BMP文件结构

## BMP文件格式是Windows系统中广泛使用的图像文件格式，属于图像数据未经压缩的图像文件（即图像中直接保存图像中像素点阵中每个像素的颜色值）。

# 2.2.2 数字图象文件格式

## BMP文件的数据按照从文件头开始的先后顺序分为四个部分：

## 1）bmp文件头(bmp file header)：提供文件的格式、大小等信息

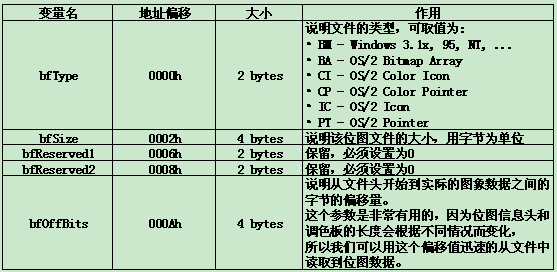
## 2）位图信息头(bitmap information)：提供图像数据的尺寸、位平面数、压缩方式、颜色索引等信息

## 3）调色板(color palette)：可选，如使用索引来表示图像，调色板就是索引与其对应的颜色的映射表

## 4）位图数据(bitmap data)：就是图像数据

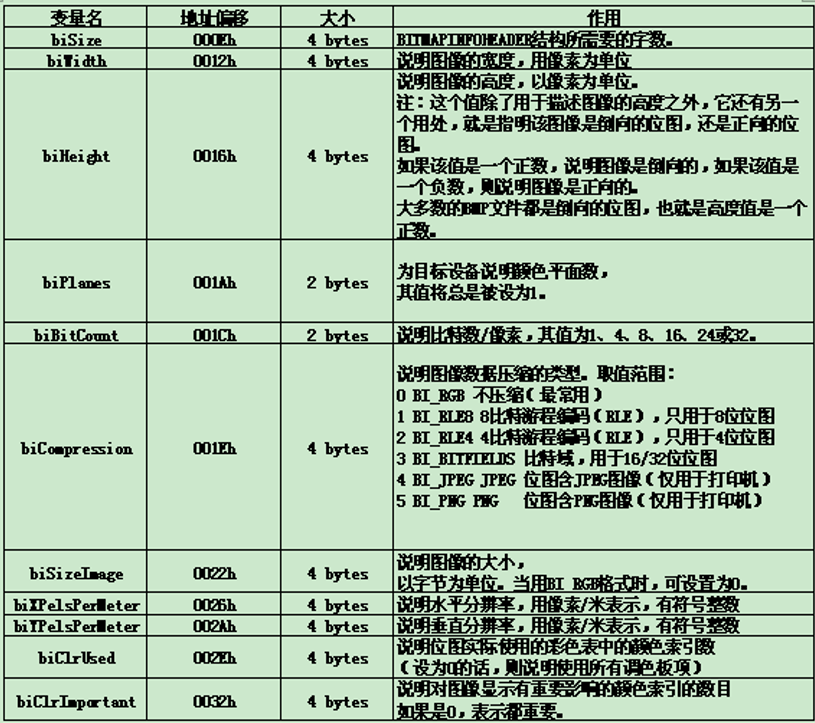
# 2.2.2 数字图象文件格式

## bmp文件头



# 2.2.2 数字图象文件格式

## bmp信息头



# 2.2.2 数字图象文件格式

## 调色板

## 调色板其实是一张映射表，标识颜色索引号与其代表的颜色的对应关系。

## 一般见到的图像以24位或32位图像，即R、G、B三种颜色各用8个bit来表示，这样的图像我们称为真彩色，这种情况下是不需要调色板的，也就是所位图信息头后面紧跟的就是位图数据了，而16位图像或256色图像是需要调色板的。

# TIF（tiff）文件格式

## 由美国Aldus Developer’s Desk和Microsoft制定

## 结构

### 文件头(8B)

### 参数指针表

### 参数数据表

### 图象数据

## 文件头

## 字节顺序(2B,表示存贮格式: II-Intel格式; MM—Motorola格式);

## 标记号(2B, 版本信息)，为固定值42,;

## 指向第一个参数指针表(4B)，相对于文件头的偏移量。

## 参数指针表

## 由每个长为12B参数块（描述图像的某一个属性）构成, 描述压缩种类、长宽、彩色数、扫描密度等参数。

## 较长参数(如调色板)只给出指针, 参数放在参数数据表中。其结构定义如下：

## *typedef struct {*

## *int tag-type（图像属性编号）;*

## *int number-size（该属性对应数据的类型）;*

## *long length（该属性对应数据的数量）;*

## *long offset（该属性数据相对于文件头的偏移量）;*

## *}TIF-FIELD;*

## 图象数据 按参数表中描述的形式按行排列（未压缩）

# PCX文件格式

## 由Z Soft公司最初制定

## 结构：

### 文件头(128字节)

### 数据部分(采用行程长度编码，压缩存储)

## 文件头结构定义

## ***typedef struct{***

## ***char manufacture; /\*always 0xa0\*/***

## ***char version；***

## ***char encoding; /\*always 1\*/***

## ***char bits-per-pixel; /\*color bits \*/***

## ***int Xmin, Ymin; /\* image origin \*/***

## ***int Xmax, Ymax; /\* image dimension \*/***

## ***int hres; /\* resolution values \*/***

## ***int vres;***

## ***char palette[48]; /\* color palette \*/***

## ***char reserved;***

## ***char color-planes; /\* color planes \*/***

## ***int bytes-per-line; /\* line buffer size \*/***

## ***int palette-type; /\* grey or color palette \*/***

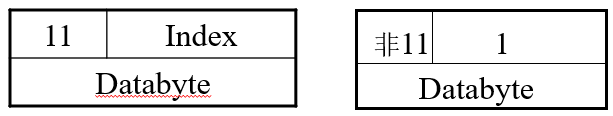
## ***char filler[58];***

## ***} PCXHEAD;***

## 其中Version若为5,文件内有个256色调色板（描述256中颜色的RGB分量值）,数据768字节,在文件最后。

## 文件体 对象素数据按行以字节为单位采用行程长度编码, 由包含Keybyte和Databyte的包组成。分2种情况:

## (1) 若Keybyte最高位为11, 则低6位为重复次数。但最多重复63次, 若再长重建一个包.



## PCX数据包的结构

## (2) 若Keybyte最高位不是11，那么该Databyte按原样写入图象文件。对一个字符的表示用长度为1的包。

# 2.3 **数字视频编码**

# 数字视频的结构

## 基本单位是帧

## 一帧即一幅图像，视频可以看做是按一定时间间隔连续播放的图像序列，这里没有考虑视频中包含的音频信息，实际上视频中还需包含音频信息，为讨论问题方便，我们这里只考虑视频中的图像数据。

## 若干同一场景的帧构成镜头（此处镜头为影视专业术语）

## 若干镜头构成情节

## 若干情节构成故事(节目)

# 国际视频标准（计算机领域）

## 计算机内能够直接处理的视频与数字电视系统中处理的视频其存储和处理方式是有区别的，原因在于，数字电视领域中一帧图像分为奇数场和偶数场显示，而计算机数字视频中的图像不需按此方式处理和显示，造成了计算机数字视频和广播电视领域数字视频的不同。

## 数字视频的标准制定机构有两个：ITU-T和ISO，其中 ITU-T发布的视频标准有H.261、H.262、H.263、H.263+、H.263++，其主要面向对象为数字广播电视领域，而ISO公布的MPEG系列标准有MPEG-1、MPEG-2、MPEG-4 和MPEG-7，并且计划公布MPEG-21，其主要应用领域为计算机。

# 国际视频标准（广播电视领域）

## **NTSC** 美国研制,是目前广泛使用的电视制式。它以525条横扫描线来组成一个屏幕帧,每秒30帧,其图象改变采用偶数线与奇数线相互交错更新的方式,造成视觉动态图象。

## **PAL** 中国、英国等国采用制式,W.Bruch1963年发明的,其基本原理类似于NTSC制式。以625条扫描线,每秒25帧,也是以奇偶数扫描线交错方式造成动态图象。

## **SECAM** 法国、俄罗斯等国采用制式。同样采用625条线和25帧, 但与NTSC和PAL相比, 其基础技术是采用频率调制, 传播方式也不同于以上两种。

# 2.4 常用数据压缩技术

# 2.4.1 数据压缩的基本原理

## 传统上用模拟方式表示声音和图象信息

## 易出故障，常产生噪音和信号丢失，且拷贝过程中噪音和误差逐步积累；

## 模拟信号不适合数字计算机加工处理。

## 数字化处理：巨大的数据量

## **采样定理:**

## 仅当采样频率≥2倍的原始信号频率时,才能保证采样后信号可被保真地恢复为原始信号。

## 采用8bit数字化（每个像素对应的比特数）,从而1秒钟电视信号的数据量约为99.2Mbits。

## 即约为100Mbps。650MB的CD-ROM仅能存约1分钟6秒钟的原始电视数据。若HDTV(1.2Gbps), 一张CD-ROM还存不下1秒的HDTV电视数据。

## 人说话的音频一般在20Hz到4KHz, 即人类语音的带宽为4KHz。依据采样定理, 设数字化精度为8b, 则1秒钟信号量为64Kbits。

## 因此, 人讲1分钟话的数据量为480KB。

## 数字化处理的关键问题－数据压缩

# 压缩的基础－数据冗余

## 空间冗余

## 时间冗余

## 信息熵冗余

## 结构冗余

## 知识冗余

## 视觉冗余

## 其它冗余

# 空间冗余

## 这是图象数据中经常存在的一种冗余。

## 在同一幅图象中,规则物体和规则背景的表面物理特性具有相关性,这些相关性的光成象结构在数字化图象中就表现为数据冗余。

# 时间冗余

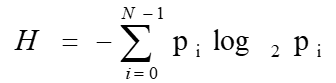
## 这是序列图象和语音数据中所经常包含的冗余。

## 图象序列中的两幅相邻的图象之间有较大的相关性,这反映为时间冗余。

## 人在说话时发音音频是一连续的渐变过程,而不是一个完全时间上独立的过程,因而存在时间冗余。

# 信息熵冗余

## 信息熵是指一组数据所携带的信息量,它定义为：



## *N*为数据类型数或码元个数, *Pi*为码元*yi*发生的概率.

## 为使信息编码单位数据量*d*接近于或等于*H*,应设：



## 其中*b(yi)*是分配给码元*yi*的比特数, 理论上应取*b(yi)=-log2Pi*. 实际一般取*b(y0)=b(y1)=…=b(yK-1)*.

## 例如, 英文字母编码码元长为7bit, 即*b(y0)=b(y1)= … =b(yK-1)=*7, 这样*d*必然大于*H*, 由此带来的冗余称为信息熵冗余或编码冗余。

# 结构冗余

## 有些图象从大域上看存着非常强的纹理结构，我们称它们在结构上存在有冗余.

## 例如布纹图象和草席图象

# 知识冗余

## 有许多图象的理解与某些基础知识有相当大的相关性。

## 例如,人脸的图象有固定的结构。比如说嘴的上方有鼻子, 鼻子的上方有眼睛, 鼻子位于正脸图象的中线上等等。

## 这类规律性的结构可由先验知识和背景知识得到, 我们称此类冗余为知识冗余。

# 视觉冗余

## 人类视觉系统对于图象场的任何变化,并不是都能感知的。

## 例如,对于图象的编码处理时,由于压缩或量化截断引入了噪声而使图象发生了一些变化,如果这些变化不能为视觉所感知,仍认为图象足够好。

## 事实上人类视觉系统一般分辨能力约为26灰度等级,而一般图象量化采用28灰度等级,这类冗余我们称为视觉冗余。

# 其他冗余

## 例如由图象的空间非定常特性所带来的冗余。

# 2.4 常用的数据压缩技术

## 根据解码后数据与原始数据是否完全一致，数据压缩方法划分为两类：

#### **可逆编码(无失真编码)** 解码图象与原始图象严格相同，压缩大约在2:1到5:1之间。如Huffman编码、算术编码、行程长度编码等。

#### **不可逆编码(有失真编码)** 还原图象与原始图象存在一定的误差，但视觉效果一般可以接受，压缩比可以从几倍到上百倍来调节。常用的有变换编码和预测编码

## 根据压缩的原理分类：

#### **预测编码** 利用空间中相邻数据的相关性来预测未来点的数据。差分脉冲编码调制(DPCM)和自适应差分脉冲编码调制(ADPCM)。

#### **变换编码** 将图象时域信号变换到频域空间处理。时域空间有强相关信号, 反映在频域上是某些特定区域内能量集中, 从而实现压缩. 正交变换如离散余弦变换, 离散付立叶变换和Walsh-Hadamard变换.

#### **量化与向量量化编码** 为了使整体量化失真最小, 就必须依照统计的概率分布设计最优的量化器。已知最优量化器是Max量化器。对象元点进行量化时, 也可以考虑一次量化多个点的向量量化。

#### **信息熵编码** 根据信息熵原理,让出现概率大的用短的码字表达,反之用长的码字表示。最常见的方法如Huffman编码、Shannon编码以及算术编码。

#### **子带编码** 将图象数据变换到频域后,按频域分带,然后用不同的量化器进行量化,从而达到最优的组合。或者分步渐近编码,随着解码数据的增加,图象逐渐清晰。离散小波变换

#### **模型编码** 编码时首先将图象中边界、轮廓、纹理等结构特征找出来,保存这些参数信息。解码时根据结构和参数信息进行合成,恢复出原图象。具体方法有轮廓编码、域分割编码、分析合成编码、识别合成编码、基于知识的编码、分形编码等。

# 2.4.1 预测编码

## 线性预测-DPCM

## 基本原理是基于图象中相邻象素之间具有较强的相关性。每个象素可根据已知的前几个象素来作预测。因此在预测编码中，编码和传输的并不是象素采样值本身，而是这个采样值的预测值与其实际值之间的差值

## 非线性预测（不讨论）

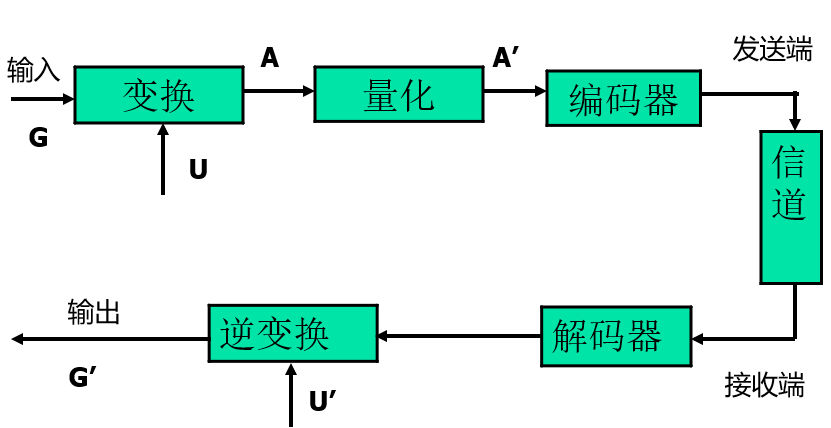
# 2.4.2 变换编码

## 输入图象G经正交变换U变换到频域空间,象素之间相关性下降,能量集中在变换域中少数变换系数上,已经达到了数据压缩的效果。

## 对变换系数A中那些幅度大元素予以保留,其它数量多的幅度小的变换系数,全部当作零不予编码,再辅以非线性量化,进一步压缩图象数据。

## 由于量化器存在,量化后变换系数A′和A间必然存在量化误差,从而引起输入图象G和输出图象G′间存在误差。图中U′是U的逆变换。

# 变换编码原理框图



## Karhunen-Loeve变换是**最佳变换**,但它的计算比较复杂。

## 实际应用中采用了一些准最佳变换如DCT,DFT和WHT等。

# 2.4.3 信息熵编码

## 又称为统计编码,它是根据信源符号出现概率的分布特性而进行的压缩编码。

## 基本思想: 在信源符号和码字之间建立明确的一一对应关系，以便在恢复时能准确地再现原信号,同时要使平均码长或码率尽量小。

## 如Huffman编码、算术编码。

# Huffman定理

## **定理** 在变长编码中,对出现概率大的信源符号赋于短码字,而对于出现概率小的信源符号赋于长码字。如果码字长度严格按照所对应符号出现概率大小逆序排列,则编码结果平均码字长度一定小于任何其它排列方式。

## Huffman定理是Huffman编码的理论基础

# 实现步骤

## (1) 将信源符号按概率递减顺序排列；

## (2) 把二个最小概率相加作为新符号的概率, 并按(1) 重排；

## (3) 重复(1)、(2), 直到概率为1；

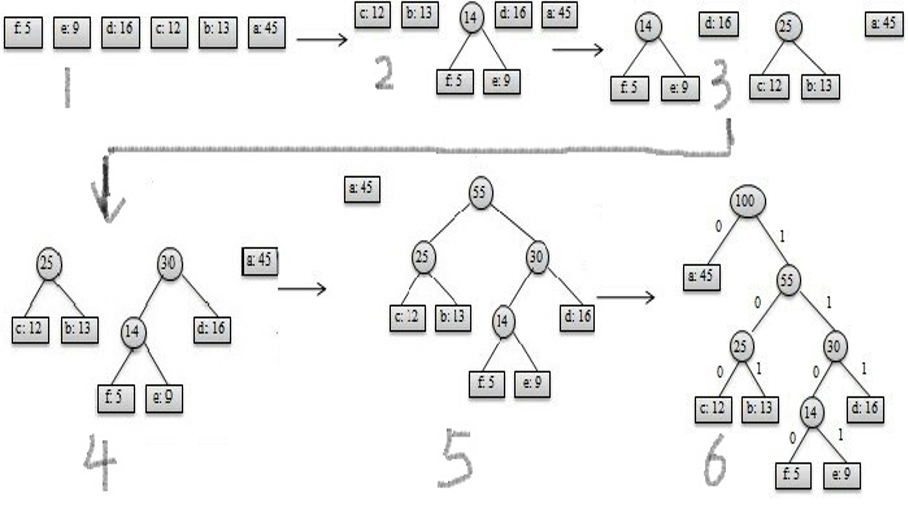
## (4) 在每次合并信源时, 将合并的信源分别赋“0”和“1”(如概率大的赋“0”,概率小的赋“1”)；

## (5) 寻找从每一信源符号到概率为1处的路径,记录下路径上的“1”和“0”；

## (6)写出每一符号的“1”、“0”序列(从树根到信源符号节点)。

# Huffman编码

# 已知一字符串中字符abcdef的出现次数如下，请确定该字符串的Huffman编码



## 上述编码的码字为:

## a:0

## b:101

## c:100

## d:111

## e:1101

## f:1100

# 算术编码

## 六十年代初,Elias提出了算术编码概念。

## 1976年, Rissanen和Pasco首次介绍了它的实用技术。其基本原理是将编码的信息表示成实数0和1之间的一个间隔(Interval),信息越长,编码表示它的间隔就越小,表示这一间隔所需的二进制位就越多。

# 算术编码举例

## 采用固定模式符号概率分配如下：

## 字符: a e i o u

## 概率: 0.2 0.3 0.1 0.2 0.2

## 范围:［0,0.2) ［0.2,0.5) ［0.5,0.6)［0.6,0.8)［0.8,1.0)

## 编码数据串为eai。令high为间隔的高端, low为低端，range为间隔的长度, rangelow为编码字符分配的间隔低端, rangehigh为编码字符分配的间隔高端。

## 初始high=1,low=0, range=high-low,

## 一个字符编码后新的low和high按下式计算：

## low=low+range×rangelow；

## high=low+range×rangehigh。

## (1) 在第一个字符e被编码时, e的rangelow=0.2, rangehigh=0.5, 因此:

## low=0+1×0.2=0.2

## high=0+1×0.5=0.5

## range=high-low=0.5-0.2=0.3

## 此时分配给e的范围为［0.2, 0.5)

## (2) 第二个字符a编码时使用新生成范围[0.2,0.5), a的rangelow=0, rangehigh=0.2, 因此:

## low=0.2+0.3×0=0.2

## high=0.2+0.3×0.2=0.26

## range=0.06

## 范围变成[0.2, 0.26)

## (3) 对下一个字符i编号, i的rangelow=0.5，rangehigh=0.6,range=0.06, 则：

## low=0.2+0.06×0.5=0.23

## high=0.2+0.06×0.6=0.236

## **结果**：用[0.23, 0.236)表示数据串eai,也就是从此区间中入一个数，作为编码进行传输，如果解码器知道最后范围是[0.23, 0.236),它马上可解得一个字符为e, 然后依次得到唯一解a、i, 最终得到eai。

## 解码过程：

## 解码的过程其实和编码的过程恰恰相反，解码是给定一个[0, 1)中的浮点数，通过解码操作之后就能完全获得原始的信号串。

## ·ÖÎöÉÏÊöÊµÀý£¬¼ÙÉèÊÜµ½Êý¾ÝÎª£º0.235则：

## 1）因0.235大于0.2，小于0.5,

## 故可判定第一个字符为e,然后计算(0.235-0.2)/(0.5-0.2)=0.117

## 2)因0.117大于0，小于0.2，

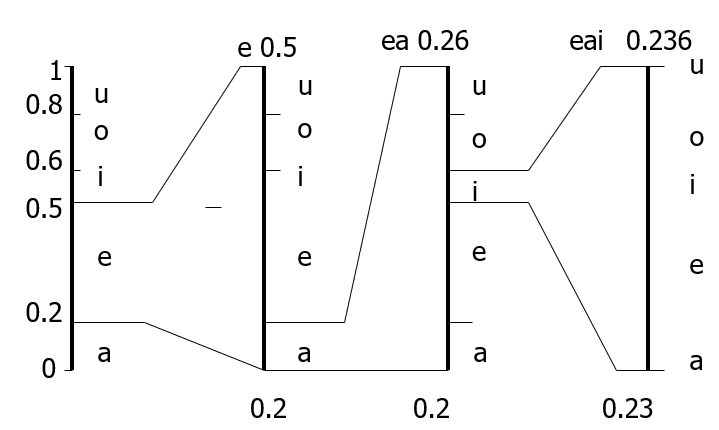
## 故可判定第二个字符为a,然后计算(0.117-0)/(0.2-0)=0.583

## 3)因0.583大于0.5，小于0.6，

## 故可判定第一个字符为i

## 实际上，此过程可以一直重复下去，所以解码端需知道发送的字符个数才可以正确解码。

# 算术编码过程表示



# 算术编码的特点

## 不必预先定义概率模型,自适应模式具有独特的优点;

## 信源符号概率接近时,建议使用算术编码,这种情况下其效率高于Huffman编码(约5%)。JPEG扩展系统采用。

# 2.5 多媒体数据转换

## 不同媒体表示不同的信息表示方式。研究媒体之间转换十分有意义。

## 有些媒体之间的转换是非常困难的事情,需要研究人类本身对各种媒体理解原理和解释过程。

## 有些媒体之间的转换则相对容易,几乎不用做什么工作。

# 部分媒体的转换关系

