

电视原理

第八章 数字视音频压缩编码

石东新

理工学部·信息工程学院·广播电视工程系

第一节 数字视频压缩编码概论

数字电视SDI（4：2：2编码、10比特量化）信号的数码率为：
 $13.5 \times 10 + 2 \times 6.75 \times 10 = 270\text{Mbps}$

一帧SDTV图像的数据量约为10.8Mb，要记录10分钟的电视节目就需要162Gb的存储器容量；而一帧HDTV图像的数据量为59.4Mb，记录10分钟的节目需要891Gb的存储器容量。

数字视频格式	色度格式	像素格式	量化比特数	无压缩有效码率
HDTV	4:2:2	1920×1080/50I	10	1037Mbps
HDTV	4:2:0	1920×1080/50I	8	622 Mbps
SDTV	4:2:2	720×576/50I	10	207Mbps
SDTV	4:2:0	720×576/50I	8	124Mbps
CIF	4:2:0	352×288/50I	8	30 Mbps

结论：
若不降低其数据量和数码率，就无法在普通的存储设备上实现数字电视信号的存储。

一、数字视频压缩机理

压缩过程：去除图像中与信息无关或对图像质量影响不大的部分，即冗余部分。

1、数据冗余

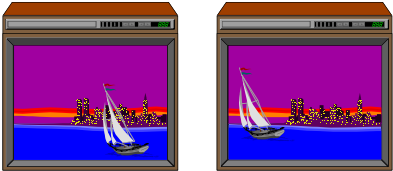
(1) 空间冗余

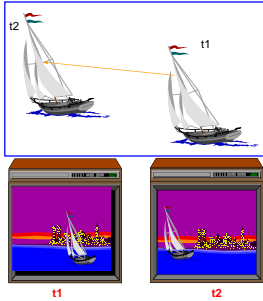
在同一幅图像中规则的物体和规则的背景都具有很强的相关性。



(2) 时间冗余

电视图像序列中相邻的两幅图像之间有较强的相关性。





(3) 结构冗余和知识冗余
图像从大面积上看存在着非常强的纹理结构，称之为结构冗余。人类对许多图像的理解是根据某些已知的知识，例如人脸的图像有固定的结构，这些规律性的结构可由先验知识和背景知识得到，称之为知识冗余。



2. 视觉冗余

人眼的视觉效果是图像质量的最直接也是最终的检验标准，对于人眼难以识别的数据或对视觉效果影响甚微的数据，都可认为是多余的数据，可以省去。这些多余部分就是视觉冗余。

- (1) 亮度辨别阈值
- (2) 视觉对不同图像内容的空间分辨力
- (3) 人眼亮色分辨力

二、 数据压缩编码方法与性能指标

压缩比= $\frac{\text{输入数据流码率}}{\text{输出数据流码率}} \geq 1$

1. 压缩编码分类

- (1) 无损压缩编码（可逆压缩编码或信息保持编码）
可使接收端解码后的信息量与发送端原信息量完全相同，因此再现的图像也与原图像严格一致，也即压缩后的图像完全可以恢复或无损伤。
- (2) 有损压缩编码（不可逆压缩编码或信息非保持编码）
编码过程中会损失一部分信息，因此接收端解码后再现的图像质量会比原图像质量有所降低，即压缩后图像有损伤，不能完全恢复。但如果视觉上能够接受甚至觉察不出质量的降低，则这种压缩就是可行的。

传统的图像编码压缩算法	1. PCM
	2. DPCM
	3. 变换编码
	Fourier 变换
	Walsh Hadamard 变换
	Hear 变换
	DCT 变换
新一代图像编码压缩算法	K-L 变换
	Slant 变换
	4. 熵编码
	游程编码
	Huffman 编码
	算术编码
	5. 矢量量化
新一代图像编码压缩算法	6. 子带编码
	7. 塔型编码
	1. 模型基编码
	2. 分形编码
新一代图像编码压缩算法	3. 小波变换
	4. 神经网络

在数字电视视频压缩编码中，通常采用混合编码方案，包括：

- 基于运动估计和补偿的预测编码（DPCM）；
- 正交变换编码（DCT）；
- 游程编码；
- 熵编码等。

2. 压缩编码性能指标——评价压缩系统的压缩质量

(1) 图像质量评价

通常采用**主观评价**和**客观评价**方法**评价经过压缩编解码后的视频图像质量**。

主观评价是观察者对被视视频质量直接打分。

客观评价是通过一定的算法模型对视频图像进行评价。

(2) 压缩编解码的复杂度

其中包括算法的运算量、存储量、编码器和解码器的复杂度等等。如果压缩编码的复杂度过高，难以实时处理。

(3) 压缩解码后带来的延时量

如果延时过大，会造成明显的可察觉到的声音和画面不同步现象，需要声音通道上加入相应的延时器加以调整。

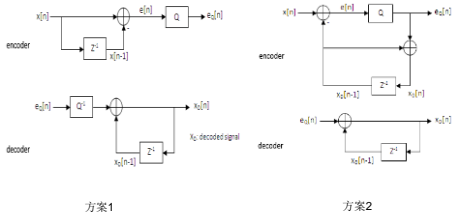
第二节 预测编码

预测编码又称**差分脉冲编码调制 (DPCM)** -----

不直接传送图像样值本身，而是对被编码的样值与参考样值间的差值 e_n 进行编码、传送。

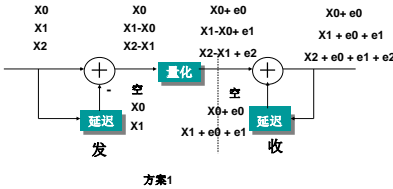
- 一、预测编码的基本原理：
- 相邻样本之间存在较强的相关性。
 - 差值信号的方差更小，具有更小的动态范围。
 - 编码和传输的并不是采样值本身，而是这个采样值的**预测值**与其实际值之间的**差值**。

一、 预测编码基本原理

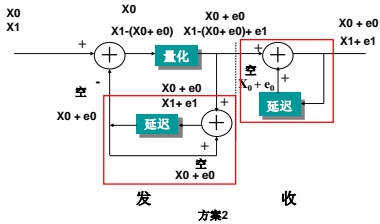


DPCM was invented by **C. Chapin Cutler** at **Bell Labs** in 1950; his patent includes both the methods.

(1) 方案1



(2) 方案2 (实际的预测编码方法)



信源相邻样值间的相关性：

帧内相邻像素间**空间相关性**——**空间冗余**——帧内 (Intra) 预测

帧间**时间相关性**——**时间冗余**——帧间 (Inter) 预测

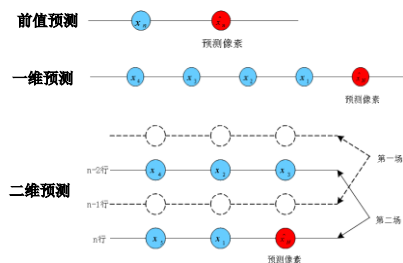
二、 帧内预测

帧内预测是基于同一帧内相邻像素存在很强的相关性。压缩图像的空间冗余。几种常用的线性预测方案

前值预测

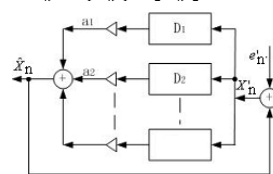
一维预测

二维预测



■预测器的设计预测公式为:

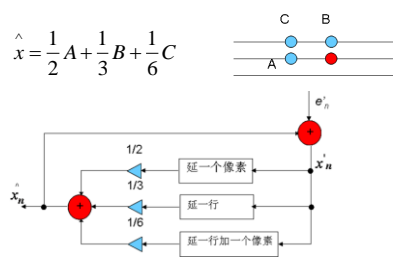
$$\hat{x}_n = a_1 x'_{n-1} + a_2 x'_{n-2} + \dots$$



图中的D1,D2,...为延迟器

■ 例子：设计一个线性预测器，其预测公式为：

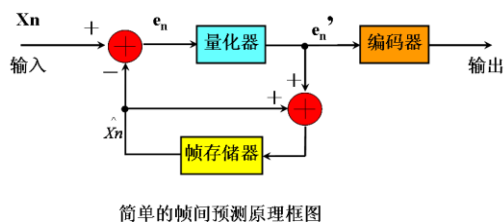
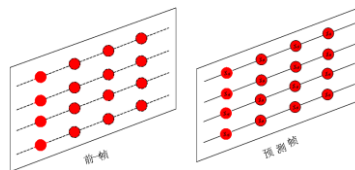
$$\hat{x} = \frac{1}{2}A + \frac{1}{3}B + \frac{1}{6}C$$



三、 帧间预测（三维预测）

利用图像信号在时间轴上的相关性来实现码率压缩，用前一帧图像中的相关像素值来预测当前帧图像的像素值。

由于要用到前一帧的像素，所以需要帧存储器来实现预测。



■根据参考帧与被编码帧的位置，帧间预测可分为：

- 前向预测：参考帧位于被编码帧之前
- 后向预测：参考帧位于被编码帧之后
- 双向预测：参考帧位于被编码帧的前后两个方向
- MPEG标准所采用的预测编码

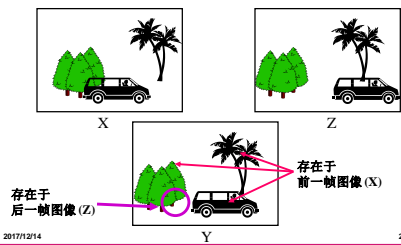
I(Intra frame)帧: 帧内编码帧

P(Predictive-coded frame)帧: 前向预测编码帧

B(Bidirectionally predictive-coded frame)帧: 双向预测编码帧

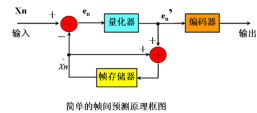


双向预测示意图

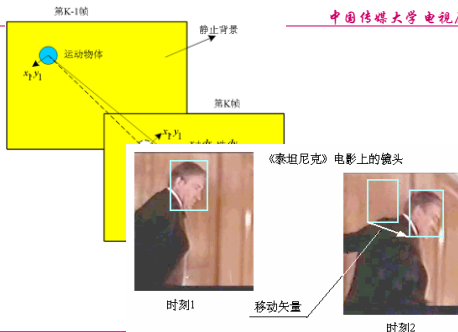


四、运动补偿技术

静止图像前后帧空间位置对应像素完全一样，简单帧间预测对于静止图像画面压缩非常有效。
存储帧存储第一帧图像数据，其后连续帧可以反复读出。

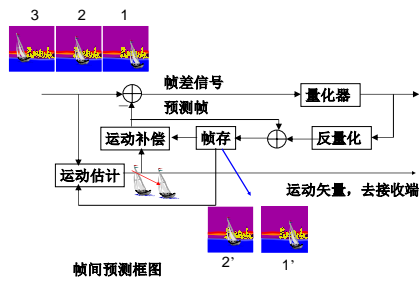
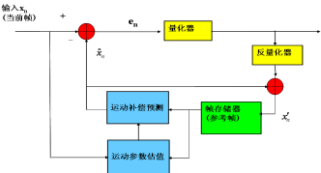


对于活动图像内容，前后帧相应位置像素差值很大，量化后比特数仍然很大，压缩很小。
解决办法——运动补偿



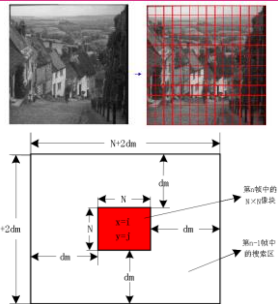
运动估计 (Motion Estimation, ME)，
运动补偿 (Motion Compensation, MC)。

如果对运动区域进行预测，首先要估计出编码对象的运动矢量，这一过程称为运动估计。
然后再根据运动矢量找出编码对象在参考帧的区域位置，用这一区域位置的信号形成当前帧运动对象的预测信号，这叫做运动补偿。

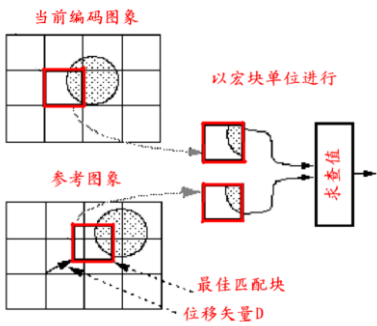
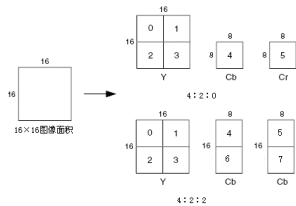


1.运动估计

块匹配法 ■
把图像分成若干子块，设子块图像是由N×N个像素组成的像块，并假设一个像块内的所有像素作一致的平移运动。
估值块大小 (N×N) 的选择应该综合考虑图像细节构成和计算量等因素。
实际常采用是基于宏块的估计。



- 通常一个宏块由一个16×16像素的亮度阵列和同区域内的Cb、Cr色差阵列共同组成，不同的色度格式包括有不同的宏块结构，其中的数字表示块的编码顺序。

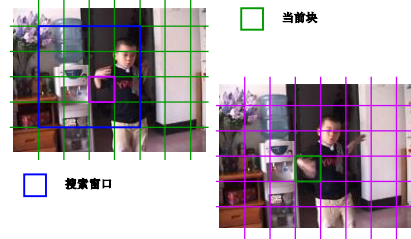


- 运动预测的位移可以是整像素、1/2甚至1/4像素为基本单位。
- 运动矢量位移的精度越高，则帧间预测后的残差就越小，压缩比也越高。



用运动矢量表明两个宏块之间的位移。例如，运动矢量=(-4, 5)，则在当前帧的宏块向左移动4个像素，向下移动5个像素，就可以在参考帧中找到最匹配的宏块。

搜索窗口示例



运动矢量与运动搜索

- 寻找运动矢量的过程称为运动搜索，可以分为2种类型：

全部搜索：

在限制的搜索区域内对每一个可能的块进行比较，找到最相似的块，这些寻找的速度比较慢。

快速搜索：

用较少的比较次数，找到最佳匹配块的近似块，这样搜索速度比较快。

2、匹配准则

对窗口中的部分或全部像素位置计算差值，从中找出差值最小者作为运动矢量

目前有三种判决标准：

- 绝对差值AE (absolute difference)最小
- 均方误差MSE (mean-square error)最小
- 平均绝对帧差MAD (mean of the absolute frame difference)最小

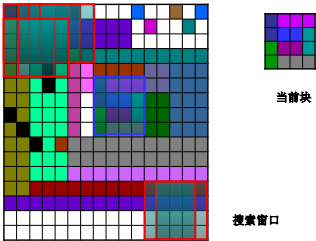
绝对差值AE: $AE = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} |f(i, j) - g(i - d_x, j - d_y)|$

均方误差MSE: $MSE = \frac{1}{I \times J} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} [f(i, j) - g(i - d_x, j - d_y)]^2$

平均绝对帧差MAD: $MAD = \frac{1}{I \times J} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} |f(i, j) - g(i - d_x, j - d_y)|$

i, j 分别表示宏块的横坐标和纵坐标;
 I, J 分别表示宏块的横坐标和纵坐标的像素个数;
 $f(i, j)$ 表示当前帧中的宏块的灰度值;
 $g(i - dx, j - dy)$ 表示参考帧中的宏块在横坐标和纵坐标的偏移量为 (dx, dy) 处的灰度值。

效果最优的搜索方法：全搜索



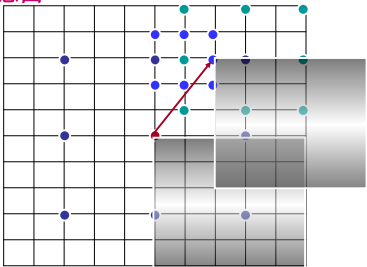
全搜索

- 对搜索窗口中的每个像素均计算其差值
- 最小的差值对应的位置即为运动矢量
- 获得的运动矢量是全局最优的
- 缺点：计算量极其庞大，假定窗口大小为 32×32 ，则：
 - 共有 $32 \times 32 = 1024$ 个位置，每个位置计算256次减法，256次乘法，255次加法
 - 从1024个位置找出最小值还需要1023次比较

速度最快的方法：三步法

- 基本思路：
 - 先用大的步长进行搜索，找到一个不太准确的运动矢量
 - 然后用小步长，在该运动矢量附近继续搜索，找到更准确的运动矢量
- 特点：
 - 可大幅度减少搜索位置
 - 使用MAD判决准则
- 缺点：
 - 运动矢量可能不是全局最优的

三步法示意图



3.运动补偿

- 重建的参考帧存放在帧存储器中，当前帧数据块和参考帧数据块经过运动估值后得到参考帧的匹配块和运动矢量，将当前帧被编码宏块与最佳匹配宏块求差值，得到残差块。因此只需要传送运动矢量和残差块数据，就可以恢复出当前帧的预测图像。
- 经过运动补偿后再求残差值，残差数据值将会很小，通过量化可以使传输的数据量大大降低。

搜索算法对质量的影响

- 搜索结果越准确, 差值图像越小, 差值图像帧内编码效果越好
- 结论:
 - 同样的数据率, 图像质量更高
 - 同样质量条件下, 数据率更低
- 搜索窗口越大, 越容易找到最佳匹配

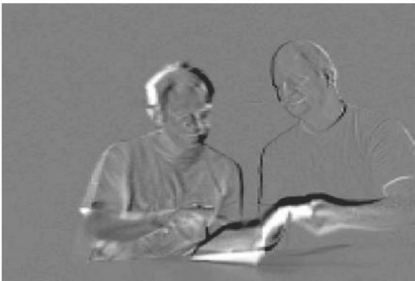
前一帧 运动补偿效果实例



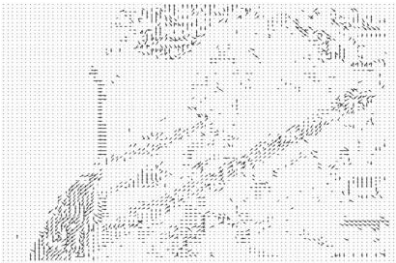
后一帧



前后帧之间残差



像素运动轨迹



前一帧 后一帧



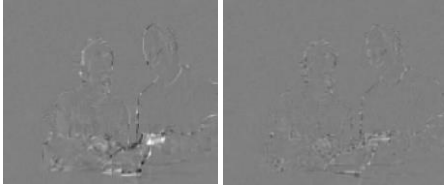
前后帧之间残差 (Residual)
(no motion compensation)

前后帧之间残差 (Residual)
(16 × 16 block size运动补偿)



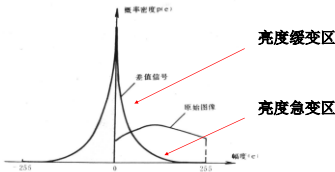
前后帧之间残差 (Residual)
(8 × 8 block size运动补偿)

前后帧之间残差 (Residual)
(4 × 4 block size运动补偿)



五、量化器设计

在预测编码中, 要对预测误差值 e_n 进行量化
预测误差值 e_n 的概率分布呈**拉普拉斯分布**, 集中在零值附近一个较窄的范围内, 随着预测误差绝对值的增大, 概率迅速下降



量化误差是预测编码产生编码失真的主要原因

人眼对图像的平坦区误差分辨能力比较高, 对图像的突变区误差的分辨能力比较低。

- 预测误差值小, 图像平坦区 (低频分量), 细量化;
- 预测误差值大, 对应图像突变区 (高频分量), 粗量化;
- 在预测编码的量化器的设计中, 利用预测误差值的概率分布特性和人眼的视觉特性, 进行以上处理, 以尽量减少编码比特数, 将量化误差控制到主观视觉可以忍受的范围之内。

第三节 正交变换编码

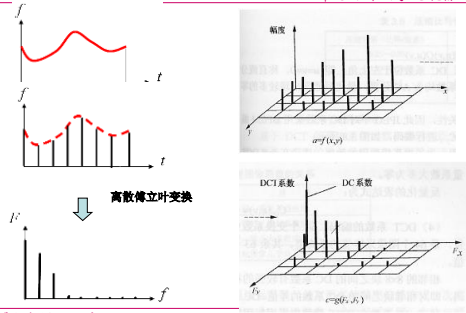
基本思想:

将空间域的图像信号变换到另外的域 (变换域) 进行描述, 再根据图像在变换域中系数的特点和人眼的视觉特性进行编码。
正交变换的变换矩阵是可逆的, 使得解码既保证有解又运算方便, 因而变换编码一般是采用正交变换。

一般图像数据在空间上具有较强的相关性, 变换到变换域后能量往往被集中在少数样值上, 通过舍弃一些较小的系数, 实现数据压缩的目的。

变换的物理意义

- 预测编码消除相关性的能力有限, 变换编码是一种更高效的压缩编码。
- 一般来说, 图像数据在空间上具有较强的相关性, 变换到变换域的变换系数可以实现去相关和能量集中。
- 变换系数的能量往往被集中在少数样值上, 再通过量化舍弃一些较小的系数, 实现数据压缩的目的。



变换的关键在于去除相关性，即使得变换后的图像数据互相之间都不相关，互相独立。

常用的正交变换有：

离散傅立叶变换DFT、离散余弦变换DCT、离散正弦变换DST等。

去除相关性能力，DCT最高，DST最低。

再有，DFT是复数运算，是其最不利之处。

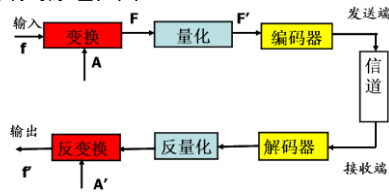
由于DCT变换具有较好的去除相关的能力，实现快速算法比较容易，因此，在数字视频压缩中得到了广泛的应用。

信息论的观点：

相关性强，信息量少；分布不均匀，信息量少。

变换域的信息量不会发生变化。因此，去除相关的能力越强，变换域的线条分布就会越不均匀。

变换编码原理框图



变换编码包括：变换、量化和编码三个步骤。

变换本身并不会产生压缩，实现压缩的是变换后的量化编码。

一、二维 $N \times N$ 图像块的DCT变换

Ahmed 和 Rao于1974年提出！

正变换DCT

$$F(u, v) = C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

反变换IDCT

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{N}} & \text{当 } u, v = 0 \\ \sqrt{\frac{2}{N}} & \text{当 } u, v = 1, 2, \dots, N-1 \end{cases}$$

像素采样值： $f(x, y)$ $x, y = 0, 1, \dots, N-1$

$u, v = 0, 1, \dots, N-1$, 分别表示图像在变换域中的水平空间频率和垂直空间频率

二维DCT的矩阵表示式

正变换DCT： $F(u, v) = [A][f][A^T]$

$$A: a(i, j) = C(i) \cos \frac{(2j+1)i\pi}{2N} \quad (0 \leq i \leq N-1, 0 \leq j \leq N-1)$$

正交变换矩阵 $[A]$

转置矩阵 $[A]^T$

根据矩阵的正交性，有 $[A]^T = [A]^{-1}$

反变换IDCT：

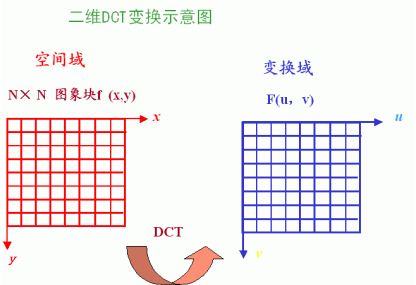
$$f(i, j) = [A]^{-1} F [A^T]^{-1} = [A^T] F [A]$$

• 二维 8×8 DCT变换 矩阵

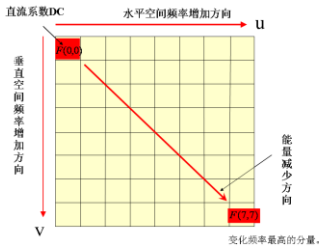
$$A: a(i, j) = C(i) \cos \frac{(2j+1)i\pi}{2N} \quad (0 \leq i \leq N-1, 0 \leq j \leq N-1)$$

$$A = \frac{1}{2} \begin{bmatrix} \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} \\ \cos \frac{\pi}{16} & \cos \frac{3\pi}{16} & \cos \frac{5\pi}{16} & \cos \frac{7\pi}{16} & -\cos \frac{9\pi}{16} & -\cos \frac{11\pi}{16} & -\cos \frac{13\pi}{16} & -\cos \frac{15\pi}{16} \\ \cos \frac{3\pi}{16} & \cos \frac{9\pi}{16} & -\cos \frac{5\pi}{16} & -\cos \frac{7\pi}{16} & \cos \frac{11\pi}{16} & \cos \frac{13\pi}{16} & -\cos \frac{15\pi}{16} & \cos \frac{17\pi}{16} \\ \cos \frac{5\pi}{16} & -\cos \frac{3\pi}{16} & \cos \frac{7\pi}{16} & -\cos \frac{9\pi}{16} & \cos \frac{13\pi}{16} & \cos \frac{15\pi}{16} & -\cos \frac{17\pi}{16} & \cos \frac{19\pi}{16} \\ \cos \frac{7\pi}{16} & -\cos \frac{9\pi}{16} & -\cos \frac{3\pi}{16} & \cos \frac{5\pi}{16} & \cos \frac{15\pi}{16} & -\cos \frac{13\pi}{16} & \cos \frac{17\pi}{16} & -\cos \frac{19\pi}{16} \\ \cos \frac{9\pi}{16} & \cos \frac{11\pi}{16} & \cos \frac{13\pi}{16} & -\cos \frac{15\pi}{16} & -\cos \frac{17\pi}{16} & \cos \frac{19\pi}{16} & \cos \frac{21\pi}{16} & -\cos \frac{23\pi}{16} \\ \cos \frac{11\pi}{16} & -\cos \frac{13\pi}{16} & \cos \frac{15\pi}{16} & \cos \frac{17\pi}{16} & -\cos \frac{19\pi}{16} & \cos \frac{21\pi}{16} & -\cos \frac{23\pi}{16} & \cos \frac{25\pi}{16} \\ \cos \frac{13\pi}{16} & \cos \frac{15\pi}{16} & -\cos \frac{17\pi}{16} & \cos \frac{19\pi}{16} & \cos \frac{21\pi}{16} & -\cos \frac{23\pi}{16} & \cos \frac{25\pi}{16} & -\cos \frac{27\pi}{16} \end{bmatrix}$$

$$C(i) = \begin{cases} \frac{1}{\sqrt{N}} & u, v = 0 \\ \sqrt{\frac{2}{N}} & u, v = 1, 2, \dots, N-1 \end{cases} \quad C(j) = \begin{cases} \frac{1}{\sqrt{8}} & u, v = 0 \\ \sqrt{\frac{2}{8}} & u, v = 1, 2, \dots, 7 \end{cases}$$

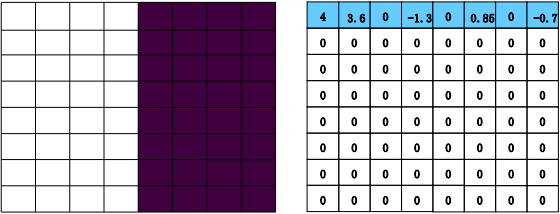


变换系数矩阵 $F(u,v)$

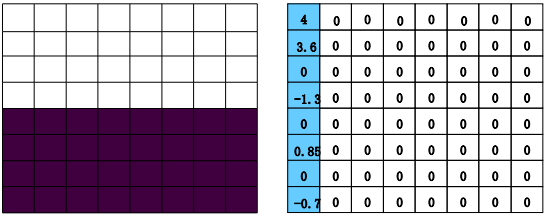


二、DCT变换举例

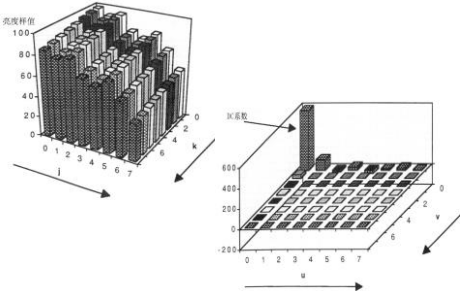
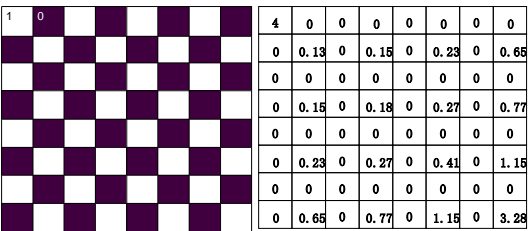
- (a): 仅具有水平空间频率成分

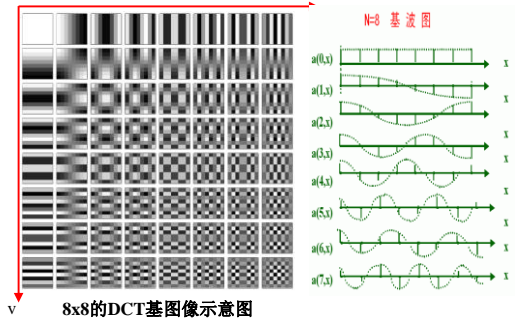
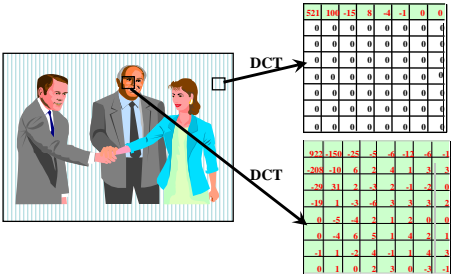


- (b): 仅具有垂直空间频率成分



- (c): 既具有水平空间频率成分，又具有垂直空间频率成分





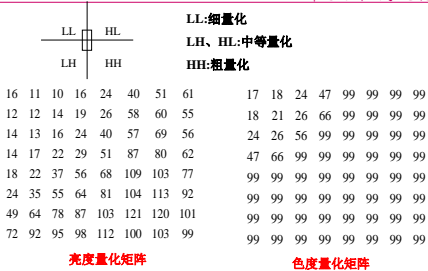
总结:

一般电视图像的构成都是以大、中面积内容为主, 精细内容较少。因而可估计到, 2维DCT系数矩阵中左上方的系数值会大些, 而越接近右下角, 系数值会越小。

也就是说, DCT变换使数值较大且概率较大的低频分量数据向左上方集中, 使数值较小且概率一般也小的高频分量向右下方集中, 具有很好的能量集中和去相关性。

三、量化器

- DCT变换具有保真性, 变换本身并不能压缩码率。
 - 只有通过量化器对系数量化和熵编码后才能降低数据量。
 - 量化器根据图像在变换域中的分布特性来进行设计:
1. 大多数图像亮度值变化少, 变换域后的变换系数的大部分能量集中在直流和低频区域, 又因为亮度突变区域如轮廓、边缘等高频细节部分调制度低, 变换后的系数一般很小。
 2. 利用预测误差值的概率分布特性和人眼的视觉特性, 对图像的平坦缓变区(对应低频分量)的部分进行细量化, 对于图像突变区(对应高频分量)的部分进行粗量化, 以尽量减少编码比特数, 将量化误差控制到主观视觉可以忍受的范围之内。



DCT变换及压缩编码实例:

— 8×8图像亮度数据矩阵及其DCT系数矩阵

139	144	149	153	156	158	155	155
144	151	153	156	159	156	156	156
150	155	160	163	168	166	166	166
159	161	162	160	160	159	160	159
159	160	161	162	162	165	165	165
161	161	161	161	160	167	167	167
162	162	161	163	162	167	167	167
162	162	161	161	163	168	168	168

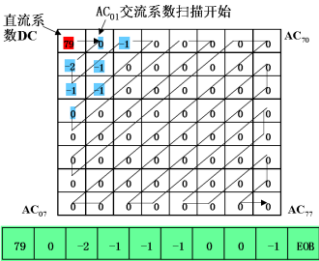
126	-1	-12	-6	2	-2	-1	1
-23	-17	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	1	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	1	1	-1
-1	0	0	-1	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

- 亮度量化矩阵及量化后的DCT系数

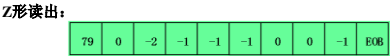
16	11	10	16	24	40	61	61
12	12	14	19	26	58	60	56
114	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	103	103	77
24	35	55	64	81	104	113	92
49	64	78	87	108	121	120	101
72	92	95	98	112	100	103	99

79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- ZigZag (Z字形) 扫描输出路径



游程编码 (RLC, Run Length Coding)
游程编码是一种简单的熵编码，一般不直接使用，为了达到较好的压缩效果，通常与其他编码方法一起使用，如和正交变换一起使用。
特点是改变信源中连续出现相同符号的表达方式来降低平均码长。



游程编码：
(0, 79), (1, -2), (0, -1), (0, -1),
(0, -1), (2, -1), EOB
熵编码 (VLC)：霍夫曼编码

第四节 视频压缩编码概述

国际标准化协会(International Standardization Organization, ISO)、国际电子学委员会(International Electronics Committee, IEC)、国际电信协会 (International Telecommunication Union, ITU) 等国际组织，于90年代领导制定了许多重要的多媒体数据压缩标准。如H. 261、JPEG、MPEG-1、H. 263、MPEG-2、MPEG-4、MPEG-7等等。标准的制定大大推动了数字电视技术及多媒体技术的发展，这些标准已在数字电视、多媒体领域得到广泛应用。

ISO/IEC:MPEG-X
ITU-R :H.26X

■Moving picture experts group (MPEG) standards

■Group of Experts

■Standards

■Multimedia Formats

■Transmission

■Conformance

MPEG-1 CD-ROM 1991

MPEG-2 Digital TV 1995

MPEG-4 Internet Video 1999

MPEG-7 Object Description 2002

MPEG-21 DRM 2005

■MPEG-1 CD-ROM

■MPEG-2 Broadcast

■MPEG-4 Internet

■MPEG-7 Object Description

■MPEG-21 DRM (数字版权管理)

第五节 JPEG数字视频压缩编码算法

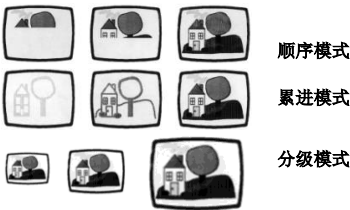
1、JPEG标准

JPEG(Joint Photographic Experts Group)是联合图像专家组的英文缩写。该组织从1986年正式开始制订静止数字图像的压缩编码标准，该标准于1992年正式通过，称为JPEG标准。
JPEG是第一个数字图像压缩的国际标准，它不仅适于静止图像的压缩，对于电视图像序列的帧内压缩也常采用JPEG算法，因此JPEG是一个适用范围广泛的通用标准。

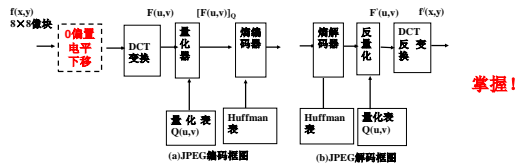
JPEG包括两种基本压缩方法压缩方法：

- (1) 有损压缩方法：它是以DCT为基础的压缩方法，其压缩比较高，是JPEG的基础。
- (2) 无损压缩方法，又称预测压缩方法。是以二维DPCM为基础的空间方式，保证解码后，能完全精确地恢复源图像采样值，其压缩比低于有损压缩方法。

在实际应用中，普遍使用的是基于DCT压缩方法的顺序编码工作模式，又称为基本顺序模式。
在顺序模式中每一个图像分量按从左到右、从上到下扫描，一次扫描完成编码。



JPEG编解码原理框图



- JPEG编解码算法主要有以下几个重要步骤：
1. DCT变换
通过DCT变换，便于去除图像数据的空间冗余。
 2. 量化
利用人眼视觉特性设计而成的矩阵量化DCT系数，减少视觉冗余。
 - 3.对量化后的直流系数和交流系数分别进行差分进行VLC编码，减少数据（编码）冗余。

JPEG推荐了亮度信号和色度信号两种量化表 中国传媒大学 电视原理 17.12.14

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	21	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

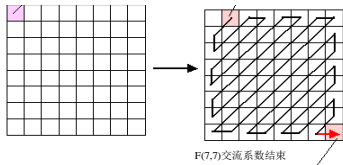
亮度量化矩阵

色度量化矩阵

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
26	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

3. 之字形扫描

在编码之前，需要把二维的变换系数矩阵转换为一维序列，由于量化之后右下角高频系数大部分为零，采用之字形扫描读取可以制造较长的零游程，提高编码效率。在扫描中，如果后续的系数全部为零，则用“EOB”表示块结束。



4. 可变长熵编码 (VLC)

(1) 直流系数

由于直流系数F(0, 0)反映了该子图像中包含的直流成分，通常较大，又由于两个相邻的子图像的直流系数通常具有较大的相关性，所以对DC系数采用差值脉冲编码 (DPCM)，即对本象素块直流系数与前一象素块直流系数的差值进行无损编码：

$$\Delta DC_i = F_i(0,0) - F_{i-1}(0,0)$$

(2) 交流系数

进行游程编码 (RLC)，并在最后加上块结束码 (EOB)；

最后对直流差值码和交流游程码再进行可变字长的霍夫曼 (Huffman) 编码。

实际编码举例：

原图像样值 f(x, y)

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

0偏置电平下移：

灰度级 2^n 的像素值，全部减去 2^{n-1} （一半），数据形式由无符号数变为有符号数（补码），单极性数据变为双极性数据。

原因：

图像内容平均亮度较高，将0电平移到中间，平均亮度降低，便于DCT变换量化后直流的系数大大降低（减少 $128 \times 8 = 1024$ ）。

$$F(u, v) = [A][128 * ones(8,8)][A^T] \quad \text{直流系数为1024，交流系数均为0}$$

电平下移后的
f(x, y)

11	16	21	25	27	27	27	27
16	23	25	28	31	28	28	28
22	27	32	35	30	28	28	28
31	33	34	32	32	31	31	31
31	32	33	34	34	27	27	27
33	33	33	33	32	29	29	29
34	34	33	35	34	29	29	29
34	34	33	33	35	30	30	30

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	159	159	159	159
159	160	161	162	162	155	155	155
161	161	161	160	157	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

原像块样值

DCT变换后的系数

1259.6	-1.0333	-12.081	-5.2029	2.125	-1.6724	-2.708	1.3238
-22.59	-17.484	-6.2405	-3.1574	-2.8557	-0.069456	0.43417	-1.1856
-10.949	-9.2624	-1.5758	1.5301	0.20295	-0.94186	-0.56694	-0.062924
-7.0816	-1.9072	0.22479	1.4539	0.89625	-0.079874	-0.042291	0.33154
-0.625	-0.83811	1.4699	1.5563	-0.125	-0.66099	0.60885	1.2752
1.7541	-0.20286	1.6205	-0.34244	-0.77554	1.4759	1.041	-0.99296
-1.2825	-0.35995	-0.31694	-1.4601	-0.48996	1.7348	1.0758	-0.76135
-2.5999	1.5519	-3.7628	-1.8448	1.8716	1.2139	-0.56788	-0.44564

11	16	21	25	27	27	27	27
16	23	25	28	31	28	28	28
22	27	32	35	30	28	28	28
31	33	34	32	32	31	31	31
31	32	33	34	34	27	27	27
33	33	33	33	32	29	29	29
34	34	33	35	34	29	29	29
34	34	33	33	35	30	30	30

零偏置后的系数

DCT变换后的系数

235.62	-1.0333	-12.081	-5.2029	2.125	-1.6724	-2.708	1.3238
-22.59	-17.484	-6.2405	-3.1574	-2.8557	-0.069456	0.43417	-1.1856
-10.949	-9.2624	-1.5758	1.5301	0.20295	-0.94186	-0.56694	-0.062924
-7.0816	-1.9072	0.22479	1.4539	0.89625	-0.079874	-0.042291	0.33154
-0.625	-0.83811	1.4699	1.5563	-0.125	-0.66099	0.60885	1.2752
1.7541	-0.20286	1.6205	-0.34244	-0.77554	1.4759	1.041	-0.99296
-1.2825	-0.35995	-0.31694	-1.4601	-0.48996	1.7348	1.0758	-0.76135
-2.5999	1.5519	-3.7628	-1.8448	1.8716	1.2139	-0.56788	-0.44564

亮度
量化
矩阵

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	97	103	121	120	101
72	92	95	98	112	100	103	99

作DCT变换后的系数

$F(u, v)$

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-18.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	-1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

量化后的系数 $[F(u, v)]_Q$

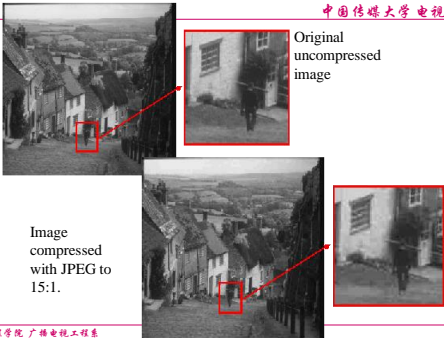
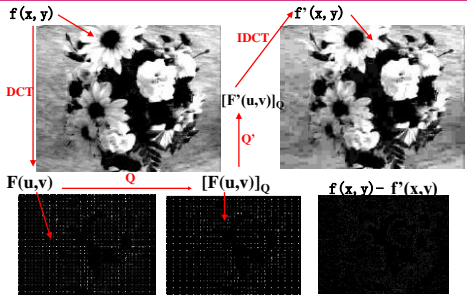
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

反量化后的系数 $[F'(u, v)]_Q$ 与DCT变换后的原系数

$[F'(u, v)]_Q$	240	0	-10	0	0	0	0	0
	-24	-12	0	0	0	0	0	0
	-14	-13	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
$F(u, v)$	235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
	-22.6	-18.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
	-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
	-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
	-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
	1.8	-0.2	-1.6	-0.3	-0.8	1.5	1.0	-1.0
	-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
	-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

反变换后的重建图像数据 $f'(x, y)$ 与原图像数据比较

$f'(x, y)$	144	146	149	152	154	156	156	156
	148	150	152	154	156	156	156	156
	155	156	157	158	158	157	156	156
	160	161	161	162	161	159	157	156
	163	163	164	163	162	160	158	156
	163	164	164	164	162	160	158	157
	160	161	162	162	162	161	159	158
	158	159	161	161	162	161	159	158
$f(x, y)$	139	144	149	153	155	155	155	155
	144	151	153	155	159	156	156	156
	150	155	160	163	158	156	156	156
	159	161	162	160	160	159	159	159
	159	160	161	162	162	155	155	156
	161	161	161	161	160	157	157	157
	162	162	161	163	162	157	157	157
	162	162	161	161	163	158	158	158
误差	5	2	0	-1	-1	1	1	1
	4	-1	-1	-2	-3	0	0	0
	5	1	-3	-5	0	1	0	-1
	1	0	-1	2	-1	0	-2	-4
	4	-3	3	1	0	5	3	1
	2	3	3	3	2	3	1	0
	-2	-1	1	-1	0	4	2	1
	-4	-3	0	0	-1	3	1	0



块效应

整幅图像作DCT变换则无块效应
实际16×16、8×8、4×4分块处理，使得块与块之间的相关性被忽略。不同块采用不同的量化，舍入高频分量不同，造成块边界不连续或整块的块斑现象。量化步长越大，块效应越明显。

块效应主要原因：
1. 高频分离被量化为0，使得块内强边缘处出现锯齿状，称为梯形噪声；
2. 由于量化的四舍五入，使得前后平缓变化的块的亮度DC系数发生跳跃，造成重建块之间的亮度突变，称为格型噪声；

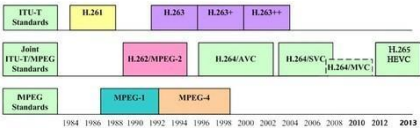
量化后的系数[F(u, v)]_q

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

之字形扫描读出后变成一维序列如下： P331
(15, 0, -2, -1, -1, -1, 0, 0, -1, EOB)
假设前一个像块DC系数为12，则直流差值系数为ΔDC₁=3，又可表示为：
(3, 0, -2, -1, -1, -1, 0, 0, -1, EOB)
把系数序列分成组：把每个非零系数和它前面相邻的全部零系数分在1组内：
{ (3) ; (0, -2) ; (-1) ; (-1) ; (-1) ; (0, 0, -1) ; EOB }
其游程幅度编码结果（用一对符号表示）为：
{ (3) ; (1, -2) ; (0, -1) ; (0, -1) ; (0, -1) ; (2, -1) ; EOB }
AC系数查表为[(Run,size),(Amplitude)]；DC系数查表为： [(size),(Amplitude)]
{ (2,3) ; [(1,2),-2] ; [(0,1),-1] ; [(0,1),-1] ;
[(0,1),-1] ; [(2,1),-1] ; EOB }
注：size是幅度大小经查表得到的位长
查表的Huffman编码值：
{ 01111, 1101101, 000, 000, 000, 111000, 1010 } 共31bit

由此可见，原图像为8×8=64个像素，如果每个像素用8bit编码，则压缩前的总比特位数为：
8×8×8=512bit
经过JPEG算法编码后，输出的总比特位数为：
31bit
平均比特率： $\frac{31bit}{64pixel} \approx 0.485(bit / pixel)$
压缩比： $\frac{512}{31} \approx 16.5 = \frac{8}{0.485}$

第六节 视频压缩编码



视频编码标准的发展历程

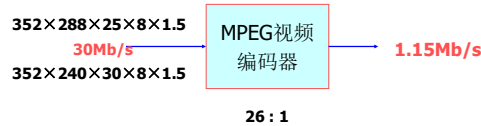
MPEG是ISO下属的活动图像专家组（Moving Picture Experts Group）的简称，是ISO和IEC两个国际组织的联合技术委员会领导下的专家组。成立于1988年，负责制定视频及其伴音的编码标准。

一、MPEG-1和MPEG-2概述

MPEG-1标准全称: Codeing of Moving Pictures
and Associated for Digital Store Media at up
to about 1.5Mbps
标准编号: ISO/IEC11172、ISO/IEC13818
标准发布日期: 1991年、1994年
主要三个部分: 系统、视频、音频

MPEG-1视频压缩标准

输入、输出指标:



- MPEG-1标准是世界上:
- 第一个集成视、音频编码标准;
 - 第一个只定义解码器(接收机)的标准;
 - 第一个独立于视频格式(NTSC/ PAL/SECAM)的视频编码;
 - 第一个由对音频和视频感兴趣的所有工业界联合制定的标准;
 - 第一个完全以软件方式制定的标准;
 - 第一个包括一个软件实施的标准。

由于MPEG-1的1.5 Mbps速率的图像质量不能满足广播电视对图像质量的要求, 于是MPEG专家组制定了MPEG-2。

MPEG-2是由ISO/IEC之下的MPEG专家组与ITU(国际电信联盟电信标准化部门)的ATV的图像编码专家组一起共同开发的, 所以MPEG-2标准也是ITU-T的建议。标准的制订工作于1990年开始, 全称为“运动图像及有关声音信息的通用编码”, 并于1994年公布。

MPEG-2广泛应用于DVD、SDTV 和 HDTV中。美国的ATSC、欧洲的DVB、日本的ISDB都已把MPEG-2作为信源编码标准。我国制定的数字卫星电视和数字有线电视的标准中, 也采用了MPEG-2标准。

MPEG-1与MPEG-2

(1) 标准组成

- 第一系统部分, 关于多路音频、视频和资料的复用和同步的规定;
- 第二视频部分, 主要涉及各种比特率的数字视频压缩编码的规定;
- 第三音频部分, 数字音频信号的压缩编码的规定。

(2) 兼容性

MPEG-2标准支持不同分辨率格式之间的兼容。

符合规定的MPEG-2视频译码器, 应能够译码MPEG-1比特流, 即后向兼容MPEG-1。MPEG-2提供的空间可分级方法允许在低分辨率即底层中使用MPEG-1编码, 即兼容MPEG-1。

MPEG-1关键词:

- 码率1.5Mbps;
- 色度格式4:2:0;
- 只能处理CIF格式;
- 只能处理逐行格式;
- 主要用于VCD;
- 音频编码分3层, 第三层为MP3的应用。

(3) 区别

掌握!

MPEG-2关键词:

- 兼容MPEG-1;
- 码率、分辨率不固定, 包括高清;
- 色度格式扩展, 包括4: 2: 2和4: 4: 4;
- 逐行、隔行皆可处理;
- 主要用于广播电视, 是三大数字电视制式的信源压缩标准;

二、MPEG2 的型和级

为了解决通用性和特殊性的矛盾：
规定了四种输入图像格式（分辨率），称为级(Levels)：
分为低级、主级、高1440级和高级。
不同的压缩处理方法，称为类或型(Profiles)：
分为简单类、主类、信杂比可分级类、空间可分级类和高类。

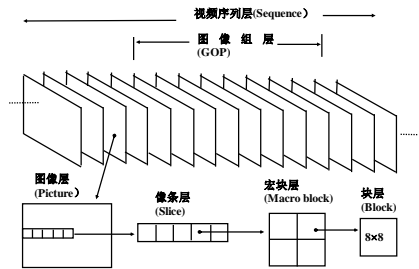
级提供了灵活的信源格式，型提供了不同的压缩质量！

级提供了灵活的信源格式，型提供了不同的压缩质量！

级 Level	参数	类 (Profile)				
		Simple	Main	SNR Scalable	Spatial Scalable	High
	帧类型 抽样格式	I帧P 4:2:0	I*P*B 4:2:0	I*P*B 4:2:0	I*P*B 4:2:0	I*P*B 4:2:0或4:2:2
High	帧类型 帧率 帧数		1920 1152 25			1920 1152 25
	最大比特率 (Mbps)		80			100 (10P+1H)
	帧类型 帧率 帧数		1440 1152 25		1440 1152 25	1440 1152 25
High-1440	最大比特率 (Mbps)		60		60	80
	帧类型 帧率 帧数	720 576 30	720 576 25	720 576 25		720 576 25
	最大比特率 (Mbps)	15	15 (MP+MH)	15		20
Main	帧类型 帧率 帧数		352 288 25			
	最大比特率 (Mbps)		4			
	帧类型 帧率 帧数					
Low	最大比特率 (Mbps)					
	帧类型 帧率 帧数					
	最大比特率 (Mbps)					

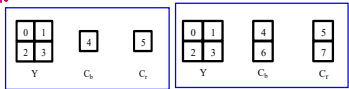
三、MPEG-1/ MPEG-2的视频结构

- MPEG对视频资料规定了层次结构，共分为六层：
1. 视频序列 (Video Sequence) (最高层)；
 2. 图像组 (Group of Pictures)；
 3. 图像 (Pictures)；
 4. 像条 (Slice)；
 5. 宏块(Macroblock)；
 6. 像块 (Block) (最低层)。



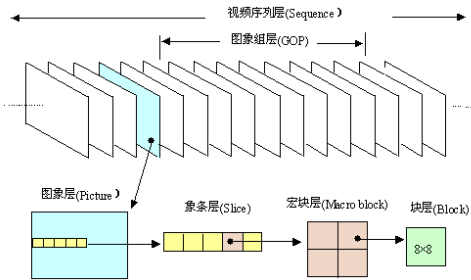
1. 视频序列 (VS)
视频序列也称图像序列，它是随机选取节目的一个基本单元。从节目内容看，一个视频序列大致对应于一个镜头，切换一个镜头，即表示开始一个新的序列。
2. 图像组 (GOP)
图像组是将一个图像序列中连续的几个图像组成一个小组，简称为GOP。它是对编码后的视频码流进行编辑的存取基本单元。
3. 图像 (P)
图像是一个独立的显示单元，也是图像编码的基本单元。可分为I、P、B三种编码图像，分别对应三种压缩编码模式，即帧内压缩编码 (I帧编码)、前向预测编码 (P帧编码) 和双向预测编码 (B帧编码)。

4. 像条 (SLICE)
像条由一系列连续的宏块组成。像条是发生误码后且不可纠正时，数据重新获得同步从而能正常译码的基本单元。
5. 宏块 (MB)
宏块是运动预测的基本单元。运动估计以宏块为单位得到最佳匹配宏块的运动向量。运动预测只对亮度数组进行，色差数组使用和亮度数组相同的运动向量。一个宏块由一个16×16像素的亮度数组和同区域内的C_b、C_r色差数组共同组成。由于在MPEG-1中图像的色度格式是4:2:0, 由6个块组成，其中4个亮度块和2个色度块。在MPEG-2中图像的色度格式还包括4:2:2和4:4:4格式。因此，一个宏块又由许多8×8的块组成。



6. 块

块是DCT变换的基本单元。一幅图像以亮度数据数组为基准被分成若干个8×8像素的数组，简称为块。可以是亮度块或色差信号块。

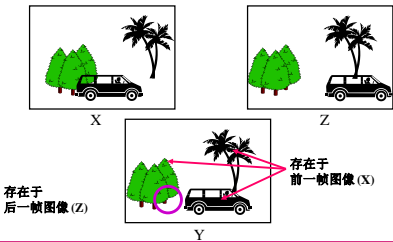


四、I、B、P帧编码原理

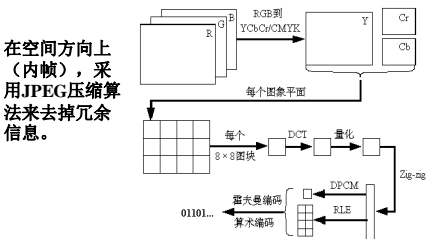
MPEG标准中三种类型图像:I帧/P帧/B帧

- I帧 (Intra coded picture): 帧内图, 仅利用该帧图像本身信息进行编码, 即直接进行DCT变换, 量化和熵编码. 属于中等压缩. 可作为P/B帧的参考帧. 压缩比 (2-5): 1
- P帧 (Predictive-coded picture): 预测图, 根据前面最近的I帧或者另一个P帧进行预测, 属于前向预测. 由于使用运动补偿, P帧比I帧压缩更大, 并可作为P/B帧的参考帧. 会传播误差. 压缩比 (5-10): 1
- B帧 (Bidirectionally predictive-coded picture): 双向预测图, 它既用过去的帧作基准, 也用未来的帧作基准, 即前向和后向预测都有. 压缩比 (20-30): 1
- B帧压缩最大, 且不传播误差.

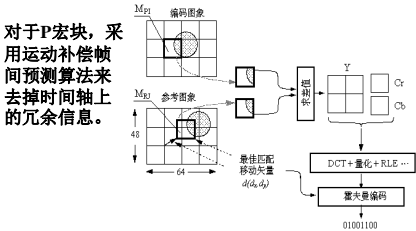
双向预测示意图



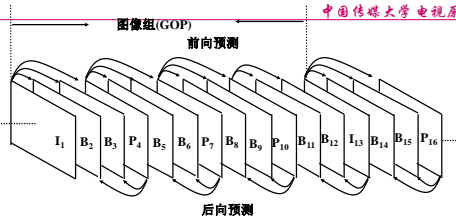
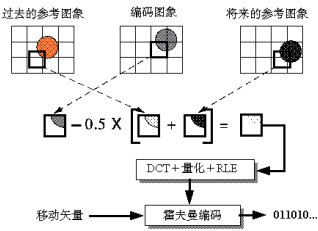
I图像帧的压缩算法:



P图像帧的压缩算法:

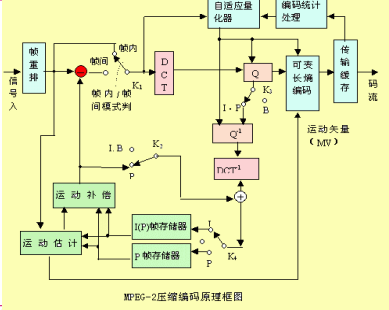


B 图像帧的压缩算法:

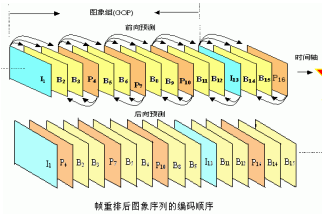


I 帧图像: 只由帧内宏块组成
P 帧图像: 包含前向运动补偿、**帧内宏块**
B 帧图像: 包含前向、后向和双向运动矢量和**帧内宏块**, B帧压缩最大, 且不传播误差。

五、编码器工作原理



由于B帧是双向预测帧(前向预测和后向预测), 在后向预测时, 需要用它将来的一个帧作为参考帧。因此需要把将来的帧先送入编码器, 就要对原始图像重新排序, 这称为**帧重排**。
例如P₄作为B₂和B₃的后向预测参考帧, 应该排在B₂和B₃之前。按编码顺序重排后的图像序列



总结: MPEG图像压缩编码
MPEG算法达到了很高的压缩比, 但仍保持了很好的图像质量, 单靠帧内编码是不可能达到的。
在MPEG压缩编码中, 主要是通过采用DCT变换编码和运动预测技术来分别去除空间冗余和时间冗余:
即先通过运动估计和运动补偿来提高去除图像序列间的时间冗余的强度, 再通过**预测**来去除图像序列间的时间冗余, 然后通过DCT变换将差值信号的空间冗余去除, 使系数能量集中在低频部分。

注意, DCT本身并不能实现数据压缩, 它是通过**量化和变长子长编码**最终达到压缩编码的目的。
因此, 在MPEG-2压缩编码算法中, 不仅包括了JPEG算法中的DCT、自适应量化和熵编码等一系列帧内编码方法, 更重要的是利用了**帧间运动补偿技术**。