

## Chapter 3 image

- ♦ Dithering and Dither Matrix

抖动：基本策略是以亮度分辨率换取空间分辨率

对于在 1 位打印机上的打印来说，如果通过增加抖动矩阵的大小来增加亮度级数量，也会增加输出图像的大小，这就降低了图像在局部的清晰度，从而降低了图像的空间分辨率。

抖动矩阵: 使用一个  $n \times n$  的二值点阵表示  $n^2+1$  级灰度分辨率

$$M_1 = \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

$$M_{n+1} = \begin{pmatrix} 4M_n & 4M_n + 2 \\ 4M_n + 3 & 4M_n + 1 \end{pmatrix}$$

抖动规则：把 0...255 重新映射到 0...4 通过用 256/5 除灰度值。如果一个灰度值比抖动矩阵的元素大，则那个元素的位置打一个点（置 1）

Ordered dithering: 不增加图像的大小。把  $n \times n$  抖动矩阵在同一时刻移动到图像在水平和垂直方向  $n \times n$  个像素上（图像值已经被降低到  $0 \sim 2^n$  区间中），如果某个像素的亮度值大于覆盖它的矩阵元素的编号则在打印机相应元素输出位填涂。

Begin //  $I(x,y)$  is the input,  $O(x,y)$  is the output,  $D$  is the  $n \times n$  dither matrix

```
for x=0 to  $x_{\max}$ 
  for y=0 to  $y_{\max}$ 
    i = x mod n
    j = y mod n
    if  $I(x,y) > D(i,j)$ 
       $O(x,y) = 1$ ;
    else  $O(x,y) = 0$ ;
```

End

- ♦ Look up table and Median Cut Algorithm

**颜色查找表**：为每个像素存储下标或者说是编号值，那么如果一个像素存储的值是 25，意为在颜色查找表中找到第 25 行。

**Median Cut:**

1. 将红色字节的值排序，并找到中值（即一半像素值比其小一半比其大），比中值小的值被标记为 0 位，比中值大的值被标记为 1 位。
2. 仅仅考虑第一步后被标记为 0 的像素，并将它们的绿色值排序。接着用另一位标记像素，0 标记绿色中那些比中值小的像素，1 标记为那些比中值打的像素。然后，把同样的规则用于第一步后被标记为 1 的像素。
3. 继续在蓝色分量上执行上述步骤，就得到了 3 位的方案。
4. 重复前面所有步骤，得到 6 位方案。
5. 再重复红色、绿色步骤，得到 8 位方案。

- ♦ Interlace scan and Adam 7 scan

Interlace scan: 一种图像编码方法使得图像在被接受的过程中逐渐清晰。

Adam 7: 7 遍隔行扫描算法。PNG 中支持的交织扫描。

1	6	4	6	2	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7
3	6	4	6	3	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7

Gif 中的交织扫描：

第一通道(Pass 1)提取从第 0 行开始每隔 8 行的数据；

第二通道(Pass 2)提取从第 4 行开始每隔 8 行的数据；

第三通道(Pass 3)提取从第 2 行开始每隔 4 行的数据；

第四通道(Pass 4)提取从第 1 行开始每隔 2 行的数据；

- ♦ Major differences between BMP, GIF, PNG and JPEG

Bmp:windows 中使用的格式。8 位 BMP 通常采用 256 各元素的颜色查找表。表中的颜色数据根据重要性排列。颜色数据是 4 个字节，前三个分别是 RGB,最后一个为 0。对于 32 位或 16 位位图，如果压缩方法是 'BI\_RGB' 则不使用颜色查找表。如果压缩方法是 'BI\_BITFIELDS' 则表中的元素是位图数据 AND 颜色表的数据。

使用游长编码压缩

占用磁盘空间大

8-bit GIF: 是一种重要的格式，因为和 WWW 以及 HTML 标记语言的历史关系，是第一个为网络浏览器识别的图像类型。

GIF 文件中的数据是颜色查找表中的当前像素颜色的索引值。最多支持 256 种颜色。可以被分成一些存储块，这样就可以存储多个图像并且控制动画和交互。流式读写，适用于网络应用。用 4 遍交织显示。允许不显示图像的一些部分。能存储图像中的文字信息。使用 LZW 无损压缩。跨平台。

Gif87a:原始的规格

Gif89a:支持简单动画，支持简单控制和透明度索引。

JPEG:现在最重要的图像压缩标准。利用人类视觉系统的限制实现高压缩率。允许用户设置质量级别（即压缩率）。

PNG:代表便携式网络图形。支持最多 48 位的颜色信息。文件包含 gamma 校正信息用于正确显示色彩图像。包含 alpha 通道信息用于控制透明度。支持 Adam7 渐进式显示。支持流式读写。跨平台。无损压缩。

区别于 GIF 的特点：

支持 48 位色彩图像。支持 16 为灰度图像。支持 16 位 alpha 通道。增加 gamma 信息。使用 LZ77 无损压缩。使用 CRC. 加速渐进式显示。标准读写工具箱。可以一个文件存储多个图像。

## Chapter 4 Color

- ♦ Image formation model

具有光谱能量分布  $E(\lambda)$  的光源发出的光照射到一个具有表面光谱反射的函数  $S(\lambda)$  的表面后被反射，然后被眼睛的是追函数  $q(\lambda)$  过滤。函数  $C(\lambda)$  成为颜色信号，是光源  $E(\lambda)$  和反射  $S(\lambda)$  的乘积：

$$C(\lambda) = E(\lambda)S(\lambda)$$

成像模型的方程：

$$(4.3) \quad \begin{aligned} R &= \int E(\lambda)S(\lambda)q_R(\lambda)d\lambda \\ G &= \int E(\lambda)S(\lambda)q_G(\lambda)d\lambda \\ B &= \int E(\lambda)S(\lambda)q_B(\lambda)d\lambda \end{aligned}$$

- Color Match Model

一门包含于心理学中的技术把基本的 R、G、B 光线的组合于某种色调匹配起来。三种基本光线可以组成特殊集合，这个集合成为原色集。为了于给定的色调匹配，要求一组观察者利用控件分别调节三原色的亮度，直到最终的光线于要求的颜色最为接近为止。这个实验装置称为色度计。

- CIE Chromaticity Diagram

CIE(国际照明组织)色度图：CIE 颜色匹配曲线有负的原型突出部分，因此设计了一套虚拟原色以

使颜色匹配函数只有正值。结果曲线通常称为颜色匹配函数。它们是  $\bar{r}$ 、 $\bar{g}$ 、 $\bar{b}$  曲线进行线性（3 \*

3 矩阵）变换得到的，用符号  $\bar{x}(\lambda)$ 、 $\bar{y}(\lambda)$ 、 $\bar{z}(\lambda)$  表示。变换矩阵根据以下条件选择：中间的颜色匹配函数  $\bar{y}(\lambda)$

与发光效率曲线  $V(\lambda)$  相等。

对于一个光谱能量分布  $E(\lambda)$  来说，表现颜色特征所需的基本色度信息是三色值 X、Y、Z 的集合。

Y 称为亮度。

$$(4.6) \quad \begin{aligned} X &= \int E(\lambda)x'(\lambda)d\lambda \\ Y &= \int E(\lambda)y'(\lambda)d\lambda \\ Z &= \int E(\lambda)z'(\lambda)d\lambda \end{aligned}$$

$$(4.7) \quad \begin{aligned} x &= \frac{X}{(X+Y+Z)} \\ y &= \frac{Y}{(X+Y+Z)} \\ z &= \frac{Z}{(X+Y+Z)} \end{aligned}$$

$$x+y+z=1, z=1-x-y$$

x,y 称为色度。

颜色匹配曲线的性质：

1. 颜色匹配曲线把相同的值加起来，每条曲线下的区域对每个颜色匹配函数是相同的。
2. 对于一个所有光谱能量分布值都为 1 的白色光源（等能量白色光），其色度值是（1/3,1/3）
3. 因为  $x,y \leq 1$ ,  $x+y \leq 1$ , 所以所有可能的色度值都必须位于虚对角线下

CIE 色度图的性质：

1. 在光谱轨迹上具有色度的颜色表现出“纯”色，是最饱和的，而靠近白点的颜色更加不饱和。
2. 对于两种光的混合，结果色度位于两种光的色度的连线上。

3. 主波长是光谱轨迹与连接白点和指定颜色的直线延长线的交点。

#### ♦ Gamma Correction

图像文件中的 RGB 数值被转换会模拟信号并驱动阴极射线管 (CRT) 中的电子枪。电子的发射与驱动电压成比例, 并且我们希望 CRT 系统产生的光线与电压线性相关。但实际上 CRT 产生的光线实际上与电压的指数大致成正比。这个指数称为 gamma.

如果文件在红色通道的值是 R, 则屏幕发射的光线于  $R'$  成正比。对需伽马校正的信号在发射前将其指数变成  $1/r$ , 因此可以获得线性信号。

实践中采用  $R \rightarrow R' = a \cdot R^{1/r} + b$ , 并在原点处特别处理

#### ♦ White Point Correction

我们要在 R、G、B 都达到最大值时, 得到白点。但是显示器规范的颜色和白点会产生错误的值。

考虑 SMPTE 规范, 设  $R=G=B=1$ , 则 X 等于  $x_r+x_g+x_b=1.095$ , Y 与 Z 的值为 1.005 和 0.9, 用  $(X+Y+Z)$  来除, 得到色度非要求的色度。

要修正这个缺陷, 先令  $Y=1$ , 由 SMPTE 规范有  $(x, y, z) = (0.3127, 0.3291, 0.3582)$ , 得到  $XYZ_{white}$

$$(4.12) \quad XYZ_{white} = MD(1,1,1)^T$$

$$(4.13) \quad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{white} = \begin{bmatrix} 0.63 & 0.31 & 0.155 \\ 0.34 & 0.595 & 0.07 \\ 0.03 & 0.095 & 0.775 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

求出  $(d_1, d_2, d_3)$  即为修正矩阵

#### ♦ Major difference between RGB, CMYK, Lab and YIQ

CMY: 减性系统, 在加性 (RGB) 系统中黑色没有光,  $RGB=(0,0,0)$ , 在减性 CMY 系统中黑色是墨水的  $C=M=Y=1$  减去所有的光线产生的。

$$(4.24) \quad \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$(4.25) \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

CMYK: 消除不足颜色

C、M、Y 混合起来实际上是图褐色, 真正的黑色的黑墨水实际上比用混合彩色墨水来制作黑墨水便宜。所以一个简单的产生准确的打印机颜色的方法是: 计算三色混合中为黑色的部分, 从颜色比例中去除之, 用真正的黑色加回来, 这被称为“消除不足颜色”

新墨水的规范:  $K \equiv \min\{C, M, Y\}$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} \Rightarrow \begin{bmatrix} C-K \\ M-K \\ Y-K \end{bmatrix}$$

Lab(CIELAB): 利用韦伯定律, 感知到同样的变化必定是相对的。如果变化的比率是相同的, 则感知到的变化是相同的, 而不管是昏暗的光线还是明亮的光线。CIELAB 空间中被量化的是感知到的颜色和亮度的不同。使用  $1/3$  指数定律而不是一个算法。使用三个值, 大致是  $L^*$  亮度,  $a^*$  和  $b^*$  组合起来制造色彩和色调。 $\Delta E$  是色差,  $c^*$  是色度,  $h^*$  是色调角

$$\Delta E = \sqrt{(L^*)^2 + (a^*)^2 + (b^*)^2}$$

$$L^* = 116 \left( \frac{Y}{Y_n} \right)^{(1/3)} - 16$$

$$a^* = 500 \left[ \left( \frac{X}{X_n} \right)^{(1/3)} - \left( \frac{Y}{Y_n} \right)^{(1/3)} \right]$$

$$b^* = 200 \left[ \left( \frac{Y}{Y_n} \right)^{(1/3)} - \left( \frac{Z}{Z_n} \right)^{(1/3)} \right]$$

with  $X_n, Y_n, Z_n$  the XYZ values of the white point.

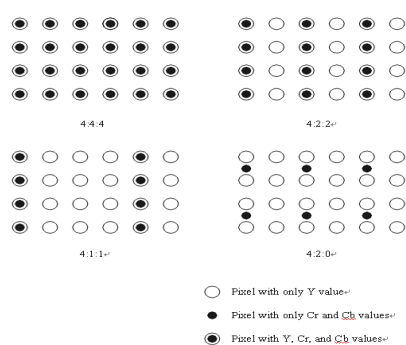
Auxiliary definitions are:

$$chroma = c^* = \sqrt{(a^*)^2 + (b^*)^2}, hueangle = h^* = \arctan \frac{b^*}{a^*}$$

YIQ:用于 NTSC 彩色电视广播中。灰度像素产生零 ( I,Q ) 色度信号。YIQ 是 YUV 的一个版本,使用相同的 Y', 但是 U 和 V 旋转了 33°。YIQ 分解对于形成图像的层次序列更有帮助,能更好的区分颜色的优先次序。

## Chapter 5 Video

- Types of Video
  - 分量视频
  - 复合视频
  - S-Video
- Interlace Scan
- Color subsample



4 : 4 : 4 表示没有使用色度二次采样,每个像素的 Y、Cb、Cr 值被传送,每个都是 4

4 : 2 : 2 表示 Cb 和 Cr 信号是因子为 2 的水平二次采样。

码流:  $Y_0 U_0 Y_1 V_1 Y_2 U_2 Y_3 V_3$

映射出像素点为:  $[Y_0 U_0 V_1] [Y_1 U_0 V_1] [Y_2 U_2 V_3] [Y_3 U_2 V_3]$

4 : 1 : 1 : 水平二次采样因子是 4

码流:  $Y_0 U_0 Y_1 Y_2 V_2 Y_3$

映射出像素点为:  $[Y_0 U_0 V_2] [Y_1 U_0 V_2] [Y_2 U_0 V_2] [Y_3 U_0 V_2]$

4 : 2 : 0 : 水平和垂直方向都进行 2 次采样,因子是 2

下面八个像素为:  $[Y_0 U_0 V_0] [Y_1 U_1 V_1] [Y_2 U_2 V_2] [Y_3 U_3 V_3]$

$[Y_5 U_5 V_5] [Y_6 U_6 V_6] [Y_7 U_7 V_7] [Y_8 U_8 V_8]$

存放的码流为:  $Y_0 U_0 Y_1 Y_2 U_2 Y_3$

$Y_5 V_5 Y_6 Y_7 V_7 Y_8$

映射出的像素点为:  $[Y_0 U_0 V_5] [Y_1 U_0 V_5] [Y_2 U_2 V_7] [Y_3 U_2 V_7]$

$[Y_5 U_0 V_5] [Y_6 U_0 V_5] [Y_7 U_2 V_7] [Y_8 U_2 V_7]$

- NTSC Specification and Modulation

## Chapter 6 Digital Audio

- Nyquist Theorem and Signal to Noise Ratio
- Linear and Non-linear Quantization (u-Law, A-Law)
- PCM and DPCM

## Chapter 7 Lossless Compression Algorithms

- ♦ Entropy and Variable-Length Coding (VLC)
- ♦ Huffman coding algorithm
- ♦ Dictionary-based Coding (LZW algorithm)
- ♦ Arithmetic Coding

## Chapter 8 Lossy Compression Algorithms

- ♦ Quantization
- ♦ Transfer Coding (DCT)

## Chapter 9 Image Compression Standards

- ♦ Block diagram for JPEG encoder

主要步骤：

1. 把 RGB 转换为 YIQ 或 YUV,并且二次采样

因为人类对灰度的视觉敏感度要远远高于对彩色的敏感度。所以 JPEG 利用这个特性采用 4 : 2 : 0 的方案进行色度图像的二次采样。转换颜色空间因为 YIQ 或 YUV 中色度和亮度是分开的，jpeg 对它们分开进行编码。

2. 对图像块进行 DCT 变换

每一个图像划分为 8\*8 的块。将 2D DCT 变换应用到每一块图像  $f(i,j)$ 。输出函数  $F(u,v)$  是每个块的 DCT 变换的系数。

为什么 8\*8 : 大于 8 的数值会使图像在低频的效果更好，但使用 8 能使 DCT 及 IDCT 变换的计算速度更快。

3. 进行量化 (量化是 JPEG 压缩中产生信息丢失的主要原因)

量化是由每个频率除以一个整数，然后取整得到

$$\hat{F}(u,v) = \text{round} \left( \frac{F(u,v)}{Q(u,v)} \right), Q(u,v) \text{ 是量化矩阵, } \hat{F}(u,v) \text{ 是量化后的 DCT 系数}$$

4. 进行 Z 编序和游长编码

Z 形扫描：用 Z 字扫描将 8\*8 的矩阵  $\hat{F}(u,v)$  变为一个 64 元素的向量。(Z 形扫描为了获得稀疏矩阵)

AC 系数的游长编码：AC 系数中每个零串用 { runlength, value } 表示。Runlength 是串里 0 的数目，value 是下一个非 0 系数。(0, 0) 紧跟着最后一个非零 AC 系数。

DC 系数的 DPCM 编码: DPCM 编码后， $d_0 = DC_0$ ,  $d_i = DC_{i+1} - DC_i$

5. 进行熵编码

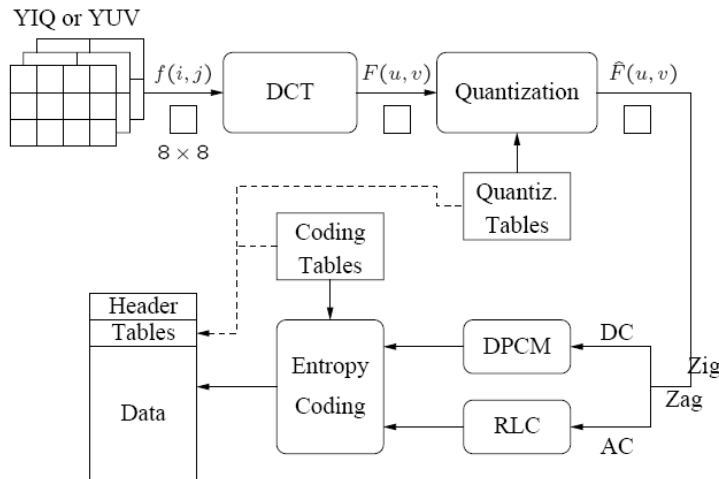
DC 和 AC 系数最后都将进行熵编码

DC 系数的霍夫曼编码 :每一个 DPCM 编码后的 DC 系数可以用 ( SIZE, AMPLITUDE ) 表示，SIZE 表示需多少位表示 DC 系数，AMPLITUDE 是实际使用的位数。例如 ( 3 , 101 )。对 SIZE 进行霍夫曼编码，AMPLITUDE 不用。编码后，定制的编码表可以保存在 JPEG 图像的头部，否则要使用默认的霍夫曼编码。

AC 系数的霍夫曼编码：VALUE 用 ( SIZE, AMPLITUDE ) 表示。得到

Symbol1: (RUNLENGTH,SIZE)和 Symbol2: (AMPLITUDE).

Symbol1 采用霍夫曼编码，Symbol2 不用。



## Chapter 10 Basic Video Compression Techniques

- Motion estimation and Motion compensation

视频压缩算法是基于运动补偿的压缩算法。

运动估计和运动补偿 每一个图像划分成  $N \times N$  的宏块。默认情况亮度图  $N=16$ , 色度图如果采用 4 : 2 : 0 的采样, 则  $N=8$ 。当前帧是目标帧, 要在目标帧和参考帧中最相似的宏块间寻找匹配。参考宏块到目标宏块的位移称作运动向量。

- Search Algorithm

**顺序搜索:** 顺序搜索参考帧中整个  $(2p+1) \times (2p+1)$  大小的窗口。将窗口中每一个宏块逐个像素的和目标帧中的宏块进行比较。使两宏块的 MAD (平均绝对误差) 最小的  $(i, j)$  即为运动向量  $MV(u, v)$

$$\text{运算量} : (2p+1) \cdot (2p+1) \cdot N^2 \cdot 3 \Rightarrow O(p^2 N^3).$$

**2D 对数搜索:** 在搜索窗口中只有 9 个位置被标记为 1, 作为基于 MAD 搜索的起始位置。当 MAD 最小值的位置确定后, 将新的搜索区域中心移动到该位置, 搜索的步长减半。在下次迭代中, 9 个新的位置标记为 2, 依次类推。

$$\begin{aligned} OPS_{per\_second} &= (8 \cdot (\lceil \log_2 p \rceil + 1) + 1) \cdot N^2 \cdot 3 \cdot \frac{720 \times 480}{N \cdot N} \cdot 30 \\ &= (8 \cdot \lceil \log_2 15 \rceil + 9) \times 16^2 \times 3 \times \frac{720 \times 480}{16 \times 16} \times 30 \\ &\approx 1.25 \times 10^9 \end{aligned}$$

运算量 :

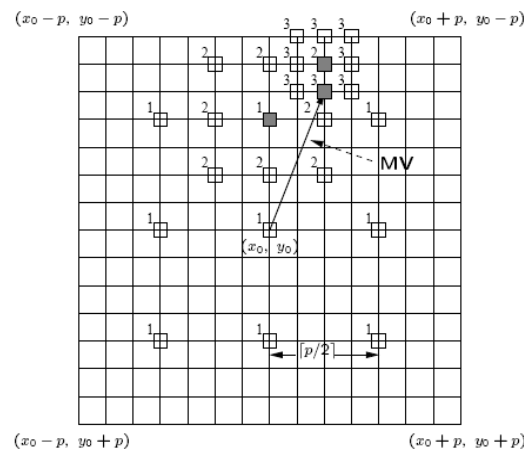


Fig. 10.2: 2D Logarithmic Search for Motion Vectors.

**分层搜索 :** 原始图像为第 0 层, 第 1 层和第 2 层是通过将上一层图像的分辨率减半获得的。初始

搜索从第 2 层开始。初始的运动向量估计值比较粗糙，但是这个值会一层层的进行修正，直到第 0 层。

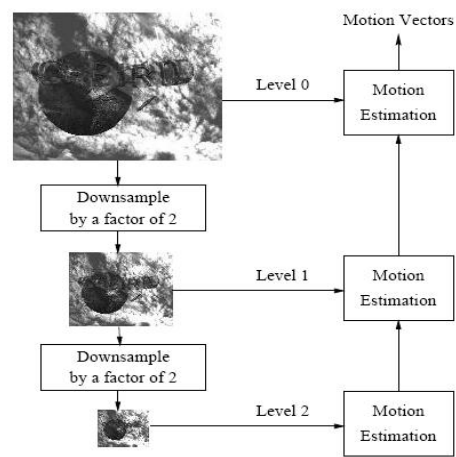
运算量：

OPS\_per\_second=

$$[(2 \cdot \frac{p}{4})^2 (N/4)^2 + 9(N/2)^2 + 9N^2] \cdot 3 \cdot (720 \cdot 480 / N / N) \cdot 3$$

0

$$= 0.51 \cdot 10^9$$



补充：代码：

顺序搜索：

```

♦ begin
♦   min MAD = LARGE NUMBER; /* Initialization */
♦   for i = -p to p
♦     for j = -p to p
♦       {
♦         cur MAD = MAD(i, j);
♦         if cur MAD < min MAD
♦           {
♦             min MAD = cur MAD;
♦             u = i; /* Get the coordinates for MV. */
♦             v = j;
♦           }
♦       }
♦   end

```

2-D 对数搜索：



分层搜索：

- ♦ Block diagram for H.261 encoder  
帧顺序：I 帧 P 帧

H.261 的编码器和解码器：

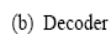
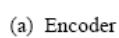


Table 10.3: Data Flow at the Observation Points in H.261 Encoder

Table 10.4: Data Flow at the Observation Points in H.261 Decoder

Current Frame	Observation Point			
	1	2	3	4
$I$	$\tilde{I}$		0	$\tilde{I}$
$P_1$	$\tilde{D}_1$	$P'_1$	$P'_1$	$\tilde{P}_1$
$P_2$	$\tilde{D}_2$	$P'_2$	$P'_2$	$\tilde{P}_2$

具体步骤：

### 1. I 帧编码：

宏块是原图 Y 帧中 16X16 的像素块。因为采用了 4:2:0 的色度二次采样，所以在 Cb 帧和 Cr 帧中，对应为 8X8 大小的区域。因此，一个宏块由 4 个 Y，1 个 Cb，1 个 Cr 的 8X8 的块组成。

对每一个 8X8 的子块，都要进行离散余弦变换 DCT。DCT 系数需要进行量化，最后通过 Z 扫描并进行熵编码。

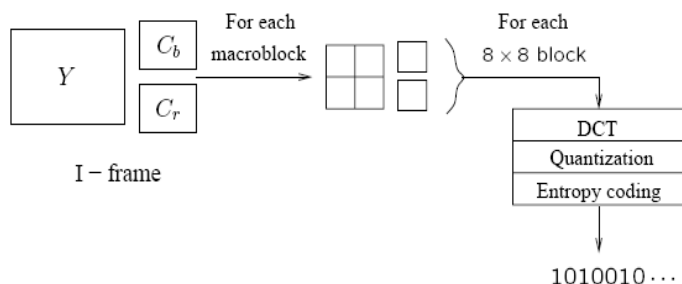
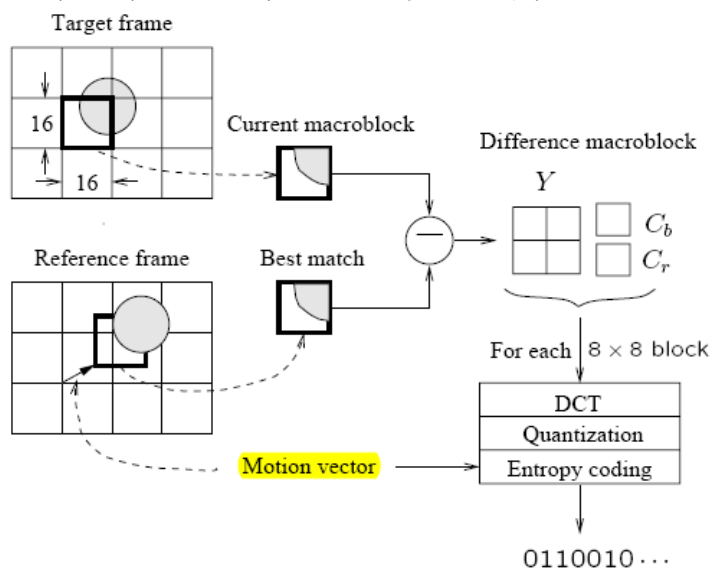


Fig. 10.5: I-frame Coding.

### 2. P 帧预测编码：

对于目标帧中的每一个宏块，通过前面任何一种进行运动向量分配。再用差值宏块测量预测误差。宏块也是由 4 个 Y，1 个 Cb，1 个 Cr 的 8X8 的块组成。这些 8X8 的子块都需要经过 DCT，量化，Z 字在扫描和熵编码四个步骤。运动向量也需要编码。



### 3. 量化：

量化控制就是反馈控制，即当输出缓冲区快要沾满时，量化步长增加，以减少编码数据的大小。编码率控制过程。

## Chapter 11 MPEG Video Coding

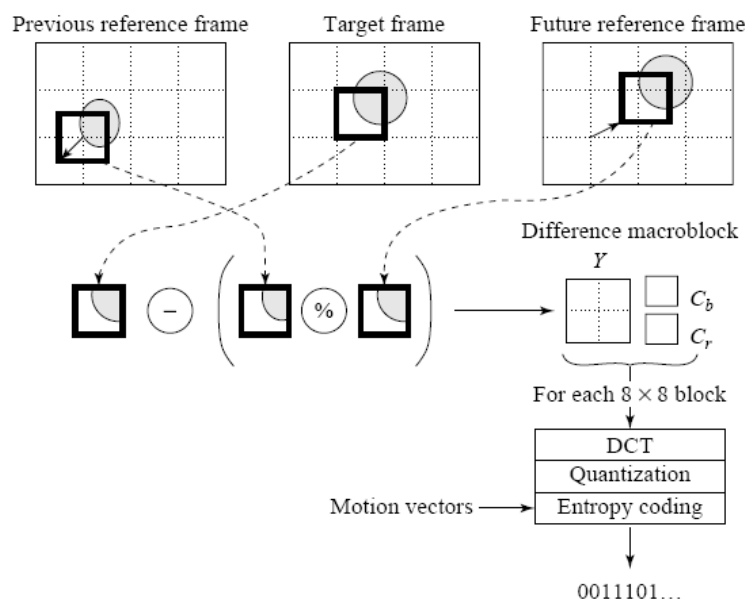
### ◆ MPEG 1

#### ■ B frame

B 帧：

MPEG 引入了第三类帧，B 帧，以及相应的双向运动步长。除了向前预测，还用到了向后预测，也就是此时用来进行匹配的宏块是从视频序列中未来的 I 帧或者 P 帧获得的。这样，B 帧的每一个宏块指向两个运动向量，一个向前预测的来，一个向后预测的来。

如果两个方向的匹配成功,那么两个运动向量都被发送,在与目标宏块进行比较产生预测误差之前,将与两个相应的匹配宏块取平均。如果只有一个参考帧的匹配时成功的,那么只有一个运动向量及其相应的宏块会被使用,或是向前预测的或者是向后预测的。



## ■ Major Differences from H.261

### 1. 运动补偿技术:

**H.261** 基于运动补偿的视频编码原理:在运动估计中,会为目标 P 帧的每个宏块分配一个从之前已编码的 I 帧或 P 帧的宏块中选出的与他最匹配的宏块,这称为预测。当前宏块与匹配的宏块之间的差称为预测误差。这个预测误差将被传送到 DCT 和接下来的编码步骤。向前预测。

**MPEG:**MPEG 引入了第三类帧, B 帧,以及相应的双向运动步长。除了向前预测,还用到了向后预测,也就是此时用来进行匹配的宏块是从视频序列中未来的 I 帧或者 P 帧获得的。这样, B 帧的每一个宏块指向两个运动向量,一个向前预测的来,一个向后预测的来。如果两个方向的匹配成功,那么两个运动向量都被发送,在与目标宏块进行比较产生预测误差之前,将与两个相应的匹配宏块取平均。如果只有一个参考帧的匹配时成功的,那么只有一个运动向量及其相应的宏块会被使用,或是向前预测的或者是向后预测的。

**Mpeg:** 由于其编码器和解码器不能用于没有后续 P 帧或 I 帧的 B 帧宏块,所以实际的译码和传输顺序和视频的实时顺序是不同的。

2. **源格式:** H.261 只支持 CIF 和 QCIF 两类源格式。MPEG1 支持 SIF 等其他格式。

3. **宏块片:** 与 H.261 中的 GOB 不同,一幅 MPEG1 图片可以被分为一个或者多个宏块片,这比 GOB 灵活。只要可以填满整个图片,他们可以包含一幅图片中的可变数目的宏块并可以开始结束于任何位置。每个宏块片都独立编码。为码率提供了灵活性。宏块片的内容对于差错恢复很重要,每个宏块片有唯一的宏块片其实嘛。Mpeg 中的宏块片类似于 H.261 中的 GOB:他处于 MPEG 曾侧结构的最底层,不需要将位流中整个变长编码组解码就可以完全恢复。

4. **量化.** MPEG1 的量化对于帧间编码和帧内编码采用不同的量化表。在一个宏块中,用于帧内编码的量化器个数是不同的。这与 H.261 不同,对 H.261,一个宏块内用于 AC 系数的量化器个数是不变的。

5. 为了增加运动步长预测的精度,减小误差,MPEG1 允许运动向量具有半像素精度。

6. MPEG1 支持 I 帧和 P 帧之间有较大差距，因此会有一个较大的运动向量搜索范围。
7. MPEG1 位流允许随即访问。

# ◆ MPEG 2

## ■ Profiles and Levels

规格和等级：规格来自应用要求，等级由规格来决定。

Table 11.5: Profiles and Levels in MPEG-2

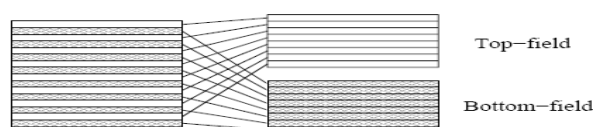
Level	Simple Profile	Main Profile	SNR Scalable Profile	Spatially Scalable Profile	High Profile	4:2:2 Profile	Multiview Profile
High		*			*		
High 1440	*	*		*	*		
Main		*	*		*	*	
Low		*	*		*		*

Table 11.6: Four Levels in the Main Profile of MPEG-2

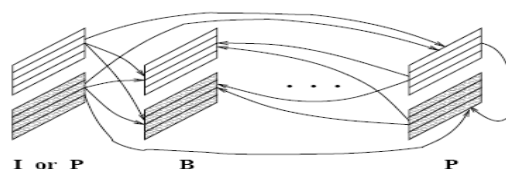
Level	Max Resolution	Max fps	Max Pixels/sec	Max coded Data Rate (Mbps)	Application
High	$1,920 \times 1,152$	60	$62.7 \times 10^6$	80	film production
High 1440	$1,440 \times 1,152$	60	$47.0 \times 10^6$	60	consumer HDTV
Main	$720 \times 576$	30	$10.4 \times 10^6$	15	studio TV
Low	$352 \times 288$	30	$3.0 \times 10^6$	4	consumer tape equiv.

## ■ Prediction modes

域预测和帧预测：隔行交织扫描：



(a) Frame-picture vs. Field-pictures



(b) Field Prediction for Field-pictures

5 种预测模式：参考课本

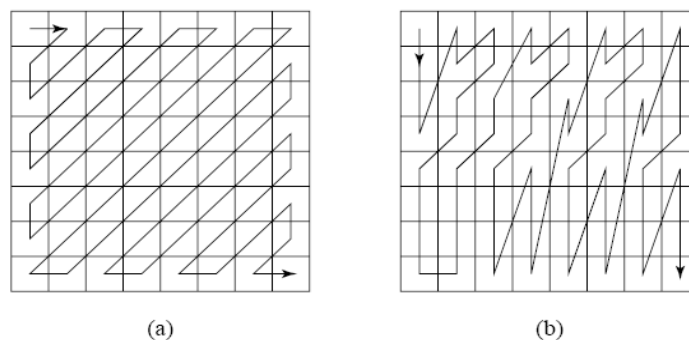
- 1. **Frame Prediction for Frame-pictures:** Identical to MPEG-1 MC-based prediction methods in both P-frames and B-frames.
- 2. **Field Prediction for Field-pictures:** A macroblock size of  $16 \times 16$  from Field-pictures is used. For details, see Fig. 11.6(b).
- 3. **Field Prediction for Frame-pictures:** The top-field and bottom-field of a Frame-picture are treated separately. Each  $16 \times 16$  macroblock (MB) from the target Frame-picture is split into two  $16 \times 8$  parts, each coming from one field. Field prediction is carried out for these  $16 \times 8$  parts in a manner similar to that shown in Fig. 11.6(b).
- 4. **MC for Field-pictures:** Each  $16 \times 16$  macroblock (MB) from the target Field-picture is split into top and bottom  $16 \times 8$  halves. Field prediction is performed on each half. This generates two motion vectors for each  $16 \times 16$  MB in the P-Field-picture, and up to four motion vectors for each MB in the B-Field-picture. This mode is good for a finer MC when motion is rapid and irregular.

- **5. Dual-Prime for P-pictures:** First, Field prediction from each previous field with the same parity (top or bottom) is made. Each motion vector **mv** is then used to derive a calculated motion vector **cv** in the field with the opposite parity taking into account the temporal scaling and vertical shift between lines in the top and bottom fields. For each MB the pair **mv** and **cv** yields two preliminary predictions. Their prediction errors are averaged and used as the final prediction error.

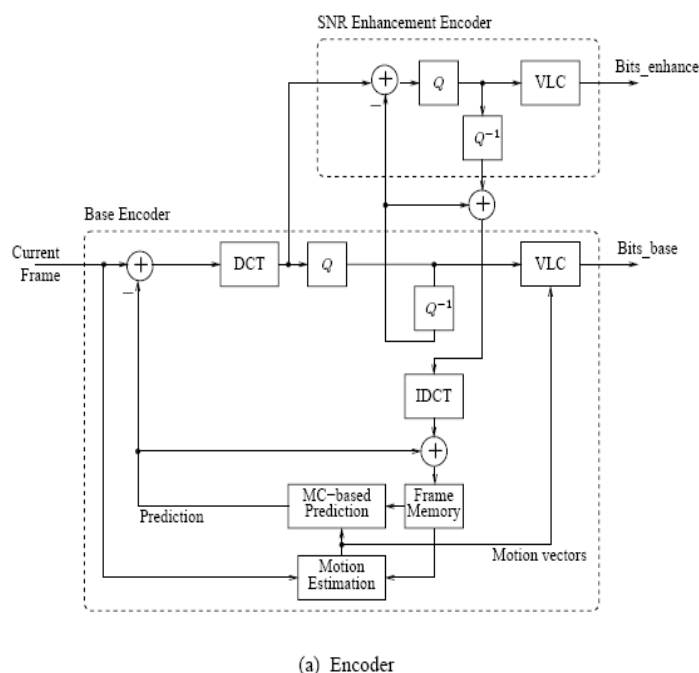
This mode mimics B-picture prediction for P-pictures without adopting backward prediction (and hence with less encoding delay).

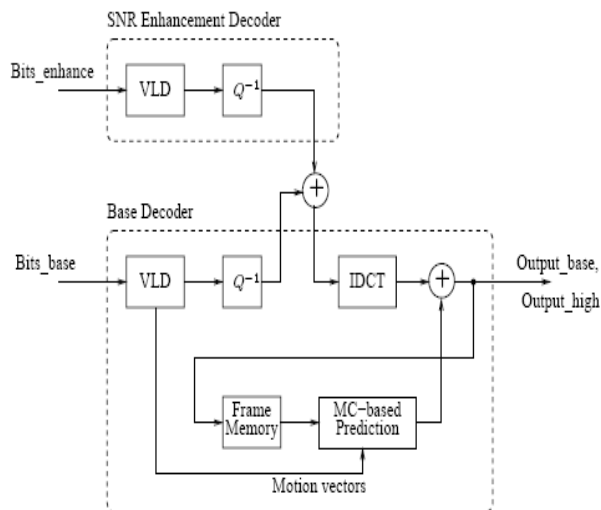
This is the only mode that can be used for either Frame-pictures or Field-pictures.

- **Alternate Scan and Field\_DCT**  
交替扫描和域 DCT :



- **MPEG-2 Scalabilities**  
SNR 可伸缩性:





(b) Decoder

空间可伸缩性：

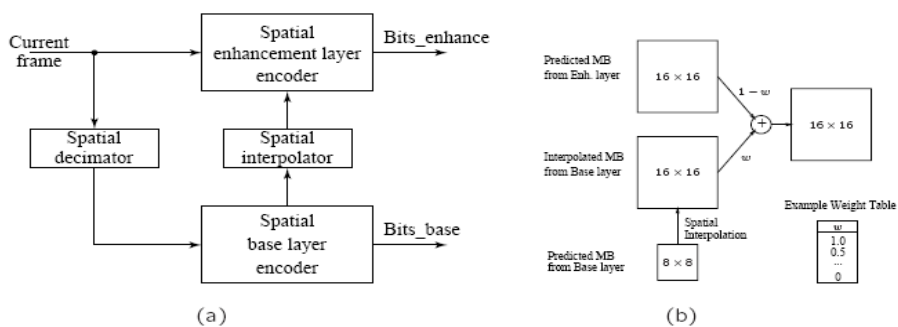
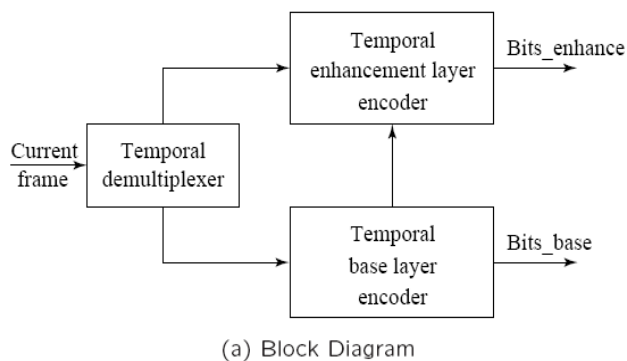


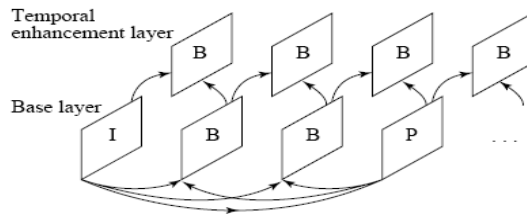
Fig. 11.9: Encoder for MPEG-2 Spatial Scalability. (a) Block Diagram. (b) Combining Temporal and Spatial Predictions for Encoding at Enhancement Layer.

时间可伸缩性

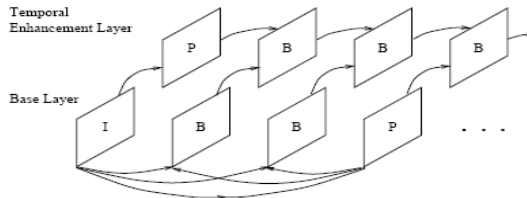


(a) Block Diagram

编码器



(b) Interlayer Motion-Compensated (MC) Prediction.



(c) Combined MC Prediction and Interlayer MC Prediction

## Chapter 13 Voice Coding

### ♦ Vocoders

■ **Vocoders** — voice coders, which cannot be usefully applied when other analog signals, such as modem signals, are in use.

- ♦ — concerned with modeling speech so that the salient features are captured in as few bits as possible.
- ♦ — use either a model of the speech waveform in time (LPC (Linear Predictive Coding) vocoding), or ... →
- ♦ — break down the signal into frequency components and model these (channel vocoders and formant vocoders).

■ Vocoder simulation of the voice is not very good yet.

■ **Channel Vocoder:**

- ♦ Vocoders can operate at low bit-rates, 1–2 kbps. To do so, a *channel vocoder* first applies a filter bank to separate out the different frequency components:

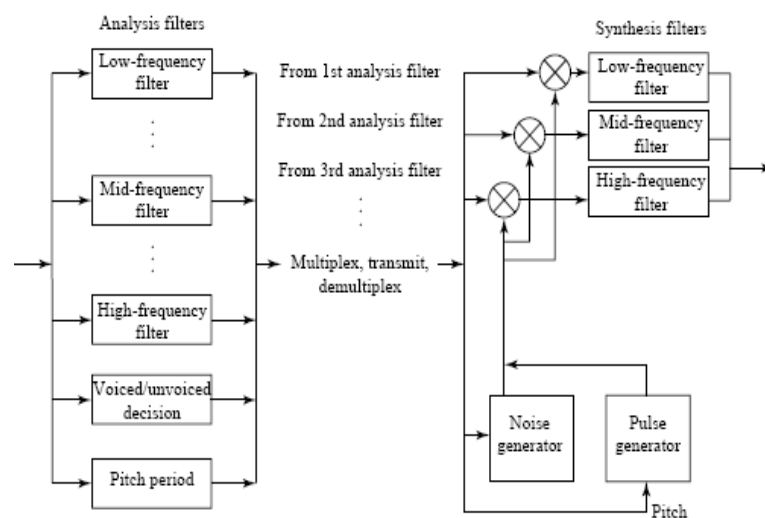


Fig 13.4: Channel Vocoder

- ♦ Due to *Phase Insensitivity* (i.e. only the energy is important):
- ♦ — The waveform is "rectified" to its absolute value.



- The filter bank derives relative power levels for each frequency range.
- A subband coder would not rectify the signal, and would use wider frequency bands.
- A channel vocoder also analyzes the signal to determine the general pitch of the speech (low — bass, or high — tenor), and also the *excitation* of the speech.
- A channel vocoder applies a vocal tract transfer model to generate a vector of excitation parameters that describe a model of the sound, and also guesses whether the sound is *voiced* or *unvoiced*.

#### ■ Format Vocoder

- Formants:** the salient frequency components that are present in a sample of speech, as shown in Fig 13.5.
- Rationale: encoding only the most important frequencies

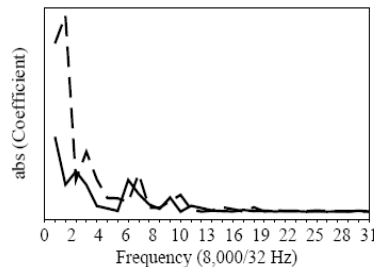


Fig. 13.5: The solid line shows frequencies present in the first 40 msec of the speech sample in Fig. 6.15. The dashed line shows that while similar frequencies are still present one second later, these frequencies have shifted.

- Hybrid Coder

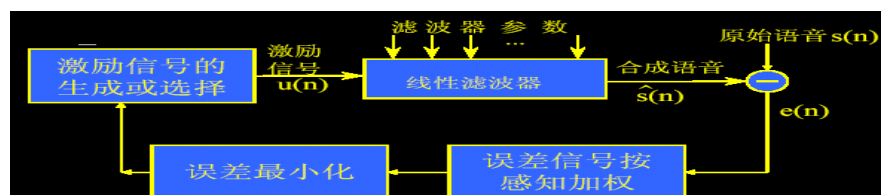
#### 问题：

- 参数编码的不足
  - 参数编码在降低码率方面有很大突破，但语音质量尚不理想。原因是：语音生成模型中的激励信号处理过于简单。
    - 不是清音就是浊音。实际上可能是两者混合；
    - 浊音的激励信号是周期性的。实际上是准周期性。
  - 改进思路：设计更好的激励信号。

#### 设计思想：

- 原理

- 使用 Analysis-by-Synthesis 来改进参数编码，其中声道滤波器模型仍与 LPC 相同。但不使用两状态（有声/无声）模型作为滤波器的输入激励信号，而是增加反馈，调节激励信号  $u(n)$ ，使语音输入信号  $s(n)$  与重建的语音信号误差  $e(n)$  为最小。
- 编码器通过“合成”许多不同的近似值来“分析”输入语音信号，所以称为 AbS。



- 激励信号  $u(n)$  的生成方法

- 多脉冲线性预测编码 MPLPC(multiple pulse LPC)
- 等间隔脉冲激励 RPE(regular-pulse excited)
- 码激励线性预测 CELP(code excited linear predictive)

## Chapter 14 Mpeg Audio

- ♦ Psychoacoustics
- ♦ MPEG Audio
  - ♦ **MPEG audio compression** takes advantage of psychoacoustic models, constructing a large multi- dimensional lookup table to transmit masked frequency components using fewer bits
  - ♦ **MPEG Audio Overview**
    - 1. Applies a filter bank to the input to break it into its frequency components
    - 2. In parallel, a psychoacoustic model is applied to the data for bit allocation block
    - 3. The number of bits allocated are used to quantize the info from the filter bank — providing the compression
  - ♦ MPEG audio offers three compatible *layers* :
    - — Each succeeding layer able to understand the lower layers
    - — Each succeeding layer offering more complexity in the psychoacoustic model and better compression for a given level of audio quality
    - — each succeeding layer, with increased compression effectiveness, accompanied by extra delay
  - ♦ The objective of MPEG layers: a good tradeoff between quality and bit-rate
  - ♦ MPEG LAYERS
    - Layer 1 quality can be quite good provided a comparatively high bit-rate is available
      - ♦ — Digital Audio Tape typically uses Layer 1 at around 192 kbps
    - Layer 2 has more complexity; was proposed for use in Digital Audio Broadcasting
    - Layer 3 (MP3) is most complex, and was originally aimed at audio transmission over ISDN lines
    - Most of the complexity increase is at the encoder, not the decoder — accounting for the popularity of MP3 players

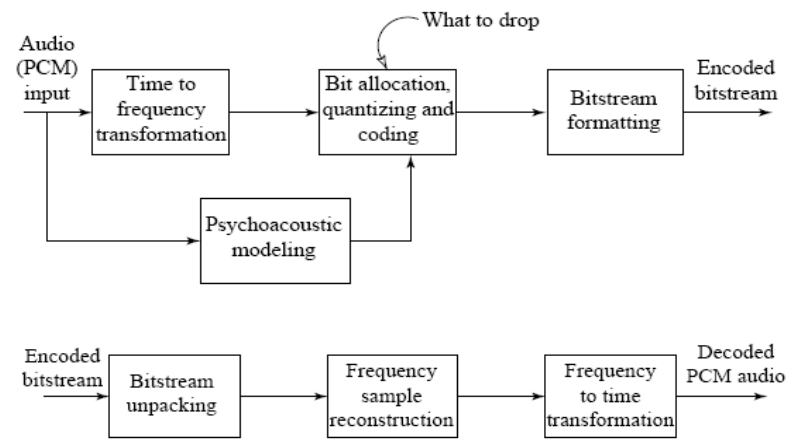


Fig. 14.9: Basic MPEG Audio encoder and decoder.