国防科学技术大学

硕士学位论文

基于H.264的视频压缩关键技术的研究

姓名：徐林玲

申请学位级别：硕士

专业：信息与通信工程

指导教师：林嘉宇

20031201

摘要

ITU-T 和 MPEG 联合组成的 JVT(Joint Video Expert Team)于 2003 年 3 月正式发布视频压缩编码国际标准 H.264。本文基于 H.264 标准研究了视频压缩的若干关键技术，包括：研究 H.264 编码器各模块的计算复杂度，由于 H.264 的主要目标之一是在同样的保真度条件下，提高编码效率，而要提高压缩比，就会相应增加了算法的复杂度，从而对算法的实际应用很不利，因此本文对 H.264 中的关键模块整数变换模块进行研究改进；H.264 主要面对 IP 和无线环境的，抗误码性能是关键，本文进一步对基于 H.264 的视频传输抗误码技术进行了研究。最后用软件实现了 H.264 视频编解码演示系统。

本论文完成的主要工作如下：

- 对视频图像编码发展史、视频图像压缩编码技术及视频编码国际标准作了简单介绍。给出了图像/视频编码领域的基本概念、发展历史和当前状况，着重介绍了视频编码国际标准 H.264 的发展情况，及其目前相关研究动态。阐明了本文选题的理论依据，并为随后展开深入的研究提供了线索与思路。
- 介绍了国际视频压缩编码标准 H.264 的基本框架，实现了 H.264 视频编解码演示系统并对编码器主要模块的复杂度进行了分析。本文以个人计算机为应用平台，利用纯软件进行视频的编解码，实现了一个从图像源处理、编码、传输、解码及显示全过程的实验演示系统。
- 对 H.264 中占编码运算量 17%左右的整数变换和量化模块进行研究及改进。将浮点 DCT、整数变换和无乘法二进制 DCT(binDCT)在结构和性能上进行比较分析，并分析了如何考虑 binDCT 各指标之间的折衷，研究了如何取用不同的 binDCT 系数以满足不同指标的需求，可实现编码质量和计算量的协调。
- 研究了基于 H.264 的视频传输抗误码技术，在 H.264 中引入了基于算术编码的具有连续检错和自动重传能力的综合抗误码方案，可在增加冗余量和提高检错性能之间取得较好的折衷，获得较好的压缩效果和抗信道误码能力。

【关键词】H.264　压缩编码　整数变换　binDCT　视频传输　抗误码技术

Abstract

In 2003, the Joint Video Team (JVT) formed by the ITU-T Video Coding Experts Group (VCEG) and the ISO Motion Picture Experts Group (MPEG) published a video compression coding international standard, which will be two identical standards: ISO MPEG4 Part 10 of MPEG4 and ITU-T H.264. The "official" title of the new standard is Advanced Video Coding (AVC); however, it is widely known by its old working title H.26L and by its ITU document number H.264.

H.264 offers significantly better video compression efficiency than previous ITU-T standards and MPEG standards, but its computational complexity is too large for implementation. In this paper, some key issues in video coding and decoding based on H.264, including complexity for implementation, integer transform and error resiliency are studied. Finally a video coding and decoding system DEMO in software is implemented.

The main work of this paper is just as below:

- Analyze the complexity for implementation of the key modules of H.264 encoder.
- Compare and analyze the DCT, the integer transforms and the binDCT, the experimental results show that the binDCT with right parameters in H.264 can not only reduce the complexity but also be more efficient.
- Introduce the usage of an ARQ scheme based on Continuous Error Detection (CED) technique, to improve error resiliency of arithmetic coding adopted by H.264. Tradeoff between added redundancy and error-detection performance is considered, achieving significant improvement in error resiliency ability of H.264.

【 keywords 】 H.264 compression coding integer transform binDCT video communication error resiliency

独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科学技术大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目：基于 H.264 的视频压缩关键技术的研究

学位论文作者签名：徐林玲　　日期：2003 年 11 月 12 日

学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密学位论文在解密后适用本授权书。）

学位论文题目：基于 H.264 的视频压缩关键技术的研究

学位论文作者签名：徐林玲　　日期：2003 年 11 月 12 日

作者指导教师签名：林嘉宇　　日期：2003 年 11 月 12 日

图 目 录

表 目 录

第一章 绪论

§ 1.1 运动图像压缩编码技术的发展历史与现状

1.1.1 运动图像压缩编码技术的发展历史

1948 年，Shannon 和他的两个学生 Oliver 与 Pierce 联合发表了对电视信号进行脉冲编码调制（PCM）的论文[1]，这标志着数字图像压缩编码技术的开端。1969 年在美国举行的首届“图像编码会议”表明图像编码以独立的学科跻身于学术界。半个世纪以来，图像编码技术早已走出实验室，广泛应用于信息社会的各个领域。

1968 年 H.C. Andrews 等人提出了变换编码[2]的方法，采用的是二维离散傅里叶变换（2D-DFT），此后相继出现了其他的变换编码方法，其中包括沃尔什－哈达玛（Walsh-Hadamard）变换[3]、K-L 变换、二维离散余弦变换（DCT）变换等；70 年代开始进行帧间预测编码的研究；80 年代开始对运动补偿（MC）所用的运动估值（ME）算法进行研究；对模型编码的研究也是始于 80 年代初。

国际电话电报咨询委员会（CCITT，现并入国际电信联盟 ITU）于 1988 年形成草案、1990 年通过主要针对可视电话和电视会议的 H.261 标准，这是图像编码技术走向实用化的重要一步，也是图像编码 40 年研究成果的结晶。90 年代相继出现的 MPEG-1、MPEG-2、MPEG-4 和 H.263 等都是在 H.261 的基础上的发展和改进的。这些国际标准普遍采用的混合编码技术也是当今最实用的高效编码方法，得到了广泛的推广和应用，业已成为当今图像编码方法的主流。

1.1.2 运动图像压缩编码技术

图像压缩编码从 1948 年电视信号数字化提出以来，经过了 50 多年的发展，不仅在理论上取得了重大进步，而且在实际中得到了广泛地应用。

表征图像/视频信息的数据量之大是惊人的，而这些数据往往是高度相关的，这些相关性引起了信息的冗余。对于视频图像来说，除了时间上和空间上的冗余外，还存在信息熵冗余、结构冗余、知识冗余、视觉冗余等[4]。图像编码的目的是去掉庞大数据中的冗余信息（去掉数据间的相关性），保留相互独立的信息分量，在保证一定重建质量的前提下，以尽量小的比特数表征图像的信息。

1.1.2.1 视频图像编码方法

传统的第一代压缩编码技术以香农（Shannon）信息论为基础，在标准化、产业化过程中已取得了巨大成功。目前建立在一种更广义信息论基础上的、基于内容的第二代编码技术也正在逐步走向成熟。第一代编码方法主要有以下几种编码方法[5]。

1、统计编码

数据压缩技术的理论基础是信息论。根据信息论的原理，可以找到最佳数据压缩编码方法，数据压缩的理论极限是信息熵。如果要求在编码过程中不丢失信息量，则要求保存信息熵，这种信息保持编码就是熵编码。它是建立在随机过程的统计特性基础上的。图像编码中应用最广的有哈夫曼（Huffman）编码、算术编码、行程编码（RLC，Run Length Code）等三种。

2、预测编码

预测编码方法是一种较为实用且被广泛采用的一种压缩编码方法，其理论基础主要是现代统计学和控制论。原理是从相邻像素之间的相关性特点考虑，比如当前像素的灰度或色度信号，数值上与其相邻像素总是比较接近，除非处于边界状态。那么当前像素的灰度或色度信号的数值，可用前面已出现的像素值进行预测估计，得到一个预测值，将实际值与预测值求差，对这个差值信号进行编码、传输，这种编码方法称为预测编码。一般有线性预测和非线性预测两种，其中线性预测编码方法也称为差值脉冲编码调制法。

3、变换编码

变换编码不是直接对空域图像信号编码，而是首先将空域图像信号映射到另一个正交矢量空间（变换域），产生一批变换系数，然后对这些变换系数进行编码处理。在编码端将原始图像分割成若干个子图像块，每个子图像块送入正交变换器作正交变换，变换器输出变换系数经滤波、量化、编码后送信道传输到接收端，接收端作解码、逆变换、综合拼接，恢复出空域图像。

变换编码中，正交变换是影响编码效率的关键。正交变换中综合性能优良的有：离散余弦变换（DCT）、小波（Wavelet）变换等。K-L 变换是均方误差准则下的最佳正交变换，当图像相邻像素间的相关系数接近 1 时，K-L 变换的基函数接近 DCT 变换的基函数。对于通常的图像，人们将 DCT 变换视为准最佳正交变换，由于 DCT 变换存在快速算法，且易于硬件实现，被广泛应用于多种图像/视频编码国际标准中[6,7,8]。

上述统计编码、预测编码、变换编码的组合可形成混合编码，是目前绝大多数视频图像编码标准的基础，JPEG、MPEG-x、H.26x 等都采用集预测估计、DCT 变换和统计编码于一体的混合编码方法。这些编码方法都是依据视频图像固有的特性进行压缩的，但仍远未成熟。伴随着感知生理-心理学的发展，人们越来越清楚的认识到：人的视觉感知特点和统计意义上的信息分布并不一致，统计上需要更多的信息量才能表征的特征，对视觉感知可能并不重要，从感知角度来讲，无需详细表征这部分特征。这时压缩技术的研究就突破

了传统信息论的框架，注重对感知特性的利用，即利用所谓的“感知熵”理论，使得压缩效率得以极大提高[9]。

随着数学理论，如小波变换、分形几何理论、数学形态学等以及模式识别、人工智能、生理心理学等学科的发展，高效的第二代压缩编码方法相继产生。如分形编码、基于模型的编码、基于对象的编码等，这里不再详细赘述。

1.1.2.2 评价视频质量的性能指标

实际上在视频编解码、视频传输过程中都可能引入失真。如何客观的度量这种失真，并且使客观度量结果和人的视觉感受一致，是视频编码、视频通信领域的一个重要问题。一般来说，视频/图像质量评价方法分为主观评价方法和客观评价方法。

对于静止图像的客观评价是用重建图像偏离原始图像的误差来衡量的，常用的有均方误差（MSE）和峰值信噪比（PSNR）。

（1）均方误差的定义为：

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[f(i,j) - f'(i,j)]^2 \tag{1.1}$$

式中

M，N ——图像宽和高的像素点数；

$f(i,j)$ ——原始图像的灰度值；

$f'(i,j)$ ——重建图像的灰度值。

（2）峰值信噪比的定义为：

$$PSNR = 10\log_{10}\frac{255\times 255}{MSE} \tag{1.2}$$

公式（1.1）和（1.2）看起来直观、严格，但这种计算独立于图像的内容，评价的结果往往和人的视觉感受不一致。这是因为 MSE 和 PSNR 是从总体上反映原始图像和重建图像的差别，并不能反映局部，比如较少的像素点有较大的灰度差和较多的像素点有较小的灰度差等各种情况。

而主观评价方法是由评价者直接对一幅图像或一段视频进行观察，从感觉上去度量其失真程度，给出质量评价级别，对所有评价者给出的分数进行加权平均，所得结果即为主观评价结果，如表 1.1 所示。这种评价结果必然符合人的视觉感受。但这种主观感受一方面不能用数学模型对其进行描述，不能直接用于视频/图像压缩编码过程中的质量评价与控制；另一方面，主观评价容易受到评价者的主观因素的影响，如年龄、性格、教育程度、背景以及评价时的心情等。因而这种评价方法在实际应用中受到了很大的限制，甚至根本不适合某些应用场合。

表1.1 主观测试分级标准

级别	质量	损伤	比较
5	优	不能觉察	好得多
4	良	刚觉察不讨厌	相同
3	中	有点讨厌	略坏
2	次	很讨厌	坏
1	劣	不能用	坏得多

人的视觉能力有限，在一定的条件下，一定的视频失真人眼根本无法觉察出来，所以对视频质量的主观评价和客观度量不同之处在于人眼的视觉特性，其中之一是视觉掩蔽。由于视觉掩蔽效应，一定的视频失真被掩蔽掉了，从主观评价的角度看，这部分损伤并没有计入视频失真之内。而从客观度量评价的角度看，任何偏离原始视频的误差都认为是失真，这是主客观评价不一致的主要原因。

因此，对视频失真进行评价时就要同时考虑到客观度量和人的视觉感受相一致，所以采用了一种基于运动特征的视频质量评价方法[10]。

视频与图像不同，它除了具有空间域的特性外，还具有时间域的特性。将空间域和时间域的特性分别用视频清晰度和视频流畅度表示。视频清晰度是指每一单个帧的清楚程度；视频流畅度是指视频序列的连续程度。为了叙述方便，假设编码器的输入原始图像序列是以恒定的高速帧频采集得到的，采集间隔定义为 T。一般情况下，不是所有帧都被编码输出。k 表示被编码输出的图像帧编号，N 表示被编码图像的总帧数，第 k 帧和第 k-1 帧之间相隔 FT_k 帧原始未编码图像，则第 k 帧和第 k-1 帧之间相隔时间为 $FT_k \cdot T$ 。

（1）视频清晰度评价

第一步，在一帧图像的清晰度评价中引入运动特征。即以现有静止图像评价策略为基础，赋予运动较为剧烈的图像块更高的权重。以 PSNR 做静止图像评价策略为例，下标 k 表示图像的帧号，将有如下公式：

$$ss_k = 10 \cdot \log_{10} \frac{255 \cdot 255}{\frac{1}{N_{MB}} \sum_{i,j} MA_k(i,j) \cdot Diff_k(i,j)} \tag{1.3}$$

式中

$$Diff_k(i,j) = \frac{1}{M_p \cdot N_p} \sum_{m=M_p \cdot i}^{M_p \cdot (i+1)} \sum_{n=N_p \cdot j}^{N_p \cdot (j+1)} \left| f_k(m,n) - f_{k-1}(m,n) \right|^2 \tag{1.4}$$

$$MA_k(i,j) = \sqrt{dx_k^2(i,j) + dy_k^2(i,j)} \tag{1.5}$$

MA_k 表示第 k 帧图像的运动剧烈程度，$f_k(m,n)$ 表示第 k 帧图像（m，n）处像素的灰度值，M_p，N_p 分别表示一个图像块的宽和高的像素点数，（dx_k，dy_k）表示第 k 帧中第（i，j）个图像块的运动矢量。

如果对所有 $i,j,k,MA_k(i,j)$ 都为 1，那么清晰度的评价就退化为静止图像的质量评价方法，即表示峰值信噪比。而当 $MA_k(i,j)$ 按照（1.5）取值时，ss_k 的值将更符合人眼视觉感受。视频清晰度可由（1.6）得到。

$$PS=\frac{1}{N}\sum_{k=1}^{N}(\frac{1}{MA_k}\cdot FT_k\cdot ss_k) \tag{1.6}$$

它包括两方面的含义：一方面，运动较为剧烈的图像帧的清晰度在整段视频清晰度评价中的作用相对较弱，因此权值低一些；另一方面，视频重放中占用时间较长的图像帧的清晰度在整段视频清晰度评价中的作用相对较强。

（2）视频流畅度评价

在固定帧频视频编码方案中，帧频参数可以在一定程度上描述画面的流畅程度；而这种简单方法，对可变帧频的视频不大适合。

在视频图像运动较为剧烈的视频段，前后帧差别较大，人眼无法辨清图像的细节部分，而对图像中对象的运动是否流畅却比较敏感；而在图像运动相对较弱的视频段，前后帧之间差别不大，人眼更关注的是每一帧的清晰度。因此在运动相对剧烈的视频段，可以适当的降低清晰度从而节约带宽来相对提高帧频，在运动较为缓和的部分，适当降低帧频提高每一帧的清晰度，这样更符合人眼的主观感受。为此，用下面的公式计算一段视频的流畅度：

$$PT=\frac{1}{N}\sum_{k=1}^{N}(MA_k\cdot FT_k) \tag{1.7}$$

PT 越小说明画面越流畅。

（3）视频质量综合评价准则

上面分别从清晰度、流畅度两方面给出了视频序列的质量评价方法。利用下式，我们可以得出对视频信号的综合评价指标。

$$P=W_s\cdot PS+W_T\cdot\frac{1}{PT} \tag{1.8}$$

其中，W_s 和 W_T 是清晰度和流畅度在综合评价中所占权重，$W_s,W_T\in[0,1]$，且 $W_s+W_T=1$。W_s 和 W_T 的选取有一定主观性，可根据自己对清晰度和流畅性的需要，调整这两个权重，从而在资源有限的环境下，获得用户最满意的效果。

1.1.2.3 视频压缩编码的国际标准[11,12,13]

近年来，国际标准化组织（ISO）和国际电信联盟（ITU）先后制定了多个图像、视频

编码的标准。如二值图像编码标准（JBIG）、静止图像编码标准（JPEG 和 JPEG2000），以及各种面向视频序列的压缩标准。下面简单介绍视频压缩编码的标准。

（1）H.261 标准

H.261 建议是国际电话电报咨询委员会（CCITT，现并入国际电信联盟 ITU）于 1990 年通过的，其应用目标主要针对可视电话和电视会议。它的传输速率为 p×64Kbps，可根据传输线路的带宽来调整图像质量，以达到刚好吻合的程度。H.261 建议是视频编码的经典之作。

（2）H.263 标准

H.263 是 ITU-T 于 1996 提出的作为 H.324 终端使用的视频编解码建议。H.263 在 H.261 建议的基础上，将运动矢量的搜索增加为半像素点搜索；同时又增加了无限制运动矢量、基于语法的算术编码、高级预测技术和 P、B 帧编码等四个高级选项；从而达到了进一步降低码速率和提高编码质量的目的，使其更适合于 IP 视频会议、可视电话等应用。

H.263 视频编码标准是专为中高质量运动图像压缩所设计的低码率图像压缩标准。与 H.261 的 p×64Kbps 的传输码率相比，H.263 的码率更低，单位码率可以小于 64Kbps，且支持的原始图像格式更多，包括了在视频和电视信号中常见的 QCIF，CIF，HDTV，ITU-R601，ITU-R709 等等。

H.263 采用运动视频编码中常见的编码方法，将编码过程分为帧内编码和帧间编码两个部分。H.263 的编码速度快，其设计编码延时不超过 150ms；码率低，在 512K 乃至 384K 带宽下仍可得到相当满意的图像效果，十分适用于需要双向编解码并传输的场合（如：可视电话）和网络条件不是很好的场合（如：远程监控）。

（3）MPEG-1 标准

MPEG-1 标准是 MPEG 组织制定的第一个视音频压缩国际标准，1990 年 12 月完成了标准草案，最后文本于 1992 年 11 月出版。主要为视频存储媒体如 VCD 制定，该标准能够适应变码流的处理，其主要目的是把 221Mbps 的 NTSC 图像压缩到 1.2Mbps，压缩率为 200：1。这是图像压缩的工业认可标准。它可针对 SIF 标准分辨率（对于 NTSC 制为 352×240；对于 PAL 制为 352×288）的图像进行压缩，传输速率为 1.5Mbps，每秒播放 30 帧，具有 CD 音质，质量级别基本与 VHS（广播级录像带）相当。MPEG 的编码速率最高可达 4-5Mbps，但随着速率的提高，其解码后的图像质量有所降低。MPEG-1 标准主要包括系统、视频编码、音频编码等几个部分。

应用 MPEG-1 技术最成功的产品非 VCD 莫属了，VCD 作为价格低廉的影像播放设备，得到广泛的应用和普及。MPEG-1 也被用于数字电话网络上的视频传输，如非对称数字用户线路（ADSL），视频点播（VOD），以及教育网络等。

（4）MPEG-2 标准

MPEG-2 标准的制定开始于 1990 年 7 月，与此同时 ITU 也开始采用 ATM 异步传输模

式的视频编码建议（H.262）的制定工作，后来 MPEG 和 ITU 决定联手开展 MPEG-2 的标准制定工作，并于 1994 年 11 月正式推出。MPEG-2 能够提供传输率在 3Mbps～10Mbps 之间，分辨率达 720×486 或 720×576 时提供广播级的图像质量和 CD 级的音质。MPEG-2 的另一特点是，可提供一个较广范围的可变压缩比，以适应不同的画面质量、存储容量以及带宽的要求。该标准已经在计算机、多媒体通信、HDTV 以及交互电视技术等专业领域得到广泛应用。目前广泛应用的 DVD 技术标准，其视频编码采用的是 MPEG-2 的视频编码技术，而其音频编码采用的是 AC-3 技术，此外 MPEG-2 还可用于为广播、有线电视网、电缆网络以及卫星直播提供广播级的数字视频。针对不同的应用，标准还规定了若干个语法子集。

（5）MPEG-4 标准

MPEG-4 国际标准的制定开始于 1998 年 11 月。MPEG-4 的本意是制定在甚低比特率下的视音频编码标准，但为了满足当今世界越来越多的视听材料要以数字形式进行相互交换而产生的种种需求，其目标已经彻底改变了。相对于 MPEG 的前两个标准，MPEG-4 已经不再是单纯的视频音频编解码标准，它更多定义的是一种格式和框架，而不是具体算法，从而为多媒体数据压缩提供了一个更广泛的平台。它揉合了各种现有的多媒体技术，包括压缩本身的一些工具、算法和图像分析与合成、计算机视觉、计算机图形、虚拟现实、语音合成等技术。其主要特征是基于对象的编码和基于模型的编码；还提供了一些基于对象的分级功能，以适应无线网和互联网等窄带网的传输。

经过这几年的发展，现在最热门的应用是利用 MPEG-4 的高压缩率和高的图像还原质量来把 DVD 里面的 MPEG-2 视频文件转换为体积更小的视频文件。经过这样处理，图像的视频质量下降不大但体积却可缩小，可很方便地用 CD-ROM 来保存 DVD 上面的节目。另外，MPEG-4 还可应用在家庭摄影录像、网络实时影像播放。

（6）MPEG-7 标准

MPEG-7 标准称为多媒体内容描述接口，2001 年 12 月制定成为标准草案。力图对各种不同类型的多媒体信息进行标准化的描述，以便实现快速有效的检索。MPEG-7 还定义了一个标准描述符集合用于描述各种类型的多媒体数据，与之相应的描述方案用于规范多媒体描述符的生成和不同描述符之间的有机联系。在此基础上，它还定义了一套标准的语言——描述定义语言（DDL，Description Definition Language）用于说明描述符和描述方案，保证其被采用的扩展性和较长的生命周期。

（7）MPEG-21 标准

随着多媒体应用技术的不断发展，有关多媒体的标准层出不穷，这些标准涉及到多媒体技术的方方面面。各种不同的多媒体信息分布式地存在于全球不同的设备上，要想通过异构网络有效的传输这些多媒体信息，必然需要综合地利用不同层次的多媒体技术标准。而现有的标准能否相互衔接，这需要一个综合性的标准来加以协调。于是在 1999 年 10 月的 MPEG 会议上提出了“多媒体框架”（Multimedia Framework）的概念。之后，这个工作

的方向被确定为 MPEG-21。它的主要目标是：讨论是否需要和如何将协议、标准、技术等不同的组件有机的结合起来，讨论是否需要新的规范，以及讨论在具备上述条件的前提下如何将不同的标准集成在一起。

表 1.2 近年来发布的或正在制定的视频图像编码标准

标准代号	标准名称	制定组织	形成日期
JBIG	Progressive Bi-level Image Compression	ISO/ITU-T	1991
JPEG	Digital Compression and Coding of continuous-tone Still Image	ISO/ITU-T	1992
JPEG2000	JPEG2000 Image Coding System	ISO/ITU-T	2000
H.261	Video Codec for Audiovisual Service at p*64Kbits/s	ITU-T	1990
H.263	Video Codec for Low Bit Rate Communication	ITU-T	1996
H.264	Advanced Video Coding	JVT	2003
MPEG-1	Coding of Moving Pictures and Associated Audio for Digital Storage Media up to 1.5Mbits/s	ISO/IEC	1992
MPEG-2	Generic Coding of Moving Pictures and Associated Audio Information	ISO/IEC	1994
MPEG-4	Coding of Audio-Visual Objects	ISO/IEC	1998
MPEG-7	Multimedia Content Description Interface	ISO/IEC	—
MPEG-21	Multimedia Framework	ISO/IEC	—

§ 1.2 视频压缩标准 H.264 的提出及应用领域

视频会议、数字存储媒体、电视广播、交互式网络流以及通信等应用领域对于运动图像的压缩编码方法的更高要求日益增大，为了充分提高编码效率和增强抗网络传输的鲁棒性，需要提出一个新的视频压缩的行业标准。ITU-T 成立视频编码专家组（VCEG，Video Coding Experts Group）开展新的视频编码标准，有三个主要的目标：语法定义简单清晰，避免过多的选择模式和子集；提高编码效率，在同样的保真度条件下，平均码率比现有的 H.263 视频压缩标准低 50%；改进 H.263 和 MPEG-4 在无线网和因特网应用中所暴露出的

不足。H.264 标准[14,15,16]即应这种要求而产生的，1998 年 1 月份开始草案征集，1999 年 9 月完成第一个草案，2001 年 5 月制定了其测试模式 TML-8，2001 年 12 月 VCEG 和 MPEG 组成联合视频小组（JVT， Joint Video Team）共同开展 H.264 标准的制定工作，2002 年 6 月的 JVT 第 5 次会议通过了 H.264 的 FCD 板，2003 年 3 月正式发布。它的用处在于可以使运动视频数据作为一种计算机可处理的数据形式，可以存储在各种存储媒体上，可以在现存或未来的网络上发送、接收，并且可以在现存或未来的广播信道上传播。

H.264 视频标准拟成为一个通用的应用于较广范围比特率、分辨率、质量和服务的标准。除了其他应用，H.264 标准至少还应涉及数字存储媒体，电视广播和实时通信等。在创建这一标准的过程中，考虑到了各个典型领域的应用要求，制定了必要的算法元素，并将其综合为一个单一的语法体系。因此，这一标准可以使得各种应用系统中的数据交换变得很容易。

H.264 和以前的标准一样，也是 DPCM 加变换编码的混合编码模式。但它采用"回归基本"的简洁设计，不用众多的选项，获得比 H.263++好得多的压缩性能；加强了对各种信道的适应能力，采用"网络友好"的结构和语法，有利于对误码和丢包的处理；应用目标范围较宽，以满足不同速率、不同解析度以及不同传输（存储）场合的需求；它的基本系统是开放的，使用无需版权。

在技术上，H.264 标准中有多个闪光之处，如统一的 VLC 符号编码，高精度、多模式的位移估计，基于 4×4 块的整数变换、分层的编码语法等。这些措施使得 H.264 算法具有很高的编码效率，在相同的重建图像质量下，能够比 H.263 节约 50%左右的码率。H.264 的码流结构网络适应性强，增加了差错恢复能力，有利于适应 IP 和无线网络的应用。

H.264 的视频标准涉及的应用领域（但并非仅限于此）如下所列：

CATV 光纤网、铜轴电缆等上的有线电视传播

DBS 直接广播卫星视频服务

DSL 数字订户在线视频服务

DTTB 数字地面电视广播

ISM 交互式存储媒体（光盘等）

MMM 多媒体邮件

MSPN 基于网络数据包的多媒体服务

RTC 实时对话服务（视频会议、电视电话等）

RVS 遥控监视

SSM 连续存储媒体（数字式的 VTR 等）

§1.3 本论文的主要工作及论文结构

本论文以由 ITU-T 和 MPEG 联合组成的 JVT（Joint Video Expert Team）公布的 H.264 视频压缩编码标准为基础，对其中若干关键技术展开研究。H.264 主要目标之一是在同样的保真度条件下，提高编码效率，而要提高压缩比，就会相应增加了算法的复杂度，从而对算法的实际应用很不利，因此本文主要研究 H.264 编解码器各模块的计算复杂度并对其中的关键模块整数变换模块进行处理改进；H.264 主要面对 IP 和无线环境的，抗误码性能是关键，本文还对基于 H.264 的视频传输抗误码技术进行了研究。

主要工作包括以下几方面：

（1）广泛研究了视频图像编码的一般方法，尤其是低速率视频编码方法，确定以 H.264 标准作为压缩编码的基本框架，并对该标准中若干关键模块与其他压缩标准进行分析比较。

（2）H.264 视频编解码演示系统的实现。本文以个人计算机为应用平台，利用纯软件进行视频的编解码。实现了一个从图像源处理、编码、传输、解码及显示全过程的实验演示系统。

（3）H.264 编码器中整数变换的研究。针对 DCT 变换存在浮点运算、运算量大等问题，H.264 中提出了整数变换。它能够很好的消除浮点运算，避免了逆变换不匹配，但仍存在计算量大的问题。因此本文在基于提升结构（lifting structure）的无乘法二进制 DCT（binDCT）的基础上，先分析了如何考虑 binDCT 各指标之间的折衷，并研究了如何取用不同的 binDCT 系数以满足不同指标的需求（比如，某些应用场合更关心小的计算量，而其他应用场合更关心好的编码质量，则可选择不同的 binDCT 系数来满足要求）。将浮点 DCT、整数变换和 binDCT 在结构和性能上进行比较分析，并通过对 binDCT 参数优化选择，实现编码质量和计算量的协调。最后提出“动态 binDCT 系数选择”的设想，针对实际中对 I 帧、P 帧，B 帧的质量要求不同，通过在传输比特流中附加少量的系数比特，使得对 I 帧采用高编码质量，对 P 帧采用中等编码质量，而对 B 帧可用低编码质量，以达到总体上更好的编码效果、更低的计算量。本章内容整理成论文《H.26L 中整数变换的研究及改进》[33]已在《通信技术》杂志 2003 年第 10 期上发表。

（4）基于 H.264 的视频传输抗误码技术的研究。先研究了传输误码的分类及目前主要的三种提高视频传输抗误码能力的处理方法。然后主要针对视频在网络传输中的错误或丢包等问题展开研究，并结合 H.264 中算术编码的特点，将一种基于算术编码的能连续检错和自动重传相结合的综合抗误码方案引入到 H.264 中，这大大的提高了 H.264 在视频传输时的抗误码能力并在增加冗余量和提高检错性能之间取得较好的折衷，并能获得较好的压缩效果和抗信道误码能力。本章内容已整理成论文《基于 H.26L 的视频传输抗误码技术

的研究》被《通信技术》杂志录用。

本论文共分五章，各章的内容安排如下：

绪论部分介绍了本文的研究背景、运动图像压缩编码技术的发展史及现状、视频压缩编码的技术、评价视频质量的性能指标及相关视频压缩编码的国际标准，论文的主要内容以及结构安排。

第二章介绍 H.264 国际标准的发展以及分析并实现了从视频源处理、压缩、传输、解码和显示的视频传输演示实验系统。

第三章对 H.264 中整数变换进行研究及改进。

第四章研究了基于 H.264 的视频传输抗误码技术，在 H.264 中引入了基于 AC 的 CED+ARQ 检错机制后能连续检错，提高了数据抗误码能力的同时能获得较好的压缩效果。

最后对全文进行了总结，指出了具有新意的若干想法，并讨论进一步的研究方向和关键技术难点。

第二章 H.264 视频压缩编码标准的分析和实现

§2.1 概论

自上个世纪 80 年代以来，ISO/IEC（国际标准化组织/国际电工委员会）制定的 MPEG-x 和 ITU-T（国际电信同盟）制定的 H.26x 两大系列视频编码国际标准的推出，开创了视频通信和存储应用的新纪元。从 H.261 视频编码建议，到 H.262/3、MPEG-1/2/4 等都有一个共同的不断追求的目标，即在尽可能低的码率（或存储容量）下获得更好的图像质量。而且，随着市场对图像传输需求的增加，如何适应不同信道传输特性的问题也日益显现出来。这就是 ISO/IEC 和 ITU-T 两大国际标准化组织联手制定的视频新标准 H.264 所要解决的问题。

H.261 是最早出现的视频编码建议[11]，目的是规范 ISDN 网上的会议电视和可视电话应用中的视频编码技术。它采用的算法结合了可减少时间冗余的帧间预测和可减少空间冗余的 DCT 变换的混合编码方法。和 ISDN 信道相匹配，其输出码率是 p×64kbps。p 取值较小时，只能传清晰度不太高的图像，适合于面对面的电视电话；p 取值较大时（如 p＞6），可以传输清晰度较好的会议电视图像。H.263 建议是低码率图像压缩标准，在技术上是 H.261 的改进和扩充，将运动矢量的搜索增加为半像素点搜索；同时又增加了无限制运动矢量、基于语法的算术编码、高级预测技术和 PB 帧编码等四个高级选项；从而达到了进一步降低码速率和提高编码质量的目的。它支持码率小于 64kbps 的应用，但实质上 H.263 以及后来的 H.263+和 H.263++已发展成支持全码率应用的建议，从它支持众多的图像格式这一点就可看出，如 Sub-QCIF、QCIF、CIF、4CIF 甚至 16CIF 等格式。

MPEG-1 标准制定于 1992 年，MPEG-1 标准视频编码部分的基本算法与 H.261/H.263 相似，也采用运动补偿的帧间预测、二维 DCT、VLC 游程编码等措施。此外还引入了帧内帧（I）、预测帧（P）、双向预测帧（B）和直流帧（D）等概念，进一步提高了编码效率。在 MPEG-1 的基础上，1994 年制定的 MPEG-2 标准[12]在提高图像分辨率、兼容数字电视等方面做了一些改进，例如它的运动矢量的精度为半像素；在编码运算中（如运动估计和 DCT）区分“帧”和“场”；引入了编码的可分级技术（Hierarchical coding），如空间可分级性、时间可分级性和信噪比可分级性等。1998 年 11 月公布的 MPEG-4 标准[12]引入了基于视听对象（AVO：Audio-Visual Object）的编码，大大提高了视频通信的交互能力和编码效率。MPEG-4 中还采用了一些新的技术，如形状编码、自适应 DCT、任意形状视频对象编码等。但是 MPEG-4 的基本视频编码器还是属于和 H.263 相似的一类混合编码器。

总之，H.261 建议是视频编码的经典之作，H.263 是其发展，并将逐步在实际上取而代之，主要应用于通信方面，但 H.263 众多的选项往往令使用者无所适从。MPEG 系列标准

从针对存储媒体的应用发展到适应传输媒体的应用，其核心视频编码的基本框架是和H.261一致的，其中引人注目的MPEG-4的“基于对象的编码”部分由于尚有技术障碍，目前还难以普遍应用。

因此，H.264是在此基础上发展起来的新的视频编码建议[14,15]，是ITU-T的VCEG（视频编码专家组）和ISO/IEC的MPEG（活动图像编码专家组）的联合视频组（JVT：joint video team）开发的一个新的数字视频编码标准，它既是ITU-T的H.264，又是ISO/IEC的MPEG-4的第10部分。1998年1月份开始草案征集，1999年1月制定了其测试模式TML-1，1999年9月，完成第一个草案，2001年5月制定了其测试模式TML-8，2002年6月的JVT第5次会议通过了H.264的FCD板。2003年3月正式发布。它克服了H.263和MPEG-4的弱点，在混合编码的框架下引入了新的编码方式，提高了编码效率，面向实际应用。同时，它是两大国际标准化组织的共同制定的，其应用前景应是不言而喻的。

§2.2 H.264的基本框架

H.264采用的是DPCM加变换编码的混合编码模式并采用“回归基本”的简洁设计，不用众多的选项，获得比H.263++好得多的压缩性能；加强了对各种信道的适应能力，采用基于“网络友好”的结构和语法，有利于为进一步进行误码及丢包处理提供良好的机制；应用目标范围较宽，以满足不同速率、不同解析度以及不同传输（存储）场合的需求；它的基本系统是开放的，使用无需版权。

H.264标准[16]与其他标准的不同之处在于它简洁的步骤以及各个功能模块细节上的优化，如基于4×4块的整数变换：H.264使用了三种变换方式，根据残余数据类型的不同来进行选择。高精度、多模式的位移估计：H.264采用了四分之一像素精度的分数像素运动补偿能获得比整数像素运动补偿更好的压缩性能，代价是大大增加了算法复杂度。熵编码部分提供了通用变长编码（UVLC）和基于上下文的二进制自适应算术编码（CABAC）两种方法等。这些措施使得H.264算法具有很高的编码效率，在相同的重建图像质量下，能够比H.263节约50%左右的码率，当然这些算法也带来了复杂度的提高。H.264的码流结构网络适应性有所提高，主要通过以下几种策略：基于率失真或SAD（Sum of Absolute Difference）判断的宏块编码模式的选择、数据分类、打包机制和错误掩盖等。但是从宏块编码模式出发，帧内编码宏块的抗误码性能需要较大的增强。

2.2.1 H.264的主要指标

1、 传输速率

传输速率是可变的，主要由终端和网络决定，一般小于64kbps。

2、 视频源格式

视频被描述为一组连续的画面，而每幅画面看作是二维的像素阵列，每一像素的彩色

表示包含三个分量：红（R）、绿（G）、蓝（B），这称为图像的 RGB 空间表示。为了利用人的视觉特性以降低数据量，通常把 RGB 空间表示的彩色图像变换到其他彩色空间。

大多数压缩算法是在 YUV 空间中进行压缩处理。在 YUV 空间中，每一彩色像素用另三个分量表示：亮度分量（Y）以及两个色差分量（U、V）。我们可以利用两个空间的联系进行空间转换，如下是两个空间相互转化的关系[14]：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1.14 & 0 \\ 1 & -0.58 & -0.394 \\ 1 & 0 & 2.03 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

YUV 表示法的重要特点是它的亮度信号（Y）和色差信号（U、V）是相互独立的，也就是 Y 信号分量构成的黑白灰度图与用 U、V 信号构成的另外两幅单色图是相互独立的。由于 Y、U、V 是独立的，所以可以对这些单色图分别进行编码。

YUV 表示法的另一个优点是可以利用人类视觉系统的特性来降低数字彩色图像所需要的存储容量。人眼对彩色细节的分辨能力远比对亮度细节的分辨能力低。在 RGB 空间中，R、G、B 三个信号有一个发生了变化，则总的图像的颜色就会发生变化，人眼是很容易察觉这种变化的；然而人眼对 Y、U、V 三个信号的变化是有不同反应的，其中对亮度信号的变化比较敏感，而对色差信号的变化不是很敏感，这样就可以更多的考虑亮度信号，而对色差信号采用一些处理方法以提高压缩比；例如即使经过亚采样或直接丢弃一部分数据等处理，但人眼对恢复时转换到 RGB 空间后的图像的变化仍然是不易察觉的。所以不管在 JPEG 标准，H.26x 标准还是 MPEG 系列标准都要将视频源从 RGB 空间变换到 YUV 空间中去，对亮度分量和色差分量采用不同的处理办法，以进行更进一步的压缩处理。在我们的编码方案里也进行了这样的考虑。

H.264 的视频源格式为 QCIF，彩色亚取样 4：2：0，帧频 29.97 帧每秒。但也支持更大的格式，如：CIF 或用户自定义的更大格式，但需要降低帧频以达到较低的比特率[16]。表 2.1 给出了各种标准视频编码图像格式。

表 2.1 数字图像/视频格式

—		Sub QCIF	QCIF	CIF	4CIF	16CIF	ITU-R 601	ITU-R 709
每秒帧数		5—15	5—15	0—30	10—30	25—30	25/30	25/30
每行像素数	Y	128	176	352	704	1408	720	1920
	U（V）	64	88	176	352	704	360	960
行数	Y	96	144	288	576	1152	576/480	1080
	U（V）	48	72	144	288	576	576/480	1080

2.2.2 H.264 的编码方法[14,16]

2.2.2.1 预测

输入图像帧的每个宏块都要通过帧内帧间选择器来判定该宏块是采用帧内预测模式还是帧间预测模式。

1）帧内预测

在先前的 H.26x 系列和 MPEG-x 系列标准中，都是采用的帧间预测的方式。在 H.264 中，当编码 Intra 图像时可用帧内预测。如果一个块以帧内预测模式编码时，一个预测块是由以前编码的块和重构的块来获得，在编码时从当前块中减去预测块。对于亮度采样，预测块 P 可以是 4×4 的子块或者是 16×16 的宏块。对于 4×4 的亮度块共有 9 种可选预测模式，对于 16×16 的亮度块共有 4 种可选预测模式，一般对于 4×4 的色度块只用一种预测模式。

（1）4×4 的亮度预测模式

并不是所有的块都在同一片（slice）内，为了保持同一片内解码的独立性，只有同一个片内的像素才可以用来预测。对于每个 4×4 块（除了边缘块特别处置以外），每个像素都可用 17 个最接近的先前已编码的像素的不同加权和（有的权值可为 0）来预测，即此像素所在块的左上角的 17 个像素（图 2.1）。显然，这种帧内预测不是在时间上，而是在空间域上进行的预测编码算法，可以除去相邻块之间的空间冗余度，取得更为有效的压缩。编码器为每块选择预测模式的准则是使预测块 P 与原始块之间的残差最小（Sum of Absolute Errors（SAE）绝对误差和）。它共有九种预测方式（如图 2.2 所示，模式 2 未标示）。

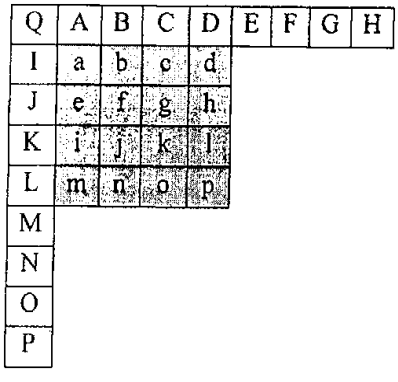

图 2.1 4×4 块及相邻像素

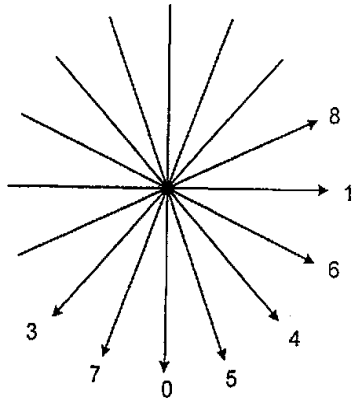

图 2.2 帧内预测模式方向

模式 0（垂直预测）：如果 A，B，C，D 与该块在同一片内，则 a，e，i，m 由 A 预测得到；b，f，j，n 由 B 预测得到；c，g，k，o 由 C 预测得到；d，h，l，p 由 D 预测得到。

模式 1（水平预测）：如果 I，J，K，L 与该块在同一片内，则 a，b，c，d 由 I 预测得

到；e，f，g，h 由 J 预测得到；i，j，k，l 由 K 预测得到；m，n，o，p 由 L 预测得到。

模式 2（直流预测）：如果 A，B，C，D，I，J，K，L 都在该块所在的片内，则该 4×4 亮度块的所有像素的预测值均为（A+B+C+D+I+J+K+L+4）>>3。如果 A，B，C，D 不在该片内而 I，J，K，L 在该片内，则该 4×4 亮度块的所有像素的预测值均为（I+J+K+L+2）>>2。如果 A，B，C，D 在该片内而 I，J，K，L 不在该片内，则该 4×4 亮度块的所有像素的预测值均为（A+B+C+D+2）>>2。如果 A，B，C，D，I，J，K，L 都不在该块所在的片内，则该 4×4 亮度块的所有像素的预测值都设为 128。大部分的块就是以这种方式来预测的。

模式 3（左下对角线预测）：该模式仅用于当 A，B，C，D，I，J，K，L，Q 都在该块所在的片内，a 由（A+2B+C+I+2J+K+4）>>3 预测得到；b，e 由（B+2C+D+J+2K+L+4）>>3；c，f，i 由（C+2D+E+K+2L+M+4）>>3 预测得到；d，g，j，m 由（D+2E+F+L+2M+N+4）>>3 预测得到；h，k，n 由（E+2F+G+M+2N+O+4）>>3 预测得到；l，o 由（F+2G+H+N+2O+P+4）>>3 预测得到；p 由（G+H+O+P+2）>>3 预测得到。

模式 4（右下对角线预测）：该模式仅用于当 A，B，C，D，I，J，K，L，Q 都在该块所在的片内，m 由（J+2K+L+2）>>2 预测得到；i，n 由（I+2J+K+2）>>2 预测得到；e，j，o 由（Q+2I+J+2）>>2 预测得到；a，f，k，p 由（A+2Q+I+2）>>2 预测得到；b，g，l 由（Q+2A+B+2）>>2 预测得到；c，h 由（A+2B+C+2）>>2 预测得到；d 由（B+2C+D+2）>>2 预测得到。

模式 5（垂直偏右预测）：该模式仅用于当 A，B，C，D，I，J，K，L，Q 都在该块所在的片内，a 由（2A+2B+J+2K+L+4）>>3 预测得到；b，i 由（B+C+1）>>1 预测得到；c，j 由（C+D+1）>>1 预测得到；d，k 由（D+E+1）>>1 预测得到；l 由（E+F+1）>>1 预测得到；e 由（A+2B+C+K+2L+M+4）>>3 预测得到；f，m 由（B+2C+D+2）>>2 预测得到；g，n 由（C+2D+E+2）>>2 预测得到；h，o 由（D+2E+F+2）>>2 预测得到；p 由（E+2F+G+2）>>2 预测得到。

模式 6（水平偏下预测）：该模式仅用于当 A，B，C，D，I，J，K，L，Q 都在该块所在的片内，a，g 由（Q+I+1）>>1 预测得到；b，h 由（I+2Q+A+2）>>2 预测得到；c 由（Q+2A+B+2）>>2 预测得到；d 由（A+2B+C+2）>>2 预测得到；e，k 由（I+J+1）>>1 预测得到；f，l 由（Q+2I+J+2）>>2 预测得到；i，o 由（J+K+1）>>1 预测得到；j，p 由（I+2J+K+2）>>2 预测得到；m 由（K+L+1）>>1 预测得到；n 由（J+2K+L+2）>>2 预测得到。

模式 7（垂直偏左预测）：该模式仅用于当 A，B，C，D，I，J，K，L，Q 都在该块所在的片内，a，j 由（Q+A+1）>>1 预测得到；b，k 由（A+B+1）>>1 预测得到；c，l 由（B+C+1）>>1 预测得到；d 由（C+D+1）>>1 预测得到；e，n 由（I+2Q+A+2）>>2 预测得到；f，o 由（Q+2A+B+2）>>2 预测得到；g，p 由（A+2B+C+2）>>2 预

测得到；h 由（B+2C+D+2）>>2 预测得到；i 由（Q+2I+J+2）>>2 预测得到；m 由（I+2J+K+2）>>2 预测得到。

模式 8（水平偏上预测）：该模式仅用于当 A，B，C，D，I，J，K，L，Q 都在该块所在的片内，a 由（B+2C+D+2I+2J+4）>>3 预测得到；b 由（C+2D+E+I+2J+K+4）>>3 预测得到；c，e 由（D+2E+F+2J+2K+4）>>3 预测得到；d，f 由（E+2F+G+J+2K+L+4）>>3 预测得到；g，i 由（F+2G+H+2K+2L+4）>>3 预测得到；h，j 由（G+3H+K+3L+4）>>3 预测得到；l，n 由（L+2M+N+2）>>3 预测得到；k，m 由（G+H+L+M+2）>>2 预测得到；o 由（M+N+1）>>1 预测得到；p 由（M+2N+O+2）>>2 预测得到。

（2）16×16 亮度预测模式

模式 0（垂直预测），模式 1（水平预测），模式 2（直流预测），模式 3（平面预测）：一个线性平面函数，这在亮度变化平缓的区域用的比较好（图 2.3）。

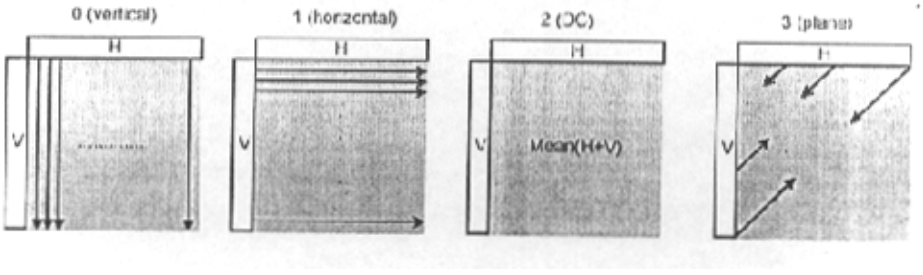

图 2.3 16×16 亮度预测模式

（3）色度预测模式

与 16×16 亮度预测模式类似，同样分为模式 0（垂直预测），模式 1（水平预测），模式 2（直流预测），模式 3（平面预测）这四种预测方式。注意：如果一个 8×8 亮度块按帧内模式编码时，色度块也同样按帧内模式编码。

（4）帧内预测模式的编码

如果将每个 4×4 块的帧内预测模式都告知解码端，这需要大量的比特。但是帧内模式时，邻近的 4×4 块必然具有高度的相关性。例如，图 2.4 中 A，B 用模式 2 编码则 C 一般也用模式 2 编码。对每一个当前块 C，编码器和解码器都要计算最可能模式（most_probable_mode）。如果 A，B 都按 4×4 帧内模式编码且在同一片内，则最可能模式是 A，B 预测模式的最小值；否则最可能模式被设置为 2（直流预测）。

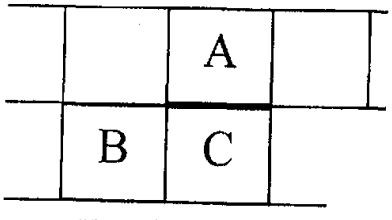

图 2.4 宏块位置图

编码器为每一个 4×4 的块设置一个标志位，使用最可能模式（use_most_probable_mode）。若该标志为“1”，则参数最可能模式被使用；若该标志为“0”，

则另一个参数剩余模式选择，（remaining_mode_selector）被设置。如果剩余模式选择小于当前的最可能模式则预测模式为剩余模式选择，否则预测模式为剩余模式选择加 1。因此剩余模式选择仅需要 8 个值（0—7）来表示当前帧内模式（0—8）。

例：A 用模式 3 预测，B 用模式 1 预测，则 C 的最可能模式为 1。若标志使用最可能模式被设为“0”，则剩余模式选择被设置。根据剩余模式选择的值，8 个剩余模式中的一个被选出来。

宏块按帧内 16×16 方式编码或一个色度块按帧内块编码必须在宏块的顶端标志。

表 2.2 选择预测模式（最可能模式=1）

剩余模式选择	块 C 的预测模式
0	0
1	2
2	3
3	4
4	5
5	6
6	7
7	8

2）帧间预测

帧间预测是从已编码的前几帧得出一个预测模型，这个模型是由参考帧中像素移位得到（运动补偿预测）。AVC 编码器使用基于块的运动补偿预测，这个准则自 H.261 以来一直被使用。与以前所不同的是，H.264 采用了不同的块大小（16×16 到 4×4）和精细的分数像素运动向量（亮度分量的 1/4 像素）。

（1）树结构运动补偿

AVC 支持的运动补偿块大小从 16×16 到 4×4 亮度块之间的多种选择。一个 16×16 亮度块可以按以下四种方式分解（图 2.5）：16×16，16×8，8×16，8×8。每一个子分割区域称作宏块分割。如果 8×8 模式被选中，这个宏块中的 4 个 8×8 又可以进一步按以下四种方式分解（图 2.6）：8×8，8×4，4×8，4×4（称作宏块子分割）。这些分割和子分割时每个宏块有多种组合。这种将宏块分割成多种块大小的运动补偿子块的方法称作树结构运动补偿。这种多模式的灵活和细致的划分，更切合图像中实际运动物体的形状，大大提高了运动估计的精确程度。在指定的范围内找出每种块的最小损耗，这些损耗包括信号的 SAD（绝对差之和）以及编码块大小信息和运动向量所需的平均比特数，然后由最小损耗得出最优的块。

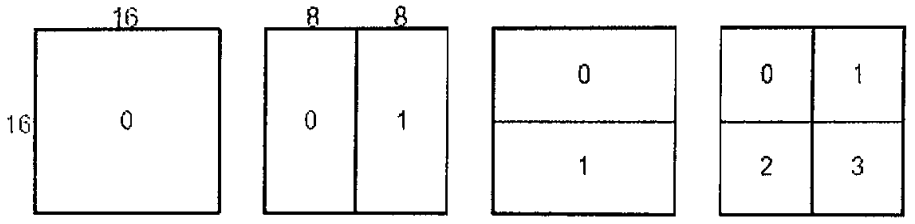

图 2.5 宏块分割：16×16，16×8，8×16，8×8

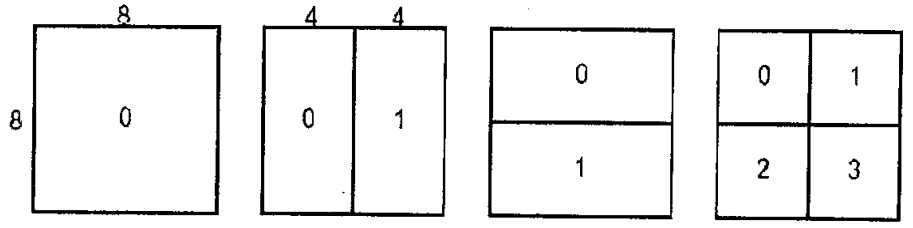

图 2.6 宏块子分割：8×8，8×4，4×8，4×4

在这种方式下，在每个宏块中可以包含有 1、2、4、8 或 16 个运动矢量。对于每个分割或子分割都需要一个独立的运动向量，每个运动向量必须被编码和传输，同时分割的选择也要编码并在压缩比特流中传输。选择一个大的分割尺寸（例如 16×16，16×8，8×16）意味着需要少量的比特来表示运动向量的选择和分割的类型，但是运动补偿后的残差可能包含大量的信息。选择一个小的分割尺寸（例如 8×4，4×8 等），运动补偿后得到一个低能量残差但需要大量的比特来表示运动向量的选择和分割的类型。因此分割尺寸的选择对于压缩性能具有重要的影响。总的来说，大的分割尺寸适用于帧内相似的区域，小的分割尺寸有利于细节区域。

对于宏块中每个色度分量（U，V）采用亮度分量一半的采样精度。每个色度块采用与亮度块同样的分割方式，只是水平和垂直的大小各一半。每个运动向量的水平和垂直大小为亮度块的一半。我们选择分割的大小是使编码残差和运动向量最小。帧内大面积部分使用大区域，细节部分使用小区域。

（2）分数采样精度

分数像素运动补偿能获得比整数像素运动补偿更好的压缩性能，代价是增加了算法复杂度。研究表明运动预测采用 1/4 像素精度的效果要明显优于采用半像素精度预测的效果，因此 H.264 采用的运动预测精度为四分之一像素。

图 2.7 中灰色块位置为整像素位置，两个整像素位置之间的是半像素位置（如图 2.7 中的 b，h，m，s），半像素位置点像素值可由相邻整像素位置像素值通过一个 6 抽头（1/32，-5/32，5/8，5/8，-5/32，1/32）有限脉冲响应滤波器得到。例如，b 值可由水平整数采样 E，

F，G，H，I 和 J 通过 2.1 式得到。

$$b=\text{round}((E-5F+20G+20H-5I+J)/32) \tag{2.1}$$

同理 h 可由 A，C，G，M，R 和 T 通过内插滤波得到。一旦与整像素位置相邻的所有半像素值都得出，则 j 值可由 cc，dd，h，m，ee 和 ff 通过内插滤波得到（也可通过 aa，bb，b，s，gg，hh 得到）。虽然 6 抽头滤波器比双线性滤波器（H.263）复杂，但它更加精确且能获得更好的运动补偿性能。

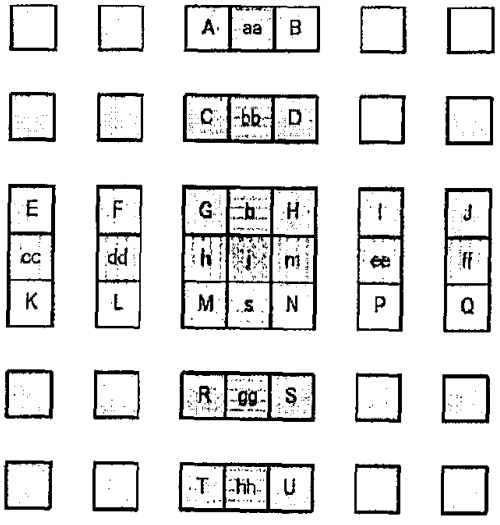

图 2.7 亮度信号半像素位置图

一旦得出所有的半像素值，则四分之一像素位置的像素值（如图 2.8 所示，a，c，i，k，d，f，n，q，e，g，p，r 位置）可由半像素值线性内插得到。

$$a=\text{round}((G+b)/2) \tag{2.2}$$

$$e=\text{round}((b+h)/2) \tag{2.3}$$

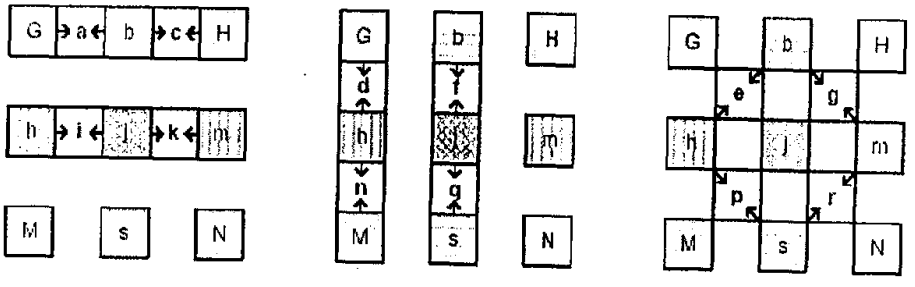

图 2.8 亮度信号四分之一像素位置图

由于彩色亚取样是 4：2：0 的采样，则亮度信号预测中亮度信号四分之一像素运动向量就要求色度信号八分之一像素运动向量。如图 2.9 中色度信号八分之一像素位置像素值

a 可由 A，B，C，D 的线性组合得到：

$$a = round([(8-d_x)\cdot(8-d_y)A + d_x\cdot(8-d_y)B + (8-d_x)\cdot d_y C + d_x\cdot d_y D]/64) \quad (2.4)$$

在图 2.9 中，d_x=2，d_y=3 因此 a=round（[30A＋10B＋18C＋6C]/64）。

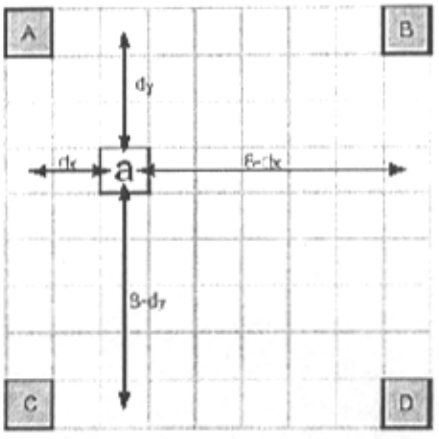

图 2.9 色度信号八分之一像素位置图

2.2.2.2 运动补偿

H.264 中的运动补偿比其他标准中的运动补偿更加灵活有效，预测的时候采用了多参考帧的方法（此前只在 2001 的 H.263 的最新版本中出现过），运动补偿时支持 7 种大小的块（H.263 只支持 2 种）。运动向量也比早期的标准具有更高的精度，在低复杂度模式时采用四分之一像素精度，而高复杂度模式是采用八分之一像素精度。运动补偿循环时采用了块滤波器，减少了视频假象并提高了预测效果。

对每个分割的运动向量进行编码需要大量的比特，特别是选用小分割尺寸的时候。临近分割的运动向量具有很高的相关性，并且可由已编码的邻近分割的运动向量来预测。预测向量（MVp）由先前运动向量计算推出，被用来编码传输的是 MVD（当前运动向量和预测向量差）。运动预测向量由运动补偿分割尺寸和邻近运动预测向量共同得到，可概括如下。

设 E 是当前宏块（分割或子分割）；A，B，C 是其邻近的分割。如果 A，B，C，E 分割的大小不同则 A 取最上面一个，B 取最左边一个。图 2.10 所示是所有宏块为 16×16，图 2.11 所示是临近分割与 E 尺寸不同时的情况。

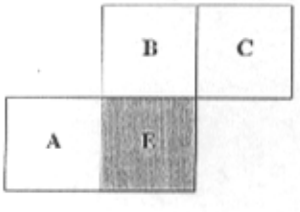

图 2.10 当前和邻近分割（同样分割大小）

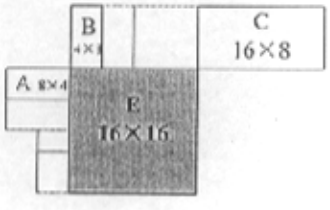

图 2.11 当前和邻近分割（不同分割大小）

（1）传输分割（不包括 16×8，8×16）：MVp 是 A，B，C 运动向量的中值。

（2）对于 16×8：上面的 16×8 的 MVp 由 B 推出，下面的由 A 推出。

（3）对于 8×16：左边的 8×16 的 MVp 由 A 推出，右边的由 C 推出。

（4）对于跳过的宏块：MVp 由（1）得出。

如果上述传输的块中有些块不在该片中，则 MVp 需要重新确定。在解码端，预测向量按同样方式获得，再加到解码出的 MVD 中。对于跳过的宏块，不需要解码向量，运动补偿宏块由 MVp 得到。

2.2.2.3 4×4 块的整数变换

H.264 与先前的标准相似，对残差采用基于块的变换编码，但它的改进之处在于变换是整数操作而不是实数运算，其过程和 DCT 基本相似。这种方法的优点在于：在编码器和解码器中允许精度相同的变换和反变换，便于使用简单的定点运算方式，并且避免了逆变换不匹配问题，也就是说，这里没有"逆变换误差"。变换的单位是 4×4 块，而不是以往常用的 8×8 块。由于用于变换块的尺寸缩小，运动物体的划分更精确，这样，不但变换计算量比较小，而且在运动物体边缘处的衔接误差也大为减小，减少了块失真。为了使小尺寸块的变换方式对图像中较大面积的平滑区域不产生块之间的灰度差异，可对帧内宏块亮度数据的 16 个 4×4 块的 DC 系数（每个小块一个，共 16 个）进行第二次 4×4 块的变换，对色度数据的 4 个 4×4 块的 DC 系数（每个小块一个，共 4 个）进行 2×2 块的变换。

与 H.263+中 8×8 点 DCT 相比，这种 4×4 点整数变换减少了方块效应，且运算速度快、结果精度高。

2.2.2.4 量化和编码

H.264 为了提高码率控制的能力，量化步长变化的幅度控制在 12.5%左右，而不是以不变的增幅变化。变换系数幅度的归一化被放在反量化过程中处理以减少计算的复杂性。为了强调彩色的逼真性，对色度系数采用了较小量化步长。

H.264 对 4×4 块使用两种扫描方式，除了与 H.263＋类似的简单 Zig-Zag（图 2.12 a）扫描外，还使用双 Zig-Zag（图 2.12 b）扫描来提高编码效率。H.264 在量化过程中使用尺

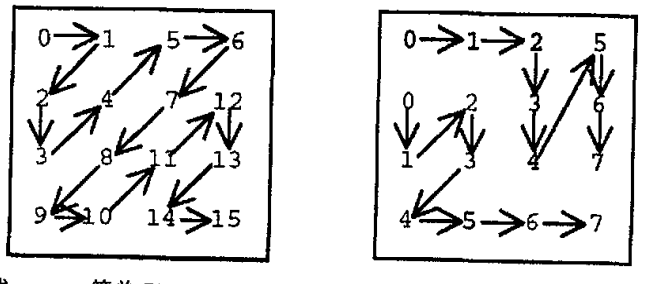

图 2.12 扫描方式　　a 简单 Zig-Zag　　b 双 Zig-Zag

度变换归一化变换编码后的系数，保证在 32 位运算系统中，计算结果有最大精度且不会

溢出。

在编码量化系数时 H.264 采用两种不同的熵编码技术：基于上下文的自适应二进制算术编码（CABAC：Context-Adaptive Binary Arithmetic Coding）和通用变长编码（UVLC：Universal VLC）。UVLC 使用一个长度无限的码字集，设计结构非常有规则，用相同的码表可以对不同的对象进行编码。这种方法很容易产生一个码字，而解码器也很容易地识别码字的前缀，UVLC 在发生比特错误时能快速获得重同步。UVLC 提供了一个简单且健壮的方法来编码模式信息和 DCT 量化系数，但在中高比特率时性能不够好。因此 CABAC 成了 H.264 的另一个选择，CABAC 具有三个独特的优势：（1）内容模型提供了编码符号的条件概率的估计；（2）算术编码允许对于每个符号分配非整数数目的比特；（3）自适应算术编码允许熵编码器能适应无固定符号统计的模式。CABAC 编码性能比 UVLC 好，但计算复杂度也高。

2.2.3 面向 IP 和无线环境

H.264 标准中包含了用于差错消除的工具，便于压缩视频在误码、丢包多发环境中传输，保证了在移动信道或 IP 信道中传输的健壮性。

为了抵御传输差错，H.264 视频流中的时间同步可以通过采用帧内图像刷新来完成，空间同步由片结构编码（slice structured coding）来支持。同时为了便于误码以后的再同步，在一幅图像的视频数据中还提供了一定的重同步点。另外，帧内宏块刷新和多参考宏块允许编码器在决定宏块模式的时候不仅可以考虑编码效率，还可以考虑传输信道的特性。

除了利用量化步长的改变来适应信道码率外，在 H.264 中，还常利用数据分割的方法来应对信道码率的变化。从总体上说，数据分割的概念就是在编码器中生成具有不同优先级的视频数据以支持网络中的服务质量 QoS。例如，采用基于语法的数据分割（syntax-based data partitioning）方法，将每帧数据按其重要性分为几部分，这样允许在缓冲区溢出时丢弃不太重要的信息。还可以采用类似的时间数据分割（temporal data partitioning）方法，通过在 P 帧和 B 帧中使用多个参考帧来完成。

在无线通信的应用中，我们可以通过改变每一帧的量化精度或空间/时间分辨率来支持无线信道的大比特率变化。可是，在实际的情况下，要求编码器对变化的各种比特率进行响应是不可能的。因此，不同于 MPEG-4 中采用的精细分级编码 FGS（Fine Granular Scalability）的方法（效率比较低），H.264 采用流切换的 SP 帧来代替分级编码。

另外，H.264 还提出了基于预测环路的去方块效应滤波、SAD（绝对误差和）的 Hadamard 变换处理以及为了满足视频内容存储的临时媒体文件格式等，它们都可以用于提高编码效率、恢复图像质量和多种视频应用的需求。

溢出。

在编码量化系数时 H.264 采用两种不同的熵编码技术：基于上下文的自适应二进制算术编码（CABAC：Context-Adaptive Binary Arithmetic Coding）和通用变长编码（UVLC：Universal VLC）。UVLC 使用一个长度无限的码字集，设计结构非常有规则，用相同的码表可以对不同的对象进行编码。这种方法很容易产生一个码字，而解码器也很容易地识别码字的前缀，UVLC 在发生比特错误时能快速获得重同步。UVLC 提供了一个简单且健壮的方法来编码模式信息和 DCT 量化系数，但在中高比特率时性能不够好。因此 CABAC 成了 H.264 的另一个选择，CABAC 具有三个独特的优势：（1）内容模型提供了编码符号的条件概率的估计；（2）算术编码允许对于每个符号分配非整数数目的比特；（3）自适应算术编码允许熵编码器能适应无固定符号统计的模式。CABAC 编码性能比 UVLC 好，但计算复杂度也高。

2.2.3 面向 IP 和无线环境

H.264 标准中包含了用于差错消除的工具，便于压缩视频在误码、丢包多发环境中传输，保证了在移动信道或 IP 信道中传输的健壮性。

为了抵御传输差错，H.264 视频流中的时间同步可以通过采用帧内图像刷新来完成，空间同步由片结构编码（slice structured coding）来支持。同时为了便于误码以后的再同步，在一幅图像的视频数据中还提供了一定的重同步点。另外，帧内宏块刷新和多参考宏块允许编码器在决定宏块模式的时候不仅可以考虑编码效率，还可以考虑传输信道的特性。

除了利用量化步长的改变来适应信道码率外，在 H.264 中，还常利用数据分割的方法来应对信道码率的变化。从总体上说，数据分割的概念就是在编码器中生成具有不同优先级的视频数据以支持网络中的服务质量 QoS。例如，采用基于语法的数据分割（syntax-based data partitioning）方法，将每帧数据按其重要性分为几部分，这样允许在缓冲区溢出时丢弃不太重要的信息。还可以采用类似的时间数据分割（temporal data partitioning）方法，通过在 P 帧和 B 帧中使用多个参考帧来完成。

在无线通信的应用中，我们可以通过改变每一帧的量化精度或空间/时间分辨率来支持无线信道的大比特率变化。可是，在实际的情况下，要求编码器对变化的各种比特率进行响应是不可能的。因此，不同于 MPEG-4 中采用的精细分级编码 FGS（Fine Granular Scalability）的方法（效率比较低），H.264 采用流切换的 SP 帧来代替分级编码。

另外，H.264 还提出了基于预测环路的去方块效应滤波、SAD（绝对误差和）的 Hadamard 变换处理以及为了满足视频内容存储的临时媒体文件格式等，它们都可以用于提高编码效率、恢复图像质量和多种视频应用的需求。

§2.3 H.264 的实现

H.264 的算法在概念上可以分为两层：视频编码层（VCL：Video Coding Layer）负责高效的视频内容表示，对视频内容进行有效的描述；网络适配层（NAL：Network Abstraction Layer）负责以网络所要求的恰当的方式对数据进行打包和传送，完成在不同网络上视频数据的打包传输。在 VCL 和 NAL 之间定义了一个基于分组方式的接口，打包和相应的信令属于 NAL 的一部分。这样，高编码效率和网络友好性的任务分别由 VCL 和 NAL 来完成。H.264 将视频编码层与网络自适应层进行有效地分离，而本文的工作重点在视频编码层。

2.3.1 H.264 的视频编码器[17]

H.264 的视频编码器的结构框图如图 2.13 所示，主要由编码器前向支路和编码器重构支路组成。

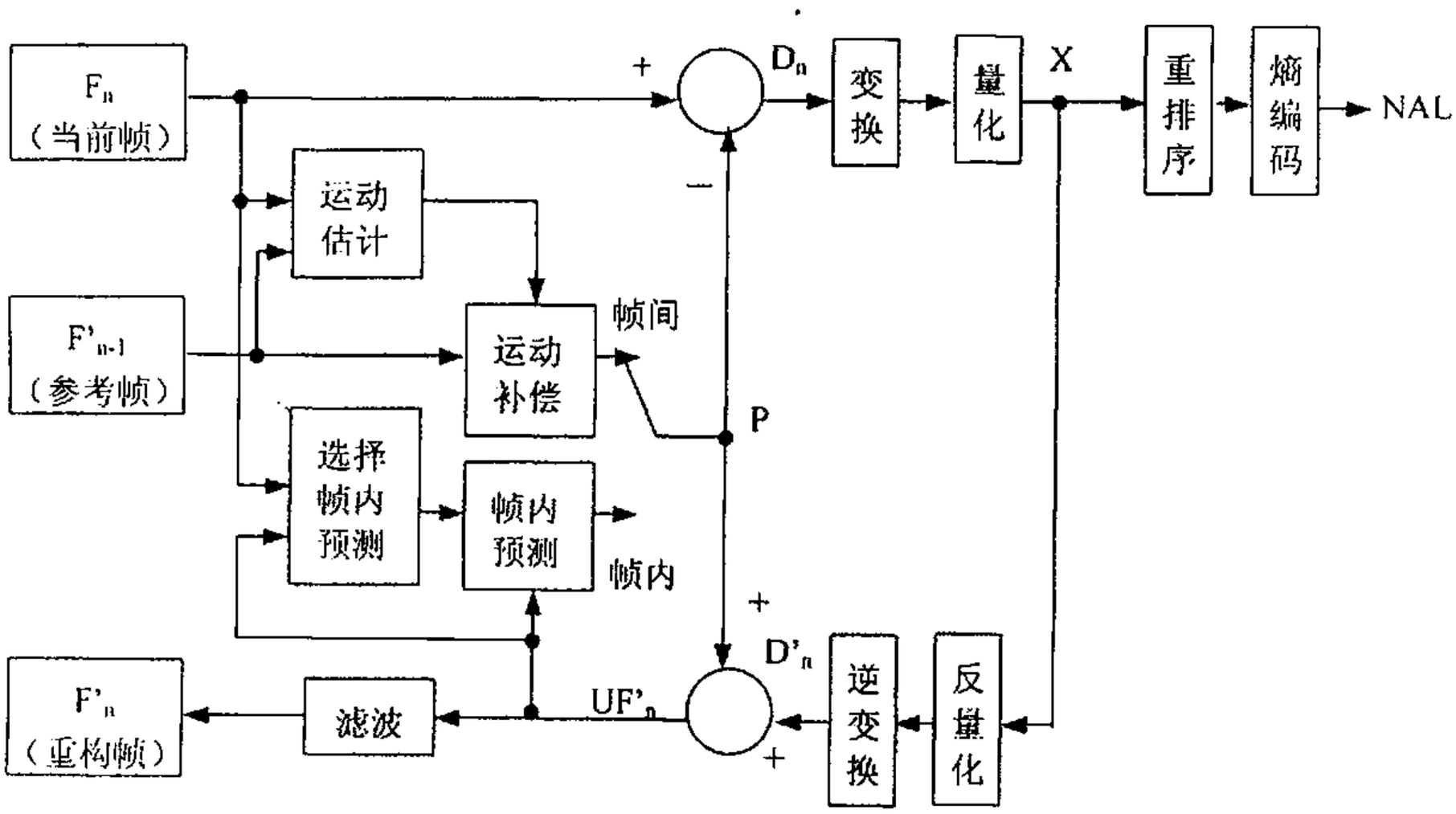

图 2.13 H.264 编码器结构框图

1.编码器前向支路

输入的视频序列包括大量的图像序列，而每一幅图像又被分成 16× 16 像素的宏块。编码时这些像素被组合成宏块片（slice），片被作为独立的编码单元，他们在解码的时候不需要参考帧内其它的片。通过一个有效的时序安排，编码器可进行高效的平行处理。

简要的编码过程：输入当前需编码帧 F_n，该帧是由一组宏块（16× 16 的原始图像）表示，每个宏块可用帧内或帧间模式来编码。基于重构帧得到预测帧 P：在帧内模式时，P 由帧 n 中已经编码部分再解码重构所得（图中的 UF'_n，注意到 P 是由未滤波的值组成）；

在帧间模式时，P 由几个参考帧用运动补偿预测的方式得到。参考帧可以是已编码重构的前几帧或后几帧（时间顺序）。

当前宏块减去预测块得到残差块 D_n，然后将其进行变换量化得到一组变换系数 x。这些系数被重排列和熵编码。熵编码系数和其他一些信息（宏块预测模式，量化步长，运动补偿，运动向量等）组成了压缩比特流。它将被送到网络提取层用来传输或存储。

2.编码器重构支路

量化后的宏块系数 x 被解码用来重构帧，以便进一步编码下面的宏块。系数 x 经过反量化，反变换得到残差宏块 D'_n，由于量化过程带来了损失，D'_n 与 D_n 并不相同。

预测宏块 P 加到 D'_n 上得到重构宏块 UF'_n（原始宏块的失真块）。使用滤波器可以减少块失真的影响并由一系列宏块 F'_n 来产生重构参考帧。

2.3.2 H.264 的视频解码器

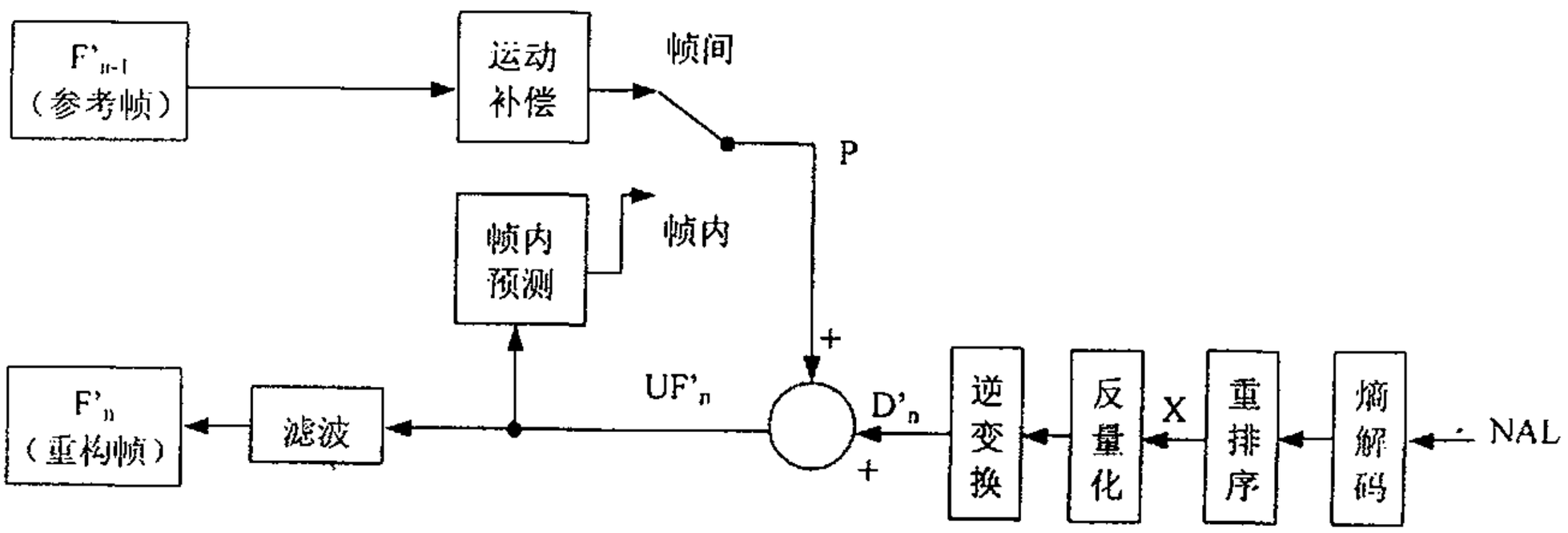

图 2.14 H.264 解码器结构框图

H.264 解码器从 NAL 中获得一个压缩比特流。数据元素经过熵解码和重排序来获得一组量化系数 X。在经过反量化和逆变换来获得 D'_n（这与编码器中的 D'_n 相同）。根据从比特流中解码获得的头信息，解码器产生一个预测宏块 P，与编码器中的 P 相同。P 和 D'_n 相加得到 UF'_n 然后再解码得到宏块 F'_n

可见，编码器的重构路径的作用是保证编码器和解码器使用同样的参考帧来产生预测 P。如果不是这样，预测 P 在编码器和解码器中就会不同，这将在编解码器之间引入误差和漂移。

H.264 标准与 H.263v2 和 MPEG-4 相比节约了高达 50%的比特率，同时提高了视频质量。对每个考虑实际应用的工程师而言，在关注 H.264 的优越性能的同时必然会衡量其实现难度。从总体上说，H.264 性能的改进是以增加复杂性为代价而获得的。

本文统计了编码过程中各主要模块所占计算时间，以 JM61e 软件为基础，采用的测试序列为 foreman_qcif.yuv，图像大小为 176×144，帧率为 10fps。实验中视频序列为 1000

帧，序列的编码帧格式是 IBBPBBPBB。采用的熵编码模式是基于上下文的自适应算术编码 CABAC。运行环境为 Celeron(R) CPU 2.2GHz，内存 256M。整个编码时间为 1331.745s，运动估计耗时 572.641s 约占了整个编码时间的 43%，H.264 采用了分数像素运动补偿能获得比整数像素运动补偿更好的压缩性能，代价是大大增加了算法复杂度。运动预测部分耗时 93.226s 约占了整个编码时间的 7%。H.264 使用了三种变换方式，根据残余数据类型的不同来进行选择。亮度 DC 系数采用 4×4 的矩阵，色度 DC 系数采用 2×2 的矩阵，对于其他的都采用 4×4 的块来变换，整数变换和量化耗时 226.496s 约占了整个编码时间的 17%，这也是编码模块中用时较多的模块，本文在接下来的工作中将对整数变换和量化模块进行讨论分析，提出改进减少其复杂度。基于上下文的自适应算术编码 CABAC 耗时 133.658s 约占了整个编码时间的 10%。

但是，随着技术的发展，这种复杂性的增加是在我们当前或不久的将来的技术可接受的范围之内的。实际上，考虑到复杂性的限制，H.264 对一些计算量特别大的改进算法未予采用，如 H.264 未采用全局运动补偿技术，这在 MPEG-4 的 ASP 中是采用的，并增加了相当的编码复杂性。

H.264 和 MPEG-4 两者都包括了 B 帧和比 MPEG-2、H.263 或 MPEG-4 的 SP（Simple profile）更为精确、更为复杂的运动内插滤波。为了更好地完成运动估计，H.264 显著地增加了可变块尺寸的种类和可变参考帧的数目。

H.264 的 RAM 需求主要用于参考帧图像，大多数编码视频使用 3～5 帧参考图像。它对 RAM 的需求并不比通常的视频编码器更多，因为 H.264 的 UVLC 对所有的各类数据采用了一个结构良好的查找表。

§2.4 本章小结

H.264 具有广阔的应用前景，例如实时视频通信、因特网视频传输、视频流媒体服务、异构网上的多点通信、压缩视频存储、视频数据库等。

H.264 建议的技术特点可以归纳为三个方面，一是注重实用，采用成熟的技术，追求更高的编码效率，简洁的表现形式；二是注重对移动和 IP 网络的适应，采用分层技术，从形式上将编码和信道隔离开来，实质上是在源编码器算法中更多地考虑到信道的特点；三是在混合编码器的基本框架下，对其主要关键部件都做了重大改进，如多模式运动估计、帧内预测、多帧预测、统一 VLC、4×4 二维整数变换等。

H.264 因其更高的压缩比，更好的信道适应性，必将在数字视频的通信或存储领域得到越来越广泛的应用，其发展潜力不可限量。但 H.264 优越性能的获得不是没有代价的，其代价是计算复杂度的大大增加，据估计，编码的计算复杂度大约相当于 H.263 的 3 倍，解码复杂度大约相当于 H.263 的 2 倍。

第三章　H.264 中整数变换的研究及改进

§ 3.1 概述

目前许多视频图像压缩标准（如：JPEG、MPEG、H.263）中都用到了 DCT 变换编码方法，在实现二维 DCT 时需要大量的矩阵运算、浮点运算和存储空间，这严重的影响了变换速度。为了减少运算量，缩短运算时间，人们一般采用的是快速浮点 DCT 变换。在实际应用中一般采用 DSP 芯片来实现 DCT 变换。由于 DCT 变换本质上是浮点运算，因此在很多实际应用中，如用 ASIC、定点 DSP 实现编解码时，这种浮点运算往往会带来很多的不便，如需要大量的存储空间，运算减慢，功耗大等。而目前 DSP 市场中，定点 DSP 芯片价格便宜且功耗低，占市场份额的 80%左右，浮点 DSP 仅占 20%左右。因此在通常情况下，把 DCT 的浮点运算定点化，但这又在很大程度上降低了运算精度，带来误差，降低信噪比。

针对这些问题，在视频图像压缩标准 H.264[14]的测试版中提出了整数变换[16]。这是一种以旋转矩阵为基础的近似 DCT 的变换，变换中只有整数运算，消除了浮点运算，这对于采用定点 DSP 实现视频编码有着积极的意义。减少运算量的同时降低了编码误差，提高了信噪比，并且精确的整数运算排除了编码器与解码器逆变换之间的不匹配。

为了进一步简化变换过程，H.264 标准改进了整数变换，采用基于提升结构的无乘法二进制 DCT（binDCT）[18]，binDCT 变换中只用加法和移位，减少乘法运算且保持了整数变换的优点并保证了变换效果。因此本章在基于提升结构（lifting structure）的无乘法二进制 DCT（binDCT）的基础上，先分析了如何考虑 binDCT 各指标之间的折衷，并研究了如何取用不同的 binDCT 系数以满足不同指标的需求（比如，某些应用场合更关心小的计算量，而其他应用场合更关心好的编码质量，则可选择不同的 binDCT 系数来满足要求）。然后对参数进行选择优化，选择 binDCT 参数使 binDCT 既能避免浮点运算又能降低运算量，且更有利于压缩。最后提出“动态 binDCT 系数选择”的设想，针对 I 帧、P 帧、B 帧的自身特点以及我们对 I 帧、P 帧、B 帧的质量要求不同，通过在传输比特流中附加少量的系数比特，使得可对 I 帧采用高编码质量，对 P 帧采用中等编码质量，而对 B 帧可用低编码质量，以达到总体上更好的编码效果和更低的计算量。

§3.2 整数变换的发展

3.2.1 Chen-Wang 快速浮点 DCT 算法

早期的视频压缩编码一般采用 N×N 为子块的浮点 DCT 变换，这种二维 DCT 变换通常被分解为两个一维 N 点的 DCT 变换，即先对图像做一维 N 点行变换，再做一维 N 点列变换。N 点一维 DCT 变换的公式为

$$y(k)=\sqrt{\frac{2}{N}}c(k)\sum_{n=0}^{N-1}x(n)\cos\frac{(2n+1)k\pi}{2N} \tag{3.1}$$

$$c(k)=\begin{cases}1/\sqrt{2} & k=0\\ 1 & others\end{cases} \qquad k=0,1,\cdots N-1\ .$$

为了减少运算时间，一般采用 DCT 的快速算法。在 H.263 后期的测试代码中采用了 Chen-Wang 快速浮点算法[19,20]，它是对 DCT 系数进行分解，得到一种快速的蝶式运算结构。图 3.1 给出了 4 点 DCT 运算结构图，该算法共有 8 次加法，6 次浮点乘法。其中 cθ 和 sθ 指的是 cos(θ)和 sin(θ)。

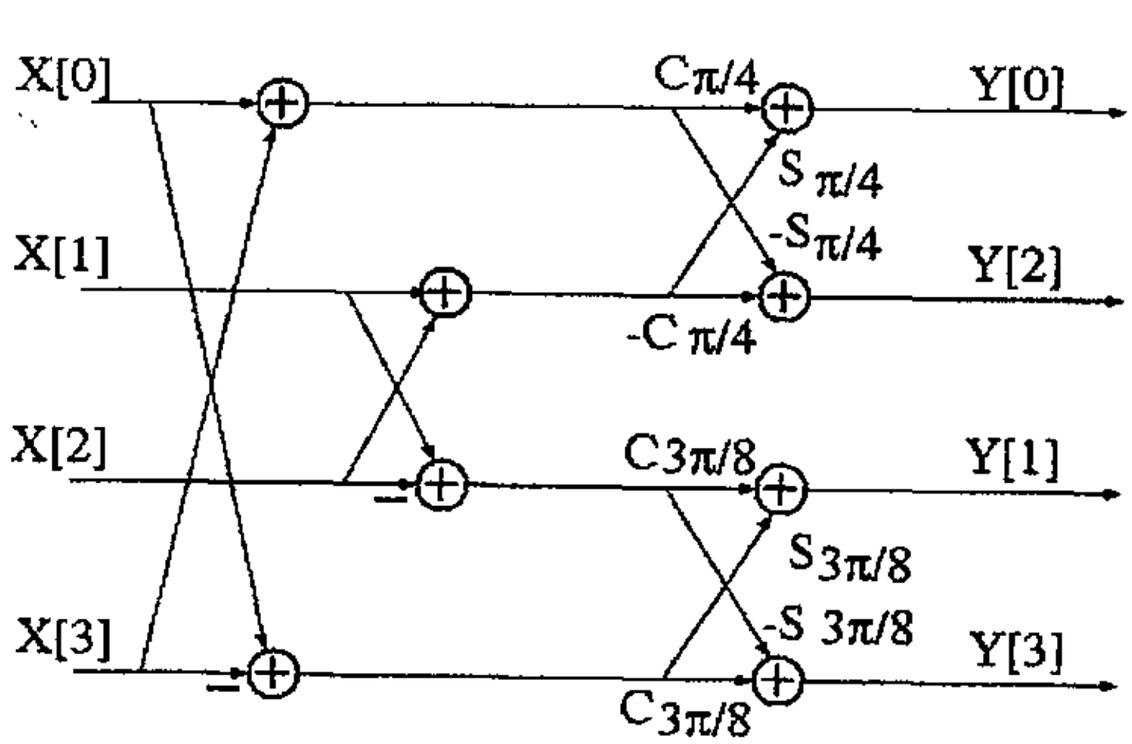

图 3.1 基于旋转变换的 4 点 DCT 的算法流程图

3.2.2 整型 DCT 变换及量化

3.2.2.1 整型 DCT 变换[18]

Chen-Wang 快速浮点算法仍用到浮点运算，为了提高运算速度，提出一种整数近似。

$$\text{整数近似:}\begin{cases}\cos\left(\frac{\pi}{4}\right)=\sin\left(\frac{\pi}{4}\right)=\frac{\sqrt{2}}{2}\Rightarrow 13\\ \cos\left(\frac{3\pi}{8}\right)=\frac{\sqrt{2-\sqrt{2}}}{2}\Rightarrow 7;\quad \sin\left(\frac{3\pi}{8}\right)=\frac{\sqrt{2+\sqrt{2}}}{2}\Rightarrow 17\end{cases}$$

且满足 $13^2+13^2=7^2+17^2$

得到了 DCT 整数变换。整型的 4×4 DCT 过程是，将像素 a，b，c，d 转换为变换系数 A，B，C，D，如下所示：

A = 13a+13b+13c+13d

B = 17a+7b-7c-17d

C = 13a-13b-13c+13d

D = 7a-17b+17c-7d

变换系数 A，B，C，D 逆转换为像素 a'，b'，c'，d'

a' = 13A+17B+13C+7D

b' = 13A+ 7B-13C-17D

c' = 13A-7B-13C+17D

d' = 13A-17B+13C-7D

a 和 a'的关系是 $a'=4\times13^2=676a$。这是因为在变换中没有进行归一化处理，归一化将在量化和反量化过程中进行，在反量化后进行平移。为了减小变换的复杂度，并保证变换运算不溢出的情况下，把移位处理放在量化过程中。

变换和反变换可在水平和垂直方向上进行。由于在反变换中使用的是精确的整数，逆变换不匹配的问题就不存在了。实现中的细节如下

原始的 4×4 块：

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

变换后的 4×4 块：

A1	B1	C1	D1
A2	B2	C2	D2
A3	B3	C3	D3
A4	B4	C4	D4

数据以如下格式从存储器中取出：

R1： a1， a2， a3， a4

R2： b1， b2， b3， b4

R3： c1， c2， c3， c4

R4： d1， d2， d3， d4

在加减操作时采用并行机制，A，B，C，D 可以这样求得：

A = 13R1+13R2+13R3+13R4

B = 17R1+7R2-7R3-17R4

C = 13R1-13R2-13R3+13R4

D = 7R1-17R2+17R3-7R4

由于是整数运算我们可以用移位加来代替乘法，例如可以用 R5：=8R1+R1 和 R5：=R5+4R1 来计算 R5：=13R1。同理垂直方向可进行相似的变换。

3.2.2.2 量化

在 H.264 标准中使用分级量化器，共支持 52 个不同的量化步长 Qstep（Quantization Step），这些量化步长用 52 个量化参数 QP（Quantization Parameter）来索引。量化步长 Qstep 和 QP 之间的关系由表 3.1 所示。

表 3.1 QP 与量化步长之间的关系

QP	0	1	2	3	4	5	6	7	8	9	10
Qstep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2
QP	11	12	13	14	15	16	17	18	19	20	21
Qstep	2.25	2.5	2.75	3.25	3.5	4	4.5	5	5.5	6.5	7
QP	22	23	24	25	26	27	28	29	30	31	32
Qstep	8	9	10	11	13	14	16	18	20	22	26
QP	33	34	35	36	37	38	39	40	41	42	43
Qstep	28	32	36	40	44	52	56	64	72	80	88
QP	44	45	46	47	48	49	50	51			
Qstep	104	112	128	144	160	176	208	224			

我们从表 3.1 中发现 QP 每增加 6，步长 Qstep 增加一倍，即 QP 每增加 1 则量化步长近似增加 12.5%。较大的量化步长范围有利于编码器精确的在比特率和视频质量之间取得很好的折衷。

由于人眼对亮度信号和色度信号的变化具有不同的反应，为了强调彩色的逼真性，对色度系数采用了较小量化步长。因此量化过程中，H.264 标准中对于亮度和色度信号采用了不同的 QP，分别记作 QP_{luma}，QP_{chroma}。这两者之间的关系是：

表 3.2 QP_I 与 QP_c 之间的关系

QP_I	<30	30	31	32	33	34	35	36	37	38	39	40
QP_c	$=QP_I$	29	30	31	32	32	33	34	34	35	35	36
		41	42	43	44	45	46	47	48	49	50	51
		36	37	37	37	38	38	38	39	39	39	39

以下两列数据将在量化和反量化中用到：

A（QP=0，..，51）=

{620，553，492，439，391，348，310，276，246，219，195，174，155，138，123，110，98，87，78，69，62，55，49，44，39，35，31，27，24，22，19，17，...}，

B（QP=0，..，51）=

{3881，4351，4890，5481，6154，6914，7761，8718，9781，10987，12339，13828，15523，17435，19561，21873，24552，27656，30847，34870，38807，43747，49103，54683，61694，68745，77615，89113，100253，109366，126635，141533，...}。

A（）和 B（）的关系为：$A(QP)\times B(QP)\times 676^2 = 2^{40}$。

传输系数 K 按以下步骤来量化：$LEVEL = (K\times A(QP) + f\times 2^{20}) / 2^{20}$，由于 $A(QP)\times B(QP)\times 676^2 = 2^{40}$ 之间是一种近似相等，加入|f|微量来进行修改补偿，其中|f|的符号与 K 相同，帧内块时 f 为 1/3，帧间块时 f 为 1/6。

反量化：$K' = LEVEL\times B(QP)$。

简要总结量化过程：

设 Y 为输入的 4×4 像素块，K_0 是它的 2 维 DCT 系数，K 为它的整数变换系数，则 $K=676K_0$。

量化后：$LEVEL \approx K\times A(QP)/2^{20} \approx 676\times K_0\times A(QP)/2^{20}$

反量化后重构信号 K'：$K' \approx LEVEL\times B(QP) \approx 676\times K_0\times A(QP)\times B(QP)/2^{20}$

由于 2 维逆变换又产生一个尺度因数 676，因此重构像素为：

$$Y'' \approx 676^2\times Y'\times A(QP)\times B(QP)/2^{20}$$

其中 Y'是由真值 DCT 重构而得，又因为 $A(QP)\times B(QP)\times 676^2 = 2^{40}$，则 $Y'' \approx Y'\times 2^{20}$。因此只要右移 20 比特（舍入）即可。

由上可见，对于真值 DCT 系数的等价量化步长为 $2^{20}/(676\times A(QP))$

§3.3 无乘法二进制 DCT（binDCT）

3.3.1 提升结构

由图 3.2 可见一个蝶型运算可以由两个提升过程[21]和两个尺度因子表示。又尺度因子可以在量化过程中加入，因此变换过程中只有两个提升过程，这使变换变得更加简洁有效。注意到在（d）中，当旋转角度接近（k+1/2）π时，这种尺度提升结构将对有限近似很敏感，且它的中间结果的动态范围也将增加。因此提出（e）这种序列改变结构，它可以很好的避免上述情况。

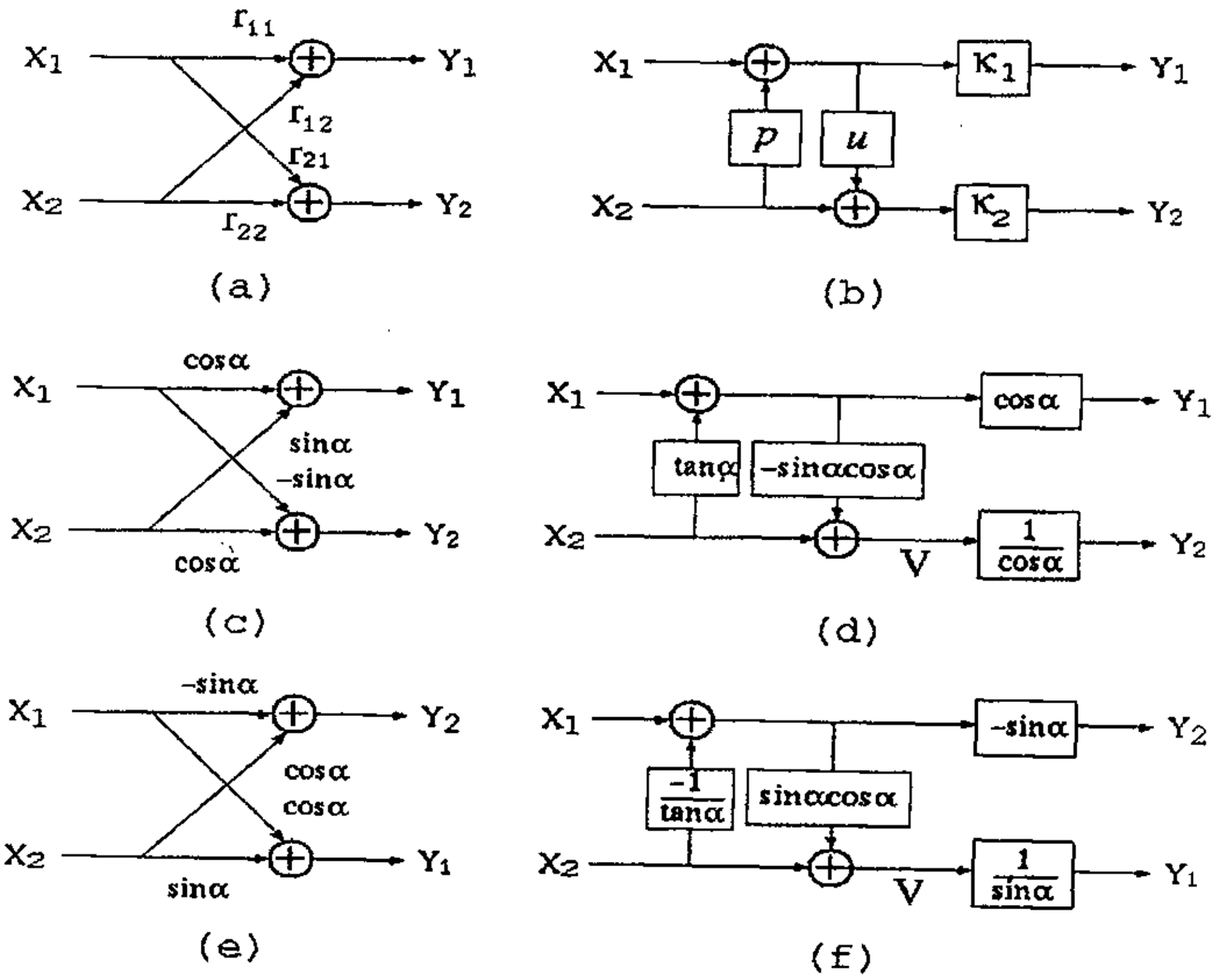

图 3.2 提升结构 （a） 普通旋转；（b） （a）的尺度提升结构图；（c） 正交旋转；（d） （c）的尺度提升结构图；（e） （c）序列改变图；（f） （e）的尺度提升结构图。

图 3.2.（b）中的参数：

$$p = r_{12}/r_{11},$$

$$u = r_{11} \times r_{21}/(r_{11} \times r_{22} - r_{21} \times r_{12}),$$

$$K_1 = r_{11}$$

$$K_2 = (r_{11} \times r_{22} - r_{21} \times r_{12})/r_{11} \text{ 。}$$

3.3.2 binDCT 的基本结构

图 3.3 为 4 点 binDCT 的基本结构图，对提升因子 p，u 取近似为 $\frac{c}{2^k}(c,k\in Z)$，则提升结构就可以实现无乘法运算，完全由移位和加法来实现。例如，$\frac{3}{8}=\frac{1}{4}+\frac{1}{8}$，提升因子$\frac{3}{8}$就可以用两个移位和一个加法来实现。

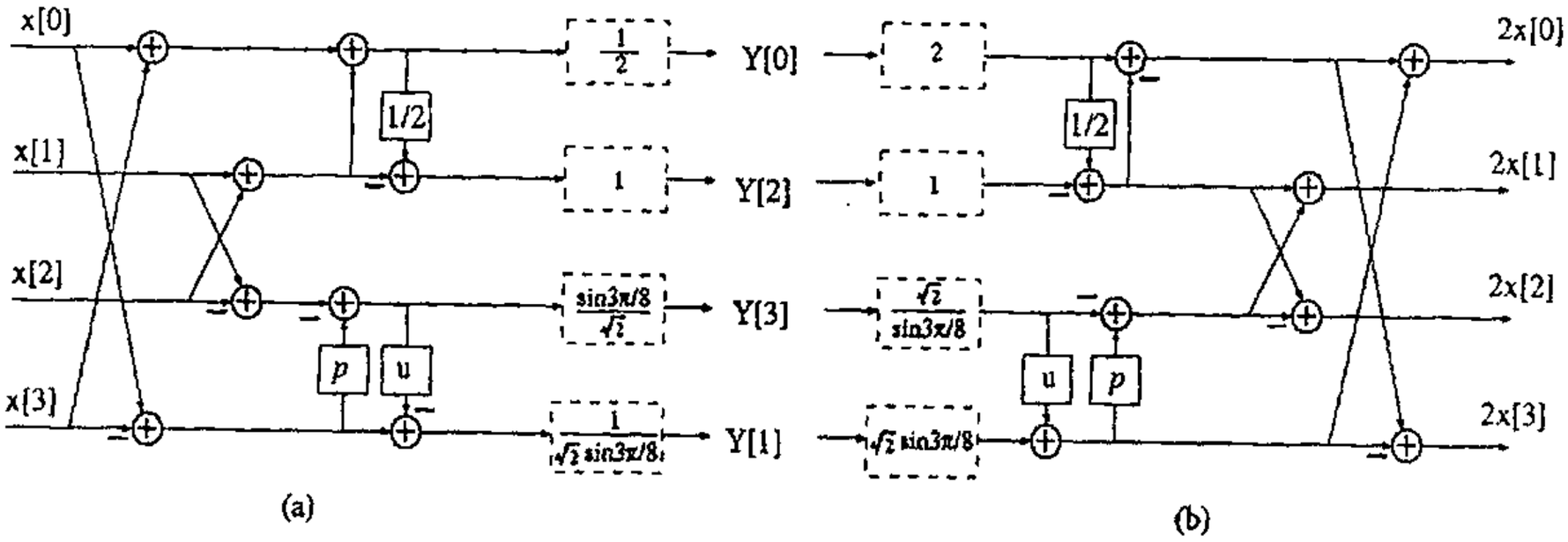

图 3.3 binDCT 变换（a）4 点 binDCT 的正变换　　（b）4 点 binDCT 的逆变换

3.3.3 尺度因子及量化

图 3.3 中的尺度因子合到量化过程中，一维的尺度因子为：

$$S1=\left[\ \frac{1}{2}\quad \frac{1}{\sqrt{2}\sin 3\pi/8}\quad 1\quad \frac{\sin 3\pi/8}{\sqrt{2}}\ \right]$$

[0.50000000000000　0.76536686473018　1.00000000000000　0.65328148243819]

相应的二维 binDCT 的尺度矩阵为：

$S2=S1^{T}\times S1=$

$$\begin{bmatrix} 0.25 & 0.38268343236509 & 0.5 & 0.32664074121909 \\ 0.38268343236509 & 0.58578643762690 & 0.76536686473018 & 0.5 \\ 0.5 & 0.76536686473018 & 1.0 & 0.65328148243819 \\ 0.32664074121909 & 0.5 & 0.65328148243819 & 0.42677669529664 \end{bmatrix}$$

其中 $S1^{T}$ 是 S1 的转置。

在目前的 H.264 测试模式中，整数变换对所有的变换系数采用的是统一的量化因子，但 binDCT 由于尺度矩阵 S2 就不能采用统一的量化因子。为了保持 binDCT 与真值 DCT 的兼容性，在 binDCT 的量化中采用一个量化矩阵 Qb，定义如下

Qb（i，j）=DCTQ（QP）/S2（i，j），　　i，j = 1 to 4。

由于首帧与其他帧及亮度与色度的 QP 不同，得到的量化矩阵也将不同。

向量 DCTQ 与 binDCT 的尺度矩阵都具有浮点运算，在实际应用中有两种方法获得整

数量化矩阵。

第一种方法，由浮点运算得到浮点的 Qb 值，然后近似为整数。

第二种方法，为了采用 16 比特定点运算得到量化矩阵，可以先将 DCTQ 扩大 8 倍，1/S2（i，j）扩大 16 倍，然后近似为整数 SDCTQ，SS2。则整数 binDCT 的量化矩阵为：

$$Qb(i,j) = (SDCTQ(QP) * SS2(i,j) + 64) / 128, \quad i,j = 1 \text{ to } 4。$$

这种方法相比于目前的测试模式，在 QP 取值较小的时候，对 binDCT 量化引入了一些舍入误差，实验发现当 QP 大于等于 8 时，这种误差可以忽略。

求得量化矩阵后，量化过程与整数变换的量化过程相似。

$$LEVEL(i,j) = (Yb(i,j) + f \times Qb(i,j)) / Qb(i,j);$$

为了加快量化速度，f×Qb（i，j），i，j=1...4 可事先求出。在目前的测试代码中，f 可以由三种值：帧内块 1/3，帧间块 1/6，0.45 率失真最优化量化。因此，可以得到 8 个 4×4 矩阵 Qb_{chroma}，$1/3Qb_{chroma}$，$1/6Qb_{chroma}$，$0.45Qb_{chroma}$，Qb_{luma}，$1/3Qb_{luma}$，$1/6Qb_{luma}$，$0.45Qb_{luma}$。其中 0.45Qb 可以由 Qb/2-Qb/20 得到。

反量化过程：$Yb'(i,j) = LEVEL(i,j) \times Qb(i,j)$。

3.3.4 动态范围分析

预测后输入 binDCT 的数据是 9 比特有符号整数（-255　255）。为了获得 binDCT 的动态范围，我们研究 binDCT 的符号，设计输入数据使输出得到极值。

例如：第二行为[107/128　3/8　-3/8　-107/128]，则输入为[255，255，-255，-255]可获得最大值，输入为[-255，-255，255，255]可获得最小值。

在提升图中，发现提升参数都是小于 1 的，它们可以由加法和移位来实现，这就减少了运算中间结果的范围（图 3.4）。第一行具有最大的绝对和，则 binDCT 的动态范围可由 DC 子带决定。一维 4 点 binDCT 变换从 9 比特到 11 比特，二维从 9 比特到 13 比特。这是最坏情况，因此 binDCT 完全适应 16 比特结构。

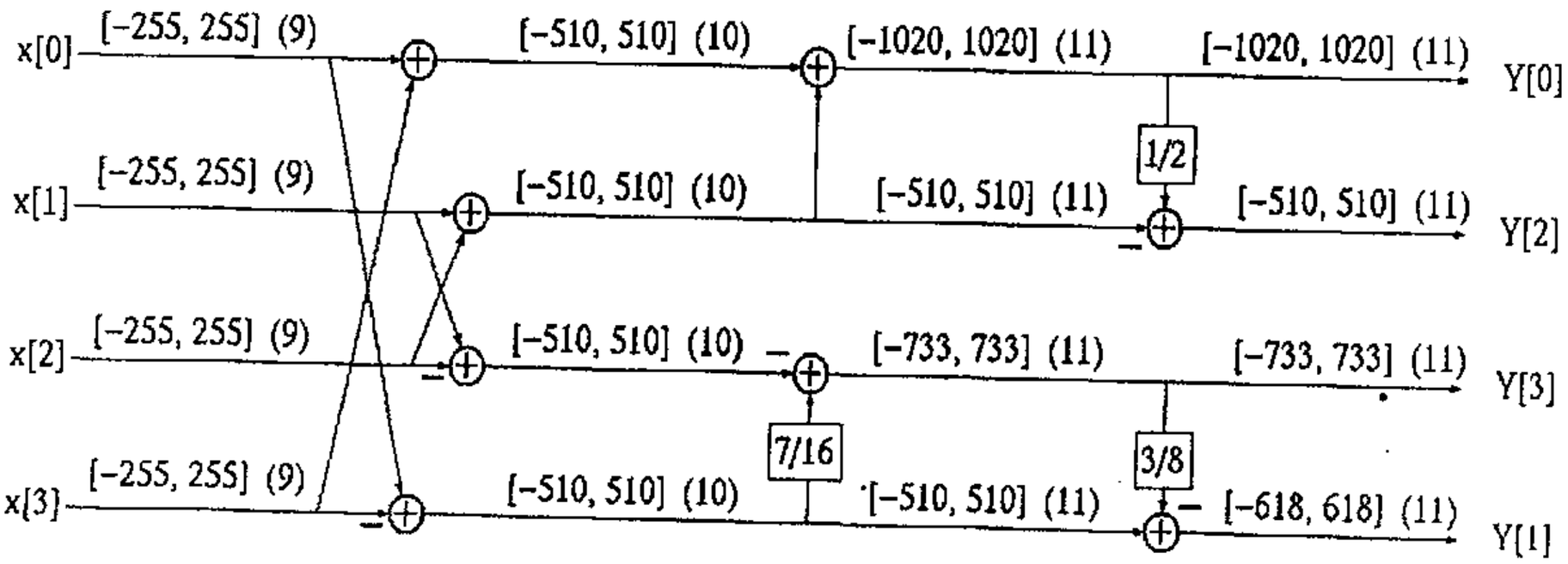

图 3.4 一维 4 点 binDCT 中间结果的动态范围

§3.4 binDCT 参数选择的优化及实验结果

将 binDCT 参数提升因子 $p = 1/tg\frac{3\pi}{8}$，$u = \sin\frac{3\pi}{8}\cos\frac{3\pi}{8}$ 取值近似为 $\frac{c}{2^k}(c,k \in Z)$，则提升结构就可以实现无乘法运算，完全由移位和加法来实现。例如，3/8 = 1/2 – 1/8，提升因子 3/8 就可以用两个移位和一个加法来实现。在表 3.3 中列出几组 p，u 的近似值，及其计算复杂度。研究发现 k 取值越大则越能更好的对 p，u 进行近似，因此也就越接近真值 DCT。当 $p=7/16$和$u=3/8$时，编码增益 7.5697 与真值 DCT 的编码增益 7.5701 十分相近，而且相比于真值 DCT 需要 6 次浮点乘法和 8 次加法，它只需要 5 次移位和 10 次加法。但注意到 k 值越大，所需的移位和加法次数也在变大。因此，在应用中我们应该根据实际需要，即在质量要求高的场合采用较大的 k 值来获得更好的编码质量，而在要求计算量小的场合，我们适当的减小 k 值，以满足实际计算量的要求。并且我们还注意到$\frac{3}{8}=\frac{1}{2}-\frac{1}{8}$，只需要进行右移，这也减少了中间结果的动态范围。

表 3.3 binDCT 的结构及性能

BinDCT 结构	真值 DCT	1	2	3	4
P	0.414213562373l0	7/16	3/8	1/2	1/2
U	0.353553390593 27	3/8	3/8	3/8	1/2
移位次数	-	5	5	4	3
加法次数	-	10	10	9	8
编码增益（dB）	7.5701（真值 DCT）	7.5697	7.5566	7.5493	7.5485

本实验以 H.264 码流结构的典型图像序列 Foreman 及 Miss American 的 QCIF 格式的各 100 帧图像序列为图像源，就 DCT，整数变换及 binDCT 变换编码的效果加以比较，采用的主要比较指标是峰值信噪比（PSNR）。

图 3.5 所示为 DCT，整数变换及 binDCT 这三种变换的峰值信噪比比较图，其中 binDCT 提升因子取值 $p = 7/16$ 和 $u = 3/8$。实验结果可见整数变换的 PSNR 要高于 binDCT 变换，略低于浮点 DCT 变换。但相比于浮点 DCT 需要 6 次浮点乘法和 8 次加法，整数变换 8 次移位和 16 次加法，binDCT 只需要 5 次移位和 10 次加法，其计算量及存储量远小于浮点 DCT 和整数变换，同时 binDCT 变换效果与整数变换相差不大且变换复杂度减少。

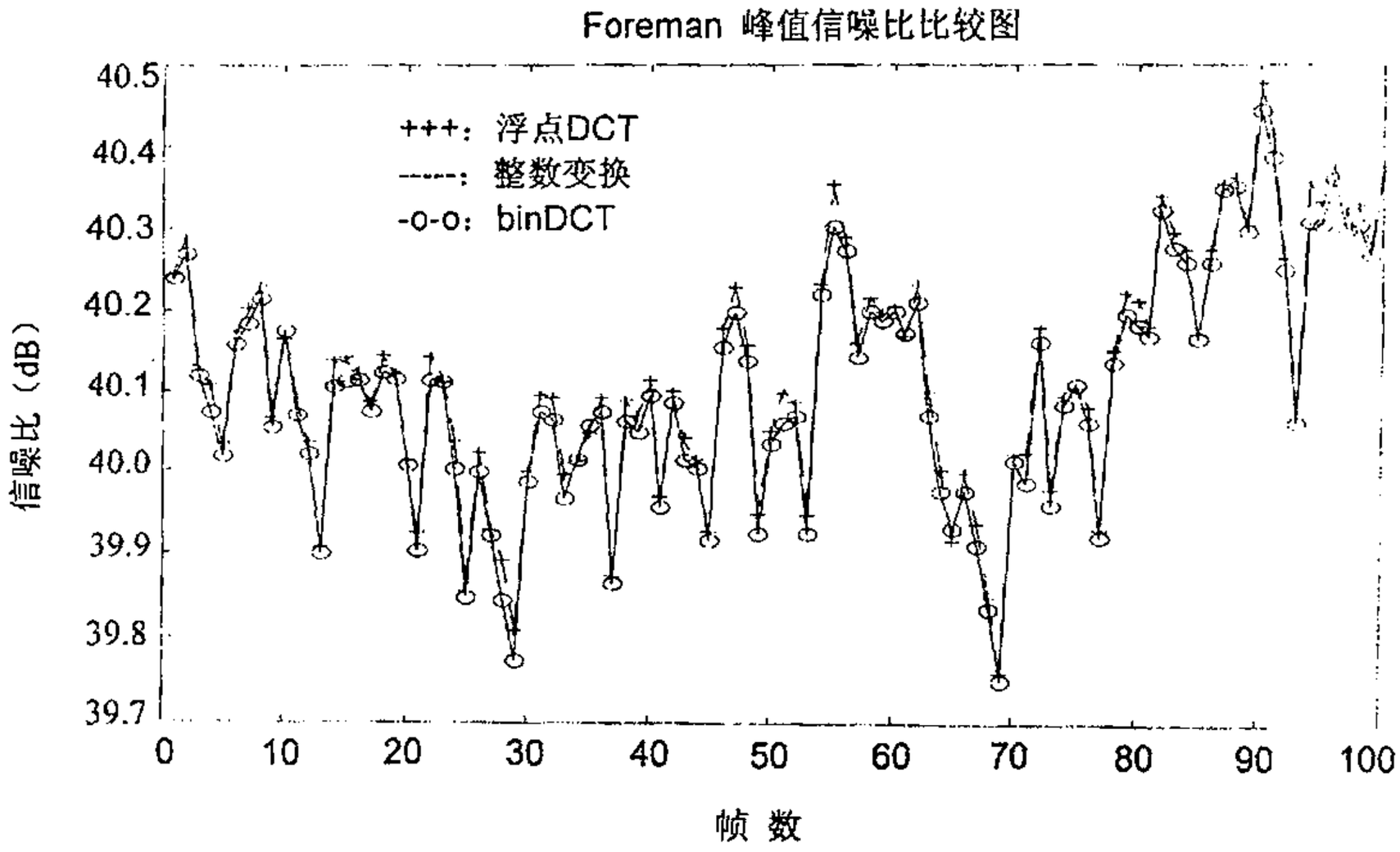

图 3.5 浮点 DCT、整数变换、binDCT 的 PSNR 比较

实验的第二步仍以典型图像序列 Foreman 及 Miss American 的 QCIF 格式的各 100 帧图像序列为视频源，在 binDCT 基本结构的基础上，选择不同的参数（表 3.4）做研究比较，采用的主要比较指标是峰值信噪比（PSNR）。

当本文对 p，u 值取进一步近似 B1 时，实验结果（表 3.5，图 3.6）表明我们所选的参数 B1 比文献[18]中的 B2、B3 更接近于浮点 DCT 的结果，但是相比于 B3 的 3 次移位和 8 次加法，B2 的 5 次移位和 10 次加法，B1 却需要 9 次移位和 14 次加法。可见精度的提高是以运算量增加为代价的。因此在应用中如何合理的选择 p，u 值，应该结合实际需要（变换质量和计算复杂度之间的折衷）进行考虑。一般情况下考虑到是对图像的像素进行变换，因此 p，u 取值 B2 是一种折衷的办法，既减少了运算量又具有很好的变换效果。

表 3.4 binDCT 的几组参数及运算复杂度

BinDCT 结构	真值 DCT	B1	B2	B3
P	$1/tg\ \frac{3\pi}{8}$	53/128	7/16	1/2
U	$\sin\frac{3\pi}{8}\cos\frac{3\pi}{8}$	45/128	3/8	1/2
移位次数	-	9	5	3
加法次数	-	14	10	8

binDCT 变换的正变换和逆变换矩阵（没有考虑尺度因子）：

$$\text{B1：正变换}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 13999/128^2 & 45/128 & -45/128 & -13999/128^2 \\ 1/2 & -1/2 & -1/2 & 1/2 \\ 53/128 & -1 & 1 & -53/128 \end{bmatrix}$$

$$\text{逆变换}\begin{bmatrix} 1/2 & 1 & 1 & 45/128 \\ 1/2 & 53/128 & -1 & -13999/128^2 \\ 1/2 & -53/128 & -1 & 13999/128^2 \\ 1/2 & -1 & 1 & -45/128 \end{bmatrix}$$

$$\text{B2：正变换}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 107/128 & 3/8 & -3/8 & -107/128 \\ 1/2 & -1/2 & -1/2 & 1/2 \\ 7/16 & -1 & 1 & -7/16 \end{bmatrix}$$

$$\text{逆变换}\begin{bmatrix} 1/2 & 1 & 1 & 3/8 \\ 1/2 & 7/16 & -1 & -107/128 \\ 1/2 & -7/16 & -1 & 107/128 \\ 1/2 & -1 & 1 & -3/8 \end{bmatrix}$$

$$\text{B3：正变换}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 3/4 & 1/2 & -1/2 & -3/4 \\ 1/2 & -1/2 & -1/2 & 1/2 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

$$\text{逆变换}\begin{bmatrix} 1/2 & 1 & 1 & 1/2 \\ 1/2 & 1/2 & -1 & -3/4 \\ 1/2 & -1/2 & -1 & 3/4 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

表 3.5 浮点 DCT、binDCT 的 PSNR 均值

PSNR(dB)	Y		U		V	
QCIF（100 帧）	Foreman	Miss American	Foreman	Miss American	Foreman	Miss American
DCT	37.5272	41.4673	40.7827	42.4877	42.0582	41.5421
B1	37.5269	41.4670	40.7817	42.4874	42.0580	41.5420
B2	37.5050	41.4233	40.7642	42.4760	42.0472	41.4958
B3	37.3788	41.2344	40.7058	42.3888	41.9935	41.2671

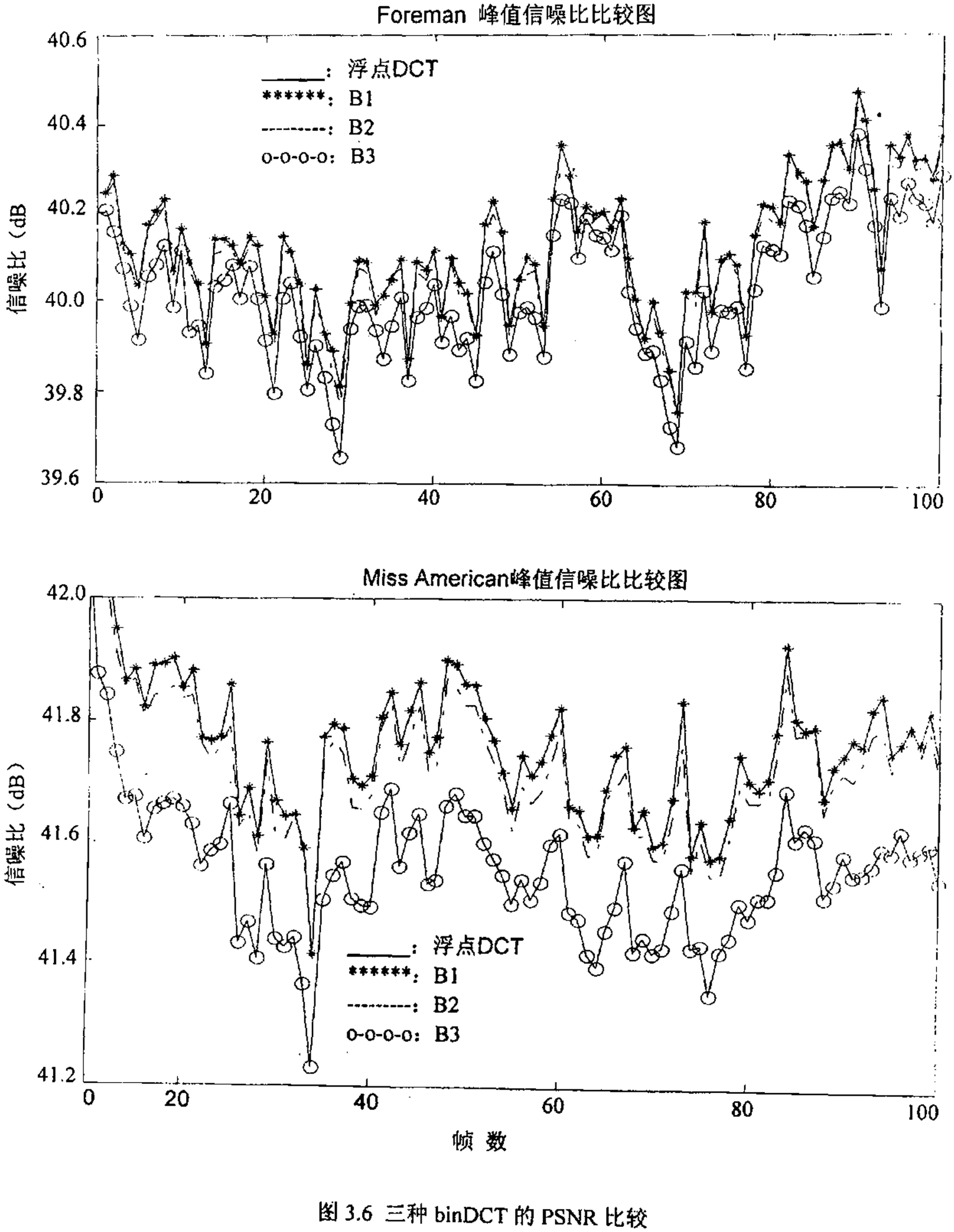

图 3.6 三种 binDCT 的 PSNR 比较

H.264 中引入了三种主要的帧类型，内部编码帧（I 帧），预测编码帧（P 帧），双向预测编码帧（B 帧）。I 帧的编码不需要参考别的帧，这类帧提供了对编码图像数据序列的访问点，使解码可由此点开始，但这类帧采用了最普通的压缩方法。P 帧采用了最有效的编

码方法，它使用运动补偿，通过过去的 I 帧或 P 帧来预测，一般也可作为后面预测的参考帧。B 帧实现了最高程度的压缩，但需要使用过去的和未来的参考帧来进行运动补偿，有助于平滑噪声，并且它不可再作为别的预测的参考帧，误差不会传播。针对 I、P、B 帧的自身特点，我们提出“动态 binDCT 系数选择”的设想。实际中对 I、P、B 帧的变换质量要求不同，对于 I 帧我们希望变换量化后失真尽量小，因为它要作为后续 P、B 帧的参考；P 帧只作为后续 P 帧的参考，相对于 I 帧，它的变换量化失真的要求可降低一些；而 B 帧不作为其它帧的参考，它对变换失真的要求最低。因此可在编码时考虑选择不同的 binDCT 参数，在编码效果和计算量之间取得很好的折衷。通过在传输比特流中附加少量的系数比特，在编解码时可实现在 I、P、B 帧之间灵活的调整 p，u 值，使得对 I 帧采用高编码质量，对 P 帧采用中等编码质量，而对 B 帧可降低编码质量，以达到总体上更好的编码效果和更低的计算量。

§3.5 本章小结

本章介绍了 H.264 中整数变换的发展及应用，分析了如何考虑 binDCT 各指标之间的折衷，并研究如何取用不同的 binDCT 系数以满足不同指标的需求。实验证明应用 binDCT 并选择合适的参数可使变换只用到加法和移位，且变换复杂度低于早期的整数变换而变换效果接近浮点 DCT，避免了浮点运算，减少了运算量，提高运算效率。本章最后提出“动态 binDCT 系数选择”的设想，在实际图像处理时，合理的动态的选择 binDCT 系数可使图像达到整体上更好的编码效果和更低的计算量。

第四章 基于 H.264 的视频传输抗误码技术的研究

§ 4.1 概论

由于信道噪声的影响，视频信息在传输过程中会发生改变或丢失，这种传输误码会导致视频重建质量严重下降。因此视频通信中的抗误码技术的研究是一个十分重要的课题。

H.264 标准采取了一些抵御传输差错的措施，视频流中的时间同步可以通过采用帧内图像刷新来完成，空间同步由条结构编码（slice structured coding）来支持。同时为了便于误码以后的再同步，在一幅图像的视频数据中还提供了一定的重同步点。另外，帧内宏块刷新和多参考宏块允许编码器在决定宏块模式的时候不仅可以考虑编码效率，还可以考虑传输信道的特性。

然而，由于 H.264 的高效率编码，使得视频码流对信道误码率非常敏感，即使单个原发性错误，也可能会造成重构视频质量的急剧下降。并且 H.264 标准中采用普通的可变长码（UVLC）和基于上下文的二进制算术编码（CABAC）的方法来编解码[22,23]。在视频传输中一旦遇到噪声干扰、出现传输误码，就会造成误码扩散，将严重影响重建视频的质量。因此从宏块熵编码模式出发，帧内编码宏块的抗误码性能需要较大的增强。

本章在 H.264 编码算法基础上，采用视频抗误码方法，来保证恢复图像的质量。根据 H.264 编解码的特点，在 H.264 测试版中引入了一种基于算术编码（AC）能连续检错（CED）的自动重传请求（ARQ）方案[24,25,26]，使得当 H.264 码流在有噪信道中传输时，加入新的检错机制后能连续检错并能发出重传请求，这大大提高了 H.264 在视频传输时的抗误码能力，并在增加冗余量和提高检错性能之间取得较好的折衷，获得较好的压缩效果和抗信道误码能力。

§ 4.2 基于 H.264 的抗误码技术的研究

4.2.1 传输误码的分类及抗误码的基本方法

任何一个视频通信系统都固有存在的问题是视频信息在传输过程中由于信道噪声使得信息改变或丢失，这种信息的改变或丢失会导致视频重建质量严重下降。由于目前绝大多数视频编码标准都采用了运动预测补偿技术以及可变长编码技术，因此，误码不仅影响

该误码数据的恢复，还会影响与之相关的其他数据的恢复，造成误码扩散（Error Propagation）。

传输误码大体上可分为两类：一类是随机误码，这主要是由于信道的物理缺陷而致，比如比特位跳转（Bit inversion）、比特位插入（Bit insertion）、比特位删除（Bit deletion）等。因编码方法不同或者信息内容不同，这种随机误码造成的影响也不同，某些随机误码可以忽略不计，某些可能非常严重。当采用定长编码方法时，一个随机比特位的错误只影响一个码字，所造成的信息失真一般是可以接受的；而当采用可变长编码（VLC，如哈夫曼编码、算术编码）时，它会导致编长码解码器同步丢失，同时产生两种明显的误码：（1）误码直接使某码字映射为一非法码字，这种情况解码器可以立即检测到该误码。（2）误码使某码字映射为另一个合法码字，或者映射为某合法的码字序列，那么解码器就很难立即检测出误码的存在，但由于误码扩散的缘故，解码器可以在随后的某个位置发现误码。一个随机比特位的错误会使比特流失去同步，使得该错误发生位置到下一个同步码之间的所有码字都变为无效码字，甚至因为解码得到的信息无法和空间位置对应起来使得同步码以后的码字都变为无用信息，因此在可变长编码中，这种误码对视频重建质量的影响很大。

另一类是突发错误，比如包交换网络中的包丢失（Packet loss）、存储媒体的物理损伤等。突发错误会导致视频质量严重下降，甚至无法解码，但这类错误一般可以根据传输协议检测到。目前视频压缩编码大多都采用可变长编码，因此这两种误码都会导致重建视频质量严重下降。

视频通信中的抗误码技术长期以来一直沿着两条路发展：（1）把数据通信中的误码控制和恢复技术扩展到视频通信中，这些技术的目标是无损恢复。例如：前向错误校正（FEC）、错误控制编码（ECC）、自动重传请求（ARQ）等；（2）旨在恢复近似于原始信息或根据人类生理特性的误码掩盖技术。比如对于视频图像，人眼可以容忍一定程度的失真，或者是某些失真人眼根本无法觉察到，而不需要像数据通信那样要求绝对的无损恢复。

目前主要有三种提高视频传输抗误码能力的处理方法：第一种是从编码端出发考虑，往编码输出码流中加一些额外的冗余信息，使得码流不需要解码器做更多的处理就能够极大的降低传输差错的影响，如前向错误校正（FEC）、错误控制编码（ECC）等。然而信道误码情况是时变的，目前还没有一个精确的数学模型来描述信道的变化。因此如果根据信道最差的情况来加入冗余信息进行保护，则在大多数信道较好的时间内浪费了信道资源；如果根据信道最好的情况加入冗余信息，则信道变差时会带来误码倍增效应。此外，还有分层编码（Layer coding）、多描述编码（Multiple description coding）等都是从编码器端考虑尽量使编码后的码流受误码影响最小。

第二种是从解码端考虑，对于已经发生的错误，利用已接收的信息来估计发生错误的信息，加以处理使其尽量不被人眼觉察出来，而不需要从编码器得到额外的信息，即误码掩盖技术。它利用人眼视觉的特性，用视频序列的空间、时间相关性来掩盖误码带来的不良影响，这一技术可分别在空域、时域、频域内进行。

第三种是编解码两端交互进行数据恢复，编码器根据解码端反馈回来的信息自适应地调整编码参数和编码模式等，最典型的是自动重传请求（ARQ）策略。这种方法要依赖于反馈信道。

本章针对 H.264 编解码的特点，将第一种和第三种方法联合考虑，将一种基于算术编码（AC）能连续检错（CED）的自动重传请求（ARQ）方案引入到现行的 H.264 测试版中。

4.2.2 H.264 中的编码方式及基于 AC 的具有 CED 的 ARQ 方案

4.2.2.1 基于上下文的二进制算术编码（CABAC）[22,23,27]

H.264 中被用来编解码的参数如表 4.1 所示。H.264[14]中片层（slice）结构以上的语法元素用定长码或变长码编码，片层或片层以下的语法元素根据熵编码模式来选择是用普通的可变长编码（UVLC）还是用基于上下文的二进制算术编码（CABAC）来编解码。

表 4.1 H.264 中的编解码参数

参数	描述
序列层，图像层及片层语法元素	—
宏块类型 Mb_type	每个编码宏块的预测方式
编码块模式	指示宏块中含有编码系数的块
量化参数	从前一个 QP 值传输
参考帧索引	帧间预测时确认参考帧
运动向量	从预测向量中得到的 MVD
残差数据	每个 4×4，2×2 块的系数数据

H.264 中 UVLC 提供了一个简单和健壮的方法来编码模式信息和 DCT 系数，但在高比特率时它的性能并不好，因此当熵编码模式被选为“1”时，H.264 中将采用一种称作基于上下文的自适应二进制算术编码（CABAC）的算术编码模式来编解码。它在如下三个条件下具有很好的压缩性能，即：（a）根据每个元素上下文来获得概率模型，（b）根据当时统计来调整概率估计，（c）使用算术编码。

CABAC 具有 3 个独特的优势。1、上下文模型提供了编码符号的条件概率的估计。2、算术编码允许给每个符号分配非整数比特。3、自适应算术编码熵编码器适合无固定符号统计，对于每个块来说，我们将原始图像和低质量预测而得的图像之间的差进行编码。

编码一个数据元素包括以下几个步骤（图 4.1）：

1、二进制化：CABAC 使用二进制算术编码，这意味着用来编码的元素仅仅是二进制元素（1，0）。一个非二进制元素（例：传输系数或运动向量）要先二进制化或者在算术

编码之前转化为二进制码（如表 4.2）。由于 MB_type 的范围限制在（0～9），因此可用另一种方法二进制化。二进制化的过程类似于将一个数据元素转化为变长码，但不同之处在于它在传输之前还要经过算术编码。

表 4.2 二进制化

（a）除 MB_type 外其它符号的二进制化

Code_number	Binarization
0	1
1	0 1
2	0 0 1
3	0 0 0 1
4	0 0 0 0 1
5	0 0 0 0 0 1
6	0 0 0 0 0 0 1
.	...
Bin_number	1 2 3 4 5 6 7 ..

（b） MB_type 的二进制化

MB_type	Binarization
0	0
1	1 0 0
2	1 0 1
3	1 1 0 0 0
4	1 1 0 0 1
5	1 1 0 1 0
6	1 1 0 1 1
7	1 1 1 0 0
8	1 1 1 0 1
9	1 1 1 1 0
Bin_number	1 2 3 4 5

2、选择上下文模型：上下文模型就是二进制符号中一个或多个二进制位的概率模型。这个模型将根据最近编码的数据符号的统计表来选择。上下文模型存储了每个二进制位“1”或“0”的概率值。

标准为每种语法元素定义了上下文模型和二进制化方式，对于不同的语法元素共有 267 种上下文模型，0 到 266（2002.9）。根据不同的片（slice）类型，一些模型具有不同的用法。例如，在 I 片中不允许有跳宏块，因此根据当前块是否是帧内编码模式来用上下文模型 0-2 编码 mb_skip 或 mb_type。每个编码片开始时，根据量化参数来初始上下文模型的（因为这对于不同数据符号将有很大的影响）。

3、算术编码：算术编码（Arithmetic coding）的思想最早在 1948 年 Shannon 的文章[28]中出现，当时 Shannon 将符号依概率排序，用累积概率的二进制表示作为对符号的编码。60 年代 Elias 发现，不需要排序，只要编码端和解码端使用相同的符号顺序就可以了[29]。但是当时人们对算术编码的理解仍然需要无限精度的浮点运算，或者随着符号的输入，需要的精度增加，同时计算时间也线性增加。1976 年前后 Rissanen 和 Pasco 分别完成一个使用有限精度运算的算术编码[30]，但是它们分别是离线的和先入先出的。Rissanen 和 Langdon 的文章[31]描述了算术编码的基本理论，并给出了一个二进制算术编码器。1987 年 Witten 等人发表了一个实用的算术编码程序[32]，即 CACM87。CACM87 被用作 H.264 的一部分。

算术编码器根据所选的概率模型对每个二进制位编码。注意：仅有“0”“1”参与编码。算术编码具有三个不同的特性：

a、概率估计通过在 LPS（LPS，“0”“1”中最小可能符号）的 64 个不同的概率状态的转换过程中完成。

b、在每一步计算新的范围前，算术编码器的当前状态 R 被量化为一个小范围预设值，使之可以使用查表法来计算新的范围（避免乘法）。

c、数据元素的简化的编解码过程具有一致的概率分布定义。

4、概率更新：根据目前编码值来更新概率模型。（例如，当前编码值为“1”则“1”的频率值增加）。

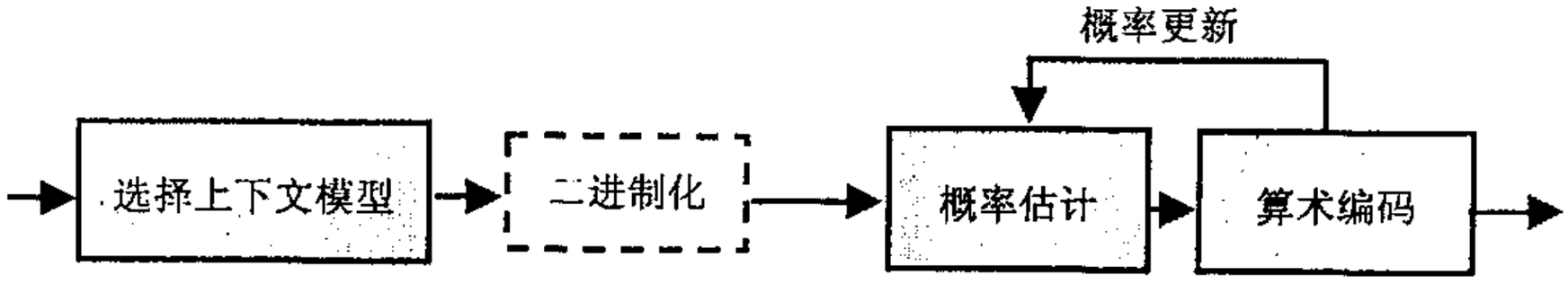

图 4.1 CABAC 编码过程简图

总的来说，CABAC 相比于 VLC 改进了编码效率，代价是增大了计算复杂度。CABAC 是根据每个元素的上下文来获得概率模型并按实时统计来调整概率模型的自适应算术编码。算术编码可以逼近信源的熵，因此 CABAC 具有很好的压缩性能。但算术编码是变长编码，随机误码会使误码扩散，导致在重同步之前的解码为误码，所以算术编码的抗误码能力较弱。

4.2.2.2 基于 AC 的具有 CED 的 ARQ 方案[24,25]

因此本章提出在 H.264 的算术编码中嵌入差错控制，使其在一定范围内能够检错。其基本思想是在编码端加入冗余，使编码区间中的部分区间，编码端不使用。当解码端发现进入这些区间时，则表明数据传输过程中有错误发生。因此解码端报错，向编码端发出自动重传请求，要求其重传一段长度为 n 的数据。由于误码的扩散，准确对误码进行定位是比较困难的，因此只在某个概率范围内确定重传数据范围 n 的值。

在 H.264 的应用中，算术编码的编解码端具有相同的当前区间，是由它的长度 δ 和低限 l 决定。为了进行有限的运算，编码端移出一些比特再将区间加倍。由于每次区间加倍表示一比特移出，这是加入冗余的理想位置，可在该位置对当前编码区间按一定的比例进行调整。该比例称为缩减因子 α [24,25]，它决定着输出冗余度的多少，同时控制着检错的速度。

设当前区间为 δ，则需要 $I(\delta) = -\log_2(\delta)$ 比特来编码表示。当前区间乘上缩减因子 α 后，则需要 $I(\alpha\delta) = I(\alpha) + I(\delta)$ 比特来编码表示，冗余量为 $I(\alpha)$。

$$I(\alpha) = -\log_2(\alpha) \text{ 比特/编码符号} \tag{4.1}$$

假设加入 x 的冗余，则选择 $\alpha = 1/2^x$。例 x=1%，则 α =0.9931；x=2%，则 α =0.9862。

缩减因子α又可理解为在算术编码过程中加入一个禁用符号μ，假设其发生概率为$e=1-\alpha$。例如，当编码一个符号$x\in\{a,b,c\}$时，设当前子区间δ，则$\alpha\delta$分配给x，$e\delta$分配给μ。由于μ不被编码，所以分配给μ的间隔不会被再分割。设$P_n(\mu)$是编完第 n 个符号后μ所占的概率。

$$P_n(\mu)=\sum_{k=1}^{k=n}e(1-e)^{k-1}=1-(1-e)^n \tag{4.2}$$

判断误码的方法是看当前区间是否在禁用区间内，因此$P_n(\mu)$又可理解为是当发生一个错误时，经过 n 比特我们能判定错误的概率。

相应的一个错误发生后经过 n 比特不能检错的概率β：

$$\beta=1-P_n(\mu)=(1-e)^n \tag{4.3}$$

可见错误发生后随着解码符号 n 的增加，不能检错的概率β成指数减少。因此，直到检出错误，错误发生在前 n 比特的概率为$(1-\beta)$。又冗余量$I(\alpha)=-\log_2(1-e)$比特/编码符号，带入（4.3）式，可以得到检错所需重传的比特 n 和冗余量$I(\alpha)$之间的关系：

$$n=\frac{\log_2(\beta)}{\log_2(1-e)}=\frac{\log_2(\beta)}{I(\alpha)} \tag{4.4}$$

由（4.4）式可知，冗余量与检错所需重传的比特之间成反比。编码信息中的冗余可由编码过程中这一简单可协调的参数α控制，但这又决定于信道条件，这对于 ARQ 很重要。某些时候要求快速检错则需加入比较大的冗余，而有些时候信道平稳更重要则可适当加长检错时间。因此参数α可根据检错要求和信道条件来进行综合的考虑和选择。

§4.3 应用设计及结果分析

由于相同的噪声对不同的帧类型（I、P、B）的影响是不同的，I 帧作为 P、B 帧的参考帧，因此误码发生在 I 帧后会出现误码扩散，对重建图像质量的影响较大。而 B 帧不作为其它帧的参考帧，发生在 B 帧的误码只会局限在该帧内，不会影响其它帧。因此应用过程中应考虑到针对不同的帧类型取不同的缩减因子α值。

本实验以 H.264 码流结构的典型图像序列 Foreman 及 Miss American 的 QCIF 格式（176×144）的 100 帧图像序列为例，序列中这三种类型的图的组织是非常灵活的，在本实验中采用 IBBPBBPBB 的组合。在相同的信道条件下（设信道产生10^{-3}的随机噪声），对 100 帧图像中的 I、P、B 帧分别加噪声，再分别采用不加任何额外检错机制和加入基于 AC 的 CED+ARQ 检错机制这两种方法得出的效果进行比较，其中缩减因子α分别取为 0.75，0.8，0.85，0.9，0.95，1（零冗余）。

应用过程中，在编码端根据α加入冗余，解码端一旦发现解码后当前区间落在缩减区间之外就报错，记下当前数据在数据包中的位置并根据（4.4）式计算 n 值。再向编码端发出重传请求，并将当前数据的位置信息及 n 值发给发送端，发送端将当前数据前 n 个数据重发，同时解码端将解码状态返回至前 n 个数据时，再根据接收到的数据解码。解码每个数据时，必须纪录当前解码状态前的若干状态，这将较大的占用了内存。实际应用中我们只保留前 100 个解码状态，由于 H.264 中算术解码是帧同步的，若 n 大于 100 则将该帧数据重发。

H.264 中数据处理是串行的，即一个宏块先 AC 解码后进行运动补偿等操作，然后再处理下一个宏块。考虑到在 AC 解码时就能报错，应用时可以利用数据处理的并行性，将一帧内的宏块统一 AC 解码然后无错就继续下面操作，一旦某块数据报错就退出，进行误码处理。

下面给出分别采用不加任何检错机制和加入基于 AC 的 CED+ARQ 检错机制这两种方法得出的实验结果图。实验的信道条件为随机产生 10^{-3} 的白噪声，本实验采用 IBBPBBPBB 的组合，编码采用 CABAC，图像序列选用了 Miss American 的 QCIF 格式（176×144）的 100 帧图像序列。由于噪声加在 I、P、B 帧对整个图像的序列的信噪比产生的影响不同，因此实验过程中我们分别对整个图像序列中的 I、P、B 帧分别加入噪声后得到不同的信噪比图。然后分别加入本文提出的基于 AC 的 CED+ARQ 检错机制，可以得到检错后的图。由于检错后得到的信噪比都近似于无噪状态下图像，所以本图只列出一条检错后曲线。H.264 中采用 YUV 表示法来表示彩色图像，其中亮度信号 Y 和色度信号 U，V 是相互独立的。图 4.2 为亮度信号 Y 的 PSNR 图，图 4.3 为色度信号 U 的 PSNR 图，图 4.4 为色度信号 V 的 PSNR 图。实验结果表明相同噪声信道条件下，采用 CED+ARQ 方案后 H.264 的抗误码能力明显优于原先不采用任何检错机制的情况。在 I 帧，P 帧加噪声时 Y 信号的信噪比提高 8dB，UV 信号也分别提高 1dB 和 3dB。在 B 帧加噪声时 Y 信号的信噪比提高 4dB，UV 信号也提高 0.5dB。对应于缩减因子α分别取为 0.75，0.8，0.85，0.9，0.95 相应的冗余度 47.9%，32.75%，21.63%，12.38%，4.98%。

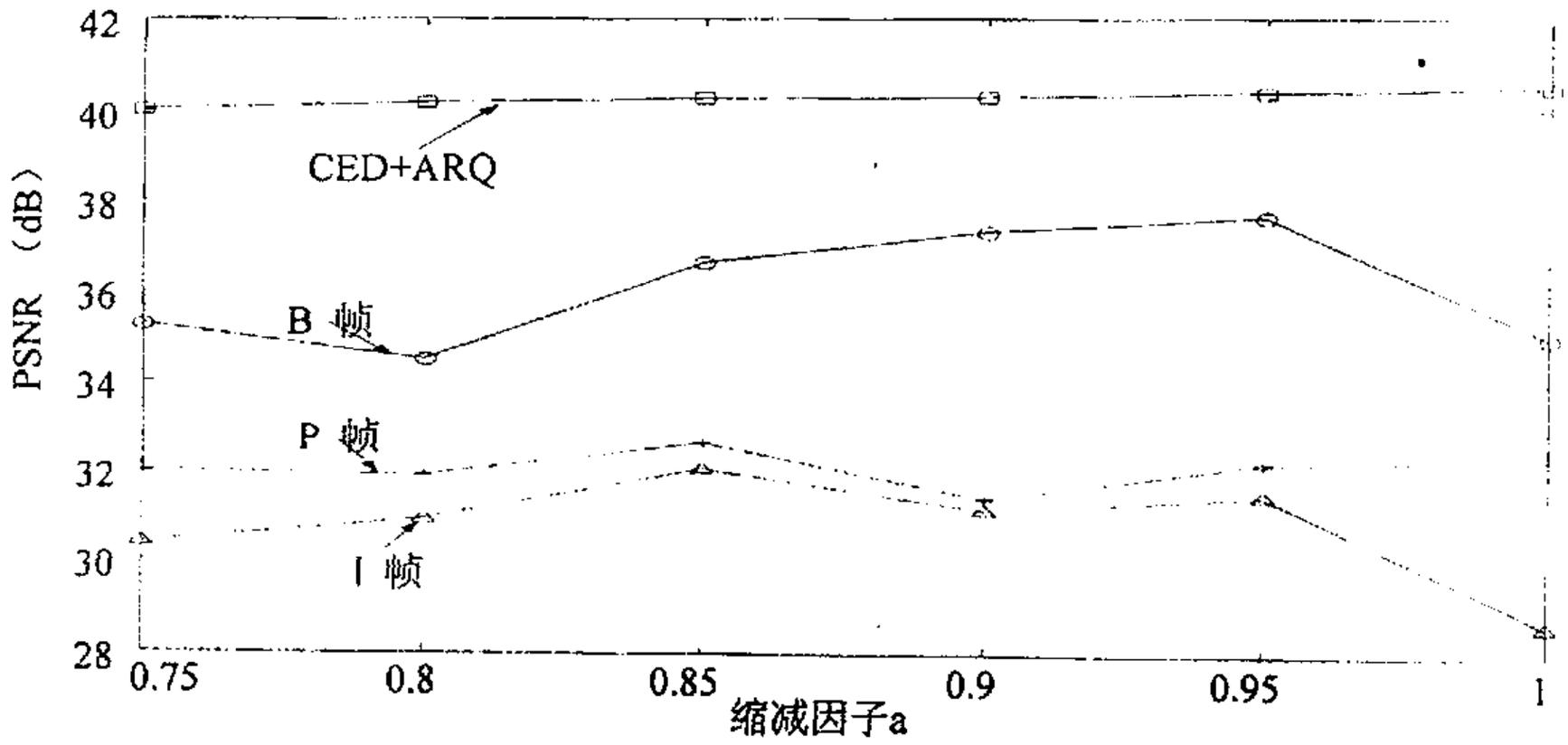

图 4.2 亮度信号 Y 的 PSNR 比较图

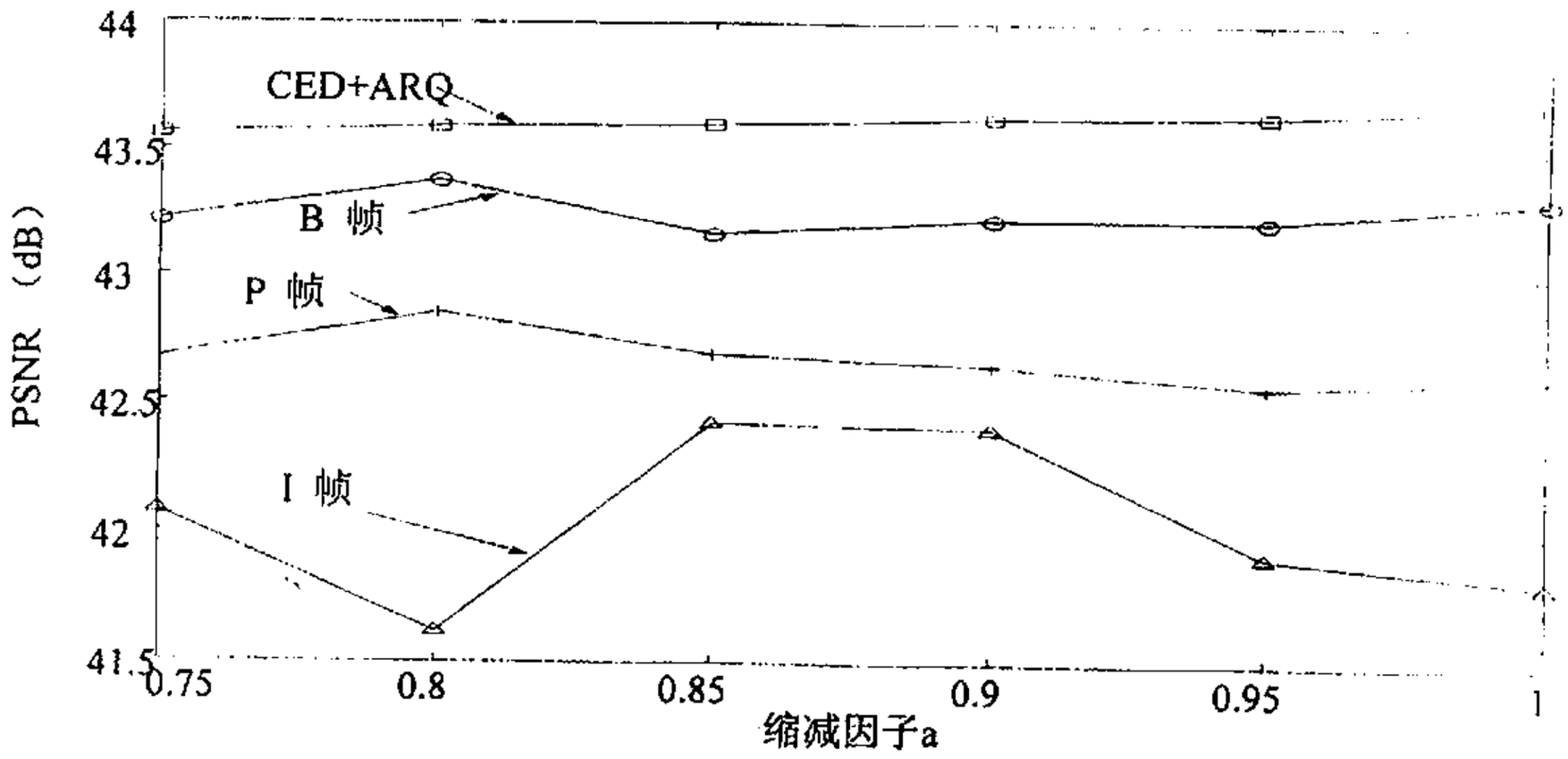

图 4.3 色度信号 U 的 PSNR 比较图

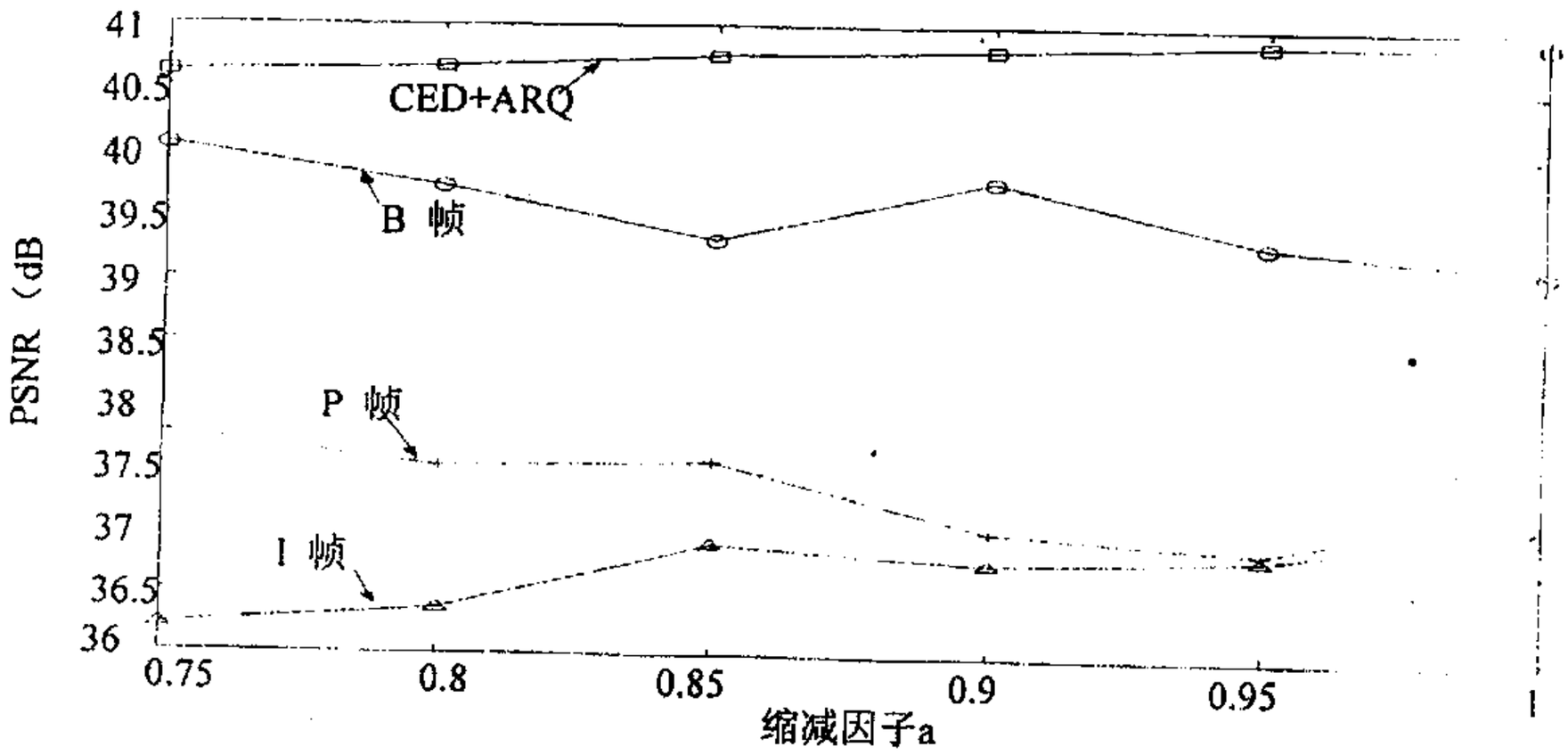

图 4.4 色度信号 V 的 PSNR 比较图

§4.4 本章小结

本章介绍了视频传输抗误码技术的发展，并结合 H.264 标准中编解码的特点对 H.264 的抗误码技术进行研究，通过实验证明在 H.264 中加入基于 AC 的 CED+ARQ 检错机制后能连续检错，且一个错误只牵制一小段数据，减少了需重发的数据量，大大减少检错时间，提高了数据抗误码能力，并能获得较好的压缩效果。

结束语

本文基于由 ITU-T 和 MPEG 联合组成的 JVT（Joint Video Expert Team）公布的视频压缩编码国际标准 H.264，研究了视频编解码的若干关键技术。包括：研究 H.264 编码器各模块的计算复杂度并对其中的关键模块整数变换模块进行研究改进；基于 H.264 的视频传输抗误码技术进行研究等。

具体工作：

1.对视频图像编码发展史、视频图像压缩编码技术及视频编码国际标准作了简单介绍。给出了图像视频编码领域的基本概念、发展历史和当前状况，着重介绍了视频编码国际标准 H.264，及其目前相关研究动态。阐明了本文选题的理论依据，并为随后展开深入的研究提供了线索与思路。

2.介绍 H.264 国际标准的发展以及 H.264 视频编解码演示系统的实现并对编码器各主要模块的复杂度进行了分析。本文以个人计算机为应用平台，利用纯软件进行视频编解码，实现了一个从图像源处理、编码、传输、解码及显示全过程的实验演示系统。

3.对 H.264 编码模块中占运算量 17%左右的整数变换和量化模块进行研究及改进。

4.研究了基于 H.264 的视频传输抗误码技术，在 H.264 中引入了一种基于算术编码的具有连续检错和自动重传相结合的综合抗误码方案后实现了连续检错，提高了数据抗误码能力同时能获得较好的压缩效果。

本文创新点：

1.在基于提升结构的无乘法二进制 DCT（binDCT）的基础上，先分析了如何考虑 binDCT 各指标之间的折衷，并研究了如何取用不同的 binDCT 系数以满足不同指标的需求。然后对参数进行选择优化，选择 binDCT 参数使 binDCT 既能避免浮点运算又能降低运算量，且更有利于压缩。最后提出“动态 binDCT 系数选择”的设想，针对实际中对 I 帧、P 帧、B 帧的质量要求不同，通过在传输比特流中附加少量的系数比特，以达到总体上更好的编码效果、更低的计算量。

2.针对 H.264 中算术编码的特点，引入了一种基于算术编码的具有连续检错能力和自动重传相结合的综合抗误码方案。该方法可以大大的提高了 H.264 在视频传输时的抗误码能力并在增加冗余量和提高检错性能之间取得较好的折衷，能获得较好的压缩效果和抗信道误码能力。

展望：

面向无线网和因特网的视频压缩传输技术是当前的研究热点，新的视频压缩国际标准H.264 也在今年刚刚公布。本文研究了 H.264 编解码的若干关键技术，但目前的研究还处于初步阶段，仍有大量的工作可做。概括为以下几点：

1.视频信息的数据量非常大，要在有限的带宽上传输日益增长的视频信息，对视频信息的高效压缩仍然显得十分重要。突破信息论的框架，根据人眼视觉特性的新一代视频压缩编码方法的发展不可限量。

2.视频通信要求很强的实时性，因此要求较高的编码速度，H.264 的编码速度还有待提高。

3.抗误码技术是视频传输的关键技术之一，也是视频传输技术得到进一步发展的前提，随着通信网络的发展，传输误码的表现形式也将发生变化，相应地要求采用不同的抗误码技术。本文对视频抗误码技术作了很初步的研究，仍然有大量工作可做。信源信道联合编码以及分层编码技术、不等错误保护技术等很具有发展潜力，值得做更为深入研究。比如：如何对视频信息进行分层，如何对不等保护的强度进行选择，如何对信源、信道编码的码率进行分配等多方面内容。

参考文献表

[1] BM Oliver, JR Pierce, and CE Shannon , " The Philosophy of PCM", Proc. IRE, vol 36, pp. 1324~1331, Nov. 1948

[2] Pratt W.K. and Andrews H.C. , Fourier transform coding of images, Proceedings of the Hawaii International Conference on System Sciences, Jan 1968

[3] PRATT WK, KANE, J., AND ANDREWS , HC Hadamard transform image coding, IEEE Proc. 57, 1 (Jan. 1969), 58~68

[4] 沈兰荪，卓力，田栋，汪孔桥，视频编码与低速率传输，电子工业出版社，2001：17~50

[5] 吴乐南，数据压缩的原理和应用，电子工业出版社，1995

[6] Chen W., Smith C.H.，and Fralick S. ，A Fast Computational Algorithm for the Discrete Cosine Transform，IEEE Trans Communication，1977，COM-25(9):1004~1009

[7] Lengwehasatit K. and Ortega A. ，DCT Computation Based on Variable Complexity Fast Approximations ，http://v.brl.ujuc.edu/ICIP98/

[8] Yu-Tai Chang and Chin-Liang Wang ，A New Fast DCT Algorithm and Its Systolic VLSI Implementation ，IEEE Trans Circuits and Systems II：Analog and Digital Signal Processing，1997, 44(11)：959~962

[9] 黄贤武等，数字图像处理与压缩编码技术，电子科技大学出版社，2000：449~456

[10] 姚志恒，基于H.263 的视频传输关键技术研究，北京：北京工业大学硕士学位论文，2002

[11] 钟玉琢等，基于对象的多媒体数据压缩编码国际标准，科学出版社，2000：14~36

[12] 钟玉琢等，MPEG-2 运动图像压缩编码国际标准及 MPEG 的新进展，清华大学出版社，2002：283~320

[13] 马小虎等，多媒体数据压缩标准及实现，清华大学出版社，1996

[14] H.26L Document Q.6/SG16-JVT-F100:Study of Final Committee Draft of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC) ，Dec. 2002

[15] H.26L Document Q.6/SG16-P-07: Draft ITU-T Recommendation H.264 (a.k.a. "H.26L") , May.2002

[16] H.26L Document Q.6/SG16: H.26L Test Model Long Term Number 8 (TML-8) Draft 0 , Jul.2001

[17] www.vcodex.com H.264/MPEG-4 Part10:Overview Oct.2002

[18] H.26L Document Q.6/SG16-M-16: A 16-bit architecture for H.26L, treating DCT transforms and quantization. Apr, 2001

[19] Chen. W, Harrison. C, and Fralick. S. A fast computational algorithm for the discrete cosine transform, IEEE Trans. Com., 1977, Vol. COM-25 (9): 1004~1011

[20] Wang. Z, Fast algorithm for the discrete W transform and for the discrete Fourier transform, IEEE Trans. ASSP, 1984, Vol. ASSP-32 (4): 803~816

[21] Daubechies. I and Sweldens.W, Factoring wavelet transforms into lifting steps, Journal of Fourier Anal. 1998, Vol. 4 (3): 247~269

[22] Jeon B., "Entropy Coding for H.26L", ITU-T Doc. Q15-J-57, May 2000

[23] Detlev Marpe , Adaptive Codes for H.26L , VCEG-L13 , Jan, 2001

[24] Boyd C., Cleary J., Irvine S., Rinsma-Melchert I., and Witten I., "Integrating error detection into arithmetic coding," IEEE Trans. Commun., vol. 45, pp. 1~3, Jan. 1997

[25] Chou J., Ramchandran K., "Arithmetic Coding-Based Continuous Error Detection for Efficient ARQ-Based Image Transmission", IEEE Joun. Commun., vol. 18, No.6, pp. 861~867, Jun. 2000

[26] Anand R., Ramchandran K., Kozintsev I. ,"Continuous Error Detection (CED) for Reliable Communication", IEEE Trans. Commun., vol. 49, No.9, pp. 1540~1549, Sep. 2001

[27] Kerofsky L., "Entropy Coding of Transform Coefficients", ITU-T Doc. Q15-K-45, August 2000

[28].Shannon C E., A mathematical theory of communication, Bell System Technical, 1948,27:379-423

[29] Norman A., Information theory and coding, New York:McGraw-Hill,1963

[30] Pascao. , Source coding algorithms for fast data compression[Ph D diss]. Stand University,USA,1976

[31] Langdon Jr G. , A introduction to arithmetic coding, IBM Journal of Research an Development, 1984,28(2):135~149

[32] Witten et al, "Arithmetic Coding for Data Compression", Comm. of the ACM, 30 (6), 1987, pp.520~541

[33] 徐林玲，林嘉宇，H.26L 中整数变换的研究及改进，通信技术，总第 142 期，2003，10:5~7

攻读硕士学位期间完成的学术论文

1. 一种抗频率选择性衰落的高速单载波系统，国防科技大学第二届研究生学术活动节论文集，2002：650~654，第一作者。
2. H.26L 中整数变换的研究及改进，通信技术，总第 142 期，2003，10：5~7，第一作者。
3. 基于 H.26L 的视频传输抗误码技术的研究，通信技术，已录用，第一作者。

致 谢

写到致谢，论文工作也就临近尾声了，这意味着我在国防科大的硕士研究生学习阶段的即将结束。回首在科大学习生活的这段时光，我从一个地方大学的本科生到一个军校的硕士生，从一个普通的年青人到一名军人，这短短的两年半我经历了人生中的一个重要转折。其间，我也经历了很多以前不曾体验的事，失去了一些东西，但也得到了一些东西，学会了很多东西。在科大我认识了很多良师益友，他们给了我无私的关心和帮助。在此，我要郑重地向他们表示感谢。

首先，我要感谢师长！很庆幸林嘉宇博士成为我的硕士课题指导老师，本论文的工作自始至终都是在他的亲切耐心指导下进行的。为了我的课题的顺利完成，林老师倾注了大量的心血。两年多来，林老师不仅在学术上以其深邃的理论造诣和严谨的科研作风为我“传道、授业、解惑”，而且在日常生活上也无微不至地关心着我。还要感谢李飚、雷菁、郑林华、熊辉、文贡坚等老师，他们对我的论文提出了不少中肯的建议，让我受益匪浅。在这里我要真心感谢老师！

感谢我的同学和朋友们，在科大两年半的时间里，我结识了很多可爱的同学和朋友，他们在学习生活上给了我很多的帮助。特别感谢刘长军、李莹莹、王学梅、马慧萍、张成、周媛媛等同学。毕业在即，祝大家一切顺利！

还要感谢我的家人，家是最好的避风港。有了家，我什么也不怕！

最后，我要感谢吴浩，一切尽在不言中！

向所有帮助过我的人表示感谢，有了你们的帮助我才能少走很多弯路，衷心感谢你们！

基于H.264的视频压缩关键技术的研究

作者：徐林玲
学位授予单位：国防科学技术大学
被引用次数：5次

引证文献(5条)

1. 李磊. 黄登山 基于H.264/AVC的自适应宏块编码算法研究[期刊论文]-信息安全与通信保密 2007(10)

2. 戴虎 基于H.264/AVC的码率控制算法的研究[学位论文]硕士 2006

3. 袁春 H.264/AVC中多参考帧下的失真度估算算法[学位论文]硕士 2005

4. 曾勇 基于H.264/AVC的率失真优化和码率控制算法研究[学位论文]硕士 2005

5. 朱靖丽 H.264视频编码算法研究及在DSP上的实现[学位论文]硕士 2005