

上海交通大学

硕士学位论文

H. 264视频编码算法研究

姓名：王涛

申请学位级别：硕士

专业：信号与信息处理

指导教师：余松煜

2003. 1. 1

H.264 视频编码算法研究

摘要

JVT/H.264 是 ITU-T VCEG 组织和 ISO/IEC MPEG 组织共同研究的新型视频压缩标准,是目前图像通信研究领域的热点问题之一。作为多媒体通信网络的重要组成部分, H.264 不再把编码和网络区别对待, H.264 综合考虑了编码效率和网络传输, 特别适合流媒体的传输, 因此它的研究和实现具有相当重要的意义。本文的研究工作正是在充分理解这个编码方案的背景、难点和重点的基础上展开的, 研究了方案中的和以往的编码方案不同的地方和新的技术应用及其相关的一些问题并提出自己的改进和措施相关的试验数据。

具体说, 本论文包含以下内容:

首先简单介绍了目前网络 and 多媒体通讯的现状和由此带来的挑战。接着介绍了在 TCP/IP 的网络环境中要达到广播级的视频播放所面临的问题和应该解决的技术。接下来回顾了视频压缩技术的背景知识, 扼要评述了国内外视频/图象编码的研究状况和相关的国际标准与建议, 分析了视频编码的现状和存在的问题; 针对这些问题, 指出了在 H.264 中相应的解决办法, 对 H.264 编码方案作了一个整体的说明。

建立在前面章节分析的基础上, 论文对 H.264 帧内编码的预测技术进行了详细的分析, 对比了 H.264、H.263、JPEG-2000 的帧内编码效率。得出了 H.264 的帧内编码效率在相同压缩比下, 它的 PSNR 性能要比 H.263 高出 5—10dB, 和 JPEG-2000 相当的结论。在此基础上, 分析了目前 H.264 的帧内预测技术不适合并行处理以及复杂度太高的缺点, 因此提出对预测块预选像素值和色度预测技术的两点改进方案, 试验结果表明改进后的方案在以极小的信噪比损失为代价的条件下, 在计算复杂度和并行处理上都有大的性能改善。

接下来详细分析了整数变换、基于内容的自适应可变长度编码技术 (CAVLC) 和结合宏块大小的自适应块变换 (ABT) 技术以及随机接入帧 SP 技术, 对整数变换、ABT 的数学基础、CAVLC 的技术创新点进行了研究和验证, 并提出改进的方向; 对 SP 帧取代 I 帧的原理和详细框图进行详细说明, 通过图表和数据对比了 I 帧、P 帧和 SP 帧的编码效率。

最后, 文章通过试验比较了 H.264 和 MPEG-4 对于标清序列的编码效率, 并引用了 JVT 关于 H.264 和 MPEG-2 对高清序列的编码效率比较。得出 H.264 不仅适用于低码率, 而且对高码率也有出色性能的结论。

关键词: H.264, 视频编码, CAVLC, ABT, 整数变换, 随机接入

VIDEO CODING ALGORITHM IN H.264

ABSTRACT

JVT/H.264 is the current project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Motion Picture Experts Group (MPEG). As one important component of multi-media communication network, H.264 design covers a Video Coding Layer (VCL), which provides the core high-compression representation of the video picture content, and a Network Adaptation Layer (NAL), which packages that representation for delivery over a particular type of network. This dissertation is just concerned with some difficult, important and new techniques of H.264 and proposed improved methods based on our analyse.

More specifically, this dissertation includes following contents:

Firstly, retrospect of some background knowledge on video compression technology is presented. After several detail, contraposed explains of the problems which curerent compressing algorithm can not overcome are illuminated, a brief survey of H.264 is then given.

Secondly, based on the analysis in the previous chapters, we just concentrated on Intra prediction technique. After much research on the characteristics of H.264 virified mode, we proposed two improved methods. First one is aimed at Pipeline processing of intra prediction. By setting the proper arrangement of submcaroblock and decude the number of pixels for prediction, we get satisfied experimental results with high processing speed. Second one is the chrome prediction method. We proposed a new plane prediction which can significant deduce the complexity with slight PSNR drops.

Further discussion is focused on the integral transforms, context-adaptive VLC coding (CAVLC), Adaptive Block Transforms (ABT) and a new picture type, namely SP-picture. Moreover, detailed explanation and block diagram of the substitution of SP frame for I frame are proposed. Coding efficiency of I frame, P frame and SP frame is compared by means of table and chart.

Key words: H.264, Video coding, CAVLC, ABT, integral transforms, random access

附件四

上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：



日期：2003年 1月17日

附件五

上海交通大学
学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本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和汇编本学位论文。

保密□，在___年解密后适用本授权书。

本学位论文属于

不保密☒。

(请在以上方框内打“√”)

学位论文作者签名：

王涛

指导教师签名：

金松堤

日期：2005年12月17日

日期：2005年12月17日

第一章 序论

1.1 目前的现状

近些年来，数字视频通信产品正迅速的进入办公室和普通家庭中。其覆盖领域已触及到通信、多媒体、广播电视和计算机等各个角落，其应用范围渗透到从低端到高端不同的视频通讯应用。随着网络技术的发展以及网络环境的改善，最终用户接入网络的带宽较以前有了巨大的改善，宽带网络技术正逐渐成为主流以及有潜力成为未来主流的接入技术，它包括 xDSL、混合光纤同轴（HFC）、光纤到户（FTTH）、交互式数字视频系统（SDV）等。另一方面随着 3G 技术的成熟和无线通讯技术的发展，交互的个人视频通讯和高质量的视频广播给当前的视频压缩技术带来了新的挑战，以往的视频压缩技术已经不能适应网络环境的改变或是需要改进。总之，新型的数字视频应用需要新一代的视频压缩标准^[1]。但是几个不变的准则为：

- 1、可靠链接的用户接收到的视频质量要好
- 2、计算能力强的用户接收到的视频质量要好
- 3、在编码端，高性能的计算机要有高的编码效率

在视频压缩算法的研究中，国际标准化组织的运动图像专家组(MPEG) 和国际电信联盟ITU-T 的视频编码专家组(VCEG)的贡献颇多。他们分别提出了MPEG-1、MPEG-2、MPEG-4和H.261、H.263、H.263++的视频编码标准。

但是这些算法都有很强的针对性，如 MPEG-1 是针对视频存储的（VCD）；MPEG-2 虽称为通用标准，但主要用于广播；MPEG-4 的应用面广，但还有需要完善和改进的地方；H.26X 系列主要针对实时的视频会议。它们都有自己的一些特殊的限制条件，如：H.26X 需要对等的编解码环境、低时延、低等待时间；而 MPEG-2 视频广播是不需要这些限制的。

前面所说，随着网络的发展，无论是 Internet 网还是无线网络，都迫切需要一种新型的压缩算法，它不仅要有高的压缩比；而且可以根据不同的网络条件提供不同质量的视频服务，可以应用在低带宽的信道，也可以应用在高带宽的信道；同时还应该有一定的网络自

第一章 序论

1.1 目前的现状

近些年来，数字视频通信产品正迅速的进入办公室和普通家庭中。其覆盖领域已触及到通信、多媒体、广播电视和计算机等各个角落，其应用范围渗透到从低端到高端不同的视频通讯应用。随着网络技术的发展以及网络环境的改善，最终用户接入网络的带宽较以前有了巨大的改善，宽带网络技术正逐渐成为主流以及有潜力成为未来主流的接入技术，它包括 xDSL、混合光纤同轴（HFC）、光纤到户（FTTH）、交互式数字视频系统（SDV）等。另一方面随着 3G 技术的成熟和无线通讯技术的发展，交互的个人视频通讯和高质量的视频广播给当前的视频压缩技术带来了新的挑战，以往的视频压缩技术已经不能适应网络环境的改变或是需要改进。总之，新型的数字视频应用需要新一代的视频压缩标准^[1]。但是几个不变的准则为：

- 1、可靠链接的用户接收到的视频质量要好
- 2、计算能力强的用户接收到的视频质量要好
- 3、在编码端，高性能的计算机要有高的编码效率

在视频压缩算法的研究中，国际标准化组织的运动图像专家组(MPEG) 和国际电信联盟ITU-T 的视频编码专家组(VCEG)的贡献颇多。他们分别提出了MPEG-1、MPEG-2、MPEG-4和H.261、H.263、H.263++的视频编码标准。

但是这些算法都有很强的针对性，如 MPEG-1 是针对视频存储的（VCD）；MPEG-2 虽称为通用标准，但主要用于广播；MPEG-4 的应用面广，但还有需要完善和改进的地方；H.26X 系列主要针对实时的视频会议。它们都有自己的一些特殊的限制条件，如：H.26X 需要对等的编解码环境、低时延、低等待时间；而 MPEG-2 视频广播是不需要这些限制的。

前面所说，随着网络的发展，无论是 Internet 网还是无线网络，都迫切需要一种新型的压缩算法，它不仅要有高的压缩比；而且可以根据不同的网络条件提供不同质量的视频服务，可以应用在低带宽的信道，也可以应用在高带宽的信道；同时还应该有一定的网络自

适应功能。所以，VCEG 在完成 H.263 之后，致力于一个短期的目标 H.263++，和一个长期的目标 H.26L—这就是 H.264 的前身。后来，MPEG 组织和 VCEG 组织联合，组成了 JVT（Joint Video Coding Experts Group）联合视频专家组，共同研究新的编码方案，分别命名为 MPEG-4 part 10 “Advanced Video Coding” (AVC)和 H.264^[2]，它的应用范围极其广泛，图 1-1 列出了它的一些应用。

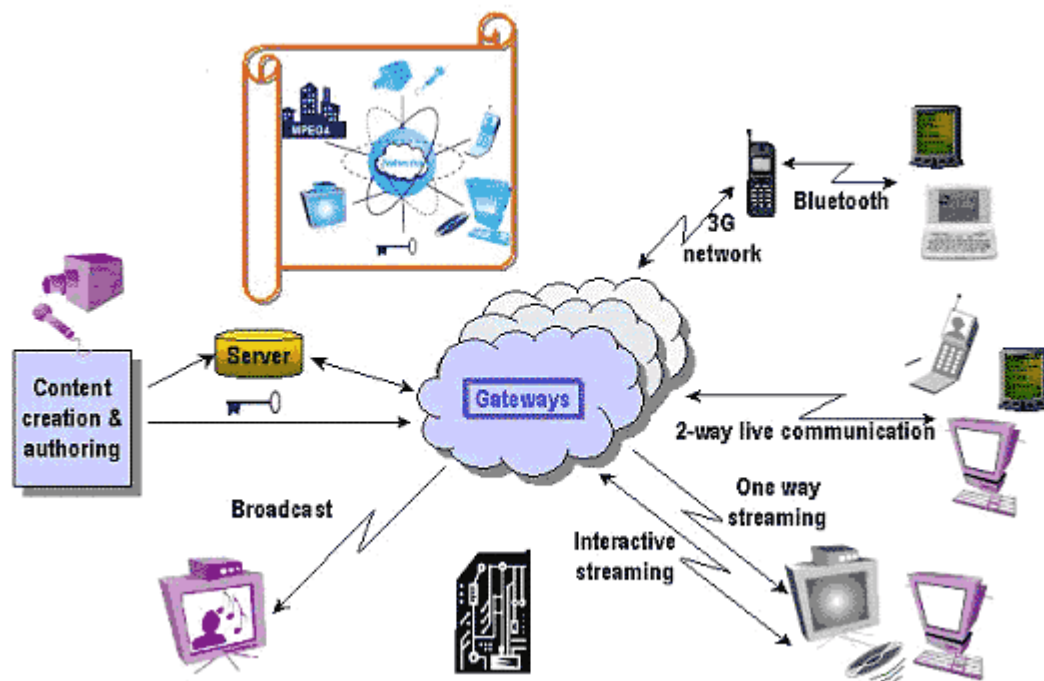


图 1-1 新的视频编码方案应用的范围

Fig.1.1 The domain of new project

1.2 新方案的目标和挑战

任何新的方案都有它的需要解决的目标和环境、面临的困难和挑战，H.264 也不例外，下面两节的内容将对此作一个表述。

1.2.1 需达到的目标

在网络环境中达到广播级的视频服务是这个新的视频编码的重要目标之一。我们知道网络环境千差万别，有的带宽可能很高，但有的却很低，并且终端用户的具体情况也各不相同，有的网络条件好、设备也好，有的却可能很差。新的编码方案考虑到终端用户情况的不同，使用了专用的视频服务器来传输视频流。视频服务器监视终端用户的链接情况、处理能力、节目选择，并且能在此基础上进行细微的调整。但是同时，基于服务器的方案

适应功能。所以，VCEG 在完成 H.263 之后，致力于一个短期的目标 H.263++，和一个长期的目标 H.26L—这就是 H.264 的前身。后来，MPEG 组织和 VCEG 组织联合，组成了 JVT（Joint Video Coding Experts Group）联合视频专家组，共同研究新的编码方案，分别命名为 MPEG-4 part 10 “Advanced Video Coding” (AVC)和 H.264^[2]，它的应用范围极其广泛，图 1-1 列出了它的一些应用。

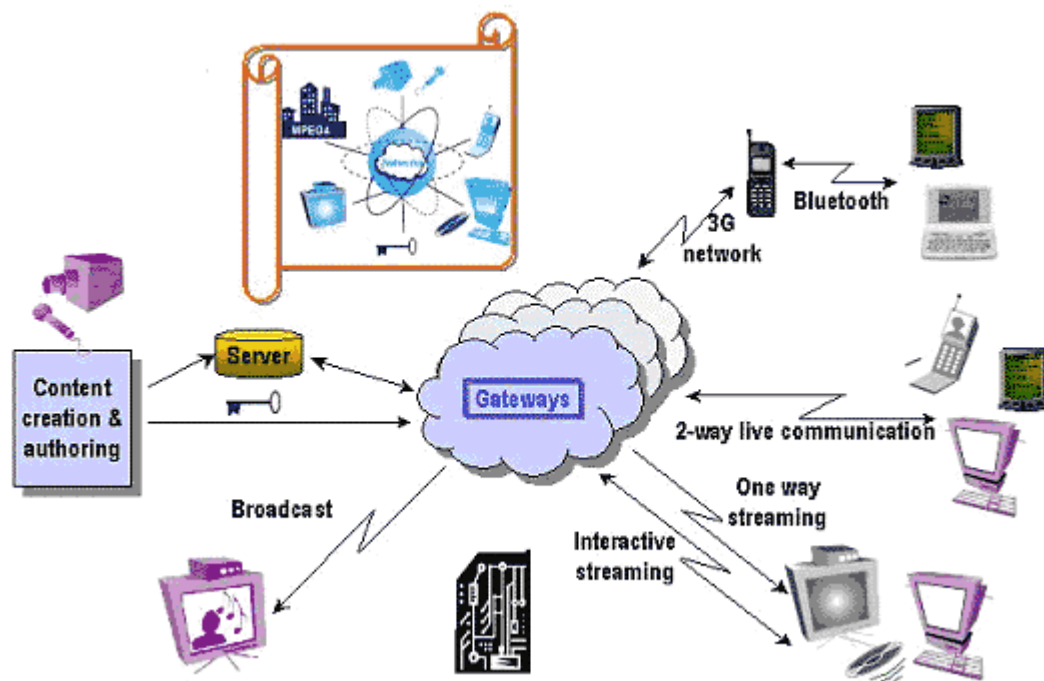


图 1-1 新的视频编码方案应用的范围

Fig.1.1 The domain of new project

1.2 新方案的目标和挑战

任何新的方案都有它的需要解决的目标和环境、面临的困难和挑战，H.264 也不例外，下面两节的内容将对此作一个表述。

1.2.1 需达到的目标

在网络环境中达到广播级的视频服务是这个新的视频编码的重要目标之一。我们知道网络环境千差万别，有的带宽可能很高，但有的却很低，并且终端用户的具体情况也各不相同，有的网络条件好、设备也好，有的却可能很差。新的编码方案考虑到终端用户情况的不同，使用了专用的视频服务器来传输视频流。视频服务器监视终端用户的链接情况、处理能力、节目选择，并且能在此基础上进行细微的调整。但是同时，基于服务器的方案

适应功能。所以，VCEG 在完成 H.263 之后，致力于一个短期的目标 H.263++，和一个长期的目标 H.26L—这就是 H.264 的前身。后来，MPEG 组织和 VCEG 组织联合，组成了 JVT（Joint Video Coding Experts Group）联合视频专家组，共同研究新的编码方案，分别命名为 MPEG-4 part 10 “Advanced Video Coding” (AVC) 和 H.264^[2]，它的应用范围极其广泛，图 1-1 列出了它的一些应用。

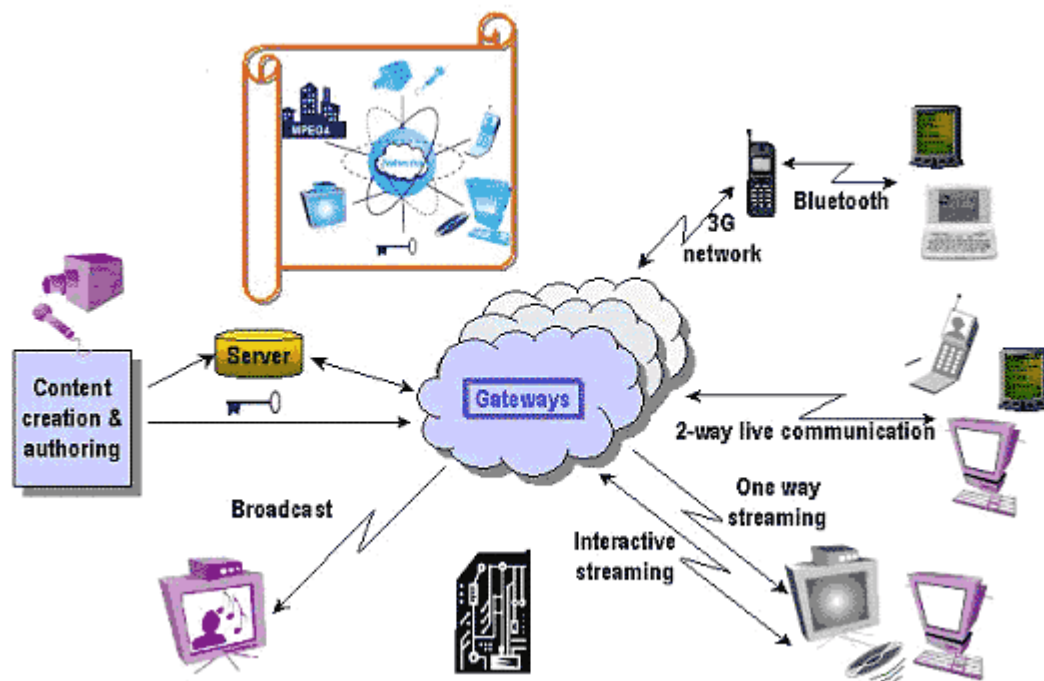


图 1-1 新的视频编码方案应用的范围

Fig.1.1 The domain of new project

1.2 新方案的目标和挑战

任何新的方案都有它的需要解决的目标和环境、面临的困难和挑战，H.264 也不例外，下面两节的内容将对此作一个表述。

1.2.1 需达到的目标

在网络环境中达到广播级的视频服务是这个新的视频编码的重要目标之一。我们知道网络环境千差万别，有的带宽可能很高，但有的却很低，并且终端用户的具体情况也各不相同，有的网络条件好、设备也好，有的却可能很差。新的编码方案考虑到终端用户情况的不同，使用了专用的视频服务器来传输视频流。视频服务器监视终端用户的链接情况、处理能力、节目选择，并且能在此基础上进行细微的调整。但是同时，基于服务器的方案

为视频压缩算法带来一个新的挑战，码流切换。

工业级的数字视频广播一大需求是通过 IP 网络来进行，因此基本的传输单元是包。为了得到最大的传输效率，码流应该尽可能的以包边界为限制(例如：一帧一包，或几帧一包)，极限值是 1500 字节。这里有两个问题：包可能迟到或根本就丢失了。网络的阻塞或路由器的缓存溢出是诱发这两个问题的原因。反映在解码端为跳帧现象（编码和解码端的重建帧不同）。在视频广播环境中，网络的低等待时间不是必须的，因此，使用大的、多重有效载荷包来减少 IP 包的头。因此，包丢失直接反映为帧丢失。

和视频会议环境不同，这种流媒体不要求对等的配置，编码、传输、解码的计算机性能可能差别很大。工业级的数字视频广播商会投资高性能的编码计算机和网络服务器，而终端用户的计算机性能差别很大。因此回放技术必须和终端用户的计算机性能结合起来。

图 1—2 图示了视频流在网络上的传输过程。编码计算机采集实时数据或从文件中读出数据，进行压缩、打包并传输到视频服务器。视频服务器负责把压缩好的数据传送到需求的用户并对用户的需求进行处理。

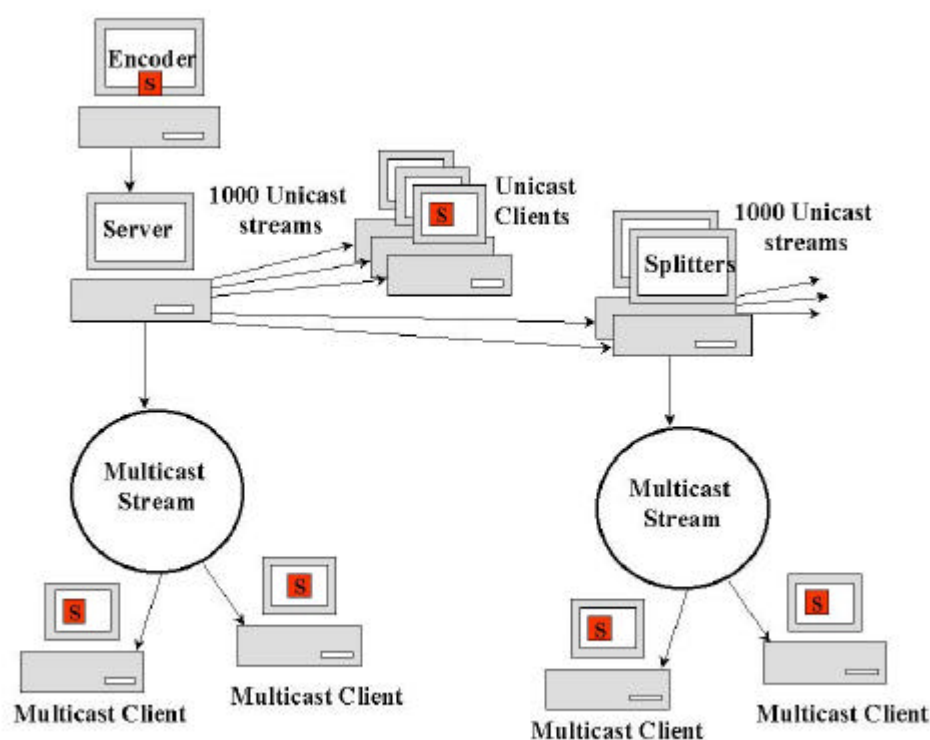


图 1—2 网络视频服务结构图

Fig.1.2 Broadcast Network Architecture

1.2.2 面临的问题和挑战

要达到在网络环境中能够进行广播级的视频服务目标，需要综合多方面因素，下面就

为视频压缩算法带来一个新的挑战，码流切换。

工业级的数字视频广播一大需求是通过 IP 网络来进行，因此基本的传输单元是包。为了得到最大的传输效率，码流应该尽可能的以包边界为限制(例如：一帧一包，或几帧一包)，极限值是 1500 字节。这里有两个问题：包可能迟到或根本就丢失了。网络的阻塞或路由器的缓存溢出是诱发这两个问题的原因。反映在解码端为跳帧现象（编码和解码端的重建帧不同）。在视频广播环境中，网络的低等待时间不是必须的，因此，使用大的、多重有效载荷包来减少 IP 包的头。因此，包丢失直接反映为帧丢失。

和视频会议环境不同，这种流媒体不要求对等的配置，编码、传输、解码的计算机性能可能差别很大。工业级的数字视频广播商会投资高性能的编码计算机和网络服务器，而终端用户的计算机性能差别很大。因此回放技术必须和终端用户的计算机性能结合起来。

图 1—2 图示了视频流在网络上的传输过程。编码计算机采集实时数据或从文件中读出数据，进行压缩、打包并传输到视频服务器。视频服务器负责把压缩好的数据传送到需求的用户并对用户的需求进行处理。

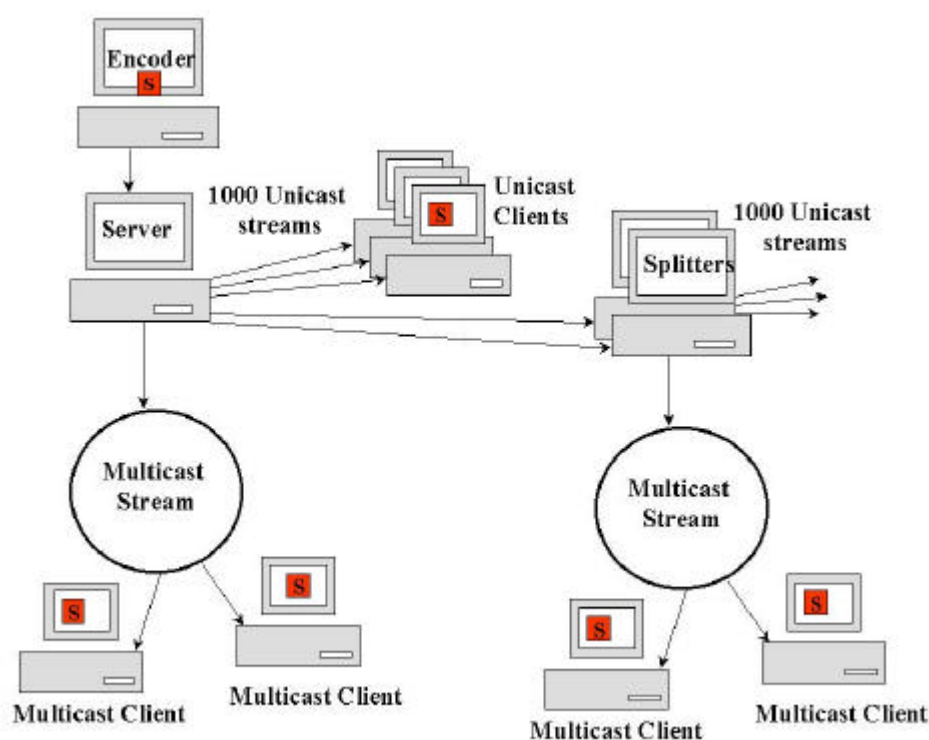


图 1—2 网络视频服务结构图

Fig.1.2 Broadcast Network Architecture

1.2.2 面临的问题和挑战

要达到在网络环境中能够进行广播级的视频服务目标，需要综合多方面因素，下面就

从三方面来讨论：（1）广播信道环境下编码效率；（2）网络服务器的平衡；（3）编码器的最小功能。这三个方面也即需要解决在复杂的网络环境中，需要达到什么样的观看效果；当网络条件发生变化时，如何去适应它；编码器应该提供什么样的功能和如何在功能和编码效率上达到一种平衡的问题。

和视频会议不同，网络广播服务由于放松了对等的编解码环境、低时延、低等待时间的限制，所以编码器性能可以有极大的提高。其中很重要的一点是在固定带宽的链接中要有连续的回放质量。体现在：（1）可适应的有限的网络时延（2）有限的算法等待时间（3）充足的解码处理。处理好这三个问题就可以达到连续稳定的播放要求。下面详细的说明问题的产生和解决办法。

1、网络引入时延

在单向的视频通讯中单纯的网络时延不是一个大的问题，这可以通过增加一定的等待时间来补偿，但是网络时延的波动性将对解码端的连续回放产生影响。时延的波动性可能是网络阻塞或是多包载荷产生的（I 或大的 P 帧）。终端用户的缓存可以减轻网络时延问题。网络 de-jitter 缓存可以减轻网络带宽波动，pre-roll 缓存可以减轻突发的数据峰值。编码器可以通过管理 pre-roll 缓存来使用不同的数据码率。这个后果是解码端有一个长的等待视频开始的时间。

2、算法引入的等待时间

在单向传播中，可以利用算法引入的等待时间来提高编码效率。长时间（多帧）参考技术可以利用已经获得的一组图像来获得更加精确的参考，也可以使用复杂的运动模型和 3D（2D+时间）编码技术。静止物体、场景变换、反复的运动都可以通过长时间的参考帧技术来获得一个很好的编码效率。但这里需要在终端用户增加一些额外的帧缓冲区。

3、非对称处理

因为编码器是独立于终端用户的，编码器可以更加有效的专注于编码，这种结构可以更好的适应复杂的编码算法。解码器性能的多样性要求解码的算法不能太复杂，在同样的码流条件下，性能高的解码器应该比性能低的解码器有更高的观看效果。

以上是从编、解码器的性能方面说明了需要解决的问题。我们知道，解码端通过反馈信道发送传送请求和本身的一些信息到视频服务器，视频服务器根据接受到的信息传送编好的码流给解码器，然后通过反馈信道监视解码器的状态作相应的调整。下面从图 1-2 的结构上分析编码器、视频服务器分开这种结构需要解决的问题。

1、错误恢复和纠错

- a、在网络的传输中，发生码字传错是难免的。在错误严重时，服务器应该能够对解码端的重传请求做出反应。这种策略需要在解码器端有一个大的重传缓存来保证在发出请求后和重传丢失的帧这段时间所需要正常播放的数据。
- b、数据包的重传并不是一个很明智的办法，所以在错误不是很严重时，视频服务器可以使用适量的错误纠正码（ECC）来补偿出错或丢失的数据。

2、视频流切换技术

在网络视频广播中，用户的网络环境可能会发生波动，好的视频流切换技术应该能够让视频服务器随着每一个解码器的参数的改变而传送不同码率的码流。但是视频切换技术面临着一些难题。通过 I 帧切换增加了处理的时间，而且会使码流变得很大，降低编码效率。通过 P 帧切换又会由于编码、解码端的重建帧不相同而产生图像的失真。并且码率控制算法还要和所有不同的码流相匹配，否则在码流切换时可能导致 VBV 缓存的溢出。但是总的说来，视频切换技术可以归纳为带宽可升级和处理可升级两种方案。

a、带宽可升级

分层可升级技术对于带宽可升级不是一个好的解决办法，因为编码效率是和层相联系的^[3]。在低带宽时，由于图像质量不好，有时达到人眼不能忍受地步，这种分层技术的缺点就特别明显。一种可能的解决办法是让编码器生成多个不同码率的视频流。视频服务器根据终端用户的具体情况传送相应的码流。编码器可以通过在编不同码率的码流时共享编码信息来减少编码的复杂度。

b、处理可升级

编码器根据解码器性能的高低生成不同码率的码流。视频服务器动态检测终端用户的处理能力并切换到相应的码流。

c、用户选择

终端用户通过反馈信道把它的参数送给视频服务器让它决定传送具体的码流。

3、随机接入

随机接入是实时直播的一项基本要求，用户应该能够在直播的过程中随时接入。用户点播要求有快进、倒带、搜索功能，用户的要求是通过一个低带宽的反馈信道传送到服务器的。I 帧是一个可行的、但是效率很低的随机接入方案，所以需要一种新的图像类型。

1.3 本文的主要内容和安排

H.264 编码方案是目前图像通信研究领域的一个热点问题。作为多媒体通信网络的重要组成部分，新型的编码方案不再把编码和网络区别对待，H.264 综合考虑了编码效率和网络传输，它的研究和实现具有相当重要的意义。本文的研究工作正是在充分理解这个编码方案的背景、难点、改进点的基础上展开的，研究了方案中的和以往的编码方案不同的地方和新的技术应用及其相关的一些问题。

全文内容安排如下：

第一章 首先简单介绍了目前网络和多媒体通讯的现状和由此带来的挑战。接着介绍了在 TCP/IP 的网络环境中要达到广播级的视频播放所面临的问题和应该解决的技术。最后指出新型的 H.264 编码方案是目前最有前途的解决方案也是当前图像通讯领域中研究的热点问题之一。

第二章 本章开头以一小节介绍了当前实际应用中的视频压缩算法的基本原理并给出一个基本的编解码结构框图。由此引入了 H.264 方案较以往的编码技术改进的地方并作一个整体的介绍。

第三章 详细地介绍了 Intra 帧的帧内预测技术并给出和 H.263 的帧内编码在图像质量上的信噪比和编码比特数的比较结果，指出 H.264 的帧内编码效果在比特数相当的条件下达到了 JPEG-2000 的效果。然后给出对帧内编码的两点改进方案并给出了试验结果，在信噪比基本不变的情况下，极大降低了计算复杂度并且可以进行并行处理。

第四章 介绍了新型的结合量化的整数变换和基于内容的可变长编码技术。详细的说明了它们的基本原理和实现的过程。并对 H.264 新引进的结合宏块大小的自适应块变换的原理和特点进行分析。

第五章 本章中对 H.264 中的随机接入帧 SP 帧的概念和功能进行了介绍和说明，分别介绍了两种编解码的框图和具体的过程，并对其适用的范围进行了说明；最后，列举了一些图表和数据对比了 I 帧、P 帧和 SP 帧的编码效率。

最后，总结了本文的工作，指出了进一步研究的方向并比较了 H.264、MPEG-4 和 MPEG-2 对标清和高清序列的编码效率。

本章参考:

- [1] Dr. Gary S. Greenbaum, “Remarks on the H.26L Project: Streaming Video Requirements for Next Generation Video Compression Standards”, doc Q15-G-11, ITU-T video coding experts group meeting, Monterey, Feb 1999.
- [2] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, “Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC) – Joint Committee Draft”, document JVT-E022d3.doc, Sep. 2002.
- [3] L. Yang, F.C.M. Martins, T.R. Gardos, “Improving H.263+ Scalability Performance for Very Low Bit Rate Applications,” Proceedings of the 1999 Video Conferencing Over IP, San Jose, January, 1999.

第二章 H.264 总体概述

2.1 视频压缩算法基本原理

视频压缩的出发点是减少视频或图像的冗余度。冗余度又分为多种, 其中最主要的是空间冗余度、时间的冗余度、编码冗余度。空间冗余是指在一幅图像中空间相邻的像素和区域之间的相似性, 例如: 在相邻的两个像素它们的数值是很相近的。时间冗余是指在相邻的一系列图像中各个图像之间有一定的相似性。编码冗余度是指从概率统计角度对常用的符号赋予短的码字。

2.1.1 压缩空间冗余度—INTRA 帧编码

目前的编码方案都是把图像分块处理的。把每一个块经过映射变换可以把像素之间的相关性变低。经过映射变换后的块含有和原始块相同的信息, 变换后的系数反映了不同频率上的信息。映射变换本身不产生任何压缩, 压缩是在量化变换后的系数上取得的, 量化系数越大, 压缩的效果越好, 当然失真也越大。由于人眼对高频不敏感, 所以可以对高频分量用大的数值来量化, 这样在频域量化会比直接对原始像素量化性能要好。在解码端, 经过映射变换的反变换后, 可以得到原始图像的重建图像。这种压缩技术称为INTRA帧编码。

常用的映射变换是discrete cosine transform (DCT) 变换, 因为它性能高并且易于实现。另一种是JPEG-2000^[1]中使用的小波变换。

2.1.2 压缩时间冗余度—INTER 帧编码

在一系列的图像中, 相邻的两幅图像之间有很强的相似性, 如果只对每一幅图像作帧内压缩 (Motion-JPEG), 显然它的压缩是低效的。如果把当前的图像和前面图像的差值编码, 显然要高效的多。差值可以用当前图像和以前图像相减获得, 下一步就是使用INTRA帧的技术来进一步压缩空间冗余度。如果视觉在某些差值上可以容忍, 就可以使用以前的像素值来代替当前的像素值, 这样可以进一步压缩。这种压缩技术称为INTER帧编码。

2.1.3 运动估计和运动补偿

运动补偿可以把INTER帧的压缩性能进一步的提高。常见的运动补偿是使用相邻帧适当位置的图像块来预测和匹配当前的图像块。目前的图像压缩标准通常是把一幅图像分为

第二章 H.264 总体概述

2.1 视频压缩算法基本原理

视频压缩的出发点是减少视频或图像的冗余度。冗余度又分为多种, 其中最主要的是空间冗余度、时间的冗余度、编码冗余度。空间冗余是指在一幅图像中空间相邻的像素和区域之间的相似性, 例如: 在相邻的两个像素它们的数值是很相近的。时间冗余是指在相邻的一系列图像中各个图像之间有一定的相似性。编码冗余度是指从概率统计角度对常用的符号赋予短的码字。

2.1.1 压缩空间冗余度—INTRA 帧编码

目前的编码方案都是把图像分块处理的。把每一个块经过映射变换可以把像素之间的相关性变低。经过映射变换后的块含有和原始块相同的信息, 变换后的系数反映了不同频率上的信息。映射变换本身不产生任何压缩, 压缩是在量化变换后的系数上取得的, 量化系数越大, 压缩的效果越好, 当然失真也越大。由于人眼对高频不敏感, 所以可以对高频分量用大的数值来量化, 这样在频域量化会比直接对原始像素量化性能要好。在解码端, 经过映射变换的反变换后, 可以得到原始图像的重建图像。这种压缩技术称为INTRA帧编码。

常用的映射变换是discrete cosine transform (DCT) 变换, 因为它性能高并且易于实现。另一种是JPEG-2000^[1]中使用的小波变换。

2.1.2 压缩时间冗余度—INTER 帧编码

在一系列的图像中, 相邻的两幅图像之间有很强的相似性, 如果只对每一幅图像作帧内压缩 (Motion-JPEG), 显然它的压缩是低效的。如果把当前的图像和前面图像的差值编码, 显然要高效的多。差值可以用当前图像和以前图像相减获得, 下一步就是使用INTRA帧的技术来进一步压缩空间冗余度。如果视觉在某些差值上可以容忍, 就可以使用以前的像素值来代替当前的像素值, 这样可以进一步压缩。这种压缩技术称为INTER帧编码。

2.1.3 运动估计和运动补偿

运动补偿可以把INTER帧的压缩性能进一步的提高。常见的运动补偿是使用相邻帧适当位置的图像块来预测和匹配当前的图像块。目前的图像压缩标准通常是把一幅图像分为

16X16像素的宏块。先在参考帧中根据一定的准则预测当前编码宏块中的每一个像素值，如果预测出没有运动矢量，说明当前宏块可以用参考帧中相应位置的宏块代替，如果有运动矢量，说明当前宏块可以用参考帧中的相应大小的块移动得到。图2—1说明了当前帧和参考帧的匹配过程。

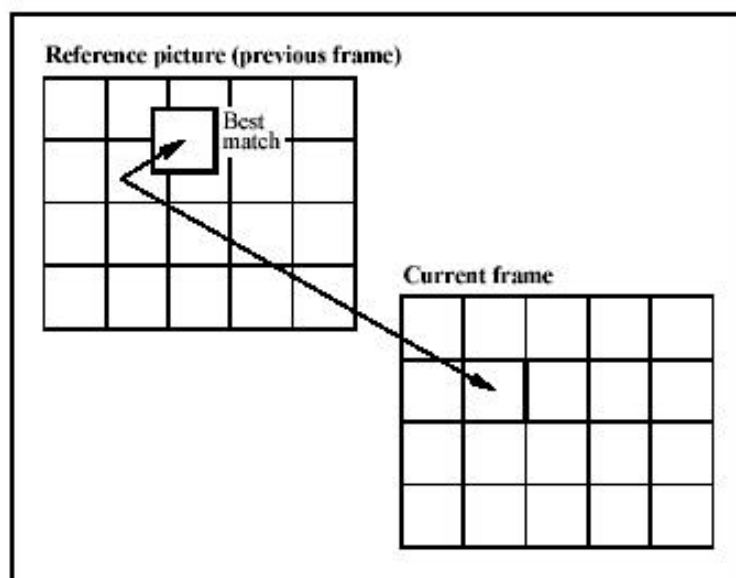


图2—1 当前帧和参考帧的匹配过程

Fig.2.1 Forward prediction using block matching

常用的匹配准则是最小均方误差（MSE）准则。当找到最佳的匹配块时，用运动矢量来描述这个位移，然后计算预测块和原始块之间的差值。解码器只要获得运动矢量和预测差值就可以解码。

2. 1. 4 熵编码

当帧经过INTRA或INTER编码后，原始的视频信号就变成了一系列的变换系数和运动矢量和辅助信息。从熵编码的角度来说，它们都是一些符号，因此可以用概率统计的方法对常用的符号建模，然后用另一种符号来表示它，这样可以进一步减少最终的码率。

游程码—如果相同的符号连续出现，用一个符号加上连续符号出现的次数，将大大的提高编码效率—这就是游程码基本思想。变换系数正好符合这个特点，所以用游程码可以极大的减小码率。并且，游程码可以很方便的结合别的熵编码方法。

变长码—可变长度编码（VLC）是目前各个编码算法中采用最多的方法。著名的霍夫曼编码就是一种变长码。VLC码的基本思想是根据概率统计对出现概率高的符号赋予短的码字，出现概率小的符号赋予长的码字。但是VLC编码有一个缺点就是它不能精确的描述符号出现的概率，因为VLC码都是整数码长。

16X16像素的宏块。先在参考帧中根据一定的准则预测当前编码宏块中的每一个像素值，如果预测出没有运动矢量，说明当前宏块可以用参考帧中相应位置的宏块代替，如果有运动矢量，说明当前宏块可以用参考帧中的相应大小的块移动得到。图2—1说明了当前帧和参考帧的匹配过程。

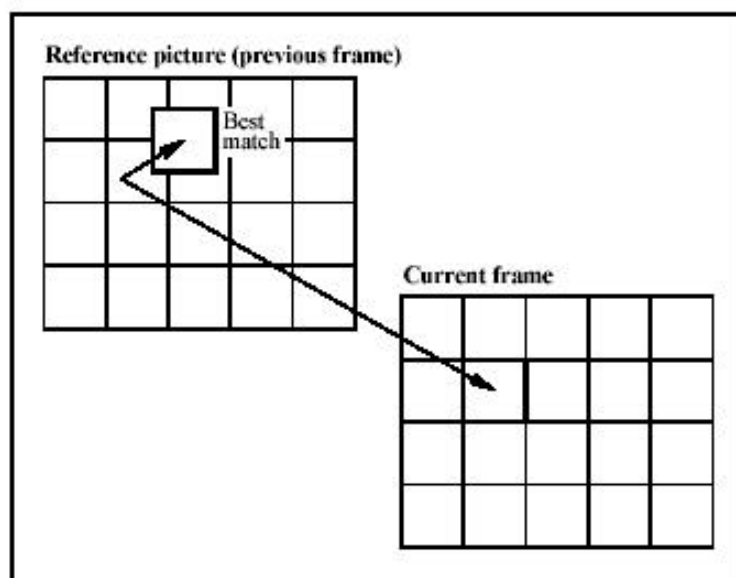


图2—1 当前帧和参考帧的匹配过程

Fig.2.1 Forward prediction using block matching

常用的匹配准则是最小均方误差（MSE）准则。当找到最佳的匹配块时，用运动矢量来描述这个位移，然后计算预测块和原始块之间的差值。解码器只要获得运动矢量和预测差值就可以解码。

2. 1. 4 熵编码

当帧经过INTRA或INTER编码后，原始的视频信号就变成了一系列的变换系数和运动矢量和辅助信息。从熵编码的角度来说，它们都是一些符号，因此可以用概率统计的方法对常用的符号建模，然后用另一种符号来表示它，这样可以进一步减少最终的码率。

游程码—如果相同的符号连续出现，用一个符号加上连续符号出现的次数，将大大的提高编码效率—这就是游程码基本思想。变换系数正好符合这个特点，所以用游程码可以极大的减小码率。并且，游程码可以很方便的结合别的熵编码方法。

变长码—可变长度编码（VLC）是目前各个编码算法中采用最多的方法。著名的霍夫曼编码就是一种变长码。VLC码的基本思想是根据概率统计对出现概率高的符号赋予短的码字，出现概率小的符号赋予长的码字。但是VLC编码有一个缺点就是它不能精确的描述符号出现的概率，因为VLC码都是整数码长。

算术码—算术码在图像数据压缩标准(如JPEG, JBIG)中扮演了重要的角色。在不知信源统计特性的情况下,只要监视一小段时间内码符号出现的频度,不管统计是平稳的还是非平稳的,编成的码率总能趋近于信源熵值。在算术码中,消息用0到1之间的实数进行编码,算术码用到两个基本的参数:符号的概率和它的编码间隔。信源符号的概率决定压缩编码的效率,也决定编码过程中信源符号的间隔,而这些间隔包含在0到1之间。编码过程中的间隔决定了符号压缩后的输出。算术码的优点就是能更精确的描述符号概率。

2.1.5 典型的编解码器结构

综合上面介绍的几种方法,就可以组合出一个典型的编解码器。图2-2a、图2-2b图示一个典型的DCT变换的编解码器,从图中可以看出,编码器端也有一个解码器,这样可以准确的进行运动补偿。有一点需要指出的是,反DCT变换后的系数应该精确匹配,这样差值才有意义,但是由于在计算机总会截去部分精度,难免会产生误差,因此不可能精确匹配,这样就会造成误差累积。

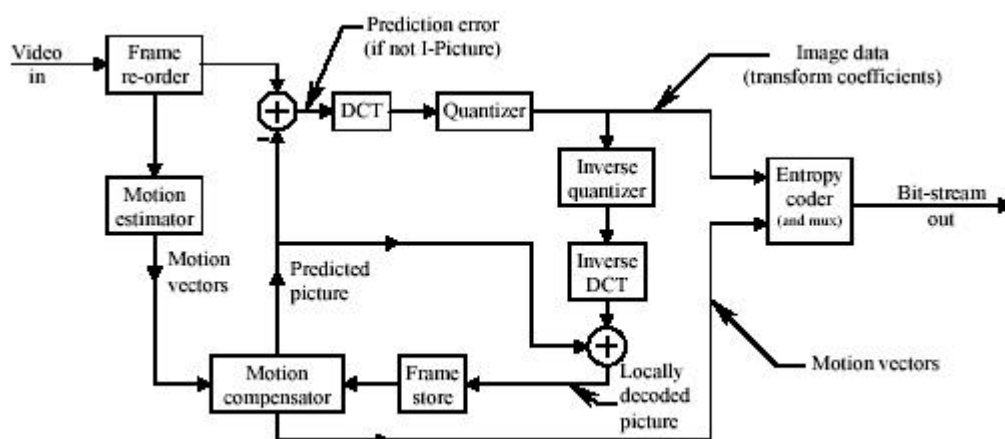


图2-2a 典型的编码器结构

Fig.2.2a Typical video coder

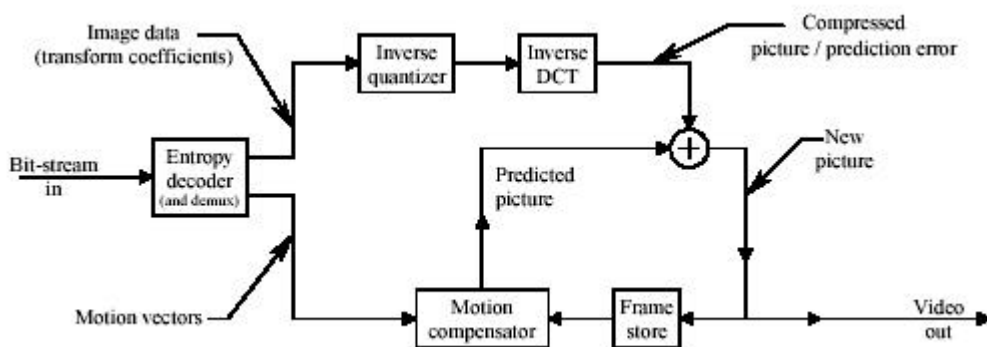


图2-2b 典型的解码器结构

Fig.2.2a Typical video decoder

算术码—算术码在图像数据压缩标准(如JPEG, JBIG)中扮演了重要的角色。在不知信源统计特性的情况下,只要监视一小段时间内码符号出现的频度,不管统计是平稳的还是非平稳的,编成的码率总能趋近于信源熵值。在算术码中,消息用0到1之间的实数进行编码,算术码用到两个基本的参数:符号的概率和它的编码间隔。信源符号的概率决定压缩编码的效率,也决定编码过程中信源符号的间隔,而这些间隔包含在0到1之间。编码过程中的间隔决定了符号压缩后的输出。算术码的优点就是能更精确的描述符号概率。

2.1.5 典型的编解码器结构

综合上面介绍的几种方法,就可以组合出一个典型的编解码器。图2-2a、图2-2b图示一个典型的DCT变换的编解码器,从图中可以看出,编码器端也有一个解码器,这样可以准确的进行运动补偿。有一点需要指出的是,反DCT变换后的系数应该精确匹配,这样差值才有意义,但是由于在计算机总会截去部分精度,难免会产生误差,因此不可能精确匹配,这样就会造成误差累积。

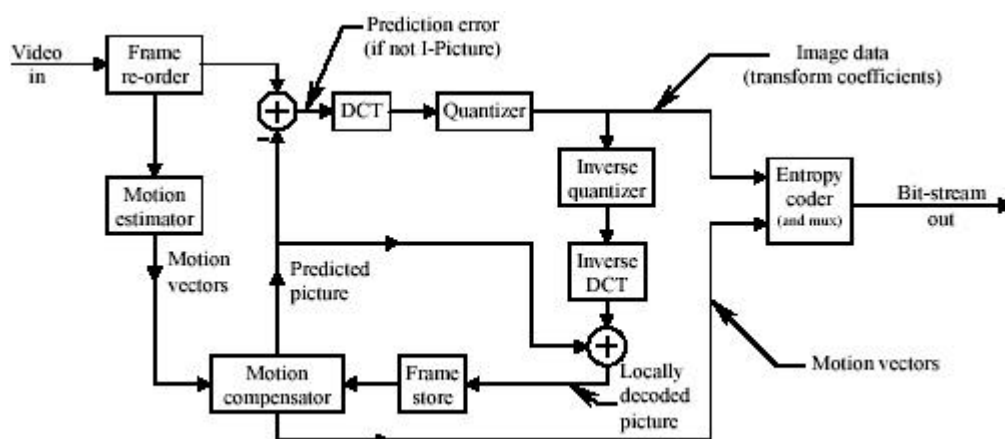


图2-2a 典型的编码器结构

Fig.2.2a Typical video coder

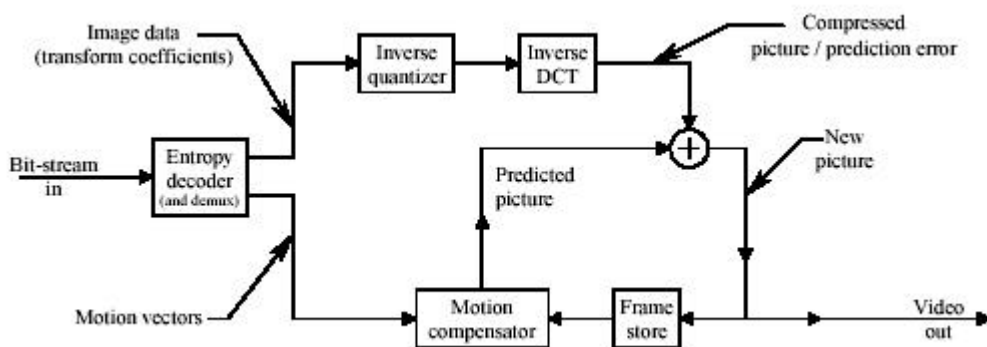


图2-2b 典型的解码器结构

Fig.2.2a Typical video decoder

2.2 图像和视频压缩的国际标准^[2]

从1990年H.261视频压缩标准正式出台以来,新的技术、算法和方案不断的涌现。主要有两大组织ITU-T和ISO/IEC贡献很多。ITU-T主要针对低码率视频编码制订的,标准为H.26x系列,ISO/IEC主要提出了MPEG标准,它不仅包括视频还有音频、系统和复用等。

2.2.1 H.261

H.261 主要是为(64kb/s)码率的实时视频应用设计的。主要应用 DCT、VLC、ME / MC 技术,是视频编码标准的原型。

2.2.2 MPEG-1

MPEG-1 是为 1.1Mbps/s 下得到录像机品质的视频质量设计的。它在 H.261 的基础上加入了 B 帧。码流分为宏块条 slice, 每一个 slice 都有自己的头, 当出现比特错误时, 用来重同步。错误的宏块用相应解好的宏块代替, 直到下一个宏块正确解码。

2.2.3 MPEG-2 / H.262

在 MPEG-1 的基础上, ISO/IEC 制定了 MPEG-2 标准, ITU-T 采纳为 H.262。MPEG-2 和 MPEG-1 的主要不同之处在于 MPEG-2 采用分层结构和支持多格式输入、支持场格式、数字电视(标清、高清)和网络通讯。分层的结构把码流分为基本层和增强层, 这样不同的网络条件和性能不同的解码器有相应的码流。此外还引入错误掩盖以及相关技术。

2.2.4 H.263

H.263 在 H.261 的基础上作进一步的改进, 它在更低的码率下有更好的视频质量。重要的一点是它引入了半像素运动估计。在第一版中加入了四种编码选项: 非限定运动矢量、算术编码、高级预测和 PB 帧。在它的后续版本 H.263+、H.263++中又加入了一些更先进的技术。H.264 是在充分吸取了它们优点的基础上才取得很大成功的。

2.2.5 MPEG-4

MPEG-4 是一个全新概念上的编码标准, 它支持的码率从 5kb/s 到 10Mbps/s。它把图像中的物体看作为独立的对象进行编码。MPEG-4 在 MPEG-2 的基础上加入纹理编码、形状编码、静止纹理编码、脸部对象编码、网格对象编码、Sprite 对象编码以及可分级编码等。

2.2.6 JPEG-2000

JPEG-2000 是 ISO/IEC 的另一个静止图像专家组的最新的静止图像压缩标准。它的早期版本也是使用 DCT 变换, 但在 JPEG-2000 中使用了小波变换来获得更高的压缩比。

2.3 H.264 总体概述^[4]

前面指出 H.264 是在充分吸取以往成功的编码标准的基础上，添加上自己的新技术或是对原有的方法进行改进。下面对这些新的或是改进的方法作一个介绍。

2.3.1 七种子宏块

宏块的大小仍是 16X16 个像素，但是划分为两层，宏块层和 8X8 子宏块层。每一层又被划分为更小的子块，如图 2—3 所示。

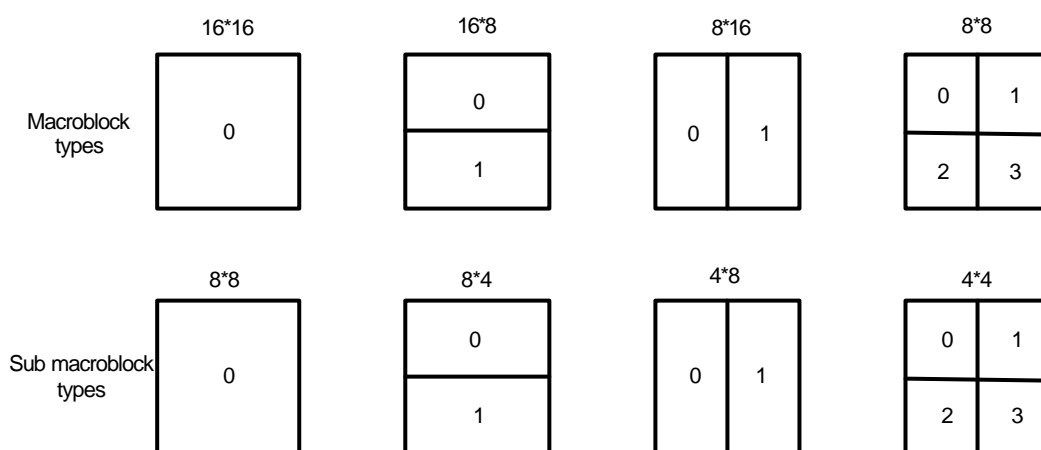


图 2—3 宏块和子宏块的划分

Fig.2.3 Macroblock and subblock number

按像素大小可分为 16X16、16X8、8X16、8X8、8X4、4X8、4X4 七种不同的块，这样对应的宏块则包括了 1、2、4、8、16 个运动矢量。

2.3.2 整数变换

H.264 将整幅图像分成 4×4 像素大小的块进行映射编码。与以往的编码标准中 DCT 变换不同的是，H.264 使用一种整数变换，其基本编码特性与 4×4 点的二维 DCT 变换相似。将二维整数变换分解为一维整数变换，先进行行整数变换，再进行列整数变换。在变换过程中系数均为整数，保证了编解码端的数据可以精确匹配。

整数变换是和量化紧密结合在一起的，在整数变换过程中，所有的系数都会扩大相应的倍数，结合量化可以减少复杂度和提高量化精度。

2.3.3 Intra 帧预测

Intra 帧预测是 H.264 的一个重要的特性，它是根据已编好的帧内相邻块的像素值，来预测当前编码块像素值。然后和原始值差，对残差进行整数变换。在预测时是按 4×4 的子块进行 9 种不同方式预测的，加上 7 种不同类型的块，所以在图像平坦区域，预测值和原始值是很接近的，这就可以节约很多的比特数，使得在相同的码率下，H.264 的帧内编

2.3 H.264 总体概述^[4]

前面指出 H.264 是在充分吸取以往成功的编码标准的基础上，添加上自己的新技术或是对原有的方法进行改进。下面对这些新的或是改进的方法作一个介绍。

2.3.1 七种子宏块

宏块的大小仍是 16X16 个像素，但是划分为两层，宏块层和 8X8 子宏块层。每一层又被划分为更小的子块，如图 2—3 所示。

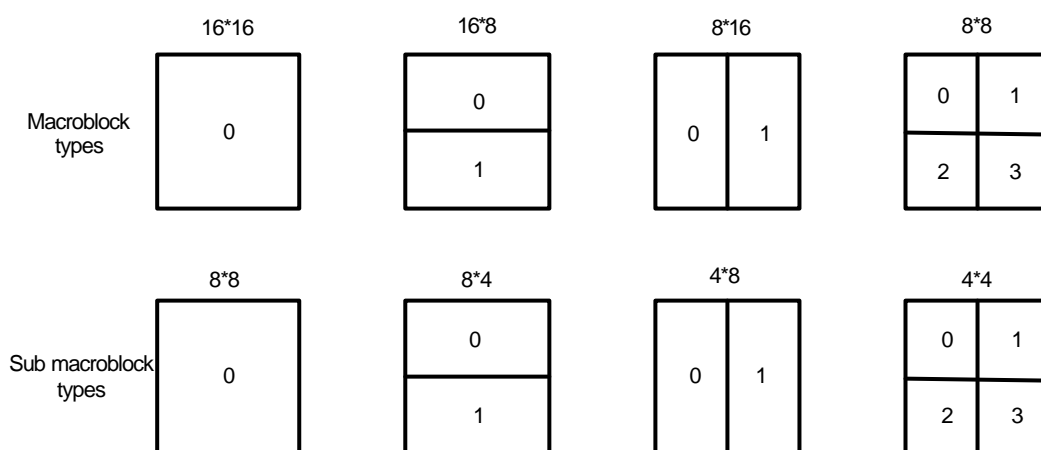


图 2—3 宏块和子宏块的划分

Fig.2.3 Macroblock and subblock number

按像素大小可分为 16X16、16X8、8X16、8X8、8X4、4X8、4X4 七种不同的块，这样对应的宏块则包括了 1、2、4、8、16 个运动矢量。

2.3.2 整数变换

H.264 将整幅图像分成 4×4 像素大小的块进行映射编码。与以往的编码标准中 DCT 变换不同的是，H.264 使用一种整数变换，其基本编码特性与 4×4 点的二维 DCT 变换相似。将二维整数变换分解为一维整数变换，先进行行整数变换，再进行列整数变换。在变换过程中系数均为整数，保证了编解码端的数据可以精确匹配。

整数变换是和量化紧密结合在一起的，在整数变换过程中，所有的系数都会扩大相应的倍数，结合量化可以减少复杂度和提高量化精度。

2.3.3 Intra 帧预测

Intra 帧预测是 H.264 的一个重要的特性，它是根据已编好的帧内相邻块的像素值，来预测当前编码块像素值。然后和原始值差，对残差进行整数变换。在预测时是按 4×4 的子块进行 9 种不同方式预测的，加上 7 种不同类型的块，所以在图像平坦区域，预测值和原始值是很接近的，这就可以节约很多的比特数，使得在相同的码率下，H.264 的帧内编

码效率有很大的提升，试验表明它和 JPEG-2000 的编码效率比较接近。另一种是针对平坦区域的 16X16 大小的预测。

2.3.4 多帧参考

H.264 吸收了 H.263++附录 U 中的多参考帧技术，使得在 P、B 帧预测时预测误差更小。同时 H.264 还对不同时间的参考帧引入不同的权重，这个特点使得 H.264 在选择参考帧时扩展到空间和时间上，使得码率控制得更精确。

2.3.5 通用变长编码 (UVLC)

在 H.264 中除了变换后的系数以外，其它的符号根据一个固定的公式来计算码字。变换的系数在此基础上加入了对已经编码系数的统计特性，这样可以更接近符号概率，提高编码效率。

2.3.6 自适应块变换 (ABT)

在量化系数较小时，整数变换不再是以 4X4 的大小进行变换，而是按照子块的大小进行 8X4、4X8 或 8X8 变换。相应的帧内预测、量化等都要根据子块的大小作相应的调整。

2.3.7 视频编码层 (VCL) 和网络适应层 (NAL)

第一章中指出，在网络环境中视频广播是 H.264 的一大目标，所以，H.264 加入了针对 IP 环境的网络适配层，这样 H.264 中的整体框架分为视频编码层 (Video Coding Layer) 和网络适配层 (Network Adaptation Layer)，如图 2-4 所示。

VCL 层以编码效率为中心，同时也适当的考虑抗误码因素。因此 VCL 层可以根据误码出现的具体情况而采取相应的措施：

- 插入帧内帧，以便可以随机接入和改善图像质量
- 引入图像片概念，防止误码扩散
- 使用独立解码的 VLC
- 根据信道的变化而改变码率

网络服务的 QoS 机制由 VCL 层把数据划分优先级来实现，不同优先级的数据有不同的重要性。NAL 层的主要任务是在一定的网络负载下，综合各种数据类型有一个最佳的传输码流。因此，NAL 层应作如下工作：

- 把 VCL 层的不同数据类型映射到码流包中
- 把打好的包传输到网络
- 如有可能，提供更好的传输机制

2.3.8 其他特点

码效率有很大的提升，试验表明它和 JPEG-2000 的编码效率比较接近。另一种是针对平坦区域的 16X16 大小的预测。

2.3.4 多帧参考

H.264 吸收了 H.263++附录 U 中的多参考帧技术，使得在 P、B 帧预测时预测误差更小。同时 H.264 还对不同时间的参考帧引入不同的权重，这个特点使得 H.264 在选择参考帧时扩展到空间和时间上，使得码率控制得更精确。

2.3.5 通用变长编码 (UVLC)

在 H.264 中除了变换后的系数以外，其它的符号根据一个固定的公式来计算码字。变换的系数在此基础上加入了对已经编码系数的统计特性，这样可以更接近符号概率，提高编码效率。

2.3.6 自适应块变换 (ABT)

在量化系数较小时，整数变换不再是以 4X4 的大小进行变换，而是按照子块的大小进行 8X4、4X8 或 8X8 变换。相应的帧内预测、量化等都要根据子块的大小作相应的调整。

2.3.7 视频编码层 (VCL) 和网络适应层 (NAL)

第一章中指出，在网络环境中视频广播是 H.264 的一大目标，所以，H.264 加入了针对 IP 环境的网络适配层，这样 H.264 中的整体框架分为视频编码层 (Video Coding Layer) 和网络适配层 (Network Adaptation Layer)，如图 2-4 所示。

VCL 层以编码效率为中心，同时也适当的考虑抗误码因素。因此 VCL 层可以根据误码出现的具体情况而采取相应的措施：

- 插入帧内帧，以便可以随机接入和改善图像质量
- 引入图像片概念，防止误码扩散
- 使用独立解码的 VLC
- 根据信道的变化而改变码率

网络服务的 QoS 机制由 VCL 层把数据划分优先级来实现，不同优先级的数据有不同的重要性。NAL 层的主要任务是在一定的网络负载下，综合各种数据类型有一个最佳的传输码流。因此，NAL 层应作如下工作：

- 把 VCL 层的不同数据类型映射到码流包中
- 把打好的包传输到网络
- 如有可能，提供更好的传输机制

2.3.8 其他特点

码效率有很大的提升，试验表明它和 JPEG-2000 的编码效率比较接近。另一种是针对平坦区域的 16X16 大小的预测。

2.3.4 多帧参考

H.264 吸收了 H.263++附录 U 中的多参考帧技术，使得在 P、B 帧预测时预测误差更小。同时 H.264 还对不同时间的参考帧引入不同的权重，这个特点使得 H.264 在选择参考帧时扩展到空间和时间上，使得码率控制得更精确。

2.3.5 通用变长编码 (UVLC)

在 H.264 中除了变换后的系数以外，其它的符号根据一个固定的公式来计算码字。变换的系数在此基础上加入了对已经编码系数的统计特性，这样可以更接近符号概率，提高编码效率。

2.3.6 自适应块变换 (ABT)

在量化系数较小时，整数变换不再是以 4X4 的大小进行变换，而是按照子块的大小进行 8X4、4X8 或 8X8 变换。相应的帧内预测、量化等都要根据子块的大小作相应的调整。

2.3.7 视频编码层 (VCL) 和网络适应层 (NAL)

第一章中指出，在网络环境中视频广播是 H.264 的一大目标，所以，H.264 加入了针对 IP 环境的网络适配层，这样 H.264 中的整体框架分为视频编码层 (Video Coding Layer) 和网络适配层 (Network Adaptation Layer)，如图 2-4 所示。

VCL 层以编码效率为中心，同时也适当的考虑抗误码因素。因此 VCL 层可以根据误码出现的具体情况而采取相应的措施：

- 插入帧内帧，以便可以随机接入和改善图像质量
- 引入图像片概念，防止误码扩散
- 使用独立解码的 VLC
- 根据信道的变化而改变码率

网络服务的 QoS 机制由 VCL 层把数据划分优先级来实现，不同优先级的数据有不同的重要性。NAL 层的主要任务是在一定的网络负载下，综合各种数据类型有一个最佳的传输码流。因此，NAL 层应作如下工作：

- 把 VCL 层的不同数据类型映射到码流包中
- 把打好的包传输到网络
- 如有可能，提供更好的传输机制

2.3.8 其他特点

H.264 还继承了 H.263++和 MPEG-4 的一些优点, 例如: 自适应的算术编码 (CABAC)、数据分割、1/4 或 1/8 像素精度、档次和分级等。

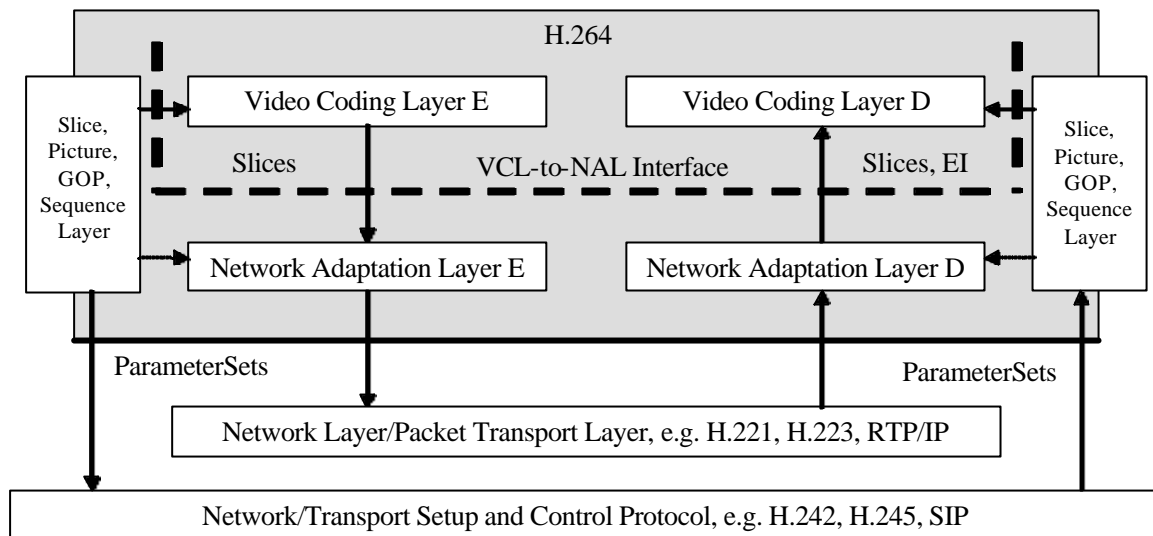


图2-4 VCL层和NAL层结构图

Fig2.4 The structure of VCL layer and NAL

2.4 小结

本章第一节中归纳总结了视频编码中常见的压缩技术, 给出了一个典型的编解码器的框图。第二节中对目前使用的动态、静态图像编码标准作了一个回顾, 简单介绍了它们使用的技术和适用的范围。第三节中对 H.264 中的新技术、新方案和改进的地方作总体的介绍。

H.264 还继承了 H.263++和 MPEG-4 的一些优点, 例如: 自适应的算术编码 (CABAC)、数据分割、1/4 或 1/8 像素精度、档次和分级等。

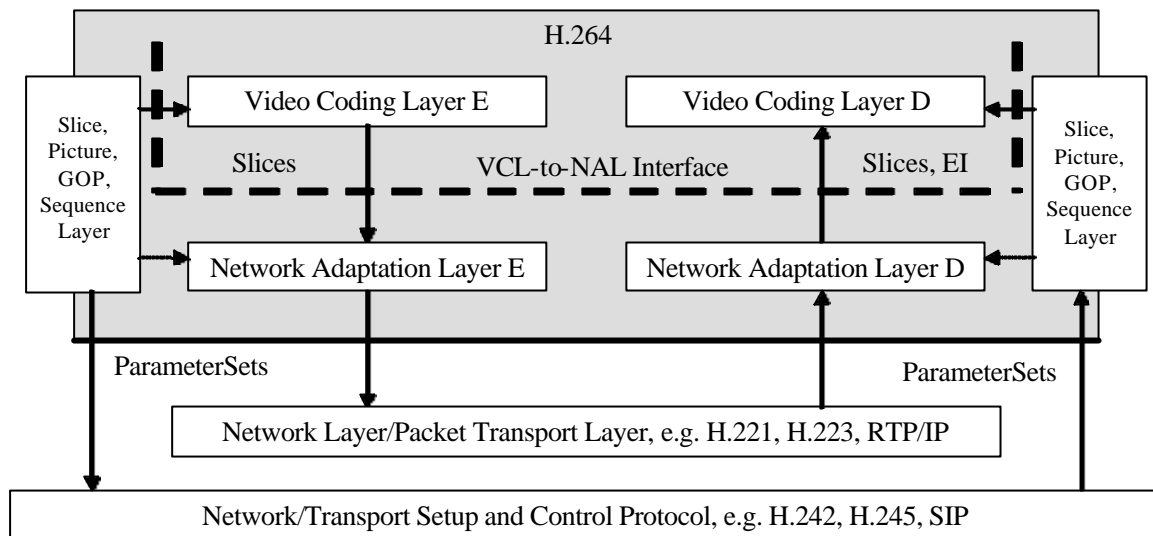


图2-4 VCL层和NAL层结构图

Fig2.4 The structure of VCL layer and NAL

2.4 小结

本章第一节中归纳总结了视频编码中常见的压缩技术, 给出了一个典型的编解码器的框图。第二节中对目前使用的动态、静态图像编码标准作了一个回顾, 简单介绍了它们使用的技术和适用的范围。第三节中对 H.264 中的新技术、新方案和改进的地方作总体的介绍。

本章参考:

- [1] Chistopoulos, Charis (editor), "JPEG2000 Verification Model 5.0."
ISO/IECJTC/SC29/WG1 N1422, August 1999.
- [2] Kristofer Dovstam, "Video coding in H.26L", Ericsson Radio Systems AB, Kista,
Sweden, Ericsson Research, Department of Audio & Visual, Technology. April 2000.
- [3] Tom McMahon, "The Emerging H.264/AVC Video Coding Standard", DGFx,
Inc., sept, 2002.
- [4] Joint Final Committee Draft (JFCD) of Joint Video Specification, ISO/IEC
JTC1/SC29/WG11 and ITU-T Q6/SG16, Document JVT-D157, Jul. 2002.

第三章 Intra 帧预测及改进

3.1 引言

当宏块是帧内方式编码时，编码的宏块根据重建宏块预测像素值，然后对预测值和原始值差值 DCT 变换。对亮度可以按照 4X4 的子块预测或 16X16 的宏块预测。当以 4X4 大小预测时有 9 种方式，以 16X16 大小预测时有 4 种方式。色度的预测方式只有 4 种。

3.2 4X4 亮度预测

图3-1^[1]中是QCIF的一帧图像中的宏块和预测的4X4的亮度子块。子块的上面和左边的宏块已经编码和重建过，因此可以用来作为预测参考。预测块P基于相邻的宏块的17个像素值，如图3-2所示。为了保证宏块条的独立性，A-Q必须在同一个宏块条中。

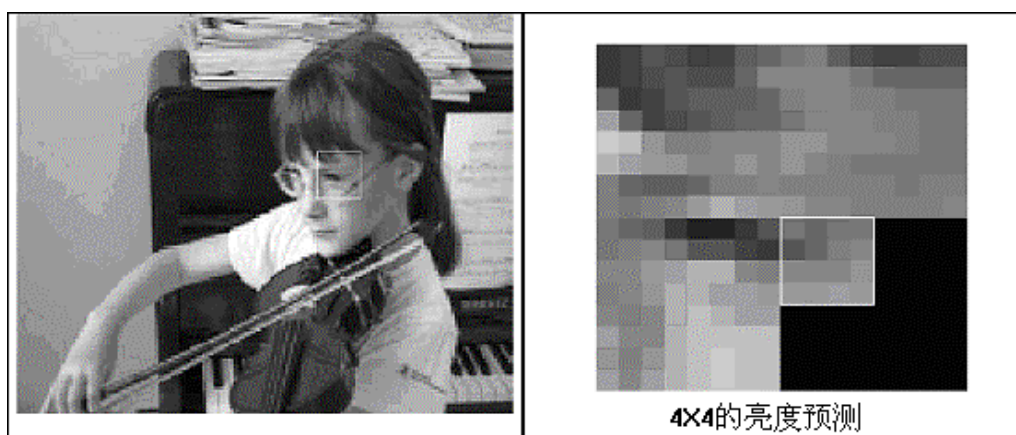


图3-1 原始宏块和4X4大小的预测块

Fig.3.1 Original macroblock; 4x4 luma block to be predicted

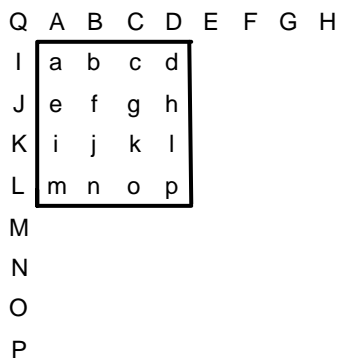


图3-2 4X4块的相邻像素值

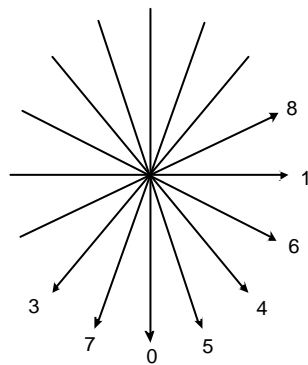


图3-3 预测方向

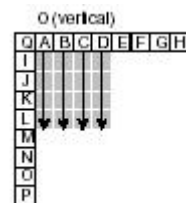
Fig.3.2 Identification of samples used for intra spatial prediction Fig.3.3 Intra prediction directions

图3—3^[1]中显示了9种预测模式的方向，下面对9种模式作说明:

3.2.1 模式 0: 垂直预测

在只有 A, B, C, D 四个像素值可用时, 预测过程为:

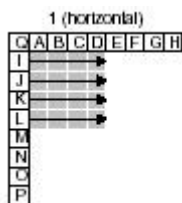
- a, e, i, m 由 A 预测,
- b, f, j, n 由 B 预测,
- c, g, k, o 由 C 预测,
- d, h, l, p 由 D 预测。



3.2.2 模式 1: 水平预测

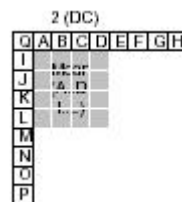
在只有 I, J, K, L 四个像素值可用时, 预测过程为:

- a, b, c, d 由 I 预测,
- e, f, g, h 由 J 预测,
- i, j, k, l 由 K 预测,
- m, n, o, p 由 L 预测。



3.2.3 模式 2: DC 预测

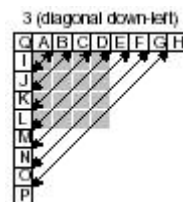
如果 A,B,C,D,I,J,K,L 都可用, 那么所有的像素值为 $(A+B+C+D+I+J+K+L+4) \gg 3$ 。如果 A,B,C,D 不可用, 而 I,J,K,L 可用, 所有像素值为 $(I+J+K+L+2) \gg 2$, 反之则为 $(A+B+C+D+2) \gg 2$ 。如果 A,B,C,D,I,J,K,L 均不可用, 所有的像素值为 128。



3.2.4 模式 3: 左下对角预测

当块周围的 17 个像素都可用时, 预测过程为:

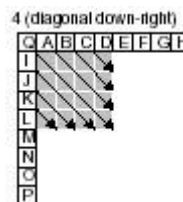
- a 预测为 $(A + 2B + C + I + 2J + K + 4) \gg 3$
- b, e 预测为 $(B + 2C + D + J + 2K + L + 4) \gg 3$
- c, f, i 预测为 $(C + 2D + E + K + 2L + M + 4) \gg 3$
- d, g, j, m 预测为 $(D + 2E + F + L + 2M + N + 4) \gg 3$
- h, k, n 预测为 $(E + 2F + G + M + 2N + O + 4) \gg 3$
- l, o 预测为 $(F + 2G + H + N + 2O + P + 4) \gg 3$
- p 预测为 $(G + H + O + P + 2) \gg 2$



3.2.5 模式 4: 右下对角预测

当 A,B,C,D,I,J,K,L,Q 可用时, 预测过程为:

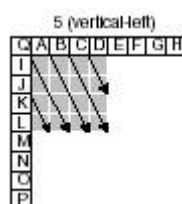
- m 预测为 $(J + 2K + L + 2) \gg 2$
- i, n 预测为 $(I + 2J + K + 2) \gg 2$
- e, j, o 预测为 $(Q + 2I + J + 2) \gg 2$
- a, f, k, p 预测为 $(A + 2Q + I + 2) \gg 2$
- b, g, l 预测为 $(Q + 2A + B + 2) \gg 2$
- c, h 预测为 $(A + 2B + C + 2) \gg 2$
- d 预测为 $(B + 2C + D + 2) \gg 2$



3.2.6 模式 5: 垂直左斜下角预测

当 A,B,C,D,I,J,K,L,Q 可用时, 预测过程为:

- a, j 预测为 $(Q + A + 1) \gg 1$
- b, k 预测为 $(A + B + 1) \gg 1$

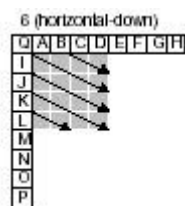


- c, l 预测为 $(B + C + 1) \gg 1$
- d 预测为 $(C + D + 1) \gg 1$
- e, n 预测为 $(I + 2Q + A + 2) \gg 2$
- f, o 预测为 $(Q + 2A + B + 2) \gg 2$
- g, p 预测为 $(A + 2B + C + 2) \gg 2$
- hs 预测为 $(B + 2C + D + 2) \gg 2$
- I 预测为 $(Q + 2I + J + 2) \gg 2$
- m 预测为 $(I + 2J + K + 2) \gg 2$

3.2.7 模式 6：水平斜下角预测

当 A,B,C,D,I,J,K,L,Q 可用时，预测过程为：

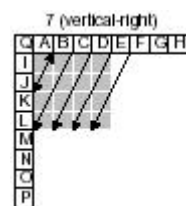
- a, g 预测为 $(Q + I + 1) \gg 1$
- b, h 预测为 $(I + 2Q + A + 2) \gg 2$
- c 预测为 $(Q + 2A + B + 2) \gg 2$
- d 预测为 $(A + 2B + C + 2) \gg 2$
- e, k 预测为 $(I + J + 1) \gg 1$
- f, l 预测为 $(Q + 2I + J + 2) \gg 2$
- i, o 预测为 $(J + K + 1) \gg 1$
- j, p 预测为 $(I + 2J + K + 2) \gg 2$
- m 预测为 $(K + L + 1) \gg 1$
- n 预测为 $(J + 2K + L + 2) \gg 2$



3.2.8 模式 7：垂直右下角预测

当 A,B,C,D,I,J,K,L,Q 可用时，预测过程为：

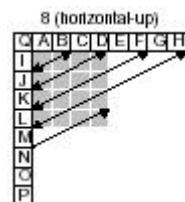
- a 预测为 $(2A + 2B + J + 2K + L + 4) \gg 3$
- b, I 预测为 $(B + C + 1) \gg 1$
- c, j 预测为 $(C + D + 1) \gg 1$
- d, k 预测为 $(D + E + 1) \gg 1$
- l 预测为 $(E + F + 1) \gg 1$
- e 预测为 $(A + 2B + C + K + 2L + M + 4) \gg 3$
- f, m 预测为 $(B + 2C + D + 2) \gg 2$
- g, n 预测为 $(C + 2D + E + 2) \gg 2$
- h, o 预测为 $(D + 2E + F + 2) \gg 2$
- p 预测为 $(E + 2F + G + 2) \gg 2$



3.2.9 模式 8：水平斜上角预测

当 A,B,C,D,I,J,K,L,Q 可用时，预测过程为：

- a 预测为 $(B + 2C + D + 2I + 2J + 4) \gg 3$
- b 预测为 $(C + 2D + E + I + 2J + K + 4) \gg 3$
- c, e 预测为 $(J + K + 1) \gg 1$
- d, f 预测为 $(J + 2K + L + 2) \gg 2$
- g, i 预测为 $(K + L + 1) \gg 1$
- h, j 预测为 $(K + 2L + M + 2) \gg 2$
- l, n 预测为 $(L + 2M + N + 2) \gg 2$
- k, m 预测为 $(L + M + 1) \gg 1$
- o 预测为 $(M + N + 1) \gg 1$
- p 预测为 $(M + 2N + O + 2) \gg 2$



3.3 16X16 亮度预测

对于图像中大块的平坦区域来说，4X4 的预测过于复杂，此时可以用 16X16 的宏块大小来预测，在此方式下共有 4 种预测模式。在 16X16 宏块的上方有 16 个像素，记为： $P(i, -1), i=0, \dots, 15$ ；下方的 16 个像素记为： $P(-1, j), j=0, \dots, 15$ 。Pred(i, j) 为预测的像素。

3.3.1 模式 0：垂直预测

$$\text{Pred}(i, j) = P(i, -1); \quad i, j = 0, \dots, 15$$

3.3.2 模式 1：水平预测

$$\text{Pred}(i, j) = P(-1, j); \quad i, j = 0, \dots, 15$$

3.3.3 模式 2：DC 预测

$$\text{Pred}(i, j) = \left[\sum_{i'=0}^{15} P(i', -1) + \sum_{j'=0}^{15} P(-1, j') + 16 \right] \gg 5 \quad i, j = 0, \dots, 15$$

如果 $P(i, -1)$ 不可用，则：

$$\text{Pred}(i, j) = \left[\sum_{j'=0}^{15} P(-1, j') + 8 \right] \gg 4 \quad x, y = 0, \dots, 15$$

如果 $P(-1, j)$ 不可用，则：

$$\text{Pred}(i, j) = \left[\sum_{i'=0}^{15} P(i', -1) + 8 \right] \gg 4 \quad x, y = 0, \dots, 15$$

如果 $P(i, -1)$ 和 $P(-1, j)$ 都不可用，则所有的像素值为 128。

3.3.4 模式 3：平坦预测

$$\text{Pred}(i, j) = ((a + b \cdot (i-7) + c \cdot (j-7) + 16) \gg 5)$$

其中：

$$a = 16 \cdot (P(-1, 15) + P(15, -1)) \quad b = (5 \cdot H + 32) \gg 6 \quad c = (5 \cdot V + 32) \gg 6$$

$$H = \sum_{x=0}^8 x \cdot (P(7+x, -1) - P(7-x, -1)) \quad V = \sum_{y=0}^8 y \cdot (P(-1, 7+y) - P(-1, 7-y))$$

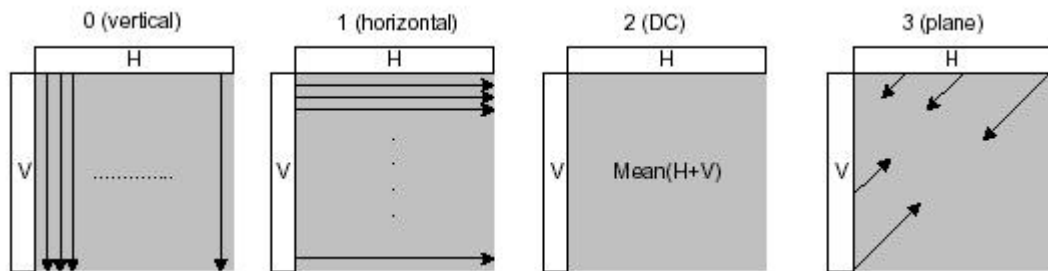


图 3-5 16X16 预测模式

Fig.3.5 H.26L Intra 16x16 prediction modes (all predicted from pixels H and V)

3.4 8X8 色度预测^[2]

色度预测和 16X16 的亮度预测很相似,只是在维度上变成了一半。

3.4.1 模式0 : 垂直预测

$$F(0,-1) = (P(0,-1) + P(1,-1) + 1) \gg 1$$

$$F(x,-1) = (P(x-1,-1) + 2 \times P(x,-1) + P(x+1,-1) + 2) \gg 2, \quad x=1, \dots, 6$$

$$F(7,-1) = (P(6,-1) + P(7,-1) + 1) \gg 1$$

$$\text{Pred}(x, y) = F(x, -1), \quad x, y=0..7$$

3.4.2 模式1 : 水平预测

$$F(-1,0) = (P(-1,0) + P(-1,1) + 1) \gg 1$$

$$F(-1, j) = (P(-1, j-1) + 2 \times P(-1, j) + P(-1, j+1) + 2) \gg 2, \quad j=1, \dots, 6$$

$$F(-1, 7) = (P(-1, 6) + P(-1, 7) + 1) \gg 1$$

$$\text{Pred}(i, j) = F(-1, j), \quad i, j=0..7$$

3.4.3 模式2 : DC 预测

$$\text{Pred}(i, j) = \left(\left(\sum_{j=0}^7 (P(-1, j) + P(i, -1)) \right) + 8 \right) \gg 4, \quad i, j=0, \dots, 7$$

如果 $P(i, -1)$ 不可用, 则:

$$\text{Pred}(i, j) = \left[\left(\sum_{j=0}^7 P(-1, j) \right) + 4 \right] \gg 3, \quad i, j=0, \dots, 7$$

如果 $P(-1, j)$ 不可用, 则:

$$\text{Pred}(i, j) = \left[\left(\sum_{i=0}^7 P(i, -1) \right) + 4 \right] \gg 3, \quad i, j=0, \dots, 7$$

如果 $P(i, -1)$ 和 $P(-1, j)$ 都不可用, 则所有的像素值为 128。

3.4.4 模式3 : 平坦预测

$$\text{Pred}(i, j) = (a + b \cdot (i-3) + c \cdot (j-3) + 16) \gg 5, \quad i, j=0, \dots, 7$$

其中:

$$a = 16 \cdot (P(-1, 7) + P(7, -1)) \quad b = (17 \cdot H + 16) \gg 5 \quad c = (17 \cdot V + 16) \gg 5$$

$$H = \sum_{i=1}^4 i \cdot (P(3+i, -1) - P(3-i, -1)) \quad V = \sum_{j=1}^4 j \cdot (P(-1, 3+j) - P(-1, 3-j))$$

3.5 预测模式编码

4X4 块的亮度、16X16 宏块的亮度、8X8 块的色度在预测后, 均可得到一个最佳的预测模式, 编码器需要把这些信息传送到解码器。如果对所有的块的预测值直接编码, 这样会

使生成的码率变大，例如：一个宏块中，有 16 个 4X4 的块，预测值范围是 0~8，那么最小占用比特数为 16，最大为 112。为 16 的情况显然是不可能的，因为所有都为模式 0 的垂直预测，还不如不要预测来的好。一般情况下要占用 80 个比特位。

研究表明预测块的最佳模式和相邻块的预测模式是密切相关的，如图 3—6 所示^[2]。当块 A 和 B 的预测模式已知时，当前块 C 的最可能预测模式也就决定了，为 A、B 中的最小模式（当 A、B 中任一个在边界时为 2）。例如：A 为模式 3 的左下对角预测、B 为模式 2 的 DC 预测，那么 C 最可能的模式为 2—DC 预测。当真实的预测模式为 2 时，则置 use_most_probable_mode 为 1，如果真实的预测模式不为 2 时，则置 use_most_probable_mode 为 0，然后把真实的模式信息通过公式 (3.5-1) 计算后编码。

```

if (remaining_mode_selector < most_probable_mode)
    remaining_mode_selector = intra_pred_mode;
else
    remaining_mode_selector = intra_pred_mode - 1;
remaining_mode_selector(编码值)      intra_pred_mode(真实值)

```

(3.5-1)

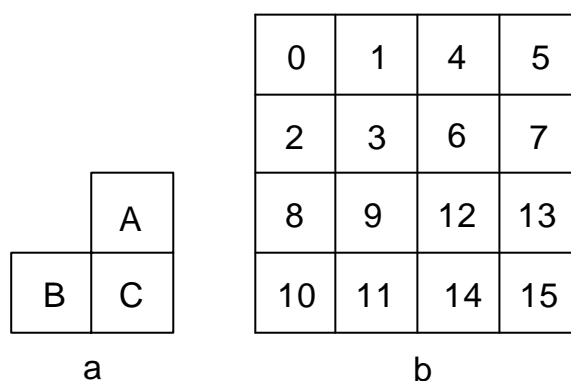


图 3—6 A、B 预测块 C 的相邻块(a) 宏块中的块预测顺序(b)

Fig.3.6 Adjacent 4x4 intra coded blocks

16X16 宏块的帧内预测模式和色度的帧内预测模式分别嵌入到宏块类型中单独编码。

3.6 帧内编码效率比较

H.264 的帧内预测可以在取得高的信噪比的同时极大的节约了比特数，下面以图表的形式对 H.264 和 H.263 的帧内压缩效果作两方面的对比。所作的比较都是在按照算术编码，码字包括可完全解码一帧需要的所有信息。使用的序列为 QCIF carphone、claire、grandma，如图 3—7 所示。三幅图都有一定的代表性，细节和块的分布各不相同，能够过反映预测效果的性能。

使生成的码率变大，例如：一个宏块中，有 16 个 4X4 的块，预测值范围是 0~8，那么最小占用比特数为 16，最大为 112。为 16 的情况显然是不可能的，因为所有都为模式 0 的垂直预测，还不如不要预测来的好。一般情况下要占用 80 个比特位。

研究表明预测块的最佳模式和相邻块的预测模式是密切相关的，如图 3—6 所示^[2]。当块 A 和 B 的预测模式已知时，当前块 C 的最可能预测模式也就决定了，为 A、B 中的最小模式（当 A、B 中任一个在边界时为 2）。例如：A 为模式 3 的左下对角预测、B 为模式 2 的 DC 预测，那么 C 最可能的模式为 2—DC 预测。当真实的预测模式为 2 时，则置 use_most_probable_mode 为 1，如果真实的预测模式不为 2 时，则置 use_most_probable_mode 为 0，然后把真实的模式信息通过公式 (3.5-1) 计算后编码。

```

if (remaining_mode_selector < most_probable_mode)
    remaining_mode_selector = intra_pred_mode;
else
    remaining_mode_selector = intra_pred_mode - 1;
remaining_mode_selector(编码值)      intra_pred_mode(真实值)

```

(3.5-1)

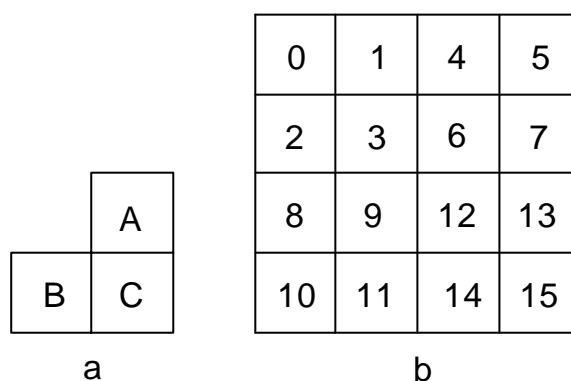


图 3—6 A、B 预测块 C 的相邻块(a) 宏块中的块预测顺序(b)

Fig.3.6 Adjacent 4x4 intra coded blocks

16X16 宏块的帧内预测模式和色度的帧内预测模式分别嵌入到宏块类型中单独编码。

3.6 帧内编码效率比较

H.264 的帧内预测可以在取得高的信噪比的同时极大的节约了比特数，下面以图表的形式对 H.264 和 H.263 的帧内压缩效果作两方面的对比。所作的比较都是在按照算术编码，码字包括可完全解码一帧需要的所有信息。使用的序列为 QCIF carphone、claire、grandma，如图 3—7 所示。三幅图都有一定的代表性，细节和块的分布各不相同，能够过反映预测效果的性能。

3.6.1 H.264 与 H.263 的 PSNR 对比

表 3—1、图 3—7 中对比了在相近的比特数下, H.264 与 H.263 对不同序列的第一帧 PSNR 大小。从表和图中可以看出, 在量化值小时, H.264 的帧内编码效率比 H.263 要高 10~20dB, 在量化值大的情况下, H.264 同样比 H.263 的编码效率高 4~5 个 dB, 反映在主观评价更是相差很远。

表3—1(a) Carphone的第一帧Y、Cb、Cr、Bits对比

Table.3.1(a) several comparisons of the first frame of Carphone

Carphone	Y(dB)	Cb(dB)	Cr(dB)	Bits
H.263	36.95	48.05	48.05	128584
	33.84	39.24	39.32	21776
H.264	54.33	54.42	54.61	124472
	37.49	40.27	41.31	22216

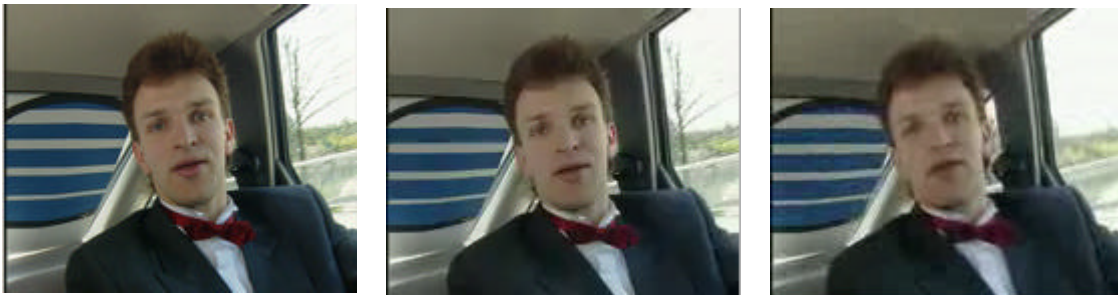


图 3—7(a) Carphone 的原始帧、H.264 在 22216bits 下重建帧、H.263 在 21776bits 下重建帧

Fig.3.7(a) The original frame and two reconstructed frames

表3—1(b) Claire的第一帧Y、Cb、Cr、Bits对比

Table.3.1(b) several comparisons of the first frame of Claire

Claire	Y(dB)	Cb(dB)	Cr(dB)	Bits
H.263	53.57	53.42	53.82	119564
	37.28	37.04	39.75	16472
H.264	67.13	66.61	66.51	123488
	40.65	40.35	41.87	13256

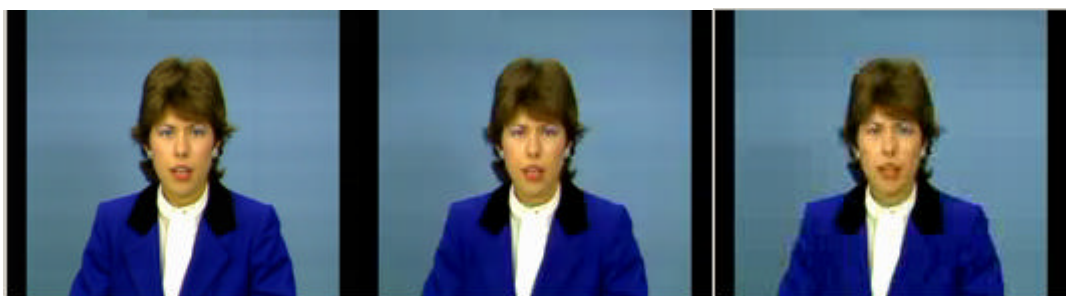


图 3—7(b) Claire 的原始帧、H.264 在 13256bits 下重建帧、H.263 在 164726bits 下重建帧

Fig.3.7(b) The original frame and two reconstructed frames

表 3-1(c) grandma 的第一帧 Y、Cb、Cr、Bits 对比

Table.3.1(c) several comparisons of the first frame of Grandma

Grandma	Y(dB)	Cb(dB)	Cr(dB)	Bits
H.263	52.47	52.83	52.88	202112
	33.78	37.33	37.85	15904
H.264	66.16	66.85	66.66	174376
	37.15	39.69	40.09	21088



图 3-7(c) grandma 的原始帧、H.264 在 21088bits 下重建帧、H.263 在 15904bits 下重建帧

Fig.3.7(c) The original frame and two reconstructed frames

需要指出的是，H.264 的量化阶数高达 52 阶，它的信噪比可以达到 H.263 达不到的程度，所以高码率也是 H.264 应用的主要方面，这一点是 H.263 达不到的。

3.6.2 H.264 与 JPEG-2000 的 PSNR 对比^[3]

H.264 的帧内编码性能和专用的静止图像压缩标准 JPEG、JPEG-2000 相比也有很高的性能表现，如表 3-2 所示。

表 3-2 H.264 与 JPEG、JPEG-2000 帧内编码性能比较

Chart 3.2 Comparison between H.264 and JPEG-2000

Sequence (Size)	Q	QP	Bit rate	PSNR			Difference JM to -	
	JPEG	H.26L	Bpp	JPEG	JPEG2000	H.26L	JPEG	JPEG2000
Container (QCIF)	93	4	2.711	42.64	48.22	46.75	4.11	-1.47
	76	12	1.446	35.58	40.54	40.19	4.61	-0.35
	25	20	0.674	29.74	33.16	34.41	4.67	1.25
	3	28	0.288	22.03	27.02	28.83	6.80	1.81
News (QCIF)	92	4	2.651	42.61	49.38	47.24	4.63	-2.14
	75	12	1.508	36.11	41.38	40.89	4.78	-0.49
	25	20	0.771	30.19	33.47	34.62	4.43	1.15
	4	28	0.347	23.52	27.31	28.62	5.10	1.31
Tempete (CIF)	95	4	3.707	43.52	50.91	46.68	3.16	-4.23
	84	12	2.123	36.86	41.87	39.69	2.83	-2.18
	43	20	1.031	30.81	33.67	32.95	2.14	-0.72
	9	28	0.407	25.49	27.52	27.01	1.52	-0.51

从表中数据可以看出，H.264 远比 JPEG 好，相比与 JPEG-2000，H.264 在小图和低码

表 3-1(c) grandma 的第一帧 Y、Cb、Cr、Bits 对比

Table.3.1(c) several comparisons of the first frame of Grandma

Grandma	Y(dB)	Cb(dB)	Cr(dB)	Bits
H.263	52.47	52.83	52.88	202112
	33.78	37.33	37.85	15904
H.264	66.16	66.85	66.66	174376
	37.15	39.69	40.09	21088



图 3-7(c) grandma 的原始帧、H.264 在 21088bits 下重建帧、H.263 在 15904bits 下重建帧

Fig.3.7(c) The original frame and two reconstructed frames

需要指出的是，H.264 的量化阶数高达 52 阶，它的信噪比可以达到 H.263 达不到的程度，所以高码率也是 H.264 应用的主要方面，这一点是 H.263 达不到的。

3.6.2 H.264 与 JPEG-2000 的 PSNR 对比^[3]

H.264 的帧内编码性能和专用的静止图像压缩标准 JPEG、JPEG-2000 相比也有很高的性能表现，如表 3-2 所示。

表 3-2 H.264 与 JPEG、JPEG-2000 帧内编码性能比较

Chart 3.2 Comparison between H.264 and JPEG-2000

Sequence (Size)	Q	QP	Bit rate	PSNR			Difference JM to -	
	JPEG	H.26L	Bpp	JPEG	JPEG2000	H.26L	JPEG	JPEG2000
Container (QCIF)	93	4	2.711	42.64	48.22	46.75	4.11	-1.47
	76	12	1.446	35.58	40.54	40.19	4.61	-0.35
	25	20	0.674	29.74	33.16	34.41	4.67	1.25
	3	28	0.288	22.03	27.02	28.83	6.80	1.81
News (QCIF)	92	4	2.651	42.61	49.38	47.24	4.63	-2.14
	75	12	1.508	36.11	41.38	40.89	4.78	-0.49
	25	20	0.771	30.19	33.47	34.62	4.43	1.15
	4	28	0.347	23.52	27.31	28.62	5.10	1.31
Tempete (CIF)	95	4	3.707	43.52	50.91	46.68	3.16	-4.23
	84	12	2.123	36.86	41.87	39.69	2.83	-2.18
	43	20	1.031	30.81	33.67	32.95	2.14	-0.72
	9	28	0.407	25.49	27.52	27.01	1.52	-0.51

从表中数据可以看出，H.264 远比 JPEG 好，相比与 JPEG-2000，H.264 在小图和低码

率时有好的性能，接近甚至超过 JPEG-2000，但在大图和低量化时，还是 JPEG-2000 的性能要好。

3.7 对于帧内编码改进的几点建议

H.264的帧内预测技术对于提高图像的质量、减小码率有很大的作用。但是，它的亮度和色度的预测模式很多，编码器要在这些模式中选择一个最佳的模式，需要进行大量的计算。因此，帧内预测模式的简化是本文研究的内容之一。

3.7.1 问题分析

在 4X4 的块预测中，用到了 A-Q，共 17 个像素点的值，如图 3—2 所示。这样可以精确预测块的值，但是也带来一些问题。

问题一：4X4 的块用到 17 个像素的值的比率不大，按照图 3—8 (b) 的扫描顺序，只占 5 / 16，如图 3—8 (c) 中打对号的块，这样造成 9 种预测模式效率并不是很高。

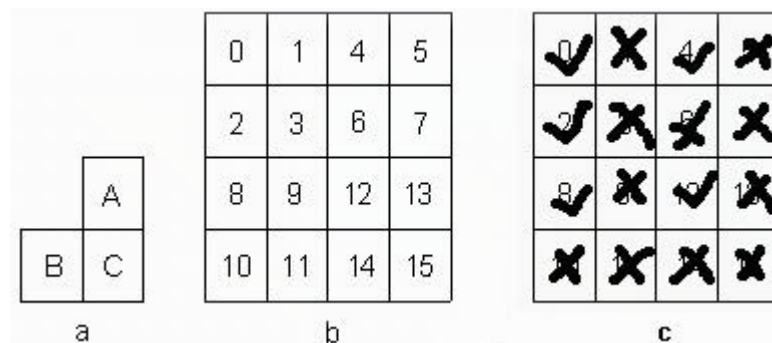


图 3—8 可以使用 17 个像素值的块

Fig.3.8 Number of usefull pixel block

问题二：由于块预测和相邻的四个块有强烈的依赖关系，只能顺序预测的，不可以并行执行，这样的结构对算法优化不利。

问题三：在表 3—1 中发现，H.264 的色度预测相比于亮度预测并没有性能上突出的表现。即在 H.263 中，色度的信噪比要比亮度信噪比相对于 H.264 要来的高。这说明色度的预测不是很突出，因此色度不必使用很复杂的预测模式。

3.7.2 图像数据分析

首先，对要预测的图像块数据做一个分析，把 Carphone 序列中的第 37、69 个宏块取出，打印出它的亮度和两个色度值，如图 3—9 所示。

从图中的数据可以看出 4X4 的块和周围 E-H、M-P 这 8 个数据的依赖关系并不是很紧密，有的地方相关，有的地方根本就没有联系，这是分块处理必然的结果，分块是不可能

率时有好的性能，接近甚至超过 JPEG-2000，但在大图和低量化时，还是 JPEG-2000 的性能要好。

3.7 对于帧内编码改进的几点建议

H.264的帧内预测技术对于提高图像的质量、减小码率有很大的作用。但是，它的亮度和色度的预测模式很多，编码器要在这些模式中选择一个最佳的模式，需要进行大量的计算。因此，帧内预测模式的简化是本文研究的内容之一。

3.7.1 问题分析

在 4X4 的块预测中，用到了 A-Q，共 17 个像素点的值，如图 3—2 所示。这样可以精确预测块的值，但是也带来一些问题。

问题一：4X4 的块用到 17 个像素的值的比率不大，按照图 3—8 (b) 的扫描顺序，只占 5 / 16，如图 3—8 (c) 中打对号的块，这样造成 9 种预测模式效率并不是很高。

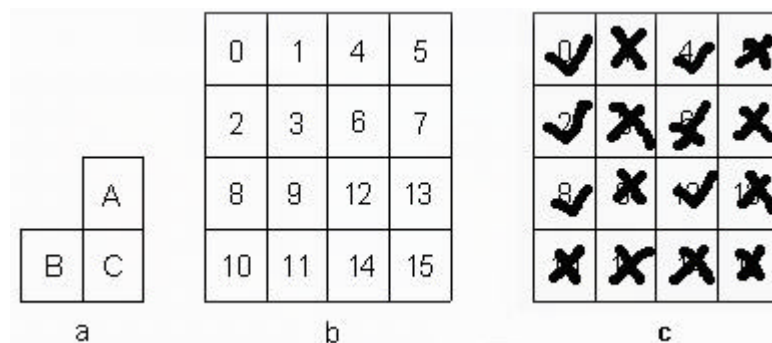


图 3—8 可以使用 17 个像素值的块

Fig.3.8 Number of usefull pixel block

问题二：由于块预测和相邻的四个块有强烈的依赖关系，只能顺序预测的，不可以并行执行，这样的结构对算法优化不利。

问题三：在表 3—1 中发现，H.264 的色度预测相比于亮度预测并没有性能上突出的表现。即在 H.263 中，色度的信噪比要比亮度信噪比相对于 H.264 要来的高。这说明色度的预测不是很突出，因此色度不必使用很复杂的预测模式。

3.7.2 图像数据分析

首先，对要预测的图像块数据做一个分析，把 Carphone 序列中的第 37、69 个宏块取出，打印出它的亮度和两个色度值，如图 3—9 所示。

从图中的数据可以看出 4X4 的块和周围 E-H、M-P 这 8 个数据的依赖关系并不是很紧密，有的地方相关，有的地方根本就没有联系，这是分块处理必然的结果，分块是不可能

率时有好的性能，接近甚至超过 JPEG-2000，但在大图和低量化时，还是 JPEG-2000 的性能要好。

3.7 对于帧内编码改进的几点建议

H.264的帧内预测技术对于提高图像的质量、减小码率有很大的作用。但是，它的亮度和色度的预测模式很多，编码器要在这些模式中选择一个最佳的模式，需要进行大量的计算。因此，帧内预测模式的简化是本文研究的内容之一。

3.7.1 问题分析

在 4X4 的块预测中，用到了 A-Q，共 17 个像素点的值，如图 3—2 所示。这样可以精确预测块的值，但是也带来一些问题。

问题一：4X4 的块用到 17 个像素的值的比率不大，按照图 3—8 (b) 的扫描顺序，只占 5 / 16，如图 3—8 (c) 中打对号的块，这样造成 9 种预测模式效率并不是很高。

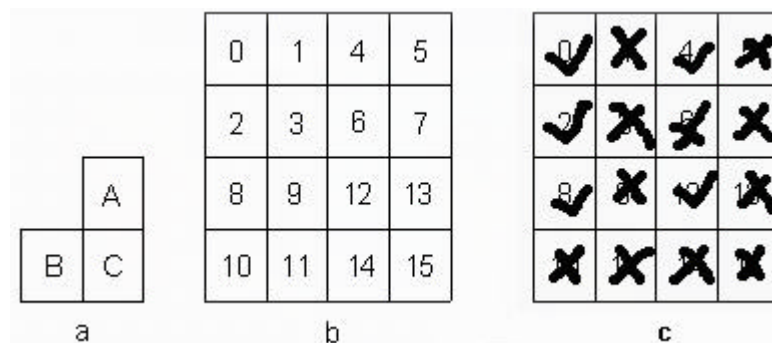


图 3—8 可以使用 17 个像素值的块

Fig.3.8 Number of usefull pixel block

问题二：由于块预测和相邻的四个块有强烈的依赖关系，只能顺序预测的，不可以并行执行，这样的结构对算法优化不利。

问题三：在表 3—1 中发现，H.264 的色度预测相比于亮度预测并没有性能上突出的表现。即在 H.263 中，色度的信噪比要比亮度信噪比相对于 H.264 要来的高。这说明色度的预测不是很突出，因此色度不必使用很复杂的预测模式。

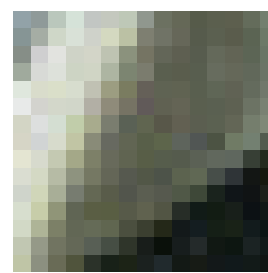
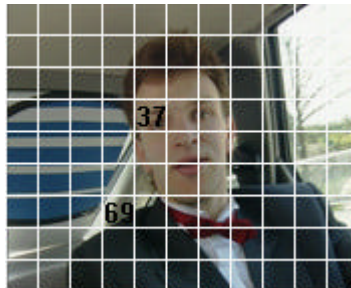
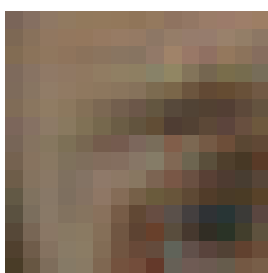
3.7.2 图像数据分析

首先，对要预测的图像块数据做一个分析，把 Carphone 序列中的第 37、69 个宏块取出，打印出它的亮度和两个色度值，如图 3—9 所示。

从图中的数据可以看出 4X4 的块和周围 E-H、M-P 这 8 个数据的依赖关系并不是很紧密，有的地方相关，有的地方根本就没有联系，这是分块处理必然的结果，分块是不可能

考虑到图像中具体目标的形状，造成分块后块的相关性不确定。

另一方面，可以观察到色度块并不象亮度块那样变化显著，色度数值基本上是平缓变化的，这也是 H.264 的色度预测不像亮度预测那样效果显著的原因。这里只列出了一幅图中一个宏块的数据，但是因为帧内编码，一个数据就具有代表性，只不过是有的宏块相关性高些，有的相关性低些的区别，从整体宏块的分布上是无规律可寻的（特殊的序列除外，如只是人肩像，背景不变时）。



第 37 块放大图

第 69 块放大图

57 101 115 119	121 121 125 126	125 125 124 122	120 119 118 117
50 66 106 116 120	124 123 124 125	125 123 123 120	121 124 122 122
54 74 109 116 121	122 124 124 126	126 124 126 124	126 126 128 127
57 80 112 116 121	124 125 124 123	124 126 127 129	131 129 130 126
53 78 114 118 123	125 125 124 123	125 128 130 130	128 125 120 117
49 74 114 121 124	124 124 124 124	124 127 127 117	105 101 107 103
51 77 112 122 124	125 123 119 119	117 114 105 93	86 90 93 91
52 79 111 122 124	126 119 108 106	99 88 81 75	72 73 83 88
55 81 110 121 125	123 106 106 97	88 83 75 76	81 84 90 95
62 79 109 120 124	115 106 100 99	108 112 110 107	99 90 84 86
64 74 104 121 123	110 113 116 121	122 117 112 104	80 64 65 74
61 68 97 122 119	117 123 124 122	115 110 98 83	82 80 76 73
55 62 91 121 119	119 122 120 115	105 92 92 94	68 55 62 70
53 58 85 117 118	120 121 115 106	83 91 85 55	68 63 69 92
59 57 80 108 120	121 123 114 94	77 86 73 71	62 56 67 89
64 55 75 100 121	123 122 113 90	88 75 109 96	69 76 83 90
92 56 70 89 118	123 123 113 101	87 85 92 102	97 97 96 98

77 109 148 140	178 206 198 193	178 147 117 101	95 94 97 105
78 85 127 151 143	209 200 197 189	166 132 108 98	95 95 100 107
73 99 152 139 188	207 199 195 181	149 118 101 95	94 97 103 107
77 127 150 154 210	202 199 192 166	132 109 97 94	96 101 104 107
94 150 135 198 207	202 199 185 149	118 103 97 95	98 101 103 106
124 142 164 213 204	202 196 170 132	109 98 96 96	100 100 104 112
147 135 207 208 204	201 188 150 119	104 96 96 98	99 101 105 121
133 174 213 205 203	200 174 132 111	99 96 97 96	98 101 109 129
141 212 207 204 203	194 154 122 104	96 95 96 95	97 101 115 134
189 212 205 204 202	178 136 113 99	94 95 94 95	97 104 121 104
214 207 205 202 196	156 124 105 97	94 93 93 92	98 106 75 38
209 206 205 203 181	138 113 99 94	93 91 92 96	91 56 35 37
206 205 203 198 162	123 102 96 93	91 89 91 77	44 35 36 36
205 205 203 187 140	111 99 95 92	91 83 57 39	39 38 37 35
205 203 200 167 123	103 97 95 90	71 45 38 39	38 38 37 35
204 202 191 146 115	101 97 80 56	42 41 42 39	38 37 37 36
203 201 178 134 110	85 57 42 42	42 41 40 38	37 37 38 34

第 37 块亮度数据

第 69 块亮度数据

138 141 142 143	143 142 143 143
133 138 140 141 143	143 143 143 143
134 139 141 142 142	141 141 142 140
134 138 140 141 139	138 139 139 139
131 138 139 138 137	137 137 138 139
134 137 138 137 139	138 139 139 140
132 136 137 137 139	140 140 137 138
132 134 137 138 141	141 133 127 133
128 135 137 137 142	142 136 136 141

136 126 127 123	120 120 121 122
145 134 126 124 122	120 121 122 121
143 127 126 123 119	121 122 122 122
134 126 124 121 119	122 122 122 122
126 125 124 118 120	122 121 122 122
126 124 122 118 121	121 123 123 123
124 123 118 119	121 122 123 125 127
122 121 117 120	122 127 127 128
123 119 118 123 125	128 128 128 128

第 37 块 V 数据

第 69 块 U 数据

图 3—9 carphone 中宏块数据分析

Fig.3.9 Analysis of Macroblock

3.7.3 建议的解决方案

结合以上的结论，针对这 3.7.2 中的三个问题，本文提出了相应的简化方案：

1、4X4 亮度预测从 17 个相邻像素简化为 9 个相邻像素，如图 3—10 所示。

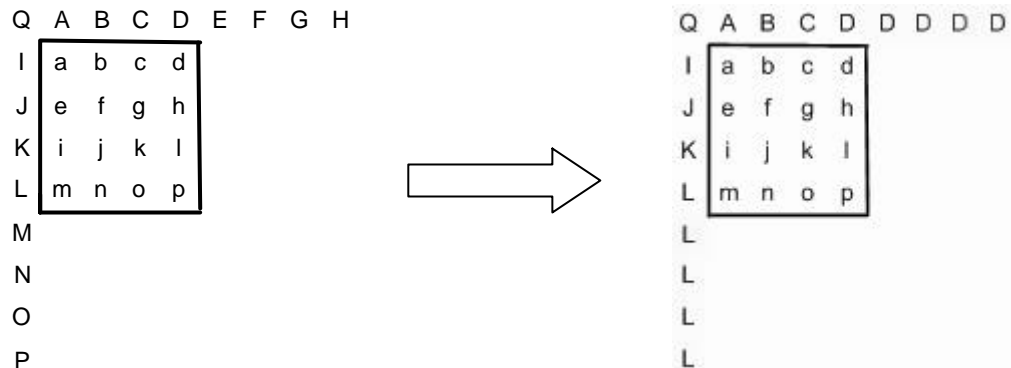


图 3—10 建议简化的方法

Fig.3.10 The suggested simplified method

用 D、L 代替原来的 E-H、M-P 的八个像素值，这样可以减少一半的计算复杂度，并且可以并行处理，如图 3—11 所示。图中标号相同的块可以并行处理，一个宏块共需要 9 次预测。

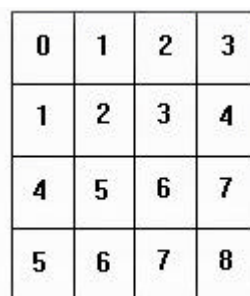


图 3—11 并行预测的块

Fig.3.11 Pipeline Issue for Intra Prediction

用 carphone、claire、grandma 等 7 种 QCIF 序列对简化的预测方法进行测试，测试的条件为：在相同的量化值下，看最后节约的比特数百分比，结果如表 3—2 所示。从表中可以看出，17 个相邻像素的预测要比 9 点预测使用的比特数要少，这是符合实际情况的，因为预测点多可以更精确的预测。但是对比于减少的计算复杂度来说，9 点预测带来的简化和并行处理还是有它的优势，因此，9 点预测可以作为一种附加选项。

表 3—2 7 个测试序列的对比结果

Chart 3.2 Comparison of seven sequence

	Qp	carphoner	Claire	grandma	Mthr_dotr	salesman	suzie	trevor	Average
CABAC	Low	0.31%	0.06%	0.22%	0.18%	0.10%	0.20%	0.45%	0.22%
	High	0.52%	0.23%	0.81%	0.54%	0.45%	0.90%	1.86%	0.62%
UVLC	Low	0.36%	0.10%	0.31%	0.20%	0.19%	0.44%	0.51%	0.30%
	High	0.58%	0.33%	0.83%	0.63%	0.83%	1.30%	2.03%	0.65%

2、在 3.7.1 中指出 8X8 色度预测的模式 3 预测效率不高，主要的原因是物体的色度是物体的本身属性，它的值变化不像亮度那样随着光线的变化而改变，所以本文提出一种简化的模式 3 预测方法。下面是具体的过程：

a、计算水平和垂直梯度

$$H = |P(0,-1) + P(1,-1) - P(6,-1) - P(7,-1)|$$

$$V = |P(-1,0) + P(-1,1) - P(-1,6) - P(-1,7)|, \quad \alpha = H/V$$

b、若 $\alpha > \beta$ ，用水平块预测方式

$$\text{Pred}(i,j) = P(i,-1) + P(-1,j) - P(-1,0), \quad i=0, \dots, 7, j=0, \dots, 3$$

$$\text{Pred}(i,j) = P(-1,j) + P(i,-1) - P(0,-1), \quad i=0, \dots, 7, j=4, \dots, 7$$

c、若 $\alpha \leq \beta$ ，用垂直块预测方式

$$\text{Pred}(i,j) = P(-1,j) + P(i,-1) - P(0,-1), \quad i=0, \dots, 3, j=0, \dots, 7$$

$$\text{Pred}(i,j) = P(i,-1) + P(-1,j) - P(-1,0), \quad i=4, \dots, 7, j=0, \dots, 7.$$

具体的过程如图 3—12 所示，图中的数字下标如 1、2、3 是相应的象素加上对应位置的增量后的值，在试验中 β 取 1。

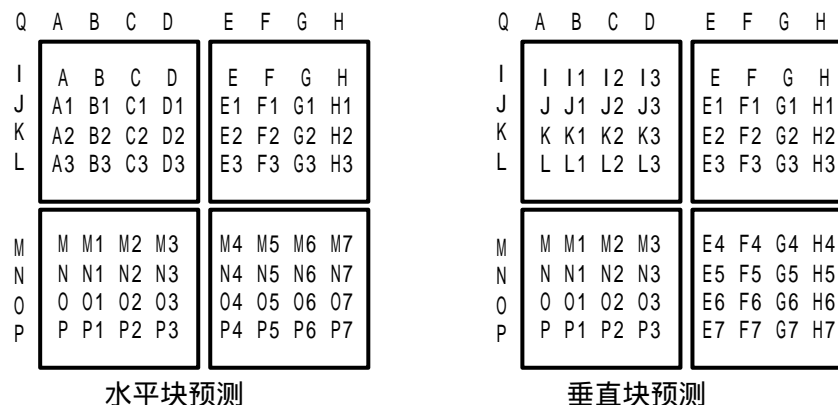


图 3—12 简化的模式 3 预测方式

Fig.3.12 The simplified of predecited Mode 3

表3—3 简化模式3的测试结果

Chart 3.3 The test result of new mode 3

		Qp	carphoner	Claire	Grandma	Mthr_dot r	salesman	suzie	Trevor	Average
U	CABAC	High	-1.32	-2.44	-0.25	-1.11	-1.32	-1.76	-0.04	-1.17
		Low	-1.54	-2.21	-0.34	-1.98	-2.01	-2.42	-0.45	-1.56
V	CABAC	High	-2.30	-3.02	-0.55	-2.21	-1.52	-3.12	-0.36	-1.87
		Low	-2.45	-3.71	-0.63	-2.82	-1.87	-3.56	-0.44	-2.21

表3—3中是色度预测简化后的模式3和当前JVT使用的测试模型JM—4.1的模式节约的比特数百分比，符号表示简化的模式比JM-4.1少用比特数。

3.8 小结

本章首先详细的说明了 H.264 的 4X4 和 16X16 的亮度块到 8X8 的色度块的帧内预测技术。对其中的重点和新颖点都以详细的图表力图使得条理清楚。然后对比了 H.264 和 H.263 的帧内编码效率以及 H.264 和 JPEG-2000 的编码效率，从试验结果上可以看出，帧内编码无论是在信噪比和节约的比特数上都较 H.263 和 JPEG 有很大的进步，接近 JPEG-2000 的效果。

接下来的章节中分析了 H.264 帧内编码存在的问题，并提出了二项相应的建议，从试验结果看，新的建议以极小的信噪比的代价，在计算复杂度和并行处理角度都有性能上的改善，因此，简化的模式可以作为帧内编码的选择项来使用。

总之，帧内编码技术是 H.264 的一大特色，它不仅仅可以使用在视频编码中，相信静止图像编码、数码相机、传真机等产品中也可以使用。

本章参考:

- [1] Iain E.G Richardson, "Prediction of Intra Macroblocks," video & image compression resources and research, October 17, 2002
- [2] Joint Final Committee Draft (JFCD) of Joint Video Specification, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Document JVT-D157, Jul. 2002.
- [3] Till Halbach, "Performance comparison: H.26L intra coding vs. JPEG2000", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)4th Meeting: Klagenfurt, Austria, JVT-D039, 22-26 July, 2002
- [4] Greg Conklin, "Low Complexity Diagonal Mode for Intra Prediction", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6) 4th Meeting: Klagenfurt, Austria, JVT-D059, 22-26 July, 2002
- [5] Minhua Zhou, "Intra Prediction with Simplified Prediction Modes", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)3rd Meeting: Fairfax, Virginia, USA, JVT-D026, 6-10 March, 2002

第四章 整数变换与熵编码

4.1 DCT 变换介绍

离散余弦变换 (DCT) 是当前视频编码的核心技术之一, 它是对图像进行分块, 然后按块进行能量集中的方法, 因为它的性能接近 K-L 变换, 而且要简单易与实现, 因此被广泛的应用在 JPEG 和 MPEG 等图像压缩算法中。

因为图像是二维的, 所以在图像压缩算法中理应使用二维 DCT 变换, 但是二维 DCT 变换复杂而且不易于实现, 所以一般都是使用一维 DCT 变换, 对图像的行和列分别进行 DCT 变换, 取得和二维 DCT 变换相近的结果。

一维 DCT 变换的基本概念是把 N 维向量 x 通过变换矩阵 H 变换成新的 N 维向量 y , 它的公式如下:

$$y(k) = \sqrt{\frac{2}{N}} c(k) \sum_{n=0}^{N-1} x(n) \cos \left((2n+1) \frac{k\pi}{2N} \right) \quad k=0,1,2,\dots,N-1 \quad (4.1-1)$$

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

写成矩阵的形式为:

$$y = Hx, \quad H_{kn} = H(k,n) = c(k) \sqrt{\frac{2}{N}} \cos \left((2n+1) \frac{k\pi}{2N} \right) \quad (4.1-2)$$

可以证明变换矩阵 H 是正交的, 但不是对称阵, 反余弦变换 IDCT 根据 H 正交性有

$$H^{-1} = H^T$$

所以 IDCT 也可以用矩阵形式表示为:

$$x = H^{-1}y = H^T y \quad (4.1-2)$$

从以上的公式中可以看出矩阵 H 的元素不是有理数, 对于输入的整数图像数据来说经过 DCT 变换之后变成了无理数, 这对于数字计算机来说是不利的, 因为即使直接 DCT 变换再反变换, 也不一定得到和原始的数据精确相等的值, 特别是在位长比较低的情况下。这在 MPEG 和 H.263 等标准中有明显的体现, 编码端和解码端的数据总是有细微的不同。

4.2 整数 DCT 变换和量化

4.2.1 整数 DCT 变换

MPEG 和 H.263 等压缩算法在计算 DCT 变换时, 针对变换系数为无理数的情况, 通常都是把系数扩大一定的倍数, 然后取整进行计算。H.264 继承了 DCT 算法的优越性, 同时也加入了改进, 它不再是把系数同时扩大一定的倍数, 而是对系数进行取整, 然后再进行变换, 这样会损失一定的能量集中性能, 试验证明, 变换效率的损失只在 0.02DB。但是因为整数 DCT 的系数相差不是很大, 同时由于它的块只有 4X4 的大小, 可以相对精确的再现原始的数据, 并且实现起来更简单, 所以这点损失和带来的性能改进是很小的。

在 H.264 中最小的块为 4X4 大小的, 所以按照公式(4.1-2)计算出的变换矩阵^[1]为:

$$Y = HXH^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \quad (4.2.1-1)$$

$$a = 1/2, b = \sqrt{1/2} \cos(p/8), c = \sqrt{1/2} \cos(3p/8)$$

矩阵 H 是变换矩阵, 从公式 (4.2-1) 中可以看出, b、c 仍为无理数, 这里和传统的 DCT 变换一样, 仍需要把他们扩大一定的倍数, 但是 H.264 在扩大之后, 还要取近似的值。下面列出两种扩大系数:

$$H_1 = \text{round}(26H) = \begin{bmatrix} 13 & 13 & 13 & 13 \\ 17 & 7 & -7 & -17 \\ 13 & -13 & -13 & 13 \\ 7 & -17 & 17 & -7 \end{bmatrix} \quad H_2 = \text{round}(2H) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

H_1 ^[2]和 H_2 分别扩大了 26 和 2 倍, H_1 是在 H.264 早期版本中的变换矩阵, H_2 是 Hadamard 矩阵。可以算出 H_1 和 H_2 是正交阵。在 [1~100] 的范围内, 只有 2 和 26 这两个系数可以满足正交性。

H_2 显然是不适合用于 DCT 变换的, 所以在早期的 H.264 版本中使用的是 H_1 。一维 DCT 变换为 $y = H_1 x$, 一维反变换为 $x' = H_1^T y$ 。这样经过一次的正、反变换后所有的系数扩大了 676 倍, 即 $X' = 676X$ 。

在 H.264 中, DCT 变换和量化、反量化和反 DCT 变换是紧密结合在一起的, 所以下面简单介绍量化策略。

4.2.2 量化、反量化

H.264 中的量化是紧密地和 DCT 变换结合在一起的, 下面的章节中给出结合原因的详细说明, 此节中只给出量化、反量化的具体实现过程^[3]。

在 JM-1.4 的校验模型以前, QP 的量化阶数有 32 级 0~31, 这和 H.263 中的量化步长是相似的。这里以亮度系数为例:

QP_{luma} 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

此外, 还引入了量化数组 A 和反量化数组 B, 分别为:

A(QP=0,...,31) =

{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,...,31) =

{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}

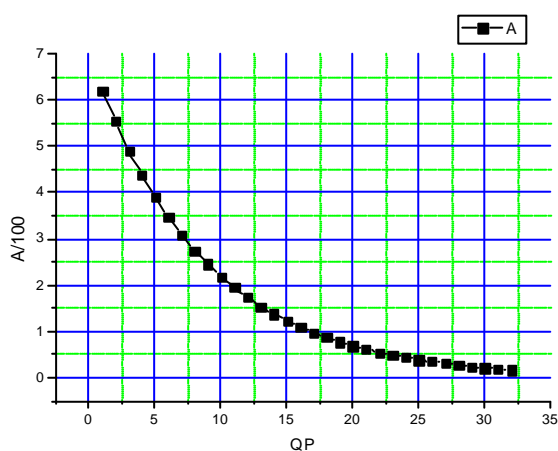


图 4—1 量化数组 A 的曲线图

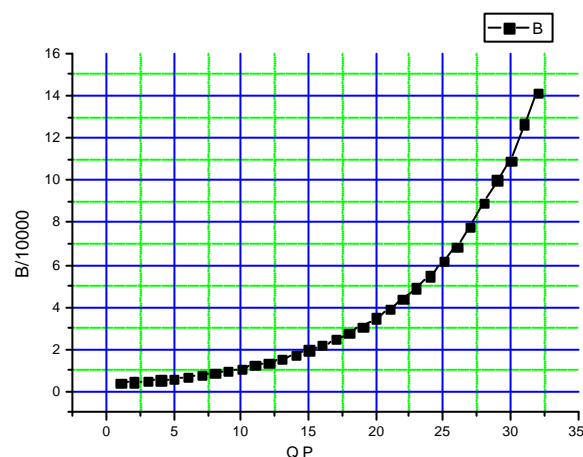


图 4—2 反量化数组 B 的曲线图

Fig.4.1 relation between QP and quant matrix A Fig.4.2 relation between QP and dequant matrix B

在图 4—1 和图 4—2 中画出了 A 和 B 的曲线图, 可以看出 A 和 B 是经过设计的, 满足对数曲线的特性。

此外, 还有以下特性:

$$2B(QP) \cong B(QP+6) \quad (4.2.2-1)$$

$$A(QP) \times B(QP) \times 676^2 \cong 2^{40} \quad (4.2.2-2)$$

在上一节的内容中，我们知道一次 DCT 变换后，系数扩大了 676 倍，公式 4.2.2-2 正好解决了这个问题，原始的象素值经过 DCT 变换、量化、反量化、IDCT 后，只要进行移位操作，不需要除法，这样就提高了执行的速度。

假设经过 DCT 变换后的系数为 K，量化后的系数为 LEVEL：

$$\text{LEVEL} = (\text{K} \times \text{A}(\text{QP}) + \text{fx}2^{20})/2^{20} \quad (4.2.2-3) \quad f = \begin{cases} 1/3 & \text{intra} \\ 1/6 & \text{inter} \end{cases} \quad f \text{ 符号和 K 的相同}$$

反量化后得到系数 K：

$$\text{K}' = \text{LEVEL} \times \text{B}(\text{QP}) \quad (4.2.2-4)$$

系数 K 再经过 IDCT 变换，再右移 20 位，就得到了重建的象素值。

从公式 4.2.2-3 和 DCT 变换扩大 676 倍后可以得出，对于真正的 4X4DCT 的量化系数为：

$2^{20} / (676 \times \text{A}(\text{QP}))$ ，即

$\text{DCTQ}(\text{QP}=0, \dots, 31) =$

[2.5019, 2.8050, 3.1527, 3.5334, 3.9671, 4.4573, 5.0037, 5.6201, 6.3055, 7.0829, 7.9546, 8.9146, 10.0074, 11.2402, 12.6110, 14.1013, 15.8280, 17.8293, 19.8865, 22.4804, 25.0185, 28.2027, 31.6561, 35.2534, 39.7730, 44.3185, 50.0370, 57.4499, 64.6312, 70.5067, 81.6394, 91.2440]

可以看出当前的 QP 和下一个 QP 之间是以约 12% 的增量递增的。

注：4.2.2 节中的量化、反量化方法在 JM-1.4 以后的版本中已经被取消，新的方法是在此基础上做了改进。

4.2.3 新型整数变换

上面的两节中介绍了 DCT 变换和 JM-1.4 以前版本中的整数 DCT 变换的基本概念和具体的实现矩阵。下面的内容介绍 JM-1.4 以后版本中的变换方法。

在公式 (4.2.1-1) 中给出了 4X4 大小的 DCT 变换矩阵：

$$Y = H X H^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

这个矩阵可以进一步写为：

$$Y = BCXC^T B = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & b \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & b \end{bmatrix}$$

其中 $d = c/b$ ，B 矩阵是对角阵，所以可以改写为：

$$Y = (CXC^T) \otimes Q = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

⊗ 符号表示的是矩阵中的对应位置上的元素相乘。这样由于矩阵 Q 只是经过变换后的各个元素的尺度伸缩，所以可以在编码器端结合量化表，解码器端结合反量化表来一次性的完成变换和量化。由于在矩阵 Q 中有三个不同的元素 a^2 、 ab 、 b^2 ，所以需要三个不同的表来进行量化、反量化。

这里系数 $d = c/b = \sqrt{2} - 1$ ，这个数值不是很理想的值，不能直接用定点二进制的方法表示和计算，所以需要选择一个计算方便的值来代替它，令 $d = 1/2$ ，这样取值还是可以得到很高的变换效率。

系数 d 是人为选择的，所以为了保证公式 (4.2.1-1) 中矩阵 H 为正交阵，即满足 $H^T H = I$ ，需要重新计算 a 、 b 、 c 的数值。

$$H^T H = \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} = \begin{bmatrix} 2a^2 + b^2 + c^2 & 0 & 0 & 2a^2 - b^2 - c^2 \\ 0 & 2a^2 + b^2 + c^2 & 2a^2 - b^2 - c^2 & 0 \\ 0 & 2a^2 - b^2 - c^2 & 2a^2 + b^2 + c^2 & 0 \\ 2a^2 - b^2 - c^2 & 0 & 0 & 2a^2 + b^2 + c^2 \end{bmatrix}$$

$$\text{所以} \begin{cases} 2a^2 + b^2 + c^2 = 1 \\ 2a^2 - b^2 - c^2 = 0 \\ d = c/b \\ d = 1/2 \end{cases}, \quad b = \sqrt{\frac{1}{2(1+d^2)}} = \sqrt{2/5}, \quad c = \sqrt{1/10}, \quad a = 1/2$$

这样得到的变换和量化的公式为:

$$Y = \begin{pmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \end{pmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

对于前向的变换, 如果系数经过除 2 后, 会造成截尾误差, 所以要把变换矩阵中的系数第 2、4 行, 转秩阵中的第 2、4 列的系数扩大 2 倍, 即:

$$Y = (C_f X C_f^T) \otimes Q_{forw} = \begin{pmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \end{pmatrix} \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \quad (4.2.3-1)$$

对于反向的变换, 因为经过量化、反量化, 截尾对于象素值的影响很小, 所以可以直接使用, 不需要扩大系数。

$$X = C_f (Y \otimes Q_{inv}) C_f^T = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix} \quad (4.2.3-2)$$

可以进一步的看出矩阵 C_f 是公式 (4.2.1-1) 中 H 的另一种系数扩大取整后的变形矩阵 $C_f = \text{round}(2.5H)$, 前面指出在 $[1 \sim 100]$ 的范围内, 只有 2 和 26 这两个系数可以满足正交性, 所以矩阵 C_f 、 C_i 的行向量、列向量的模是不一样的, 和系数扩大 26 倍时的模为 676 不同。所以, 这里量化系数对变换后得到的 4X4 矩阵中的每个元素都是不同的。

在上一节指出, $2B(QP) \cong B(QP+6)$, 即量化的系数增加 6 时变为原来的两倍。所以, 只需要保存量化系数为 $[0, 5]$ 区间范围内的值, 通过循环、乘 2 就可以得到 $[6,$

51] 区间内的值 (在 JM-1.4 以后的版本中, 量化变得更精细, 所以 QP 值的范围变成 [0, 51])。下面给出新的量化矩阵和量化的方法。

4.2.4 JM-1.4 版本后的量化、反量化方法

在 JM-1.4 的版本中, 量化值 QP 的范围为[0,51], 这么大的一个量化范围为码率和质量提供了一个很好的权衡。量化的基本公式为: $Z_{ij} = \text{round}(Y_{ij}/Q_{\text{step}})$, 表 4-1 列出了对应于 QP 的 Q_{step} 。

表 4-1 对应于 Qp 的 Qstep 值

Table.4.1 The relation between Qp and Qstep

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	...	18	...	24	...	30	51
Qstep	2.25	2.5	...	5	...	10	...	20	224

但是, 由公式 4.2.3-1 可知, 变换矩阵各行向量的模不相等, 所以对变换后矩阵中的每个系数应扩大相应的倍数来使得最终的模相等, 结合量化的 Q_{step} , 因此引入了量化矩阵 (这里的量化矩阵和通常意义上的量化矩阵不是同一个概念)。最终的量化公式为:

$$Z_{ij} = \text{round}(Y_{ij} \frac{PF}{Q_{\text{step}}}) \quad (4.2.4-1)$$

PF 对应于 4.2.3-1 中的 Q_{forw} , Y_{ij} 对应于 4.2.3-1 中的矩阵 C_i 变换后的系数, Z_{ij} 为量化后的矩阵。在 H.264 中用移位操作代替除法运算, 同时为了避免小数运算, 所有的系数扩大 2^{15} 倍。

$$Z_{ij} = (Y_{ij} * MF + f) >> \text{qbits} \quad (4.2.4-2)$$

$$\text{其中 } \frac{MF}{2^{\text{qbits}}} = \frac{PF}{Q_{\text{step}}}, \text{qbits} = 15 + \text{floor}(QP/6), \text{INTRA 帧时 } f \text{ 为 } 2^{\text{qbits}}/3, \text{INTER}$$

帧时为 $2^{\text{qbits}}/6$ 。

表 4-2 列出了量化系数从 0~5 的各个系数量化值:

表 4-2 QP 从 0~5 的量化值

table.4.2 a^2 , $ab/2$, $b^2/4$ relevant to QP

QP	0	1	2	3	4	5
a^2	13107	11916	10082	9364	8192	7282
$ab/2$	8066	7490	6554	5825	5243	4559
$b^2/4$	5243	4660	4194	3647	3355	2893

反量化的原理和量化一样, 这里只给出相应的公式和反量化矩阵的系数值。表 4-3

51] 区间内的值 (在 JM-1.4 以后的版本中, 量化变得更精细, 所以 QP 值的范围变成 [0, 51])。下面给出新的量化矩阵和量化的方法。

4.2.4 JM-1.4 版本后的量化、反量化方法

在 JM-1.4 的版本中, 量化值 QP 的范围为[0,51], 这么大的一个量化范围为码率和质量提供了一个很好的权衡。量化的基本公式为: $Z_{ij} = \text{round}(Y_{ij}/Q_{\text{step}})$, 表 4-1 列出了对应于 QP 的 Q_{step} 。

表 4-1 对应于 Qp 的 Qstep 值

Table.4.1 The relation between Qp and Qstep

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	...	18	...	24	...	30	51
Qstep	2.25	2.5	...	5	...	10	...	20	224

但是, 由公式 4.2.3-1 可知, 变换矩阵各行向量的模不相等, 所以对变换后矩阵中的每个系数应扩大相应的倍数来使得最终的模相等, 结合量化的 Q_{step} , 因此引入了量化矩阵 (这里的量化矩阵和通常意义上的量化矩阵不是同一个概念)。最终的量化公式为:

$$Z_{ij} = \text{round}(Y_{ij} \frac{PF}{Q_{\text{step}}}) \quad (4.2.4-1)$$

PF 对应于 4.2.3-1 中的 Q_{forw} , Y_{ij} 对应于 4.2.3-1 中的矩阵 C_i 变换后的系数, Z_{ij} 为量化后的矩阵。在 H.264 中用移位操作代替除法运算, 同时为了避免小数运算, 所有的系数扩大 2^{15} 倍。

$$Z_{ij} = (Y_{ij} * MF + f) >> \text{qbits} \quad (4.2.4-2)$$

$$\text{其中 } \frac{MF}{2^{\text{qbits}}} = \frac{PF}{Q_{\text{step}}}, \text{qbits} = 15 + \text{floor}(QP/6), \text{INTRA 帧时 } f \text{ 为 } 2^{\text{qbits}}/3, \text{INTER}$$

帧时为 $2^{\text{qbits}}/6$ 。

表 4-2 列出了量化系数从 0~5 的各个系数量化值:

表 4-2 QP 从 0~5 的量化值

table.4.2 a^2 , $ab/2$, $b^2/4$ relevant to QP

QP	0	1	2	3	4	5
a^2	13107	11916	10082	9364	8192	7282
$ab/2$	8066	7490	6554	5825	5243	4559
$b^2/4$	5243	4660	4194	3647	3355	2893

反量化的原理和量化一样, 这里只给出相应的公式和反量化矩阵的系数值。表 4-3

列出了量化系数从 0~5 的各个系数反量化值:

表 4-3 QP 从 0~5 的反量化值
table.4.3 a^2 , ab , b^2 relevant to QP

QP	0	1	2	3	4	5
a^2	10	11	13	14	16	18
ab	13	14	16	18	20	23
b^2	16	18	20	23	25	29

假设用公式 (4.2.3-1) 整数变换后得到的矩阵是 Y , 公式 (4.2.4-1) 给出了 JM-4 的量化公式:

$$Y_Q(i, j) = \left[Y(i, j) \cdot Q(QP \% 6, i, j) + f \right] / 2^{15+QP/6} \quad i, j = 0, \dots, 3 \quad f = \begin{cases} \frac{1}{3}(2^{15} + QP/6) & \text{intra} \\ \frac{1}{6}(2^{15} + QP/6) & \text{inter} \end{cases} \quad (4.2.4-1)$$

Y_Q 是量化后的矩阵, Q 是量化系数(表 4-1), 帧内时 $f=2^{15+QP/6}/3$, 帧间时 $f=2^{17+QP/6}/6$, f 和被量化的 Y 有同样的符号。可以看出, 中间结果是 32bit 的, 但是最后得到的 Y_Q 仍是 16bit 的。

公式 (4.2.4-2) 给出了 JM-4 的反量化公式:

$$Y(i, j) = Y_Q(i, j) \cdot R(QP \% 6, i, j) \ll QP/6, \quad i, j = 0, \dots, 3 \quad (4.2.4-2)$$

Y_Q 是量化后的矩阵, R 是反量化系数(表 4-2)。

接着用公式 (4.2.3-2) 整数反变换后得到的矩阵是 X' , 因为所有的系数都扩大了 64 倍, 所以变换之后再右移 6 位。

4.3 自适应的块变换 (ABT)

前面 H.264 中采用了 16X16、16X8、8X16、8X8、8X4、4X8、4X4 共 7 种不同的宏块类型, 编码器必需对宏块的类型编码, 解码器才能选择合适的块进行宏块运动估计、补偿等操作, 相对于 H.263 中的固定 8X8 大小的块来说, 这些宏块类型码是多余的。另一方面, 不同大小的块的图像分布特性是不同的, 硬要把不同大小的块拆成 4X4 进行变换也不是最佳的选择。因此, 针对这两个问题, H.264 中引入了自适应的块变换 ABT [5]

(Adaptive Block Transforms), ABT 的基本思想是块变换不限定为 4X4, 而是根据块的

具体的大小来选择变换矩阵，目前的版本支持 4X4、4X8、8X4、8X8 的块变换，这样变换矩阵的大小就可以表示宏块类型，不必额外编码，从而节约了比特数。但是，由于 Intra 帧预测是按 4X4 块来进行的，所以对于 ABT 必需引入新的帧内预测方法。下面两节将对 ABT 变换和 ABT 帧内预测进行说明。

4.3.1 ABT 变换

假设矩阵 $X_{m,n}$ 是原始的系数矩阵， T_m 是列变换矩阵， T_n 是行变换矩阵，则 ABT 变换的矩阵表达式为：

$$Y_{m,n} = T_m * X_{m,n} * T_n^T, \quad (4.3.1-1)$$

m, n 取值为 4 或 8，相应的矩阵如图 4—3 所示：

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad T_8 = \begin{bmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{bmatrix}$$

图 4—3 T_4 和 T_8 变换矩阵

Fig.4.3 transform matrix of T_4 and T_8

T_4 变换矩阵和 4X4 的块变换矩阵相同，由于不是正交的矩阵，所以和 4X4 块变换一样，也引入了量化矩阵，并且使用同样的量化方法。 T_8 [6] 阵是正交的，所以不需要特别引入量化矩阵，只要对所有的系数统一的量化就可以了，并且 T_8 性能和 DCT 相当，只是在 4、6 行略有不同，可以使用经典的 DCT 算法，如 LLM，AAN 等。

由公式 4.3.1-1 可以得出 ABT 的反变换公式 4.3.1-2：

$$X_{m,n} = T_m^T * Y_{m,n} * T_n, \quad (4.3.1-2)$$

同正变换一样，当块的大小为 4X4 时，反变换仍先使用 4X4 变换时的反量化方法，然后再进行矩阵变换。 T_8 时直接量化，然后反变换就可以了。

4.3.2 ABT 预测

对于帧内的宏块，ABT 方式编码时有不同于 4X4 块的块预测。假设预测的块为 $P[m,n]$ ， $0 \leq m \leq M$ ， $0 \leq n \leq N$ ， $M, N = \{4, 8\}$ ，上边相邻宏块的像素值为 T_k ，左边相邻宏块的像素值为 L_k ， $0 \leq k \leq (M+N-1)$ ， q 是左上角的像素值。如图 4—5 所示。

当 4X4 的块时，预测方法和上一章中相同，当块大于 4 个像素时，由于块的扩大，它的边缘可能不是光滑的，如下图所示：



图 4—4 左边为直接预测的块右边为滤波后预测块

Fig.4.4 Left: Mode 1 is applied without prefiltering. Right: Prefiltering is applied first.

这些像素值直接用来预测会影响预测准确性，因此要先对周围的像素滤波^[7]，公式如 4.3.2-1 所示：

$$EP[k] = (EP'[k-1] + 2*EP'[k] + EP'[k+1] + 2) >> 2 \quad (4.3.1-1)$$

EP 中顺时针保存了 $[L_{m+n-1}, \dots, L_0, q, T_0, \dots, T_{m+n-1}]$ 数据。

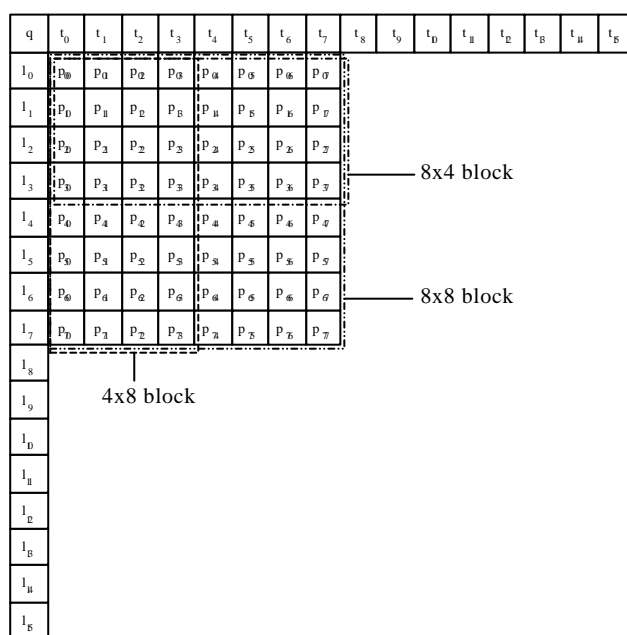


图 4—5 4X8、8X4、8X8 亮度块预测的相邻像素值

Fig.4.5 samples for ABT intra prediction for 4x8, 8x4, and 8x8 luma blocks

从图 4—5 可以看出，相邻的水平垂直像素经过补充到 $M+N$ 个后，就可以使用类似 4X4 块的 9 种预测模式。

4.3.3 ABT 模式相关技术

以上两节中介绍了 ABT 模式的关键技术，这一小节对相关的内容作简单介绍。

ABT 分块为 4X4、4X8、8X4、8X8 的块，其中正方形块的扫描方式为传统的 zigzag 形状，4X8 和 8X4 的扫描方式如图 4-6 所示。



图 4—4 左边为直接预测的块右边为滤波后预测块

Fig.4.4 Left: Mode 1 is applied without prefiltering. Right: Prefiltering is applied first.

这些像素值直接用来预测会影响预测准确性，因此要先对周围的像素滤波^[7]，公式如 4.3.2-1 所示：

$$EP[k] = (EP'[k-1] + 2*EP'[k] + EP'[k+1] + 2) >> 2 \quad (4.3.1-1)$$

EP 中顺时针保存了 $[L_{m+n-1}, \dots, L_0, q, T_0, \dots, T_{m+n-1}]$ 数据。

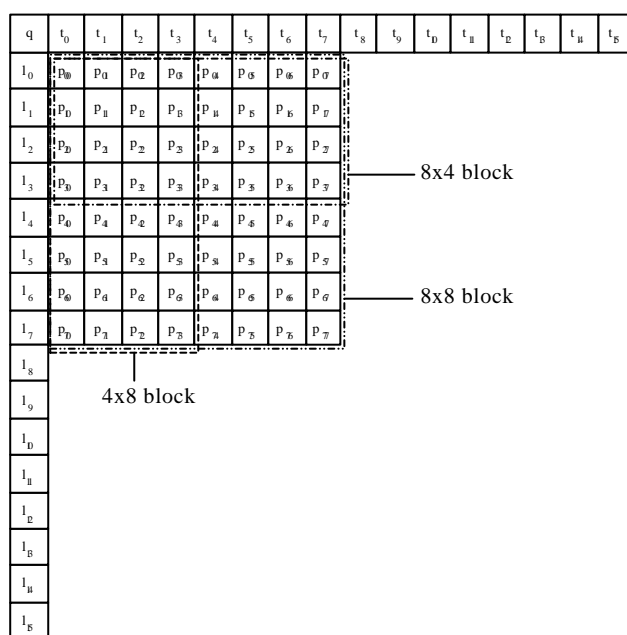


图 4—5 4X8、8X4、8X8 亮度块预测的相邻像素值

Fig.4.5 samples for ABT intra prediction for 4x8, 8x4, and 8x8 luma blocks

从图 4—5 可以看出，相邻的水平垂直像素经过补充到 $M+N$ 个后，就可以使用类似 4X4 块的 9 种预测模式。

4.3.3 ABT 模式相关技术

以上两节中介绍了 ABT 模式的关键技术，这一小节对相关的内容作简单介绍。

ABT 分块为 4X4、4X8、8X4、8X8 的块，其中正方形块的扫描方式为传统的 zigzag 形状，4X8 和 8X4 的扫描方式如图 4-6 所示。

帧内宏块ABT模式判决分为两步，判决的标准依据编码器的性能分为低复杂和高复杂度两种，低复杂度时使用SATD判决，高复杂度使用R-D判决。首先，确定宏块的最优帧内预测模式，对于每一个子块有一个最小的判决代价，并统计出一帧中总的代价。接着，每一个子块的最小代价和一帧的总代价比较，然后决定子块的最终模式。

ABT编码后的系数为 ± 1 的情况很多 [8]，这种系数对于图像质量并无很大的改善，但是它本身和相关占用的比特数却很多，所以ABT引入一种 (Level, Run) 的判决方法，当 Level=1 时，对于不同的Run引入不同的权重因子，如果Run很小，那么权重因子就大，反之就小。然后所有的权重因子之和如果低于一个阈值，就不对此块编码，从而节约比特数，这也是一种意义上的R-D判决。

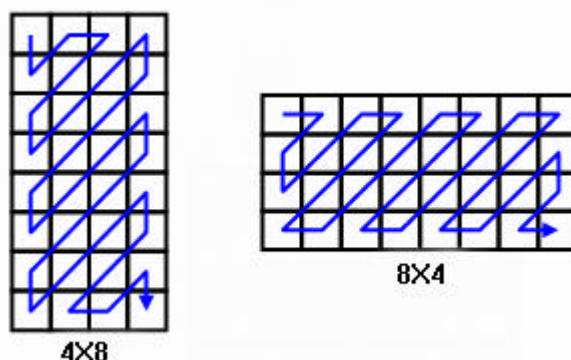


图 4-6 ABT 中 4x8 与 8x4 块的扫描方式
Fig.4.6 Scans for 4x8 and 8x4 blocks

4.4 H.264 中的熵编码

H.264 采用了两种熵编码方法：自适应的算术编码 (CABAC) 和可变长度编码 (VLC)。这两种编码的基本原理在第二章中已经阐述过，CABAC 和 VLC 编码各有优点和缺点，CABAC 的编码增益比 VLC 高，但是它的复杂度也高，并且抗误码的性能不好，与此相反，它的缺点却是 VLC 的优点，反之亦然。因此，H.264 规定在它的 Baseline 档只使用 VLC，在 Main 档加入 CABAC。本文对只对 VLC 编码方案进行详细的讨论。

在 H.264 中，VLC 编码是以 Exp-Golomb (Exponential Golomb Coding) 编码为基础，并辅以 CAVLC (Context-Adaptive Variable Length Coding) 编码的方案。CAVLC 对量化后的系数进行编码，Exp-Golomb 则对除此之外的一些信息 (如：序列头信息、图像头信息、宏块类型、量化参数等) 进行编码。

帧内宏块ABT模式判决分为两步，判决的标准依据编码器的性能分为低复杂和高复杂度两种，低复杂度时使用SATD判决，高复杂度使用R-D判决。首先，确定宏块的最优帧内预测模式，对于每一个子块有一个最小的判决代价，并统计出一帧中总的代价。接着，每一个子块的最小代价和一帧的总代价比较，然后决定子块的最终模式。

ABT编码后的系数为 ± 1 的情况很多 [8]，这种系数对于图像质量并无很大的改善，但是它本身和相关占用的比特数却很多，所以ABT引入一种 (Level, Run) 的判决方法，当 Level=1 时，对于不同的Run引入不同的权重因子，如果Run很小，那么权重因子就大，反之就小。然后所有的权重因子之和如果低于一个阈值，就不对此块编码，从而节约比特数，这也是一种意义上的R-D判决。

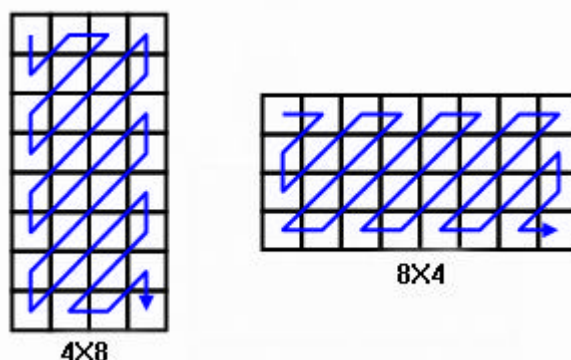


图 4-6 ABT 中 4x8 与 8x4 块的扫描方式
Fig.4.6 Scans for 4x8 and 8x4 blocks

4.4 H.264 中的熵编码

H.264 采用了两种熵编码方法：自适应的算术编码 (CABAC) 和可变长度编码 (VLC)。这两种编码的基本原理在第二章中已经阐述过，CABAC 和 VLC 编码各有优点和缺点，CABAC 的编码增益比 VLC 高，但是它的复杂度也高，并且抗误码的性能不好，与此相反，它的缺点却是 VLC 的优点，反之亦然。因此，H.264 规定在它的 Baseline 档只使用 VLC，在 Main 档加入 CABAC。本文对只对 VLC 编码方案进行详细的讨论。

在 H.264 中，VLC 编码是以 Exp-Golomb (Exponential Golomb Coding) 编码为基础，并辅以 CAVLC (Context-Adaptive Variable Length Coding) 编码的方案。CAVLC 对量化后的系数进行编码，Exp-Golomb 则对除此之外的一些信息 (如：序列头信息、图像头信息、宏块类型、量化参数等) 进行编码。

4.4.1 Exp-Golomb 编码技术

Exp-Golomb 码是 Yuji Itoh 在[9]中提出的 UVLC (Universal Variable Length Coding) 编码方案在阶为 2 时的特例。它有固定的编码结构, 如下:

$$\underbrace{[0, \dots, 0]}_M [1] \underbrace{[INFO]}_M$$

它是由 M 个比特前导的 0, 1 比特的 1 和 M 比特的信息位构成。其中 M 由公式 4.4.1-1 确定:

$$\begin{aligned} M &= \lfloor \log_2(\text{code_num} + 1) \rfloor \\ INFO &= \text{code_num} + 1 - 2^M \end{aligned} \quad (4.4.1-1)$$

code_num 也即所要编的符号值, 符号 0 没有前导和 INFO, 直接编码为 1。下表列出了 Exp-Golomb 码表的前 8 个结构:

表4-4 Exp-Golomb 码表
Fig.4.4 Exp-Golomb code table

Symbol No.	No. of bits	Code
0	1	1
1	3	010
2	3	011
3	5	00100
4	5	00101
5	5	00110
6	5	00111
7	7	0001000
...

可以看出这个码表的优点是结构简单, 无论是软件还是硬件上都易于实现, 更重要的是它不需要以往的编码算法中那样存储码表。

在H.264中需要编码的信息可分为三种类型映射到符号中去。第一种是直接映射, 例如宏块的类型, 参考帧等; 第二种是带符号位的映射, 例如运动矢量的差值, ΔQP 等; 第三种是查表映射, 例如CBP信息等。下表列举这三种映射方法:

表4-5 三种映射到符号的方法

Table 4.5 three symbol mapping methods

符号	直接映射 (4X4 预测类型)	符号映射 (运动矢量)	查表映射 (CBP)
0	模式 0	0	0(所有块系数均为 0)
1	模式 1	1	16 (色度 DC 块)
2	模式 2	-1	1 (第一个亮度块)

3	模式 3	2	2 (第二个亮度块)
4	模式 4	-2	4 (第三个亮度块)
5	模式 5	3	8 (第四个亮度块)
6	模式 6	-3	32 (色度 DC、AC 块)
7	模式 7	...	7 (第一、二个亮度块)
...

每一种映射方法都是经过调整后按照出现的概率重新分配的，概率大的赋予短的符号，概率小的赋予长的符号。例如在运动矢量差值中0是最常的出现的，所以赋予符号0，-3的概率就要小，所以赋予符号6。

4.4.2 CAVLC 编码技术

4X4的像素块经过编码、量化后得到的矩阵系数有如下的特性：

- 1、矩阵为稀疏矩阵，大量的系数值为0。
- 2、非零系数经过Zigzag扫描后，通常最后的一些系数为±1构成的序列，如...,-1,-1,0,1。
- 3、相邻块的系数是相关的，块的非零系数个数和相邻块非零系数个数密切相关。
- 4、低频部分的系数通常会有大的数值，越往高频系数值越小。

针对这些特性，H.264使用了CAVLC [10] 技术专门对亮度、色度的系数进行编码。根据以上列举的四个特点，采取以下方法：

- 1、使用 (run、level) 技术来压缩0的个数。
- 2、结合结尾为±1序列和子块非零系数与相邻块相关特点，使用联合码表编码。
- 3、根据相邻块的系数值选择合适的码表编码当前块的系数值。

这就构成了CAVLC的五个段，标设段、符号段、系数段、零系数段、零游程段，针对以上五个段，下面分五个步骤进行说明，并给出一个具体的实例，下图为某一4X4的块经过变换、量化后的系数。

3	0	0	0
0	1	1	0
2	0	0	0
0	-1	0	0

经过Zigzag扫描后的序列为：3002100100-1

图4—7 某一4X4的块经过变换、量化后的系数和扫描后序列

Fig.4.7 the sequence after Zifzag scan

4.4.2.1 标设段

这一段编码的信息是序列中的非零系数个数 (NumCoeff) 和序列结尾为±1的个数 (T1)，它们的取值范围为 $0 \leq \text{NumCoeff} \leq 16$ ， $0 \leq T1 \leq 3$ ，如果T1个数大于3，则大于的部分当作普通的非零系数编码。在此例中，NumCoeff=5，为3,2,1,1,-1；T1=3，为1,1,-1。

NumCoeff和T1联合起来编码，首先根据上面块和左边块的非零系数个数NU、NL选择合适的码表，然后查找码表来编码NumCoeff_T1s。一共有三个常见的NumCoeff_T1s的码表，如果NU、NL的组合没有合适的码表来选择，那么就使用6bits的定长编码，前面4bits代表NumCoeff，后面2bits代表T1的个数。

	Nu	
Nl		
Nu	Nl	N
X	X	$(\text{Nu} + \text{Nl}) / 2$
X		Nu
	X	Nl

图4—8 TotalCoeffs_T1编码表选择示意图

Fig.4.8 The code table of TotalCoeffs_T1

上图中，通过统计NU、NL来得到N决定码表：

$0 \leq N < 2$ ：选择码表0； $2 \leq N < 4$ ：选择码表1；

$4 \leq N < 8$ ：选择码表2； $8 \leq N$ 使用6bits的定长码表。

例中NumCoeff_T1为 (5, 3)，假设使用的是码表0，那么编好的二进制序列为：0000100

4.4.2.2 符号段

这一段编码是对结尾序列 (T1) 的符号进行编码，正号(+)为0，负号(-)为1。不过编码的顺序和扫描的顺序相反。例中结尾序列为1,1,-1，符号为十十一，逆向编码为：100。

4.4.2.3 系数段

这一段编码的是除了结尾序列 (T1) 之外的非零系数，同样，它也是逆向编码的，从高频分量向DC分量编码。这里共有7个不同的码表，每一个码表有独立的门限值。首先第一个系数使用码表0 (如果NumCoeff>10并且T1<3,则使用码表1)，接下来的系数和前一个编码系数使用的码表的门限值进行比较，若高于门限值编号则使用下一个码表，由此可见是根据系数值来自适应编码的。若T1<3，则第一个系数的绝对值减1后在编码。码表的门限值为：0,3,6,12,24,48。

例中为，level(1)=2,使用VLC0编码为001，level(0)=3,使用VLC1编码为：0010。

4.4.2.4 零系数段

这一段编码的是在序列中所有的零系数的个数，因为非零系数个数和零系数个数是相关的，所以使用一个二维的码表查表得到对应的比特流。

例中TotalZeros为6，NumCoeff为5，查表得编码为：100。

4.4.2.5 零游程段

这一段编码的是非零系数前的零的个数，这里也是逆向编码的。例：在序列3002100100-1中，系数-1前零的个数为2，系数1前零的个数为2，系数2前的零的个数为2。这些零的个数也称为run_before。很明显，run_before和TotalZeros每一次编码后剩下的零的个数(ZerosLeft)也是相关的，因此也使用了二维的(run_before/ZerosLeft)码表来查表编码。

例中run_before(4): run_before/ZerosLeft=2/6,编码为001;

run_before(3): run_before/ZerosLeft=2/4,编码为01;

run_before(2): run_before/ZerosLeft=0/2,编码为1;

run_before(1): run_before/ZerosLeft=2/2,编码为00;

run_before(0): run_before/ZerosLeft=0/0,不需要编码。

下面把整个例子的编码过程以表格的形式列出：

表 4—6 本例中的编码流程
Chart4.6 steps of coding one block

编码元素	数值	输出码流
NumCoeff_T1s	NumCoeff=5, T1s=3	0000100
T1 sign(4)	-	1
T1 sign(3)	+	0
T1 sign(2)	+	0
Level(1)	+2(Level_VLC0)	001
Level(0)	+3(Level_VLC1)	0010
TotalZeros	6	100
run_before(4)	run_before/ZerosLeft=2/6	001
run_before(3)	run_before/ZerosLeft=2/4	01
run_before(2)	run_before/ZerosLeft=0/2	1
run_before(1)	run_before/ZerosLeft=2/2	00
run_before(0)	run_before/ZerosLeft=0/0	

CAVLC编码充分考虑到系数本身以及系数中非零系数和零系数个数的相关性，因此它的编码效率对比以前的VLC编码有很大的改进，特别是在QP比较小的时候，它的性能的改善是很明显的。下表^[10]对比了应用CAVLC后，比特节约的百分比。

LOW: QP = 0, 4, 8, 12 High: Qp=16, 20, 24, 28

Table 4.6 comparison of bits used in different sequences

表 4—6 (a) 不同序列的帧内比特节约百分比

QP	Contnr	Foremn	News	Silent	Tempt	Mobile	Paris	Flower
Low	-16.70	-12.45	-16.95	-14.95	-19.21	-23.11	-17.66	-25.25
High	-6.61	-3.16	-8.03	-2.14	-6.53	-9.93	-6.92	-11.78

表 4—6 (b) 不同序列的帧间比特节约百分比

QP	Contnr	Foremn	News	Silent	Tempt	Mobile	Paris	Flower
Low	-7.67	-9.92	-12.59	-10.36	-15.33	-16.05	-14.27	-17.24
High	-2.55	-0.85	-3.53	-1.13	-2.34	-3.28	-3.15	-3.43

4.5 小结

本章详细地说明了H.264中新型的结合量化的整数变换技术，对这一变换方法的原理、具体步骤和改进的性能都详细明了的做了说明。在最后一节中，对H.264特有的、结合宏块类型的、ABT变换的原理和特点进行了分析。

这里，需要指出的是无论4X4的变换和可变块大小的ABT变换，目前的JM校验模型中都没有给出好的优化算法。但是，从计算复杂度的统计来看，变换和量化占用了总的计算复杂度的1/3强，因此目前迫切需要一个好的优化算法，并且这个算法也可以运用到模式判决使用的Hadamard变换中。本文做了一些尝试，认为可以从两个方面突破，其一，找出切实可行的矩阵计算的优化算法，如类似于LLM，AAN的算法等。其二，从变换矩阵入手，改变矩阵的系数，这样可能会带来一些损失，但是通过系数的改变可以使得矩阵计算的优化算法更有效。

总之，结合量化的4X4变换和ABT变换在不增加复杂度的前提下，把变换和量化联合起来，使得图像的质量有很大的提高，同时可变大小的块也更符合图像本身的属性，减小了人工的边缘，很大程度上改善了主观视觉效果。

最后一节中，详细说明了JVT新引入的基于内容的变长编码(CAVLC)的整个过程，从中可以看出CAVLC编码的优越性，它在比特节约方面的性能是巨大的，并且它是复杂度也比CABAC要小的多。

表 4—6 (a) 不同序列的帧内比特节约百分比

QP	Contnr	Foremn	News	Silent	Tempt	Mobile	Paris	Flower
Low	-16.70	-12.45	-16.95	-14.95	-19.21	-23.11	-17.66	-25.25
High	-6.61	-3.16	-8.03	-2.14	-6.53	-9.93	-6.92	-11.78

表 4—6 (b) 不同序列的帧间比特节约百分比

QP	Contnr	Foremn	News	Silent	Tempt	Mobile	Paris	Flower
Low	-7.67	-9.92	-12.59	-10.36	-15.33	-16.05	-14.27	-17.24
High	-2.55	-0.85	-3.53	-1.13	-2.34	-3.28	-3.15	-3.43

4.5 小结

本章详细地说明了H.264中新型的结合量化的整数变换技术，对这一变换方法的原理、具体步骤和改进的性能都详细明了的做了说明。在最后一节中，对H.264特有的、结合宏块类型的、ABT变换的原理和特点进行了分析。

这里，需要指出的是无论4X4的变换和可变块大小的ABT变换，目前的JM校验模型中都没有给出好的优化算法。但是，从计算复杂度的统计来看，变换和量化占用了总的计算复杂度的1/3强，因此目前迫切需要一个好的优化算法，并且这个算法也可以运用到模式判决使用的Hadamard变换中。本文做了一些尝试，认为可以从两个方面突破，其一，找出切实可行的矩阵计算的优化算法，如类似于LLM，AAN的算法等。其二，从变换矩阵入手，改变矩阵的系数，这样可能会带来一些损失，但是通过系数的改变可以使得矩阵计算的优化算法更有效。

总之，结合量化的4X4变换和ABT变换在不增加复杂度的前提下，把变换和量化联合起来，使得图像的质量有很大的提高，同时可变大小的块也更符合图像本身的属性，减小了人工的边缘，很大程度上改善了主观视觉效果。

最后一节中，详细说明了JVT新引入的基于内容的变长编码(CAVLC)的整个过程，从中可以看出CAVLC编码的优越性，它在比特节约方面的性能是巨大的，并且它是复杂度也比CABAC要小的多。

本章参考:

- [1] Hallapuro, M. Karczewicz, and H. Malvar, "Low Complexity Transform and Quantization – Part I: Basic Implementation," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 Document JVT-B038, January 2002.
- [2] H. S. Malvar, "Low-Complexity Length-4 Transform and Quantization with 16-Bit Arithmetic," ITU-T Q.6/SG16 Document VCEG-N44, September, 2001.
- [3] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T Q.6/SG16 (VCEG) Document VCEG-M33, March 2001.
- [4] A. Hallapuro, M. Karczewicz, and H. Malvar, "Low Complexity Transform and Quantization – Part II: Extensions," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 Document JVT-B039, January 2002.
- [5] Mathias Wien, Claudia Mayer, "New Integer Transforms for H.26L", ITU-T SG16 Video Coding Experts Group (Question 15) Q15-J-41, Tenth Meeting: Osaka, Japan, 16-18 May 2000.
- [6] G. Bjontegaard, "Addition of 8x8 transform to H.26L", ITU-T SG16 Video Coding Experts Group (Question 15) 15-I-39, Red Bank, New Jersey, 19-22 October, 1999.
- [7] Joint Video Team, "Draft ITU-T Recommendation H.264(a.k.a. "H.26L")", ITU-T Q.6/SG16 Document VCEG-P07, 16th Meeting: Fairfax, Virginia, USA, 6-10 May, 2002
- [8] Mathias Wien, Claudia Mayer, Jens-Rainer Ohm, "Integer Transforms for H.26L using Adaptive Block Transforms", ITU-T Q.6/SG16 Document Q15-K-24, Eleventh Meeting: Portland, Oregon, USA, 22-25 August, 2000.
- [9] Yuji Itoh, "Bi-directional motion vector coding using universal VLC," Signal Processing: Image Communication, Vol.14, pp. 541-557, May 1999.
- [10] Gisle Bjontegaard and Karl Lillevold, "Context-adaptive VLC (CVLC) coding of coefficients," Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6) JVT-C028, 3rd Meeting: Fairfax, Virginia, USA, 6-10 May, 2002
- [11] Iain E.G Richardson, "H.264 Variable Length Coding." video & image compression resources and research, October 17, 2002

第五章 S 帧技术研究

5.1 引言

随着网络条件的不同，视频接收者应该能够接收到不同视觉质量的码流，例如，当移动用户从户外转到室内时，网络环境可以从低带宽的信道转移到相对带宽高的无线局域网中，那么应该接收到更好质量的视频。在实时的点对点系统中可以通过返回信道来调整编码器的参数（如：量化系数、帧率）。但是在典型的视频流服务器系统中，服务器中存储的是同一视频的不同质量的码流，根据接收端的网络环境传输相应的压缩码流，因此码流的切换只能是从一个质量差的码流切换到另一个质量好的码流，而不是通过返回信道来调整编码器的参数。

最简单的码流切换可以传输 I 帧来实现，但是这样的代价无论对于网络或是接收端都是太大了，因为 I 帧的数据量很大。P 帧也可以作为切换帧，但是由于不同码率的码流之间的精度不同，低码率的 P 帧如果用来作为下一高码率帧的参考帧的话，误码的累积将会造成解码器质量的严重下降，甚至不可解。为此目的，H.264 引入一种新型的帧 S 帧^[1]来更好的解决这个问题。同时，S 帧对于错误掩盖、带宽可升级、码流切换、处理可升级、随机接入、快进、快退等都有很大的帮助。

码流切换：图 5-1 中详细说明了码流切换的方法。码流 2 和码流 1 是同一视频的不同码率下的压缩好的流，假设从码流 1 切换到码流 2，在想要切换的地方加入 S 帧（码流 2 中的 S_2 ，码流 1 中的 S_1 ），当切换的时候，另一个 S 帧（ S_{1_2} ）被传输到码流 2。 S_{1_2} 帧的参考帧是码流 1 中的前一 P 帧， S_2 帧的参考帧是码流 2 中的前一 P 帧，但是它们的重建帧是相同的。

随机接入：图 5-2 中详细说明了随机接入的方法。在码流 1 中的 S_1 是需要接入的帧，新的解码器只需要接收到码流 2 中的 I_2 帧和 S_2 帧就可以随机的接入到码流 1 中，而不需要等待码流 1 中的 I 帧，这样就可以在不破坏码流 1 结构的情况下随机接入到码流 1 中去。

第五章 S 帧技术研究

5.1 引言

随着网络条件的不同，视频接收者应该能够接收到不同视觉质量的码流，例如，当移动用户从户外转到室内时，网络环境可以从低带宽的信道转移到相对带宽高的无线局域网中，那么应该接收到更好质量的视频。在实时的点对点系统中可以通过返回信道来调整编码器的参数（如：量化系数、帧率）。但是在典型的视频流服务器系统中，服务器中存储的是同一视频的不同质量的码流，根据接收端的网络环境传输相应的压缩码流，因此码流的切换只能是从一个质量差的码流切换到另一个质量好的码流，而不是通过返回信道来调整编码器的参数。

最简单的码流切换可以传输 I 帧来实现，但是这样的代价无论对于网络或是接收端都是太大了，因为 I 帧的数据量很大。P 帧也可以作为切换帧，但是由于不同码率的码流之间的精度不同，低码率的 P 帧如果用来作为下一高码率帧的参考帧的话，误码的累积将会造成解码器质量的严重下降，甚至不可解。为此目的，H.264 引入一种新型的帧 S 帧^[1]来更好的解决这个问题。同时，S 帧对于错误掩盖、带宽可升级、码流切换、处理可升级、随机接入、快进、快退等都有很大的帮助。

码流切换：图 5-1 中详细说明了码流切换的方法。码流 2 和码流 1 是同一视频的不同码率下的压缩好的流，假设从码流 1 切换到码流 2，在想要切换的地方加入 S 帧（码流 2 中的 S_2 ，码流 1 中的 S_1 ），当切换的时候，另一个 S 帧（ S_{1_2} ）被传输到码流 2。 S_{1_2} 帧的参考帧是码流 1 中的前一 P 帧， S_2 帧的参考帧是码流 2 中的前一 P 帧，但是它们的重建帧是相同的。

随机接入：图 5-2 中详细说明了随机接入的方法。在码流 1 中的 S_1 是需要接入的帧，新的解码器只需要接收到码流 2 中的 I_2 帧和 S_2 帧就可以随机的接入到码流 1 中，而不需要等待码流 1 中的 I 帧，这样就可以在不破坏码流 1 结构的情况下随机接入到码流 1 中去。

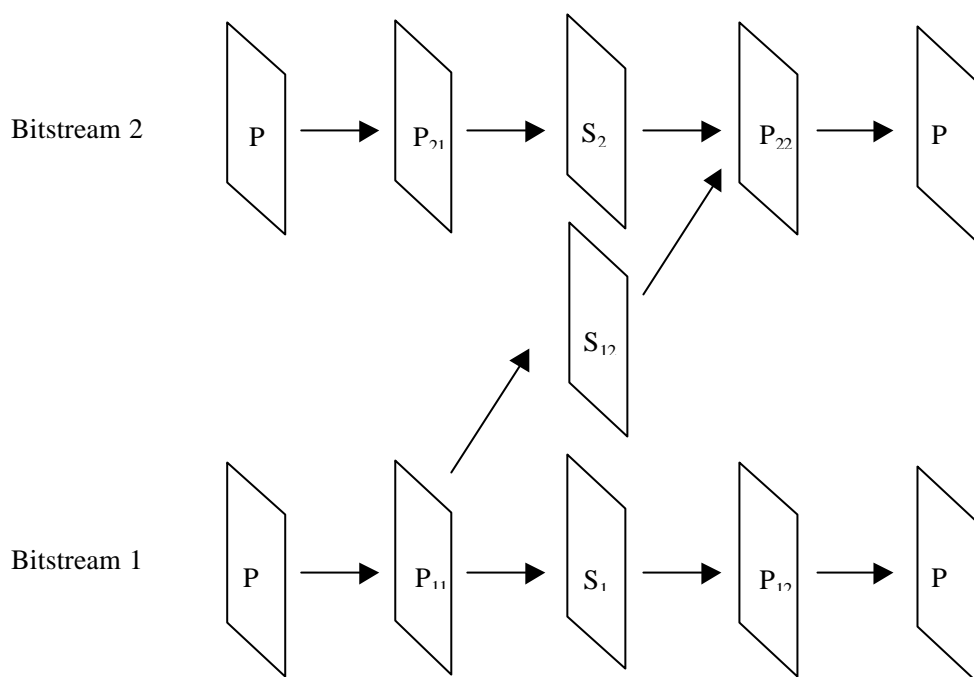


图 5-1 使用 SP 帧从码流 1 切换到码流 2

Fig.5.1 Switching between bitstreams 1 and 2 using SP-pictures

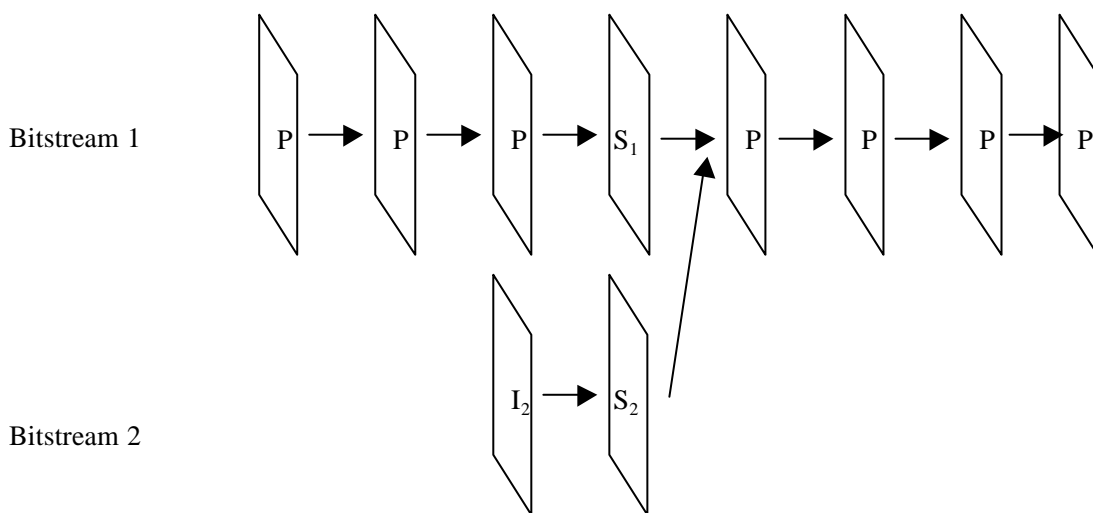


图 5-2 使用 SP 帧的随机接入

Fig.5.2 Random access using SP-pictures

快进 / 快退：图 5-3 中详细说明了快进 / 快退的方法。码流 3 全是由量化系数大的 I 帧组成，它的时间间隔也是大的（例如 2S），码流 2 是对应 S 帧，码流 1 是正常播放的码流。当接收端接收到相应的 I 帧时，由于 SP 帧的可随机接入性，只要再接收到对应的 SP 帧，就可以接入到码流 1 中去，这样就实现了快进，快退的原理也是一样的。

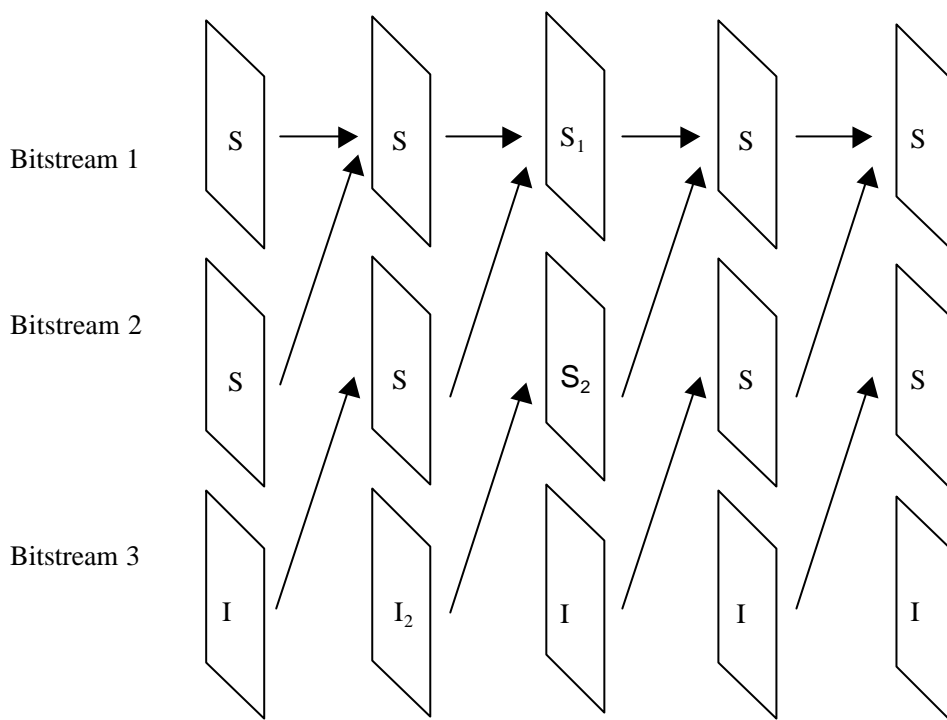


图 5-3 使用 SP 帧的快进 / 快退

Fig.5.3 Fast-forward using SP-pictures

H.264 中有两种 S 帧 SP 帧、SI 帧。SP 帧和 P 帧相似，利用运动补偿技术对视频的冗余信息进行压缩；SI 帧和 I 帧相似，利用帧内预测技术对视频冗余信息进行压缩。SI 帧的编码比较简单，下面将对 SP 帧进行详细的分析。

5.2 SP 帧编解码

SP 帧中最基本的问题就是参考帧和重建帧的问题。在图 5-1 中，码流 2 中的 S_2 帧的参考帧是 P_{21} 帧，重建帧是 P_{22} 帧，码流 1 中的 S_1 也有对应的参考帧 P_{11} 帧、重建帧 P_{12} 帧。当从码流 1 切换到码流 2 时，传输的 S_{12} 帧的重建帧必须和码流 2 的重建帧 P_{22} 一样，这样才能保证码流 2 以后能正确解码，但是 S_{12} 帧的参考帧是 P_{11} ，这样 S_{12} 的参考帧和重建帧分属两个不同码率的码流，因此必须对 S_{12} 帧加入额外的信息来保证重建帧的完全匹配。因此，才会有正常码流中普通 S 帧（如 S_1 、 S_2 ）和切换码流 S_{12} 帧（反之， S_{21} ）。

首先，分析 SP 帧和 P 帧的编、解码方法的不同之处，图 5-4，图 5-5 中详细的给出了它们的区别与联系。当 Ptype 指示当前帧是 P 帧时，编解码器执行的是图示中的上部分框图，当指示当前帧是 SP 帧时，执行的是下部分框图。

首先，分析编码框图（图 5-4）中 SP 帧的整个流程：

- 1、原始图像经过 DCT 变换得到的 $Korig$ 。

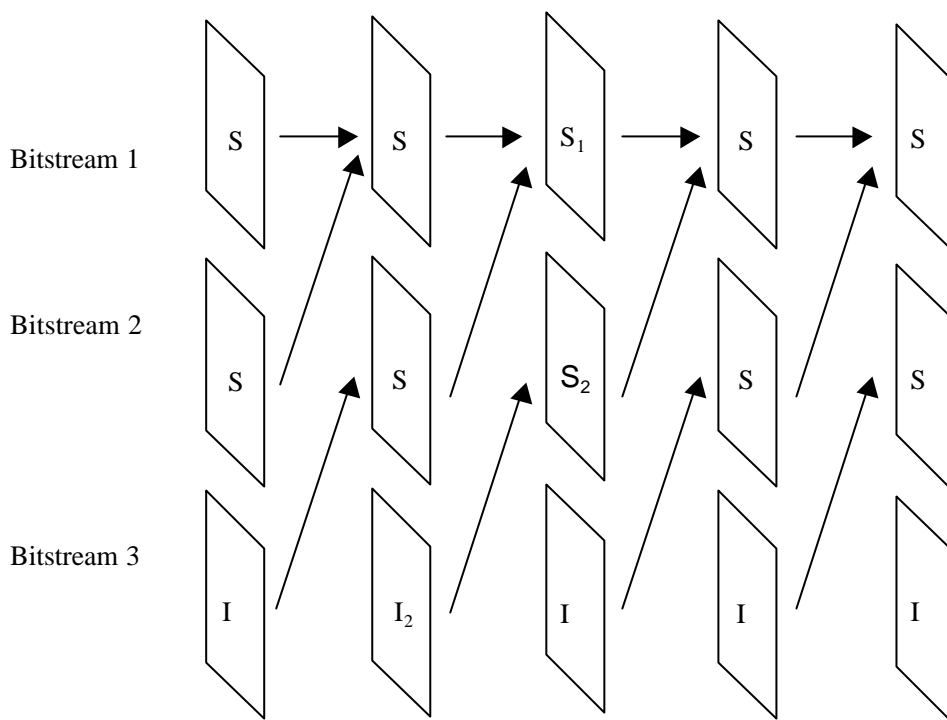


图 5-3 使用 SP 帧的快进 / 快退

Fig.5.3 Fast-forward using SP-pictures

H.264 中有两种 S 帧 SP 帧、SI 帧。SP 帧和 P 帧相似，利用运动补偿技术对视频的冗余信息进行压缩；SI 帧和 I 帧相似，利用帧内预测技术对视频冗余信息进行压缩。SI 帧的编码比较简单，下面将对 SP 帧进行详细的分析。

5.2 SP 帧编解码

SP 帧中最基本的问题就是参考帧和重建帧的问题。在图 5-1 中，码流 2 中的 S_2 帧的参考帧是 P_{21} 帧，重建帧是 P_{22} 帧，码流 1 中的 S_1 也有对应的参考帧 P_{11} 帧、重建帧 P_{12} 帧。当从码流 1 切换到码流 2 时，传输的 S_{12} 帧的重建帧必须和码流 2 的重建帧 P_{22} 一样，这样才能保证码流 2 以后能正确解码，但是 S_{12} 帧的参考帧是 P_{11} ，这样 S_{12} 的参考帧和重建帧分属两个不同码率的码流，因此必须对 S_{12} 帧加入额外的信息来保证重建帧的完全匹配。因此，才会有正常码流中普通 S 帧（如 S_1 、 S_2 ）和切换码流 S_{12} 帧（反之， S_{21} ）。

首先，分析 SP 帧和 P 帧的编、解码方法的不同之处，图 5-4，图 5-5 中详细的给出了它们的区别与联系。当 Ptype 指示当前帧是 P 帧时，编解码器执行的是图示中的上部分框图，当指示当前帧是 SP 帧时，执行的是下部分框图。

首先，分析编码框图（图 5-4）中 SP 帧的整个流程：

- 1、原始图像经过 DCT 变换得到的 $Korig$ 。

- 2、Buffer0 中的图像和原始图像进行运动补偿，然后得到预测的图像。
- 3、预测的图像经过 DCT 变换得到 K_{pred} ，再经过系数为 Q_s 量化得到本地预测帧 L_{pred} ，再经过反量化，然后和 K_{orig} 相减。
- 4、相减得到的差值数据经过 Q_p 量化，在经过熵编码，得到 SP 帧码流。
- 5、另一个分枝是量化好的系数，经过 Q_p^{-1} 反量化，加上 K_{pred} ，然后 Q_s 量化得到本地重建帧 L_{rec} ，再反量化，反 DCT，经过循环滤波器存到 Buffer0 中。

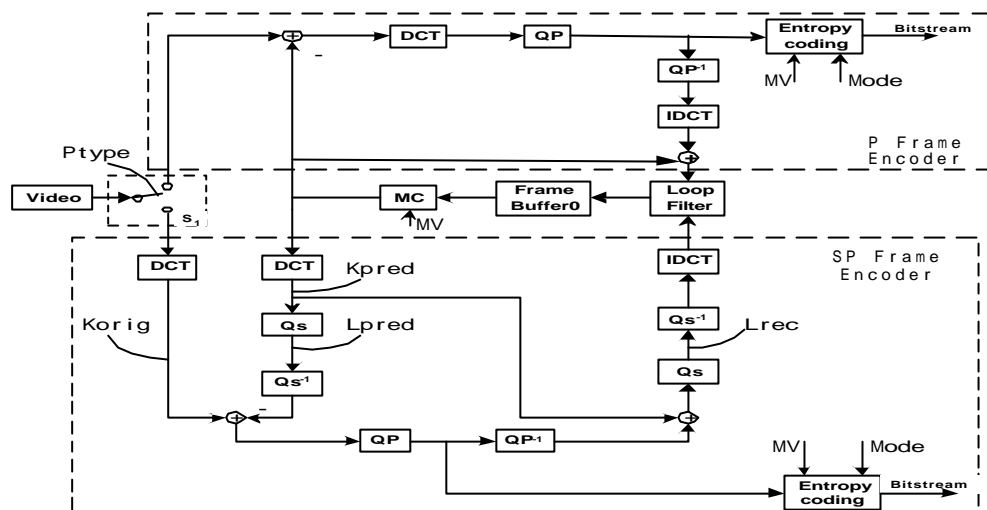


图 5-4 SP 帧和 P 帧的编码器图例

Fig.5.4 Encoder for P frame and SP frame

再来看解码器的框图（图 5-5）中 SP 帧的整个流程：

- 1、接收到的码流经过熵解码，得到运动矢量 MV ，本地差错 $Lerr$ 等信息， $Lerr$ 再经过 Q_p^{-1} 反量化。
- 2、Buffer0 中的图像经过运动补偿，然后 DCT 变换，加上 1 中的结果，得到 K_{rec} ，然后 Q_s 量化， Q_s^{-1} 反量化，反 DCT 变换。
- 3、反 DCT 的图像经过循环滤波器，更新 Buffer0，再显示出来。

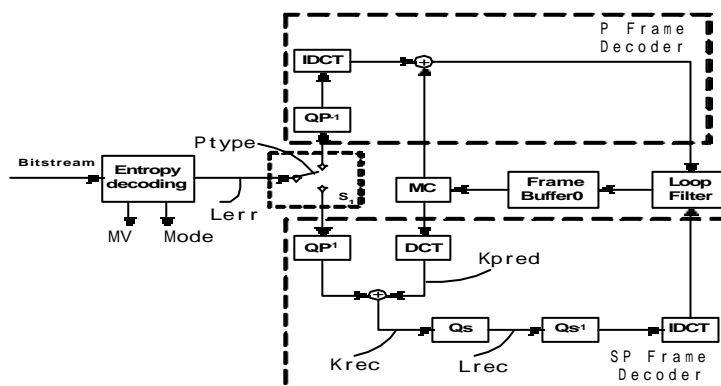


图 5-5 SP 帧和 P 帧的解码图例

Fig.5.5 Decoder for P frame and SP frame

★这里只是图示了普通的 S 帧，切换帧编码方法在图 5-6 中给出。

从以上分析中,可以看出,SP 帧和 P 帧的主要不同之处在于 SP 帧的预测图像要经过 Q_s 量化和 Q_s^{-1} 反量化才和原始图像进行差值计算,不象 P 帧那样预测图像直接和原始图像进行差值。这里的 Q_s 是当前的码流 (bitstream1) 和将切换去的码流 (bitstream2) 共同的量化系数。正是因为这样,才能保证 S12 和 S2 的重建帧的完全一致。

5.3 SP 帧编解码方案

SP 帧是相对较新的帧,因此在目前的 H.264 草案中没有对 SP 帧的编解方案给出详细的说明,只给出了一个参考模型。本章以下内容介绍两种 SP 帧的编解码方案和一些试验数据。

5.3.1 JM-1 中的 SP 帧结构

在图 SP-6 [2] 中,详细的说明了 JM-1 中的 SP 帧编码流程:

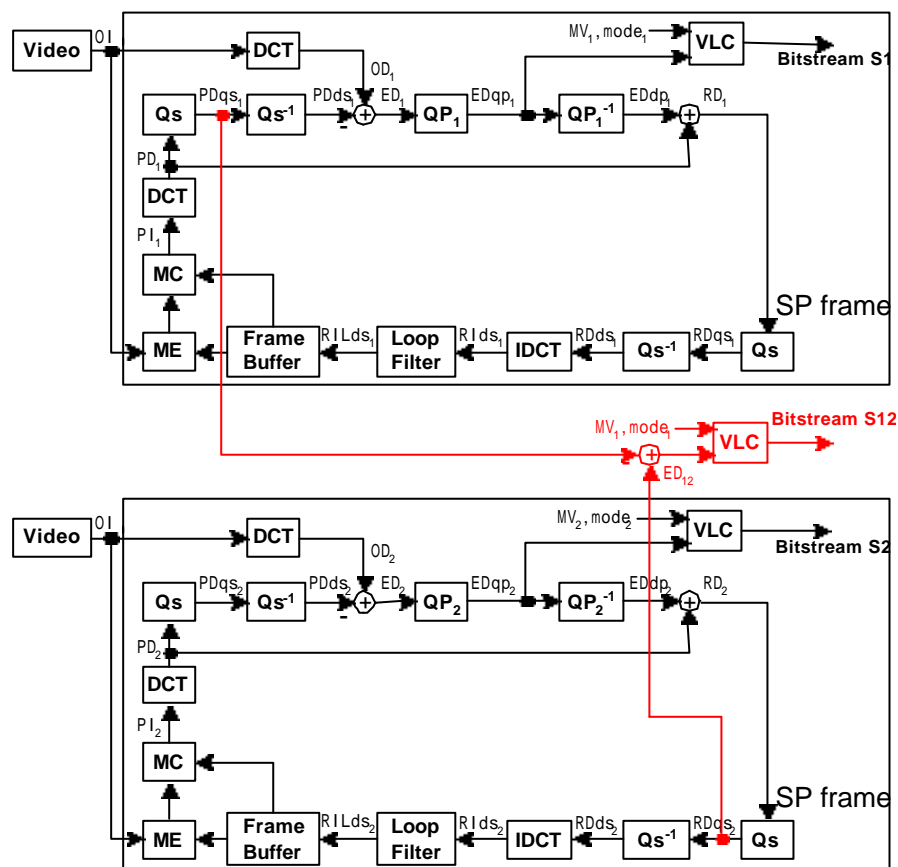


图 5—6 JM-1 中的普通 S1、S2 帧和切换 S12 帧编码框图

Fig.5.6 The block diagram of compressing S1, S2 and S12 in the JM-1

首先,对图中所用的标记作说明:

- 第一个字母
 - O: 宏块或块大小的原始图像 P: 宏块或块大小的预测图像
 - R: 宏块或块大小的重建图像 E: 宏块或块大小的预测差值图像

从以上分析中,可以看出,SP 帧和 P 帧的主要不同之处在于 SP 帧的预测图像要经过 Q_s 量化和 Q_s^{-1} 反量化才和原始图像进行差值计算,不象 P 帧那样预测图像直接和原始图像进行差值。这里的 Q_s 是当前的码流 (bitstream1) 和将切换去的码流 (bitstream2) 共同的量化系数。正是因为这样,才能保证 S12 和 S2 的重建帧的完全一致。

5.3 SP 帧编解码方案

SP 帧是相对较新的帧，因此在目前的 H.264 草案中没有对 SP 帧的编解方案给出详细的说明，只给出了一个参考模型。本章以下内容介绍两种 SP 帧的编解码方案和一些试验数据。

5.3.1 JM-1 中的SP 帧结构

在图 SP-6^[2] 中, 详细的说明了 JM-1 中的 SP 帧编码流程:

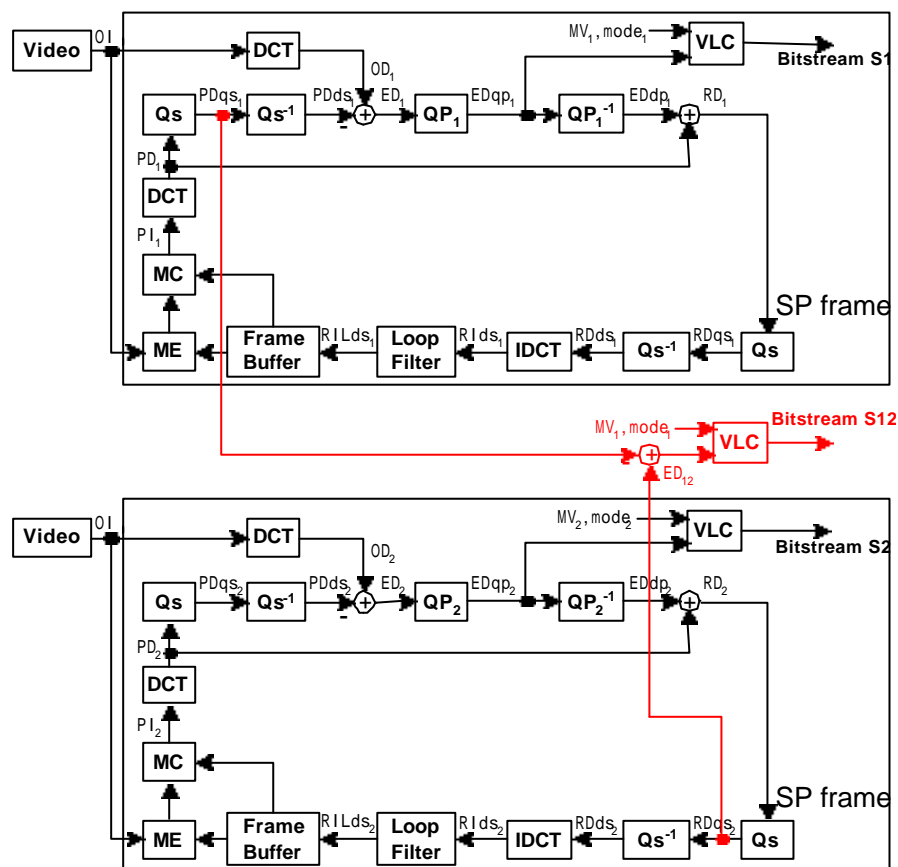


图 5-6 JM-1 中的普通 S1、S2 帧和切换 S12 帧编码框图

Fig.5.6 The block diagram of compressing S1, S2 and S12 in the JM-1

首先，对图中所用的标记作说明：

- 第一个字母
 - O : 宏块或块大小的原始图像 P : 宏块或块大小的预测图像
 - R : 宏块或块大小的重建图像 E : 宏块或块大小的预测差值图像

从以上分析中,可以看出,SP 帧和 P 帧的主要不同之处在于 SP 帧的预测图像要经过 Q_s 量化和 Q_s^{-1} 反量化才和原始图像进行差值计算,不象 P 帧那样预测图像直接和原始图像进行差值。这里的 Q_s 是当前的码流 (bitstream1) 和将切换去的码流 (bitstream2) 共同的量化系数。正是因为这样,才能保证 S12 和 S2 的重建帧的完全一致。

5.3 SP 帧编解码方案

SP 帧是相对较新的帧，因此在目前的 H.264 草案中没有对 SP 帧的编解方案给出详细的说明，只给出了一个参考模型。本章以下内容介绍两种 SP 帧的编解码方案和一些试验数据。

5.3.1 JM-1 中的 SP 帧结构

在图 SP-6^[2] 中, 详细的说明了 JM-1 中的 SP 帧编码流程:

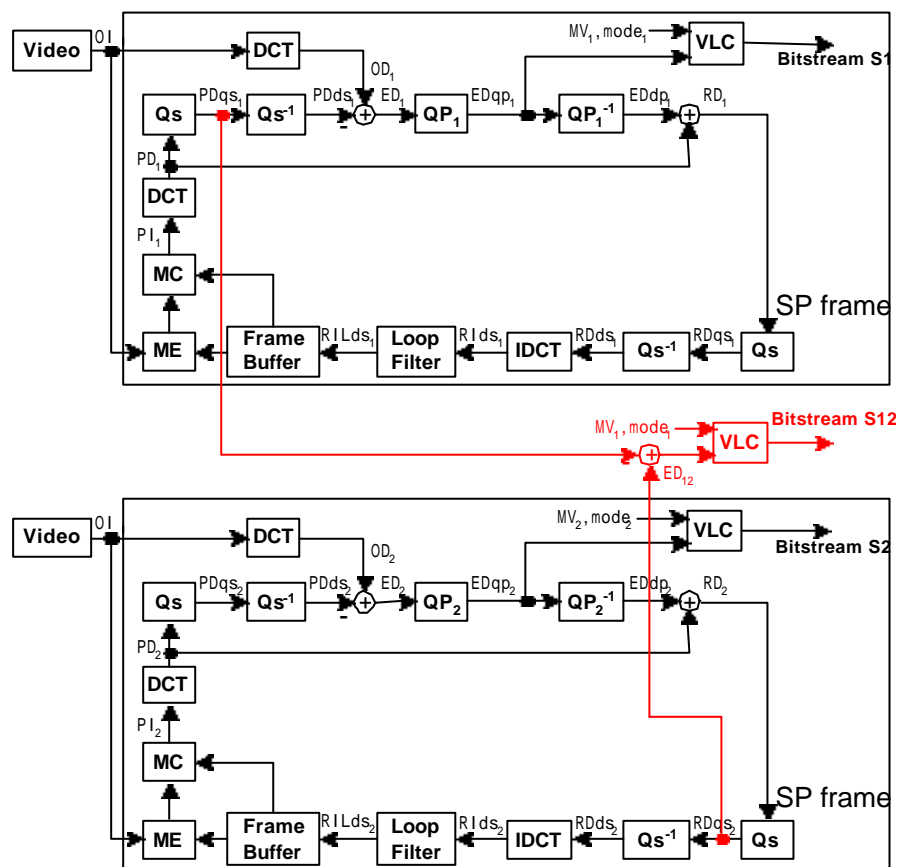


图 5-6 JM-1 中的普通 S1、S2 帧和切换 S12 帧编码框图

Fig.5.6 The block diagram of compressing S1, S2 and S12 in the JM-1

首先，对图中所用的标记作说明：

- 第一个字母
 - O : 宏块或块大小的原始图像 P : 宏块或块大小的预测图像
 - R : 宏块或块大小的重建图像 E : 宏块或块大小的预测差值图像

- 第二个字母
 - D : DCT 域
 - I : Pixel 域
- 第一个可选字母
 - L : 宏块或块大小滤波后的重建图像
- 第二个可选字母
 - q : 量化
 - d : 反量化
- 第三个可选字母
 - p : QP 的量化系数
 - s : Q_s 的量化系数

下面对图 5—6 作详细的流程说明, S_1 、 S_2 帧作为普通的 SP 码流, S_{12} 作为切换码流, 假定从 S_1 切换到 S_2 。

首先, 看普通 S_1 帧的编码过程:

- 1、原始图像 O_1 经过 DCT 变换得到变换域的原始图像 OD_1 。
- 2、帧存中的图像经过运动补偿后得到预测图像 PI_1 , 再经过 DCT 变换得到变换域的预测图像 PD_1 。
- 3、用 Q_s 量化和解量化变换域预测图像 PD_1 , 得到变换域 Q_s 量化和 Q_s 反量化图像 $PDqs_1$ 和 $PDds_1$ 。
- 4、变换域的原始图像 OD_1 减去变换域 Q_s 量化图像 $PDds_1$, 得到差值图像 ED_1 。
- 5、用 QP_1 量化差值图像 ED_1 , 得到量化后的差值图像 $EDqp_1$, 使用熵编码 (UVLC 或 CABAC) 方法编码 $EDqp_1$, 加上运动矢量 MV_1 和宏块类型 $mode_1$ 等信息, 组成码流 S_1 。
- 6、用 QP_1 反量化 $EDqp_1$, 得到变换域的反量化差值图像 $EDdp_1$ 。
- 7、 $EDdp_1$ 加上变换域的预测图像 PD_1 , 得到变换域重建图像 RD_1 。
- 8、用 Q_s 量化变换域重建图像 RD_1 , 得到变换域 Q_s 量化重建图像 $RDqs_1$, 再反量化, 反 DCT 变换, 得到重建图像 $RI ds_1$ 。
- 9、重建图像 $RI ds_1$ 经过循环滤波器的得到滤波后的图像 $RI L ds_1$, 更新帧存并显示。

其次, 看切换 S_{12} 帧的编码过程:

- 1、码流 S_2 的变换域 Q_s 量化重建图像 $RDqs_2$ 减去码流 S_1 变换域 Q_s 量化图像 $PDqs_1$, 得到码流 S_2 和码流 S_1 的差值图像 ED_{12} 。
- 2、差值图像 ED_{12} 加上运动矢量 MV_1 和宏块类型 $mode_1$ 等信息, 组成码流 S_{12} 。

具体的切换过程和传输的码流信息如下:

当解码器觉察到自身网络环境改变的时候, 向视频服务器发出请求, 希望改变正在传

输码流的码率，可以从低码率跃到高码率，也可以是高码率转向低码率。这时视频服务器会在当前的码流中查找切换帧所在的位置 $S1$ （假定从码流 1 切换到码流 2），然后发一个信息给解码器，通知它将在 $S1$ 的前一帧进行码率切换。接着，在切换的位置 $S1$ 发送 $S12$ 帧，解码器接收到 $S12$ 帧后，因为它有码流 1 的前一 P 帧做参考帧，所以可以正确解码，解好的帧作为码流 2 中 $S2$ 的后一 P 帧的参考帧，因为 $S2$ 帧和 $S12$ 帧的重建帧是一样的，既，参考 $S2$ 帧的重建帧和参考 $S12$ 帧的重建帧是无差别的。所以，以后收到的码流 2 可以正确的解码。

在图 5—7^[2]中，详细的说明了 JM-1 中的 SP 帧解码流程：

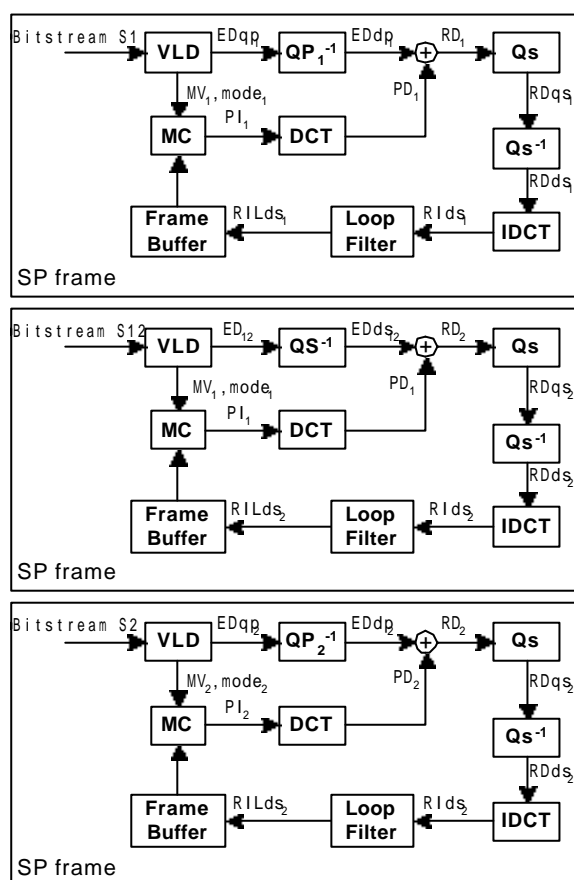


图 5—7 JM-1 中的普通 $S1$ 、 $S2$ 帧和切换 $S12$ 帧解码框图

Fig.5.7 The decoder for SP pictures in the JM-1

首先，看普通 $S1$ 帧的解码过程：

- 1、经过熵解码后，得到 DCT 域量化后的差值图像 $EDqp_1$ ，和运动矢量 MV_1 和宏块类型 $mode_1$ 等信息。
- 2、用 QP_1 对 $EDqp_1$ 进行反量化，得到 DCT 域反量化的差值图像 $EDdp_1$ 。
- 3、对帧存中的图像进行 MV_1 的运动补偿，然后进行 DCT 变换，得到变换域的预测图像 PD_1 。

- 4、PD1 和 EDdp1 相加,得到 DCT 域重建图像 RD1。
- 5、用 Qs 对 RD1 进行量化和反量化,得到 DCT 域的 Qs 反量化重建图像 RDds1。
- 6、再经过反 DCT 变换,循环滤波器,得到解码端重建的图像 RILds1,用来更新帧存和显示。

切换 S12 帧的解码过程和普通 S1 帧相似,只是 QS 代替了码流 1 中的量化系数 QP1。

5.3.2 另一种 SP 帧结构

JM-1 中的结构是比较复杂的,用到了较多的 DCT 变换和量化,并且码流中 SP 帧的结构是固定的。下面介绍另一种 SP 帧的结构。在这个方案中,码流中 SP 帧的结构是灵活的。切换时,码流 1 中的 S1 帧不规定必须是 SP 帧编码,如果别的码流不要求在此时切换到码流 1,那么码流 1 中的 S1 帧可以当作普通的 P 帧编码。这种策略为码流的切换提供了更加灵活的措施。在 TCP/IP 环境中,从高码率向低码率的切换要比从低码率向高码率切换的情况更多,这种灵活的措施可以让高码率的码流以更大的时间间隔来插入 SP 帧,降低本身的码率。

图 5—8 [2] 中,详细的说明了此方案中的 SP 帧编码流程:

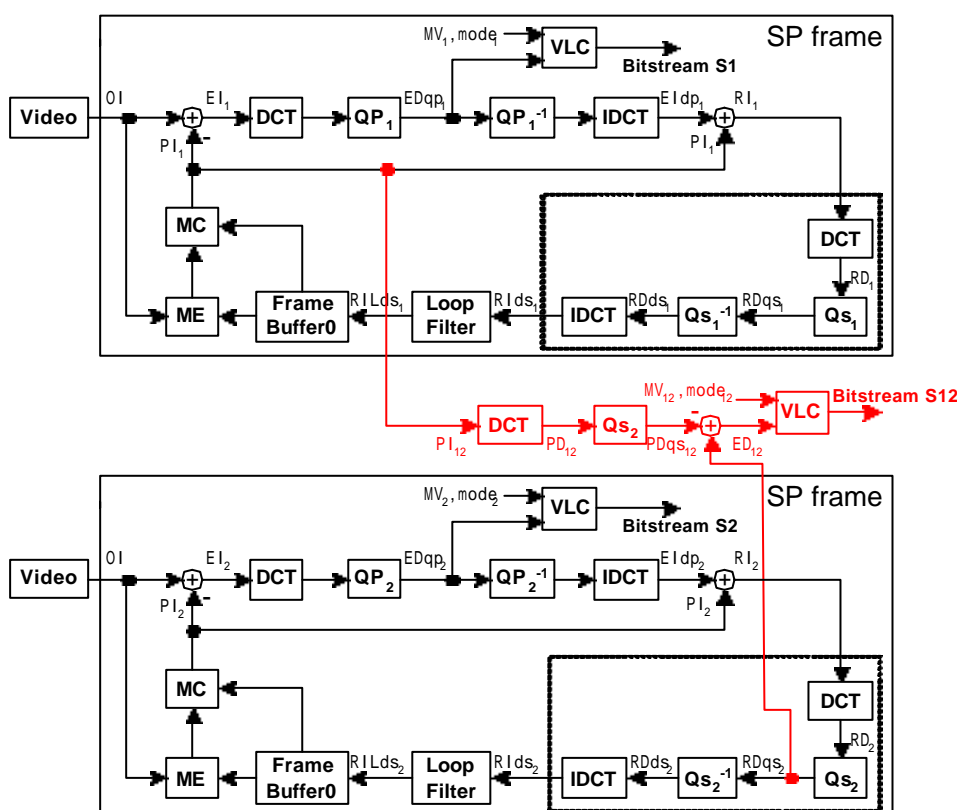


图 5—8 此方案的普通 S1、S2 帧和切换 S12 帧编码框图

Fig.5.8 The block diagram of compressing S1, S2 and S12

从框图中可以看出,此方案中 SP 帧和 P 帧的主要不同之处在于虚线框中的部分,对

- 4、PD1 和 EDdp1 相加,得到 DCT 域重建图像 RD1。
- 5、用 Qs 对 RD1 进行量化和反量化,得到 DCT 域的 Qs 反量化重建图像 RDds1。
- 6、再经过反 DCT 变换,循环滤波器,得到解码端重建的图像 RILds1,用来更新帧存和显示。

切换 S12 帧的解码过程和普通 S1 帧相似,只是 QS 代替了码流 1 中的量化系数 QP1。

5.3.2 另一种 SP 帧结构

JM-1 中的结构是比较复杂的,用到了较多的 DCT 变换和量化,并且码流中 SP 帧的结构是固定的。下面介绍另一种 SP 帧的结构。在这个方案中,码流中 SP 帧的结构是灵活的。切换时,码流 1 中的 S1 帧不规定必须是 SP 帧编码,如果别的码流不要求在此时切换到码流 1,那么码流 1 中的 S1 帧可以当作普通的 P 帧编码。这种策略为码流的切换提供了更加灵活的措施。在 TCP/IP 环境中,从高码率向低码率的切换要比从低码率向高码率切换的情况更多,这种灵活的措施可以让高码率的码流以更大的时间间隔来插入 SP 帧,降低本身的码率。

图 5—8 [2] 中,详细的说明了此方案中的 SP 帧编码流程:

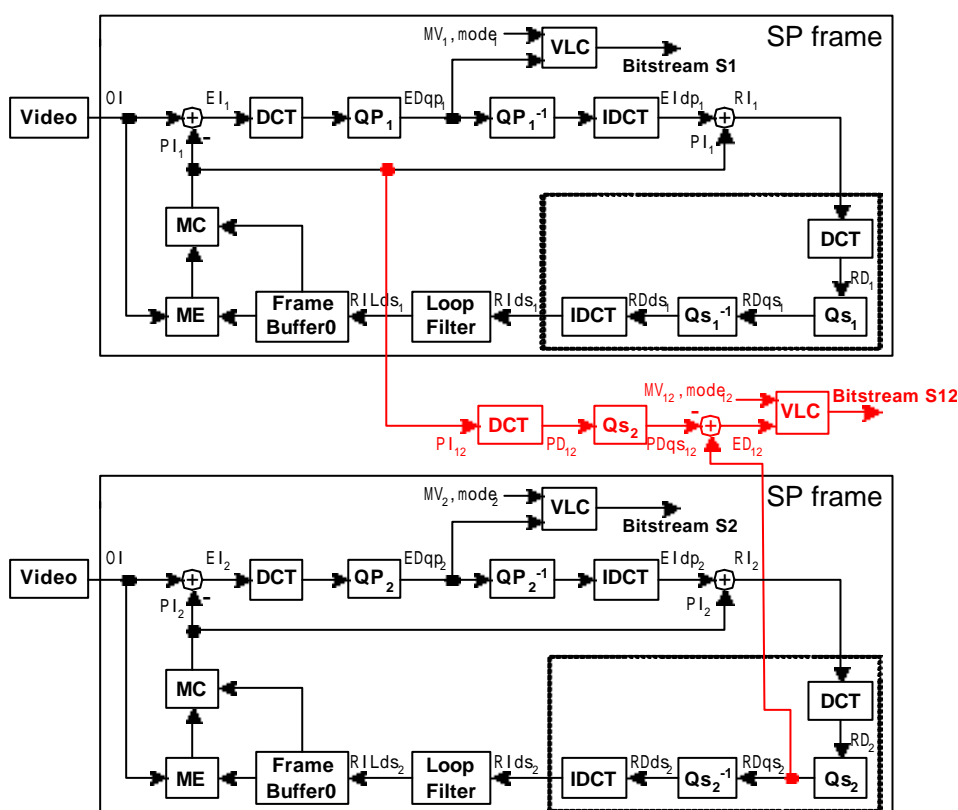


图 5—8 此方案的普通 S1、S2 帧和切换 S12 帧编码框图

Fig.5.8 The block diagram of compressing S1, S2 and S12

从框图中可以看出,此方案中 SP 帧和 P 帧的主要不同之处在于虚线框中的部分,对

原来的结构不要进行很大的改动，只要在原框架上加上虚线框中的部分。和 JM-1 的不同之处在于，各自码流中的 Q_s 也可以有不同的量化阶，可以有更大的灵活性，由于 Q_s 的不同，切换帧 S12 要进行 DCT 变换、 Q_{s2} 的量化然后求差值编码。

这种方案的优点在于不必对原有的框图结构进行大的改动，处理的也相对简单，特别适合量化 QP 是定值的情况，但是对于码流的量化 QP 改变的情况则没有 JM-1 的方案来的好，因此可以把两者的优点结合起来考虑。

在图 5-9^[2]中, 详细的说明了此方案中的 SP 帧解码流程:

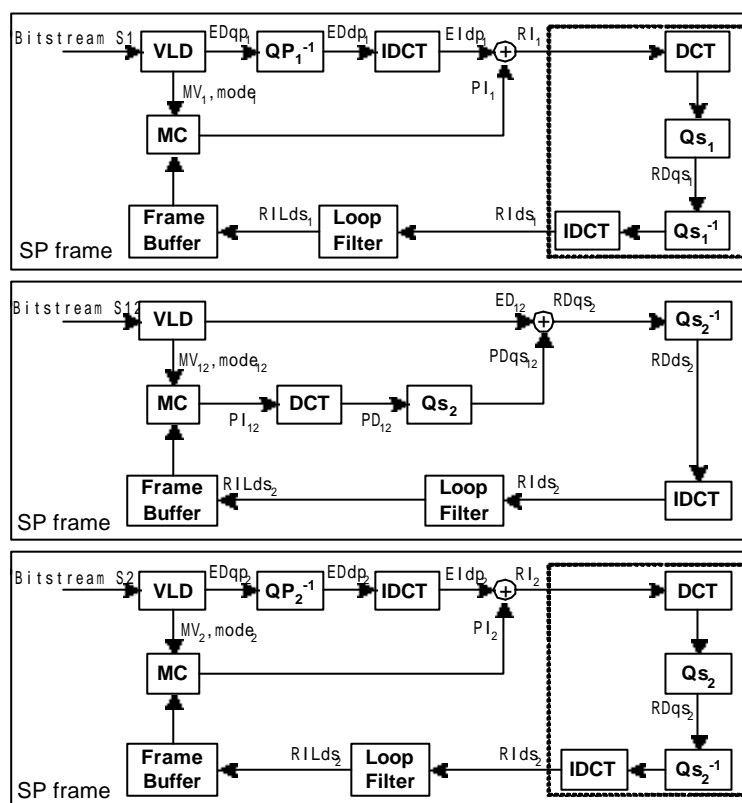


图 5-9 此方案的普通 S1、S2 帧和切换 S12 帧解码框图

Fig.5.9 The decoder for SP frames S1, S2, and S12

此方案的解码框和 JM-1 中的解码框图对比起来也是要精简一些，它的优缺点和编码框图一样，在此就不详述。

还有一些别的 SP 帧的编码方案，这些方案都是从不同的编码复杂度和编码的性能作一些综合的考虑，但是其中不变的原则是切换帧和要切换去的帧的重建帧一定要相同，这样才能保证无缝的切换。

5.4 SP 帧编解效率

从以上的章节中可以看出 SP 帧的引入为码流切换、随机接入、快进 / 快退等功能的实

原来的结构不要进行很大的改动，只要在原框架上加上虚线框中的部分。和 JM-1 的不同之处在于，各自码流中的 Qs 也可以有不同的量化阶，可以有更大的灵活性，由于 Qs 的不同，切换帧 S12 要进行 DCT 变换、 $Qs2$ 的量化然后求差值编码。

这种方案的优点在于不必对原有的框图结构进行大的改动，处理的也相对简单，特别适合量化 QP 是定值的情况，但是对于码流的量化 QP 改变的情况则没有 JM-1 的方案来的好，因此可以把两者的优点结合起来考虑。

在图 5—9^[2]中，详细的说明了此方案中的 SP 帧解码流程：

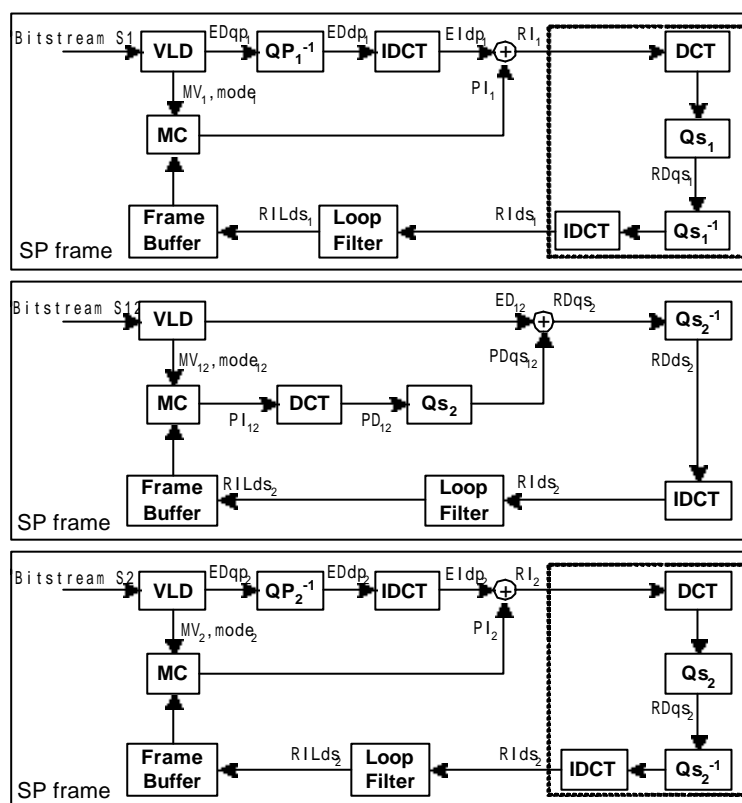


图 5—9 此方案的普通 S1、S2 帧和切换 S12 帧解码框图

Fig.5.9 The decoder for SP frames S1, S2, and S12

此方案的解码框和 JM-1 中的解码框图对比起来也是要精简一些，它的优缺点和编码框图一样，在此就不详述。

还有一些别的 SP 帧的编码方案，这些方案都是从不同的编码复杂度和编码的性能作一些综合的考虑，但是其中不变的原则是切换帧和要切换去的帧的重建帧一定要相同，这样才能保证无缝的切换。

5.4 SP 帧编解效率

从以上的章节中可以看出 SP 帧的引入为码流切换、随机接入、快进 / 快退等功能的实

现提供了更加简便的方法，同时 SP 的编码效率也是很高的。下面从单帧的 SP 帧、间隔 SP 帧和切换 SP 帧来比较和 I 帧、P 帧的编码效率。

下面对比一下 I、P、SP 帧的编码效率：

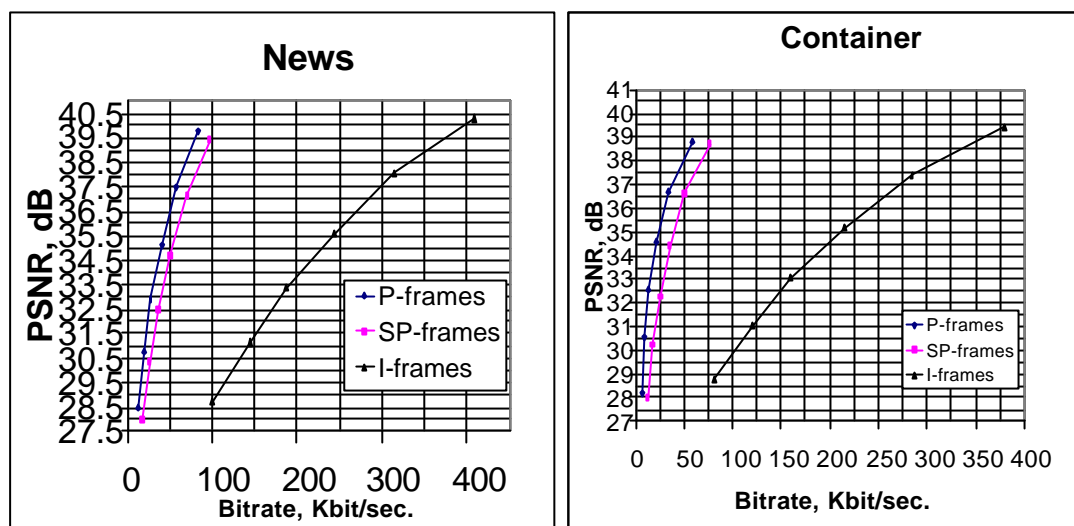


图 5—10 I、P、SP 帧编码效率对比图

Fig.5.10 Compare coding efficiency of I、P、SP

图 5—10 中对比了 I、P、SP 帧的编码效率，对 News 和 Container 序列分别进行编码的比较。黑色的线是所有的帧都为 I 帧；粉红色的线是第一帧为 I 帧，其余为 SP 帧；蓝色的线是第一帧为 I 帧，其余为 P 帧。由两个图中可以看出，SP 帧的编码效率和 P 帧差不多，比 I 帧的编码效率要高很多。

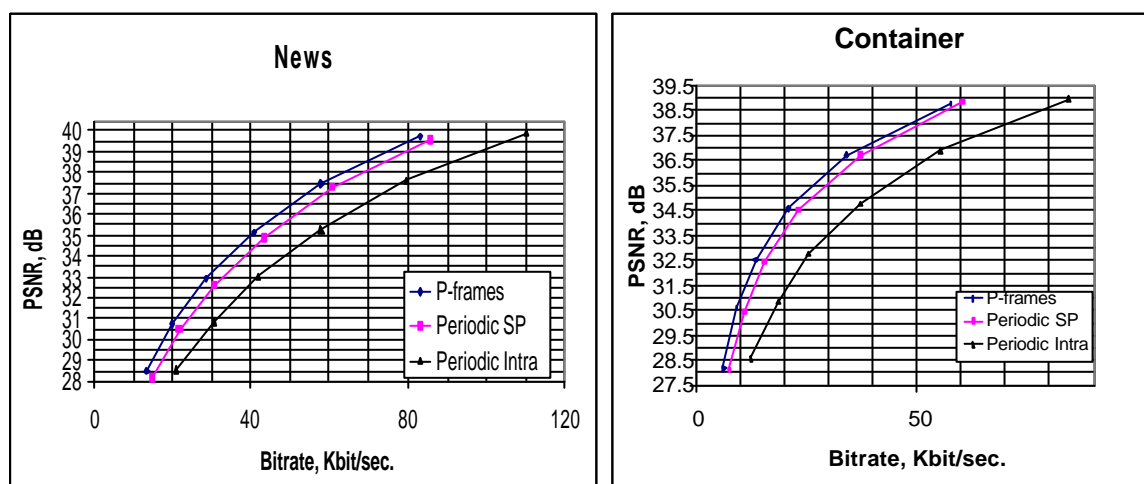


图 5—11 1 秒间插 I、P、SP 帧编码效率对比图

Fig.5.11 coding efficiency of I、P、SP in 1 sec.

图 5—11 中对比了 1 秒钟间插 I、P、SP 帧的编码效率，对 News 和 Container 序列分别进行编码的比较。黑色的线是第一帧为 I 帧，间隔 1 秒插入 I 帧，其余为 P 帧；粉红色的线是第一帧为 I 帧，间隔 1 秒插入 SP 帧，其余为 P 帧；蓝色的线是第一帧为 I 帧，

其余为 P 帧。由两个图中可以看出，在间插情况下，SP 帧的编码效率和 P 帧差不多，比 I 帧的编码效率要高。

在表 5-1，表 5-2 中给出切换帧 S12 和 I 帧大小的比较结果：

第一行是码流 1 的量化系数，第一列是码流 2 的量化系数；“intra”行是码流 2 在不采用 SP 帧时，各个量化系数的 I 帧平均比特数；其余为切换帧 S12 的比特数。

注：为了对比码率大小的±50%改变，码流 1 和码流 2 量化系数 QP 之差不超过 6

例：在表 5-1 中，第三行第二列为 22006 表示当 QP=13 的码流 1 切换到 QP=16 的码流 2 时需要的切换帧 S12 为 22006 比特。

表 5-1 “hall”序列中切换帧 S12 和 I 帧大小比较

Chart5.1 comparsion of S12 and I in “hall”sequence

QP	13	16	19	22	25	28
13		16264	13694			
16	22006		13664	11556		
19	26329	18191		11709	9220	
22		22160	15580		9476	6798
25			18495	12588		7420
28				15314	10293	
Intra	33657	25889	19810	14950	11225	7788

表 5-2 “foreman”序列中切换帧 S12 和 I 帧大小比较

Chart5.2 comparsion of S12 and I in “foreman”sequence

QP	13	16	19	22	25	28
13		23026	18091			
16	30142		18060	15563		
19	33567	25720		15540	11320	
22		28304	19975		11160	7626
25			21970	16671		7360
28				15314	10293	
Intra	37214	27427	19927	14435	10211	7018

从表 5-1、表 5-2 中我们可以看出切换帧 S12 的大小是和两个切换的的码流密切相关的。总的来说切换帧 S12 占用的比特数要比 I 帧要少，特别是从高码率切换到低码率的情况。

特别需要说明的是 S12 帧是附加的码流，它只出现在需要切换的时候，而不象 I 帧那样必须包含在整个码流中，因此，它在具有 I 帧那样的功能的同时，还极大的降低码流的大小。

5.5 小结

在本章节中，对 SP 帧的功能和编解码的方法进行一个完整、详尽的分析。

首先，对 SP 帧的概念和功能进行了介绍和说明；其次，分别介绍了两种编解码的框图和具体的过程，并对其适用的范围进行了说明；最后，列举了一些图表和数据对比了 I 帧、P 帧和 SP 帧的编码效率。

因为 SI 帧的框图和方法与 SP 帧比较相近，只是去掉运动估计的部分，因此本章没有详细的说明。

从以上的分析中可以看出 S 帧的引入无论是在功能上还是在编码效率以及灵活性方面都有很大的提高。它对于错误掩盖、带宽可升级、码流切换、处理可升级、随机接入、快进、快退等都有很大的帮助。

本章参考:

- [1] M. Karczewicz and R. Kurceren, "A proposal for SP-frames", doc VCEG-L27, ITU-T video coding experts group meeting, Eibsee, Germany, Jan 2001.
- [2] F. Wu, S. Li, R. Kurceren and J. Lainema, "The core experiment on the SP frame coding", doc JVT-B112, Joint video term group meeting, Geneva, Switzerland, Jan 2002.
- [3] X. Sun, F. Wu, S. Li, W. Gao and Y.-Q Zhang, "Improved SP coding technique", doc JVT-B097, Joint video term group meeting, Geneva, Switzerland, Jan 2002.
- [4] R. Kurceren and M.Karczewicz, "Improved SP-frame encoding", doc VCEG-M73, ITU-T video coding experts group meeting, Austin, USA, April 2001.

结语和建议

广播级的视频效果是近年来视频通信领域方兴未艾的研究重点，它覆盖了数字电视、DVD、家庭影院等多种新兴产业，这一切是和高效的视频压缩算法分不开的。其中，MPEG-4 和 H.263 算法由于性能上的优越得到了广泛的应用并且相关的产品层出不穷。随着技术的发展和通讯手段的提高，无论是在高码率信道还是在低码率信道，迫切的需要一种新的高效的算法能够在高带宽信道中到达 HDTV 的视觉效果也能够在低带宽信道中有连续的好的播放质量。特别是近年来随着 Internet 的普及和发展，越来越需要能够在网上观看高质量的视频。正是这些需求推动了新的视频压缩算法—H.264 的出现。可以说，H.264 是目前图像通讯领域中最热门也最领先的研究方向，它本是 ITU-T 下一代的视频会议通讯标准，它采纳以往算法的优点并采用了新的技术，博览众长融会贯通，它的出色性能得到了各方面的广泛认同，因此，MPEG 采纳它为 MPEG-4 的 Part-10 部分。它在取得好的视频质量的同时也极大的降低了码率，因此，它的应用范围极广，从 HDTV 到实时视频会议，从 Internet 上网络点播到 3G 移动图像通讯，都有它的用武之地。相信在不久基于 H.264 的各种视频通讯产品将会大量涌现，本文正是在这种情况下，对 H.264 编码方案中的一些新技术新应用力图做一个详细的具体的说明并提出一些自己的看法和改进。需要指出的是，到目前为止 H.264 还没有形成最终的标准，文中介绍的内容可能会有变化，但是基本的思想和方法应该不会有大的改动。

本文的主要内容集中于对 H.264 视频编码新技术的研究和分析，进行一些验证并提出了自己的改进措施。具体为对 H.264 帧内编码的预测技术进行了详细的分析，得出了 H.264 的帧内预测在相同压缩比下，它的 PSNR 性能比 H.263 要高出 5—10dB，和 JPEG-2000 相当的结论。在此基础上，分析了目前 H.264 的帧内预测技术不适合并行处理以及复杂度太高的缺点，因此提出对预测块预选像素值和色度预测技术的两点改进方案，试验结果表明改进后的方案在以极小的信噪比损失为代价的条件下，在计算复杂度和并行处理上都有大的性能改善。接下来详细分析了整数变换、基于内容的自适应可变长度编码技术（CAVLC）和结合宏块大小的自适应块变换（ABT）技术以及随机接入帧 SP 技术，对整数变换、ABT 的数学基础、CAVLC 的技术创新点进行了研究和验证，并提出改进的方向；对 SP 帧取代 I 帧

的原理和详细框图进行详细说明，通过图表和数据对比了 I 帧、P 帧和 SP 帧的编码效率。

总体来说，作者认为 H.264 将会是下一代领先的视频压缩标准，它在图像质量和码流大小方面都有目前其他标准无法比拟的优越性，但是必须承认，H.264 的算法复杂度是很高的，由于它是多帧参考、可变宏块大小、自适应算术编码、 $1/4$ 像素精度等导致了 H.264 编码器非常复杂，在编同样的图像序列下，它使用的时间要大大超过 MPEG-4 使用的时间，不过 H.264 解码器复杂度不大，和 MPEG-4 相当。H.264 正是牺牲了算法复杂度取得一个高的图像质量。所以，作者认为 H.264 在单向视频广播或是 PS 流中将会首先应用，然后在此基础上它将逐渐延伸到实时视频传输等对时间要求比较高的场合。

当前 H.264 的主要工作分为两大部分，其一是对 H.264 算法的完善，例如码率控制算法、抗误码等。目前 H.264 还没有一个好的码率控制算法，这是因为 H.264 使用 R-D 率失真模型和宏块级的码率控制导致码率控制算法非常复杂；抗误码也是 H.264 要在无线信道条件下必须解决的问题。其二是对 H.264 算法上的优化，例如在 $1/4$ 像素下的快速运动估计、宏块预测的并行处理等。

基于以上的认识和结合研究的兴趣，以后的工作主要致力于标清或高清的高效压缩和高质量的视频上。附录中列出了本文使用 JM-4 编码器对标清和高清序列的编码效率和 MPEG-4 的比较，也摘录了一些别人的试验结果。考虑到目前的 JM-4 编码器一直使用 QCIF 或 CIF 大小的序列测试，并没有专门针对标清或高清进行优化，作者相信 H.264 应用于标清或高清一定大有可为。

对于本文的欠缺之处，敬请老师和同学指正。

附录

随着数字电视的发展,消费者希望能够观看清晰度高的视频节目。目前数字电视主要使用 MPEG-2 压缩算法来处理数字电视节目并得到了广泛的应用。随着 H.264 标准的即将出台,把 H.264 应用于标准清晰度或高清晰度视频是目前研究的热点之一。本文也做了一些尝试,使用 H.264 校验模型 JM-4 对标准清晰度的视频进行编码,并和 MPEG-4 进行比较,下面的表格和图表是对比的结果。

试验一:

测试序列: 720X576 的 football 序列一 此序列的特点是运动剧烈,图像复杂度中等

测试条件: 编码次序为 IPPP, JM-4 选择 RD 优化、UVLC 编码; MPEG-4 使用微软编码器,两者都没有加入码率控制算法

表一 a 编好的 football 序列中 P 帧占用的比特数

Table1.a bits results of football

Frame	P1	P2	P3	P4	P5	P6	P7	P8	P9
JM-4	167344	169432	170096	166480	173840	171112	168304	169048	163184
MPEG-4	206536	230512	231568	225472	234376	234336	226352	223904	221960
Frame	P10	P11	P12	P13	P14	P15	P16	P17	P18
JM-4	170304	170304	176456	174072	162760	164432	154152	170088	171752
MPEG-4	228224	229984	228872	217296	215168	206696	215816	224920	220312

表一 b 编好的 football 序列中 P 帧的信噪比数

Table1.b PSNR results of football

PSNR	P1	P2	P3	P4	P5	P6	P7	P8	P9
JM-4	33.9475	34.086	34.4213	34.3571	34.2803	34.2727	34.334	34.3246	34.1262
MPEG-4	34.3138	34.4932	34.6344	34.6035	34.4965	34.5226	34.5781	34.5848	34.3551
Frame	P10	P11	P12	P13	P14	P15	P16	P17	P18
JM-4	34.1262	34.1189	34.1472	34.1911	34.1281	34.2404	34.2572	34.169	34.0863
MPEG-4	34.3477	34.3717	34.2867	34.2702	34.3497	34.3599	34.2829	34.2486	34.4113

试验二:

测试序列: 720X576 的 mobile 序列一 此序列的特点是运动简单,图像复杂度高

测试条件: 编码次序为 IPPP, JM-4 选择 RD 优化、UVLC 编码; MPEG-4 使用微软编码器,两者都没有加入码率控制算法

表二 a 编好的 mobile 序列中 P 帧占用的比特数

Table2.a bits results of mobile

Frame	P1	P2	P3	P4	P5	P6	P7	P8	P9
JM-4	129728	125528	106720	120656	159056	135968	122848	140608	150392
MPEG-4	269120	308472	299472	305432	315408	311392	305736	295392	252112
Frame	P10	P11	P12	P13	P14	P15	P16	P17	P18
JM-4	146920	144176	138680	154344	164232	152336	145928	144920	143592
MPEG-4	256272	268584	281832	276768	238120	230280	226176	216032	218232

表二 b 编好的 mobile 序列中 P 帧的信噪比数

Table2.b PSNR results of mobile

PSNR	P1	P2	P3	P4	P5	P6	P7	P8	P9
JM-4	32.82735	32.9829	32.9063	32.9188	32.8513	32.7916	32.7343	32.7372	32.7312
MPEG-4	32.9814	32.8711	32.8115	32.7103	32.651	32.6129	32.6534	32.678	32.6383
Frame	P10	P11	P12	P13	P14	P15	P16	P17	P18
JM-4	32.7432	32.7268	32.7205	32.7556	32.7952	32.77	32.7263	32.7433	32.7539
MPEG-4	32.549	32.5433	32.6022	32.6082	32.5531	32.5663	32.6042	32.6647	32.6408

从以上的试验结果可以看出，在编码标准清晰度序列时，1) JM-4 性能比 MPEG-4 的性能要好；2) 图像本身复杂度越高，JM-4 效果越好。

下面的数据是摘录于 JVT 组织的 JM-4 与 MPEG-2 的对比试验结果，参见 JVT-F007.doc。

表三 a 720p(60)序列的对比结果

Table3.a results of 720p(60)

720p(60)		6 Mbps			10 Mbps		
		AVC	MPEG-2	MPEG-2	AVC	MPEG-2	MPEG-2
Crew	Encoding by	Aachen	InterVideo	UBV-RD	Aachen	InterVideo	UBV-RD
	Y PSNR	38.3004	36.6775	36.57	39.7971	38.8961	38.31
	U PSNR	43.1288	41.3906	41.47	43.891	43.079	42.73
	V PSNR	43.6108	41.227	41.32	44.8836	43.7365	43.17
Harbor	Encoding by	TUM	InterVideo	UBV-RD	TUM	InterVideo	UBV-RD
	Y PSNR	33.83	31.8597	31.56	35.64	34.2488	33.73
	U PSNR	42.3	40.7634	40.4	43.25	41.9087	41.44
	V PSNR	44.25	43.2475	40.84	45.25	44.2446	43.72
Sailormen	Encoding by	Aachen	InterVideo		Aachen	InterVideo	
	Y PSNR	35.5617	33.5394		36.7817	35.3011	
	U PSNR	39.3832	37.3703		40.3978	38.9341	
	V PSNR	41.162	39.5256		42.0584	40.7499	
Night	Encoding by	Aachen	InterVideo	UBV-RD	Aachen	InterVideo	UBV-RD

	Y PSNR	36.0869	33.1054	32.21	37.7148	35.5077	34.53
	U PSNR	40.0545	37.4014	36.86	41.3584	39.3344	38.61
	V PSNR	42.0156	39.3501	38.46	43.2914	41.1223	40.18
City	Encoding by	Aachen	InterVideo	UBV-RD	Samsung	InterVideo	UBV-RD
	Y PSNR	37.1278	34.3173	33.98	38.78	36.2475	35.82
	U PSNR	44.272	42.7499	42.51	45.01	43.5698	43.39
	V PSNR	44.7502	43.2189	42.68	45.45	44.016	43.84

表三 b 1080i(30)序列的对比结果

Table3.b results of 1080i(30)

1080i (30)		10 Mbps			20 Mbps		
		AVC	MPEG-2	MPEG-2	AVC	MPEG-2	MPEG-2
Flamingo	Encoding by	Sony	Broadcom	InterVideo	Sony	Broadcom	InterVideo
	Y PSNR	33.76	27.74	30.53	36.69	32.35	34.42
	U PSNR	38.37	36.94	37.94	39.32	39.28	39.68
	V PSNR	38.22	35.19	36.97	39.4	38.65	39.45
	Bitrate	13.42776	12.14	10.05	23.43612	20.11	20
Mountain	Encoding by	Sony	Broadcom	InterVideo	Sony	Broadcom	InterVideo
	Y PSNR	33.82	32.22	34.45		35.54	37.09
	U PSNR	39.17	40	40		41.52	41.05
	V PSNR	42.62	43.76	43.54		44.54	44.15
	Bitrate	8.32238	10.1	10		10.09	19.97
Kayak	Encoding by	Sony	Broadcom	InterVideo	Sony	Broadcom	InterVideo
	Y PSNR		25.4	26.26	30.89	28.21	29.84
	U PSNR		37.51	37.8	39.74	40.31	40.37
	V PSNR		37.17	37.64	39.84	40.62	40.59
	Bitrate		13.19	10.59	18.35822	20.08	20
Crossroad	Encoding by	Sony	Broadcom	InterVideo	Sony	Broadcom	InterVideo
	Y PSNR	32.17	26.48	30.95	36.36	32.86	35.03
	U PSNR	37.9	36.87	38.13	39.59	39.74	40.17
	V PSNR	38.23	37.08	38.64	40.05	40.15	40.7
	Bitrate	9.15461	10.12	10.07	19.63668	20.14	20

以上的试验数据可以看出，H.264 在编码标准清晰度的序列时要明显好于 MPEG-2，在
高清晰度的序列时，对于运动不剧烈的图像有好的性能，运动剧烈的图像则优势不明显，
这是因为 MPEG-2 有针对不同档次不同水平的序列优化造成的。

致 谢

在本论文即将付梓之际，我谨向尊敬的导师余松煜教授致敬！我在攻读硕士学位的两年半时间里，得到余老师的悉心指导和亲切关怀。余老师留给我的印象不仅是和蔼可亲、学识渊博和严谨的治学精神，更是他那忘我工作、废寝忘食的敬业精神。余老师的深深教诲使我在两年半时间里受益匪浅，并且将激励我在以后的工作中不断前进，再次衷心感谢余老师为我所付出的辛勤汗水！

我很高兴自己在硕士期间身处于一个团结友爱、奋发向上的团体中。我的同学和课题组的各位同事在作者的研究工作中给予了很大的支持，正是由于很多有益的讨论使我受益匪浅。他们是我的同学陈峰硕士、王建国硕士、傅红燕硕士、马智勇硕士、林梦冬硕士、王尧硕士、鞠啸东硕士、陈亮硕士。在我参加的几个课题组中，总是能从中学到很多东西。陈川博士的勤奋和聪明，曾昭平博士的踏实肯干、为人热情忠厚都给我留下了深深的影响。

还要感谢周源华老师、孙军老师、周军老师、王嘉老师、张瑞老师、薛震民老师等的帮助和指导，感谢张志明提供的专业资料和有益的探讨。此外，还有同所的许多其他师生予以不同方面的帮助，恕不在此一一列出他(她)们的名字。

在结束本篇论文之际，我要感谢我的父母对我成长的关心和鼓励。他们的支持和激励鼓舞是我完成硕士学业的基本保证。

二零零三年一月

作者攻读硕士学位期间撰写和发表的论文

1. 王涛, 余松煜. “结合 DSP 与 PC 的 G.723.1 语音编码技术”, 《计算机工程》, 2002, 12

H. 264视频编码算法研究

作者: [王涛](#)
学位授予单位: [上海交通大学](#)
被引用次数: 8次

引证文献(8条)

1. [张真](#), [黄登山](#), [汤加跃](#) 视频编码中帧内预测算法研究及性能比较[期刊论文]-[计算机测量与控制](#) 2007(2)
2. [王玉新](#) 基于H. 264视频流的运动目标的检测[学位论文]硕士 2006
3. [黄志鹏](#) 基于新视频编码标准的运动估计算法研究[学位论文]硕士 2006
4. [杨坚锐](#) 基于TMS320DM642的H. 264视频编解码系统的研究和实现[学位论文]硕士 2005
5. [陈光法](#) 基于MAP-CA BSP-15处理器AVS视频编码器的设计与实现[学位论文]硕士 2005
6. [吴参毅](#) 基于媒体处理器的MPEG-4网络视频编码器[学位论文]硕士 2005
7. [肖强](#) 基于无线移动设备的交互式多媒体E-learning系统[学位论文]硕士 2005
8. [刘晓峰](#) H. 264/AVC 编码器帧内预测算法优化及DSP实现[学位论文]硕士 2005

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y490746.aspx