

基于 H.264 标准的视频编解码器的 实现、性能研究与优化

Realization, Performance Research and Optimization
of Video Codec Based on H.264


姓名：孙同伦

学号：SA02006032

导师：尹 东 副教授

专业：信号与信息处理

中国科学技术大学电子工程与信息科学系
2005 年 5 月


2005.5.19

中文内容摘要

随着信息时代的到来,人们对现有通信网络条件下的多媒体服务提出了更高的要求,如更高质量要求的可视电话和视频会议,具有更高清晰度的视频存储和播放,无线网络中的视频通信等。这些应用中一个关键的技术环节是视频图像的压缩和解压缩。

视频压缩算法的日益成熟,编码标准的相继推出,有力的支持了视频通信的发展。H.264 是关于视频编码的最新国际标准,由国际电信联盟 ITU-U(International Telecommunication Union, Telecommunication Standardization Sector) 和国际标准化组织 ISO(International Organization for Standardization)/国际电工委员会 IEC(International Electrotechnical Commission)共同制定的联合标准。新一代标准 H.264 可以得到更好的压缩图像效果、拥有更多的功能和更大的灵活性,它可以在保证相同的图像质量的前提下,比 H.263(Baseline)和 MPEG-4(SP)节省多达 50%的码率。但是由于 H.264 中引入了多种新的编码技术,使得编码器的计算量急剧增加。如何降低 H.264 实现复杂度,使得它可以有效的应用在实际的通信系统中,已经成为学术界普遍关注的课题。

本文针对如何降低 H.264 编码器复杂度的关键问题进行一些探讨,在深入研究 H.264 的标准文档、性能和计算复杂度的基础上,对 H.264 的帧内/帧间预测模式选择算法进行了研究,采用基于局部边缘方向信息的帧内预测模式选择快速算法,提出了基于大块分裂的帧间预测模式快速选择算法。采用本文中的快速模式选择算法可以有效的减少预测的候选模式数,在作 IBPBP 编码时可以节约大约 50%-70%的编码时间,同时图像质量和码率基本不变。同时,本文中又对 H.264 的编解码器的参考软件的主要耗时模块与程序结构进行了优化调整,有效减少计算量与降低 CPU 的开销,提高了编解码效率。

本文第二章简要说明了视频数据压缩的必要性和可能性,并介绍了视频压缩编码的基本方法,基于块匹配混合编解码器的典型结构以及现有主要视频编码标准的特点及其应用。

第三章首先给出了 H.264 视频编解码器的结构框图,对其中的主要功能模块及其包含的新技术进行了详细论述。接着对 H.264 视频编解码器的性能与复杂性进行了分析,对主要模块进行耗时统计以指出编解码器的瓶颈所在,为后续优化提供指引。

第四章根据 H.264 的特点,采用基于局部边缘方向信息的帧内预测模式选择快速算法、提出了基于大块分裂的帧间预测模式快速选择算法,在保证编码图像质量和码率基本不变的情况下,有效降低计算复杂度。

最后,最后以 JVT 发布的 H.264 参考模型 JM9.4 为基础,使用 Intel 的 MMX/SSE 技术对 H.264 编解码器的主要的耗时模块进行了改写,包括: SAD 计算、正反整数变换、像素内插等,并对部分程序结构进行了调整,包括:程序的内存分配与 I/O 操作,程序的初始化,以及减少函数的调用,优化程序中的循环结构,尽量减少计算量,减少 CPU 的开销,有效的提高了编解码效率。

关键词: H.264; 视频编解码; 帧内/帧间预测模式; 算法优化

Abstract

With the coming of IT era, people put forward higher demand of the multimedia service under present condition of communication network. For example, the higher quality video phone and video conference, video store and display with higher resolution, video communication in wireless network, etc. But a key technology in all these applications is video coding and decoding.

The development of video communication is strongly supported by the advances in the research on video compression algorithms and release of coding standards. H.264 is the emerging video coding standard of ITU-T and ISO/IEC. Compared with prior standards, H.264 has better compression performance, more functions and more agility. Especially H.264 can provide equivalent objective quality at a data rate about 50% lower than that required by H.263 (Baseline) and MPEG-4 (SP). However, the algorithm complexity of H.264 dramatically increases due to the adoption of a number of video coding tools. How to cut down the complexity of H.264 and make it be a practical standard in real communication has been a popular issue.

This paper finds out the major parts that lead to the high computation based on studying the standard documents, performance, and the computation complexity of H.264, proposes a fast intra-prediction mode selection method based on local edge information and a fast inter-prediction mode selection based on split procedures. Experimental results have demonstrated that the new method can achieve a reduction of 50%~70% encoding time on average, with a negligible average PSNR loss and a mere bit rate increase compared with the original H.264 reference software. In addition, the parts that fit to optimize by MMX/SSE instruction set are found out and realized, the structure of H.264 reference software is adjusted. So the complexity of H.264 encoding is reduced obviously.

Chapter Two discusses the need and the possibility of video coding, video coding tools, a video codec model, and main features of prior standards.

Chapter Three firstly shows the model of H.264 video codec, provides a detailed analysis of the individual coding gains provided by several key features of the standard, and illustrates the improved coding efficiency enabled by H.264 relative to prior standards. Then the bottleneck affecting the speed is pointed out based on the analysis of the architecture of H.264 codec.

Chapter Four proposes a fast intra-prediction mode selection method based on local edge information and a fast inter-prediction mode selection based on split procedures, with rate-distortion optimization that reduces complexity of the H.264 encoding with a negligible average PSNR loss and a mere bit rate increase.

In the end, the parts that fit to optimization by MMX/SSE instruction set are found out and realized. Then, optimization methods in data structure, memory allocation and program design are proposed and implemented.

Key: H.264; video codec; intra/inter-prediction mode; algorithm optimization.

致 谢

值此论文完成之际，我要对我的导师尹东老师表示深深的谢意。在尹老师的悉心关怀和精心指导下，我的论文工作得以顺利的完成。在攻读硕士学位的期间里，尹老师一直以他对学术研究孜孜以求、认真求是和积极奋发的精神指引着我，尹老师的悉心指导帮助了我在科研能力方面的成长。在此论文即将完成之际，谨向导师致以深深的敬意和衷心的感谢！

感谢实验室的徐守时老师、吴秀清老师、陈学俭老师。他们的辛勤工作为实验室创造了一个良好的环境，他们勤勉的工作作风和对事业执着的追求精神将是我永远的学习的榜样。

感谢实验室的同学们，特别是张勋、侯蕾、任志清、唐群元、谭剑峰、王庆临等同学，感谢他们在我毕业论文的研究工作中对我的热情帮助，与他们的讨论令我受益匪浅。还要感谢我的老乡兼高中同学陈尧，是他的帮助让我有了一次珍贵的机会去北京永新同方公司实习，感谢永新同方公司的领导同事杜永明、洪均、赵欣等给予我的指导和帮助。

特别要感谢我的父母、哥哥、亲人们，感谢他们给予我无私的爱和无限的关怀。从小学到大学，从考研到读研，他们时刻都是我最坚强的后盾，我的一点一滴的进步都有他们的一份子。

最后要感谢各位对论文进行评审的专家学者，他们在百忙中对我的论文不吝给予指正，在此表示由衷的感谢。

孙同伦
2005年5月

第1章 绪论

1.1 引言

近年来,多媒体技术迅猛发展,人们对视频编码技术及其在网络中的应用如:可视电话、会议电视、远程监控等越来越感兴趣。但由于受信道传输带宽和存储容量的限制,多媒体技术的各项应用都离不开高效的视频图像压缩算法。数据量巨大的视频图像信息处理技术成为多媒体通信中研究的热点问题^[1.1]。

二十世纪九十年代以来, JPEG、H.261、H.263、MPEG-1、MPEG-2、MPEG-4等图像多媒体国际标准相继出台,它们成功的应用于多种实用场合。但是,随着IT技术的迅猛发展、Internet的普及,以及对视频传输需求不断提高,现有的这些视频压缩标准明显存在着一些缺陷和局限性。为了克服这些缺点,结合目前软硬件发展的情况,ISO/IEC和ITU-T两大标准组织联合制定了新的视频编码标准H.264/AVC。新一代标准H.264可以得到更好的压缩图像效果、拥有更多的功能和更大的灵活性^[1.2,1.3]。

H.264作为新一代的视频编解码标准,采用了成熟的技术,在追求更高的编码效率和简洁的表达形式的同时,也提供了非常好的视频质量,是当前最高效的视频压缩方法,具有广阔的应用前景^[1.4-1.7]。

新一代标准H.264可以得到更好的压缩图像效果、拥有更多的功能和更大的灵活性,它可以在保证相同的图像质量的前提下,比H.263(Baseline)和MPEG-4(SP)节省多达50%的码率。但是由于H.264中引入了多种新的编码技术,使得编码器的计算量急剧增加。如何降低H.264实现复杂度,进一步提高编解码器的效率以及高效新算法的采用,使得它可以有效的应用在实际的通信系统中,已经成为学术界普遍关注的课题^[1.8-1.10]。

本论文的研究工作正是在充分理解H.264视频编码标准的背景、难点、改进点的基础上展开,研究了这种新一代视频编解码标准和以往的视频编解码标准不同的地方和新的技术应用以及其相关的一些问题,针对如何降低H.264编码器复杂度的关键问题进行一些探讨。在深入研究H.264的标准文档、性能和计算复杂度的基础上,对H.264的帧内/帧间预测模式选择算法进行了研究,采用基于局部边缘方向信息的帧内预测模式选择快速算法,提出了基于大块分裂的帧间预测模式快速选择算法。采用本文中的快速模式选择算法可以有效的减少预测的候选模式数,在作IBPBP编码时可以节约大约50%-70%的编码时间,同时图像质量和码率基本不变。同时,本文中又对H.264的参考软件编解码器的主要耗时模块与程序结构进行了优化调整,有效减少计算量与降低CPU的开销,提高了编解码效率。

1.2 国内外研究现状

1.2.1 H.264 视频编码标准

H.264是目前关于视频编码的最新国际标准,其主要特点如下^[1.11,1.12]:

- 高压缩率:在给定的固定精度下,比现有的其他视频标准节省50%以上的

比特率;

- 采用简洁的设计方式: 语法描述简单, 避免过多的选项和配置, 尽量利用现有的编码模块;
- 灵活的延时处理: 对不同的业务, 灵活的利用相应的时延限制;
- 错误恢复: 加强对误码和丢包的处理, 增强解码器的差错恢复能力;
- 提高网络适应性: 采用网络友好的结构和语法, 以适应 IP 网络、移动网络的应用;
- 在编解码器中采用复杂度可分级设计, 在图像质量和编码处理之间可分级, 以适应高和低复杂性的应用;
- H.264 编码基本框架的使用无需版权。

H.264 从开始制定时, 很多公司就开始着手相关产品的研发, 现在已经出现了很多基于 H.264 标准的产品。Envivio 公司 2004 年发布的 4-Caster™ 4550 编码器提供了全分辨率、全帧速率视频的实时 MPEG-4/H.264 编码能力 (<http://www.envivio.cn/products/4caster.html>)。DG2L 公司 2005 年发布的 Phoenix Live Broadcast Encoder 能够实现 IPTV 应用中的 H.264 实时编码 (<http://www.ivast.com/products/lbe.html>)。LSI logic 公司 2005 年发布的最新实时编码器 H.264 VLE4000 能够实现在 Main Profile@Level 3 框架下的实时编码 (http://www.lsilogic.com/products/video_broadcasting/vle4000.html)。由于 H.264 的算法复杂度较大, 现有的各种产品大部分都是基于专用硬件实现。

1.2.2 H.264 视频编解码器的快速算法研究

目前国内外关于 H.264 的研究工作主要集中在如何提高编码速度和编码效率, H.264 码率的传输以及 H.264 码率控制策略等。文献[1.13]提出了新的运动估计和模式选择优化策略; 文献[1.14,1.15]讨论了快速帧内预测模式选择; 文献[1.16-1.18]讨论了快速帧间模式选择; 文献[1.19,1.20]讨论了多帧参考的快速算法; 文献[1.21,1.22]讨论了码率控制的方法; 文献[1.23,1.24]讨论了在容易发生数据包丢失的网络上进行 H.264 码率传输的问题; 文献[1.25]介绍了在无线环境下的常用编码和传输技术; 文献[1.26,1.27]讨论了 H.264 编解码器的优化。

1.3 论文主要工作及结构安排

1.3.1 论文工作

H.264 是国际上关于视频编码的最新国际标准, 拥有优秀的编码效率和广阔的应用前景, 必定会广泛应用在 DTV/HDTV 和 IP 视频、移动通信等诸多领域。本文以 H.264 为研究对象, 在深入研究 H.264 的标准文档、性能和计算复杂度的基础上, 对 H.264 的帧内/帧间预测模式选择算法进行了研究, 采用基于局部边缘方向信息的帧内预测模式选择快速算法, 提出了基于大块分裂的帧间预测模式快速选择算法。采用本文中的快速模式选择算法可以有效的减少预测的候选模式数, 在作 IBPBP 编码时可以节约大约 50%-70% 的编码时间, 同时图像质量和码率基本不变。同时, 本文中又对 H.264 的参考软件编解码器的主要耗时模块与程序结构进行了优化调整, 有效减少计算量与降低 CPU 的开销, 提高了编解码效率。

1.3.2 论文结构安排

比特率;

- 采用简洁的设计方式: 语法描述简单, 避免过多的选项和配置, 尽量利用现有的编码模块;
- 灵活的延时处理: 对不同的业务, 灵活的利用相应的时延限制;
- 错误恢复: 加强对误码和丢包的处理, 增强解码器的差错恢复能力;
- 提高网络适应性: 采用网络友好的结构和语法, 以适应 IP 网络、移动网络的应用;
- 在编解码器中采用复杂度可分级设计, 在图像质量和编码处理之间可分级, 以适应高和低复杂性的应用;
- H.264 编码基本框架的使用无需版权。

H.264 从开始制定时, 很多公司就开始着手相关产品的研发, 现在已经出现了很多基于 H.264 标准的产品。Envivio 公司 2004 年发布的 4-Caster™ 4550 编码器提供了全分辨率、全帧速率视频的实时 MPEG-4/H.264 编码能力 (<http://www.envivio.cn/products/4caster.html>)。DG2L 公司 2005 年发布的 Phoenix Live Broadcast Encoder 能够实现 IPTV 应用中的 H.264 实时编码 (<http://www.ivast.com/products/lbe.html>)。LSI logic 公司 2005 年发布的最新实时编码器 H.264 VLE4000 能够实现在 Main Profile@Level 3 框架下的实时编码 (http://www.lsillogic.com/products/video_broadcasting/vle4000.html)。由于 H.264 的算法复杂度较大, 现有的各种产品大部分都是基于专用硬件实现。

1.2.2 H.264 视频编解码器的快速算法研究

目前国内外关于 H.264 的研究工作主要集中在如何提高编码速度和编码效率, H.264 码率的传输以及 H.264 码率控制策略等。文献[1.13]提出了新的运动估计和模式选择优化策略; 文献[1.14,1.15]讨论了快速帧内预测模式选择; 文献[1.16-1.18]讨论了快速帧间模式选择; 文献[1.19,1.20]讨论了多帧参考的快速算法; 文献[1.21,1.22]讨论了码率控制的方法; 文献[1.23,1.24]讨论了在容易发生数据包丢失的网络上进行 H.264 码率传输的问题; 文献[1.25]介绍了在无线环境下的常用编码和传输技术; 文献[1.26,1.27]讨论了 H.264 编解码器的优化。

1.3 论文主要工作及结构安排

1.3.1 论文工作

H.264 是国际上关于视频编码的最新国际标准, 拥有优秀的编码效率和广阔的应用前景, 必定会广泛应用在 DTV/HDTV 和 IP 视频、移动通信等诸多领域。本文以 H.264 为研究对象, 在深入研究 H.264 的标准文档、性能和计算复杂度的基础上, 对 H.264 的帧内/帧间预测模式选择算法进行了研究, 采用基于局部边缘方向信息的帧内预测模式选择快速算法, 提出了基于大块分裂的帧间预测模式快速选择算法。采用本文中的快速模式选择算法可以有效的减少预测的候选模式数, 在作 IBPBP 编码时可以节约大约 50%-70% 的编码时间, 同时图像质量和码率基本不变。同时, 本文中又对 H.264 的参考软件编解码器的主要耗时模块与程序结构进行了优化调整, 有效减少计算量与降低 CPU 的开销, 提高了编解码效率。

1.3.2 论文结构安排

1.3.2 论文结构安排

比特率;

- 采用简洁的设计方式: 语法描述简单, 避免过多的选项和配置, 尽量利用现有的编码模块;
- 灵活的延时处理: 对不同的业务, 灵活的利用相应的时延限制;
- 错误恢复: 加强对误码和丢包的处理, 增强解码器的差错恢复能力;
- 提高网络适应性: 采用网络友好的结构和语法, 以适应 IP 网络、移动网络的应用;
- 在编解码器中采用复杂度可分级设计, 在图像质量和编码处理之间可分级, 以适应高和低复杂性的应用;
- H.264 编码基本框架的使用无需版权。

H.264 从开始制定时, 很多公司就开始着手相关产品的研发, 现在已经出现了很多基于 H.264 标准的产品。Envivio 公司 2004 年发布的 4-Caster™ 4550 编码器提供了全分辨率、全帧速率视频的实时 MPEG-4/H.264 编码能力 (<http://www.envivio.cn/products/4caster.html>)。DG2L 公司 2005 年发布的 Phoenix Live Broadcast Encoder 能够实现 IPTV 应用中的 H.264 实时编码 (<http://www.ivast.com/products/lbe.html>)。LSI logic 公司 2005 年发布的最新实时编码器 H.264 VLE4000 能够实现在 Main Profile@Level 3 框架下的实时编码 (http://www.lsillogic.com/products/video_broadcasting/vle4000.html)。由于 H.264 的算法复杂度较大, 现有的各种产品大部分都是基于专用硬件实现。

1.2.2 H.264 视频编解码器的快速算法研究

目前国内外关于 H.264 的研究工作主要集中在如何提高编码速度和编码效率, H.264 码率的传输以及 H.264 码率控制策略等。文献[1.13]提出了新的运动估计和模式选择优化策略; 文献[1.14,1.15]讨论了快速帧内预测模式选择; 文献[1.16-1.18]讨论了快速帧间模式选择; 文献[1.19,1.20]讨论了多帧参考的快速算法; 文献[1.21,1.22]讨论了码率控制的方法; 文献[1.23,1.24]讨论了在容易发生数据包丢失的网络上进行 H.264 码率传输的问题; 文献[1.25]介绍了在无线环境下的常用编码和传输技术; 文献[1.26,1.27]讨论了 H.264 编解码器的优化。

1.3 论文主要工作及结构安排

1.3.1 论文工作

H.264 是国际上关于视频编码的最新国际标准, 拥有优秀的编码效率和广阔的应用前景, 必定会广泛应用在 DTV/HDTV 和 IP 视频、移动通信等诸多领域。本文以 H.264 为研究对象, 在深入研究 H.264 的标准文档、性能和计算复杂度的基础上, 对 H.264 的帧内/帧间预测模式选择算法进行了研究, 采用基于局部边缘方向信息的帧内预测模式选择快速算法, 提出了基于大块分裂的帧间预测模式快速选择算法。采用本文中的快速模式选择算法可以有效的减少预测的候选模式数, 在作 IBPBP 编码时可以节约大约 50%-70% 的编码时间, 同时图像质量和码率基本不变。同时, 本文中又对 H.264 的参考软件编解码器的主要耗时模块与程序结构进行了优化调整, 有效减少计算量与降低 CPU 的开销, 提高了编解码效率。

1.3.2 论文结构安排

本论文由以下七个部分组成:

第一章 介绍了课题背景、国内外相关领域的研究现状、本课题的主要研究内容、本课题的意义以及论文结构。

第二章 简要说明了视频数据压缩的必要性和可能性,并介绍了视频压缩编码的基本方法,基于块匹配混合编解码框器的典型结构以及现有主要视频编码标准的特点。

第三章 首先给出了 H.264 视频编解码器的结构框图,对其中的主要功能模块,如小尺寸多模式帧内预测、多尺寸高精度多帧参考帧间预测、整数块变换与可变步长量化、去块效应滤波、增强的熵编码以及 SP-slice 和 SI-slice 等模块及其包含的新技术进行了详细论述。接着对 H.264 视频编解码器的性能与复杂性进行了分析,对主要模块进行耗时统计以指出编解码器瓶颈所在,为后续优化提供指引。

第四章 根据 H.264 的特点,采用基于局部边缘方向信息的帧内预测模式选择快速算法、提出基于大块分裂的帧间预测模式快速选择算法,在保证编码图像质量和码率基本不变的情况下,有效降低计算复杂度。

第五章 以 JVT 发布的 H.264 参考模型 JM9.4 为基础,使用 Intel 的 MMX/SSE 技术对 H.264 编解码器的主要的耗时模块进行了改写,包括: SAD 计算、正反整数变换、像素内插等,并对部分程序结构进行了调整,包括:程序的内存分配与 I/O 操作,程序的初始化,以及减少函数的调用,优化程序中的循环结构,尽量减少计算量,减少 CPU 的开销,有效的提高了编解码效率。

第六章 “工作总结与展望”总结自本课题的研究工作,并对本课题今后进一步的研究方向进行讨论。

参考文献

- [1.1] 钟玉琢,王琪,贺玉文,基于对象的多媒体数据压缩编码国际标准—MPEG-4 及其校验模型.北京:科学出版社.2000 年 10 月.
- [1.2] Joint Video Team of ITU-t and ISO/IEC JTC 1, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification(ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC). JVT-G050, 2003.
- [1.3] Wiegand T., Sullivan G.J., Bjntegaard G., Luthra A., Overview of the H.264/AVC video coding standard. Circuits and Systems for Video Technology, IEEE Transactions on Volume 13, Issue 7, July 2003 Page(s):560-576.
- [1.4] Kamaci N., Altunbasak Y., Performance comparison of the emerging H.264 video coding standard with the existing standards. Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on Volume 1, 6-9 July 2003 Page(s):I - 345-8 vol.1.
- [1.5] Raja G, Mirza M.J., Performance comparison of advanced video coding H.264 standard with baseline H.263 and H.263+ standards. Communications and Information Technology, 2004. ISCIT 2004. IEEE International Symposium on Volume 2, Oct. 26-29, 2004 Page(s):743-746.
- [1.6] 周祥平,王嵩,新标准 H.264 的核心技术与应用前景.中国有线电视, 2004 No.15. Page(s):55-61.

- [1.7] 秦岭,王煜坚,李东新,吴镇扬, 视频编码标准 H.264 的主要技术特点及其应用前景. 微计算机应用,2004 Vol.25 No.4.Page(s):449-455.
- [1.8] Ostermann J., Bormans J., List P., Marpe D., Narroschke M., Pereira F., Stockhammer T., Wedi T., Video coding with H.264/AVC: tools, performance, and complexity. Circuits and Systems Magazine, IEEE Volume 4, Issue 1, First Quarter 2004 Page(s):7-28.
- [1.9] Tu-Chih Wang, Yu-Wen Huang, Hung-Chi Fang, Liang-Gee Chen. Performance analysis of hardware oriented algorithm modification in H.264. Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on Volume 3, 6-9 July 2003 Page(s):III - 601-4 vol.3.
- [1.10] 周敬利, 金毅, 余胜生, 郑俊浩, 基于 H.264 视频编码技术的研究. 华中科技大学学报(自然科学版), 2003 Vol.31 No.8.Page(s):32-34.
- [1.11] Iain E.G.Richardson, The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression-Video Coding for Next-generation Multimedia[M/OL].
<http://nmlab.zju.edu.cn/~mm/book/H.264.And.MPEG-4.Video.Compression.Video.Coding.For.Next.Generation.Multimedia.pdf>.
- [1.12] Kamaci, N., Altunbasak, Y., Performance comparison of the emerging H.264 video coding standard with the existing standards. Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on Volume 1, 6-9 July 2003 Page(s):I - 345-8 vol.1.
- [1.13] Jianfeng Xu, Zhibo Chen, Yun He, Efficient fast ME predictions and early-termination strategy based on H.264 statistical characters. Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on Volume 1, 15-18 Dec. 2003 Page(s):218 -222 Vol.1.
- [1.14] Feng PAN, Xiao LIN, Rahardja SUSANTO, Keng Pang LIM, Zheng Guo LI, Ge Nan FENG, Da Jun WU, and Si WU, Fast Mode Decision for Intra Prediction. JVT-G013, 7th Meeting: Pattaya II, Thailand, 7-14 March, 2003.
- [1.15] Zhang Yong-dong, Dai Feng, Lin Shou-xun, Fast 4×4 intra-prediction mode selection for H.264. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on , Volume: 2 , 27-30 June 2004. Pages:1151-1154 Vol.2.
- [1.16] Kim Y.-H., Yoo J.-W., Lee S.-W., Shin J., Paik J., Jung H.-K., Adaptive mode decision for H.264 encoder. Electronics Letters , Volume: 40 , Issue: 19 , 16 Sept. 2004, Pages:1172-1173.
- [1.17] K.P. Lim, S. Wu, D.J. Wu, S. Rahardja, X. Lin, F. Pan, ZG Li, Fast INTER Mode Selection. Doc. I020, Sep. 2003.
- [1.18] Jeyun Lee, Byeungwoo Jeon, Fast mode decision for H.264. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume 2, 27-30 June 2004 Page(s):1131-1134 Vol.2.
- [1.19] Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-Yi Chen, Shyh-Yih Ma, Chun-Fu Shen, Liang-Gee Chen, Analysis and reduction of reference frames

- for motion estimation in MPEG-4 AVC/JVT/H.264. Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on Volume 2, 6-9 July 2003 Page(s):II - 809-12 vol.2.
- [1.20] Yeping Su, Ming-Ting Sun, Fast multiple reference frame motion estimation for H.264. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume 1, 27-30 June 2004 Page(s):695-698 Vol.1.
- [1.21] Peng Yin, Boyce J., A new rate control scheme for H.264 video coding. Image Processing, 2004. ICIP '04. 2004 International Conference on Volume 1, 24-27 Oct. 2004 Page(s):449-452.
- [1.22] Li Z.G., Pan F., Lim K.P., Lin X., Rahardja S., Adaptive rate control for H.264. Image Processing, 2004. ICIP '04. 2004 International Conference on Volume 2, Oct. 24-27, 2004 Page(s):745-748.
- [1.23] Yuan Zhang, Wen Gao, Huijiang Sun, Qingming Huang, Yan Lu, Error resilience video coding in H.264 encoder with potential distortion tracking. Image Processing, 2004. ICIP '04. 2004 International Conference on Volume 1, 24-27 Oct. 2004 Page(s):163-166.
- [1.24] Yen-Lin Tung, Hsiu-Chen Shu, Jin-Jang Leou, An error detection and concealment scheme for H.264 video transmission. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume 3, 27-30 June 2004 Page(s):1735-1738 Vol.3.
- [1.25] Gnani S., Grangetto M., Magli E., Olmo G., Comparison of rate allocation strategies for H.264 video transmission over wireless lossy correlated networks. Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on Volume 2, 6-9 July 2003 Page(s):II - 517-20 vol.2.
- [1.26] Juyup Lee, Sungkun Moon, Wonyong Sung, H.264 decoder optimization exploiting SIMD instructions. Circuits and Systems, 2004. Proceedings. The 2004 IEEE Asia-Pacific Conference on Volume 2, Dec. 6-9, 2004 Page(s):1149-1152.
- [1.27] 张帆, 张旭东. H.264 编码器的 SSE-2 优化实现[J]. 中国图象图形学报 A, 2003, Vol.8 No.1:286-289.

第2章 视频压缩原理和方法

2.1 视频数据压缩的必要性与可能性

信息时代的重要特征是信息的数字化,数字化了的信息带来了“信息爆炸”。数字化了的视频和音频信号的数据量之大是非常惊人的。例如:

➤ SIF (Source Input Format) 格式数字电视图像, NTSC 制、彩色、4:4:4 采样:

每帧数据量: $352 \times 240 \times 3 = 253(\text{KB})$;

每秒数据量 (位率): $253 \times 30 = 7.603(\text{MB/s})$;

一张 CD-ROM 可存帧数: $650 \div 0.253 = 1.226\text{K}(\text{帧/张})$;

一张 CD-ROM 可存节目时间: $(650 \div 7.603) / 60 = 1.42(\text{分/张})$ 。

➤ CCIR (International Consultative Committee for Radio) 格式数字电视图像, PAL 制、4:4:4 采样:

每帧数据量: $720 \times 576 \times 3 = 1.24(\text{MB})$;

每秒数据量 (位率): $1.24 \times 25 = 31.3(\text{MB/s})$;

一张 CD-ROM 可存帧数: $650 \div 1.24 = 0.524\text{K}(\text{帧/张})$;

一张 CD-ROM 可存节目时间: $650 \div 31.1 = 20.9(\text{秒/张})$ 。

庞大的数据量无疑给存储器的存储容量、通信干线的信道传输率以及计算机的处理速度都增加了太大的负荷,成为多种视频应用技术实现的瓶颈问题。解决这一问题的办法,单纯通过扩大存储器容量、增加通信干线的传输率是不现实的。数据压缩技术是行之有效的办法,通过数据压缩手段把信息数量压下来,以压缩形式存储和传输,既节约了存储时间,又可以提高通信干线的传输效率,同时也使计算机实时处理视频、音频信息,以保证播放出高质量的视频、音频节目成为可能。

同时,视频图像在时间上和空间上都包含许多冗余信息,对视频图像进行压缩就是将庞大数据中的冗余信息去掉(去除数据之间的相关性),保留相互独立的信息分量。视频图像压缩的基本方法和思想主要概括为两点:

(1) 在空间方向上,图像数据采用静态图像的压缩算法来去掉冗余信息;

(2) 在时间方向上,图像数据压缩采用运动补偿(Motion Compensation)算法来去掉冗余信息。

数据压缩可以分为两种类型:无损压缩和有损压缩。无损压缩是指压缩后的数据在进行解压缩后与压缩前的数据完全一致,没有任何信息损失。无损压缩的方法是构筑在香农的信息理论基础之上的,根据信号的统计特性进行编码。有损压缩是指压缩后的数据重构后,与原始数据有所不同,但不会影响人们对原始资料所要表达的信息的理解。在多媒体压缩技术中,往往先对多媒体数据采用有损压缩算法进行处理,然后再对处理过的数据进行无损压缩。目前,广泛运用于视频编码技术的压缩算法主要有三种:统计编码、预测编码和变换编码。这三种编码方法在上个世纪 80 年代已经非常成熟,现在广泛应用在各种图像视频编码标准中^[2-1]。

2.2 视频压缩的基本编码技术

2.2.1 压缩空间冗余度—帧内编码

目前的编码方案都是把图像分块进行处理的。其方法是对每一个块经过变换编码以降低相邻像素之间的相关性。经过变换编码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。变换编码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,经过变换编码的反变换后,可以得到原始图像的重建图像。这种压缩技术称为帧内编码。

离散余弦变换 DCT (Discrete Cosine Transform) 编码性能高,计算复杂度适中,且有快速算法,是一种常用的变换编码技术,被广泛应用于 MPEG 和 H.26x 系列视频编码标准中。另一种 JPEG2000 中使用的小波变换 WT (Wavelet Transform),它是一种具有很好局域化的空-频分析的综合方法,针对不同类型的图像特点中的不同区域采用不同的空-频分辨率,有可能得到比其他变换方法更高的压缩比。

在 H.264 视频编码标准中,增加了单个图像的帧内预测编码,充分利用被预测块左边和上边已解码重构的邻近宏块像素对当前块进行预测,只将实际块与预测块的残差进行编码,帧内编码的像块信息可以用较少的比特数有效表达,减少了空间相关的冗余比特数^[22]。

2.2.2 压缩时间冗余度—帧间编码

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,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,下一步就是使用帧内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。这种压缩技术称为帧间编码。

2.2.3 运动估计和运动补偿

运动补偿可以把帧间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。运动补偿预测通常是以 16×16 宏块为预测单元,把当前宏块认为是先前某一时刻图像宏块的位移,位移的内容包括运动方向和运动幅度。所以运动补偿预测是用先前的局部图像,来预测当前的局部图像, 16×16 的运动矢量块是预测误差,它必须进行编码、传送,供解码时恢复图像用。图2-1 说明了当前帧和参考帧的匹配过程。

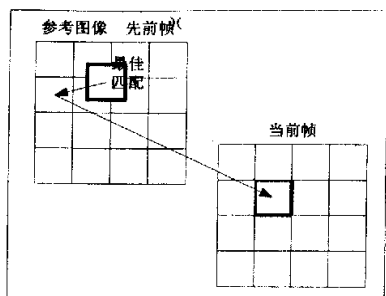


图2-1 当前帧和参考帧的匹配

H.264的运动补偿支持以往的视频编码标准中的大部分关键特性，而且灵活的添加了更多的功能，一是为了更精确的对图像进行预测补偿，新标准将宏块更进一步的划分为7种模式（ 16×16 、 16×8 、 8×16 、 8×8 、 8×4 、 4×8 、 4×4 ），小块模式的运动补偿为运动详细信息处理提高了性能，减少了方块效应，提高了图像的质量；二是采用 $1/4$ 像素精度的运动估值；三是提供可选的多参考帧预测功能，多帧参考的使用大大提高了对图像传输的容错性，抑制了错误在空间和时间上的蔓延^[2.2]。

2.2.4 熵编码

当帧经过帧内或帧间编码后，原始的视频信号就变成了一系列的变换系数、运动矢量以及辅助信息。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。

常用的熵编码方法有三种：一是利用概率分布特性的著名的哈夫曼(Huffman)编码方法，它是无失真变长编码的最佳编码方法；二是利用概率分布特性的熵编码方法——算术编码(Arithmetic Coding)；三是利用相关特性的游程(RLC)编码方法（具体的编码思想可参考文献[2.1]）。

H.264支持两种熵编码方案，即基于上下文的变长编码(CAVLC)和基于上下文的算术编码(CABAC)^[2.3]。前者更为简单，易于实现；后者则能进一步提高熵编码的性能，与CAVLC比较，在相同图像质量下编码视频信号使用后者将会使比特率减少10%~15%^[2.4]。

2.2.5 视频编解码器的典型结构

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，如图 2-2a、图 2-2b 所示。编码器采用了一种混合的信源压缩编码算法，即通过利用帧间预测去除时间上的冗余度，利用变换编码减少预测余量信号空间上的冗余度。解码器具有运动补偿能力，最后采用熵编码形成传输码流^[2.1]。

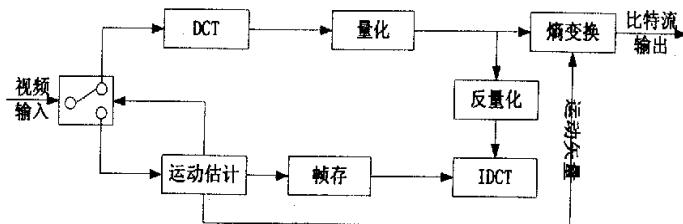


图2-2a 视频编码器的典型结构

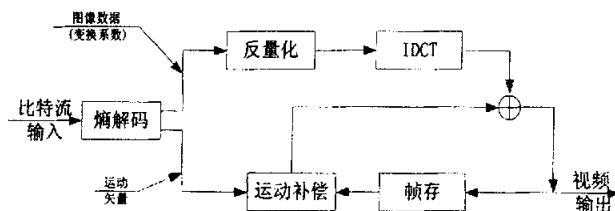


图2-2b 视频解码器的典型结构

2.3 现有视频压缩标准

多媒体计算机的关键技术是关于多媒体数据的压缩编码和解码算法。二十世纪 80 年代以来，国际标准化协会(ISO)、国际电子学委员会(IEC)和国际电信联盟—电信标准分部 (ITU-T) 等国际组织制定了两种重要的视频编码标准：H. 26x 标准和 MPEG 标准，取得了辉煌的业绩。ISO/IEC 的建议标准 MPEG 系列主要用于数字电视广播 DVD 等。ITU-T 的建议标准 H.26x 系列（如 H.261、H.263）主要用于实时视频通信，如视频会议、可视电话等。H.26x 系列标准与 MPEG 系列标准的演进如图 2-3 所示。

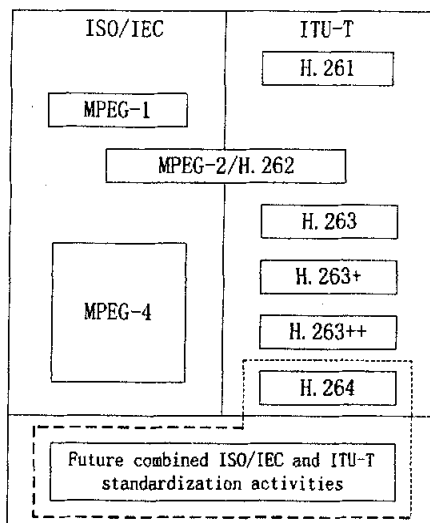


图 2-3 H.26x 标准与 MPEG 标准的演进

2.3.1 H.261

1990 年，ITU-T 公布了第一个视频编码标准 H.261，在帧间编码时采用了基于 16×16 的宏块和整像素精度的运动估计，而在帧内编码时采用了 8×8 数据块的 DCT 运算。这些算法有效的压缩了视频序列在时间和空间上的冗余度，使得 H.261 具有较高的压缩比，适用于 $p \times 64\text{Kbps}$ 的视听业务($p=1 \sim 32$)，可以用于 ISDN^[2,5]。

2.3.2 MPEG-1

MPEG-1 发布于 1992 年，称为运动图像和声音的压缩算法，是针对 1.5 Mbit/s 以下数据传输率的数字存储媒质运动图像及其伴音编码的国际标准，主

要包括系统、视频、音频等3部分,具有随机存取、快速正向/逆向搜索、逆向重播、视听同步、容错性、编码/解码延迟等功能。MPEG-1旨在达到家用录像系统(VHS)质量,对于不激烈的动作图像画面视频信号能获得较好的图像质量,但如果图像对象动作激烈时,图像有可能产生马赛克现象^[2.6,2.7]。

MPEG-1主要用于家用VCD,它使VCD取代了录像带。也可用于数字电话网络上的视频传输,例如非对称数字用户线路(ADSL)、视频点播(VOD),以及教育网络等。

2.3.3 MPEG-2

MPEG-2(ISO/IEC 13818)标准于1994年正式推出,主要用于高清晰度视频及其音频的编码。MPEG-2解决了MPEG-1不能满足的日益增长的多媒体应用对分辨率和传输率的要求,支持固定比特率传送、可变比特率传送、随机访问、分级编码、比特流编辑等功能。MPEG-2能够提供广播级的视像和CD级的音质^[2.8,2.9]。

MPEG-2主要用于数字存储媒体、高清晰度电视和数字视频广播等领域。从技术角度看,MPEG-2和MPEG-1在细节上存在着一些差别,MPEG-2可以说是MPEG-1的超集,在MPEG-1的基础上附加了一些特征帧格式和编码选项。MPEG-2音频与MPEG-1差别不大,其优于MPEG-1的地方主要体现在视频方面:设置了“按帧编码”和“按场编码”两种模式;在MPEG-2中,亮度信号与色度信号的比例由MPEG-1的4:2:0扩展为4:2:2或4:4:4;规定了四种图像预测和运动补偿方式;视频编码采用了分级编码技术,按类别的不同分为5种档次和4个不同的等级。

2.3.4 H.263

1996年3月ITU-T公布了H.263视频编码标准。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、高级预测、PB帧和基于语法的算术编码。在其后续版本H.263+中,提供了12种新的编码选项和其他特征,进一步提高了压缩编码性能。2000年ITU-T又推出了H.263++,对H.263又做了一些新的扩展,增加了一些新的特性,从而适应于各种网络环境,并增强差错恢复的能力^[2.10]。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。

2.3.5 MPEG-4

1999年1月MPEG-4(ISO/IEC 14496)第1版正式公布,1999年12月第2版公布。MPEG-4标准是一个适合多种多媒体应用的视听对象编码标准,它定义了一种框架而不是具体的算法,使视频产品具备更大的灵活性和可扩展性。MPEG-4采用基于对象的方式,通过对不同的视听对象(自然的或合成的)独立进行编码实现较高的压缩效率,同时可实现基于内容的交互功能,满足了多媒体应用中人机交互的需求。设备厂商在应用MPEG-4标准进行产品研发时,可根据应用领域的不同,适当选择标准工具的子集^[2.11,2.12]。

MPEG-4标准与MPEG-1和MPEG-2标准最根本的区别在于MPEG-4采用基于对象的方法,可以支持基于对象的互操作性。为适应通用访问,MPEG-4标准中加入了面向功能的传送机制,其中的错误鲁棒性、错误恢复的处理和速率控制等功能使编码能适应不同信道的带宽要求。MPEG-4编码系统是开放性质的,可随时加入新的编码算法模块,可根据不同的应用需求,现场配置解码

器。

MPEG-4 标准的开发目标是实现多媒体业务在各个领域的应用, 涉及面非常广泛, 不同的应用对应的码率、分辨率、质量和服务也不同。目前基于 MPEG-4 标准的应用有: 数字电视、实时多媒体监控、视频会议、低比特率下的移动多媒体通信、PSTN 网上传输的可视电话等。

表 2-1 中是现有视频编码标准及其应用的总结。

2.3.6 MPEG-7

MPEG-7(ISO/IEC 15938)一般称为多媒体内容描述接口, 侧重于媒体数据的信息编码表达, 是一套可用于描述多种类型的多媒体信息的标准^[2,13,2,14]。MPEG-7 定义了一个关于内容描述方式的互操作的框架, 它超越了传统的元数据概念, 具有描述从低级元素信号特征, 如颜色、形状、声音特质到关于内容搜集的高级结构信息的能力。MPEG-7 通过定义的一组描述符与多媒体信息的内容本身相关联, 支持用户快速有效的搜索其感兴趣的信息。通过给携有 MPEG-7 数据的多媒体信息加上索引, 用户就可方便的进行信息检索了。

MPEG-7 使多媒体信息查询更加智能化, 它对多媒体内容进行描述的功能对现有的 MPEG-1、MPEG-2、MPEG-4 标准将起到功能扩展的作用。MPEG-7 的应用可以分成三大类: 第一类是索引和检索类应用; 第二类是选择和过滤类应用, 可以帮助使用者只接受符合需要的信息服务数据; 第三类是与 MPEG-7 中“元(meta)”内容表达有关的专业化应用。

MPEG-7 目前已经实现了的应用包括数字图书馆、广播媒体选择、多媒体目录服务、多媒体编辑、远程教育、医疗服务、电子商务、家庭娱乐等, 涉及教育、新闻工作、旅游、娱乐、地理信息系统、医疗应用、商业、建筑等诸多领域。

2.3.7 MPEG-21

由于多媒体内容的处理涉及到了许多不同的平台, 关系到数字资产权利保护等诸多问题, 所以虽然目前用于多媒体内容的传输和使用的许多标准都已存在, 但想要建立一个统一的完整体系还有很多问题需解决。为了将不同的协议、标准和技术结合在一起, 使得用户可以在现有的各种网络和设备上透明的使用多媒体内容, 实现互操作 (Interoperability), 需要建立一个开放的多媒体框架, 所以出现了 MPEG-21 标准^[2,15,2,16]。

MPEG-21 标准(ISO/IEC21000)的正式名称是多媒体框架, 其制订工作于 2000 年 6 月开始。MPEG-21 将创建一个开放的多媒体传输和消费的框架, 通过将不同的协议、标准和技术结合在一起, 使用户可以通过现有的各种网络和设备透明的使用网络上的多媒体资源。MPEG-21 中的用户可以是任何个人、团体、组织、公司、政府和其它主体, 在 MPEG-21 中, 用户在数字项的使用上拥有自己的权力, 包括用户出版/发行内容的保护、用户的使用权和用户隐私权等。

MPEG-21 包括 7 个基本要素: 数字项声明 (Digital Item Declaration), 数字项识别和描述, 内容处理和使用, 知识产权管理和保护, 终端和网络, 内容表示, 事件报告。数字项是 MPEG-21 框架中的基本单元, 它由资源、原数据 (metadata) 和结构共同组成, 是一个带有标准化的结构化数字对象。要素中的资源包括了采用 MPEG-1、MPEG-2、MPEG-4 标准的多媒体信息。通过数字项的定义, MPEG-21 集成了 MPEG 系列的其它标准, 由此也可以看出, MPEG-21

是建立在其它标准的基础之上的。

MPEG-21 标准支持以下功能：内容创建，内容生产，内容分配，内容的消费和使用，内容的分组，知识产权管理和保护，内容识别和描述，用户权限，终端和网络资源提取，内容表示和事件报告等。该标准是从商业内容和与内容相关服务的前景等角度开发的，将同已有的其它 MPEG 系列标准等进行适当结合，从而使用户对视频、音频的处理更加方便和有效，最终为多媒体信息的用户在全球范围内提供透明而有效的视频通信应用环境。MPEG-21 的出台可以将现有的标准统一起来，消费者将可以自由使用音视频内容而不被不兼容的格式、编解码器、媒体数据类型及诸如此类的东西所干扰。

表 2-1 现有视频编码标准及其应用

		码率	应用
MPEG 标准	MPEG-1 视频	0.8kbps-1.5Mbps	VCD, CD-ROM, Internet
	MPEG-2 视频	3Mbps-10Mbps MP/ML	数字广播,
		16Mbps-50Mbps MP/HL	DVD, D-VHS
	MPEG-4 视频	50Mbps-100Mbps 422P@MP/HL	专业视频处理
		10kbps-384kbps SP@L1/2/3	Cellar, Internet
		10kbps-2Mbps CP@L1/2	交互式电视
		2Mbps-38Mbps MP@L2/3/4	
ITU-T 标准	50Mbps-1200Mbps Studio@L1/2/3/4		视频剪辑
	H.261	64kbps-1.5Mbps	视频电话(ISDN)
	H.262	与 MPEG-2 视频相同	
	H.263	10kbps-384kbps	视频电话(PSTN, Internet)

2.4 本章小结

本章中，首先简要说明了视频数据压缩的必要性和可能性，介绍了视频压缩编码的基本方法，如帧内编码、帧间编码、运动估计与运动补偿、熵编码等，接着给出了基于块匹配混合编解码框器的典型结构，然后对现有主要视频编码标准——H.26x 系列与 MPEG 系列标准的特点进行了分析，说明了 H.26x 系列标准与 MPEG 系列标准的演进，简单介绍了他们使用的关键技术以及应用范围。

参考文献

- [2.1] 钟玉琢, 王琪, 贺玉文. 基于对象的多媒体数据压缩编码国际标准——MPEG-4 及其校验模型. 北京: 科学出版社. 2000 年 10 月.
- [2.2] Iain E.G.Richardson, The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression-Video Coding for Next-generation Multimedia[M/OL].
<http://nmlab.zju.edu.cn/~mm/book/H.264.And.MPEG-4.Video.Compression.Video.Coding.For.Next.Generation.Multimedia.pdf>.
- [2.3] Marpe D, Blattermann G, et al, Video compression using context-based

- adaptive binary Arithmetic coding. 2002.
<http://bs.hhi.de/~marpe/cabac.html>.
- [2.4] Marpe D., Schwarz H., Blattermann G., et al. Context-Based Adaptive Binary Arithmetic Coding in JVT/H.26L. Proc. IEEE International Conference on Image Processing (ICIP-02), 2002 Vol.2. Page(s):513-516.
- [2.5] ITU-T, Video codec for audio/visual services at PX64 kbits. Draft ITU-T Recommendation H.261. Jan 1993.
- [2.6] ISO/IEC CD 11172, Coding of moving pictures and associated audio for digital storage media at up to 1.5Mbits/sec - Part 2: Coding of moving pictures information, Dec.1991.
- [2.7] Shlien, S., Guide to MPEG-1 audio standard. Broadcasting, IEEE Transactions on Volume 40, Issue 4, Dec. 1994 Page(s):206-218.
- [2.8] ISO/IEC 13818-2, Information technology – Generic coding of moving pictures and associated audio Part 2: Video, 1995.
- [2.9] Lauzon, D., Vincent, A., Limin Wang, Performance evaluation of MPEG-2 video coding for HDTV. Broadcasting, IEEE Transactions on Volume 42, Issue 2, June 1996 Page(s): 88-94.
- [2.10] ITU-T, Video Coding for low bit rate communication, ITU-T Recommendation H.263; version 1, Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.
- [2.11] ISO/IEC FDIS 14496-2, Information technology – Generic coding of audio-visual objects Part 2: Visual, Oct. 1998.
- [2.12] ISO/IEC JTC1/SC29/WG11 N3342, Overview of the MPEG-4 Standard. Mar.2000.
- [2.13] Martinez J.M., Standards - MPEG-7 overview of MPEG-7 description tools, part 2. Multimedia, IEEE Volume 9, Issue 3, July-Sept. 2002 Page(s):83 – 93.
- [2.14] Shih-Fu Chang, Sikora T., Puri A., Overview of the MPEG-7 standard. Circuits and Systems for Video Technology, IEEE Transactions on Volume 11, Issue 6, June 2001 Page(s):688 – 695.
- [2.15] Bormans J., Gelissen J., Perkis A., MPEG-21: The 21st century multimedia framework. Signal Processing Magazine, IEEE Volume 20, Issue 2, March 2003 Page(s):53 – 62.
- [2.16] Tseng B.L., Ching-Yung Lin, Smith J.R., Using MPEG-7 and MPEG-21 for personalizing video. Multimedia, IEEE Volume 11, Issue 1, Jan-Mar 2004 Page(s):42 – 52.

第3章 H.264 视频编解码器的设计与实现

3.1 H.264/AVC 视频编码标准概述

二十世纪九十年代以来, JPEG、H.263、MPEG-1、MPEG-2 等图像多媒体国际标准相继出台, 成功的应用于多种实用场合。但是, 随着 Internet 的普及, 以及对视频传输需求的不断提高, 现有的视频压缩标准明显存在着一些缺陷和局限。为了克服这些缺点, 结合软硬件发展趋势, 2001 年, ITU-T 的 VCEG 视频专家组和 ISO/IEC 的 MPEG 专家组组成联合专家组 JVT (Joint Video Team), 共同推进视频压缩技术的进步。2003 年 3 月 JVT 形成了最终标准草案分别提交 ITU-T 和 ISO/IEC。在 ISO/IEC 中, 该标准的正式名称为 MPEG-4 AVC (Advanced Video Coding) 标准, 作为 MPEG-4 标准的第十部分; 在 ITU-T 的正式名称为 H.264 标准^[3.1]。

一方面, H.264 把 H.263 中的一些已经证明行之有效的可选模式作为 H.264 中的基本模式固定下来, 例如先进的帧内编码模式和基于语法的算术编码模式等; 另一方面, H.264 又加入了一些新的研究成果, 在系统结构、编码方法、变换、运动估计和运动补偿等方面都提出了许多新的概念, 例如可选的 $1/4(1/8)$ 像素的运动估计、多模式的运动矢量估计、可交换帧、整数变换、多帧参考等, 从而使得 H.264 在压缩率上具有更好的性能, 编码性能也有显著提高^[3.2]。

3.2 H.264 视频编解码器结构

H.264 标准有三个框架 (profile), 分别是 Baseline profile、Main profile 和 Extend profile, 每个 profile 有相应的算法组成和语法结构^[3.3]。Baseline profile 是基本框架, 面向复杂度低, 传输延迟小的应用对象; Main profile 面向运动特性复杂、快速, 传输延迟大的应用对象; Extend profile 面向应用要求更高的对象, 如图 3-1 所示。

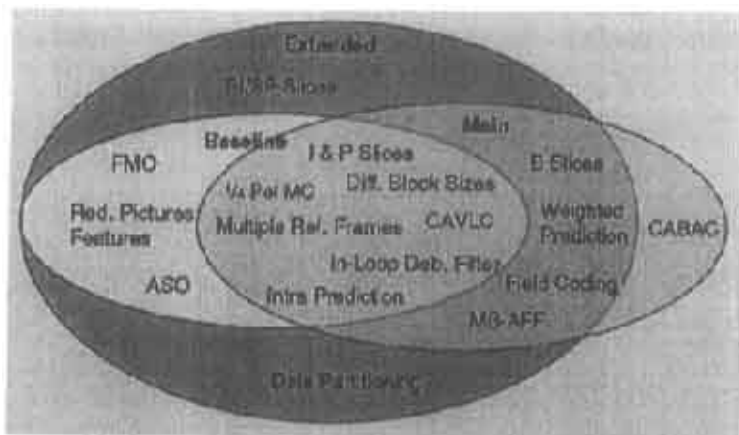


图 3-1 H.264 的 Baseline, Main and Extended profiles

H.264 在算法上分为两层：视频编码层（VLC: Video Coding Layer）和网络提取层（NAL: Net Abstraction Layer）。视频编码层负责高效的视频内容处理；网络提取层负责网络的分段格式封装数据，包括组帧、逻辑信道的指令、定时信息的利用或序列结束信号等。其码流结构的网络适应性强，容错性好，对误码和丢包处理能力强。VCL 和 NAL 之间定义了基于分组方式的接口，高效编码和良好的网络适应性分别由 VCL 和 NAL 完成。

H.264 标准的视频编码的基本结构与早先的 H.263 相似，采用图像预测和变换编码相结合的编码结构。H.264 编码器和解码器的基本结构如图 3-2a、图 3-2b 所示^[3.3]。

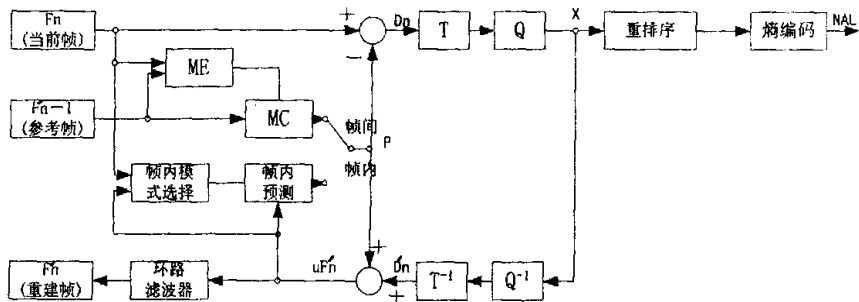


图 3-2a H.264 视频编码器

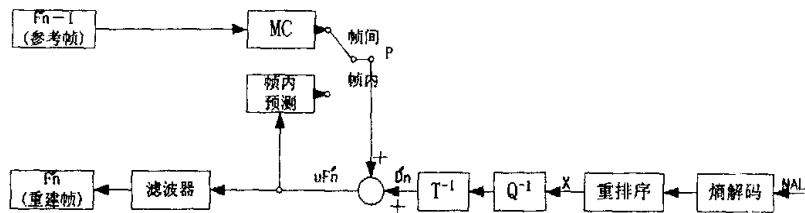


图 3-2b H.264 视频解码器

以编码为例简要叙述工作过程：对输入帧 F_n 编码时，实际上是对宏块（由原始图像依一定规则分割得到， 16×16 像素）编码。宏块编码分为帧内编码和帧间编码。帧内编码模式下，预测宏块 P 由当前帧中的已编码的宏块经解码、重构后，通过一定预测方法获得。帧间编码模式下， P 由一个或多个参考帧经运动补偿获得（如图 3-2a 所示，参考帧由已编码的帧 F'_{n-1} 表示）。 P 与当前宏块相减获得残差宏块 D_n ，经块变换和量化得到 X （一串量化过的变换参数）。 X 通过重排序和熵编码后获得新的参数，该参数与解码宏块所需的说明组成压缩后的流，送往 NAL 层进行传递和存储。同时 X 需要进行解码重构，为编码后续宏块做准备。 X 经反量化和逆变换产生宏块 D'_n （由于量化误差 D'_n 不同于 D_n ）， P 与 D'_n 相加产生重构宏块 $\mu F'_n$ 。环路滤波器（LoopFilter）用于降低块失真。通过组合一连串相应的宏块得到重构的参考帧 F'_n 。解码器的工作过程请参见参考文献[3.3]。

H.264 的基本算法是通过帧内预测和运动补偿消除空间冗余，通过帧间预测和运动补偿消除时间冗余，通过变换编码消除频域冗余，基本的功能模块与 H.263 相比没有变化。H.264 与 H.263 的区别主要表现在宏块（Macroblock）的分割，DCT 变换的算法，熵编码算法，以及其他一些技术细节上。

3.3 H.264 视频编码标准关键技术分析

H.264是建立在块匹配混合编码的基础上,采取一系列高效压缩编码技术的开放式标准。其亮点在于:基于宏块层率失真优化(Rate Distortion Optimization, RDO)理论,用小尺寸、多模式预测提高空域和时域压缩率,用整数变换量化提高频域压缩率,用增强的熵编码高效压缩符号冗余,详细规定了新标准内涵,又提供了灵活算法,以便根据不同的应用背景调整相应的算法和编码参数,力求在各种开放环境下实现最小失真的最佳熵编码,达到高压缩率、低时延、容错性好、编/解码复杂度可分级、对IP网和移动网适应性强等主要功能的设计目标^[3,4]。

为了实现这个核心技术,增强编码视频对复杂、异构的视频传输网的适应性,H.264在设计上将整个编/解码系统分成视频编码层(VCL, Video Coding Layer)和网络适配层(NAL, Network Abstraction Layer)两个具有不同功能的层次,并在每个层次上进行了一系列的性能提升。如多尺寸块模式帧内和帧间预测编码、1/4和1/8像素精度和多参考帧运动估计算法、4×4块的整数正交变换、可变步长量化、去除块效应的环路滤波器、基于内容的变长编码(CAVLC)和基于内容的自适应算术编码(CABAC)、引入完成流切换的刷新帧技术代替过去的分级编码进行码率控制和差错控制等^[3,4]。

3.3.1 小尺寸、多模式帧内预测

传统的对I帧图像进行帧内压缩是将单帧内相邻块之间存在的形状、纹理和彩色冗余,以及视频序列中存在的所有空域冗余(如动物后面的静止区)都进行直接变换编码,存在着较多的空间相关冗余。H.264增加了单个图像的帧内预测编码,充分利用被预测块左边和上边已解码重构的邻近宏块像素对当前块进行预测,只将实际块与预测块的残差进行编码,帧内编码的像块信息可以用较少的比特数有效表达,减少了空间相关的冗余比特数。

H.264对I帧作帧内预测时将宏块分割为3种,即对静止或慢变的亮度图像用16×16宏块预测模式,对复杂或剧烈运动的亮度图像用4×4块预测模式,对彩色图像用8×8宏块预测模式。每一种预测模式又提供多种方向的空间预测方法,即提供了9种4×4亮度块、4种16×16亮度宏块和4种8×8色度宏块(包含U、V两个相等的色度宏块,每个色度宏块有64个被预测像素)共17种帧内空间预测方法。帧内预测前要对当前块左/上边邻近的解码重构块,根据RDO和SAE进行最佳模式分类和选择,使预测帧内属于不同位移物体的像素数减少,在空域内更加贴近原始帧。这种多尺寸宏块和多方向预测,提高了帧内编码的空域压缩率。对于4×4的亮度块预测方式见图3-3。

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

图 3-3 帧内预测模式(4×4)

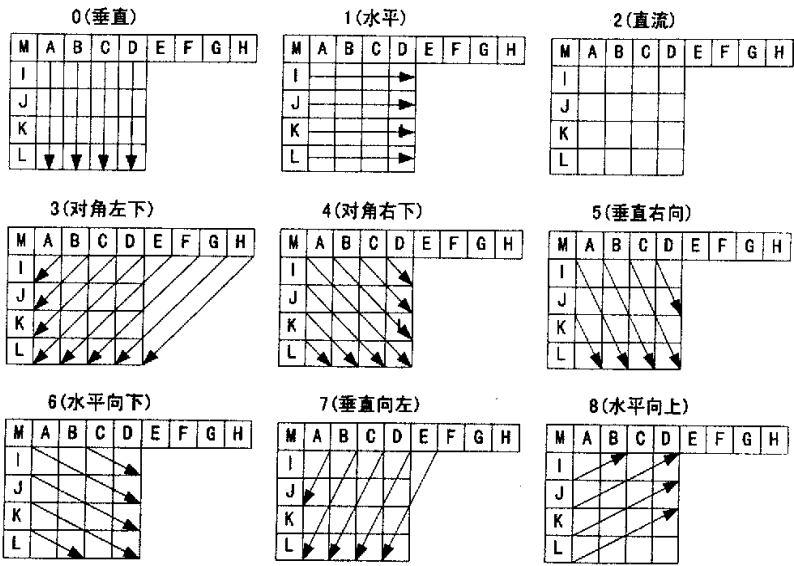


图 3-4 4×4 亮度块预测模式

其中，大写字母表示来自临近块并已经解码重构的像素（当这些像素在图像外部或编码次序上滞后于被预测像素时，是不可得的），小写字母表示将要被预测的像素。共有 9 种预测方式，其中模式 2 预测（不包括在方向预测图中），其余见上图 3-4，它们的预测方向示意图如图 3-5 所示。

模式 0：垂直预测。a，e，i，m 用 A 预测；b，f，j，n 用 B 预测；c，g，k，o 用 C 预测；d，h，l，p 用 D 预测。

模式 2：DC 预测。若 A，B，C，D，I，J，K，L 都是可得的，则所以被预测样本值为 $(A+B+C+D+I+J+K+L+4) \gg 3$ ；若仅 A，B，C，D（或仅 I，J，K，L）可得，则被预测样本值为 $((A+B+C+D+2) \gg 2)$ ；若都不可得，则被预测样本值为 128。

其他几种 4×4 的帧内预测模式和 4 中 16×16 宏块（包括 8×8 的色度宏块）的帧内预测模式可以用类似的方法获得，详细算法见参考文献^[3,3]，这里不再详述。

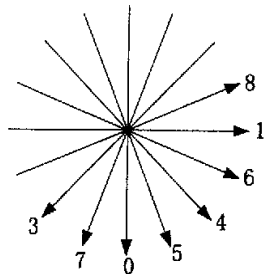


图 3-5 帧内预测方向示意图

3.3.2 多尺寸、高精度、多参考帧帧间预测

过去帧间预测是利用邻帧相关性以 1 种 16×16 或将其 1 个分为 4 个 8×8（非限制运动矢量模式时）两种正方形宏块作为基本单元，来描述所覆盖的全部运动细节，

特别是对于复杂或剧烈的运动区域，时域预测精度误差较大。H.264采用7种树型宏块结构作为帧间预测的基本单元，即将过去16×16模式分割为16×16、16×8、8×16、8×8宏块，8×8模式再分割为8×8、8×4、4×8、4×4子宏块，如图3-6a、6b所示，通过RDO来选择不同的宏块尺寸。在多种不同尺寸和形状宏块的预测模式下，1个宏块可以用1个或最多16个正方形或矩形的运动矢量来描述运动细节，更利于贴近实际实现最佳的块匹配，提高了运动补偿的精度。

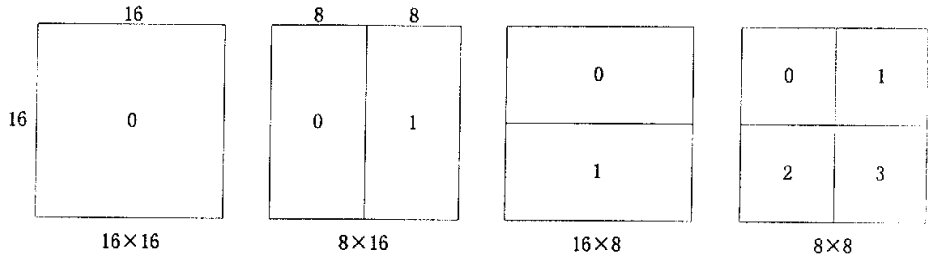


图 3-6a 宏块划分：16×16、16×8、8×16、8×8

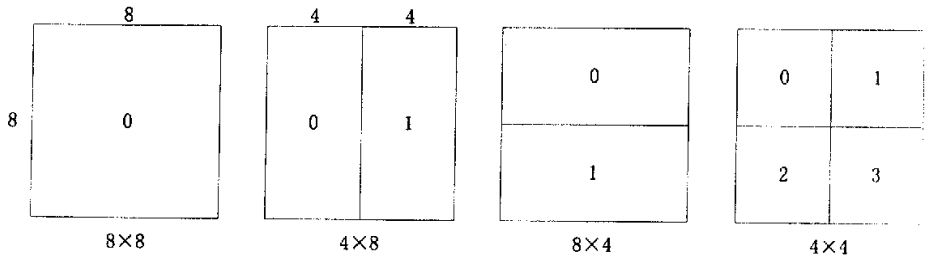


图 3-6b 子宏块划分：8×8、8×4、4×8、4×4

从H.261到MPEG-4运动矢量的估值精度，只从整像素、半像素提高到1/4像素，使搜索区内块匹配误差较大，影响时域压缩率。而H.264的运动估计从支持亮度分量的1/4像素精度提高到支持色度分量的1/8像素精度，并详细定义了相应更小分数像素的插值实现算法，如利用6抽头FIR滤波器产生1/2像素、继续利用1/2像素双向线性内插产生1/4像素、利用4抽头滤波器产生1/8像素（详细算法见参考文献[3.3]）。因此，帧间运动矢量估值精度的提高，使搜索到的最佳匹配点（块或宏块中心）尽可能接近原图，减小了运动估计的残差，提高了运动视频的时域压缩效率。

H.264与以往的单参考帧预测不同，它支持多参考帧预测编码（最多可达16个），即通过在当前帧过去的多个参考帧中进行运动搜索，寻找出当前编码块或宏块的最佳匹配。在出现复杂形状和纹理的物体、快速变化的景物、物体互相遮挡或摄像机快速的场景切换等一些特定情况下，多参考帧的使用会体现更好的时域压缩效果。

3.3.3 整数块变换与可变步长量化

3.3.3.1 整数块变换

帧内、帧间预测的差值要从空域、时域变换到频域进一步压缩处理。传统的DCT变换以8×8像素块为基本单元，变换过程采用复杂的浮点运算，因取舍误差引起像素值漂移，造成反变换失配。较大的像块变换时不仅计算量较大，还会因邻块间相关性降低、强度不连续而出现块效应。H.264主要采用基于DCT技术的新型4×4块整数变换，变换的目的在于在给定的失真条件下，通过整数变换在频

域内给出最大的零系数值，通过有效量化编码达到较高的压缩率。

H.264主要采取3种方式对每一个4×4块进行整数变换，如图3-7所示，即4×4块残差AC变换（含有量化校正矩阵信息），4×4亮度DC系数哈达码（DHT）变换（16×16帧内模式下16个4×4块的DC系数组成的二次4×4块），2×2色度DC系数DHT变换。两个色度块(U,V)的变换在完成相应的亮度块变换后进行（标号-1...25是变换时的次序）。

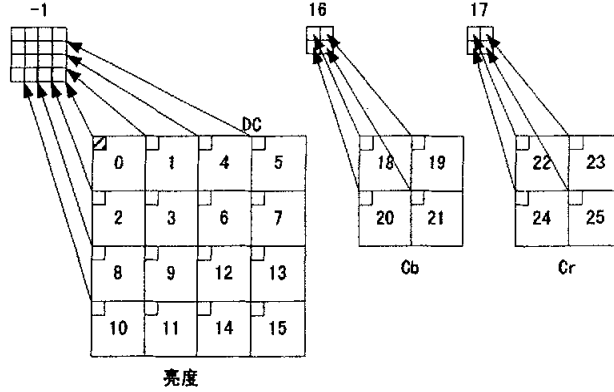


图 3-7 H.264 的 4×4 整数变换

H.264中使用4×4的类DCT整数变换作为残差宏块的基本变换，这种变换的对象是经过运动补偿预测或者帧内预测后的包含残差数据的4×4块。

输入数组 X 的 4×4DCT 变换表示为：

$$Y = AXA^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} X \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \quad (3.1)$$

其中， $a = \frac{1}{2}$ ， $b = \sqrt{\frac{1}{2}} \cos(\frac{\pi}{8})$ ， $c = \sqrt{\frac{1}{2}} \cos(\frac{3\pi}{8})$

为了便于进行移位计算，式 3.1 可以近似写为：

$$Y = (CXC^T) \otimes E = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} X \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/2 & ab/2 & b^2/4 \end{bmatrix} \quad (3.2)$$

其中， $a = \frac{1}{2}$ ， $b = \sqrt{\frac{2}{5}}$

反变换为：

$$X^I = C_i^T (Y \otimes E_i) C_i^T =$$

$$\begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left[\left[X \right] \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix} \quad (3.3)$$

其中 $\pm 1/2$ 可以不失精确的用右移实现。

与 H.263 的 8×8 DCT 相比, H.264 的 4×4 整数变换 DCT 有以下几个优点^[3.5]:

- (1) 变换的核心算法是没有乘法的, 它只进行加法和移位操作, 可以提高运算速度;
- (2) 算法简单明了。 4×4 算法在编码和解码端都有严格相等的反变换, 而且系数都为整数, 避免了截取误差, 使得运动边缘块的编码噪声变小;
- (3) 减少了方块效应。方块效应主要是由于 DCT 变换、量化后, 高频系数的改变造成了图像细节的丢失。由于 H.264 是对 4×4 的块进行变换, 所以较 8×8 的方块效应要小。

当宏块的编码在 16×16 帧内模式下, 此时整个 16×16 块的亮度分量是由相邻像素点预测得到的。宏块的 16 个 4×4 块的直流分量组成一个 4×4 的直流系数矩阵 X_D , 这个矩阵的变换采用的是离散哈达码变换(Discrete Hadamard transform, DHT)。

16×16 帧内模式下的 H.264 的 4×4 亮度直流系数变换, 正向变换公式为:

$$Y_D = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} X_D \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} / 2 \quad (3.4)$$

反变换公式:

$$X_{QD} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} Z_D \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (3.5)$$

式中: Z_D 是 Y_D 量化后的系数矩阵; 而 X_{QD} 经过反量化可以得到 X' 。

因为在帧内编码的宏块里, 大多数的能量都集中在直流系数上, 这种变换有利于进一步压缩 4×4 的亮度直流信号的相关性。

一个宏块中, 每种色度分量都有 4 个 4×4 的块。每个 4×4 的块都作 4×4 的残留变换。变换后所有 4×4 块的直流系数组成一个 2×2 的块 W_D 要在量化前作一次离散哈达码变换。

H.264 的 2×2 色度直流系数变换正向变换公式为:

$$Y_D = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} [W_D] \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (3.6)$$

反向变换公式为:

$$W_{QD} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} [Z_D] \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (3.7)$$

3.3.3.2 可变步长量化

H.264 采用和以往定步长量化不同的可变步长量化, 等比例分级实现, 以进

进一步增强在一定失真条件下的码率控制能力。变换系数矩阵先与量化矩阵进行元素相乘，再对量化起点值修正后进行量化。正向量化公式为

$$Z_{ij} = \text{round}(Y_{ij}/Q_{\text{step}})$$

(3.8)

反向量化为 $Y_{ij} = Z_{ij} \cdot Q_{\text{step}}$

其中 Z_{ij} 为量化后的系数；round为取整； Y_{ij} 为变换后的像素系数； Q_{step} 为量化步长。

每个宏块的量化步长由 52 个量化参数 QP 值决定，色度分量步长比亮度分量略小，以提高彩色的逼真度。52 个量化矩阵含有量化步长信息和量化校正信息，帧内预测时校正值为 $2^{Q/3}$ ，帧间预测时为 $2^{Q/6}$ 。表 3-1 中所示的是亮度信号的量化步长 QP 的参数表(色度信号的量化步长 QP_{chroma} 略有不同)。可以看出：QP 值每加 1， Q_{step} 变化幅度增加 12.5%，QP 每加 6， Q_{step} 加倍。由于量化后的系数要取整，减小了量化精度，有效降低了数据量。

表 3-1 量化步长 QP 的参数表

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	...
QStep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	...
QP	...	18	...	24	...	30	...	36	...	42	...	48	...	51
QStep		5		10		20		40		80		160		224

为了降低变换量化计算的复杂度，H.264 将变换系数幅度的归一化放在反量化过程中处理，整数变换和量化有机融合在一起进行。输入的取样值X先变换为 $W = C_fXC_f$ ，然后按 $Z = W \cdot PF/(Q_{\text{step}} \cdot 2^Q)$ 公式量化。这样，在量化过程中有机结合变换系数矩阵和调整的量化参数，分级实现高精度量化和低精度量化，大大提高了压缩图像的质量和编码效率。

3.3.4 去块效应滤波

为了降低由 H.264 高压缩比产生的明显的块失真效应，所有宏块按扫描顺序进行有条件的滤波。在编码端反变换后，也就是在此宏块重建和存储于预测其他宏块之前，应用去方块滤波器 (Deblocking Filter)；在解码端，在重建和显示此宏块之前，也要应用去方块滤波器。根据宏块中每一个块的位置和量化参数的不同，对每一条块边界设置不同的滤波强度 (Strength)，自适应的调整滤波效果。在宏块中按下面的顺序对 4×4 块的水平和竖直边界滤波。首先对亮度分量的 4 个垂直边界滤波，其次对亮度分量的 4 个水平边界滤波，再次对色度分量的 2 个垂直边界滤波，最后对色度分量的 2 个水平边界滤波。去方块滤波器的滤波将影响临近块边界的至多 3 个像素。通过这种自适应调整强度的去方块滤波，有效的改善解码图象的主观视觉质量。并且在编码器中用滤波的宏块做运动补偿时，可以减小预测残差^[3.6]，提高压缩效率。

3.3.5 增强的熵编码

H.264 在 H.263 多码表变长编码 (VLC) 和 H.26L 单码表通用变长编码 (UVLC) 的基础上，采用统一的两种高性能的熵编码，一种是基于内容的自适应变长编码 (Context-adaptive Variable Length Coding, CAVLC)，另一种是基于内容的自适应二进制算术编码 (Context-based Adaptive Binary Arithmetic Coding, CABAC)。仿真测试表明 CABAC 比 CAVLC 压缩率高 15%^[3.7]。

CAVLC 利用游程码对经预测、变换、量化后的系数块内大量的连零串进行压缩编码，对块内非零系数+1、-1 序列和相邻块间相关的非零系数，采用从

高端向接近直流处逆向“之”字扫描 (Zig-Zag), 如图 3-8 所示, 根据数据的统计特性进行自适应选择量级参数查表编码, 充分利用信源相关性和人眼视觉特性, 有效压缩代码数据冗余, 降低码率。

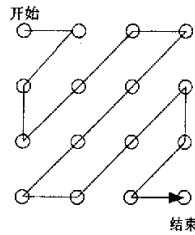


图 3-8 帧模式下 4×4 亮度块 Zig-Zag 扫描

当 `entropy_code_mode` 被设为 0 的时候, 剩余块中的数据使用 CAVLC 进行编码, 而其他变长编码的单元使用 Exp-Golomb 编码。CAVLC 用于对 Zig-Zag 扫描之后的 4×4 或者 2×2 块中变换系数的编码。它利用了 4×4 块的一些特点来提高编码效率:

- (1) 在预测、变换和量化之后, 块中的数据是十分稀疏的, 存在大量的零。CAVLC 使用游程编码高效的对这些零串进行编码;
- (2) Zig-Zag 扫描之后的最高频系数常常是 ± 1 , CAVLC 将这些高频的 ± 1 系数表示为 Trailing 1, 或者 T1;
- (3) 相邻块中非零系数的个数是相关的。系数的个数用查表来编码, 表的选择取决于相邻块中非零系数的个数;
- (4) 非零系数的大小在扫描后的开始 (也就是靠近直流系数附近的位置) 是比较大的, 而在高频位置较小。CAVLC 利用这一特点, 根据最近编码的大小值自适应的选择 VLC 表。

当 `entropy_code_mode` 被设为 1 的时候, H.264 使用一种基于内容的自适应算术编码对其语法元素进行编码和解码。这种算术编码根据语法元素的内容为其选择概率模型, 根据本地的统计数字自适应的进行概率预测, 并使用算术编码。

CABAC 分 3 步进行: ①二值化。将所有经预测、变换和量化的待编码符号, 根据不同的语法元素所要求的二进制映射规则 (如常用的一元二进制), 映射成二进制符号串 (Bins), 送到后续的二进制算术编码器。②选择内容模型。根据上下文宏块类型、运动矢量、参考帧数、预测模式等不同的语法元素预先定义概率模型; 据当前待编码符号的类型及左/上邻块符号给当前待编码符号选定相应已定义概率的内容模型; 对当前待编码符号进行更加准确的概率估计。H.264 定义了 64 种概率模型和 267 (编号 0~266) 种内容模型, 以便为编码符号提供准确的条件概率估计, 压缩符号间冗余, 从其中一种概率状态转化到最为可能符号的状态即完成概率估计。③自适应算术编码并自动更新概率。根据已完成待编码符号的概率估计, 给每个符号分配 1 个分数比特, 随着编码符号的增加, 对应小数概率区间不断缩小, 符号可以逼近它的熵编码极限。CABAC 可以在每编码 1 个符号后, 根据该符号位的上下文变化, 不断动态调整内容模型, 自动完成新的概率估计。

CAVLC 与 CABAC 相比, 具有运算简单、易于实现的特点, 但 CABAC 比 CAVLC 可以提高 15% 的编码效率, 主要是因为^[3.7]:

- (1) 内容模型提供编码符号概率分布的估计。利用适当的内容模型, 在

编码当前符号时, 根据已编码的临近符号的概率统计, 在不同的概率模型间转换, 充分利用符号间的冗余。

- (2) 算术编码可以给每一个符号的字母分配非整数的比特, 因此符号可以接近它的熵率被编码。如果选择了高效的概率模型, 符号概率常常大于 0.5, 这时分数比特就比 CAVLC 的整数 (至少 1 比特) 比特高效得多。
- (3) 自适应的算术编码可以使编码器自适应采用动态的符号概率统计。例如, 运动矢量的概率统计随空间、时间的不同, 或序列、比特率的不同可以发生巨大的变化。因此自适应模型可以充分利用已编码符号的概率累计, 使算术编码更好的适应当前符号的概率, 提高了编码效率。

3.3.6 SP-slice 和 SI-slice

在 H.264 中, picture 是场和帧的统称。slice 是 picture 的分割, slice 由宏块或宏块对 (Macroblock pairs) 组成。slice 中的宏块或宏块对在 picture 中的序号不一定是连续的。slice 是一个独立的编码单元, 它不需要参考同一个 picture 中的其他 slice 进行编码。根据图像中多个运动物体的不同运动状态将图像分割成不同或相同性质的 slice, 条件允许可以对多个 slice 并行处理, 能够增加图像精度, 减少计算量, 提高运算速度。

类似于 H.263 中的 I 帧、P 帧、B 帧, H.264 中有 I-slice、P-slice、B-slice。另外, 为了满足不同视频码流间快速有效的切换和视频解码器快速有效的随机访问 (例如: 一幅视频图像在 internet 网传播时以不同的比特率编码, 网路流量下降时解码器需要从高比特率视频码流自动切换到低比特率视频码流), H.264 标准建立了 SP-slice 和 SI-slice 的概念和相应的语法。

如图 3-9 所示, 解码器将从 StreamA 切换到 StreamB, 假设 StreamB 中都是编码后的 P-slice。通常情况下若切换到 B3, 需要将 (B2 编码成 I-slice, 这样就增加了码率。H.264 定义了 SP-slice (用于视频流转换的 P-slice)。这时解码器的输入是 SP-slice AB2, 解码后的 A1 作为运动补偿参考帧, 解码器的输出是 B2 帧。再以 B2 作为参考帧去重构 B3。SP-slice 的编解码方法参见参考文献^[3,8]。

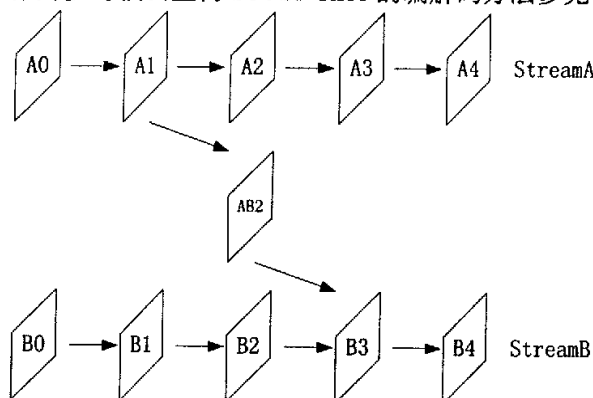


图 3-9 使用 SP-slice 进行切换

SP-slice 的使用, 在不明显增加码率的前提下, 为不同图像的视频流之间、同一图像序列的不同比特率的码流之间、或同一视频码流的随机访问提供了快速高效的切换方式, 在满足人的视觉感受的前提下, 避免使用 I-slice, 节约了时间和资源。

SI-slice (用于视频流转换的I-slice) 的使用方法与SP-slice基本相同, SI-slice 主要运用在两个完全不相关、无法使用运动补偿预测的序列的切换。

3.4 H.264 视频编解码器的性能与复杂度分析

H.264 视频编码标准与现有的视频编码标准相比有了明显的改进。本节中, 在同等视觉质量前提下, 对 H.264 的编码效率与其他标准在不同的应用环境中进行了比较, 并对其采用的新的关键技术进行了复杂度分析, 找出编解码器运行的瓶颈和热点, 并据此确定了下一步编解码器优化的重点。

3.4.1 H.264 标准与现有其他编码标准的编码效率的比较

文献[3.9]中对现存的几种视频编码标准在流媒体应用、视频会议、娱乐应用方面的不同编码效率进行了详细全面的比较。这里所有的编码器均采用 RDO 优化控制, 对在流媒体应用与视频会议应用方面的编码效率进行比较时, 采用的是 CIF (the Common Intermediate Format, 352×288 picture elements) 与 QCIF (the Quarter Common Intermediate Format, 176×144 picture elements, progressive) 格式的测试序列; 而对于娱乐应用进行编码效率比较时采用的是 ITU-R CCIR601 (720×576 picture elements, interlaced) and HDTV (High Definition Television, 1280×720 picture elements, progressive) 格式的测试序列。

在视频流媒体应用方面, 比较的标准是 H.264/AVC MP (Main Profile), MPEG-4 Visual ASP (Advanced Simple Profile), H.263 HLP (High Latency Profile) 与 MPEG-2 Video ML@MP (Main Level at Main Profile)。对于 Tempete 测试序列 (15 Hz), 在相同码率条件下的亮度分量的 PSNR 如图 3-10a 所示。表 3-2a 中描述的是在同等视觉质量前提下, H.264 标准能够节省的平均码率。显然, H.264 MP 相对于 MPEG-2 ML@MP 与 MPEG-4 Visual ASP 的平均码率降低分别约为 63%、37%。

对于视频会议应用, 比较的标准是 H.264/AVC BP (Baseline Profile), MPEG-4 Visual SP (Simple Profile), H.263 Baseline, and H.263 CHC (Conversational High Compression)。对于 Paris 测试序列 (15Hz), 图 3-10b 是测试结果, 表 3-2b 是在同等视觉质量前提下, H.264 标准能够节省的平均码率。从表中可以看出, H.264 BP 相对于 H.263 Baseline 与 H.263 CHC 的平均码率降低分别约为 40%、27%。

而对于视频图像要求更高的娱乐应用方面的比较显示, H.264 标准相对于 MPEG-2 Video ML@MP 与 HL@MP 标准可以节省 45% 的平均码率^[3.9]。

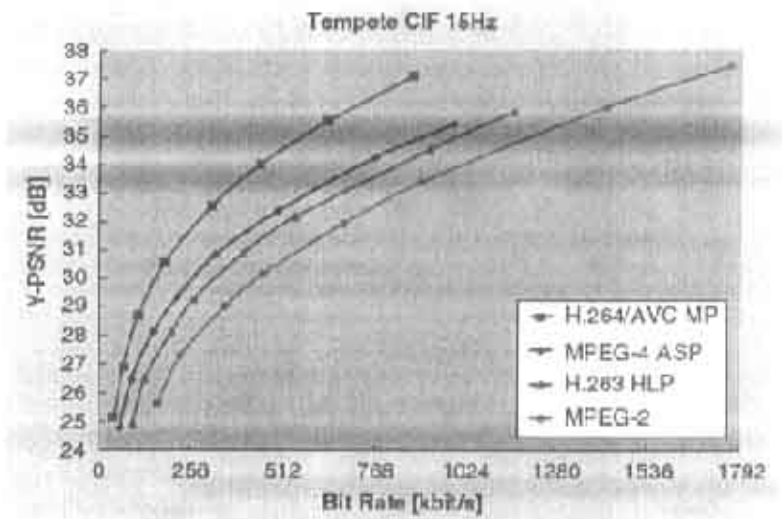


图 3-10a H.264 标准在视频流媒体应用情况下相对其他标准的平均码率的比较^[3.9]

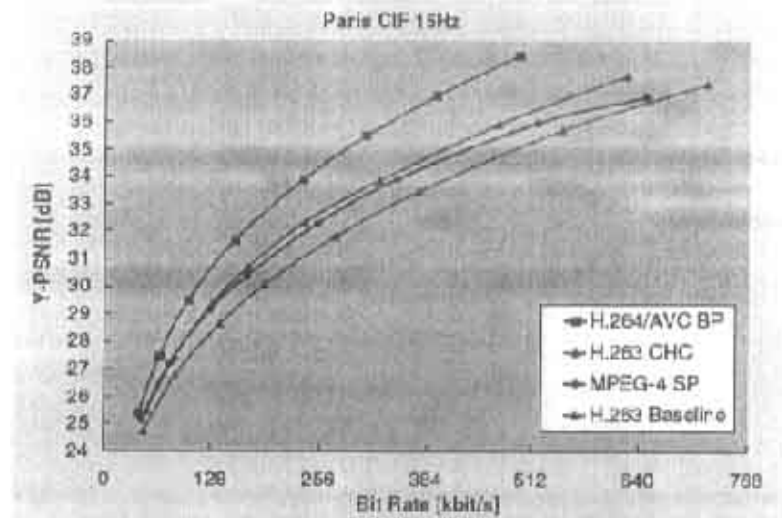


图 3-10b H.264 标准在视频会议应用情况下相对其他标准的平均码率的比较^[3.9]

表 3-2a H.264 MP 标准相对其他标准的编码比特率的平均节省量^[3.9]

	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC MP	37.44%	47.58%	63.57%
MPEG-4 ASP	—	16.65%	42.95%
H.263 HLP	—	—	30.61%

表 3-2b H.264 BP 标准相对其他标准的编码比特率的平均节省量^[3.9]

	H.263 CHC	MPEG-4 SP	H.263 Base
H.264/AVC MP	27.69%	29.37%	40.59%
H.263 CHC	—	2.04%	17.63%
MPEG-4 SP	—	—	15.69%

3.4.2 H.264 标准的复杂度分析

评估一种新的视频编码标准的复杂度不是一件容易的事情，因为它与软件的执行平台有直接的关系，如 DSP 处理器、FPGA 与 ASIC 等。本节中将要讨论的对象是与硬件平台无关的软件复杂度。

复杂度分析的缺省软硬件环境（论文后续部分中同样使用这个配置）：实验的硬件环境采用 P4-2.6G/DDR 256M 内存，软件测试模式为 JVT 发布的联合模型 JM9.4^[3 10]，模块耗时统计和编码器性能分析使用 Intel Vtune Performance Analyzer 7.0。

进行复杂度分析的测试序列：Tempete 30Hz CIF、Foreman 30 Hz QCIF&CIF 与 Mobile 30 Hz QCIF&CIF（码率范围均为 40 Kbits/s~2 Mbits/s），以及 Container 15Hz QCIF。

JVT JM9.4的缺省编码参数：采用1个参考帧、支持RDO优化、CABAC熵编码、搜索范围32。分别对小尺寸多模式帧内预测、多尺寸高精度多帧参考帧间预测、整数变换与可变步长量化、去块效应滤波、以及增强的CABAC熵编码等模块的性能进行测试。

3.4.2.1 帧内预测

H.264 采用了帧内预测编码，充分利用被预测块左边和上边已解码重构的邻近宏块像素对当前块进行预测，只将实际块与预测块的残差进行编码，帧内编码的像块信息可以用较少的比特数有效表达，减少了空间相关的冗余比特数。对Tempete CIF 30Hz序列测试帧内预测技术采用与否的RD曲线如图3-11所示，由图可以看出，帧内预测在有效降低码率的同时提高了图像编码质量。

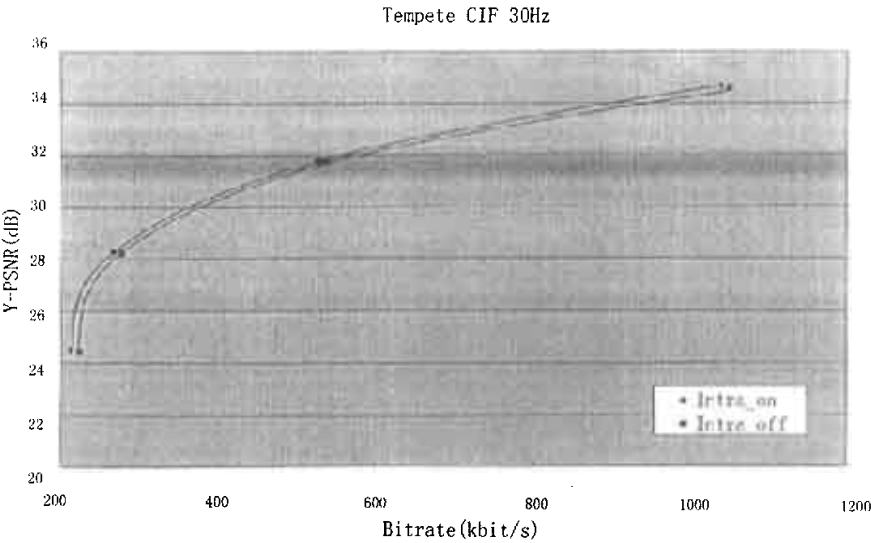


图3-11 帧内预测R-D曲线

3.4.2.2 可变块模式

采用多种运动矢量块模式进行运动估计，与只用 16×16 模块相比较，码率可节约超过 10%。对 mobile.qcif 原始视频序列，只改变运动矢量模块的大小，其它参数相同，图像信噪比和码率的实验结果见表 3-3。

由表 3-3 可以看出，采用多种运动矢量块模式的运动估计，在保证图像相

同质量的前提下，可以得到较低的码率。

H.264 选取最优运动矢量块模式，是需要在运动矢量值编码和系数编码中取平衡点，所以，为了达到码率最低的效果，采取全部运动矢量模块的图像信噪比并不一定是最好的。

表 3-3 mobile.qcif 不同运动模式块编码实验结果

运动模块大小	16×16	16×8	8×16	8×8	8×4	4×8	4×4	全部
PSNR_Y	32.33	32.31	32.33	32.36	32.51	32.48	32.53	32.62
PSNR_U	35.11	35.10	35.10	35.13	35.12	35.15	35.16	35.18
PSNR_V	34.53	34.54	34.54	34.58	34.60	34.58	34.57	34.60
Bitrate(kb/s)	359.23	357.70	355.26	353.98	345.03	345.14	346.15	324.57

3.4.2.3 像素插值

H.264的运动估计支持亮度分量的1/4像素精度与色度分量的1/8像素精度的插值。帧间运动矢量估值精度的提高，使搜索到的最佳匹配点（块或宏块中心）尽可能接近原图，减小了运动估计的残差，提高了运动视频的时域压缩效率。如图3-12所示，采用1/4或者1/8像素的运动估值精度，和整数精度相比，码率节约了10%，视频图像编码质量要求越高，可以节省的码率越大。

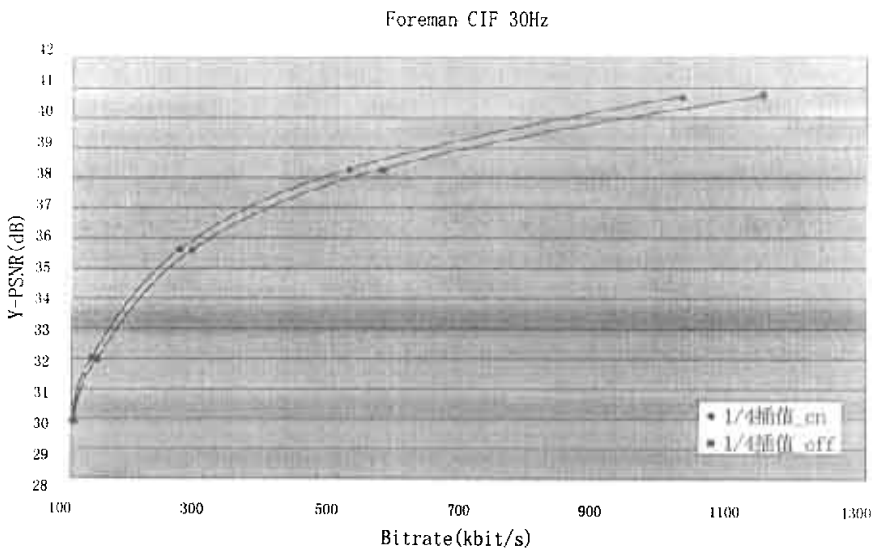


图3-12 1/4像素插值R-D曲线

3.4.2.4 多帧参考

H.264 与以往的单参考帧预测不同，它支持多参考帧预测编码（最多可达16个），即通过在当前帧过去的多个参考帧中进行运动搜索，寻找出当前编码块或宏块的最佳匹配。在出现复杂形状和纹理的物体、快速变化的景物、物体互相遮挡或摄像机快速的场景切换等一些特定情况下，多参考帧的使用会体现更好的时域压缩效果。

但是，采用多参考帧帧间编码方式，使用16帧参考帧比只使用1帧参考帧节约码率5~10%。测试中，对Tempete.cif原始视频序列，做IBPBP编码，采用多参考帧，其它参数相同，图像信噪比和码率的实验结果见表3-4。

表 3-4 Tempete.cif 多帧参考帧实验结果

参考帧数	1	2	3	4	5	16
PSNR_Y	34.03	34.30	34.38	34.42	34.44	34.48
PSNR_U	36.95	37.05	37.09	37.13	37.12	37.16
PSNR_V	39.08	39.25	39.34	39.34	39.35	39.38
Bitrate(kb/s)	874.06	756.08	735.69	728.64	726.60	722.23

由实验结果可知，多参考帧有利于提高运动估计的准确性，在一定程度上可以提高图像质量和降低码率。但是，多参考帧增加了编码的时间和存储容量。多帧参考在高码率时效果较好，但此时编码器的计算和内存开销将成倍增加。

3.4.2.5 整数变换与可变化步长

采用 4×4 整数变换代替原来的 DCT 变换，更适合于多种运动矢量块运动估计的编码，也更能很好的适应 DSP（定点运算）的实现。从图 3-13 可以看出在量化和反量化过程中，整数变换比 DCT 变换计算精度高，所得到的重建恢复图像质量好。

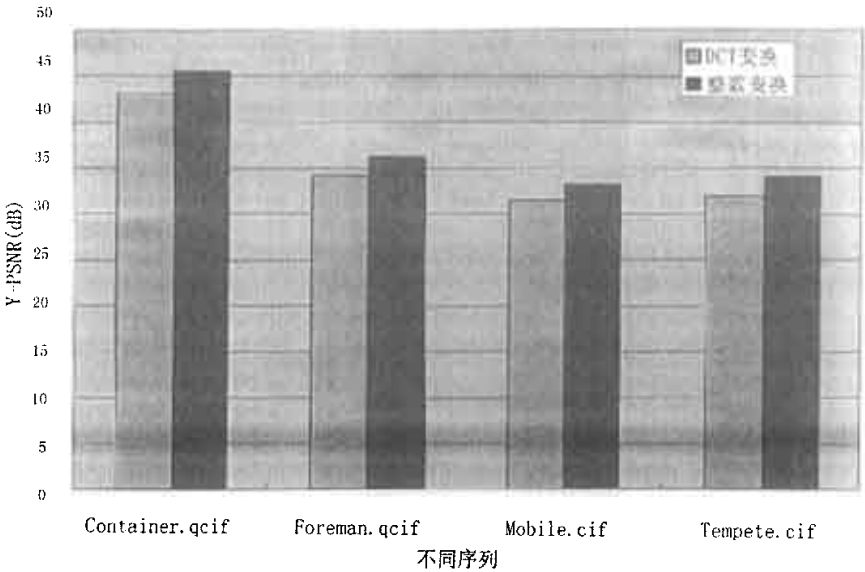


图 3-13 DCT 变换和整数变换重建图像信噪比的对比

3.4.2.6 去块效应滤波器

为了降低由 H.264 高压缩比产生的明显的块失真效应，标准中采用了去块效应滤波技术。去块效应滤波器不但能够有效的改善主观视频质量，而且可以在低码率条件下提高图像的信噪比。对低码率序列 Container QCIF 15Hz 的测试结果如图 3-14 所示。

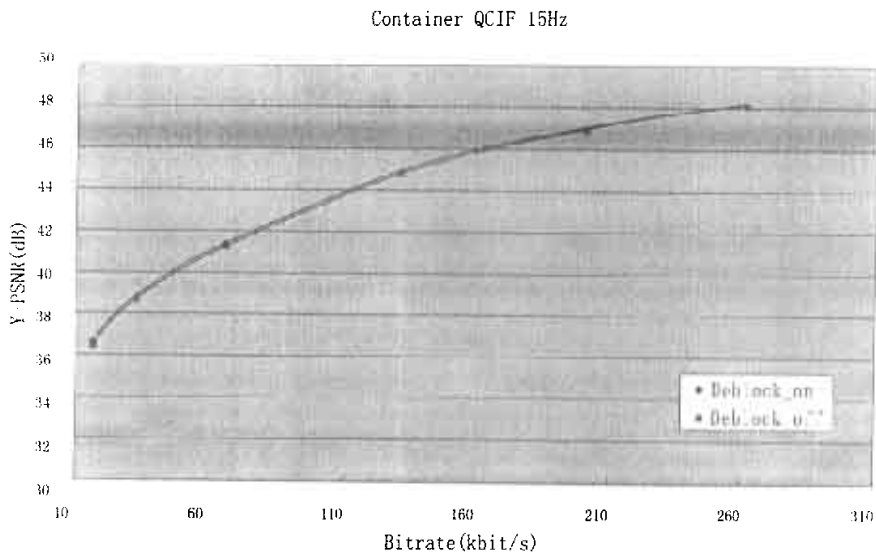


图3-14 去块效应滤波器R-D曲线

3.4.2.7 熵编码

H.264支持两种熵编码方案,即基于上下文的变长编码(CAVLC)和基于上下文的算术编码(CABAC)。前者更为简单,易于实现;后者则能进一步提高熵编码的性能。相对于CAVLC,同等图像质量下,Foreman CIF序列采用CABAC编码可以节省10%的码率,测试结果如图3-15所示。

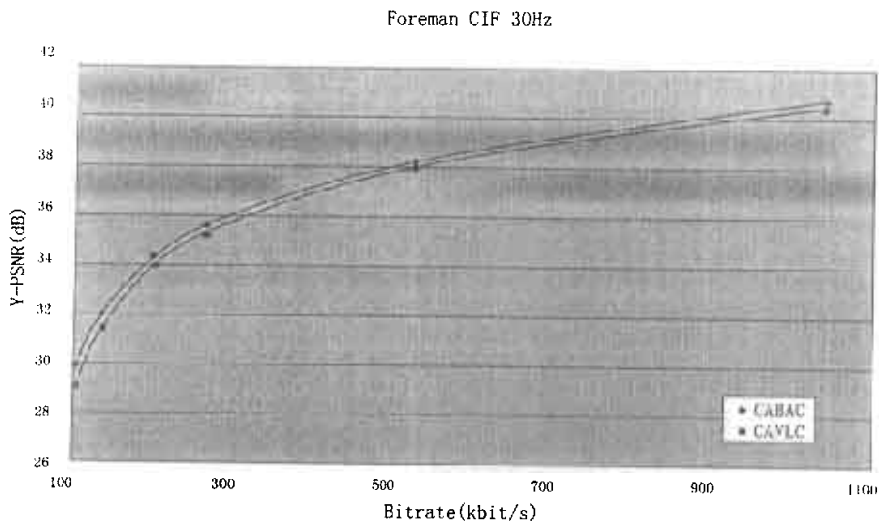


图3-15 CABAC与CAVLC的R-D曲线

3.4.3 H.264 视频编解码器中主要模块的耗时统计

分析复杂度是有针对性的进行高效编码器优化的一个重要步骤,而 H.264 极高的计算开销也一直受到国内外学者的关注。本节中对 JM9.4 编码器的主要模块耗时进行了测试,仿真测试的环境和缺省的编码参数与性能分析中的一致,

对于不同的测试序列，测试结果的统计数据见表 3-5。其中率失真开销指的是基于率失真优化 RDO 的编码模式选择的计算开销。

表 3-5 H.264 视频编码器主要模块的耗时统计(单位: %)

序列	格式	运动估计	率失真	变换和量化	其他
Container	QCIF	31.23	33.68	12.39	22.70
Foreman	QCIF	34.79	34.26	11.35	19.60
Mobile	CIF	60.17	21.19	6.12	12.52
Tempete	CIF	64.72	17.68	5.25	12.35

从表 3-5 可以看出，H.264 参考编码器各模块的计算复杂度随序列的不同有明显的变化，但总体上运动估计、基于 RDO 的模式选择以及变换和量化的开销最大，并且随图像的尺寸增大，运动估计所占比例越高。因此本文的后续部分，将编码器优化的重点放在运动估计、帧内帧间模式的快速选择以及对变换量化模块的改写。

对 JM9.4 解码器的主要模块耗时的测试，仿真测试的环境性能分析中的一致，测试已编码文件为 Foreman.264，主要耗时模块的统计如图 3-16 所示。测试结果显示运动补偿中的像素插值、内存分配与 I/O 操作是解码器中主要的耗时模式，是影响 JM9.4 解码速度的瓶颈，需要进行优化改写以提高解码效率。

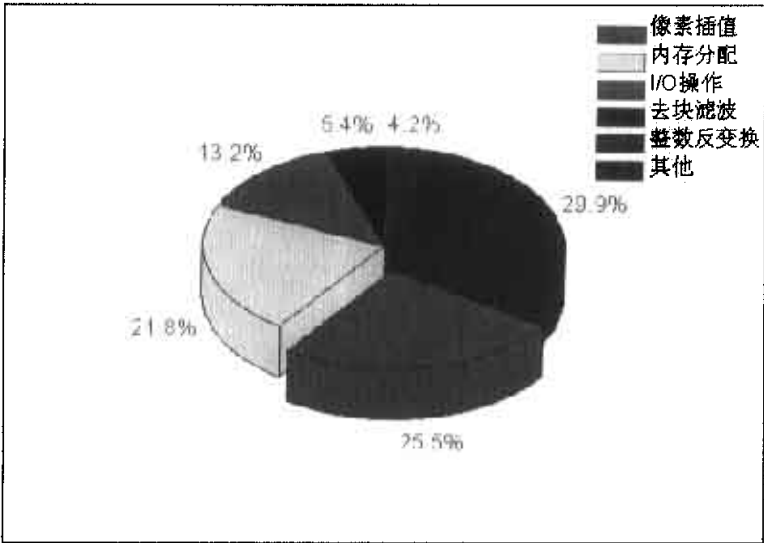


图 3-16 Foreman.264 解码过程中主要模块的耗时统计

3.5 本章小结

H.264 作为目前所有视频压缩标准中压缩效率最高、网络适应能力最好的一个可分级的、面向未来的开放标准，3 个不同的复杂层次分别面向不同的应用范围。其基本系统无需使用版权，在 DTV/HDTV 和 IP 视频、移动通信等领域将会有更多、更深入的视频应用和服务。

本章中，首先给出了 H.264 视频编解码器的结构框图，对其中的主要功能模块，如小尺寸多模式帧内预测、多尺寸高精度多帧参考帧间预测、整数块变

换与可变步长量化、去块效应滤波、增强的熵编码以及 SP-slice 和 SI-slice 等模块及其包含的新技术进行了详细论述。接着对 H.264 视频编解码器的性能与复杂性进行了分析。仿真结果表明, 相同的 PSNR 条件下 H.264 的码率比 MPEG-4, H.263 和 MPEG-2 平均码率降低分别约为 37%, 47%和 60%。但编码效率提高的同时却增加了巨大的计算开销, H.264 编码复杂度约是 MPEG-2 的 10 倍以上, 解码复杂度约是 MPEG-2 的 3 倍以上。因此, 解决 H.264 的实用问题, 关键在于如何在不降低 H.264 编/解码器功能的前提下, 优化 JVT 算法, 降低编解码复杂度。

参考文献

- [3.1] Joint Video Team of ITU-t and ISO/IEC JTC 1, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC). JVT-G050, 2003.
- [3.2] Wiegand T., Sullivan G.J., Bjntegaard G., Luthra A., Overview of the H.264/AVC video coding standard. Circuits and Systems for Video Technology, IEEE Transactions on Volume 13, Issue 7, July 2003 Page(s):560-576.
- [3.3] Iain E.G.Richardson, The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression-Video Coding for Next-generation Multimedia[M/OL]. <http://nmlab.zju.edu.cn/~mm/book/H.264.And.MPEG-4.Video.Compression.Video.Coding.For.Next.Generation.Multimedia.pdf>.
- [3.4] Ostermann J., Bormans J., List P., Marpe D., Narroschke M., Pereira F., Stockhammer T., Wedi T., Video coding with H.264/AVC: tools, performance, and complexity. Circuits and Systems Magazine, IEEE Volume 4, Issue 1, First Quarter 2004 Page(s):7-28.
- [3.5] 曾昭平, 马仲华, 张文军, H.26L 中变换编码和量化的原理与分析, 通信学报, 2002, 23(2): 102-106.
- [3.6] Hong M.-C. Hahn H.-S, A Loop/Post Filter to Suppress Blocking and Ringing Artifacts for H.26L VideoCodec. SPIE Proceeding of Visual Communication and Image Processing, 2001(1):940-947.
- [3.7] Marpe D., Schwarz H., Blattermann G., et al, Context-Based Adaptive Binary Arithmetic Coding in JVT/H.26L. Proc. IEEE International Conference on Image Processing (ICIP-02), 2002 Vol.2. Page(s):513-516.
- [3.8] Karczewicz, M., Kurceren, R., The SP- and SI-frames design for H.264/AVC Circuits and Systems for Video Technology, IEEE Transactions on Volume 13, Issue 7, July 2003 Page(s):637-644.
- [3.9] T. Wiegand, H. Schwarz, A. Joch, and F. Kossentini, Rate-constrained coder control and comparison of video coding standards, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, pp. 688-703, July 2003.
- [3.10] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. Sullivan, Performance comparison of video coding standards using Lagrangian coder control, Proc. of the IEEE ICIP 2002, part II, pp. 501-504, Sept. 2002.

第4章 H.264的帧内/帧间编码模式快速选择算法

H.264 定义了 9 种 4×4 块的帧内预测模式($M_4=9$)、4 种 16×16 宏块的帧内预测模式($M_{16}=4$)、4 种色度宏块的帧内预测模式($M_8=4$)，以及 7 种不同大小的帧间宏块模式，为了提高编码效率，编码时对每一种模式都进行运动搜索，并利用利用率失真优化(RDO)分别计算编码代价，选择编码代价最小的模式都作为帧内或帧间编码模式。巨大的计算量使模式选择成为帧内/帧间编码的瓶颈所在。本章中，针对帧内/帧间编码模式选择，分别提出一种选择快速算法。实验结果表明，采用本文中的快速模式选择算法可以有效的减少预测的候选模式数，在作 IPBPB 编码时可以节约大约 50%-70%的编码时间，同时图像质量和码率基本不变。

4.1 帧内模式快速选择算法

4.1.1 帧内预测的模式选择

H.264 为了实现高质量高压压缩比，采用了 RDO 技术。在对 I 帧编码时，需要通过遍历所有可能的帧内编码模式组合，寻找率失真开销最小的那一种作为最优编码模式。而对于 P, B 帧编码，则需遍历所有的帧内和帧间编码模式。以编码 I 帧为例，其一个宏块组合(指一个 16×16 的亮度宏块和相应的两个 8×8 的色度宏块)的 RDO 模式选择过程^[4,11]如下：

- 1) 对于亮度宏块中某一 4×4 块，按某种 4×4 帧内预测模式 $\text{mode}_{4 \times 4}$ 建立相应的帧内预测块；
- 2) 计算预测的 4×4 块和原始的 4×4 块之间的绝对差($\text{SAD}_{4 \times 4}$)；
- 3) 计算该模式的率失真开销 $\text{RDcost}_{4 \times 4}$ ；
- 4) 在该 4×4 块中重复以上 1 到 3 步，遍历所有的 9 种 4×4 帧内预测模式；
- 5) 选取具有最小率失真开销 $\text{RDcost}_{4 \times 4}$ 的模式作为该预测块的最佳 4×4 帧内预测模式；
- 6) 对宏块中 16 个 4×4 块重复以上 1 到 5 步，获得每一个 4×4 块的最佳编码模式和相应的 $\text{RDcost}_{4 \times 4}$ ，进而获得整个宏块的率失真开销 $\text{RDcost}_{16 \times 16}$ ；
- 7) 按类似的方法遍历 4 种 16×16 宏块帧内预测模式，并计算相应的宏块率失真开销 $\text{RDcost}_{16 \times 16}$ ，选择宏块率失真开销 $\text{RDcost}_{16 \times 16}$ 最小的模式为最佳 16×16 宏块的帧内预测模式；
- 8) 根据 6 和 7 中最小的宏块率失真开销 $\text{RDcost}_{16 \times 16}$ ，判断亮度宏块的 4×4 或 16×16 帧内预测模式；
- 9) 对每一种 8×8 色度宏块的帧内预测模式(两个色度宏块使用相同的模式)计算相应的率失真开销 $\text{RDcost}_{8 \times 8}$ ，并重复以上 1 到 8 步，获得相应的宏块组合率失真开销 $\text{RDcost}_{\text{MB}}$ ，选择最小的宏块组合率失真开销 $\text{RDcost}_{\text{MB}}$ 作为该宏块组合的最佳帧内预测模式。

由此，完成了一个宏块组合的帧内预测模式选择。显然对一个宏块组合共有 $M_8 \times (M_4 \times 16 + M_{16} \times 4)$ 种不同的模式组合，因此共需完成 640 次模式计算，RDO 才能获得最佳的帧内预测模式选择。巨大的计算量使模式选择成为帧内编码的瓶颈所在。

4.1.2 基于局部边缘方向信息的帧内预测快速算法

Pan 等^[4,2]提出了一种基于边缘方向直方图的快速帧内预测模式选择算法,这种算法可以有效减少帧内预测的模式数目,但是边缘方向直方图的计算仍然十分复杂,没有能够有效降低计算复杂度。

文献[4.3]中提出了一种基于边缘特征的4×4帧内快速算法,在保持图像编码质量与码率有较少提升的前提下有效降低了计算复杂度。但是其算法仅仅对4×4块的预测实现了快速算法,对于16×16块与色度分块的预测没有考虑。本节中,结合文献[4.3]中提到的基于当前块局部边缘特征的算法思想,在保持图像质量和码率不变的情况下,对帧内预测模式进行快速选择。

4.1.2.1 4×4块内预测的快速算法

首先对当前4×4块进行局部边缘方向信息提取。这里将一个4×4块分割成4个2×2块,如图4-1所示。

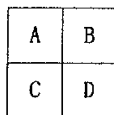


图4-1 4×4块分割示意图

图中A、B、C、D分别代表相应2×2块所有像素值之和, $P(x,y)$ 为相应位置的像素值, A、B、C、D与 $P(x,y)$ 的关系如式(4.1)所示:

$$A = \sum_{y=0}^1 \sum_{x=0}^1 P(x,y) \quad B = \sum_{y=0}^1 \sum_{x=2}^3 P(x,y) \quad C = \sum_{y=2}^3 \sum_{x=0}^1 P(x,y) \quad D = \sum_{y=2}^3 \sum_{x=2}^3 P(x,y) \quad (4.1)$$

为了有效表征局部边缘方向信息,文献[4.3]引入两个边缘特征参数:垂直边缘参数 F_v 与水平边缘参数 F_h 以表征局部边缘方向信息。

$$F_v = \left\lfloor \frac{(A+C) - (B+D)}{S} \right\rfloor \quad (4.2a)$$

$$F_h = \left\lfloor \frac{(A+B) - (C+D)}{S} \right\rfloor \quad (4.2b)$$

其中 $\lfloor \cdot \rfloor$ 表示取整函数(the floor function), S 为量化因子(a scaling factor)。

F_v 与 F_h 分别代表的是4×4块左右两部分与上下两部分之间的相对灰度差值,如图4-2所示。

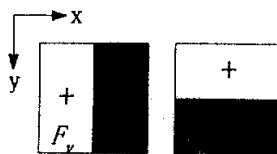







图4-2 F_v 与 F_h 的示意图

根据这两个参数,可以获得当前4×4块的边缘方向信息,如表4-1所示:

表 4-1 F_v 与 F_h 的关系与局部边缘方向对应表

Case	F_v 与 F_h 的关系	边缘方向
1	$ F_v = F_h =0$	不存在显著边缘 
2	$ F_h =0$ 且 $ F_v >0$	垂直边缘 
3	$ F_v =0$ 且 $ F_h >0$	水平边缘 
4	$ F_v = F_h >0$ 且 $F_v/F_h>0$	左下对角边缘 
5	$ F_v = F_h >0$ 且 $F_v/F_h<0$	右下对角边缘 
6	$ F_v > F_h >0$	存在显著的垂直边缘
7	$ F_h > F_v >0$	存在显著的水平边缘

由表 4-1 中的对应关系, 可以根据 F_v 与 F_h , 为当前块选定不同的候选模式块 (CBPMs: Candidate Block of Prediction Modes), 又由于 DC 模式相对与其他 8 种模式最有可能成为最优模式, 所以这里对每一种情况都将 DC 模式作为 CBPM, 如表 4-2 所示。

表 4-2 F_v 与 F_h 由决定的 CBPMs

Case	CBPMs
1	2
2	0,2
3	1,2
4	3,2
5	4,2
6	0,5,7,3,4,2
7	1,6,8,3,4,2

4×4 块快速帧内预测步骤如下:

1) 计算当前块 F_v 与 F_h ;

量化因子 S 由量化步长 Q_p 决定, 如式 4.3 所示:

$$S = \begin{cases} 8 & Q_p < 20 \\ 16 & 20 \leq Q_p < 30 \\ 32 & 30 \leq Q_p < 40 \\ 64 & Q_p \geq 40 \end{cases} \quad (4.3)$$

- 2) 根据 F_v 和 F_h 为当前块选定 CBPMs;
- 3) 为候选模式生成 4×4 预测块;
- 4) 计算候选模式代价;

5) 充分 3、4 步骤, 计算所有候选模式的代价并选择最合适的 4×4 模式。

4.1.2.3 16×16 亮度块与 8×8 色度块的帧内预测的快速算法

基于同样的观察, 对 16×16 亮度块以及两个 8×8 色度块计算局部边缘方向信息以选择合适的 CBPMs, 并同样将 DC 模式作为常用的 CBPMs, 以减少 FS 模式下 CBPMs 的数目, 有效降低计算复杂度。

4.1.3 实验结果与分析

对基于局部边缘方向信息的快速帧内预测编码模式选择快速算法的实验分析使用本文中的缺省软硬件环境, 缺省编码参数为: 采用 1 个参考帧、支持 RDO 优化、CABAC 熵编码、运动估计搜索算法采用全搜索 FS, 测试序列为具有不同运动复杂度信息的 Tempete 30Hz CIF、Mobile 30Hz CIF、Foreman 30Hz QCIF、Container QCIF 15 Hz

实验分析结果如表 4-3 所示, 利用原始图像的局部边缘方向信息可以有效的减少帧内预测的候选模式数, 在作 IBPBP 编码时可以节约大约 40%-65% 的编码时间, 同时, PSNR 和码率与原 JM 软件遍历的模式选择算法相比, 基本不变。图 4-3 表明, 在不同码率情况下, 基于局部边缘方向信息的快速帧内预测编码模式选择的快速算法在降低候选块模式 CBPMs 的情况下, PSNR 和码率基本不变。

表 4-3 帧内快速选择算法性能比较

序列	Bitrates(%)	Y-PSNR(%)	Times(%)
Container QCIF	+1.57	-0.02	-64.2
Foreman QCIF	+2.12	-0.01	-57.6
Mobile CIF	+1.64	+0.01	-45.3
Tempete CIF	+1.71	-0.01	-41.3

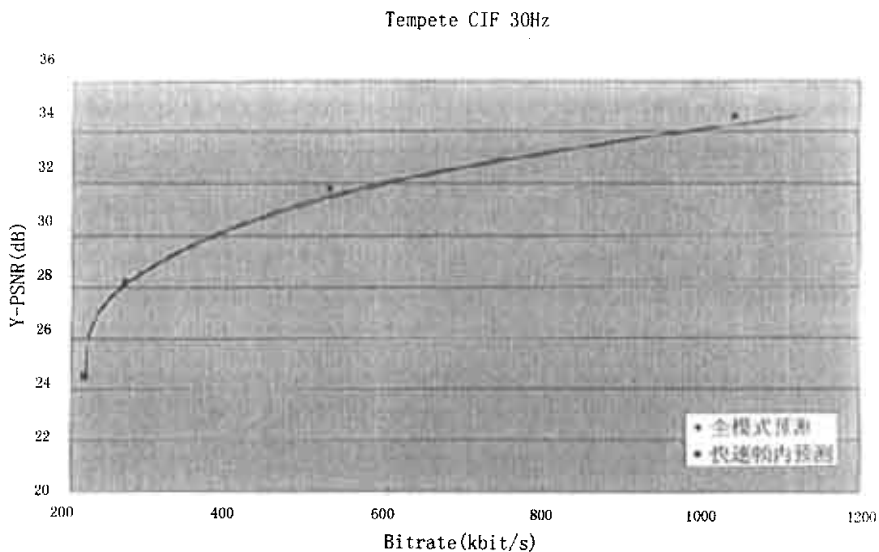


图 4-3 Tempete CIF 帧内预测算法 RD 曲线

4.2 帧间模式快速选择算法

H.264 编码效率的提升是以计算复杂度的增加为代价的,为了加速多块模式运动估计,本节中,提出了基于 16×16 块分裂的快速帧间模式选择算法,并综合考虑全 0 块、SKIP 模式与帧内模式。实验结果表明,新算法与其他算法比较,图像质量和码率只有少量变化,而编码速度有显著提高。

4.2.1 多块模式运动估计的观察

H.264 定义了 7 种不同大小的帧间宏块模式,以及 9 种 4×4 块的帧内预测模式、4 种 16×16 宏块的帧内预测模式、4 种色度宏块的帧内预测模式。在对 I 帧编码时,需要通过遍历所有可能的帧内编码模式组合,寻找率失真开销 RDcost 最小的那一种作为最优编码模式。而对于 P, B 帧编码,则需遍历所有的帧内和帧间编码模式。H.264 采用拉格朗日方法来选择最优运动矢量和最优编码模式,使得运动矢量和残差编码的比特分配最小^[4.4]。

JVT 发布的 H.264 参考模型 JM9.4,采用全搜索运动估计时,需要对每一种编码模式进行计算,包括: $\text{inter}16 \times 16$ 、 $\text{inter}16 \times 8$ 、 $\text{inter}8 \times 16$ 、 $\text{inter}8 \times 8$ 、 $\text{intra}4 \times 4$ 和 $\text{intra}16 \times 16$ 。特别的,对于 $\text{inter}8 \times 8$,需要进一步分割成更小的块模式如: 8×4 、 4×8 、 4×4 。

4.2.2 基于大块分裂的帧间预测快速算法

文献[4.5]中,提出了一种基于全 0 系数块检测的快速模式选择算法,并且图像质量和压缩率不变。算法中利用全 0 块的性质(AZCB: all-zero coefficients block),有效跳过无用块模式。此算法没有能够很好解决算法中使用的门限的问题,并且其算法只考虑了帧间模式。

在文献[4.6]中,作者根据帧内模式的预测信息和边缘映射以进行帧间模式选择。文献[4.7]对 7 种不同模式出现在最终编码中的概率进行了统计分析。统计数据显示,在 P 帧图像中,采用帧内模式作为最终编码模式的最大概率只有 9%,平均概率仅为 3%。因此,这种根据帧内预测模式信息以进行帧间编码模式选择的算法显然不是合适的选择。如果能够事先确定最佳的块模式是帧间模式,那么就不需要再检测或计算帧内模式,进而降低计算复杂度。Lee 等^[4.8]提出的算法,正是基于这种统计观察的结论,在选定了帧间编码方式后,根据帧间模式信息决定帧内模式的选择。并且他们只计算被认为有价值的帧内模式。此算法综合考虑的 SKIP 模式、帧内/帧间模式,但计算复杂度仍然较高;

对于多尺寸方块的运动向量估计, Li[4.9]提出以合并为基础的算法,用来降低 H.264 中的运动向量估计的计算复杂度。文章中对具有相同 MVs 的较小方块进行合并,由于帧间编码最终采用较大模式的可能性较大^[4.7],这种对任何图像都从 4×4 块进行合并的方法并不能做到最优。本文中,提出了一种更为优化的快速模式选择算法,对帧内预测采用上节中提到的基于局部边缘方向信息的快速算法,对帧间模式采用基于大块分裂的快速算法,综合考虑 SKIP 模式与全 0 块的性质,更进一步降低计算复杂度,同时图像质量和码率只有少量变化。

文献[4.7]的统计结果显示,对于不同运动性质的图像,采用 SKIP 与 16×16 模式作为最优编码模式的平均概率为 70%,同时采用 8×8 模式的比例也相当高。因此本文中的算法以此统计结果为基准,提出以 16×16 块为初始块进行分裂的算法进行帧间模式的快速选择,同时考虑全 0 块、SKIP 模式与帧内模式。

4.2.2.1 AZCB 的检测

AZCB 定义为^[4,5]:

$$AZCB(Block \quad 8 \times 8) = \begin{cases} 1 & \text{if } C8 \leq Tc8 \\ 0 & \text{otherwise} \end{cases} \quad (4.4a)$$

$$AZCB(Block \quad 16 \times 16) = \begin{cases} 1 & \text{if } C16 \leq Tc16 \\ 0 & \text{otherwise} \end{cases} \quad (4.4b)$$

$$\text{其中, } C8 = \sum_{i=0}^3 \sum_{j=0}^{15} cc(i, j), \quad C16 = \sum_{i=0}^3 C8(i)$$

$$cc(i, j) = \begin{cases} \infty & \text{if } level(i, j) > 1 \\ 3 & \text{if } level(i, j) = 1 \text{ and } run_before = 0 \\ 2 & \text{if } level(i, j) = 1 \text{ and } 1 \leq run_before \leq 2, \\ 1 & \text{if } level(i, j) = 1 \text{ and } 3 \leq run_before \leq 5 \\ 0 & \text{if } otherwise \end{cases}$$

C8 与 C16 分别表示 8×8 与 16×16 块的全部系数值, $cc(i, j)$ 为相应系数值,

$level(i, j)$ 表示量化变换系数的绝对值, run_before 表示在全部系数变为 0 前的量化距离。

由于 H.264 采用 4×4 整数变换, 对于 8×8 或者 16×16 宏块, 产生单系数非 0 块的可能性很大。如果一个较大模式的块通过检测判断为全 0 块, 那么就跳过此块的运动搜索。

4.2.2.2 SKIP 模式的检测

跳过宏块也就是不需进行编码, 而可以直接采用帧间预测的方法从参考帧恢复的宏块。H.264 的跳过宏块只能是 16×16 的块模式, 而且必须符合以下准则:

- 最佳的运动补偿块是 16×16 块;
- 运动估计的参考帧是当前帧的前一帧;
- 当前块 MV 与其预测 MV 相同;

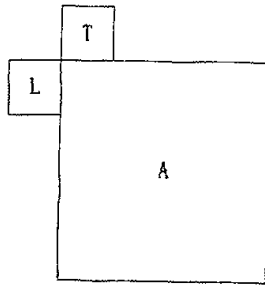


图 4-4 SKIP 模式检测示意图

如图 4-4 所示, A 块为当前编码宏块, 其运动向量为: $MV-A$, 其预测向量为: $PMV-A$ (可通过 H.264 运动向量预测方法计算得出)。L 块与 T 块分别是左边与上边最小块模式, 其运动向量分别为 $MV-L$, $MV-T$ 。

$MV-A$ 需满足以下要求:

$$\text{if } (MV-L == 0 \mid MV-T == 0) \quad MV-A = ZERO - MV;$$

else MV-A = PMV-A;

其中, ZERO - MV 为零向量。

➤ 帧间预测后的残余矩阵经整数变换与量化后全为 0。

通过对 SKIP 模式的检测,可大大加快运动估计的速度,特别是静止块较多的序列。

4.2.2.3 基于帧间模式信息的帧内模式预测

帧间编码模式利用的是像素的时间相关性,而帧内编码模式利用的是像素的空间相关性。所以,如果当前块有更强的空间相关性,那么其最终采用帧内编码模式的可能性就很大。

在 P 帧编码中,选择帧内还是帧间模式需要对图像的时空相关性进行计算。为了降低计算量, Lee 等^[4,8]引入平均边界误差 (ABE: average boundary error, 当前块的边缘像素与其相邻已编码块的边缘像素差值) 作为空间相关性度量, 平均比特率 (AR: the average rate, 编码运动补偿残差数据的平均比特数) 作为时间相关性度量。

AR 的计算公式如式 4.5 所示:

$$AR = \frac{1}{384} (\#bits \text{ of texture data}) \times \lambda \quad (4.5)$$

其中 $\lambda = 0.85 \times 2^{Qp/3}$, $384 = (16 \times 16)_{luma} + (8 \times 8)_{chroma} \times 2$

这里, AR 被定义为当前块纹理数据的比特数 (the bits of texture data), 因为在计算 RDcost 的过程中, 这个值已经被计算过了, 所以 AR 不需要重复计算。帧内预测利用的是相邻已编码块的像素值, 因此 ABE 的计算公式如式 4.6 所示:

$SBE(\text{Sum of Boundary Error}) =$

$$\begin{aligned} & + \sum_{i=0}^{15} |Y_{Orig}(x, y+i) - Y_{Rec}(x-1, y+i)| + |Y_{Orig}(x+i, y) - Y_{Rec}(x+i, y-1)| \\ & + \sum_{i=0}^7 |Cb_{Orig}(cx, cy+i) - Cb_{Rec}(cx-1, cy+i)| + |Cb_{Orig}(cx+i, cy) - Cb_{Rec}(cx+i, cy-1)| \\ & + \sum_{i=0}^7 |Cr_{Orig}(cx, cy+i) - Cr_{Rec}(cx-1, cy+i)| + |Cr_{Orig}(cx+i, cy) - Cr_{Rec}(cx+i, cy-1)| \end{aligned}$$

$$\therefore ABE = \frac{1}{64} SBE \quad (4.6)$$

其中 $64 = (16 \times 16)_{luma} + (8 \times 8)_{chroma} \times 2$,

Y_{Orig} , Cb_{Orig} , Cr_{Orig} 分别是当前块的亮度色度像素值, Y_{Rec} , Cb_{Rec} , Cr_{Rec} 分别表示相邻块已编码块的像素值。算法中假定:

if (AR < ABE) 忽略所有帧内模式的计算, 最终编码模式选择帧间块模式;

else 计算所有帧内块模式的 RDcost, 选择最佳帧内块模式并与已选定的最佳帧间块模式进行比较, 进而选择最终的最佳编码模式。

4.2.2.4 基于大块分裂的帧内/帧间预测模式选择算法

分裂法进行最佳帧间模式选择分两步进行:

➤ 步骤 1, 将 16×16 块分裂成 16×8 、 8×16 、 8×8 块如图 4-5(a) 所示。使用 ADSS^[4,10] 搜索算法计算 16×16 块的 ME。由于 16×16 块的 MV 将作为后续分裂过程中的预测 MV, 因此需要准确计算。分裂进程中, 如果 16×8 块与 8×16 块的 MVs 相等, 那么就将此 16×16 块的 MV 作为 8×8 块的 MV, 并跳转到步骤 2 进一步分裂; 否则计算 16×16 、 8×16 与 8×8 块的 RDcost, 并选择其

- 中一种作为最佳帧间编码模式；
- 步骤 2，将 8×8 块更进一步分裂成 8×4 、 4×8 、 4×4 块如图 4-5(b)所示。在此过程中，使用 SDSP^[4,11]搜索算法进行运动搜索，如果 8×4 块与 4×8 块的 MVs 相等，那么就将此 8×8 块的 MV 作为 4×4 块的 MV；否则计算 8×8 、 4×8 与 8×4 块的 RDcost，并选择其中一种作为最佳帧间编码模式。

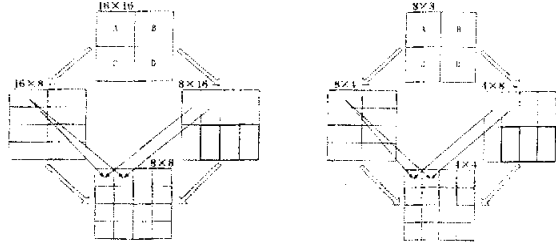


图 4-5(a) 16×16 块的分裂 图 4-5(b) 8×8 块的分裂

综上所述，基于大块分裂的帧间模式选择，同时考虑全 0 块、SKIP 模式与帧内模式的快速模式选择算法的总体算法流程如图 4-6 所示：

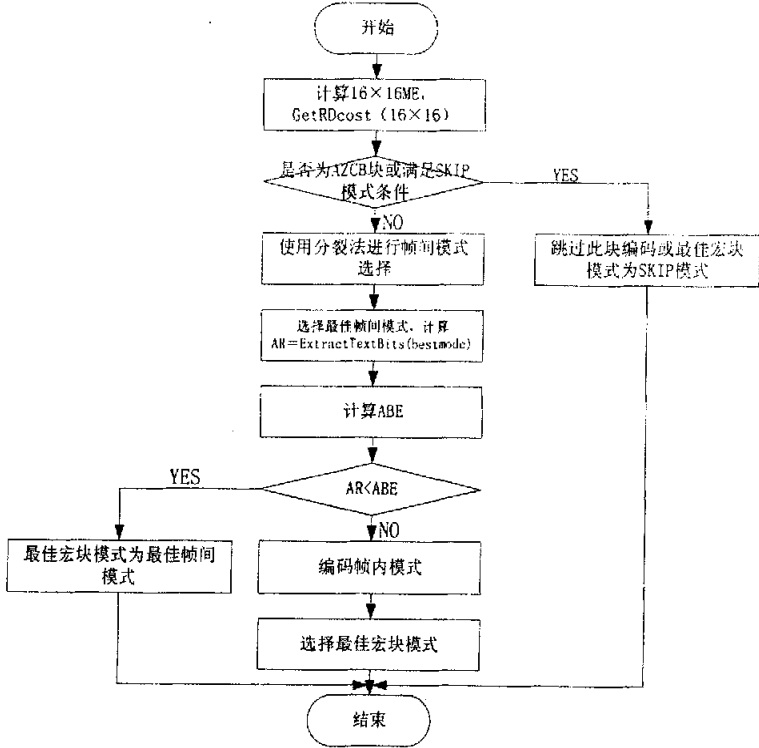


图 4-6 快速模式选择优化算法流程

4.2.3 实验结果与分析

对基于大块分裂的帧间预测模式快速选择算法的实验分析使用本文中的缺省软硬件环境，缺省编码参数为：采用1个参考帧、支持RDO优化、CABAC熵编码，测试序列为具有不同运动复杂度信息的Tempete 30Hz CIF、Mobile 30Hz CIF、Foreman 30 Hz QCIF与 Container 15Hz QCIF。

在H.264中，基于RDO的帧间模式选择的基本思想仍然是采用遍历的方式搜索所有7种可能的块模式，所以在这个过程中必然包含大量的冗余操作。统计结

果表明，在不同复杂度的序列中大块模式所占比例更大，因此最佳的模式选择算法应从16×16逐步向更小的块进行，并且在相对较大的块中找到足够好的匹配后，立即跳出其后所有的较小块模式的搜索。本文中采用基于大块分裂的帧间模式选择算法，同时考虑全0块、SKIP模式与帧内模式。

实验分析结果如表4-4所示，采用本文中的快速模式选择算法可以有效的减少预测的候选模式数，在作IBPBP编码时可以节约大约50%-70%的编码时间，同时，PSNR和码率与原JM软件遍历的模式选择算法相比，基本不变。图4-6表明，在不同码率情况下，基于局部边缘方向信息的快速帧内预测编码模式选择的快速算法在降低候选块模式CBPMs的情况下，PSNR和码率基本不变。

表 4-4 快速模式选择算法性能比较

序列	Bitrates(%)	Y-PSNR(%)	Times(%)
Container QCIF	+1.27	-0.01	-69.5
Foreman QCIF	+1.01	-0.01	-61.7
Mobile CIF	+1.25	0.00	-52.4
Tempete CIF	+1.43	-0.02	-47.0

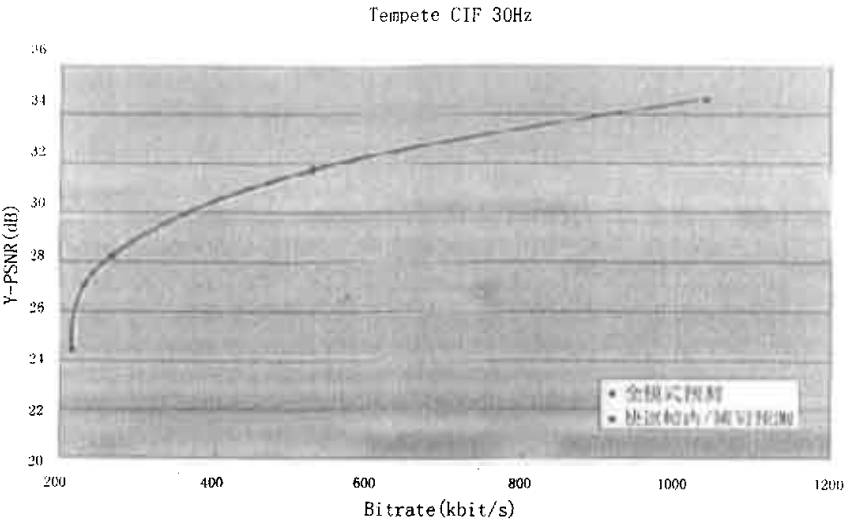


图 4-6 Tempete CIF 快速模式选择算法的 RD 曲线

4.3 本章小结

H.264 定义了 9 种 4×4 块的帧内预测模式(M4=9)、4 种 16×16 宏块的帧内预测模式(M16=4)、4 种色度宏块的帧内预测模式(M8=4)，以及 7 种不同大小的帧间宏块模式，为了提高编码效率，编码时对每一种模式都进行运动搜索，并利用利用率失真优化（RDO）分别计算编码代价，选择编码代价最小的模式都作为帧内或帧间编码模式。巨大的计算量使模式选择成为帧内/帧间编码的瓶颈所在。

本章中，根据 H.264 的特点，采用基于局部边缘方向信息的帧内预测模式选择快速算法，提出了基于大块分裂的帧间预测模式快速选择算法，同时综合

考虑全 0 块、SKIP 模式与帧内预测。对不同格式和运动复杂性的序列的测试结果表明,本章中的算法可以节约 50%到 70%的编码时间,同时保证编码图像质量和码率基本不变的情况,有效降低计算复杂度。

但同时还应该说明的是对于 H.264 的预测模式选择的快速算法,从实时编码的角度看还相有一定的距离,而这一切有待于今后更进一步的研究加以解决。

参考文献

- [4.1] Iain E.G.Richardson, The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression-Video Coding for Next-generation Multimedia[M/OL].
[http://nmlab.zju.edu.cn/~mm/book/H.264 And MPEG-4 Video Compression Video Coding For Next Generation Multimedia.pdf](http://nmlab.zju.edu.cn/~mm/book/H.264%20And%20MPEG-4%20Video%20Compression%20Video%20Coding%20For%20Next%20Generation%20Multimedia.pdf).
- [4.2] Feng PAN, Xiao LIN, Rahardja SUSANTO, Keng Pang LIM, Zheng Guo LI, Ge Nan FENG, Da Jun WU, and Si WU, Fast Mode Decision for Intra Prediction. JVT-G013, 7th Meeting: Pattaya II, Thailand, 7-14 March, 2003.
- [4.3] Zhang Yong-dong, Dai Feng, Lin Shou-xun, Fast 4×4 intra-prediction mode selection for H.264. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on, Volume:2, 27-30 June 2004. Pages:1151-1154 Vol.2.
- [4.4] Thomas, Wiegand, Heiko Schwarz and Anthnoy Joch, Rate-Constrained Coder Control and Comparison of Video Coding Standards. IEEE Trans. CSVT, VOL.13, pp.688-703, July 2003.
- [4.5] Kim Y.-H., Yoo J.-W., Lee S.-W., Shin J., Paik J., Jung H.-K., Adaptive mode decision for H.264 encoder. Electronics Letters, Volume: 40, Issue: 19, 16 Sept. 2004, Pages:1172-1173.
- [4.6] K.P. Lim, S. Wu, D.J. Wu, S. Rahardja, X. Lin, F. Pan, ZG Li, Fast INTER Mode Selection. Doc. I020, Sep. 2003.
- [4.7] Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-Yi Chient, Shyh-Yih Ma, Chun-Fu Shen, Liang-Gee Chen, Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264. Proceedings of 2003 IEEE International Conference; on Acoustics, Speech, and Signal Processing, Volume:3, Pages:145-148.6-10 April 2003.
- [4.8] Jeyun Lee, Byeungwoo Jeon, Fast mode decision for H.264. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume 2, 27-30 June 2004 Page(s):1131-1134 Vol.2.
- [4.9] 李升家, 加速 H.264 运动向量估计模式的选择. 台湾成功大学硕士学位论文, 2004 年 6 月.
- [4.10] J. Xin, M.T. Sun, and V. Hsu, Diversity-based Fast Block Motion Estimation. Proc. IEEE International Conference on Multimedia & Expo (ICME), Baltimore, MD, USA, July 2003.
- [4.11] S. Zhu and K.K. Ma, A new diamond search algorithm for fast block-matching motion estimation. IEEE Trans. Image Processing, vol. 9, pp. 287-290, Feb. 2000.

第5章 基于 MMX/SSE 技术的代码优化

视频编解码程序的优化通常包括算法优化、多媒体指令优化、代码层次优化和编译器的优化^[5.1,5.2]。本章中,先使用 Intel 的 MMX/SSE 技术对 H.264 编解码器的主要的耗时模块进行了改写,包括: SAD 计算、正反整数变换、像素内插等,并对部分程序结构进行了调整,包括程序的内存分配与 I/O 操作,程序的初始化,以及减少函数的调用,优化程序中的循环结构,尽量减少计算量,减少 CPU 的开销,有效的提高了编解码效率。

5.1 MMX/SSE 技术概述

5.1.1 MMX/SSE 指令集简介

MMX (Multi Media Extension, 多媒体扩展) 指令集是 Intel 公司于 1996 年推出的一项多媒体指令增强技术。Intel 的 MMX™ 技术是对 Intel 体系结构(IA)指令集的扩展^[5.3]。该技术使用了单指令多数据技术(SIMD)技术,以并行方式处理多个数据元素,从而提高了多媒体和通讯软件的运行速度。MMX™ 指令集提供了 8 个 64 位通用寄存器,命名为 MM0~MM7,在此基础上构成了新的数据类型,见图 5-1 所示,增加了 57 条新的操作码。

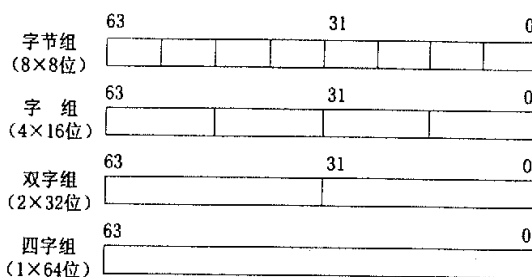


图 5-1 MMX™ 的成组数据类型

MMX™ 技术提高了很多应用程序的执行性能,例如活动图像、视频会议、二维图形和三维图形。几乎每一个具有重复性和顺序性整数计算的应用程序都可以从 MMX™ 技术中受益。对于 8 位、16 位和 32 位数据元素的处理,改善了程序的性能。一个 MMX™ 指令可一次操作 8 个字节,且在一个时钟周期内完成两条指令,也就是说,可在一个时钟周期内处理 16 个数据元素。另外,为增强性能,MMX™ 技术为其它功能释放了额外的处理器周期。以前需要其它硬件支持的应用程序,现在仅需软件就能运行。给更小的处理器占用率更高程度的并发技术提供了条件,在当今众多的操作系统中这些并发技术得到了利用。在基于 Intel 的分析系统中,某些功能的性能提高了 50% 到 400%。这种数量级的性能扩展可以在新一代处理器中得到体现。在软件内核中,其速度得到更大的提高,其幅度为原有速度的三至五倍。

MMX 的缺点: 由于 MMX 的运算指令必须在数据配对整齐的时候才能使用,所以使用 MMX 指令要比普通的汇编指令多余许多分组配对的指令,如果运算不是特别的整齐的话,就要浪费大量的时间在数据的配对上,所以说 MMX 指令也不是万能的,也有其很大的缺陷。同时 MMX 指令在处理 16 位数据的时候才能发挥最大的作用,处理 8 位数据要有一点技巧。而处理 32 位数据,MMX

指令几乎没有什么加速能力。

SSE (Streaming SIMD Extensions, 单指令多数据流式扩展) 是 Intel 提出的继 MMX 之后的 CPU 指令集, 能够有效增强 CPU 浮点运算的能力。最早应用在 PIII 系列 CPU 上, 现在已经得到了 Intel PIII、P4、Celeron、Xeon、AMD Athlon、Duron 等系列 CPU 的支持。具有 Intel SSE 指令集支持的处理器有 8 个 128 位的寄存器, 每一个寄存器可以存放 4 个 (32 位) 单精度的浮点数。SSE 同时提供了一个指令集, 其中的指令可以允许把浮点数加载到这些 128 位的寄存器之中, 这些数就可以在这些寄存器中进行算术逻辑运算, 然后把结果放回内存。

SSE 指令集包括了 70 条指令, 其中包含提高 3D 图形运算效率的 50 条 SIMD (单指令多数据技术) 浮点运算指令、12 条 MMX 整数运算增强指令、8 条优化内存中连续数据块传输指令。理论上这些指令对目前流行的图像处理、浮点运算、3D 运算、视频处理、音频处理等诸多多媒体应用起到全面强化的作用。

5.1.2 下一代处理器指令集 SSE2

基于 P4 CPU 的 SSE2 指令集, 在 SSE 的基础上又引入新的指令, 使得一条指令可以同时多达 16 个 8 位数据进行相同的运算。

SSE2 有两个不同的部分所组成: SSE 部分和 MMX 部分。前者主要负责处理浮点数, 后者则专门计算整数。SSE2 寄存器容量是 MMX 寄存器的两倍, 寄存器存储数据也增加了两倍, 在指令处理速度保持不变的情况下, 通过 SSE2 优化后的应用软件运行速度也将提高 2 倍。由于 SSE2 指令集与 MMX 相兼容, 因此被 MMX 优化过的程序很容易被 SSE2 再次深层次的优化。

5.1.3 基于 MMX、SSE 与 SSE2 技术的程序设计

Visual Studio .NET 2003 提供了对 MMX、SSE 与 SSE2 指令集的编程支持, 从而允许用户在 C++ 代码中不用编写汇编代码, 而是使用 intrinsic 函数封装, 就可直接使用 SSE/SSE2 指令的功能^[5,4]。C++ 编程人员在使用 MMX、SSE 与 SSE2 指令函数编程时不必关心这些 128 位的寄存器, 完全可以使用 128 位的数据类型“__m128”和一系列 C++ 函数来实现这些算术和逻辑操作, 而决定程序使用哪个 MMX 或 SSE/SSE2 寄存器以及代码优化是 C++ 编译器的任务。

本文中, 正是使用 MMX、SSE 与 SSE2 指令集的 intrinsic 函数封装对 H.264 编码器中的耗时较多而又非常适合应用 MMX、SSE 与 SSE2 技术的模块, 如: 正反整数变换与量化、运动估计中的 SAD 值计算以及 1/4 和 1/8 像素内插等进行了优化。实验表明, 优化后速度有了明显提高。

5.2 H.264 视频编码器的优化分析

5.2.1 SAD 计算的代码优化

由第三章的 H.264 视频编码器复杂度分析的结果可以知道, 运动估计是 H.264 视频编码器中运算量最大的模块之一, 上一章对运动估计中帧内/帧间预测模式的选择提出了快速算法。在运动估计过程中, 常采用 SAD (Sum of Absolute Differences) 作为差值评价标准, 假设运动补偿的块尺寸为 $N \times N$, 其计算公式如式 5.1 所示:

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (5.1)$$

其中, C 是当前块 (即需要进行运动补偿的块), R 是参考块。从公式可以看出, SAD 的计算包含大量的加法和绝对值以及求和运算, 这正是 SSE/SSE2 的强项。由于 H.264 的运动补偿采用了七种不同的块模式, 对不同的块模式的 SAD 的计算, 采用不同的 intrinsic 函数进行优化。对 16×16 和 16×8 的块, 由于其每行 16 个像素连续存放, 我们使用 `_mm_add_epi8` 函数, 这个函数可以一次计算 2 组 8 点的 SAD 值。

对于 8×16 、 8×8 、 8×4 的块, 每行只有 8 个像素, 可以采用 `_mm_add_epi16` 函数, 这个函数一次可以计算 2 组 4 点 SAD 的值 (丢弃符号位)。

至于 4×8 、 4×4 的块则采用 `_mm_add_epi32` 函数, 同样丢弃符号位, 一次计算 4 点 SAD 的值。

5.2.2 正反整数变换的优化

在第三章中的视频编码器复杂度的分析中发现 H.264 的整数变换模块是另一个耗时较多的模块。H.264 中的 DCT 变换采用的是整数变换, 相对于传统的 DCT 变换, 其运算量已有所减少, 但在 H.264 编码器的编码过程的总运算量仍然占到 12%。

H.264 中使用的 4×4 整数变换如式 5.2 所示, 其中 X 为输入块数据, C 为变换系数, C^T 是 C 的转置, Y 为变换输出。

$$Y = CXC^T \quad (5.2)$$

其中,

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix},$$

对上述变换式进行改写, 如式 5.3 所示:

$$Y = (X^T C^T) C^T \quad (5.3)$$

通过 2 次循环可以完成上式的计算, 第一次循环时, 计算 $X^T C^T$, 结果保存在 X 中, 第二次再计算 $X^T C^T$ 。

设 $X = (\text{data}[0], \text{data}[1], \text{data}[2], \text{data}[3])$,

$$C^T = (\text{matrix}[0], \text{matrix}[1], \text{matrix}[2], \text{matrix}[3]),$$

X 转置后结果依然保存在 $\text{data}[n]$ 中 (n 从 0 到 3)。

$X^T C^T$ 分四次计算 (一次循环), 每一个 $\text{data}[n]$ 乘 X^T 得一个 4×4 矩阵, 这是一个中间数据, 需要将这个 4×4 矩阵转置后计算行向量 (刚好是一个 `_m128` 数据) 相加, 结果保存在第一行中。这样一次循环就可以计算出 $X^T C^T$ 矩阵相乘的一行值。通过 4 次循环完成 $X^T C^T$ 的计算。

针对 4×4 整数变换的特点, 并充分利用 SSE2 指令的特性, 优化改写的方法如下:

- 数据表示: 由于输入输出数据和中间结果不会超过 16 位, 使用 16 位来表示变换中的输入输出数据和中间结果, 可以提高数据访问的效率;
- 寄存器组织: 一个 SSE2 寄存器可以存储 8 个 16 位数据, 所以一条 SSE2

指令可以同时处理两行数据的运算，极大的提高了变换速度；

- 指令的选择：由于 SSE2 提供的乘累加指令不能一次累加所有中间结果，所以优化中没有使用 `_mm_madd_epil6`，而使用 `_mm_add_epil6` 和 `_mm_mullo_epil6` 指令结合，充分利用了寄存器长度并减少指令个数，否则需要额外的移位和加法。

5.2.3 其他模块的优化算法

5.2.3.1 初始化

数据的初始化是每个项目都要进行的工作，在视频编码器中，也不例外。在通常的程序中，数据的初始化占据的运行时间不会太长，但是在视频编码器中，需要为相当一部分全局变量申请空间，并进行初始化。这是一个对编码器进行赋初值的过程，与此同时，在程序运行到一定阶段，如当前帧为 1 帧，则需要对参考帧置零。这些数据的初始化或复位的过程，在整个编码器中还是占据了一定的开销。

对这一问题的解决方法，一是在申请空间时，尽量减少申请的次数，对于可以重复使用的地址空间，只需要申请一次，重复利用，避免原来测试模型中多次申请、释放空间的情况；二是利用同时拷贝一块数据，而不是利用循环，对每一个数据进行赋值，加快了初始化或数据复位的速度。

5.2.3.2 减少函数的调用

函数的嵌套调用对系统资源的要求较高，由于函数的调用是一个跳转的过程，不仅要传递参数，还要保护跳转前的程序状态，所以减少函数之间的相互调用也是很有必要的。对于一些操作简单，调用次数不多的函数，将被调用到的函数体直接放到主掉函数中，不仅可以减少参数的传递，避免跳转，在某些情况下还可以减少数据的赋值操作，节约了系统资源，提高系统效率，减少编码时间。

5.3.3.3 循环结构的优化

在整个 JM9.4 参考编码器的代码中，循环结构占据了很大的部分。对于循环结构，优化的原则是：

- 尽量不要采用循环嵌套，减少循环次数，增加循环内部操作。
- 对循环进行合并、调整。当相邻的两个循环有相关性时，并且循环的次数也有一定的关联，可以把两个循环的内部操作合并，减少了 for 语句的同时也就减少了计算上的冗余。对于 if 语句也可以用同样的方法，尽量减少程序中的判断；对于效率较低的 switch 分支，可考虑函数指针。
- 程序的判断分支放在循环体外面。判断分支如果放在循环体内部不利于 CPU 的分支预测机制，所以在可能的情况下，将判断分支放在循环体的外面，也可以对程序的运行速度有很大的帮助。

5.2.4 优化效果及分析

本节中，先使用 Intel 的 MMX/SSE 技术对 H.264 编解码器的主要的耗时模块进行了改写，包括：SAD 计算、正反整数变换等，并对部分程序结构进行了调整，包括程序的初始化，以及减少函数的调用，优化程序中的循环结构，尽量减少计算量，减少 CPU 的开销。表 5-1 是各模块优化前后在总运算量中所占的比重的对比。从表中可以看出，优化后运动估计在总运算量中的比重有了很多幅度的下降，整数变换模块的比重也有不同程度的下降。图 5-2 显示了编码器优化前后的编码性能的比较。实验使用的测试序列长度均为 100 帧，时间单

位为秒，帧率单位为帧/s。

从图 5-2 可以看出，经过优化的编码器比纯 C 代码的编码器速度有了很大的提升。平均编码速度提升了 5~10 倍。

表 5-2 各主要耗时模块占总运算量在优化前后的比较(单位：%)

序列	运动估计		整数变换		其他	
	优化前	优化后	优化前	优化后	优化前	优化后
Container QCIF	31.23	20.51	12.39	4.12	56.38	75.37
Foreman QCIF	34.79	22.48	11.35	4.85	53.86	72.67
Mobile CIF	60.17	31.54	6.12	6.87	33.71	63.59
Tempete CIF	64.72	35.91	5.25	5.76	30.03	58.33

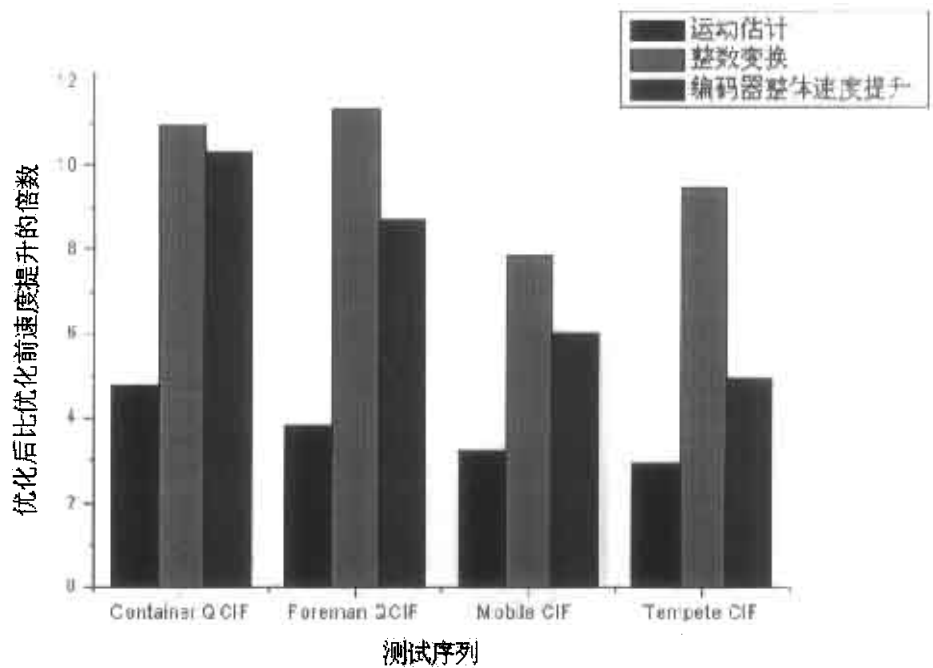


图 5-2 H.264 编码器优化后比优化前速度提升的倍数

5.3 H.264 视频解码器的优化分析

5.3.1 像素插值

H.264 的运动估计从支持亮度分量的 1/4 像素精度提高到支持色度分量的 1/8 像素精度。H.264 编码器在把参考帧存入帧存之前，要对整幅图像在水平方向和垂直方向上分别进行 4 倍上取样，使图像达到 1/4 精度。其中上取样是通过插值来实现的，利用 6 抽头 FIR 滤波器[1,-5,20,20,-5,1]产生 1/2 像素点，在此基础上继续利用 1/2 像素双向线性内插产生 1/4 像素点的值，利用 4 抽头滤波器产生 1/8 像素点的值，如图 5-3 所示。

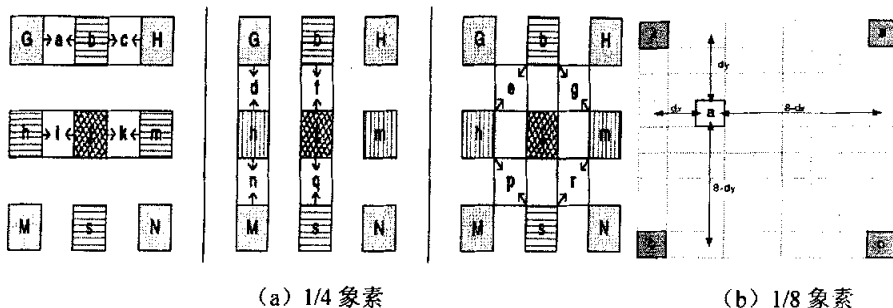


图 5-3 1/4 和 1/8 像素精度预测的插值图

而在解码器中，不需要像编码器那样对整个参考帧进行上取样，帧存中只保存参考帧的整数像素表示。计算 1 个帧间编码 4×4 块的预测块时，首先根据相应的参考帧号在帧存中选择正确的参考帧，如果该块的运动矢量指向整数位置，那么直接复制参考帧中相应的整数像素点的值就可以构成预测块，否则就要根据运动矢量指向的分数位置选择插值方法，并进行即时在线插值。由于运动矢量支持 $1/4$ 像素精度，所以除了整数像素点 G,H,M,N 外，还有 15 种分数像素点，如图 5- 中的小写字母点，分数预测块相应也有 15 种，且每种预测块中只包含一种类型的点。

1/4、1/8 像素插值的过程中使用了很多的乘法、加法运算，因此这个预测过程的运算量也很大。针对 1/4 和 1/8 像素插值的特点，采用 intrinsic 封装中的移位操作的函数 `_mm_slli_epi16` 等以及加法操作的函数进行改写。优化后的像素插值的过程如图 5-4 所示：

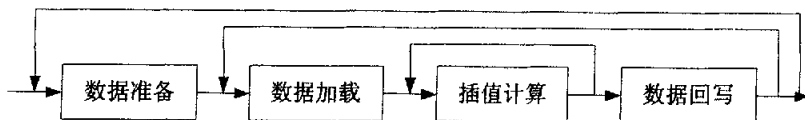


图 5-4 插值优化过程示意图

其中数据准备负责准备计算所需要的整数像素数据,必要时用边界像素代替引用到的边界外像素,而数据加载直接从内存中把准备好的数据读入到MMX寄存器中。从图5- 中可以看出,准备好一次数据后,其余3个部分可以根据需要多次执行,同样每加载一次数据都可以用于多次插值计算,而多次计算的结果将一次性回写入内存。这种层次结构具有模块清晰、加载后的数据可以重复利用、内存读写次数大大减少等多项优势。

在整数像素准备时首先要明确的是需要的整数像素个数。 $1/4$ 像素点由于依赖于整数像素点和 $1/2$ 像素点, 可以重复利用已准备好的整数像素点和已计算好的 $1/2$ 像素点的值, 一般不增加需要的整数像素个数。

在插值优化过程中, 有大量的数据转置操作, 其实数据转置并不是一个单独的步骤, 而是融合在数据准备和数据回写过程中进行的, 只是数据准备主要针对整数像素数据, 而数据回写主要针对中间数据, 它们本质上都是把垂直方向上内存离散的数据调整到水平方向上的连续内存中。这样两个方向的插值操作可以通过巧妙组织数据而统一起来, 大大降低了整体算法复杂度。

5.3.2 内存分配与 I/O 操作

内存分配与 I/O 操作也是 H.264 视频解码器中占用时间较大的模块, 需要进行优化。

JM 在分配 3 维内存结构时,所分配的内存是不连续的,这不利于使用 memset 或 memcpy 等函数进行批量内存操作。为此,笔者修改了 JM 中 3 维内存分配函数,使内存连续分配。

JM 在进行运动补偿时,使用运动向量在参考图像中寻找预测块。当运动向量指向参考图像以外时,预测块超出参考图像边界的那些点使用最近的边界点代替^[5,6]。这在 JM 中是通过数组下标取 min 和 max 函数实现的,无形中增加了处理时间。为此,考虑将参考图像的四周各扩展若干字节,扩展出来的像素点取参考图像中的最近的边界点。扩展后坐标原点发生了移动,需要对参考图像首地址的指针进行调整,使通过下标[0][0]访问到的像素仍然是未扩展前下标为[0][0]的像素。在实际中,运动向量的有效范围很大,如在 Level 4 下,运动向量的水平分量和垂直分量的范围分别为[-2 048, 2 047.75]和[-512, +511.75]^[5,6],但考虑到当运动矢量使计算一个预测块所需的像素完全处于参考图像以外,那么不论运动向量的水平分量或垂直分量延伸多远,所得到的预测块都是相同的。因此,将扩展的字节数取为 16 就可以了,同时将运动向量的两个分量分别裁减到不超过参考图像左边和上边的边界 8 个字节以及下边和右边的边界 2 个字节。扩展后的参考图像见图 5-5。

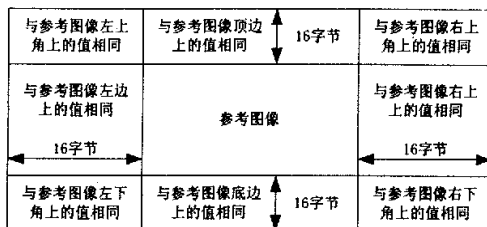


图 5-5 扩展后的参考图像

在 H.264 视频解码器中,I/O 操作模块的优化的方法是参考 MPEG2/4 编解码器中对数据读写的方法,利用缓冲,对数据的读写也作了改进。

在 JM9.4 中,数据的读取是以位为单位的,读取一段数据需要通过大量的循环、移位和位操作。这里,将此种数据读取的方法进行改进,每次读取一定长度的数据,如 4 个字节,然后通过移位的方法,得到需要的数据段。利用这种方式对数据进行读取,减少了循环的次数,提高了运行效率。JM9.4 中数据的输出是以字节为单位的,利用循环分别输出 Y、U、V 三个分量,循环内部操作简单,而循环语句本身会占据较多的 CPU 时钟,所以采用了输出缓冲的方法,先将当前输出帧的 Y、U、V 分量写入缓冲,然后一次写到输出文件,提高了文件输出的速度。

5.3.3 其他模块的优化算法

以上主要对 H.264 解码参考模型中占用处理时间较多的模块进行了改写,而程序细节的优化对解码速度的提高也显得非常重要,可以参考上节中对编码器程序的优化对解码器中的部分程序进行改写,如:尽量采用 memcpy,memset 等库函数减少循环减少内存的访问与修改;尽量不要采用循环嵌套,而使用循环展开;尽量减少程序中的判断;对于效率较低的 switch 分支,可考虑函数指针;尽量减少计算量,减少 CPU 的开销。

5.3.4 优化效果及分析

经过代码和算法优化后，H.264 解码器性能提高了 10 倍，总体性能增益和各个模块速度提升见表 5-3 所示。由于解码器是编码器的逆过程，通过码流得到图像，故不会产生图像信噪比的下降，解码后的图像质量完全和编码时生产从图像一致。

表 5-3 Foreman.264 解码测试各模块优化前后的性能对比

		像素插值	内存分配	I/O 操作	其他	总计
优化前	比例(%)	25.5	21.8	13.2	39.5	100
	时间(s)	6.590	5.634	3.412	10.209	25.845
优化后	比例(%)	18.3	15.1	16.8	49.8	100
	时间(s)	0.434	0.358	0.398	1.180	2.370

5.4 本章小结

视频编解码程序的优化通常包括算法优化、多媒体指令优化、代码层次优化和编译器的优化。本章中，先使用 Intel 的 MMX/SSE 技术对 H.264 编解码器的主要的耗时模块进行了改写，包括：SAD 计算、正反整数变换、像素内插等，并对部分程序结构进行了调整，包括程序的内存分配与 I/O 操作，程序的初始化，以及减少函数的调用，优化程序中的循环结构，尽量减少计算量，减少 CPU 的开销，有效的提高了编解码效率。测试结果表明，对于编码器，编码速度比优化前提高了 5-10 倍；而解码器的解码速度比优化前提高了 10-15 倍。

参考文献

[5.1] Richard Gerber, The Software Optimization Cookbook. Copyright@Intel Corporation. 2002.

[5.2] Ville Lappalainen1 and Timo D.H, Unified Method for Optimization of Several Video Coding Algorithms on General-Purpose Processors. In P roc.2002,IEEE Int. Conf Information Technology, vol.1, pp .431-439,Jan.2002.

[5.3] Intel 软件说明书（Intel Software manuals）[EB/OL].
<http://developer.intel.com/design/archives/processors/mmx/index.htm>,2003.

[5.4] MSDN, SSE 技术主题[EB/OL].
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vclang/html/vcrefstreamingsimdextensions.asp>,2003.

[5.5] Document JVT-D153: JVT/H.26L decoder complexity analysis Joint Video Team of ISO/IEC MPEG & ITU-T VCEC, Klagenfurt, Austria, 22~26 July, 2002.

[5.6] ITU-T Rec. H.264 ISO/IEC 14496-10 AVC. Joint Final Committee Draft (JFCD) of Joint Video Specification. JVT Doc: JVT-D157, 2002.

第6章 总结与展望

6.1 全文总结

H.264 是国际上关于视频编码的最新国际标准,拥有优秀的编码效率和广阔的应用前景,必定会广泛应用在 DTV/HDTV 和 IP 视频、移动通信等诸多领域。但是由于 H.264 中引入了多种新的编码技术,使得编码器的计算量急剧增加。如何降低 H.264 实现复杂度,使得它可以有效的应用在实际的通信系统中,已经成为学术界普遍关注的课题。本文以 H.264 为研究对象,针对如何降低 H.264 编码器复杂度的关键问题进行一些探讨,在深入研究 H.264 的标准文档、性能和计算复杂度的基础上,对 H.264 的帧内/帧间预测模式选择算法进行了研究,采用基于局部边缘方向信息的帧内预测模式选择快速算法,提出了基于大块分裂的帧间预测模式快速选择算法。采用本文中的快速模式选择算法可以有效的减少预测的候选模式数,在作 IPBPB 编码时可以节约大约 50%-70%的编码时间,同时图像质量和码率基本不变。同时,本文中又对 H.264 的参考软件编解码器的主要耗时模块与程序结构进行了优化调整,有效减少计算量与降低 CPU 的开销,提高了编解码效率。

本论文的主要研究工作概括如下:

第二章简要说明了视频数据压缩的必要性和可能性,并介绍了视频压缩编码的基本方法,基于块匹配混合编解码器的典型结构以及现有主要视频编码标准的特点。

第三章首先给出了 H.264 视频编解码器的结构框图,对其中的主要功能模块及其包含的新技术进行了详细论述。接着对 H.264 视频编解码器的性能与复杂性进行了分析,对主要模块进行耗时统计以指出编解码器瓶颈所在,为后续优化提供指引。

第四章根据 H.264 的特点,采用基于局部边缘方向信息的帧内预测模式选择快速算法,提出了基于大块分裂的帧间预测模式快速选择算法,在保证编码图像质量和码率基本不变的情况下,有效降低计算复杂度。

最后以 JVT 发布的 H.264 参考模型 JM9.4 为基础,使用 Intel 的 MMX/SSE 技术对 H.264 编解码器的主要的耗时模块进行了改写,包括: SAD 计算、正反整数变换、像素内插等,并对部分程序结构进行了调整,包括程序的内存分配与 I/O 操作,程序的初始化,以及减少函数的调用,优化程序中的循环结构,尽量减少计算量,减少 CPU 的开销,有效的提高了编解码效率。

6.2 工作展望

本文以 H.264 为研究对象,在深入研究 H.264 的标准文档、性能和计算复杂度的基础上,对 H.264 的帧内/帧间预测模式选择算法进行了研究,采用基于局部边缘方向信息的帧内预测模式选择快速算法,提出了基于大块分裂的帧间预测模式快速选择算法,并在此基础上对 JVT 发布的联合参考软件模型进行了优化。

作为一种新的视频压缩编码标准, H.264 表现出了非常卓越的编码压缩性能, 同时也显示出巨大的计算开销和内存开销。因此如何根据 H.264 的特点, 找到适合 H.264 的快速算法就成为当前研究的一个热点。下面提出本课题进一步研究的设想:

➤ 编码模式选择的快速算法

H.264 定义了 9 种 4×4 块的帧内预测模式($M4=9$)、4 种 16×16 宏块的帧内预测模式($M16=4$)、4 种色度宏块的帧内预测模式($M8=4$), 以及 7 种不同大小的帧间宏块模式, 为了提高编码效率, 编码时对每一种模式都进行运动搜索, 并利用利用率失真优化(RDO)分别计算编码代价, 选择编码代价最小的模式都作为帧内或帧间编码模式。巨大的计算量使模式选择成为帧内/帧间编码的瓶颈所在。虽然本文采用的方法可以有效的减少模式选择的开销, 但从实时的角度看还是远远不够的, 还需要更有效的算法。

➤ 码率控制

视频传输面临的问题就是如何在给定带宽和低延迟条件下实时传输数据, 并保证良好和稳定的接收端质量。这就需要在编码端采用合适的码率控制策略, 对编码参数进行调整, 使码率尽可能和目标码率一致, 同时获得高的视频质量。以往的视频压缩标准都对其进行了广泛的研究。然而, 由于 H.264 的编码算法与以往的标准有些不同, H.264 中采用了 RDO 技术, 宏块的编码模式与量化参数的选取有关, 因此以往各种经典的码率控制方法对 H.264 并不适用, 使得在其中实现高效的码率控制更加困难。虽然码率控制不是标准的必要组成部分, 但在视频编码的实际应用领域中, 码率控制是一项非常重要和实用的技术。

➤ CABAC 熵编码模块的研究和优化

基于内容的自适应二进制算术编码 CABAC 是 H.264 中可选的熵编码方式之一, 它的设计基于二进制定化、选择内容模型、自适应算术编码并自动更新概率等 3 个步骤, 充分考虑了视频流的相关统计特性, 把熵编码的性能尽可能的发挥到了极致。通过各种实验可以证明, CABAC 的编码效率均高于 CAVLC, 尤其是在 QP 参数较大的, 码率较低时, CABAC 的高编码效率优势更明显。但 CABAC 的计算开销相对 CAVLC 需增加 40%左右, 目前还没有很好的快速算法。

➤ SP/SI 帧技术

SP-slice 的使用, 在不明显增加码率的前提下, 为不同图像的视频流之间、同一图像序列的不同比特率的码流之间、或同一视频码流的随机访问提供了快速高效的切换方式, 在满足人的视觉感受的前提下, 避免使用 I-slice, 节约了时间和资源。SI-slice (用于视频流转换的 I-slice) 的使用方法与 SP-slice 基本相同, SI-slice 主要运用在两个完全不相关、无法使用运动补偿预测的序列的切换。正式由于 SP/SI 帧的使用, H.264 在网络友好性方面才有了很大改进: 对网络传输具有更好的支持, 支持灵活的流媒体服务应用; 具有很强的抗误码性能, 以适应在噪声干扰大、丢包率高的无线信道中传输。但是因为 SP/SI 帧的算法复杂度较高, 所需执行时间较长, 对算法的优化和完善将成为进一步研究的方向。

附 录

攻读硕士学位期间发表的学术论文：

- (1) 孙同伦, 尹东, 唐群元. 基于 Gabor 滤波增强的指纹识别. 山东医学工程, 已录用.
- (2) 孙同伦, 尹东, 候蕾. H.264 软件编码器的设计与优化. 中国电子学会第十九届电路与系统学术年会会议论文, 已录用.

科研工作：

- (1) 北京永新同方信息工程有限公司监测事业部：“H.264 视频编解码器优化”.
- (2) 北京永新同方信息工程有限公司监测事业部：“MPEG2 TS 码流录制”.
- (3) “十五”中科院知识创新方向性研究：“信息融合关键技术预先研究”.
- (4) “十五”期间总装备预研项目：“数据融合处理技术”.

主要研究内容有：

- (1) H.264 识别编解码器的优化.
- (2) MPEG2 TS 码流的录制.
- (3) SAR 图像中道路的识别.
- (4) 中高分辨率图像中铁路编组站的识别.

作者: [孙同伦](#)
学位授予单位: [中国科学技术大学](#)

参考文献(72条)

1. 参考文献

2. [钟玉琢, 王琪, 贺玉文](#) [基于对象的多媒体数据压缩编码国际标准--MPEG-4及其校验模型](#) 2000
3. [JVT-G 050. ITU-t and ISO/IEC JTC 1. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification \(ITU-T Rec. H. 264 | ISO/IEC 14496-10 AVC\)](#) 2003
4. [Wiegand T, Sullivan G J, Bjntegaard G, Luthra A](#) [Overview of the H.264/AVC video coding standard](#) 2003(07)
5. [Kamaci N, Altunbasak Y](#) [Performance comparison of the emerging H.264 video coding standard with the existing standards](#) 2003
6. [Raja G, Mirza M J](#) [Performance comparison of advanced video coding H.264 standard with baseline H.263 and H.263+ standards](#) 2004
7. [周祥平, 王嵩](#) [新标准H.264的核心技术与应用前景](#)[期刊论文]-[中国有线电视](#) 2004(15)
8. [秦岭, 王煜坚, 李东新, 吴镇扬](#) [视频编码标准H.264的主要技术特点及其应用前景](#)[期刊论文]-[微计算机应用](#) 2004(4)
9. [Ostermann J, Bormans J, List P, Marpe D, Narroschke M, Pereira F, Stockhammer T, Wedi T](#) [Video coding with H.264/AVC: tools, performance, and complexity](#) 2004(01)
10. [Tu-Chih Wang, Yu-Wen Huang, Hung-Chi Fang, Liang-Gee Chen](#) [Performance analysis of hardware oriented algorithm modification in H.264](#) 2003
11. [周敬利, 金毅, 余胜生, 郑俊浩](#) [基于H.264视频编码技术的研究](#)[期刊论文]-[华中科技大学学报\(自然科学版\)](#) 2003(8)
12. [Iain E G Richardson](#) [The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression- Video Coding for Next-generation Multimedia](#)
13. [Kamaci N, Altunbasak Y](#) [Performance comparison of the emerging H.264 video coding standard with the existing standards](#) 2003
14. [Jianfeng Xu, Zhibo Chen, Yun He](#) [Efficient fast ME predictions and early-termination strategy based on H.264 statistical characters. Information](#) 2003
15. [Feng PAN, Xiao LIN, Rahardja SUSANTO, Keng Pang LIM, Zheng Guo LI, Ge Nan FENG, Da Jun WU, and Si WU](#) [Fast Mode Decision for Intra Prediction](#) 2003
16. [Zhang Yong-dong, Dai Feng, Lin Shou-xun](#) [Fast 4×4 intra-prediction mode selection for H.264](#) 2004
17. [Kim Y -H, Yoo J -W, Lee S -W, Shin J., Paik J., Jung H. -K](#) [Adaptive mode decision for H.264 encoder](#) 2004(19)
18. [K P Lim, S Wu, D J Wu, S. Rahardja, X. Lin, F. Pan, ZG Li](#) [Fast INTER Mode Selection](#) 2003
19. [Jeyun Lee, Byeungwoo Jeon](#) [Fast mode decision for H.264](#) 2004
20. [Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-Yi Chen, Shyh-Yih Ma, Chun-Fu Shen, Liang-Gee Chen](#) [Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264](#) 2003

21. [Yeping Su, Ming-Ting Sun Fast multiple reference frame motion estimation for H.264](#) 2004
22. [Peng Yin, Boyce J A new rate control scheme for H.264 video coding](#) 2004
23. [Li Z G, Pan F, Lim K P, Lin X., Rahardja S Adaptive rate control for H.264](#) 2004
24. [Yuan Zhang, Wen Gao, Huijiang Sun, Qingming Huang Yan Lu Error resilience video coding in H.264 encoder with potential distortion tracking](#) 2004
25. [Yen-Lin Tung, Hsiu-Chen Shu, Jin-Jang Leou An error detection and concealment scheme for H.264 video transmission](#) 2004
26. [Gnavi S, Grangetto M, Magli E, Olmo G Comparison of rate allocation strategies for H.264 video transmission over wireless lossy correlated networks](#) 2003
27. [Juyup Lee, Sungkun Moon, Wonyong Sung H.264 decoder optimization exploiting SIMD instructions](#) 2004
28. [张帆, 张旭东 H.264编码器的SSE-2优化实现\[期刊论文\]-中国图象图形学报A辑](#) 2003(z1)
29. [参考文献](#)
30. [钟玉琢, 王琪, 贺玉文 基于对象的多媒体数据压缩编码国际标准—MPEG-4及其校验模型](#) 2000
31. [Iain E G Richardson The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression—Video Coding for Next-generation Multimedia](#)
32. [Marpe D, Blatermann G Video compression using context-based adaptive binary Arithmetic coding](#) 2002
33. [Marpe D, Schwarz H, Blatermann G Context-Based Adaptive Binary Arithmetic Coding in JVT/H.26L](#) 2002
34. [Draft ITU-T Recommendation H.261. ITU-T, Video codec for audio/visual services at PX64 kbits](#) 1993
35. [ISO/IEC CD 11172. Coding of moving pictures and associated audio for digital storage media at up to 1.5Mbits/sec – Part 2: Coding of moving pictures information](#) 1991
36. [Shlien S Guide to MPEG-1 audio standard](#) 1994(04)
37. [ISO/IEC 13818-2. Information technology – Generic coding of moving pictures and associated audio Part 2: Video](#) 1995
38. [Lauzon D, Vincent A, Limin Wang Performance evaluation of MPEG-2 video coding for HDTV](#) 1996(02)
39. [ITU-T Recommendation H.263. ITU-T, Video Coding for low bit rate communication](#) 2000
40. [ISO/IEC FDIS 14496-2. Information technology – Generic coding of audio-visual objects Part 2: Visual](#) 1998
41. [ISO/IEC JTC1/SC29/WG11 N3342. Overview of the MPEG-4 Standard](#) 2000
42. [Martinez J M Standards – MPEG-7 overview of MPEG-7 description tools](#) 2002(03)
43. [Shih-Fu Chang, Sikora T, Purl A Overview of the MPEG-7 standard](#) 2001(06)
44. [Bormans J, Gelissen J, Perkins A MPEG-21: The 21st century multimedia framework](#) 2003(02)
45. [Tseng B L, Ching-Yung Lin, Smith J R Using MPEG-7 and MPEG-21 for personalizing video](#) 2004(01)
46. [ITU-t and ISO/IEC JTC 1. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification \(ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC\). JVT-G050](#) 2003
47. [Wiegand T, Sullivan G J, Bjntegaard G, Luthra A Overview of the H.264/AVC video coding standard](#) 2003(07)
48. [Iain E G Richardson The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression—](#)

49. Ostermann J. Bormans J. List P. Marpe D., Narroschke M., Pereira F., Stockhammer T., Wedi T Video coding with H.264/AVC: tools, performance, and complexity 2004(01)
50. 曾昭平. 马仲华. 张文军 H.26L中变换编码和量化的原理与分析[期刊论文]-通信学报 2002(2)
51. Hong M -C. Hahn H -S A Loop/Post Filter to Suppress Blocking and Ringing Artifacts for H.26L VideoCodec 2001(01)
52. Marpe D. Schwarz H. Blattermann G Context-Based Adaptive Binary Arithmetic Coding in JVT/H.26L 2002
53. Karczewicz M. Kurceren R The SP- and SI-frames design for H.264/AVC Circuits and Systems for Video 2003(07)
54. T. Wiegand, H. Schwarz, A. Joch, F. Kossentini Rate-constrained coder control and comparison of video coding standards 2003
55. A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. Sullivan Performance comparison of video coding standards using Lagrangian coder control 2002
56. Iain E G Richardson The Robert Gordon University, Aberdeen, UK. H.264 and MPEG-4 Video Compression- Video Coding for Next-generation Multimedia
57. Feng PAN, Xiao LIN, Rahardja SUSANTO, Keng Pang LIM, Zheng Guo LI, Ge Nan FENG, Da Jun WU, and Si WU Fast Mode Decision for Intra Prediction 2003
58. Zhang Yong-dong, Dai Feng, Lin Shou-xun Fast 4×4 intra-prediction mode selection for H.264 2004
59. Thomas Wiegand, Heiko Schwarz, Anthnoy Joch Rate-Constrained Coder Control and Comparison of Video Coding Standards 2003
60. Kim Y -H, Yoo J -W, Lee S -W, Shin J., Paik J., Jung H. -K Adaptive mode decision for H.264 encoder 2004(19)
61. K P Lim, S Wu, D J Wu, S. Rahardja, X. Lin, F. Pan, ZG Li Fast INTER Mode Selection 2003
62. Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-Yi Chient, Shyh-Yih Ma, Chun-Fu Shen, Liang-Gee Chen Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264 2003
63. Jeyun Lee, Byeungwoo Jeon Fast mode decision for H.264 2004
64. 李升家 加速H.264运动向量估计模式的选择 2004
65. J Xin, M T Sun, V Hsu Diversity-based Fast Block Motion Estimation 2003
66. S Zhu, K K Ma A new diamond search algorithm for fast block-matching motion estimation[外文期刊] 2000
67. Richard Gerber The Software Optimization Cookbook 2002
68. Ville Lappalainenl, Timo D H Unified Method for Optimization of Several Video Coding Algorithms on General-Purpose Processors 2002
69. Intel 软件说明书 (Intel Software manuals) 2003
70. MSDN, SSE 技术主题 2003
71. Document JVT-D153, JVT/H.26L decoder complexity analysis Joint Video Team of ISO/IEC MPEG & ITU-T VCEC 2002

72. ITU-T Rec.H.264 ISO/IEC 14496-10 AVC. Joint Final Committee Draft(JFCD) of Joint Video Specification. JVT Doc:JVT-D157 2002

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y819950.aspx