

西安电子科技大学

硕士学位论文

基于H. 264的高效视频转码关键技术研究

姓名：刘海啸

申请学位级别：硕士

专业：通信与信息系统

指导教师：宋彬

20100101

摘要

本文基于 H.264/AVC 视频编码标准,重点研究 H.264 标准内高效码率转码和 H.263 到 H.264 标准间高效转码的关键技术。

首先,针对 H.264/AVC 标准内码率转码,通过分析 H.264/AVC 率失真优化算法和级联全解全编转码结构,本文提出了基于率失真的快速模式选择算法,通过充分利用参考帧的图像信息和解码宏块的运动矢量、编码类型和残差等信息,估计宏块失真来直接选择宏块模式,并据此提出了高效码率转码结构。实验结果表明,与级联全解全编转码相比,本文提出的算法能够大大简化编码器的运算复杂度,编码时间减少 90%且 PSNR 下降小于 0.5dB,对实时多媒体传输及其它数据传输有着重要意义。

其次,针对目前实时视频通信中 H.263 仍然应用较多,本文还重点研究了 H.263 到 H.264 标准间的高效转码算法。针对两种标准的转换,分别从码率转码、空间分辨率 2:1 下采样转码、时间分辨率转码和混合转码四个方面进行了研究。本文提出了 H.263 宏块模式、运动矢量、CBP、COD 等相关信息复用的策略,并在此基础上提出了宏块模式快速判决算法;对于复杂度比较高的 P8x8 模式判决,采用了提前终止判决的策略;对于分辨率和帧率降低转码还提出了相应的运动矢量合成、细化和重建算法;最后结合混合转码提出了 H.263 到 H.264 标准间的高效转码整体框架。实验结果表明,在相同转码参数配置下,与级联转码器相比该高效转码算法具有稳定的率失真性能,能够很好的保证视频质量,PSNR 下降小于 0.8dB,同时转换编码速度提高 25%~45%,极大的提高了转码器的转码能力,可用于实际视频转码系统中。

关键词: 视频转码 H.263 H.264 率失真 模式判决

Abstract

This thesis is based on H.264/AVC standard, mainly focusing on efficient homogeneous bit-rate transcoding for H.264 and heterogeneous transcoding from H.263 to H.264.

First, for homogenous H.264 bit-rate transcoding, the direct mode selection algorithm is proposed based on the analysis of the RDO algorithm and the cascaded architecture. From the macroblock mode statistics and residual calculation of original bitstream as well as the reference frame, the optimal macroblock mode can be decided by using the estimated distortion. The experimental results show that the proposed transcoding method provides acceptable performance, decreasing less than 0.5 dB and saving 90% encoding time, compared with the cascaded transcoding.

Second, for H.263 to H.264 transcoding, the bit rate reduction, 2:1 ratio resolution reduction, frame rate reduction and combined transcoding are respectively focused on. The fast decision of inter-frame macroblock is presented on the basis of reusing H.263 macroblock information such as mode, motion vectors, CBP and COD. For the resolution and frame rate reduction, different methods for motion vector reconstruction and refinement are also proposed. Finally, the overall framework of transcoding from H.263 to H.264 is presented. The experimental results show that compared with the cascaded architecture the proposed algorithm can promote the speed of transcoding by 25%~45%, as well as reduce PSNR less than 0.8 dB.

Keyword: Video transcoding H.263 H.264 RDO Mode decision

西安电子科技大学

学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：刘海峰

日期 2010.3.15

西安电子科技大学

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名单位为西安电子科技大学。

（保密的论文在解密后遵守此规定）

本学位论文属于保密，在____年解密后适用本授权书。

本人签名：刘海峰

日期 2010.3.15

导师签名：宋林

日期 2010.3.15

第一章 绪论

1.1 研究背景

随着图像压缩编码技术、宽带数字通信技术和多媒体网络技术的发展,多媒体数据通信已经在广播电视、计算机和网络通信领域获得了广泛的应用。这些应用得以实现的关键就是数字视频压缩标准的制定。自 ITU-T(International Telecommunication Union-Telecommunication)的视频编码专家组于 1990 年提出视频压缩标准 H. 261 以来,视频压缩编码标准体系得到了迅猛发展,其中包括各种 MPEG^{[1][2][3]}和 H.26x^{[4][5][6][7][8]}等标准的成功制定。这些标准除语法格式不同外,在压缩效率、码率、压缩格式上都有所不同,它们分别针对特定的应用场合和不同的网络环境。例如,H.263 用于可视电话和视频会议等场合;MPEG-2 用于数字视频广播、高清晰度数字电视 HDTV 和 DVD 等场合;MPEG-4 用于 Internet 的流式视频应用。

2003 年,ITU-T 的 VCEG 和 ISO/IEC 的 MPEG 组织联合制定了最新的视频压缩编码标准 H.264/AVC^[9]。该编码标准在以往基于 MC+DCT(运动补偿+离散余弦变换)模式的各种视频编码标准的基础上,采用了许多新技术,具有能够降低 50%或更多带宽的能力,能以少于 1Mbps 的码率传输基于互联网协议(IP)的广播质量级的视频内容。这是目前版本的 MPEGx 等编解码方案根本所不能比拟的^[10]。可以预见,基于 H.264 标准的多媒体应用仍将成为未来一段时间内的研究热点。

另一方面,不同用户终端(可以是视频工作站、家用电脑、移动电脑或是手机等通信设备),通过不同的网络(如:局域网(LAN)、无线局域网(Wireless LAN)、综合业务数字网(ISDN)或是电话线拨号上网等),透明地请求和接收各种各样的视频服务(包括视频广播、视频会议、视频点播、交互视频、视频监控等),从而实现通用多媒体访问(Universal Multimedia Access)的功能。不同终端的处理能力、存储能力和显示能力各不相同,终端的视频解码器类型也各有差异。这些异构网络除带宽不同外,在误码比特率和数据包丢失率等方面也不同。另外,由于各种编码标准的存在,使得大量的视频数据按照不同的标准和格式压缩编码,分布存储在不同的设备上。

国外对转换编码器的研究最早是针对信道带宽的要求实现视频码率的变换。此后,对转换编码器的结构和码率控制算法的研究广泛展开,其中许多方法虽然具有好的效果,但考虑到实时性以及复杂程度的要求,这些算法无法实用化。随着多媒体应用的发展,仅降低比特率的转码已不能满足需要,其后陆续有对图像

分辨率转码、帧率转码和不同标准间转码的研究。转码中的关键问题成为研究的热点,包括不同标准间对应数据的映射;新的运动矢量映射问题;新的宏块模式决策;像素域转换时纹理数据的处理;以及相关的压缩域图像/视频处理算法和编码器性能优化等研究。

综上所述,随着多媒体应用的发展,越来越多的场合要求视频数据能在不同格式、不同标准间进行交换。为了使得异构网络之间、不同种类的接入设备之间能够无缝连接,需要研究压缩视频流之间的转码。不同时期、不同用途的多媒体终端使用多种不同的视频编码标准,要实现它们之间的互连互通,也必须研究视频编码转换算法。

1.2 H.264 视频转码概述

视频转码本质上还是研究视频编码,解码技术。但它与编码和解码过程又有所不同,转码的输入和输出都是压缩编码后的码流,它们在视频流格式、比特率、帧率、空间分辨率以及所遵循的压缩编码标准等方面都有可能不同,而编码和解码的输入和输出只有一端为压缩视频流。尽管如此,视频编解码中的许多优化技术仍可应用于转码过程中,提高转码速度。

转码过程中可以获得许多输入码流信息,如视频序列头信息、宏块编码模式信息、运动矢量信息、频域的压缩系数等。因此在设计转码器时,可以利用原始码流中的信息来简化转码结构,降低转码运算复杂度。

转码算法的优劣主要表现在两个方面:(1)转码性能,即转码过程中引入的图像失真度;(2)算法的运算复杂度。实际应用中转码器一方面要求运算复杂度低,另一方面实现成本足够低,所以处理运算时要尽量避免复杂的运算瓶颈,同时要保持视频质量、降低硬件要求和延迟。

H.264/AVC 视频编码标准是 ITU 和 ISO 联合研究制定的编码效率高、网络适应性强的最新视频编码国际标准,由于它可交互式和非交互式应用,适合各种媒体的传输和存储,它必将会成为新一代的视频编码标准。因此基于 H.264/AVC 视频转码技术的研究具有现实上的必要性,但同样也有技术上的困难性。例如,H.264/AVC 编码标准的高复杂度 RDO 模式判决需要计算所有可选模式的率失真代价,所以在 H.264/AVC 的编码转换中,宏块模式的重新选择往往耗费大量时间。如果能够充分利用转码过程中得到各种信息实现 H.264/AVC 的快速编码,将对转码的速度有很大提高,对实时转码中有着不可估量的意义。

另一方面,目前在实时视频通信中 H.263 仍然应用较多,而 H.264 取代 H.263 是不可逆转的趋势,在一定时期内二者必将同时存在,因此研究 H.263 到 H.264 的标准间转码算法有着重要的实际意义。虽然两种标准在技术细节上有一定的差

异,但由于二者都是基于运动补偿和块变换的视频编码标准,因此在转码过程中,如何能够尽可能的利用各种 H.263 的解码信息实现 H.264 快速编码成为标准间高效转码的关键。

1.3 主要内容和章节结构

本文工作来源于华为合作项目《H.263/MPEG-4/H.264 转码技术开发》,作为该项目中的主要部分,本文重点研究基于 H.264/AVC 标准的高效转码关键技术,对 H.264 标准内码率转码和 H.263 到 H.264 标准间转码所面临的问题展开分析和研究。文章组织结构如下:

第一章 主要介绍本文的研究背景和研究意义,概括地描述了基于 H.264/AVC 视频转换编码急需解决的问题和面临的技术难点,为本文的后续研究指明了方向。接着概述了本文的章节结构和主要内容。

第二章 首先介绍了视频压缩基本原理和国际上几种主要的视频压缩标准,尤其是对 H.264/AVC 主要的技术特点进行了详细的分析,接下来回顾了国内外转码技术研究的现状,深入剖析了视频转码中的关键技术,最后对基于 H.264 转码涉及到的问题进行了简单分析,为本文后续基于 H.264/AVC 视频转码技术的研究提供了必要的知识准备。

第三章 主要研究 H.264 标准内的高效码率转码。首先对 H.264 标准的率失真优化算法进行了回顾总结,然后在级联转码器的构架基础上,提出了基于率失真的快速模式选择算法,通过充分利用输入码流解码图像和参考帧的图像信息,估计宏块的失真,进而直接选择宏块模式,降低了编码运算复杂度,提高转码速度。

第四章 重点研究 H.263 到 H.264 标准间高效转码。分别从码率转码、空间分辨率 2:1 下采样转码、帧率转码和混合转码四个方面进行研究,本章提出了 H.263 解码宏块信息的复用策略,并在此基础上提出了快速模式判决算法及相关的运动矢量重建方法,最后结合混合转码提出了 H.263 到 H.264 标准间高效转码的整体框架。本章算法在保证质量的前提下大大降低转码计算量,具有实际应用价值。

第五章 总结了本文所做的工作,给出了相关结论,并对将来的研究工作展开展望。

第二章 视频转码技术

2.1 引言

视频转换编码是一种处理过程，它对输入的压缩码流进行端到端的处理，使得转码后的压缩码流更能适应传输信道带宽及终端的需要。它包括对视频码流压缩标准、比特率、空间分辨率、帧率的转换以及在视频码流中嵌入图标、水印信息、差错控制数据等操作^{[1][12]}，如图 2.1 所示。

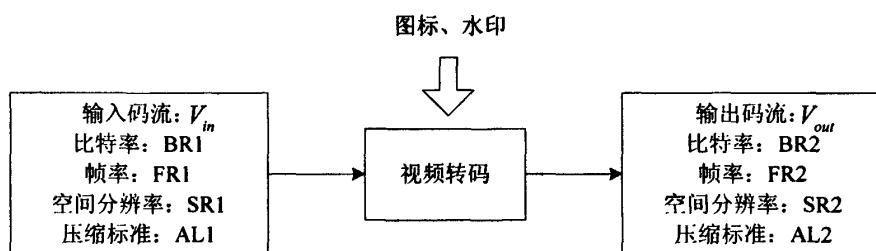


图 2.1 视频转换编码

视频转码分为同类视频转码和异类视频转码。同类视频转码是指在不改变压缩标准的前提下，对比特率，空间分辨率和帧率等进行转换。异类视频转换编码是在不同压缩标准之间进行的，是为解决跨越不同网络平台、采用不同视频编码标准引起的不兼容问题而提出的解决方案。异类视频转换编码是在充分分析不同标准之间异同的基础上，采用类似于同类视频转换编码的结构和算法来设计视频转码结构的，但是由于涉及不同的编码标准转换，所以要考虑标准间的语法转换问题，并且要考虑不同标准之间的兼容性。

图 2.2 所示为转码所能实现的功能及其分类。

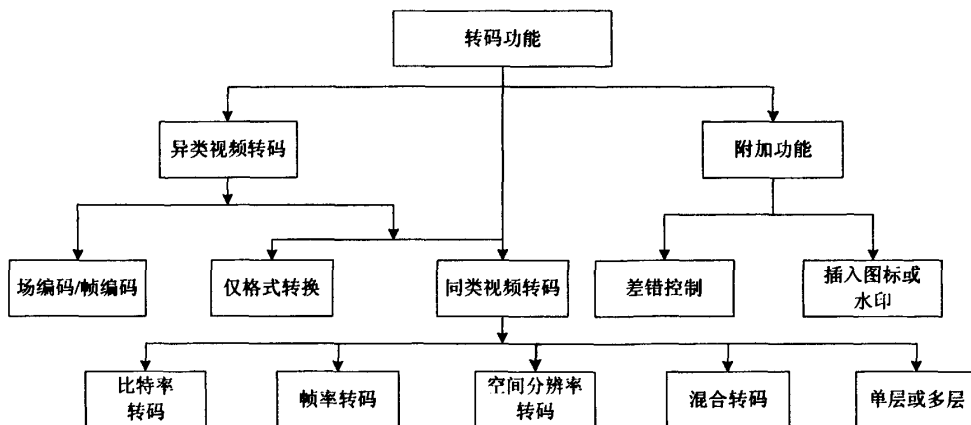


图 2.2 各种转码功能和转码分类

视频转码要考虑消除或者最大程度的减小误差的影响,提高转码后码流的解码恢复质量;还要考虑对转码结构进行简化,对转码模块进行优化,以降低转码运算复杂度,提高转码应用的实时性。

为了研究和设计转码器,必须要对转码器两端的编码标准(可能是相同或不同标准)都有深入了解,下面我们先介绍一下视频编码技术及压缩标准。

2.2 视频编码技术及标准

各种视频压缩标准在编码时都利用了图像中的两种特性:空间相关性和时间相关性。一帧图像内的任何一个场景都是由若干像素点构成的,因此一个像素通常与它周围的某些像素在亮度和色度上存在一定的关系,这种关系叫做空间相关性。另外,视频序列通常由若干帧连续的图像构成,前后帧连续图像间也存在一定的关系,即时间相关性。这两种相关性使得图像中存在大量的冗余信息。如果我们将这些冗余信息去除,只保留少量非相关信息进行传输,就可以大大节省传输频带。一个好的编码压缩方案就是能够最大限度地去掉图像中的冗余信息。

目前针对各种不同应用领域,诸多国际压缩编码标准不断产生,例如国际电信联盟标准化部门(ITU-T)制定的 H.261, H.262, H.263, H.264 和 ISO/IEC 的活动图像专家组(MPEG)制定的 MPEG-1, MPEG-2, MPEG-4。它们的提出与制定,加速了编码技术的商品化和产业化,从 VCD 到 DVD,从会议电视到可视电话,从数字电视到机顶盒、数字广播和高清晰度电视,这些日新月异、层出不穷的高科技产品将不断激励视频压缩与通信领域继续发展前行。下面介绍几种主要的国际视频压缩编码标准。

2.2.1 MPEG-4

MPEG 组织于 1999 年 2 月正式公布了 MPEG-4(ISO/IEC14496)标准第一版本。同年年底 MPEG-4 第二版亦告完成,且于 2000 年年初正式成为国际标准。MPEG-4 与 MPEG-1 和 MPEG-2 有很大的不同,MPEG-4 不只是具体压缩算法,它是针对数字电视、交互式绘图应用(影音合成内容)、交互式媒体(WWW)等整合与压缩技术的需求而制定的国际标准。MPEG-4 标准将众多的多媒体应用集成于一个完整的框架内,旨在为多媒体通信及应用环境提供标准的算法及工具,从而建立起一种能被多媒体传输、存储、检索等应用领域普遍采用的统一数据格式。

MPEG-4 标准同以往标准的最显著差别在于它是采用基于对象的编码理念,即在编码时将一幅景物分成若干在时间和空间上相互联系的视频音频对象,分别编码后,在经过复用传输到接收端,然后再对不同的对象分别解码,从而组合成

所需要的视频和音频。这样既方便我们对不同的对象采用不同的编码方法和表示方法，又有利于不同数据类型间的融合，并且这样也可以方便的实现对于各种对象的操作及编辑。

2.2.2 H.263

国际电信联盟的标准化部门 ITU-T 主要制定的是与通信相关的码率视频压缩标准，如 H.26X 系列。近几年来，新出现和即将出现的视频业务，使低码率视频通信得到广泛的应用。会议电视、会议电话发展很快，而公用电话网和无线网络上的传输速率仍然很有限而且误码率高，H.263 等标准能很好的解决这类应用问题。

H.263 是最早用于低码率视频编码的 ITU-T 标准，随后出现的第二版(H.263+)及 H.263++增加了许多选项，使其具有更广泛的适用性。H.263 是在 H.261 基础上发展起来的，其标准输入图像格式可以是 S-QCIF、QCIF、CIF、4CIF 或者 16CIF 的彩色 4:2:0 亚取样图像。H.263 与 H.261 相比采用了半像素的运动补偿，并增加了 4 种有效的压缩编码模式。无限制的运动矢量模式允许 H.263 的运动矢量指向图像以外的区域。基于句法的算术编码模式使用算术编码代替霍夫曼编码，可在信噪比和重建图像质量相同的情况下降低码率。它的预测模式允许一个宏块中 4 个 8x8 亮度块各对应一个运动矢量，从而提高预测精度；两个色度块的运动矢量则取这 4 个亮度块运动矢量的平均值。补偿时，使用重叠块运动补偿，8x8 亮度块的每个像素的补偿值由 3 个预测值加权平均得到。使用该模式可以产生显著的编码增益，特别是采用重叠的块运动补偿，会减小块效应，提高主观质量。

2.2.3 H.264/AVC 视频压缩标准

H.264/AVC 是由 ITU-T 和 ISO/IEC 的联合开发组共同开发的最新国际视频编码标准。在 ISO/IE 中该标准命名为 AVC(Advanced Video Coding)，作为 MPEG-4 标准的第 10 部分，在 ITU-T 中正式命名为 H.264/AVC 标准。

相对于早期的视频压缩标准，H.264/AVC 在编码框架上还是沿用以往的 MC-DCT 结构，即运动补偿加变换编码的混合结构，因此它保留了一些先前标准的特点。H.264/AVC 还引入了很多先进的技术，包括 4x4 整数变换、空域内的帧内预测、1/4 像素精度的运动估计、多参考帧与多种尺寸块的帧间预测技术等。新技术带来了较高的压缩比，在相同的重建图像质量下，H.264/AVC 比 H.263+和 MPEG-4 简单档次(Simple profile)减小 50%码率，但同时算法复杂度也有所增加。

下面具体介绍一下 H.264/AVC 的技术特点。

- 4x4 整数变换

H.264/AVC 将整幅图像分成 4×4 像素大小的块进行编码,其使用 4×4 的 DCT 变换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。

- 基于空域的帧内预测技术

视频编码是通过去除图像的空间与时间相关性来达到压缩的目的。空间相关性通过有效的变换来去除,如 DCT 变换、H.264/AVC 的整数变换;时间相关性则通过帧间预测来去除。这里所说的变换去除空间相关性,仅仅局限在所变换的块内,如 8×8 或者 4×4 块,并没有块与块之间的处理。H.264/AVC 是在空域中,利用当前块的相邻像素直接对每个系数做预测,更有效地去除相邻块之间的相关性,极大地提高了帧内编码的效率。

- 运动估计

H.264/AVC 的运动估计具有 3 个新的特点:1/4 像素精度的运动估计;7 种大小不同的块进行帧间预测;前向与后向多帧参考。H.264/AVC 的帧间预测中,一个宏块(16×16)可以被分为 16×8、8×16、8×8 的块,而 8×8 的块又可以分为 8×4、4×8、4×4 的块。另外与以往标准的 P 帧、B 帧不同,H.264/AVC 采用了前向与后向多个参考帧的预测。半像素精度的运动估计比整像素运动估计有效地提高了压缩比,而 1/4 像素精度的运动估计可带来更好的压缩效果。

- 帧编码和场编码自适应方式选择

对于运动量较小的图像序列,帧编码邻行之间空间相关性大,故采用帧编码模式;对于运动量大的,场编码邻行间的时间相关性大,故采用场编码模式。有时同一帧中有一部分内容运动量大,有一部分内容运动量小,则可在帧编码中进一步分成图像自适应帧场(PAFF)和宏块自适应帧场(MBAFF)等模式。

- 熵编码自适应技术

在 H.264/AVC 中采用了两种自适应的熵编码:上下文自适应变长编码(Context-adaptive variable-length coding CAVLC)和上下文自适应二进制算术编码(Context-adaptive binary arithmetic coding CABAC),它们都利用待编码视频信息邻近已编码像素的统计特性,因而码率压缩比(编码效率)进一步提高,而后者编码效率更高,但计算更加复杂些。

- 去块效应滤波器

H.264/AVC 标准引入了去块效应滤波器,对块的边界进行滤波,滤波强度与块的编码模式、运动矢量及块的系数有关。去块效应滤波器在提高压缩效率的同时,改善了图像的主观效果。

● H.264/AVC 中率失真优化技术

不同于以往的编码标准，在 H.264/AVC 标准中采用了率失真优化技术（Rate Distortion Optimization, RDO）。率失真优化策略是在率失真理论^[13]的基础上提出的一种代价函数方案，它在网络传输和编码技术中有着广泛的应用。H.264/AVC 引入了这种代价函数策略，并将其合理地运用于运动搜索、参考帧选择和模式判决中，相对于运用传统代价函数的编码方法，编码效果有了大幅度提高。其主要思想是，在计算代价函数时，同时考虑码率和失真度两方面因素的制约，在保证低失真度的同时保证低码率，这样更加有利于视频流的传输。具体的率失真优化技术在第三章中详细介绍。

2.3 视频转码技术分析

如前所述，从实现方法的角度划分视频转码可分为码率转码、分辨率转码和帧率转码，下面分别对这三类转码技术进行分析。

2.3.1 码率转码

最早的视频转码研究主要集中在降低输入视频码流码率以满足网络的传输带宽或者使其与特定的编码类/级相兼容。全解全编结构是实现码率降低的最直观最简单的转码框架，如图 2.3 所示。但是，由于重新编码过程需要复杂度很高的运动估计和模式选择过程，这种直接的架构很难达到实时的视频转码要求，这就极大的限制了码率转码器的使用。因为在很多情况下，这种类型的视频转码器都配置在服务器端(Server)或者代理端(Proxy)，它们需要根据实际需求实时的完成码流的转换。因此，码率降低的视频转码研究，关键在于在保持一定图像质量的前提下，有效的利用原始码流中的编码信息来降低重新编码的复杂度。

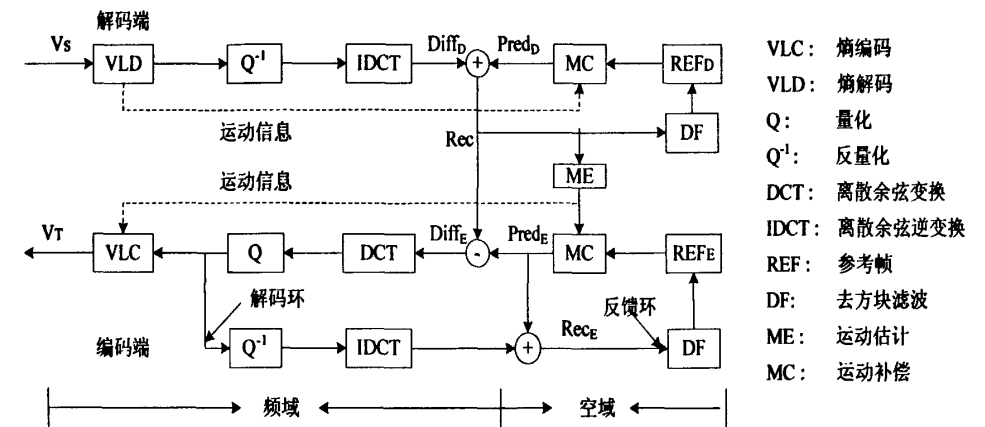


图 2.3 全解全编级联转码器

对码率转码的研究主要集中在有效转码结构的探讨上, 这些结构主要分为两大类: 开环转码结构和闭环转码结构。

在开环结构的视频转码器中, 通过可变长解码和反量化后, 直接按照新的码率进行重新量化和编码, 因此其运算复杂度低, 转码速度快, 而且转码结构简单, 容易实现。但是在转码过程中由于头端编码器和末端解码器重构帧的失配会引起累加误差, 即漂移误差^[14](drift error)。若不消除会导致解码恢复图像质量在一个 GOP 间隔内逐步恶化, 转码后图像有很大的损失。为了克服这种漂移误差带来的较大图像失真, 又引入了闭环结构的视频转码器, 闭环结构转码器在编码部分引入了闭合回路补偿差错来消除漂移误差的影响。

闭环转码结构又可以分为像素域的级联转码结构和频域转码结构。像素域的级联转码结构, 与全解全编的级联转码结构相比部分编码不做运动估计, 而是将解码出的运动矢量直接映射到编码器中, 但是它需要将原始数据解码到像素域对漂移误差进行补偿。

为进一步简化计算, 在假设 DCT、IDCT 和 MC 是线性运算的基础上, 通过对像素域级联转码结构的分析, 我们可以得到一个功能相同但结构更加简化的频域的转码结构^{[15][16][17][18]}。在解码端只是进行了变长解码和反量化得到宏块 DCT 值, 而在编码端通过运动补偿残差, 然后重新量化和变长编码得到目标码流。编码端缓存里存放反量化后的 DCT 值, 此值被反馈到频域运动补偿模块从而对漂移误差进行补偿。

有关频域运动补偿的信息可以在文献^[19]中得到。针对频域的运动补偿操作的计算量简化, 目前已提出一些方法^[20]。例如, 利用 DCT 块的能量集中在低频部分的特点进行简化、矩阵分解或者将二维的 MC 视为 2 个一维的操作。

2.3.2 分辨率转码

视频空间分辨率转码是指通过改变输入码流中图像的空间分辨率, 将图像从一种分辨率缩小为另一种分辨率以满足视频终端显示和网络带宽要求的过程, 它不仅能满足相应设备显示分辨率的要求, 也大大地降低了码率, 节省了传输带宽。空间分辨率转码主要有三种结构: 开环转码器, 像素域级联转码器和像素域高效转码器。开环转码结构如前文所述, 实现结构简单, 直观, 但是由于会引入漂移误差, 导致解码恢复图像质量显著的下降, 因此在空间分辨率转码中, 很少使用开环转码结构。

像素域级联转码器是最简单的空间分辨率转码实现方式是将输入码流完全解码到像素域, 对解码数据进行下采样, 然后再重新进行运动估计、DCT 变换、熵编码等编码操作, 最终得到目标码流。这种全解全编的像素域级联转码结构, 其

解码部分、图像缩放模块、再编码部分相互独立，转码结构灵活，转码后解码恢复图像质量高，但是由于它要重新进行运动估计、宏块编码模式判决等编码操作，运算量大，不能适应实时转码要求，因而它仅适用于对转码质量要求高而实时性要求不太高的场合。

像素域高效转码器是在全解全编级联转码器的基础上，通过复用输入码流中的图像头信息、运动矢量、宏块编码类型、预测残差等信息，简化编码端模式判决、运动估计等操作以降低转码运算复杂度，同时在转码框架中保留了反馈环，对漂移误差进行补偿，保证了转码视频质量，其转码结构如图 2.4 所示。

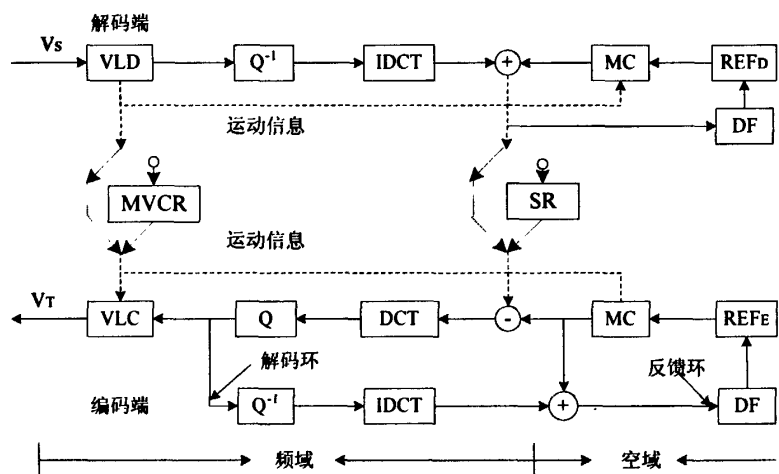


图 2.4 像素域高效空间分辨率转码器

图中 SR(spatial resolution reduction)指视频转码中的空间分辨率降低模块；MVCR(MV composition and refinement)指转码操作时因进行空间分辨率降低而对 MV 的合成和细化处理，其具体算法在后续章节介绍。

高效转码器与像素域级联转码器结构类似，都在编码部分使用了反馈环补偿差错消除漂移误差。与像素域级联转码器相比，高效转码器充分利用输入码流信息，省略了复杂的运动估计过程，降低了转码器的运算复杂度，提高了转码应用的实时性，是一种较优的转码方案。

2.3.3 帧率转码

通常用于电视广播的视频数据帧率为每秒 25 帧或者 30 帧，而在实际应用中因为一些无线终端设备的处理能力限制只能处理帧率为每秒 10 帧或者 15 帧的视频数据，这样降低视频帧率成了迫切的需求。

如果把视频信号看成时间域与空间域三维的采样信号，对视频帧率处理则看成视频在时间域上的转码。因而在一些文献中，又把帧率降低转码称为时间

分辨率降低转码。一般来说, 视频图像在时间域上的变化很小, 邻近的几帧中只有很小的变化, 以一定的速度丢弃几帧图像, 对人的视觉没有太大影响, 不会造成视频质量大幅度下降, 但是却可以节约了大量比特率。

对于视频帧率下采样, 当视频中有足够的非参考帧时(B 帧), 丢弃非参考帧来降低帧率, 不需要对码流进行太多处理, 但是如果涉及到丢弃码流中的参考帧, 则面临着参考帧丢弃后受影响的运动矢量重构问题

帧率转码结构与空间分辨率转码结构相似, 其高效转码结构仍然是通过对输入码流信息的复用, 简化转码中的再编码过程, 降低转码运算的复杂度, 并且维持良好视频质量。对应于空间分辨率降低模块 SR, 帧率转码结构采用帧率降低模块 TR(temporal resolution reduction), 它采用一定的丢帧策略进行。而 MVCR(MV composition and refinement)模块是转码操作时因进行帧率降低而对 MV 的合成和细化处理。在帧率降低转码中, 如何对丢失帧的运动矢量进行合成与细化以及宏块模式的快速判决成为其实现的关键。

2.4 视频转码中的关键技术

2.4.1 运动重估计技术

运动估计是视频编码中比较费时的过程, 如果在转码过程中重新计算运动矢量, 将会极大加重转码器的运算负担, 而不可能实现实时转码。因此利用输入码流中的原始运动信息, 在此基础上进行运动矢量和编码模式的优化, 是转码的一项关键技术。直接利用输入码流中运动矢量来推导新运动矢量的算法成为运动矢量重用算法。运动矢量重用避免了运动估计过程, 大大降低了转码运算的复杂性, 而且它的转码质量与重新进行完全运动估值时几乎相同。因此, 多数视频转码研究都集中于运动矢量的重用上。但是, 由于输出视频图像尺寸与输入视频图像尺寸可能不同或者转码两侧的编码标准不同, 输入码流的运动信息往往不能直接复用, 所以运动矢量和宏块编码模式需要重新进行推导。这是非常重要的操作, 将直接影响到转码视频质量和运算复杂度。

2.4.1.1 运动矢量合成

- 空间分辨率变化时运动矢量的合成

空间分辨率转码前后的宏块不再是一一对应的关系, 因而原始码流中的运动矢量不再有效, 需要重新映射新宏块的运动矢量。

在过去几年中, 运动矢量合成算法得到了深入研究, 比较常见的算法有平均

值法、加权平均法、中值法等。平均值法是取四个相邻宏块的运动矢量的平均值，当遇到四个运动矢量比较分散或者其中某一个宏块的运动矢量明显不同于其余三个时，这种求平均的方法会产生很大的误差^{[21][22]}，导致转码视频质量下降；文献^[23]提出的加权平均算法，利用预测残差宏块的活动性对相应运动矢量进行加权处理，从而获得下采样后运动矢量；DC 最大值法是选择和最大的宏块对应的运动矢量作为下变换后的运动矢量，这种方法比平均值法运算复杂，但是性能优于平均值法和加权平均法^[24]；中值法是计算每个运动矢量与其他运动矢量的距离和，选择值最小的对应的运动矢量作为下变换后宏块的运动矢量，这种方法虽然确定中值 MV 时需要一定的耗时^{[25][26]}，但是复杂度不高且合成的运动矢量比较精确，适用范围较广。

● 时间分辨率变化时运动矢量的合成

要求时间分辨率下降时，通常采用丢帧的方法，但是会造成丢弃帧无法正常解码，因而需要重建转码输出后的运动矢量，直观的做法是进行像素域运动矢量重新搜索，但是运算量比较大。转码算法一般利用码流中已有信息来推导新的运动矢量以减少运算量，以图 2.5 为例，宏块 B 是 A 的最佳匹配宏块，C 是 B 的最佳匹配宏块，因为第 $n-1$ 帧被丢弃，所以要寻找宏块 A 在第 $n-2$ 帧中的最佳匹配块，即如何利用码流中已有的第 $n-1$ 帧信息建立第 n 帧与第 $n-2$ 帧在时间域上的关系，使得第 $n-2$ 帧成为第 n 帧的参考帧。

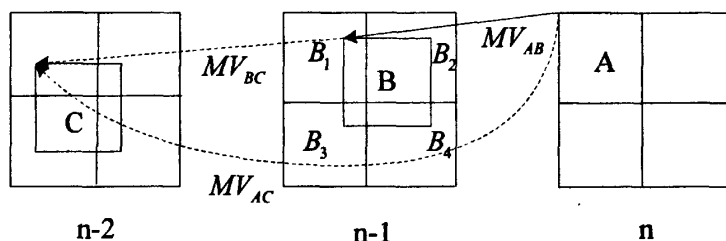


图 2.5 丢帧运动矢量重建模型

丢弃第 $n-1$ 帧后，需要在第 $n-2$ 帧中找到一个参考宏块 C，利用 B 宏块与 C 宏块之间的运动矢量来建立一个 A 宏块到 C 宏块的运动矢量。一般而言，A 宏块的参考宏块在第 $n-1$ 帧中所覆盖的四个宏块应该存在不同的运动矢量，因而可以利用这四个不同运动矢量的权重来重建一个新的运动矢量作为转码后当前帧到上一个未跳过视频帧之间的运动矢量。这种利用线性加权来重建运动矢量的方法，称为双线性插值算法^[27]。另外，J.Youn 等还提出了只采用一个主导运动矢量的算法(FDVS)^{[28][29][30]}，利用参考块所覆盖的面积大小来选择一个占主导地位的運動矢量。如图 2.5 所示，B 块覆盖了第 $n-1$ 帧中四个宏块，FDVS 算法在这四个宏块中寻找一个覆盖面积最大占主导地位的宏块即 B_2 ，其运动矢量判定为最佳运动矢量

MV_{BC} ，则转码后的重建运动向量为 $MV_{AC} = MV_{AB} + MV_{BC}$ 。

2.4.1.2 运动矢量精细化

在视频转码过程中，用各种矢量合成算法得到新的运动矢量以后，往往要进行一个精细化过程（motion vector refinement）。所谓精细化就是在运动估计得到的目标运动矢量周围一个较小的区域内进行局部搜索，将补偿效果最好的运动矢量作为最终结果，得到一个更精确的运动矢量。

运动矢量细化窗口通常选择很小，以避免视频转码中运动重新估计过程的高复杂度。不过搜索窗的大小与合成后的新运动矢量精度有关，若合成后的新运动矢量精度较高，则搜索窗可以设置的小一点，反之则要适当增加搜索窗大小，进行更精确的运动估计。通常在高效转码过程中采用小钻型搜索方式：分别计算当前搜索起点与其邻近的上下左右四个整像素点对应的代价值，选取其中代价最小的点作为整像素最优点，然后可以用相同的方式寻找最优的半像素和 1/4 像素点。这种搜索方法的搜索点较少，在保证视频质量的同时，引入的复杂度比较小。

在运动矢量搜索过程中，有三种常用的块匹配准则用来确定最佳的运动矢量：绝对误差和准则（SAD, Sum of Absolute Difference），均方误差准则（MSE, Mean of the Square Error）和平均绝对误差准则（MAE, Mean of Absolute Difference）。

$$SAD(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)| \quad (2-1)$$

$$MSE(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|^2 \quad (2-2)$$

$$MAE(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)| \quad (2-3)$$

其中 (i, j) 为待求的运动矢量， $f_k(m, n)$ 和 $f_{k-1}(m+i, n+j)$ 分别为当前帧在 (m, n) 处和参考帧在 $(m+i, n+j)$ 处的像素值，M 和 N 为宏块或子块的宽和高。

2.4.2 宏块模式决策

视频编码从一种格式向另一种格式转换时，往往需要重新确定宏块模式。比如空间分辨率下降情况下，图像尺寸减小，纹理有所变化，使用小的宏块尺寸可能更适合于宏块编码；再如，不同的编码标准在运动估计时提供的宏块模式种类可能不同，在 MPEG-2 中只有 16x16，在 H.263 中允许 8x8 和 16x16 两种模式，而在 H.264 中，则有多达 16x16、16x8、8x16、8x8、8x4、4x8 和 4x4 等 7 中不同

尺寸的宏块模式。解决这种宏块模式不匹配的问题,就需要将输入端的宏块模式根据每个宏块自身特点映射到输出端的宏块模式。另外,对于 H.264,其高复杂度率失真优化算法要求在模式判决时计算每一种可选模式的率失真代价,通过这种极其复杂的计算代价换取编码的高效率,无疑给实时转码带来了很大的困难,对此本文将在第三章中进行详细分析。

2.5 基于 H.264/AVC 的视频转码

作为最新的一种视频压缩标准, H.264/AVC 已经成为当前视频编码研究领域的热点,其良好的应用前景使得它必将成为一个主流的应用标准。一方面,随着 H.264 标准的大力推广和普及,大量基于 H.264 编码的视频流出现,而 H.264 编码标准内转码技术,包括 H.264 标准内的码率转码和分辨率转码等目前还存在大量技术问题需解决。另一方面由于目前实时视频通信应用中仍有很多设备采用 H.263 标准,完全过渡到 H.264 还需要很长一段时间。因此在现阶段,无论从保存现有的设备投资,还是从节省存储空间和网络传输带宽的角度出发,研究开发出高效的 H.263 到 H.264 转码器也是当前的一个当务之急。

H.264 由于引入了许多新的技术使其与以往的视频压缩标准 H.263、MPEG-4 等相比具有更高的编码效率,如采用多模式的帧间预测、采用多参考帧的预测技术、运动搜索精度达到 1/4 像素以及采用帧内预测等。但同时这些新技术也使得实现复杂度成倍增加,使得基于 H.264 标准的转码器要比基于以前编码标准的同类型转码器要复杂得多。H.264 在频域转码中由于使用帧间预测导致了漂移误差进而使得视频质量逐步恶化;半像素和 1/4 像素插值中的取整(rounding)和限幅(clipping)操作使得运动补偿是一种非线性操作;编解码器中的去方块滤波也是一种非线性操作,这些非线性技术的引入导致级联的全解全编转码器不可能实现数学上的完全等价,一切基于线性操作假设推导出来的视频转码简化结构不再适合于 H.264/AVC 转码,如频域的转码结构。

本文重点研究基于 H.264/AVC 标准的转码关键技术,针对 H.264/AVC 中包含众多的宏块编码模式,提出了基于率失真的快速模式判决算法,通过充分利用参考帧的图像信息和解码宏块的运动矢量、编码类型和残差等信息,估计宏块失真来直接选择编码模式,简化编码器复杂度,并据此提出了 H.264 标准内高效码率转码结构;针对 H.263 和 H.264 两种标准将长期共存的状况,本文还提出了 H.263 到 H.264 标准间的高效转码算法,根据级联全解全编转码的性能分析,指出了实现高效转码的关键和途径;根据两种标准的差异,提出了 H.263 解码宏块等相关信息复用的策略;并提出了 H.263 到 H.264 标准间码率转码、分辨率转码和帧率转码的宏块模式快速判决算法以及运动矢量细化算法,对于其中复杂度比较高的

P8x8 模式判决, 使用了提前终止判决的策略, 最后结合混合转码提出了 H.263 到 H.264 标准间高效转码的整体框架。

2.6 本章小结

本章首先简单介绍了 MPEG-X 和 H.26X 视频压缩标准, 尤其是 H.264/AVC 的技术特点, 其次对视频转码技术概况进行了介绍, 重点侧重于转码中运动重估计和宏块模式判决等关键技术。最后对基于 H.264/AVC 转码的必要性以及所面临的困难进行了详细分析。接下来的章节, 本文将对 H.264/AVC 标准内码率转码与 H.263 到 H.264 标准间转码中宏块编码模式快速判决等关键算法进行研究与实现。

第三章 H.264 标准内码率转码

3.1 引言

视频码率转码是视频转码中最早研究的内容，其目的是在保持低复杂度和高图像质量的基础上实现压缩视频流的高码率到低码率的转换，从而适应不同信道的带宽。码率转码研究的重点集中在两个方面：一是如何减小系统的复杂度，二是如何在目标压缩码率下达到最高的图像质量。

码率转码最直接的方式是采用级联全解全编结构。参照图 3.1，它分为两个部分：先由解码器对输入视频流解码，再由一个编码器按照要求对解码后的视频进行编码压缩。通过将解码后的视频流按目标码率进行重新编码，从而能够得到最佳的图像质量。该技术方案实现时需要输入码流进行完全解码，然后对码流重新进行运动估计、编码模式判决等编码操作，运算复杂度高，不能满足实时视频转码要求，在实际视频通信系统中对运算/存储等资源的占用率很高，从而造成设备成本居高不下、实际推广应用困难。

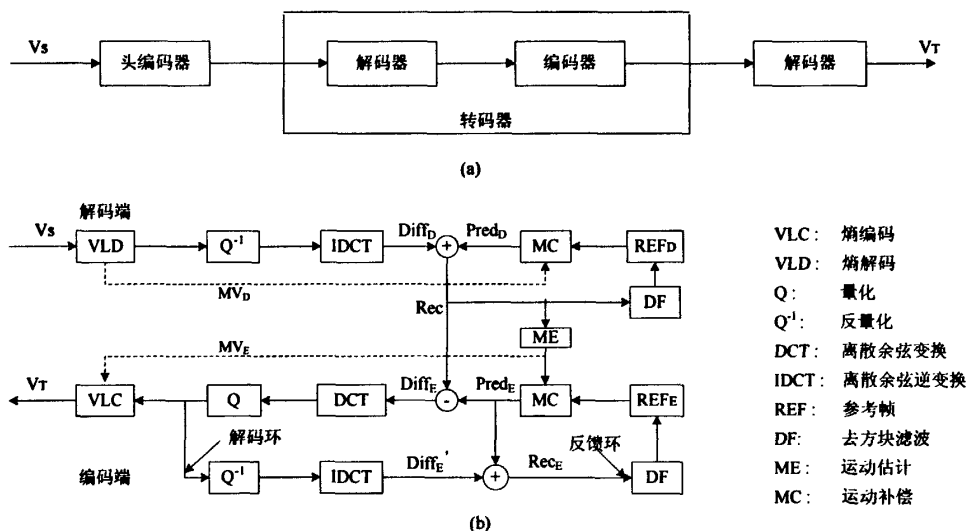


图 3.1 级联码率转码 (a) 结构框图 (b) 详细框图

本章首先介绍 H.264 标准的率失真优化 (Rate Distortion Optimization, RDO) 策略，然后提出基于 RDO 的快速模式选择算法和转码结构，利用解码信息估计宏块失真直接对编码模式进行判决，从而实现编码时间的大幅度节省。

3.2 H.264 率失真优化算法

在视频传输过程中,为了在目标数据率的限制下使得传输的视频信号的失真达到最小,需要用到信息论中的率失真问题,即在编码比特率和图像失真之间必需选择一个恰当的折衷。可描述如下:在保证比特率 R 不超过最大比特率 R_{MAX} 的条件下,使失真 D 达到最小,即 $\min\{D\}$ 限制条件:

$$R \leq R_{\max} \quad (3-1)$$

可以通过选择最优的编码参数给出"最好"的图像质量(最低的失真),并不超过目标比特率。在实际中,用一套编码参数(量化步长、块模式选择等)对视频序列进行编码,得到相应的编码比特率(R)和解码图像质量(或失真 D),两者结合,即形成一个 R - D 工作点。用不同套的编码参数重复上述编码过程,获得不同的 R - D 工作点,如图 3.2 所示。

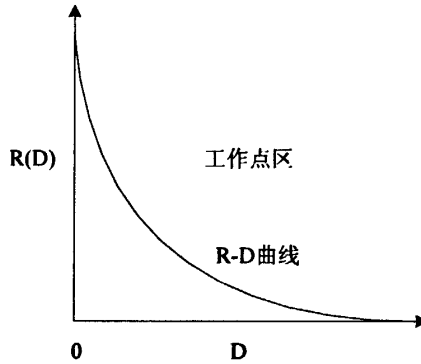


图 3.2 率失真曲线

率失真理论指出,理想的率失真曲线(R - D 曲线)位于众多的工作点的凸边上。而率失真优化的目标就是寻找一套编码参数,这套参数代表的工作点尽可能地处于或接近这条曲线。式(3-1)是很难直接最小化的,目前常用的优化模型是拉各朗日优化法,具体如下:

$$\min\{J = D + \lambda * R\} \quad (3-2)$$

J 是包含 D 、 R 以及拉各朗日乘数 λ 的代价函数。

如上所述, RDO 的主要思想是,在计算代价函数时,同时考虑码率和失真度两方面因素的制约,在保证低失真度的同时保证低码率,这样更加有利于视频流的传输。H.264 引入了这种代价函数策略,并将其合理地运用于运动搜索、参考帧选择和模式判决中,相对于运用传统代价函数的编码方法,编码效果有了大幅度提高。在 JVT (Joint Video Team) 的参考软件 JM12.0 中运动估计与模式选择算法有低复杂度模式和高复杂度模式之分^[31],并且由 RDO 选项来设定。在其高复杂度模式下, H.264 引入的代价函数策略分别如下。

- 运动矢量 RDO 代价

运动矢量代价主要运用于运动搜索模块中,为搜索点的择优提供了代价依据。H.264 直接运用了 RDO 运动搜索代价函数,其主要构成如下:

$$J(\mathbf{m}, \lambda_{\text{MOTION}}) = \text{SAD}(s, c(\mathbf{m})) + \lambda_{\text{MOTION}} \cdot R(\mathbf{m} - \mathbf{p}) \quad (3-3)$$

其中运动矢量 $\mathbf{m} = (m_x, m_y)^T$, 预测运动矢量 $\mathbf{p} = (p_x, p_y)^T$, λ_{MOTION} 为拉格朗日常数, $R(\mathbf{m} - \mathbf{p})$ 为编码运动矢量预测误差的比特, s 为原始宏块, $c(\mathbf{m})$ 为预测宏块, SAD 计算两个宏块的绝对误差和。

- 参考帧 RDO 代价

H.264 采用了多参考帧,因此定义了参考帧 RD 代价选择最优参考帧,其主要构成如下:

$$J(\text{REF} | \lambda_{\text{MOTION}}) = \text{SAD}(s, c) + \lambda_{\text{MOTION}} \cdot (R(\mathbf{m} - \mathbf{p}) + R(\text{REF})) \quad (3-4)$$

与运动矢量的代价构成类似,但在计算比特时要加上编码参考帧的相关比特数。

- 宏块模式 RDO 代价

模式代价主要运用于宏块模式判决中,在帧内预测块和帧间预测块各模式选择过程中定义有所不同。对于帧间预测宏块的模式判决,其 RD 代价为:

$$J(s, c, \lambda_{\text{MODE}}) = \text{SSD}(s, c) + \lambda_{\text{MODE}} \cdot R(s, c, \text{MODE} | \text{QP}) \quad (3-5)$$

其中 λ_{MODE} 为拉格朗日常数, SSD 为原始宏块 s 和重构宏块 c 之间 YUV 分量的均方误差和。

3.3 宏块模式快速判决算法

3.3.1 级联转码结构分析

如图 3.1 所示,级联方法是对输入压缩视频流解码后重新再编码,是两个视频网络中传统的互连操作,这种方法包含了完全编码的整个过程,需要重新进行运动估计搜索最佳运动矢量和重新判决宏块模式。简单起见这里仅考虑单参考帧而且输入是帧间模式的宏块。

1) 转码器解码端

转码器解码端的输入为已经编好的 H.264 码流,经过熵解码、重排序、反量化和反变换后得到残差块 Diff_D , Pred_D 为运动补偿得到的预测块, Rec 为解码端的重构块,因有

$$\text{Pred}_D = \text{MC}(\text{Ref}_D), \quad \text{Rec} = \text{Diff}_D + \text{Pred}_D \quad (3-6)$$

2) 转码器编码端

在级联转码结构里, 转码器编码端同头编码器类似, 首先通过运动估计得到最佳运动矢量 MV_E , 然后在模式判决中计算每个宏块每种模式的 RD 代价, 选择最佳模式 $Mode_E$, 代价函数如式 (3-5)。

Rec 为编码端的输入宏块即解码端的重构块, Rec_E 为重构块, $Pred_E$ 为运动补偿得到预测块, $Diff_E$ 为残差块, 并且有

$$Pred_E = MC(Ref_E), \quad Rec = Diff_E + Pred_E \quad (3-7)$$

3) 转码器解码环和反馈环路

转码器为了消除漂移误差, 一般采用闭环结构, 引入解码环路和反馈环路, 从图 3.1 中可以得知:

$$Rec_E = Diff'_E + Pred_E \quad (3-8)$$

式中 $Diff'_E$ 为 $Diff_E$ 经过 T、Q、Q⁻¹、T⁻¹ 后的残差块。

并且根据式 (3-6)、(3-7) 我们可以得到如下的关系式:

$$Diff_E = Rec - Pred_E = (Diff_D + Pred_D) - Pred_E = Diff_D + \Delta MC \quad (3-9)$$

其中 $\Delta MC = Pred_D - Pred_E = MC(Ref_D) - MC(Ref_E)$

由以上分析可知, 在这种级联的全解全编转码结构中, 采用高复杂度的 RDO 模式判决为了得到最佳模式需要计算每种可选模式的代价, 包含了运动搜索、变换和量化等过程, 这无疑大大增加了 H.264 编码器的复杂度。正是基于这种原因, 以往的转码器都采用复用运动矢量和宏块模式的方法来减少转码器的时间复杂度, 但同时极大的损失了图像质量。针对上述问题, 本章提出了一种宏块模式的快速判决方法, 既保证了图像质量又大幅减小了转码器的复杂性。

3.3.2 快速模式判决算法

高效转码的关键在于能否充分利用解码得到的各种信息简化编码器复杂度, 从而实现高效转码。对于 H.264 而言, 能否利用解码宏块信息直接选择宏块的模式, 无疑可以大大减少编码复杂度。文献^[32]提出了一种利用帧活动性判决宏块模式的方法, 通过解码码流中宏块模式的分布情况和运动矢量来衡量帧的活动性, 进行宏块模式的选择。该方法能较好的保证转码质量, 但是在衡量帧活动性时带来了额外的复杂度。本节在级联转码器的构架基础上, 提出了一种基于 RDO 的快速模式判决算法, 通过充分利用输入码流解码图像和参考帧的图像信息, 估计宏块的失真, 进而直接选择宏块模式, 降低编码运算的复杂度。

3.3.2.1 宏块失真估计

根据式 (3-7)、(3-8), 在编码侧进行 RDO 模式判决时, 拉格朗日代价函数

中计算编码器的输入图像和重构图像的失真为：

$$\begin{aligned}
 D = \text{SSD}(\text{Rec}, \text{Rec}_E) &= \sum_{i,j} (\text{Rec}(i,j) - \text{Rec}_E(i,j))^2 \\
 &= \sum_{i,j} (\text{Diff}_E(i,j) + \text{Pred}_E(i,j) - \text{Diff}'_E(i,j) - \text{Pred}_E(i,j))^2 \\
 &= \sum_{i,j} (\text{Diff}_E(i,j) - \text{Diff}'_E(i,j))^2
 \end{aligned} \quad (3-10)$$

这里为了分析方便, 根据 J.Moon 等人的讨论^[33], 这里我们假设当前 4x4 残差块 U , 经过 DCT 变换为 V , 然后量化、反量化结果为 V' , 最后经过反 DCT 变换为 U' :

$$U \xrightarrow{\text{DCT}} V \xrightarrow{Q} V_Q \xrightarrow{Q^{-1}} V' \xrightarrow{\text{DCT}^{-1}} U'$$

其中在量化过程中假设 Δ 为量化步长, 则有如下关系, 如图 3.3 所示。

$$\begin{aligned}
 V_Q &= \left\lfloor \frac{V+d}{\Delta} \right\rfloor * \text{sign}(V), \text{ where } d = \begin{cases} \Delta/3 & \text{for Intra} \\ \Delta/6 & \text{for Inter} \end{cases} \\
 V' &= V * \Delta
 \end{aligned} \quad (3-11)$$

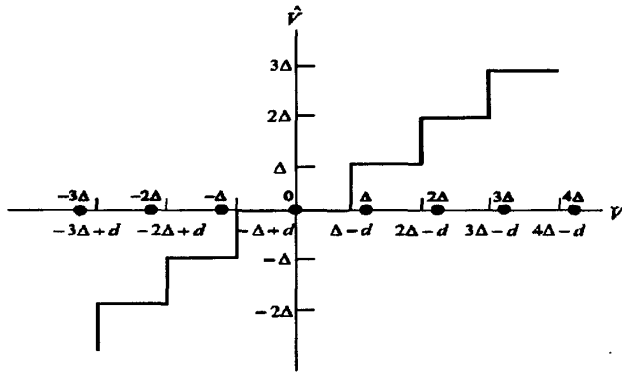


图 3.3 H.264 量化示意图

不难看出, 当变换系数 V 满足 $-\Delta+d < V < \Delta-d$ 时, 量化误差就等于 V 。

根据式 (3-10), 4x4 块的失真可表示为残差块 U 和 U' 差的内积即

$$D_{4 \times 4} = \langle U - U', U - U' \rangle \quad (3-12)$$

又因为 DCT 变换是酉变换^[1], 则有下列式成立,

$$\text{DCT}(U - U') = V - V' = \varepsilon_q \quad (3-13)$$

其中 ε_q 代表 4x4 块在 DCT 域里的量化误差, 进一步可以得出失真 $D_{4 \times 4}$ 即等于量化误差的能量和:

$$D_{4 \times 4} = \langle \varepsilon_q, \varepsilon_q \rangle = \sum_{i,j} \varepsilon_q(i,j)^2 \quad (3-14)$$

^[1]设矩阵 U , n 阶复方阵 U 的 n 个列向量是 U 空间的一个标准正交基, 则 U 是酉矩阵. 一个简单的充分必要判别准则是: 方阵 U 的共轭转置乘以 U 等于单位阵, 则 U 是酉矩阵

基于以上的分析, J.Moon 等人得出了结论:

$$\varepsilon_q(i, j)^2 \approx \begin{cases} V(i, j)^2 & |V| < \Delta - d \\ E(\varepsilon_q^2) & otherwise \end{cases} \quad (3-15)$$

这里对结论 (3-15) 式进行讨论, 我们知道 DCT 变换的系数 V , 一般满足 Laplacian distribution 分布^[34], 如图 3.4 所示:

$$f_v(V) = \frac{\lambda}{2} e^{-\lambda|V|} \quad \text{for } -\infty < V < +\infty, \quad 0 < \lambda < 1 \quad (3-16)$$

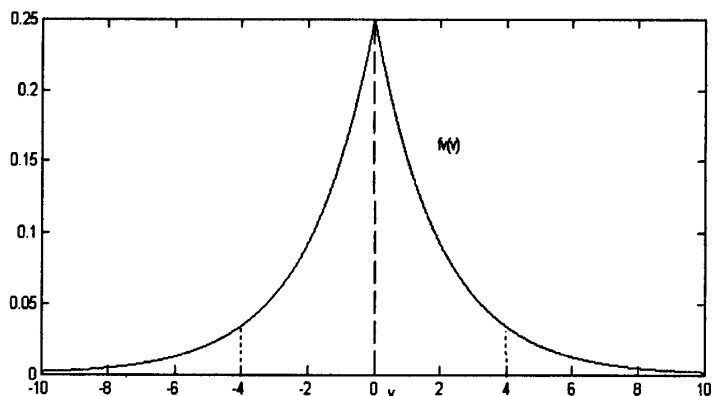


图 3.4 DCT 系数的 Laplacian 分布 ($\lambda = 5$)

所以有满足 $|V| < \Delta - d$ 的概率为

$$P = \int_{|V| < \Delta - d} f_v(V) dV = \int_{-\Delta + d}^{\Delta - d} \frac{\lambda}{2} e^{-\lambda|V|} dV = 1 - e^{-\lambda(\Delta - d)} \quad (3-17)$$

λ 为 Laplacian 系数, 且 $0 < \lambda < 1$, Δ 为量化步长与量化系数 QP 相关。 λ 取 0.5 时根据上式可得到图 3.5, 可以看出在中低码率条件下, $QP \geq 20$ 通常很容易得到满足, 此时 $|V| < \Delta - d$ 的概率近似等于 1。

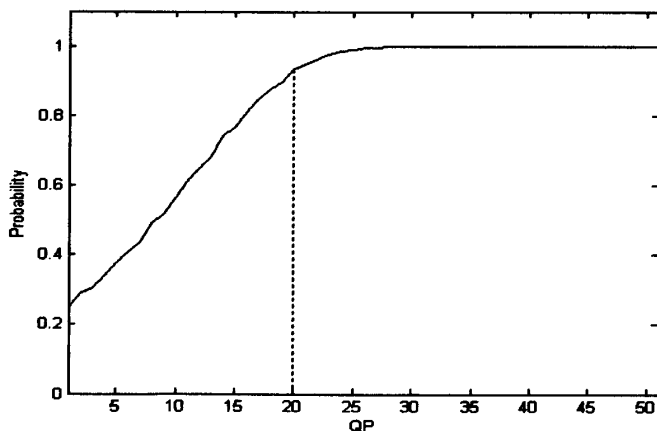


图 3.5 $|V| < \Delta - d$ 概率与 QP 关系图 ($\lambda = 5$)

因此式 (3-15) 退化为: $\varepsilon_q(i, j)^2 \approx V(i, j)^2$ 。考虑到 DCT 的酉变换性, 我们可以得知:

$$\begin{aligned} D_{4 \times 4} &= \langle U - U', U - U' \rangle = \langle V - V', V - V' \rangle \\ &= \sum_{i,j} \varepsilon_q(i, j)^2 \approx \sum_{i,j} V(i, j)^2 \\ &= \sum_{i,j} U(i, j)^2 \end{aligned} \quad (3-18)$$

基于以上分析, 我们得知在拉格朗日代价函数中, 失真可以近似为:

$$\begin{aligned} D &= SSD(Rec, Rec_E) \\ &= \sum_{i,j} (Diff_E(i, j) - Diff_E'(i, j))^2 \\ &\approx \sum_{i,j} Diff_E^2(i, j) \quad \text{from(3-12)} \\ &= \sum_{i,j} (Diff_D(i, j) + \Delta MC(i, j))^2 \quad \text{from(3-9)} \\ &= \sum_{i,j} Diff_D^2(i, j) + 2 * \sum_{i,j} (\Delta MC(i, j) * Diff_D(i, j)) + \sum_{i,j} \Delta MC^2(i, j) \end{aligned} \quad (3-19)$$

其中, $\sum Diff_D^2(i, j)$ 表示当前块解码后得到的残差系数取平方求和。在码率降低转码中, 我们可以认为输入高码率码流的质量足够好, 解码得到的运动矢量 $MV_D = (mv_{di}, mv_{dj})^T$ 是头编码器经过 RDO 的运动搜索得到的最佳运动矢量, 反映了物体的实际运动。转码器编码端进行运动重估计得到的运动矢量 $MV_E = (mv_{ei}, mv_{ej})^T$ 也一定在 MV_D 周围, 所以预测图像的差异主要取决于参考帧, 可用 MAD 或者 MSE 来衡量, 如式 (3-14)。

$$\begin{aligned} \Delta MC(i, j) &= Pred_D(i, j) - Pred_E(i, j) \\ &= MC(Ref_D(i, j))_{MV_D} - MC(Ref_E(i, j))_{MV_E} \\ &\approx Ref_D(i + mv_{di}, j + mv_{dj}) - Ref_E(i + mv_{ei}, j + mv_{ej}) \\ &\approx \alpha * MAD_{Ref} \\ \sum \Delta MC^2(i, j) &= \sum (Pred_D(i, j) - Pred_E(i, j))^2 \\ &\approx \sum (Ref_D(i + mv_{di}, j + mv_{dj}) - Ref_E(i + mv_{ei}, j + mv_{ej}))^2 \\ &\approx \alpha^2 * MSE_{Ref} \end{aligned} \quad (3-20)$$

式中 $i + mv_{di}, j + mv_{dj}, i + mv_{ei}, j + mv_{ej}$ 代表运动矢量指向的最近的整像素点, 因为 H.264 运动搜索采用 1/4 像素精度, 如果 MV 的垂直和水平分量为整数, 参考块相应的像素已经存在; 如果其中一个或两个为分数, 预测图像则要通过参考帧中相应像素内插获得, 这里 $i + mv_{di}, j + mv_{dj}, i + mv_{ei}, j + mv_{ej}$ 分别取其最近的整像素位置, 因为利用临近已编码点进行内插也是对其距离最近的整像素点取最大权重。 MAD_{Ref} 和 MSE_{Ref} 分别代表参考帧的平均绝对误差和均方误差, 这里用平均绝对误差和均方误差来估计当前参考帧相应像素的误差和相应宏块的均方误差, α 为

经验系数，分别定义为：

$$MAD_{Ref} = \frac{1}{WH} \sum_{i,j}^w,h |Ref_D(i,j) - Ref_E(i,j)|$$

$$MSE_{Ref} = \frac{1}{WH} \sum_{i,j}^w,h (Ref_D(i,j) - Ref_E(i,j))^2 \quad (3-21)$$

$$\alpha = \frac{mv_{di}^2 + mv_{dj}^2}{\overline{mv_{di}^2} + \overline{mv_{dj}^2}} \quad (3-22)$$

其中 W 、 H 为宏块的宽和高， $\overline{mv_{di}}$ 、 $\overline{mv_{dj}}$ 为当前解码帧所有宏块运动矢量两个分量的平均值。

对于经验系数 α ，一方面因为基于块匹配算法的运动矢量估计中，对于任何一个图像块，使用均方误差最小匹配准则搜索得到的运动矢量 (mv_i, mv_j) ，其运动矢量的模值与该图像块的均方误差变化趋势一致，因此可以通过当前解码宏块的运动矢量 mv_d 的模值与其平均值相比来调整 MSE；另一方面因为本章提出的宏块模式的快速判决算法是通过估计失真利用门限值直接判决，通过引入加权系数 α 可以在运动剧烈程度相对较高的区域里，引入更多 P8x8 的判决，进一步提高宏块模式预判的准确度。

根据式 (3-20)，式 (3-19) 最终可以改写为：

$$D = SSD(Rec, Rec_E) \approx \sum Diff_D^2 + 2\alpha * MAD_{Ref} * \sum Diff_D + \alpha^2 * MSE_{Ref} \quad (3-23)$$

3.3.2.2 宏块模式判决

根据式 (3-23)，我们可以很容易利用解码宏块的残差信息和参考帧信息估算出该宏块失真，如果可以通过估计宏块失真直接判决宏块模式，则可以大大节省编码时间，减小转码器的复杂度。我们首先利用 JVT 会议参考软件 JM 统计了几种标准视频序列在 30fps, 1024kbps=>512kbps 条件下编码第一个 P 帧后得到的各最佳模式的平均失真值，结果如表 3.1 所示。

表 3.1 最佳模式的平均失真

序列	P16x16	P16x8	P8x16	P8x8
Foreman	219	206	244	271
Mobile	132	212	159	249
News	110	197	146	334
Akiyo	113	77	143	193
Bridge-close	277	256	334	419
Bridge-far	219	192	317	260
Bus	180	245	167	274

Coastguard	205	256	167	274
Container	162	199	174	239
Flower	143	112	123	198
Hall	255	207	227	316
Highway	231	274	252	546

从表中可以看出, 绝大部分序列编完第一个 P 帧后, 得到的 Inter 宏块模式类型大致可以分为三类: P16x16, P8x8, P16x8 与 P8x16, 而且 P8x8 模式的平均失真最大, P16x16 的最小, 而 P16x8 与 P8x16 介于两者之间, 因此可以作为门限值有效地对模式进行区分。如果能够通过估算出失真值, 则能很好的判决出最优帧间编码模式, 而不必进行 RDO 遍历每种模式计算比较代价。

对于门限值, 采用固定某一个 P 帧用全解全编高复杂度 RDO 模式判决, 并记录该帧最佳帧间模式的平均宏块失真代价, 作为后续 n 帧的模式判决的门限值。如果发生了场景切换, 导致相邻帧之间相关性遭到破坏, 这时如果仍然采用原门限值, 将造成宏块模式的错误选择, 最直接的后果是该帧 PSNR 下降会相对比较严重, 因此我们采取加入自适应反馈的方法, 如果前者所述情况发生, 则采取常规的 RDO 模式判决, 并记录新的判决门限, 这样就可以有效的保证判决门限和当前帧宏块失真的相关性, 以及模式判决的准确性。

如上所述, 我们可以根据式 (3-23), 利用解码宏块的残差信息和参考帧信息估算该宏块的失真, 并选择最接近的门限值对应的宏块模式进行编码。另外, 由于转码采用增大 QP 实现码率降低, 其转码后的宏块模式一般会比转码前的更复杂^[35], 如图 3.6 所示。为了进一步地修正模式判决结果, 如果上述方法得到的宏块模式比转码器解码得到的宏块模式还要复杂, 则直接复用输入宏块的模式。同样, 如果宏块模式最终判为 P16x16, 则加入 Skip 的模式判决可以进一步节省后续的变换量化时间, 提高编码效率。

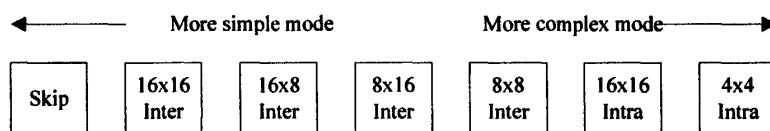


图 3.6 宏块模式复杂度示意图

对于转码器里另一个比较耗时的运动搜索, 不直接进行 MV 的重新搜索, 而是以输入的解码 MV 和预测 MV 为起点, 做小钻型细化。

3.4 高效码率转码结构

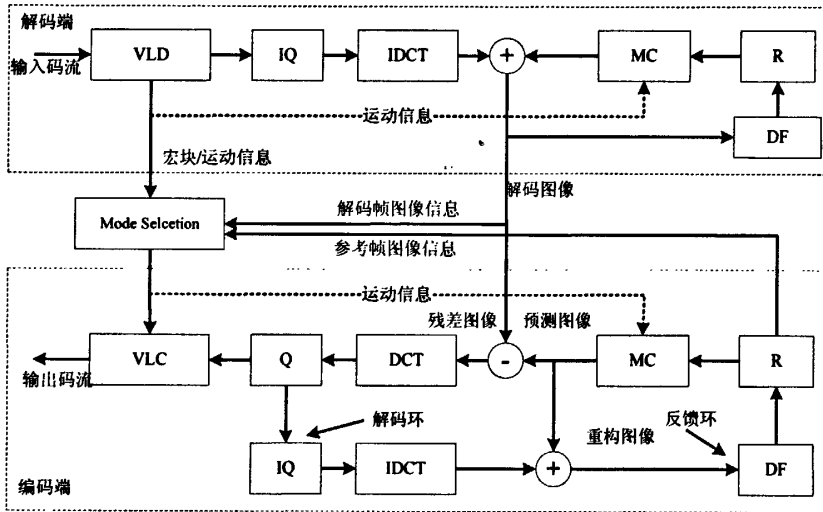


图 3.7 基于快速模式判决的 H.264 高效转码框图

综上所述，本文提出了如图 3.7 所示的 H.264 高效转码结构，其中 Mode Selection 利用解码器得到的解码帧图像数据和编码器参考帧图像数据进行快速模式判决和运动矢量细化。

基于快速模式判决的 H.264 的高效码率转码，首先解码端接收输入的压缩码流，进行解码和相关的解码图像信息的采集；然后利用相关的解码图像信息和参考帧图像信息进行宏块模式的快速判决和运动矢量细化；最后编码端进行重新编码。

其中核心步骤快速模式判决充分利用了相关解码图像信息和参考帧图像信息进行宏块失真的估计和模式的快速判决，步骤如下，参见流程图 3.8：

- (1) 判断当前编码帧是否为 I 帧，如果是则不参与快速转码；
- (2) 判断当前编码 P 帧在当前的 GOP 里是否为第一个 P 帧，如果是也不参与快速转码转到 (3)，否则对当前帧的每个宏块执行步骤 (4) 和 (5)；

(3) 对当前帧每个宏块利用 JVT 会议参考软件 JM 的高复杂 RDO 模式判决和运动搜索计算宏块模式和运动矢量，并且统计该帧最佳帧间宏块模式的失真平均值作为后续编码帧模式判决的门限 Threshold，退出模式判决；

(4) 如果当前宏块模式为帧内模式或者 SKIP 模式，则直接复用输入宏块模式；否则通过利用解码宏块的残差系数和 $\sum Diff_d$ 、平方和 $\sum Diff_d^2$ ，以及参考帧平均绝对误差 MAD_{Ref} 和均方误差 MSE_{Ref} 估计宏块失真，估计的方法为：
$$D \approx \sum Diff_d^2 + 2\alpha * MAD_{Ref} * \sum Diff_d + \alpha^2 * MSE_{Ref}$$
，其中 α 为经验系数，根据估计的失真，直接选择与其最接近的门限所对应的宏块模式，并且对判决结果根据输入解码宏块模式予以适当修正，如果判决结果模式比输入模式还要细则直接复

用输入模式;

(5) 对当前帧的每个宏块以输入的解码宏块运动矢量和预测运动矢量为运动搜索起点, 做小范围的细化搜索, 得到新的运动矢量传送至编码器;

(6) 当前帧编码后统计重构帧和解码帧的 PSNR, 若与上一帧统计 PSNR 结果相差超过一个门限, 则返回 (3), 否则退出。

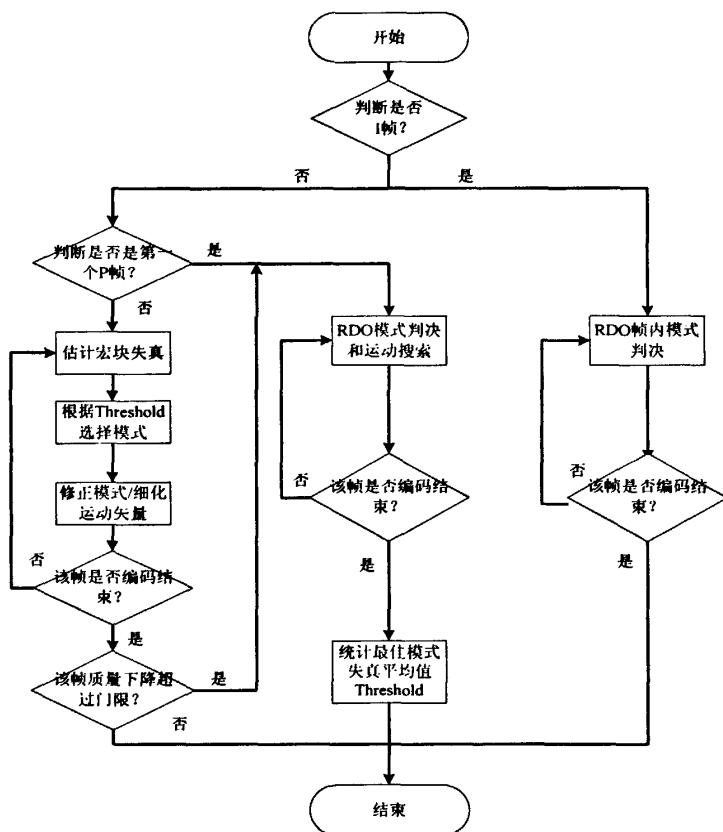


图 3.8 快速模式判决流程图

3.5 实验结果

为了衡量本章提出的高效转码算法性能, 我们做了比较实验。本章所有实验硬件环境为 CPU Intel Pentium(R)4, 3.0GHZ, 1.0G 内存; 软件测试模型为 JM12.0; GOP 结构为“IPPP...”; 参考帧数目为 1; 搜索范围为 16 像素; 搜索精度为 1/4 像素精度; RDO 采用 JVT 会议参考软件 JM 规定的高复杂度 RDO; 快速模式判决的经验系数 α 为 1。参考序列采用标准 akiyo、bridge-close、bidge-far、container、flower、hall 序列; 输入码流的源码率为 1024kbps, 分辨率为 352x288 CIF 格式 (352x288), 帧率 30fps; 输出码流的目标码率为 512kbps, 分辨率为 352x288 CIF 格式, 帧率 30fps。

首先，统计各参考序列在上述实验条件下分别采用级联全解全编方法和本章提出的基于 RDO 快速模式判决的高效转码方法，转码 10 帧得到的宏块模式判决相同的比例。统计结果见表 3.2，由表可以看出采用本章提出的快速模式判决得到的宏块模式与级联高复杂度 RDO 模式判决得到的宏块模式相同的比例可达到 60%~80%，宏块模式的正确判决有效的保证了高效转码的图像质量。

表 3.2 分别采用级联与高效转码判决宏块模式相同的比例

模式	Akiyo	bridge-close	bridge-far	container	flower	hall
Skip	82.98%	86.75%	76.94%	82.14%	94.01%	62.3%
P16x16	55.26%	67.41%	47.46%	50.31%	79.38%	48.91%
P16x8	49.17%	63.39%	54.25%	51.11%	71.12%	60.89%
P8x16	47.18%	74.39%	66.67%	53.17%	54.24%	53.25%
P8x8	73.18%	94.09%	17.09%	76.03%	89.14%	59.15%
I4x4	44.44%	100%	94.29%	0%	100%	75%
I16x16	50.00%	100%	49.43%	100%	83.33%	38.67%
total	74.19%	84.43%	66.30%	73.99%	85.55%	57.13%

其次，我们统计了各参考序列分别采用级联和高效转码方法，转码 100 帧所消耗的编码时间、转码时间及 Y、U、V 分量的 PSNR。其中级联转码结果见表 3.3，高效转码结果见表 3.4，两者性能比较结果见表 3.5，由此我们可以看出采用本章提出的基于 RDO 快速模式判决的高效转码方案在质量损失很小的情况下(Y 分量 PSNR 损失 0.3~0.5db)，编码时间节省可达 90%之多。

表 3.3 级联转码性能

参考序列	PSNR_Y(db)	PSNR_U(db)	PSNR_V(db)	编 码 时 间 (s)	转 码 时 间 (s)
akiyo	47.59	50.69	51.72	571.096	744.11
bridge-close	39	40.63	42.48	573.039	746.359
bridge-far	43.07	42.91	44.24	541.023	714.891
container	40.51	47.35	47.26	603.655	777.61
flower	29.3	37.09	41.56	552.052	725.984
hall	41.8	42.76	45.4	558.577	732.75

表 3.4 高效转码性能

参考序列	PSNR_Y(db)	PSNR_U(db)	PSNR_V(db)	编 码 时 间 (s)	转 码 时 间 (s)
akiyo	47.24	50.27	51.4	48.831	221.562
bridge-close	38.64	40.61	42.43	48.686	221.656
bridge-far	42.65	42.79	44.1	49.808	223.937
container	40.19	47.14	46.95	50.827	224.578
flower	28.95	37.03	41.38	55.69	257.015
hall	41.24	42.54	45.1	51.5	225.485

表 3.5 级联与高效转码性能比较

参考序列	ΔPSNR_Y (db)	ΔPSNR_U (db)	ΔPSNR_V (db)	编码时间 节省(%)	转码时间 节省(%)
akiyo	-0.35	-0.4	-0.32	91.45	70.2246
bridge-close	-0.36	-0	-0.05	91.504	70.3017
bridge-far	-0.42	-0.1	-0.14	90.794	68.6754
container	-0.32	-0.2	-0.31	91.58	71.1195
flower	-0.35	-0.1	-0.18	89.912	64.5977
hall	-0.56	-0.2	-0.3	90.78	69.2276

最后，为了进一步衡量本章提出的基于 RDO 快速模式判决算法性能，在上述实验条件下我们又对 flower 序列分别采用级联和文献^[32]提出的基于帧活动性的宏块模式判决方法对本章提出的方法进行比较。实验结果图 3.9 为不同码率下的 PSNR，图 3.10 为相对编码时间（实际编码时间除以最大级联编码时间然后乘以 100），图 3.11 为转码前后图像主观质量比较。结果表明，本文提出的基于 RDO 快速判决模式的高效转码算法，在 PSNR 上介于其余二者之间，主观质量无较大差异，但所用的编码时间最少，其性能优于基于帧活动性的快速模式判决算法。

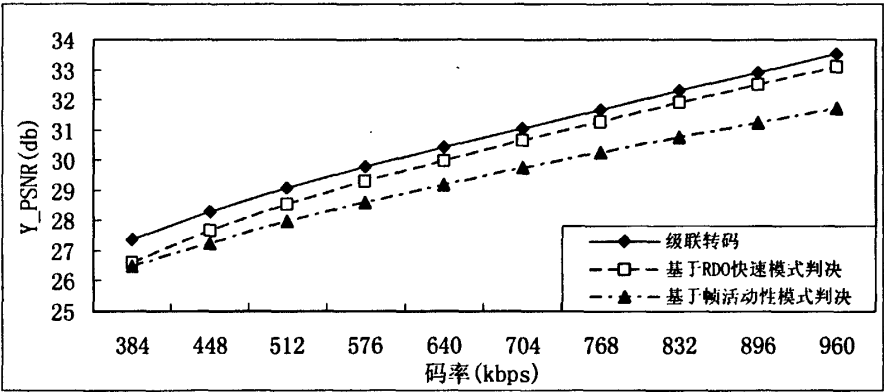


图 3.9 转码器 PSNR 性能

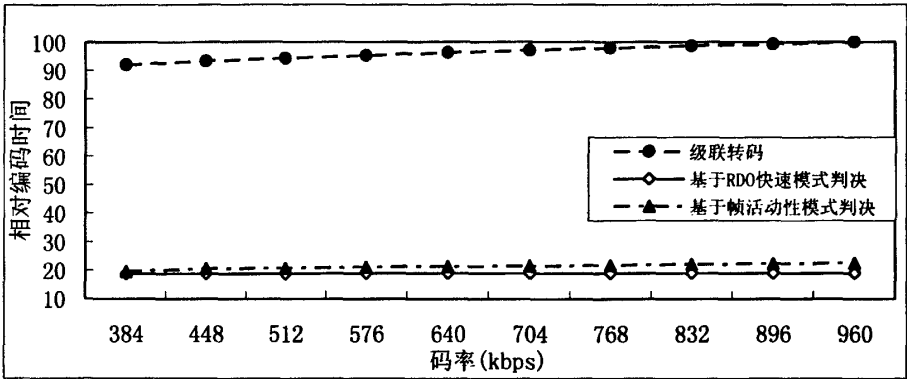


图 3.10 转码器性能相对编码时间



(a)转码前图像

(b)基于帧活动性模式判决转码后



(c)基于 RDO 快速模式判决转码后

图 3.11 转码前后主观图像质量比较(flower 序列)

3.6 本章小结

本章提出的 H.264 高效码率转码结构是在级联转码器的构架基础上, 由解码器、快速模式判决器和编码器三部分组成。模式的快速判决通过充分利用输入码流解码图像和参考帧的图像信息、运动矢量、宏块编码类型和残差等信息, 估计宏块的失真, 直接选择宏块模式, 并且进行运动矢量的细化, 简化编码器模式判决、运动估计等操作以降低转码运算的复杂度, 提高转码速度, 具有复杂度低和易于实现的优点, 可用于实时多媒体传输及其它数据传输。

第四章 H.263 到 H.264 标准间转码

4.1 引言

由 2.2 节所述可知,H.263 是面向实时视频通信的视频编码标准,它是由 ITU-T 在 H.261 的基础上完成的,结构与 H.261 相似。但由于采用了半像素运动估计及四种新的编码选项(无限制运动矢量,高级预测模式,基于语法的算术编码和 PB 帧模式),它的效率比 H.261 高得多。在相同压缩比下,PSNR 可以提高 3-4dB。后来,ITUT 的专家对 H.263 标准进行了改进,在原有的基础上加入了更多的编码选项,以提高原有的压缩效率和图像质量,这就是后来的 H.263+及 H.263++,他们与 H.263 是向前兼容的。而 H.264 与以前的任何标准相比,压缩效率都有更大的提高,在相同视频图像质量下,H.264 比 H.263 要节省 50%左右的比特率,也就是说它可以仅用一半的编码量就能得到同样清晰的图像。

目前在实时视频通信中 H.263 仍然应用较多,而 H.264 取代 H.263 是不可逆转的趋势,在一定时期内,二者必将同时存在,因此研究 H.263 到 H.264 标准间的转码算法有着重要的实际意义。

表 4.1 给出了两个标准之间关键技术的对比。

表 4.1 H. 263 和 H. 264 的关键技术指标

参数	H.263	H.264
变换类型	DCT	整数 DCT
变换块大小	8×8	4×4
帧内预测	DCT 域	像素域
宏块大小	16×16	16×16
运动矢量精度	1/2 像素	1/4 像素
1/2 像素插值方式	双线性	六抽头滤波器
1/4 像素插值方式	无	双线性
参考帧数量	1	=<16
运动补偿块大小	16×16 (8×8)	7 种块大小
环路去块效应滤波器	无	支持
X 方向运动矢量	[-16,15.5]	[-1024,1023.75]
Y 方向运动矢量	[-16,15.5]	[-64,63.75]

由表 4.1 可知, H.263 到 H.264 的转码问题,主要集中在 DCT 变换方式、MV 的重新估计、编码模式的判决等方面。虽然 H.263 与 H.264 两种标准在技术

细节上有一定的差异，但由于二者都是基于运动补偿和块变换的视频编码标准，因此在转码过程中，很多前端编码的信息都可以直接或经过修正后应用到后端的编码上，这也是我们降低转码计算量的主要思想，如何能够尽可能的利用各种解码信息则成为实现 H.263 到 H.264 标准间高效转码的关键。

4.2 级联转码耗时性能分析

对于 H.263 到 H.264 的标准间全解全编级联转码器，可以看作由 H.263 的解码器和 H.264 的编码器的简单串联而成，和 H.264 标准内级联转码结构类似。为了确定合理的转码策略及优化方案，需要对级联转码器中解码和编码部分模块的耗时进行测试和分析。下面对级联转码中 H.263 解码器和 H.264 编码器的耗时及于 H.264 编码端各个模块的耗时比例统计进行统计分析，分别测试了 Foreman、Mobile 两组码流，其中输入码流：300 帧，分辨率 4CIF，帧率 30fps，码率 2048kbps 的编码码流；转码后目标码流：300 帧，分辨率 4CIF，帧率 30fps，码率分别为 2024kbps、768kbps、64kbps。

实验结果如表 4.2、表 4.3 所示，分别给出了在给定的测试条件下，H.263 解码器与 H.264 编码器的耗时比例，以及编码端主要模块的耗时比例情况。

表 4.2 级联转码中 H. 263 解码和 H. 264 编码耗时比例

输入码流	目标码率 (kbps)	解码时间 (%)	编码时间 (%)	其它 (%)
Foreman (2048kbps)	2024	10.86	88.94	0.2
	768	12.36	87.53	0.11
	64	14.91	85.00	0.09
Mobile (2048kbps)	2024	13.29	86.61	0.1
	768	11.44	88.43	0.3
	64	13.75	86.05	0.2

表 4.3 级联转码中 H. 264 编码器各模块耗时比例统计

输入码流	目标码率 (kbps)	模式判决 (%)	变换量化 (%)	速率控制 (%)	编码缓存 (%)	熵编码 (%)	去方块滤波 (%)	半像素插值 (%)
Foreman	2024	35.65	16.33	8.8	6.88	5.12	7.54	0.24
	768	35.15	13.17	10.93	8.27	3.29	7.43	0.3
	64	30.08	7.41	14.06	10.52	2.19	6.99	0.37
Mobile	2024	31.67	17.2	9.3	7.32	5.51	7.85	0.26
	768	32.93	13.32	11.22	8.64	3.52	7.53	0.3
	64	31.15	10.86	12.64	9.63	2.69	7.45	0.34

从表 4.2 可以得出结论，转码过程的耗时比例，H.263 解码侧约占 10%，而 H.264 编码侧约占 90%，说明所使用的 H.263 解码器复杂度远低于 H.264 编码器。对于 H.264 编码器的优化将比较重要，如何减小 H.264 的编码复杂度仍是 H.263 到 H.264 标准间高效转码的关键。

从表 4.3 可以得出结论，当由 H.263 码流转换为 H.264 码流时，除了前端 H.263 解码之外，后端 H.264 编码所需的所有运动矢量和预测模式等都要重新进行判断，对于 H.264 编码器而言，这个计算开销非常之大，很难实现实时转码处理。其中编码侧宏块模式判决模块所占耗时占到了整个转码器耗时的 30%~35%，这部分是降低转码复杂度的主要模块，也是高效转码实现的主要手段之一。为了保证高效转码中的视频质量，尽可能的利用解码侧的宏块模式、运动矢量等信息是标准间高效转码的核心思想。

4.3 高效码率转码

为了达到高效转码，充分利用解码侧参考帧、解码侧当前帧和编码侧参考帧的相关信息，在满足特定条件下应直接复用解码帧输入运动矢量或者直接判决编码侧当前帧各宏块的模式，从而简化模式判决和相应的编码操作。对于特定的 H.263 到 H.264 的标准间转码，如果能够充分挖掘利用 H.263 解码码流中隐藏的信息，无疑可以带来更多的便利。

4.3.1 H.263 宏块信息的复用

H.263 按分层方式定义码流结构，共四层。自上向下分别为：图像层（picture layer）、块组层（GOB layer）、宏块层（macroblock layer）和块层（block layer）。其中宏块层的数据由宏块头和其后的块数据组成，数据结构如图 4.1 所示。

COD	MCBPC	MODB	CBPB	CBPY	DQUANT	MVD	MVD2	MVD3	MVD4	MVDB	Block Data
-----	-------	------	------	------	--------	-----	------	------	------	------	------------

图 4.1 H. 263 宏块层码流语法

(1)宏块编码标志(COD)(lbit)

当 COD 为“0”时表示该宏块被编码。如果 COD 为“1”，则表示该宏块中没有被编码的信息，该宏块的运动矢量和量化后的系数全部为 0。帧内编码图像没有 COD 字段。

(2)宏块类型及色度编码模式(MCBPC)(变长码)

MCBPC 表明宏块的编码类型。宏块有多种编码类型，如 INTER(帧间)，INTER+Q(帧间、新量化级)，INTER+4V(帧间、四个运动矢量)，INTRA(帧内)及 INTRA+Q(帧内、新量化级)。还有一种填充(Stuffing)的类型，用来增加系统的码

流。MCBPC 也同时表明色度块的编码形式,例如标明在 U 块和 V 块内除了 INTRADC 外是否还有需要传递的变换系数。

(3) B 宏块形式(MODB)(变长码)

MODB 仅当 PTYPE 指示为“PB-frame”时才出现,它指示 CBPB 或 MVDB 是否出现。

(4) B 宏块编码形式(CBPB) (6bits)

CBPB 仅出现在 PB 帧中, CBPB 共 6 位, 分别对应 B 宏块的 6 个子块, 最左边的一位对应块 1, 最右边的一位对应块 6, 若块 N 中有非零量化系数, 则 CBPBN=“1”, 反之为零。

(5)亮度块编码形式(CBPY)(变长码)

给出了四个亮度块的编码形式(在 PB 帧模式下指 P 宏块的亮度块), 当 CBPYN=“1”时, 表示块 N 中有非零的帧内交流系数被编码传送。

(6)量化信息(DQUANT) (2bits/变长码)

这个字段给出了当前宏块量化级和前一宏块量化级的变化量, 这个变化量限制在-2 和+2 之间。

(7)差分运动矢量(MVD)(变长码)

可变长码字, 由水平分量可变长码字和垂直分量可变长码字组成。所有的 INTER 宏块(包括 PB 帧模式下的 INTEA 宏块)都包含 MVD。

(8)先进预测方式下的差分运动矢量(MVD2-4)(变长码)

宏块类型为 INTER+4V 时, 每个宏块对应了四个运动矢量 (MVD,MVD2,MVD3,MVD4), 即每个亮度块对应一个运动矢量。

(9)B 宏块运动矢量(MVDB)(变长码)

在采用 PB 帧模式编码时, B 宏块在前一 P 帧作运动预测, 可由 MVDB 给出最佳匹配块的运动矢量。

因此在 H.263 向 H.264 转码的过程中, 为了减少编码器复杂度, 需要尽可能挖掘 H.263 码流中宏块的解码信息, 本文提出的方法具体如下:

● 复用宏块 CBP 信息

在 H.264/AVC 中, CBP(Coded Block Pattern) 是宏块层的码流语法元素, 用来标记一个 8×8 的亮度块或色度块是否含有非 0 的变换系数。如图 4.2 所示, CBP 由 6 位二进制数 $b_5b_4b_3b_2b_1b_0$ 组成。其中 b_5b_4 用于标记 2 个 8×8 色度块, $b_3b_2b_1b_0$ 用于标记 4 个 8×8 亮度块。

CBP 主要作用是指宏块残差(亮度与色度)经 DCT/量化后为 0 的部分, 由于实际物体的运动可看做是一种刚性运动, 而且输出 QP 与输入 QP 接近, 故可以近似认为输出宏块的残差与输入宏块的残差分布相似, 此时可以认为输入宏块中为 0 的部分经编码转换后仍然为 0, 从而省去输出宏块在该部分的 DCT、量

化、反量化，IDCT 的时间，进一步的还可省去该宏块运动矢量的细化操作，直接复用模式映射后的 MV，从而节约一定转码复杂度。

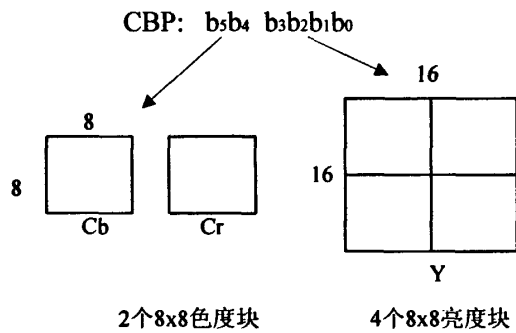


图 4.2 H.264 CBP 位构成示意图

● 复用宏块 COD 信息

如上所述，在 H.263 标准中编码的宏块指示 COD（Coded Macroblock Indication），置“0”时该比特标示宏块编码。若置“1”，对该宏块没有任何进一步的信息传输；在此情况下，译码器应把该宏块当作一个 INTER 宏块处理具有整块的运动矢量等于 0 并且无任何系数数据。对这些图像中的每个宏块而言，在不是“INTRA”模式类型的图像中，COD 才存在。在高效转码中 H.263 码流解码出来，提取 COD 值。当 COD 值为 1 时，当前宏块直接判为 P16x16，提前终止模式判决。

4.3.2 快速模式判决

H.263 到 H.264 的标准间转码时，由于输入的 H.263 宏块模式只有 Intra、Inter 两种类型(都是 16x16 块)，所以基本上而言，对于输出的 H.264 宏块模式都需要重新判决。

4.3.2.1 级联转码模式变换分析

为了确定合理的模式判决算法，我们首先对级联转码中输入宏块的模式和输出宏块的模式做了统计分析，测试序列为“foreman”和“mobile”，300 帧，帧率为 25fps，码率 2024kbps，分辨率为 352x288，GOP 结构的“IPPP...”，目标码率分别为 64kbps、576kbps 和 832kbps，结果分别见表 4.4 和表 4.5。

表 4.4 级联转码输入输出模式 (foreman)

目标码率 (kbps)	输入 模式	输出模式(%)						
		I4x4	I16x16	Skip	P16x16	P16x8	P8x16	P8x8
64	Intra	6.88	47.6	22.92	22.6	0	0	0
	Inter	0.63	5.12	70.06	24.19	0	0	0
576	Intra	54.79	6.98	1.04	16.77	4.9	5.1	10.42
	Inter	2.4	2.2	20.19	48.89	9.82	9.85	6.64
832	Intra	58.54	5	0.52	13.54	5.21	5.83	11.35
	Inter	2.59	2.18	13.12	52	10.79	10.89	8.43

表 4.5 级联转码输入输出模式 (mobile)

目标码率 (kbps)	输入 模式	输出模式(%)						
		I4x4	I16x16	Skip	P16x16	P16x8	P8x16	P8x8
64	Intra	1.86	39.75	14.29	44.1	0	0	0
	Inter	0.42	2.57	70.76	26.24	0	0	0
576	Intra	10.56	4.97	0	62.73	3.11	5.59	13.04
	Inter	0.71	0.33	17.25	60.73	8.62	9.06	3.3
832	Intra	11.8	4.35	0.62	57.14	6.21	7.45	12.42
	Inter	0.71	0.29	10.88	61.59	10.46	11.04	5.02

根据级联全解全编转码的输入输出宏块模式映射的统计结果，做出了高效转码中采用的模式映射表，见表 4.6。

表 4.6 H. 263/H. 264 输入输出宏块编码模式映射表

输入 H.263 模式	可能的 H.264 输出模式
Intra	{ SKIP, I16x16, I4x4, P16x16 }
Inter	{ SKIP, P16x16,P8x8,P16x8,P8x16,I16x16}

对于 I 帧，仅考虑 I16x16 和 I4x4 两种宏块模式。对于输入是 P 帧的 Intra 块，输出宏块为 I16x16 的可能性最大，为了进一步提高质量可加入 P16x16 模式判决；对于输入 Inter 块，由于宏块大小为 16x16，所以在模式判决时首先考虑 P16x16，这样做同时也是考虑到 H.264 编码器中 P8x8 判决的复杂度要远比 P16x16 高。如果 P16x16 的模式过于单一而不能保证质量，就需要引入 P8x8，并且根据 P8x8 搜索的最佳 MV 采用模式合并的方法加入 P16x8 和 P8x16 模式(在 4.3.2.2 节中详细介绍)。

另外在编码过程中，还通过预先判决的方法直接将宏块判为 SKIP，从而省略运动估计及后续的编码过程，大幅度提高转码速度。根据 4.3.1 复用 COD 信息的讨论可知，若输入 H.263 宏块的 COD 值为 1，并且当前宏块在 H.264 编码器中计

算出的预测运动矢量为零, 则可以直接预判为 SKIP 块。若 H.263 的输入运动矢量和 H.264 预测运动矢量相同, 而且在当前宏块周围至少有一个 SKIP 块时, 也预判当前宏块为 SKIP 块。通过预先判决的方法, 可以提前终止模式判决和复杂的 MV 搜索细化, 大大减少了时间复杂度。

4.3.2.2 帧间 P8x8 块的判决

由上述的分析可知对于输入 H.263 Inter 块, 在模式判决时首先考虑 P16x16, 但是 P16x16 的单一模式过于简单而不能保证质量, 这时就需要引入 P8x8 以及更复杂的模式。但是在 H.264 编码器中 P8x8 的模式判决要远比 P16x16 耗时, 因此在保证质量的前提下需要考虑 P8x8 模式判决的提前终止, 以节约复杂度。

在这里定义了宏块预测误差的平均近似 $AAPerror^{[36]}$ (Average Approximation of the Prediction Error):

$$AAPerror = \frac{1}{N} \sum_{i=0}^{N-1} DCT_i \quad (4-1)$$

DCT_i 代表宏块 i 中非零 DCT 系数的和, N 是宏块数目。若 $DCT_i > AAPerror$ 则表示该宏块预测误差较大, 需要对其进一步的细化 MV 或者寻找更合适的宏块尺寸。因此这里为了在保证质量前提下尽可能提高转码速度, 采用 $AAPerror$ 作为一个门限来判定是否增加 P8x8 模式判决, 使用此门限可以最大限度的减少输出模式的复杂度, 减少模式判决时间提高转码速度。

如果计算出 P8x8 模式的代价小于 P16x16 代价, 并不立即将宏块直接判为 P8x8, 而是根据 4 个 8x8 子块搜索出的最佳 MV 进行模式合并, 如图 4.3 所示, 检查是否可以判为 P16x8 或 P8x16 模式。具体的方法就是分别比较四个子块的 MV:

若子块 0、子块 1 的 MV 相等且子块 2、子块 3 的 MV 也相等, 则该宏块判为 P16x8 ;

同理若子块 0 与子块 2、子块 1 与子块 3 的 MV 也分别相等, 则宏块判为 P8x16; 否则该宏块仍判为 P8x8。

这种模式合并的方法, 可以进一步增加模式的复杂性同时也不会带来过多的时间复杂度, 在后续介绍的高效分辨率和帧率转码结构中也都有应用。



图 4.3 P8x8 宏块的合并

综上所述,本章提出了如图 4.4 所述的 H.263 Inter 宏块的快速模式判决算法。

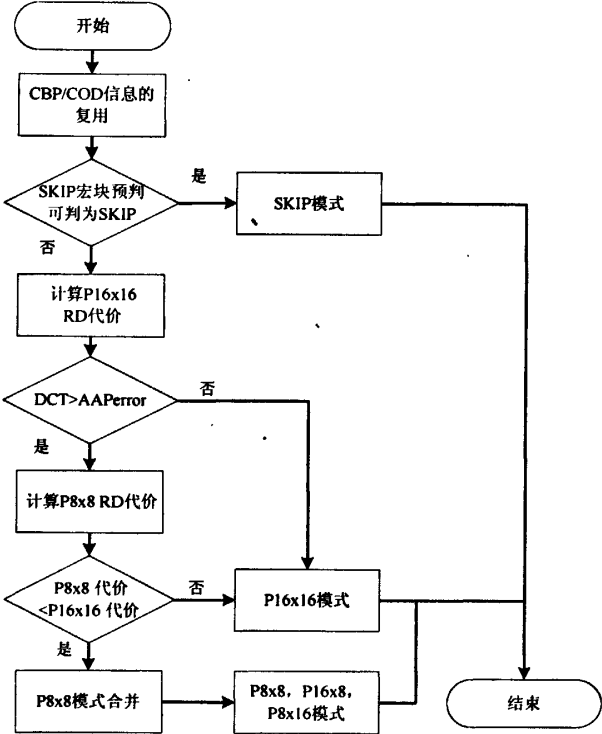


图 4.4 Inter 宏块的快速模式判决

4.3.3 运动信息估计

如上所述,与 H.263 标准相比 H.264 有着复杂的分区模式,因为 H.263 的输入码流仅有 Inter、Intra 两种宏块分区模式,意味着在编码侧可以利用的输入宏块信息非常少,而且可以直接利用的作为编码器的起始 MV 非常少。H.264 编码器为了保证质量要求,不仅需要模式重新判决,还需要进一步的 MV 搜索细化,这就难免带来了实时转码的困难。

从表 4.1 中我们知道 H.263 的运动补偿精度是 1/2 像素,而 H.264 的运动补偿精度是 1/4 像素,因此 H.263 解码得到的 MV 信息首先需要进行转换(乘以 2),然后作为 H.264 编码器运动估计的起始位置搜索点 (MVstart),进行钻石搜索。我们采取的搜索方式是以 MVstart 为起始位置搜索点,分别进行一轮整像素钻石搜索,一轮 1/2 像素钻石搜索和一轮 1/4 像素钻石搜索。

图 4.5 具体显示了如何从 H.263 解码得到的 MV 信息映射到 H.264 编码器运动估计的起始位置搜索点 MVstart。

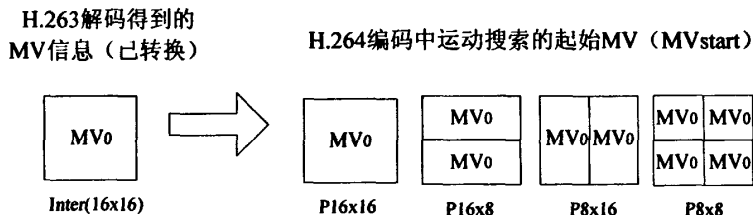


图 4.5 MV 信息的映射

为了进一步减小 H.264 编码器的复杂度, 在运动搜索里加入零块判决^[37], 如果当前块的 SAD(sum of absolute difference)小于一个和量化步长相关的常数 ($M * N * 25Q_{step} / 192$, $M \times N$ 为宏块大小), 则提前终止运动搜索, 也不再继续进行变换和量化操作, 以此节省了编码时间。

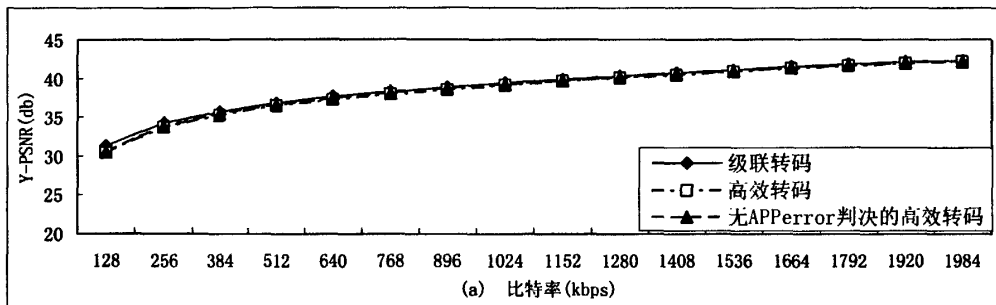
4.3.4 实验结果与性能分析

本章所有实验测试环境均为: Pentium(R) M CPU 1.6GHz, 512MB 内存; 操作系统: Microsoft Windows XP Home Edition 版本 2002 Service Pack 2。

测试序列是“foreman”和“carphone”, 它们的空间分辨率分别为 CIF(352x288) 和 QCIF(176x144)。H.264 编码器采用 x264, 生成 foreman 序列时, 采用 GOP 结构的“IPPP...”, 码率为 2024Kbps, 帧率为 25fps; 生成 carphone 序列时, 编码采用 GOP 结构的“IPPP...”, 码率为 768Kbps, 帧率为 25fps。

为了衡量本节所提出的快速模式判决算法性能, 分别比较了级联转码、高效转码和无 AAPerror 判决的高效转码性能, 其中无 AAPerror 判决的高效转码采用的快速模式判决不加入 P8x8 模式判决的提前终止 (4.3.2.2 节所述), 其余转码配置条件相同: 对测试序列“foreman”, 目标帧率是 25fps, 目标分辨率 352x288, 目标码率: 128—1984Kbps (间隔为 64Kbps); 对测试序列“carphone”, 目标帧率是 25fps, 目标分辨率 176x144, 目标码率: 128—768Kbps (间隔为 64Kbps)。

实验结果分别如图 4.6、图 4.7 所示, 图 4.8 为转码前后主观质量比较。



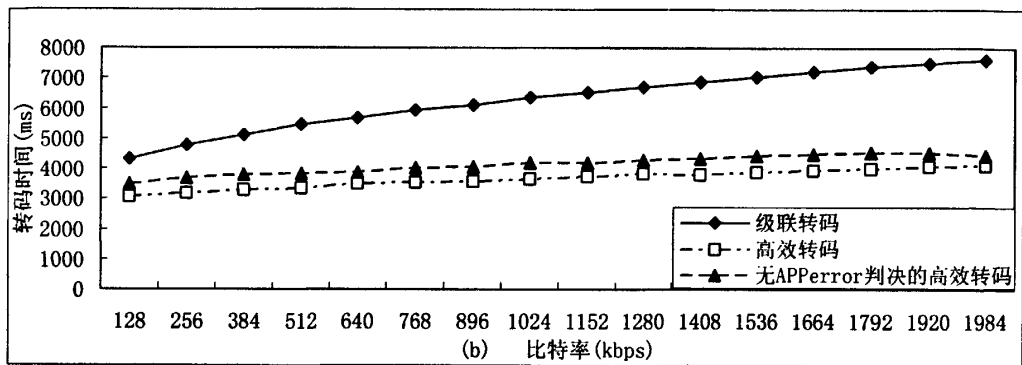


图 4.6 (a)foreman 序列的转码 PSNR 性能曲线
(b)foreman序列的转码时间性能曲线

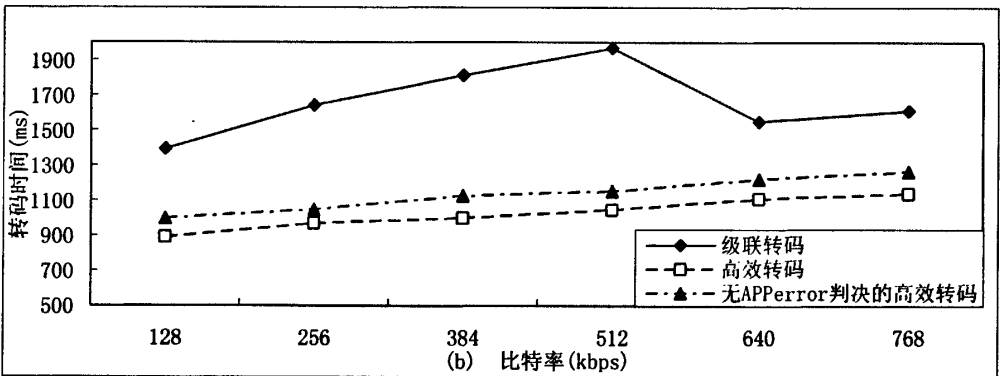
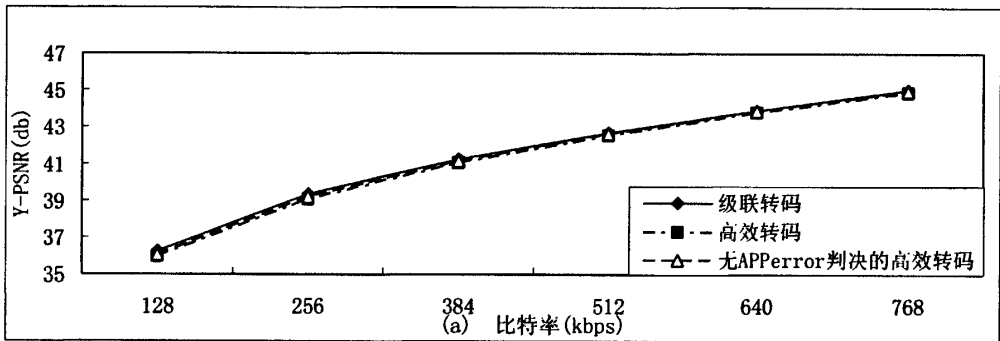


图4.7 (a)carphone序列的转码PSNR性能曲线
(b)carphone序列的转码时间性能曲线

从图 4.6 和图 4.7 中可以看出：在相同的参数配置条件下，高效转码同级联转码结果相比较，在 PSNR 下降很小的情况下（foreman 序列多数码率下在 0.1 dB 到 0.3dB 之间，极低目标码率下在 0.3 dB 到 0.8dB 之间；carphone 序列基本上在 0.1 dB 到 0.2dB 之间），转码速度有所提高(foreman 序列提高 30%~45%，carphone 序列提高 29%~46%)。不加入 AAPerror 提前终止判决 P8x8 模式的高效转码算法同级联相比较，PSNR 下降很小的情况下（foreman 序列多数码率下在 0.1 dB 到

0.2dB 之间, 极低目标码率下在 0.3 dB 到 0.7dB 之间; carphone 序列基本上在 0.02 dB 到 0.19dB 之间), 转码速度也有所提高(foreman 和 carphone 序列提高 20%~40%), 但是同原高效转码算法相比, PSNR 虽然有很小幅度的增加(0.058~0.207dB), 但是转码速度却降低了近 10%。

从图中我们可以看出, 随着目标码率的降低, 转码速度的提升幅度也随之降低, 这是因为级联转码中模式判决的流程如下: (1) 执行快速 SKIP 判决, 如果满足条件, 则提前终止。(2) 执行 P16x16 的运动估计。(3) 执行 P8x8 的运动估计。(4) 如果 P8x8 的代价 < P16x16 的代价, 则执行 P8x8 子模式的运动估计。(5) 如果 P8x8 的代价 < (P16x16 的代价 + 门限值), 则执行 P16x8 和 P8x16 的运动估计。(6) 执行帧内模式分析。(7) 选择最佳模式。

在目标码率较低时, 编码最佳模式为 SKIP 和 P16x16 的可能性较大, 因此上述判决流程提前终止的可能性也较大。另外, 编码最佳模式为其它模式的可能性相对较小, 因此执行 P8x8 子模式、P16x8 和 P8x16 的运动估计的可能性也相对较小。以上两个原因导致目标码率较低时级联转码的耗时也相应的较少, 所以快速转码的转码速度的提升幅度在目标码率较低时也相应的较低。

另外, 对于 carphone QCIF 序列目标码率 576kbps 处的时间性能产生拐点是本章实验采用的级联转码结构里 H.264 编码器在输入输出码率比很小的情况下不再考虑 8x8 以下的模式, 运动搜索采用小钻型, 从而导致了级联编码时间迅速减少, 在后面的高效分辨率和帧率转码中也有类似的情形。



图 4.8 转码前后主观图像质量(foreman, cif, 25fps, 2024kbps→1024kbps)

4.4 分辨率 2:1 下采样高效转码

分辨率降低高效转码中 I 帧的处理和码率转码相比, 只是增加了分辨率的下采样处理模块, 输入 H.263 码流的 Intra 块在模式判决上, 只进行 I16x16 和 I4x4 的判决, 其他部分的具体过程同码率转码。以下主要介绍 P 帧的分辨率 2:1 下

采样高效转码。

4.4.1 快速模式判决

当分辨率变换时，输入码流解码后得到的输入宏块信息已经不再适合于输出码流，需要重新确定新宏块的编码模式。在 H.263 中，由于宏块分区模式相对比较少，所以这里的模式判决准则比较简单。

分辨率 2: 1 下降时，输入四个宏块下采样时与一个输出宏块对应。分为以下几种情况考虑：

1) 若当前宏块可预判为 SKIP 块(4.3.2 节所述)，则直接判为 SKIP，结束模式判决。

2) 若输入 4 个宏块中全为帧内块，则下采样宏块映射为帧内编码块。

3) 除上述情况外，则下采样宏块映射为帧间编码块。

因此，我们给出如图 4.9 所示的 2: 1 分辨率下采样的快速模式判决算法。

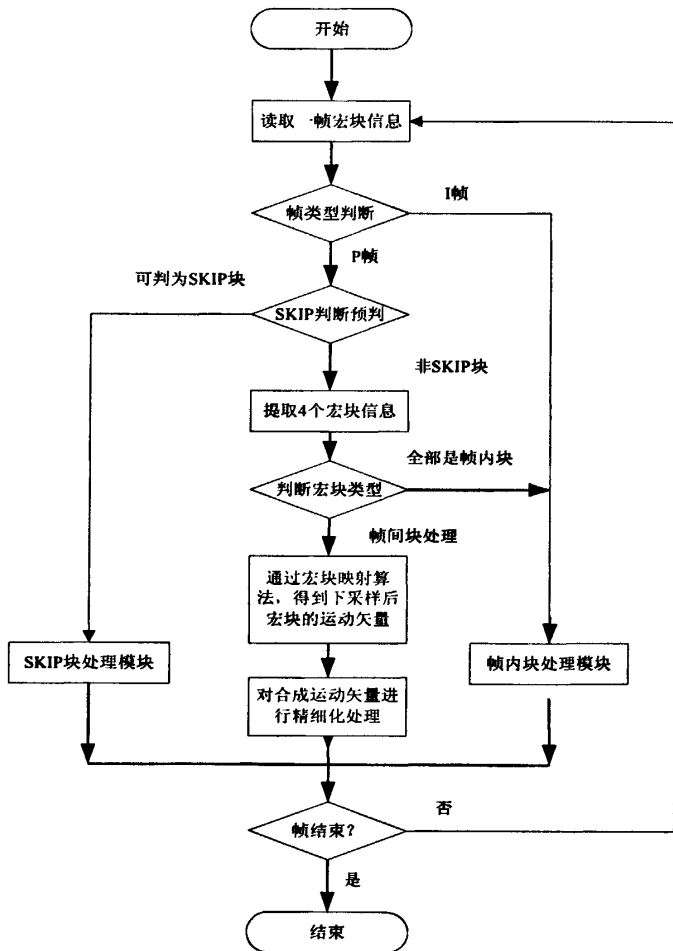


图 4.9 分辨率 2: 1 下采样中的模式判决

在情况 3) 中图像分辨率下采样后, 宏块模式仍需进一步细化, 而且解码输入码流得到的运动矢量不能被编码端复用, 需要对输入码流进行合成并且进行精细化处理。

本章采用单个宏块映射方案, 即输入的四个宏块分别对应输出宏块的每一个 8×8 块, 且下采样后的宏块取 4 个 MV, 即每个子 8×8 块对应 1 个 MV。根据其结果首先按照 4.3.2.2 节所述的 P8x8 模式合并方法做模式的预判决, 同时把 4 个子块的 MV 保存做为运动搜索的起始 MV。

为了尽可能的利用输入信息以节省编码时间, 通过计算四个输入子块的 MV 的最大夹角^[38], 来判断相邻宏块运动矢量是否足够分散, 进而可以判断是否需要宏块模式细化。

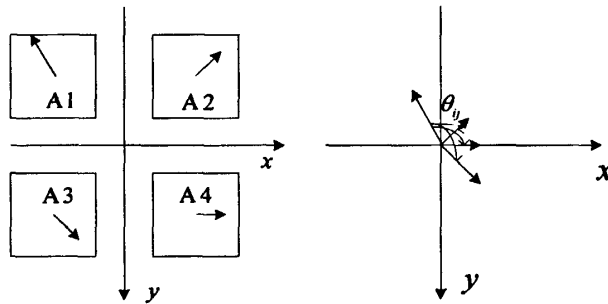


图 4.10 运动矢量夹角法

如所图 4.10 示, 假设有宏块 A1、A2、A3 和 A4, 其运动矢量为 mv_i , $i=1, \dots, 4$ 。将这些运动矢量移到同一个坐标下, 来计算它们相互之间的夹角 θ_{ij} , $i \neq j$, $i, j=1, \dots, 4$ 。在这些夹角值中寻找一个最大值 θ_{\max} :

$$\theta_{\max} = \max\{\theta_{ij}\}, i, j=1, \dots, 4 \quad (4-2)$$

通过式 (4-2) 最大夹角值来判定宏块运动方向是否分散。如果 θ_{\max} 小于预先给定的运动矢量夹角门限值, 则说明相邻宏块的运动方向比较一致, 下采样后对应的宏块被判决为 P16x16 块, 把输入的 4 个 MV 作为起始 MV 进行搜索; 反之, 则说明相邻宏块的运动方向比较分散, 下采样后对应宏块的模式需要细化, 首先进行 P8x8 模式的 MV 搜索和代价计算, 然后按照码率转码的方法进行模式合并。对于运动矢量的细化, 也是采用同 4.3.3 节的方法。

4.4.2 实验结果与性能分析

使用标准序列 Foreman 的 H.264 编码码流作为转码器的输入, 共 300 帧, 对于源码流: 分辨率为 CIF 格式, 帧率为 25fps; 输出码流: 分辨率为 QCIF 格式, 帧率为 25fps。仅考虑分辨率转码的条件下, 设置目标码率为源码率的 1/4。实验结果如下:

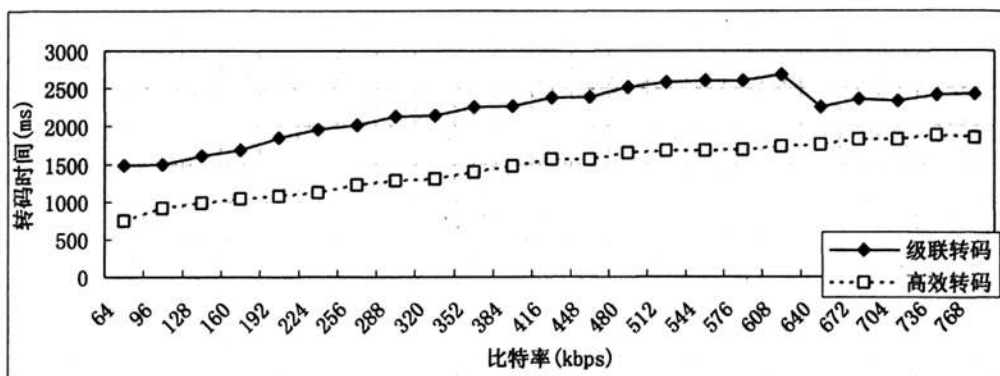


图 4.11 分辨率 2:1 转码转码时间对比 (foreman)

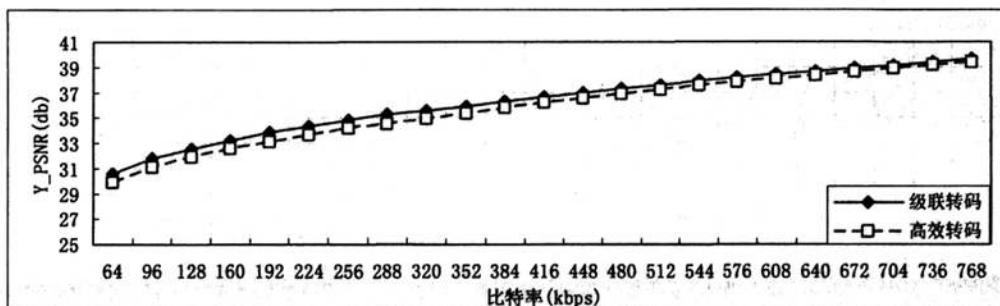


图 4.12 分辨率 2:1 转码转码平均 PSNR 对比 (foreman)

从以上图的实验结果可以看到, 在相同参数配置下, 高效转码相对级联转码平均 PSNR 相差了 0.2~0.6dB 左右, 而节省了 20%~40%左右的运算复杂度。图 4.13 为转码前后主观质量比较。



图 4.13 转码前后主观图像质量 (foreman, 25fps, cif→qcf, 2024kbps→506kbps)

下面再以 4CIF 格式码流作为转码输入码流, 做分辨率 2:1 下采样转码。采用标准序列 Coastguard, 目标分辨率为 CIF, 共 300 帧, 其他参数配置同上。实验结果如下所示, 可以看到, 相比全解全编级联转码, 高效转码的平均 PSNR 相差了仅 0.2~0.5dB, 复杂度节省了 30%~35%左右。

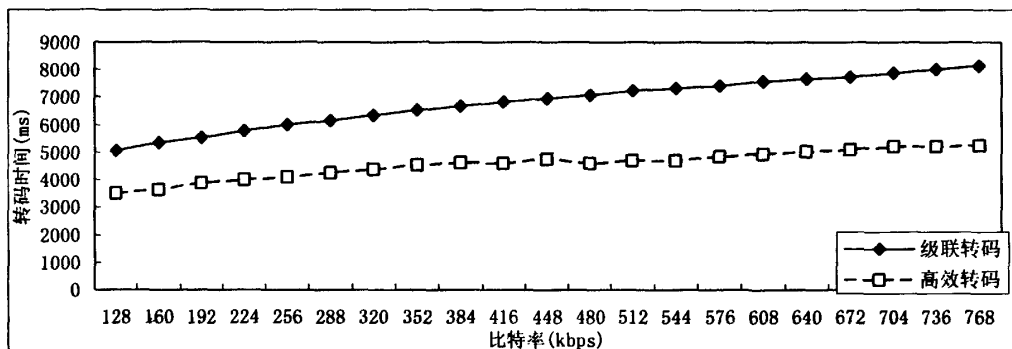


图 4.14 分辨率 2:1 转码转码时间对比(carphone)

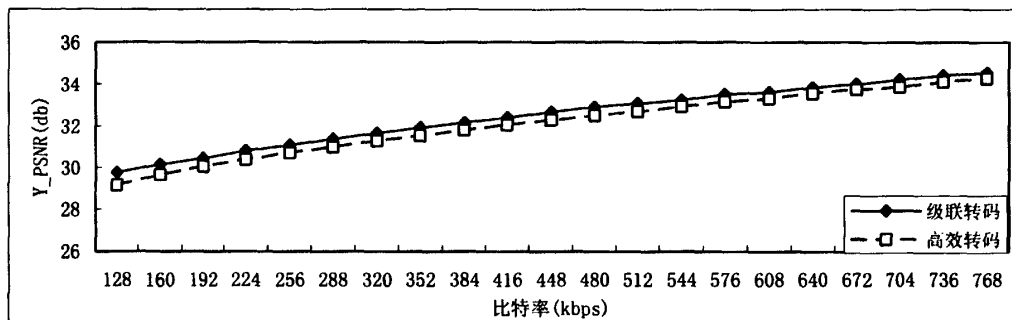


图 4.15 分辨率 2:1 转码转码平均 PSNR 对比(carphone)

4.5 高效帧率转码

4.5.1 丢帧策略

一般来说, 视频图像在时间域上的变化很小, 邻近的几帧中只有很小的变化, 以一定的速度丢弃几帧图像, 对人的视觉没有太大影响, 不会造成视频质量大幅度下降, 但是却可以节约了大量比特率。

帧率降低转码中, 设源帧率为 F , 目标帧率为 G , 则首先每 F_L 个源帧输出一个目标帧, 当目标帧数达到 G_L 时, 每 F_S 个源帧输出一个目标帧。每当源码流中出现一个 I 帧, 则重置上述过程。这一问题等效于将长度为 F 的块分割为 G 个长度几乎相等的子块, 即 G_L 个长度为 F_L 的子块和 G_S 个长度为 F_S 的子块。

$$F = G_L \cdot F_L + G_S \cdot F_S \quad (4-3)$$

其中,

$$F_L = \lceil F/G \rceil \quad (4-4)$$

$$F_S = \lfloor F/G \rfloor \quad (4-5)$$

$$G_L = F - F_s \cdot G \quad (4-6)$$

$$G_s = G - G_L \quad (4-7)$$

4.5.2 快速模式判决

视频转码中丢弃参考帧将不可避免地造成图像质量下降并且不断积累,而 I 帧是 GOP 的开始,作为后面 P 帧的参考帧有着比 P 帧更高的图像质量,考虑到丢弃码流中的 I 帧,将会破坏原有的 GOP 结构而导致误差积累范围加大,所以帧率下采样转码不允许丢弃 I 帧,故只能丢弃 P 帧。

输入 H.263 码流 I 帧的 Intra 块在模式判决上,只进行 I16x16 和 I4x4 的判决,具体过程同码率转码。对于帧率降低转码,P 帧的处理也只是在码率转码的基础上增加了帧率下采样、丢帧运动信息的保存和运动矢量的重建处理模块。在进行模式判决时,分以下两种情形考虑:

- 1) 若输入为 Intra 块,除了考虑帧内块,还加入了 P16x16 的判决;
- 2) 若输入为 Inter 块,并且当前帧的参考帧是不是丢弃帧,则无需考虑 MV 的重建,等同于码率转码;
- 3) 否则若输入为 Inter 块并且当前帧的参考帧是丢弃帧,就需要利用丢帧的运动矢量进行 MV 重建,并且重新判决宏块模式,以下详细介绍。

● 运动矢量重建

运动矢量的重建算法主要有双线性插值法,FDVS 运动矢量重建算法等。其中,双线性插值法虽然节省了转码运算量,但同时引入两个明显缺点:

(1) 当连续丢帧时需要保存所有到上一个非丢弃帧的前几帧的信息,这需要很大的缓存;

(2) 由于参考块所覆盖有四个宏块,四个宏块的运动矢量可能比较分散,用这四个运动矢量来组成的新的运动矢量,有时反而会导致重建的运动矢量与实际的最优运动矢量出现比较大的偏差。

为克服上述缺点,高效转码器将采用 FDVS 运动矢量重建算法。具体的 FDVS 运动矢量合成算法参见 2.4.1 节。

● 宏块模式重新判决

输入为 H.263 的 Inter 块且参考帧是丢弃帧,则宏块模式需要重新判决。如图 4.16 所示,首先进行 P16x16 模式判决,计算其 RD 代价;然后按照码率转码 4.3.2 节里 SKIP 预判和 P16x16 的预判方法进行模式的预判,通过预判可以立即终止模式判决、运动搜索等操作减小编码器复杂度;最后进行 P8x8 的模式判决,这里我们仍然利用码率高效转码中模式合并的方法对其结果进行合并。

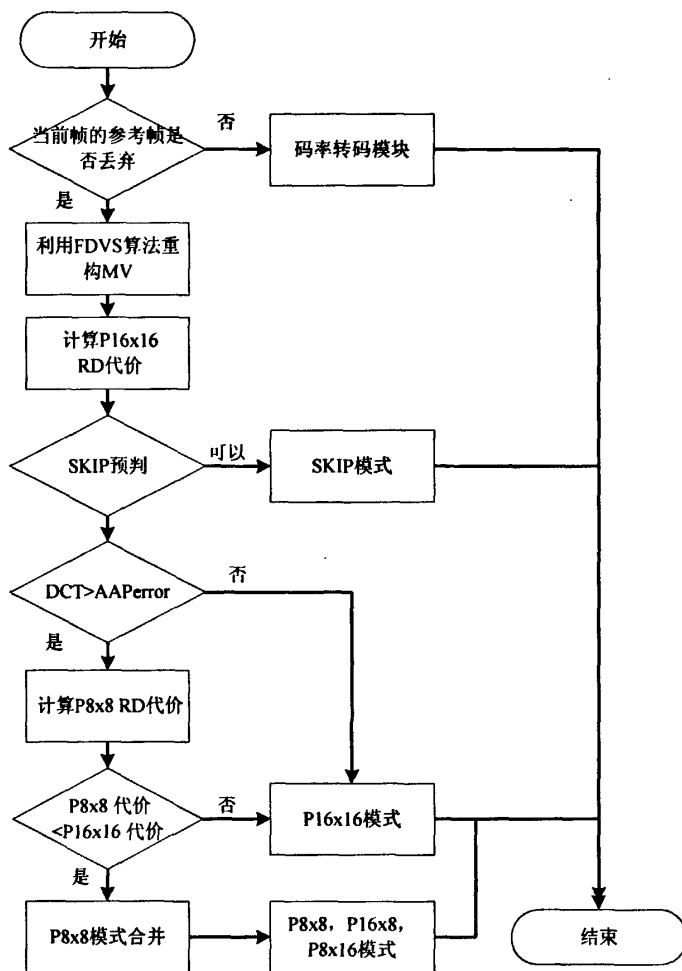


图 4.16 帧率转码 H.263Inter 宏块的快速模式判决

4.5.3 实验结果与性能分析

分别使用标准图像序列 Foreman 和 Carphone 的 H.263 编码码流作为转码器的输入码流。对于 Foreman, 分辨率为 CIF, 源码率为 2024kbps, 源帧率为 25fps, GOP 为 250。对于 Carphone, 分辨率为 QCIF, 源码率为 768kbps, 源帧率为 25fps, GOP 为 250。仅考虑降低帧率转码条件下, 目标码率根据下式求得:

$$Br_{dst} = Br_{src} \frac{Fps_{dst}}{Fps_{src}} \quad (4-8)$$

式中, Fps_{dst} 为目标帧率, Fps_{src} 为源帧率, Br_{src} 为源码率。实验结果如下:

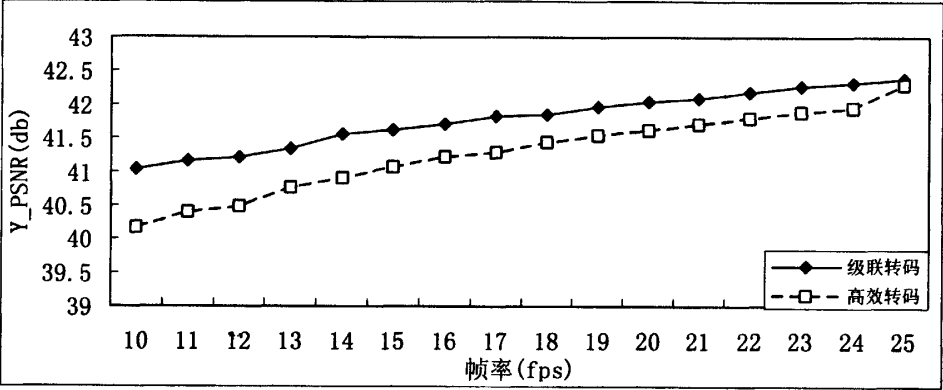


图 4.17 帧率转码平均 PSNR 对比 (Foreman)

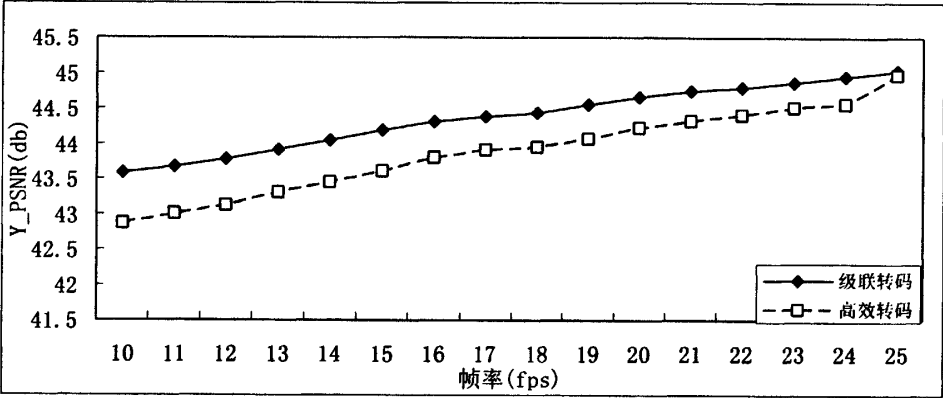


图 4.18 帧率转码平均 PSNR 对比 (Carphone)

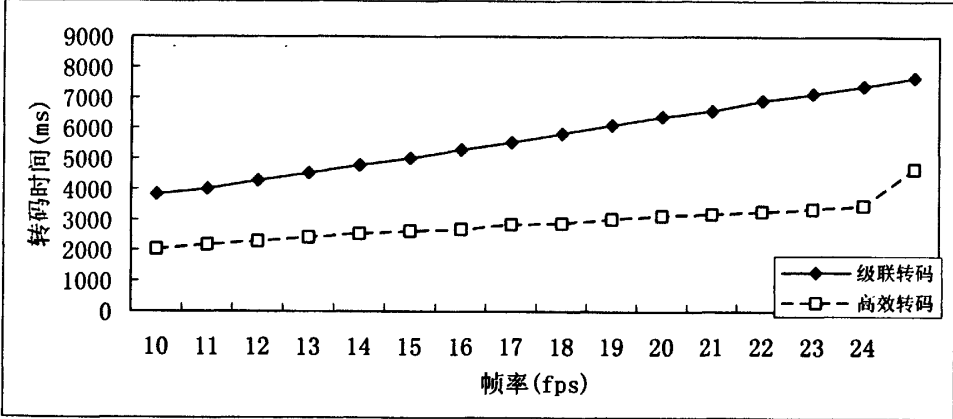


图 4.19 帧率转码耗时对比 (Foreman)

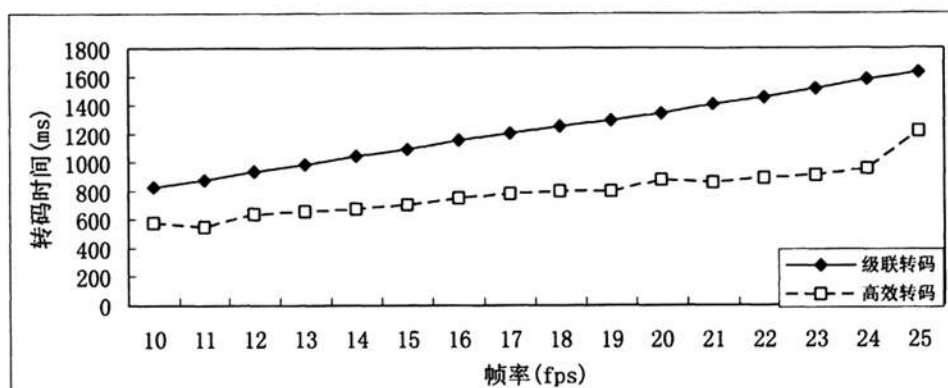


图 4.20 帧率转码耗时对比 (Carphone)

从图中可以看到, 对于 Foreman 序列, 与级联转码方法相比, 使用高效转码算法平均 PSNR 下降了 0.3~0.8 个 dB 左右, 而复杂度节省了约 45%; 对于 Carphone 序列, 高效转码相比级联转码, 平均 PSNR 下降了 0.3~0.7 个 dB, 而复杂度节省了约 30%~40%。并且由图 4.21 可知, 转码前后图像主观质量相差不大。

另外, 随着目标帧率的降低, 高效转码节省的耗时比例越来越小, 这主要是因为: (1) 丢帧数越多, 编码的帧数就越少, 故级联转码中编码耗时越来越少; (2) 丢帧数增多, 导致高效帧率转码器中运动矢量重建和细化的运算增加。

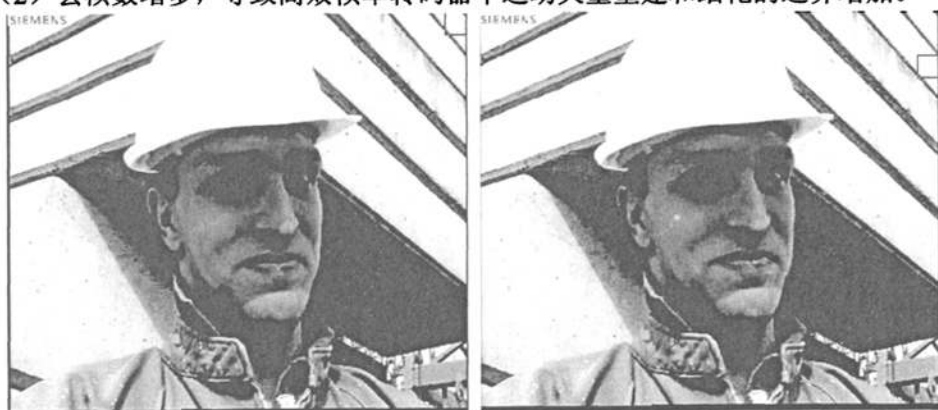


图 4.21 转码前后主观图像质量 (foreman, cif, 25→15fps, 2024kbps→1214kbps)

4.6 混合转码

4.6.1 空间/时间分辨率混合转码

在实际转码中, 通常需要将空间分辨率下采样转码和帧率下采样转码以及码率降低转码联合进行, 联合的空间分辨率下采样和帧率下采样及码率降低算法包含三个运算过程, 其一是帧率下采样, 其二为空间分辨率下采样, 其三则为码率编码, 由于码率变换主要在于编码侧量化参数的选取, 情况比较简单。

如前所述, I 帧不参与帧率下采样转码, 故 I 帧的转码流程与仅分辨率下采样转码流程相同。P 帧的高效联合转码, 帧率下采样需要对丢失帧的运动矢量进行保存处理及利用 FDVS 算法重建运动矢量; 空间分辨率下采样时需要对运动矢量进行下采样操作以及下采样后宏块类型的确定。

实际操作上, 在混合转码中包含了两层含义: 其一, 当前帧的参考帧是否是丢弃帧, 需要考虑帧率下采样; 其二, 图像分辨率下采样, 需要考虑 2:1 比例 (或者其他任意比例) 的分辨率下采样。不同的层次上对运动矢量的操作也不同。如图 4.22 所示, 具体过程为:

如果当前帧的参考帧不是丢弃帧, 则完全参照分辨率下采样转码操作, 仅需要对映射后的宏块所覆盖的输入宏块运动矢量进行映射合成;

若有丢弃帧, 则需要保存丢弃帧的运动矢量, 并且参照任意分辨率下采样转码的操作, 对所覆盖的每个输入宏块运动矢量进行映射, 然后利用丢弃帧的运动矢量通过 FDVS 算法进行重建。

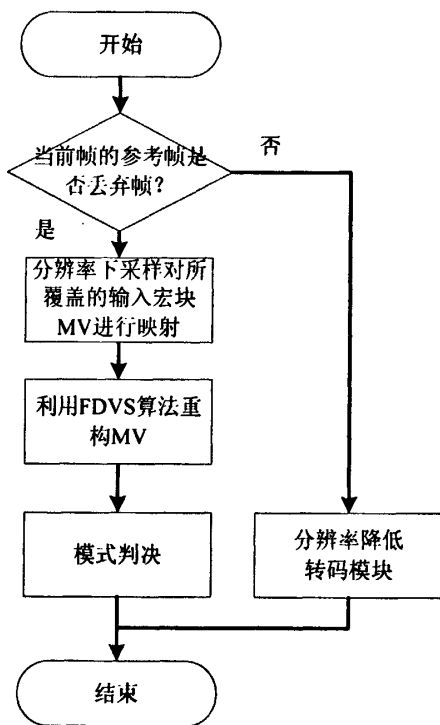


图 4.22 混合转码流程

其中模式判决过程先计算 P16x16 和 P8x8 的模式代价, 若 P16x16 的代价更小则直接判为 P16x16 宏块模式, 否则根据 P8x8 的判决结果按照 4.3.2.2 节的方法对宏块进行模式合并, 最后才判为 P8x8 模式。由此可见混合转码过程与分辨率降低转码相比只是在有帧率下采样要求时加上帧率转码中 MV 的重建处理模块即可。

4.6.2 整体转码框架

基于本章以上的分析,对于 H.263 到 H.264 标准间的整体转码结构可按照是否有帧率或分辨率降低的要求,分别转为码率降低、分辨率降低、帧率降低或者混合转码,如图 4.23 所示。如果有帧率降低,则需要保存丢弃帧的宏块信息和运动矢量等信息,此时若有分辨率降低,则为混合转码,否则仅为帧率降低转码。

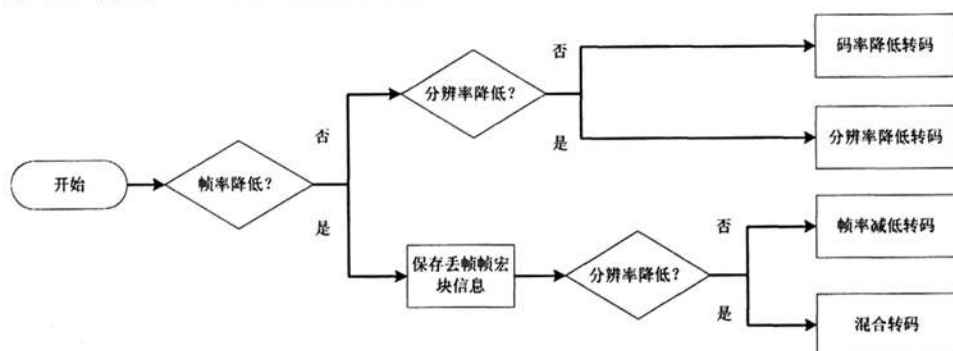


图 4.23 整体转码框架

4.6.3 实验结果与性能分析

为了衡量空间/时间分辨率混合高效转码性能,我们分别使用标准图像序列 Foreman 和 Coastguard 的 H.263 编码码流作为转码器的输入,对于 Foreman,分辨率为 CIF,源码率为 3072kbps,源帧率为 25fps, GOP 为 250;对于 Coastguard,分辨率为 4CIF;源码率为 3072kbps,源帧率为 25fps, GOP 为 250。分辨率下采样仅考虑 2:1 比例,分别为 QCIF 和 CIF 格式,目标码率根据下式求得:

$$Br_{dst} = \frac{Br_{src}}{4} \frac{Fps_{dst}}{Fps_{src}} \quad (4-9)$$

式中, Fps_{dst} 为目标帧率, Fps_{src} 为源帧率, Br_{src} 为源码率。实验结果如下:

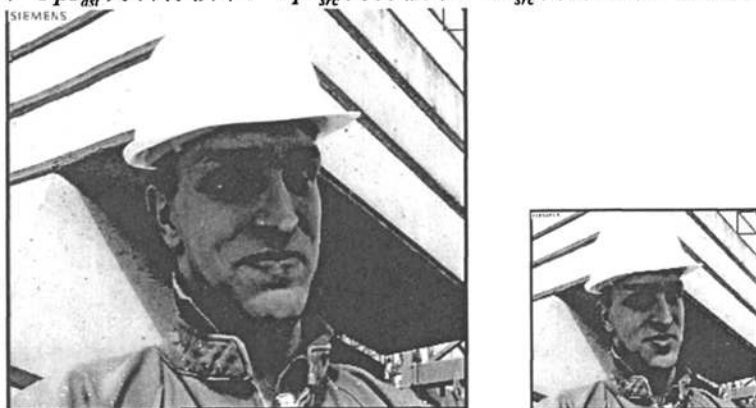


图 4.24 转码前后主观质量比较(foreman, cif→qcf, 25→15fps, 3072→460kbps)

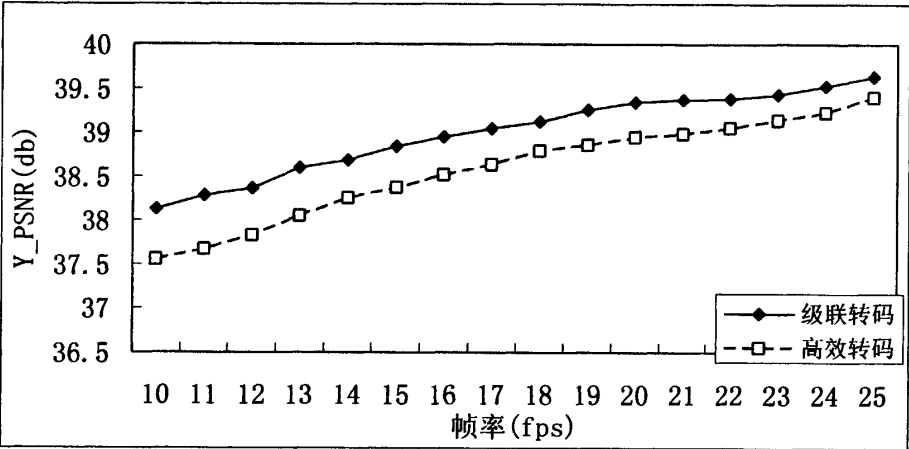


图 4.25 混合转码平均 PSNR 对比 (Foreman)

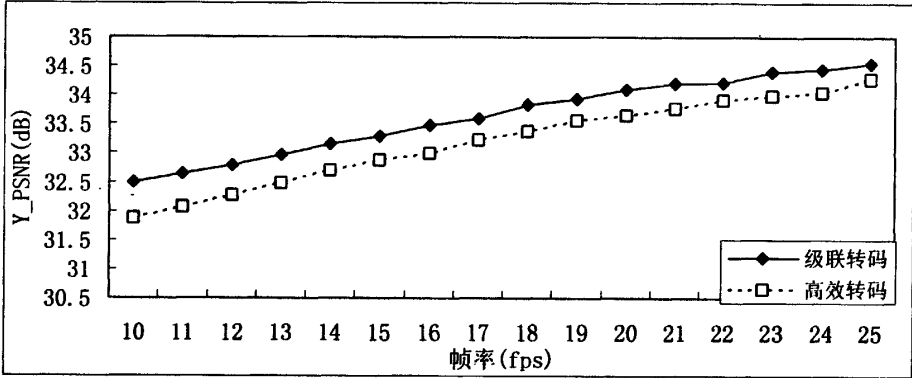


图 4.26 混合转码平均 PSNR 对比 (Coastguard)

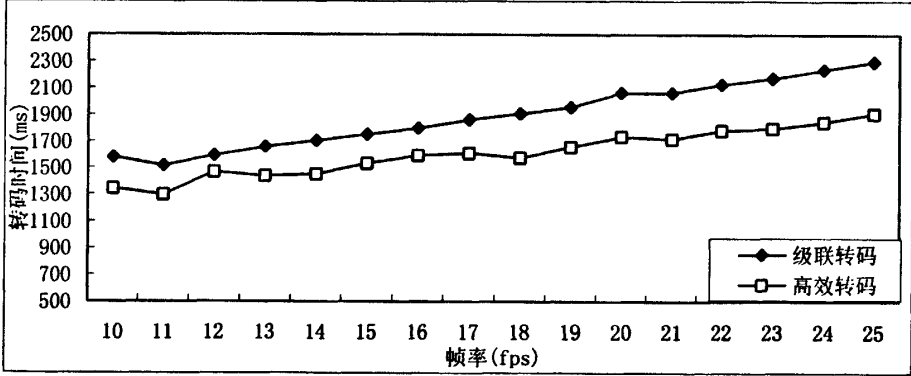


图 4.27 混合转码耗时对比 (Foreman)

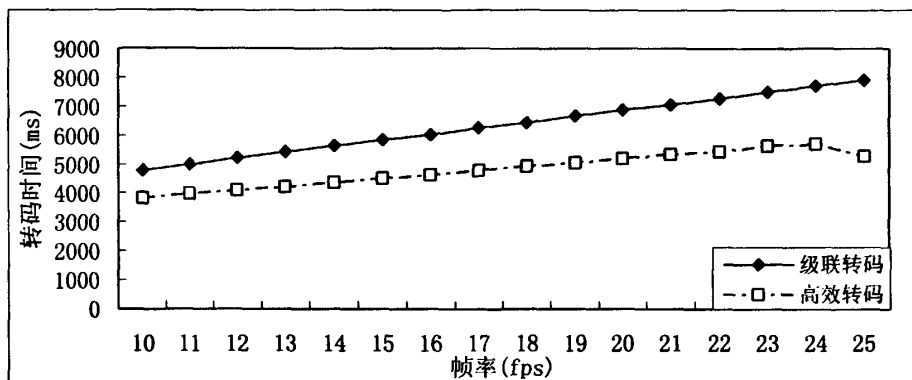


图 4.28 混合转码耗时对比 (Coastguard)

从图中可以看到, 对于 Foreman 序列, 与级联转码方法相比, 使用高效转码算法后平均 PSNR 下降了 0.2~0.5 个 dB 左右, 而复杂度节省了约 15%~20%; 对于 Carphone 序列, 高效转码相比级联转码, 平均 PSNR 下降了 0.2~0.7 个 dB, 而复杂度节省了约 20%~30%。

混合转码包括了码率, 分辨率和帧率的转换, 由于这种转码方式既损失输入码流的空间相关性, 也损失了输入码流的时间相关性, 从时间方面考虑, 即必须考虑分辨率下采样所引起的多个输入宏块运动矢量的映射, 同时还要考虑因为帧率转换导致的运动矢量重建问题, 因此实现高效转码难度很大, 时间复杂度的节省与单方面的分辨率或帧率转码相比都要小。另外随着目标帧率的降低, 高效转码节省的耗时比例越来越小, 这主要是因为丢帧数越多, 编码的帧数就越少, 将大大减少故级联转码中编码耗时, 同时输出码流相对输入码流来说分辨率变为原来的 1/4, 也意味着编码时间在整个转码时间中的比重越来越少, 使得高效转码时间节省相比于级联转码很小。

4.7 本章小结

随着多媒体技术的不断发展, 出现了多种编码标准共存的状态, 不同的编码标准都有各自的应用环境。另一方面在实际应用过程中, 由于网络条件和接收设备类型的差异, 也使得标准间转码技术正变得越来越重要。本章着重讨论了降低 H.263 与 H.264 之间转码计算量的方法以及需要考虑的问题, 主要针对两种标准的差异, 分别从码率转码、空间分辨率 2:1 下采样转码、帧率转码和混合转码四个方面分析, 提出了 H.263 宏块信息的复用策略以及快速模式判决算法, 对于空间和时间分辨率转码还提出了运动矢量的合成、细化和重建算法。最后结合混合转码提出了 H.263 到 H.264 标准间高效转码的整体框架。通过简单分析可以看出, 本章算法可以大大降低转码计算量, 具有实际应用价值。

第五章 总结与展望

5.1 全文总结

随着多媒体压缩编码技术和网络技术的不断发展,系统与网络之间的兼容性越来越重要。面对不同的终端和不同的网络状况,通用多媒体接入设备需要为用户提供透明的视频数据的交换与操作,必须具有视频压缩编码实时转换的能力。另一方面,新一代视频压缩标准 H.264/AVC 采用了近几年视频编码方面的先进技术,具有较高的编码效率,因此,基于 H.264/AVC 标准的转码对后续转码研究具有深远意义。目前在实时视频通信中 H.263 仍然应用较多,在一定时期内二者必将同时存在,因此研究 H.263 到 H.264 标准间的转码算法有着重要的实际意义。

本文工作主要围绕 H.264 标准内码率转码和 H.263 到 H.264 标准间转码展开,提出了相应的解决方案。本文的主要贡献是:

首先,通过分析 H.264/AVC 率失真优化算法和级联全解全编转码结构,本文提出了基于率失真的快速模式判决算法,通过充分利用参考帧的图像信息和解码宏块的运动矢量、编码类型和残差等信息,估计宏块失真来直接选择宏块模式,大大简化编码器的运算复杂度,提高转码速度,对实时多媒体传输及其它数据传输有着重要意义。

其次,针对 H.263 到 H.264 标准间的高效转码,本文分别从码率转码、空间分辨率 2:1 下采样转码、帧率转码和混合转码四个方面进行了研究。首先对级联全解全编转码进行了性能分析,指出实现高效转码的关键,第一是充分提取 H.263 解码隐藏的信息,第二简化 H.264 编码器复杂度。本文根据两种标准的差异,提出了 H.263 宏块模式、运动矢量、CBP、COD 等相关信息复用的策略,并根据此提出了宏块模式快速判决算法;对于复杂度比较高的 P8x8 模式判决,使用了提前终止判决策略;对于分辨率降低和帧率降低转码还提出了相应的运动矢量合成、细化和重建算法;最后结合混合转码提出了 H.263 到 H.264 标准间的高效转码整体框架。本文提出的 H.263 到 H.264 标准间的高效转码算法,在相同转码参数配置下与级联转码器相比,大大提高了转换编码的速度,极大的提高转码器的转码能力,同时具有稳定的率失真性能,保证了视频主客观质量,可用于实际视频转码系统中。

5.2 展望

在本文的研究基础上，还可以进行以下改进工作：

1. 利用输入输出宏块之间的联系，更准确的判定输出宏块编码预测模式，进一步提高宏块模式预测的准确性。
2. 利用参考帧和当前帧的运动信息，更准确的实现运动矢量定位，进一步减小运动矢量细化算法的复杂度。
3. 可以将 H.264/AVC 标准内码率转码的相关算法推广到分辨率、帧率以及混合转码体系中，进一步加快转码速度。
4. 可以将 H.263 到 H.264 标准间转码的相关算法推广到其他标准到 H.264 转码的体系结构中，进一步完善整个转码体系。

致谢

回顾三年来的学习和生活，要感谢许多人给予我的帮助，使我的研究生生活充实而又有意义，也使得我能够克服各种各样的困难完成这篇论文。

首先诚挚的感谢我的导师宋彬教授，在两年多的学习和工作中，他提供给我许多建议，使我终生受益，引领着我明确了人生的方向。宋老师知识广博精深，治学态度严谨求实，工作精益求精，这些都深深影响了我，是我将来学习和工作中用之不竭的精神财富。

深深的感谢秦浩副教授，他旺盛的工作热情、活跃的思维方式、高度的敬业精神、兢兢业业的工作作风和大胆创新的进取精神都对给我产生重要影响，与他的每一次讨论都使我受益匪浅，这篇论文的完成也离不开您细心的指导。

特别感谢我的家人对我学业上的支持和生活上的照顾，没有他们在我求学路上一贯的坚定支持，我不会取得今天的成绩，深深感谢他们。

感谢实验室的古立峰、李子诺、秦艳辉、贺红，和已经毕业的师兄罗广涛、邹箭、李兵伟和张东川，几年来得到他们许多的支持和帮助，让我在课题研究中让我学到很多经验和方法，使我能够在一个轻松愉悦、积极向上的实验室氛围内工作学习。还要感谢时振帅、权安以及许多我身边的朋友们，在我求学期间，他们给我许多的欢乐、帮助和关怀，使我觉得充实而温暖。

衷心感谢给予我支持与帮助的每一个人！

参考文献

- [1] ISO/IEC IS 11172, Coding of Moving Pictures and Associated Audio for Digital Storage Media Up to About 1.5Mbit/s, 1993. (MPEG-1).
- [2] ISO/IEC IS 13818-2, Information technology-Generic coding of moving pictures and associated audio information:Video,1995.(MPEG-2 video).
- [3] ISO/IEC IS 14496-2, Information technology-Generic coding of audio-visual objects part 2: Visual, 1999. (MPEG-4 Video).
- [4] ITU-T Recommendation H.261, Video codec for audiovisual services at p x 64 kbits, Dex, 1990, Mar.1993(revised).
- [5] ITU-T Recommendation H.263, Video coding for low bitrate communication, ver. 1, Nov. 1995
- [6] ITU-T Recommendation H.263, Version 2("H.263+"), Video coding for low bitrate communication, Jan. 1998.
- [7] ITU-T "H.263++" Annexes U, V, and W to Recommendation H.263, Nov. 2000
- [8] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC), JVT-G050r1, May 2003
- [9] ITU-T Recommendation H.264(Draft), draft text of final draft international standard for advanced video coding, Mar. 2003
- [10] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra, Overview of the H.264/AVC video Coding Standard, IEEE Trans, on circuits and systems for video technology, 2003, vol. 13, no. 7, pp, 560-576
- [11] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: an overview of various techniques and research issues," IEEE Transactions on Multimedia, vol. 7, no. 5, pp. 793-804, 2005.
- [12] J. Xin, C.-W. Lin, M.-T. Sun, "Digital video transcoding, " IEEE Proceeding, vol. 93, no.1, pp. T. Berge. Rate Distortion Theory[M]. A Mathematical Basis for Data Compression, Prentice-Hall, Englewood Cliffs, 1971.
- [13] I. Ahmad, X. Wei, Y. Sun and Y. -Q. Zhang, "Video Transcoding: An overview of various Techniques and Research Issues," IEEE Transactions on Multimedia, Vol. 7, No. 5, pp. 793-804, Oct. 2005.

- [14] "A frequency-domain video transcoder for dynamic bitrate reduction of MPEG-2 bit streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, Dec. 1998
- [15] S.-F. Chang and D. G. Messerschmitt, "Manipulation and composition of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol.13, no. 1, pp. 1–11, Jan. 1995.
- [16] A. Eleftheriadis and D. Anastassiou, "Constrained and general dynamic rate shaping of compressed digital video," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Washington, DC, Oct. 1995, pp. 396–399.
- [17] S.-Z. Liu and A. C. Bovik, "Local bandwidth constrained fast inverse motion compensation for DCT-domain video transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 309–319, May 2002.
- [18] Shanableh, T. Ghanbari, M. "Transcoding architectures for DCT - domain heterogeneous video transcoding , " *Proc. IEEE Conf. Image Processing 2001*, vol. 1, Oct. 2001, pp. 433–436
- [19] Xin J, Lin J W , Sun M T, Digital video transcoding[J]. *Proceedings of the IEEE*, 2005, 93(1): 84-97
- [20] B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video down scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 929–936, Sep. 1999.
- [21] P. Yin and M. Wu, "Video transcoding by reducing spatial resolution," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2000, pp. 972–975.
- [22] G. Shen, B. Zeng, Y.-Q. Zhang, and M. L. Liou, "Transcoder with arbitrarily
- [23] resizing capability, " in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS) 2001*, vol. 5, Sydney, NSW, Australia, May 2001, M.-J. Chen, M.-C. Chu, and C.-W. Pan, "Efficient motion-estimation algorithm for reduced frame-rate video transcoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 269–275, Apr. 2002.
- [25] N. Bjork and C. Christopoulos, "Transcoder architecture for video coding," *IEEE Trans. Consumer Electron.*, vol. 44, pp. 88–98, Feb. 1998.
- [26] J. Xin, M.-T. Sun, K. Chun, and B. S. Choi, "Motion re-estimation for HDTV to SDTV transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS) 2002*, vol. 4, Geneva, Switzerland, May 2002, pp. 715–718
- [27] J.-N. Hwang, T.-D. Wu, and C.-W. Lin, "Dynamic frame-skipping in video transcoding , " *IEEE Second Workshop on Multimedia Signal Processing*,

- pp.616-621,Dec.1998.
- [28] J.Youn,M.-T.sun,and C.-W.Lin, "Motion Vector Refinement forHigh-Performance Transcoding, " IEEE Trans.Multimedia,vol.1,no.1,pp.30-40,Mar.1999.
- [29] J.Youn,M.-T.sun,and C.-W.Lin,"Motion estimation for high performance transcoding," IEEE Trans,ConsumerElectron,vol.44,no.3,pp.649-658,Aug.1998.
- [30] J.Youn,M.-T.sun,and C.-W.Lin, "A fast motion vector composition method for temporal transcoding," Proc IEEE Int. Symp. Circults and Syatems, vol.4, pp.243-246, Jul.1999.
- [31] K.Lim, G.Sullivan, T.Wiegand, "Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods", JVT-X101, Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, Jul.2007.
- [32] H. Nam, B.Dan, H.Kim, J.Jeong, and S.ko, "Low Complexity H.264 Transcoder for Bitrate Reduction", Communications and Information Technologies 2006. ISCIT'06, pp.679-682, Sep.2006.
- [33] J. Moon, Y.Moon, J. Kim, "A Computation Reduction Method for RDO Mode Decision Based on An Approximation of the Distortion", Image Processing, 2006 IEEE International Conference, pp.2481-2484, Oct.2006.
- [34] E. Y. Lam and J. W. Goodman, "A Mathematical Analysis of the DCT Coefficient Distributions for Images", IEEE Transactions on Image Processing, Vol. 9, No. 10, pp.1661-1666, 2000.
- [35] P. Zhang, Q. M. Huang, and Wen Gao, "Key technique of bitrate reduction for H.264 streams", PCM 2004, LNCS 3332, pp. 985-992, 2004.
- [36] Kai-Tai Fung, Wan-Chi Siu "Low complexity H.263 to H.264 video transcoding using motion vector decomposition", Proc. IEEE ISCAS, Vol.2 pp.908-911, May 2005.
- [37] X.W. Wang, Q. Li, H.J. Cui, and K. Tang, "Motion search based on best zero-block detection criterion", Journal of Tsinghua University (Sci & Tech), Vol. 43, No.7, pp.938-941, 2003.
- [38] 张春荣. 数字视频DCT域转码技术研究.上海交通大学博士论文. 2007年2月

研究成果

申请国家技术发明专利：

秦浩，刘海啸，宋彬，古立峰. 基于快速模式判决的 H.264 高效转码器. 已提交.

参加科研项目：

华为技术有限公司合作项目《H.263/MPEG-4/H.264 转码技术开发》.

基于H. 264的高效视频转码关键技术研究

作者：[刘海啸](#)

学位授予单位：[西安电子科技大学](#)

本文链接：http://d.g.wanfangdata.com.cn/Thesis_Y1669093.aspx