

浙江大学

硕士学位论文

基于重要块保护的视频压缩容错编码

姓名：邵根富

申请学位级别：硕士

专业：计算机应用技术

指导教师：陈纯；卜佳俊

20060501

摘 要

近年来,伴随着视频编解码技术的突飞猛进,特别是 MPEG-4 和 H.264 的出现,使得视频编码效率得到了很大的提升。另一方面,随着 Internet 和无线网运载能力的增强,基于 Internet 和无线网的视频传输成为可能。因此,各种各样的视频应用,例如视频点播,视频会议,视频监控等,开始在 Internet 和无线网上出现,并逐步流行起来。

然而,相对于当前网络的有限带宽来说,视频数据仍然过于庞大;而且大部分的视频应用都要求有一定的实时性,从而不可避免地造成网络数据包的丢失。由于压缩过的视频数据对于差错非常敏感,假如没有任何差错控制机制,则丢包会造成解码端视频质量的严重下降。因此,为了减轻信道误码对于解码端视频质量的影响,在编码端和解码端都必须引入一定的差错控制机制。

在深入研究现有差错控制机制的基础上,基于其中非平衡差错保护的思想,设计了一种基于重要块保护的 video 压缩容错编码方法,其中的重要块是那些在解码端不容易使用差错隐藏方法恢复的宏块。该机制利用统计学的方法,建立了一个以帧间块模式(Inter-block Mode)、运动矢量差(MVD)和绝对误差和(SAD)这三个量作为影响因素的模型,作为重要块决策方法。由于该机制的目的是通过保护重要块,再配合解码端的差错隐藏方法,提高整体的视频质量,因此,所设计的模型需要预先在编码端考虑解码端的差错隐藏方法。通过对大量序列的训练和结果的分析,可以得到一些系数,用以表示各影响因素的影响值和权重。最后,对所找出的重要块施加保护措施。实验结果表明,该机制能够准确地找出重要块,且相应的保护措施能够有效地减少错误发生时视频质量的损失。

关键词: 视频编解码 差错控制 容错机制 差错隐藏 重要块

Abstract

With the rapid development of video coding techniques, especially after the occurrence of MPEG-4 and H.264/AVC, video compression rate has been improved a lot. On the other hand, the *underlying networks (e.g. Internet, wireless, and etc.)* have also been improved to support video transmitting. So more and more video applications over network come into use, e.g. Video on Demand, video conferencing, and video surveillance.

However, the extremely high demand of video data still exceeds the ability of underlying Internet/wireless networks. Since most of the video applications don't allow too much delay, packets that carry video data often suffer from loss. The compressed video data are very sensitive to errors due to residues reduction. With no error control schemes, the decoded video often suffers from large quality degradation and causes visual artifacts. To reduce the effect of errors, a large number of error control schemes are introduced both in video encoder and decoder.

Based on the study of current error control schemes, we devise a novel error resilient video coding scheme. This scheme is called significant macroblock (SMB) protection-based error resilient video coding. It is a UEP-like method for inter macroblock protection. Statistical methods are used to establish an offline Inter-block mode, MVD and SAD-based SMB determination model. Error concealment (EC) function at decoder side is taken into consideration in advance and several parameters for further significant degree (SD) evaluation for an MB are generated. During *encoding*, we build an SD table for each frame based on the parameters and pick up those MBs with the largest SD values as SMBs. Few additional computations are induced in SMB determination, thus make our scheme practical in real time scenarios. Simulations show that this scheme has an acceptable SMB determination accuracy and the corresponding protections can prevent errors effectively.

Keywords: Video CODEC, Error Control, Error Resilience, Error Concealment, Significant Macroblock, SMB

第一章 绪论

1.1 引言

当今时代,信息技术和计算机互联网技术飞速发展,传统的文本型信息载体已经越来越不能满足人类获取信息的需要。多媒体信息集声音、图像和文字于一体,以最直观的方式表达信息,因此越来越受到人们的欢迎。

多媒体信息技术集声音、图像和文字于一体,集传统广播电视技术、计算机存储技术、计算机处理技术和计算机通信技术之大成,它将把计算机应用引入一个无所不能的广阔领域,成为未来的主要技术之一。多媒体技术不仅使计算机应用更有效,更接近人类习惯的信息交流方式,而且将开拓前所未有的应用领域,使信息空间走向多维化,使人们思想的表述不再局限于顺序的、单调的、狭窄的范围,而有了一个充分的自由空间。多媒体技术必将引起信息社会的一场划时代的革命,并推动着人类社会的不断进步。

由于图像、视频和声音本身数据量庞大,给存储和传输带来了很大的不便。因此,多媒体信息技术的研究,特别是音视频压缩技术的研究,有着重大意义。它使得各种各样的多媒体应用得以实现,例如可视电话、视频点播、视频会议系统、视频监控系统等。同时,由于多媒体文字、声音、静止图像、视频动态图像等信源数据有极强的相关性,即存在大量的冗余信息,只要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,多媒体数据压缩是完全可以实现的。因此,研究高效的多媒体数据压缩编码方法,对多媒体数据的存储和传输都有着重要的意义。

然而,高效的压缩编码在降低多媒体信息冗余度的同时,也使得多媒体信息对于差错异常敏感。尽管目前的 Internet 和无线网的带宽和鲁棒性都有了很大提高并已经能够适应多媒体数据的传输,但是仍然存在较高的误码率。因此,网络的丢包对于接收端视频的质量有很大的影响。但是,如果在编码端有目的地加入一定的冗余度,则能够尽可能地减少差错带来的影响。在传输信道误码率较高的情况下,如何在压缩效率和鲁棒性间做一个权衡,已经成为一个研究的热点。

1.2 研究背景

从 1985 年起,国际标准化组织(ISO)和国际电信联盟(ITU)发布了一系列用于图像和视频编码的标准,分别为 JPEG/MPEG 标准和 H.26x。随着标准的不断更新,视频的压缩效率也越来越高。另一方面,底层网络和协议层次的发展,使得压缩视频数据的传输成为可能。

但是, 压缩效率的提高一般是以减少原始图像的冗余度来获得的, 从一定程度上来说, 也就降低了其鲁棒性。当压缩视频数据在 Internet 或无线网上传输时, 如果信道发生误码, 则极易造成接收端视频质量的下降。误码对视频质量的下降不仅体现在当前帧, 还体现在后续帧中。这是由于大多数的视频压缩技术都采用了预测技术(帧内预测和帧间预测), 从而使当前帧中的错误会导致后续帧使用一个错误的基值预测, 而产生差错传播。因此, 如果能够通过增加一定的差错控制, 来提高差错恢复的能力, 则对于那些误码率高的视频传输系统来说, 是非常有帮助的。

传统的差错控制技术可以分为三类: 1) 在编码器端引入的; 2) 在解码器端引入的; 3) 需要由编码器和解码器交互的。在编码器端引入的差错控制主要是有目的地加入一些冗余信息, 预防错误发生时所带来的影响。在解码器端, 则主要通过差错隐藏的方法, 来降低由于差错所引起的视觉不顺畅感。由编码端和解码端交互的差错控制方法, 则是通过一个反馈信道, 传输解码端对差错的诊断信息, 从而在编码端有目的地预防差错和避免差错。对应于这三类差错控制技术, 可以衍生出许多差错控制的具体方法。由于不同的标准采取不同的视频编解码技术, 因此, 针对不同的标准, 也可设计不同的差错控制方法。

在常用的差错控制技术中, 非平衡差错保护(UEP)其中常用的方法之一。其核心思想是对重要程度不同的信息做不同级别的保护, 例如, 对于压缩视频来说, 头信息总是最重要的, 因此要重点保护; Intra 编码的数据次之, 则施加相对低层次的保护; Inter 编码的数据重要性最低, 因此施加最少的保护。然而, 必须注意到, 对于同一 Inter 帧, 并不是所有的宏块都是同等重要的, 即假设该帧中宏块丢失, 对接收端视频质量所造成的影响是不同的。假如能够将这些重要块检测出来, 并施加保护, 则将使得由于差错所造成的接收端视频质量下降得到控制。因此, 如何检测重要块和如何施加保护, 已经成为差错控制领域的一个热点。

1.3 研究的意义与目的

从主观上来看, 视频图像中运动较大的区域总是人们关心的重点; 从客观统计分析来看, 也可以得出那些运动较大或者运动无序的区域, 当发生差错时, 对所属的帧视频质量的下降影响最大, 对后续帧的差错传播影响也很大。因此, 假如能够将这些重要的区域或块重点保护起来, 则对于整体视频的质量将有较大的提高。

以往, 对于重要块的决策都是基于在编码器端模拟解码端的差错隐藏方法, 来寻找出那些最不容易隐藏错误的块。这种方法非常有效, 但是, 需要对每个块做模拟差错隐藏和二次编码, 因此, 计算复杂度非常高, 不能适用于实时环境, 只能结合转码使用于视频点播之类的应用中。

常用的重要块保护方法是对重要块做 Intra 编码。Intra 编码是一种有效的抑制错误传播的方法。但是, 由于 Intra 编码较之 Inter 编码需要近乎 10 倍的码流大小, 因此在网络带宽

受限的环境下,如果仅仅采用 Intra 编码,所保护到的块数量是很少的。同时,如何合理地选择一帧中的块做 Intra 编码,也是需要研究的难点之一。反之,如果设计一种合理的对重要的 Inter 数据做保护的方法,则将以较低的比特率获得视频质量的明显提高。

本文着重研究在保证可接受的查找精度的前提下,如何快速地找出重要块;然后设计一种基于重要块的差错保护方法。本文的研究结合人眼观察的特性和统计学的分析,摒弃了以往高复杂度的块决策方法,提出了一种新的快速的重要块决策方法和保护方法,使其适用于实时环境和嵌入式设备,这对于压缩视频差错控制方法领域有重大的意义。

1.4 论文的组织

在后面的章节中,将概要介绍视频编解码技术和压缩视频传输技术,着重介绍压缩视频差错控制技术和容错编码,并在此基础上提出快速的重要块决策和保护方法。本文共分为七章,分别安排如下:

第一章分析了当前多媒体技术和视频编解码技术的发展,概要介绍了压缩视频传输的发展状况,从而提出了差错控制的重要性和本文提出的方法的重要性。

第二章概要介绍了视频编解码相关技术和视频编解码标准的演化,分析了压缩视频在网络上传输的需求和可能性,并由此引入了差错控制机制。

第三章着重介绍了传统的视频压缩差错控制技术。根据差错控制技术在编解码器中的实施,可以分为前向容错,差错隐藏和交互式抗误码技术。

第四章和第五章分别就提出的基于统计分析的快速重要块决策方法和基于决策出的重要块保护的差错控制机制做了展开,并通过模拟实验给出了查找进度和差错控制效果。

第六章给出了本文提出的方法在移动视频中的应用情况。

第七章对全文的研究做了总结,包括本文的贡献和不足,并对接下来的研究方向做了展望。

第二章 压缩视频及其在网络上的传输

进入信息时代,多媒体信息越来越成为人们获取信息的主要工具。但由于原始采样的多媒体数据往往具有海量性,尤其是视频数据,因此,必须经过压缩。另一方面,随着 Internet 和无线网的发展,压缩视频在网络上的传输需求越来越大。视频的传输需要底层网络的支持。不同的底层网络和不同的应用需求,都需要定义不同的协议规范来保证视频的传输质量。致力于网络传输的 ITU-T 组织为视频传输定义了多个多媒体通信系列标准建议族(H.32x 系列),分别面向不同的场景和应用。同时,在保证视频传输质量方面,几种常用的控制机制也越来越成为研究的热点领域。而差错控制机制就是其中一种重要的方法。

2.1 视频编解码相关技术及其标准

随着电子信息化的推广,人们越来越依靠计算机获取和利用信息。由于数字化后的视频和音频等媒体数据仍然具有海量性,因此,当前的存储能力和网络资源显然不能满足未压缩媒体数据存储和传输的需要。这样,对多媒体信息的压缩就显得尤其重要。从目前计算机的软硬件发展水平及发展趋势来看,可以断言,在将来很长一段时间内,数字化的媒体信息数据以压缩形式存储和传播将仍是惟一的选择。

2.1.1 视频编解码概述

数据压缩的基本原理是把低效(长的)表达方式改为高效(短的)表达方式。减少冗余度的方法,是数据压缩的通用法则。这对于视频压缩来说,也一样。由于视频数据庞大,因此,必须进行压缩;同时,视频数据中存在的大量冗余性使得视频压缩成为可能。近年来,由于多媒体技术应用越来越广泛,视频编解码相关技术一直是个热点。本节主要对视频压缩、视频编解码及其相关技术做一个简单介绍。

2.1.1.1 视频数据压缩

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”。数字计算机面临的是数值、文字、语言、音乐、动画、静图像、电视视频图像等多种媒体承载的由模拟信号转化成数字信号的吞吐、存储和传输问题。数字化了的视频和音频信号的数量之大是非常惊人的。例如单幅 640×480 分辨率条件下的彩色视频图像(24 位真彩色),其总数据量约为 7.37Mbit/帧。如果是连续运动的视频图像,假设以每秒 30 帧的速度播放,那么视频

信号传输速率为 220Mbit/s; 如果把这种信号存放在 650MB 的光盘中, 一张光盘只能播放 20 多秒钟。如此巨大的数据量无疑给存储器容量、通信干线的信道传输率以及计算机速度都增加了极大的压力。同时, 普通 PC 机 ISA 总线的数据传输率最大只能达到 5MB/秒。为进行多媒体数据的实时处理, 必须采用某些技术例如视频数据压缩技术来降低多媒体数据的数据量, 用尽可能少的数据来表达信息, 实现快速的编解码算法, 节省传输和存储的开销。

多媒体视频信号是可以压缩的, 而且压缩比非常高。研究表明, 图像和视频数据都是高度相关的, 一幅图像的內部以及视频序列中的相邻图像之间存在着大量的冗余信息。最常见的是空间冗余、时间冗余和视觉冗余。空间冗余描述了同一帧图像中不同像素点之间的相关性。例如, 同一景物表面上各采样点的颜色之间往往存在着空间连贯性, 但是基于离散像素采样来表示物体颜色的方式通常没有利用景物表面颜色的这种空间连贯性, 从而产生了空间冗余。这种冗余可以通过改变物体表面颜色的像素存储方式, 达到减少数据量的目的。时间冗余的产生则是由于序列图像一般是位于一时间轴区间内的一组连续画面, 其中的相邻帧往往包含相同的背景和移动物体, 只不过移动物体所在的空间位置略微不同, 所以后一帧数据与前一帧的数据有许多共同的地方, 这种共同性是由于相邻帧记录了相邻时刻的同一场景画面, 也即所谓的时间冗余。视觉冗余的产生则是基于人类的视觉系统对图像场非均匀和非线性的敏感性, 而原始图像数据在采集时, 通常是线性和均匀采样的, 并未考虑人眼的特性。理想编码应当能够把视觉敏感和不敏感的部分区分开来编码, 从而去除视觉冗余。除了视觉冗余、空间冗余和时间冗余外, 媒体数据中常见的冗余信息还包括: 结构冗余、知识冗余、图像区域的相同冗余、纹理的统计冗余等。

通过去除那些冗余数据可以使原始图像数据极大的减少, 从而解决图像数据量巨大的问题。对应于不同的冗余数据, 可以采用不同的压缩方法。例如, 颜色空间转换通过去除或者降低视觉不敏感的分量, 从而消除视觉上的冗余; 通过运动估计 (Motion Estimation) 和运动补偿 (Motion Compensation) 可以消除视频的时间冗余; 通过对预测差值的离散余弦变换 (DCT), 可以消除空间冗余; 通过量化系数的可变长编码 (VLC), 可以消除统计冗余。

以压缩/解压后数据与压缩前原始数据是否完全一致作为衡量标准, 可将数据压缩划分为无失真压缩 (即可逆压缩) 和有失真压缩 (即不可逆压缩) 两类。像颜色空间的转换, 量化等方法都是属于失真压缩的方法; 而运动估计/运动补偿技术、DCT 变换等都是属于无失真压缩技术。一个好的视频编码器, 往往是有失真压缩和无失真压缩的结合体。

2.1.1.2 传统视频编解码过程分析

一、编码过程分析

视频编码的过程主要是消除冗余的一个过程。传统的混合 DPCM/DCT 编码主要是通过分块、运动估计与运动补偿、DCT 变换、量化、熵编码等方法, 将原始视频数据进行压缩

编码。图 2.1 给出了传统的混合 DPCM/DCT 编码框架图。在编码过程中，视频帧 n (F_n) 经过编码处理产生编码后的数据流 (Coded Bistream)。具体编码过程如下：

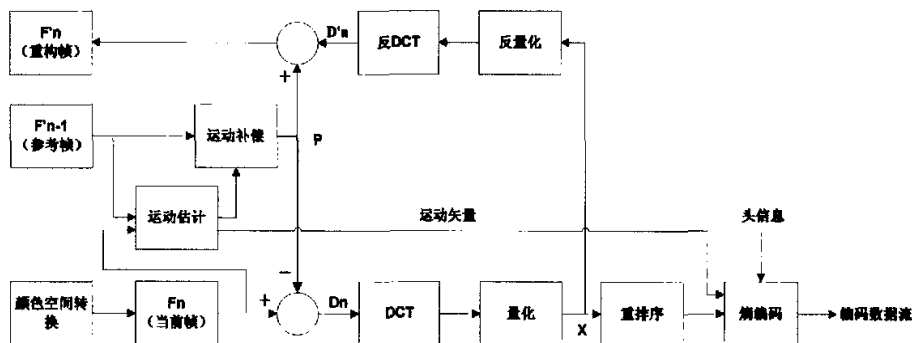


图 2.1 DPCM/DCT 编码框架图

- 1) 对于输入的原始帧数据，首先进行颜色空间的转换，将人眼敏感的分量和相对不敏感的分量分开，得到转换后的帧 F_n 。
- 2) 为了更好地进行后续的压缩处理，将整帧数据进行分块处理。后面的编码过程，都是基于划分好的块来进行的。
- 3) 使用运动估计算法从参考帧 F_{n-1} 中找出与当前宏块最匹配的区域，从而确定运动矢量 (MV) 和残差值 (Residual)，在这里，将由 ME 得到的运动矢量和残差分别定义为 MV 和 D 。
- 4) 对残差值 D 进行 DCT 变换。DCT 变换是基于更小的块进行的，一般为 8×8 或者 4×4 的子块。
- 5) 对每一子块进行量化得到量化后的残差系数 X 。量化是传统 DPCM/DCT 编码过程中唯一有损的环节，因此，使得解码端重构出的帧与原始帧有出入。为了保持编码端和解码端的一致性，在编码端编码某一帧时，不能直接使用其参考帧的原始数据，而需使用反量化、反 DCT 变换和运动补偿后得到的重构帧 F'_n 。
- 6) 由于量化后的残差系数会有许多的零，并且非零系数主要集中在左上角部分，因此对量化后的 DCT 系数重新扫描排序 (Reordered) 并进行行程编码 (Run-level Coding)。
- 7) 最后，行程编码后的 DCT 系数，运动矢量信息和相关头信息经过熵编码产生压缩的数据流。

二、解码过程分析

解码过程主要是一个反编码的过程，但又不全相同。解码主要包括反熵编码、重排序、反量化、反 DCT 变换等过程。示例 DPCM/DCT 解码过程可见图 2.2。具体流程如下：

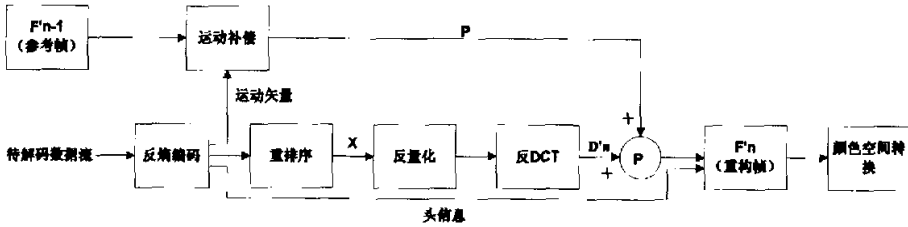


图 2.2 DPCM/DCT 解码框架图

- 1) 压缩后的数据流通过反熵编码提取出每一个宏块的系数，运动矢量和相关的头信息。
- 2) 通过行程编码和重新扫描排序，恢复出编码端 DCT 变换和量化后的残差 X 。
- 3) X 进行反量化和反 DCT 变换得到解码后的残差值 D' ；
- 4) 利用反熵编码中提取出的运动矢量和解码端已经解码的参考帧做运动补偿，将其与残差值 D' 相加，即得到解码后的重构帧 F'_n 。
- 5) F'_n 即提供给播放器播放的帧数据，同时也要保存在缓冲区中作为后续帧 F'_{n+1} 解码的参考帧。
- 6) 如播放器对所提供的颜色空间不支持，则需要进行反颜色空间转换。

通过编解码过程的分析，可以得到，当前流行的混合 DPCM/DCT 编解码包含的主要技术有：颜色空间转换（Colorspace Transform），运动估计和运动补偿（Motion Estimation & Motion Compensation），离散余弦变换（Discrete Cosine Transform），量化（Quantization），Zig-zag 扫描（Zig-zag Scan），熵编码（Entropy Coding）。这些方法将在下面 2.1.1.3 节给出。

2.1.1.3 视频编解码相关技术

上一小节给出了传统视频编解码的基本流程，本节概要讲述视频编解码相关技术及相关背景知识。传统的视频编解码技术在上世纪 80 年代就开始了研究，如今已经非常成熟了。本文此只做一个简单的介绍[1][2][3]。

● 颜色空间转换

RGB (Red/Green/Blue) 是基于光学理论中的三元色得到的，它可以用来表示万千世界中的各种各样的色彩，且简单直观，在电视机和计算机的颜色空间显示系统中应用广泛。但是，RGB 颜色空间没有考虑到任何人眼的特性，因此视觉冗余度很高。为了满足广播电视的需要，结合人眼的视觉特性，开发了 YUV、YCrCb (Luminance/Red Chrominance/Blue Chrominance) 等利于视频压缩的颜色空间。颜色空间的转换，发生在编码端的开始和解码端的结束。也可以将其看成是编解码过程的前处理和后处理阶段。在编码端，将 RGB 颜色空间转换成 YUV 或 YCrCb 空间，以利于编码压缩；在解码端，将 YUV 或 YCrCb 颜色空间转换成 RGB 颜色空间，以利于显示。

颜色空间转换之后, 仍然没有去除视觉冗余。去除视觉冗余的过程, 实际上是一个亚采样的过程。以 YUV 颜色空间为例, 常用的采样方法有 4:4:4、4:2:2、4:2:0, 如图 2.3 所示。由于亮度信息 (Y 分量) 是人眼最敏感的信息, 色度信息则相对较为不敏感, 因此, 对色度信息亚采样, 不会对视频质量造成大的影响。

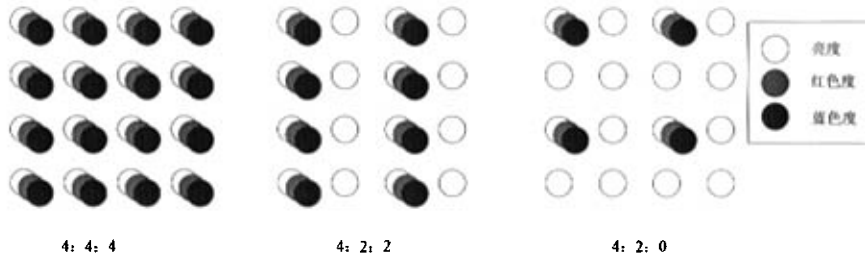


图 2.3 色度信息亚采样示例

● 运动估计和运动补偿

对于视频序列图像, 其相邻帧之间存在很大的时间相关性, 即时间冗余 (Temporal Redundancy)。运动图像多数情况下只是其中的很少一部分在运动, 并且运动不会很大。只要是在同一场景中, 相邻两幅图像之间在内容上的诧异也不会太大, 前后两帧重复的部分很多。因此, 在编码端, 并不需要把每帧图像上所有的像素信息都原模原样地传给接收端, 而只需将其与前一帧做比较, 找出最佳的匹配信息, 并计算残差。找出最佳匹配块的过程, 即为运动估计的过程; 计算残差的过程, 即为运动补偿的过程。对做过运动估计和运动补偿的帧来说, 在传输给解码端时, 只需将最佳匹配块的位置信息及计算得出的残差信息编码传送即可。在接收端, 基于前一帧图像的内容, 就可以根据解码出的运动信息和残差, 获得当前帧图像。运动估计和运动补偿方法比全部传送每帧图像的具体细节所需的数据量要小得多, 因此可以大幅度提高视频编码的效率。

图 2.4 给出了一个简单的运动估计示例。它的基本思想是将图像序列的每一帧分成许多互不重叠的宏块, 并认为宏块内所有像素的位移量都相同, 然后对于当前帧中的每一块到参考帧中某一给定搜索范围内根据一定的匹配准则找出与当前块最相似的块, 即匹配块。由匹配块与当前块的相对位置计算出运动位移, 所得的运动位移即为当前块的运动矢量。利用搜索得到的运动矢量在参考帧上进行运动补偿, 而得到的残差值 (Residual) 经 DCT 变换、量化、行程编码后与运动矢量共同经熵编码, 然后以比特流形式传出去。

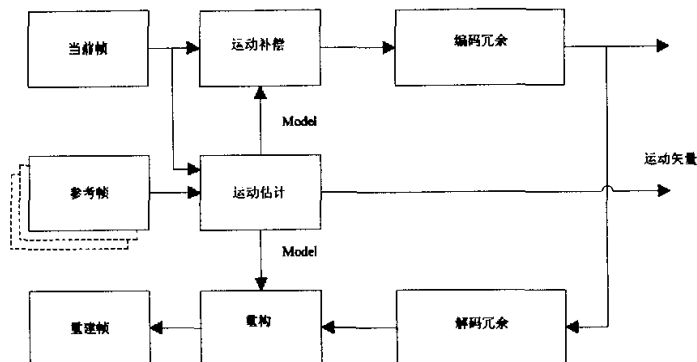


图 2.4 运动估计

由于运动估计算法是整个编码过程中最耗时的一个模块，因此，它仍然是编解码算法中的一个研究热点。常用的运动估值技术主要有像素递归法和块匹配法。由于基于块匹配的运动估计（Block-based Motion Estimation）算法简单，并且易于在硬件上实现，已经成为最常用的运动估计算法，并被多种标准所采纳。但是，运动估计算法中仍然有许多问题需要解决，例如如何选择模式，如何提高搜索效率和精度等。

● 变换编码

变换编码的主要目的是去除图像的空间冗余度。空间冗余度在图像中是非常大的。人们早就发现，在同一副图像中，相邻的像素点之间有着极强的相关性。由于能量分布在了图像的各个区域，从而使得图像压缩很难实现。变换编码的目的，即使用一种合理的转换方法将原始图像转换到转换域（Transform Domain），使图像能量得以集中，降低其空间冗余度，方便编码压缩。

常用的变换编码方法有离散余弦变换（DCT）和离散小波变换（DWT）。DCT 主要应用于小的、有规则的图像区域，例如 8×8 块；DWT 则主要应用于较大的图像区域。DCT 变换是目前应用最广泛的变换编码方法，被应用于多种常用的视频编解码标准中；而 DWT 由于在静态图像编码上的性能要优于 DCT，因此，在 JPEG-2000 等静态图像编码标准中也被采纳。

对于序列图像来说，主要采用 DCT 变换。简单的 DCT 变换示例可见图 2.5。

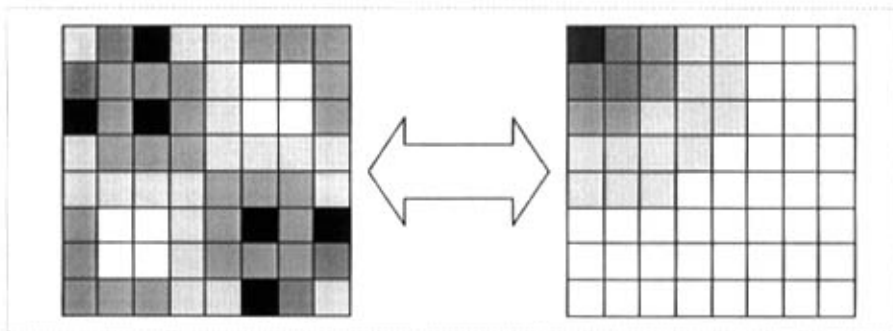


图 2.5 离散余弦变换（DCT）

● 量化

量化的目的是去除经变换编码后的视频分量中的那些不重要的分量。量化的原理很简单，量化器就是把一个在值域 X 的信号量化减为到较小的值域 Y 中。和原来 X 值域中的信号相比，要保证在值域 Y 中使用更少的位数来表示信号，这样才能起到压缩数据的目的。如图 2.6，通过比较源数据 Src 和反量化后的数据 Dst ，可以得到两者之间的差值 Err 。因此，量化过程是一个有损压缩的过程。鉴于此特征，往往可以通过控制量化阶（Scale Factor 或 Step Size）来调节对图像的压缩是精细还是粗糙。当量化阶较大时，反量化恢复后的图像往往比较粗糙，但是压缩比较高；反之，量化阶较小时，由于保留了更多的残差分量，则恢复后的图像质量较好。

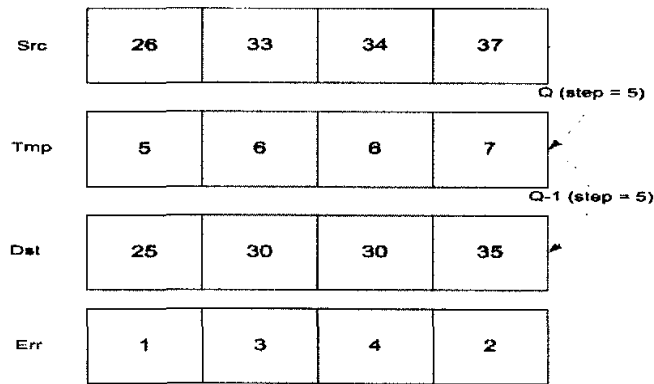


图 2.6 量化

● Zigzag 扫描

经 DCT 变换和量化后的残差系数，大部分都为零，只有少数几个非零系数，这些系数主要集中在图像块的左上角。因此，只需对这些非零系数编码即可。由于量化后的系数一般采用游程编码，因此，如果能够将大部分的零系数集中到一块，则更利于编码。因此，常用的扫描方式为之字形（Zigzag）扫描，如图 2.7 所示。

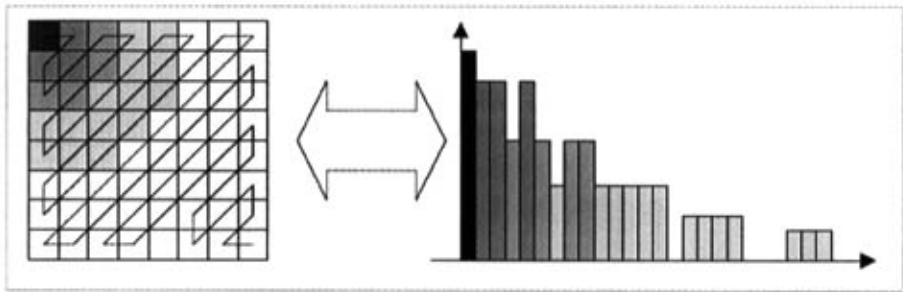


图 2.7 Zigzag 扫描

● 熵编码

熵编码把一系列用来表示视频序列的元素符号转变为一个用来传输或是存储的压缩码流。熵编码的基本原理是将那些常用的符号用短码字编码,而将那些不常用的符号用长码字编码。视频编解码过程中,常用的熵编码方法有哈夫曼编码和算术编码。

2.1.2 视频编解码相关标准演化

目前,大部分的视频编解码器都是基于某个国际公认的视频编解码标准的。国际标准化组织 (ISO) 和国际电信联盟 (ITU) 分别制订了一系列的标准,这些标准大大推进了视频编解码技术的发展。每个视频编码标准实际上都是常用视频编解码技术的一个子集,并且往往面向某一类型的应用。由 ISO 成立的联合图像专家组 (JPEG) 主要制订面向静态图像的压缩标准,主要包括 JPEG[4]和 JPEG-2000[5];由 ISO 和 IEC (International Electro-technical Commission, 国际电工委员会) 共同成立的移动媒体专家组 (Moving Picture Expert Group) 制定的标准主要针对视频数据的存储、广播电视和视频流的网路传输的应用,其标准以 MPEG-x 命名,主要包括 MPEG-1[6]、MPEG-2[7]、MPEG-4[8]、MPEG-7[9]、MPEG-21[10];由 ITU 视频编码专家组 (VCEG) 制定的标准主要针对实时视频通讯的应用,如视频会议和可视电话等,其标准以 H.26x 命名,主要包括 H.261[11]、H.263[12]、H.263+[13]、H.263++[14]和 H.264[15]。自 2001 年起,ISO 和 ITU 联合组建了联合视频组 (JVT),着重将 H.26L 草案发展成为一个国际性标准,并有了 H.264/AVC (MPEG-4 Part 10)。此外,由我国信息产业部支持的具备自主知识产权的第二代信源编码标准《信息技术—先进音视频编码》(AVS) [16]目前也已出台。这些标准都是传统的或者当前的主流视频编码标准,其整体演化过程可参见表 2.1。在 2.1.2.1, 2.1.2.2 和 2.1.2.3 分别对各个标准做了简单介绍。

表 2.1 视频编码标准

标准	主要应用	制订年份
JPEG/JPEG-2000	图像 (Image)	1992-1999/2000
JBIG	传真 (Fax)	1995-2000
H.261	视频会议 (Video Conferencing)	1990
H.262/H.262+	标清数字电视 (DTV/SDTV)	1995/2000
H.263/H.263++	视频电话 (Videophone)	1998/2000
MPEG-1	VCD (Video CD)	1992
MPEG-2	标清/高清数字电视、DVD (DTV, SDTV, HDTV, DVD)	1995
MPEG-4	交互式视频 (Interactive Video)	2000
MPEG-7	多媒体内容描述接口 (Multimedia Content Description Interface)	2001

MPEG-21	多媒体框架 (Multimedia Framework)	2002
H.264/MPEG-4 part 10	AVC (Advanced Video Coding)	2003
	Fidelity Range Extensions (High Profile), Studio editing, Post processing, Digital cinema	2004 August
AVS	高分辨率数字广播、高密度激光数字存储媒体、无线宽带多媒体通讯、互联网宽带流媒体	2006

2.1.2.1 JPEG 和 MPEG

JPEG 的全称是 Joint Photographic Experts Group (联合图像专家组), 由 ISO 成立, 主要致力于静态图像压缩编码, 推出的标准主要有 JPEG 和 JPEG-2000。MPEG 的全称是 Moving Pictures Experts Group (即活动图像专家组), 由 ISO 与 IEC (International Electronic Committee) 于 1988 年联合成立, 致力于运动图像 (MPEG 视频) 及其伴音编码 (MPEG 音频) 标准化工作。MPEG 前后共推出了多个版本, 有 MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-21。各个流行的标准分别介绍如下:

● JPEG 和 JPEG-2000

JPEG 标准是国际上, 彩色、灰度、静止图像的第一个国际标准。JPEG 标准不仅适于静图像的压缩; 电视图像序列的帧内图像的压缩编码, 也常采用 JPEG 压缩标准。JPEG 标准定义了两种基本压缩算法: 一是基于 DCT 变换有失真的压缩算法; 二是基于空间预测编码 DPCM 的无失真压缩算法。JPEG-2000 是 JPEG 标准的改进版, 由于使用了离散小波变换 (DWT), 其压缩性能比原来的 JPEG 标准要好。

● MPEG-1

MPEG-1 制定于 1992 年, 为工业级标准而设计, 是一个面向数字存储的运动图像及其伴音的编码标准, 它完成的基本任务就是质量适当的图像 (包括伴音) 数据必须成为计算机数据的一种, 和已有的数据 (如文字、绘图等数据) 在计算机内兼容, 并且这些数据必须在现有的计算机网络和广播电视等通信网络中兼容传输。MPEG-1 标准包括三个组成部分: MPEG 视频、MPEG 音频、MPEG 系统, 因此, 它涉及的问题包括视频压缩、音频压缩及多种压缩数据流的复合和同步问题。MPEG-1 标准可适用于不同带宽的设备, 如 CD-ROM、Video-CD、CD-I 等。MPEG-1 适用于比特率 1.5Mbps 的传输。

● MPEG-2

MPEG-2 制定于 1995 年, 其设计目标是高级工业标准的图像质量以及更高的传输率。MPEG-2 的编码码流分为 6 个层次, 从上至下依次为: 视频序列层 (Sequence); 图像组层 (GOP: Group of Picture); 图像层 (Picture); 像条层 (Slice); 宏块层 (MacroBlock) 和像块层 (Block)。MPEG-2 所能提供的传输率在 3-10Mbits/s 间, 其在 NTSC 制式下的分辨率

可达 720×486, MPEG-2 能够提供广播级的视像和 CD 级的音质。除了作为 DVD 的指定标准外, MPEG-2 还可用于为广播, 有线电视网, 电缆网络以及卫星直播 (Direct Broadcast Satellite) 提供广播级的数字视频。

● MPEG-4

MPEG-4 标准于 2000 年制订, 提供了一个基于对象的低码率视频压缩编码标准, 旨在为视(音)频数据的通信、存取与管理提供一个灵活的框架与一套开放的编码工具。MPEG-4 与传统的 MPEG-1、MPEG-2 标准最大的区别是基于对象的编码, 即加入了视觉对象 (Visual Object) 和音频对象 (Audio Object), 这一特性为交互性提供了有力的底层机制支持, 也为在压缩域对图像或视频的 VO 内容进行灵活的表示和管理提供了有利条件。基于 MPEG-4 的特性, 它具有基于内容的交互性、高压缩率和灵活多样的存取模式等优点。

● MPEG-7 和 MPEG-21

与 MPEG-1、MPEG-2 和 MPEG-4 不同, MPEG-7 和 MPEG-21 关心的重点不是视频编解码过程中的编码和压缩技术。MPEG-7 的目标是提供对多媒体内容描述的接口, 使得日渐庞大的多媒体信息得到管理, 并能迅速搜索; MPEG-21 则旨在提供一种多媒体框架, 使多媒体资源在大范围的网络上能够实现透明传输和充分利用。

2.1.2.2 H.26x

ISO 的 MPEG 系列标准主要是面向视频数据的存储和分发, 而 ITU 以往则主要致力于电信传输, 因此, 其开发组 VCEG 开发的一系列标准 H.261、H.263 和 H.26L (H.264 的前身) 分别面向实时 (Real-time)、点对点 (Point-to-point) 和多点 (Multi-point) 的传输。

● H.261

H.261 于 1990 年发布草稿, 是 ITU-T 发布的第一个图像压缩标准。H.261 是专门为综合业务数字网 (ISDN) 开展可视电话、视频会议而制定的, 主要用于双向视频通信, 速率为 64Kbps 的整数倍。在 H.261 中, 输入的图像格式有两种, 为 CIF (352×288) 和 QCIF (176×144), 帧率为 30、15、10、7.5 四种, 隔行扫描, 宽高比为 4: 3。H.261 是基于 DCT 和 DPCM 的预测编码算法, 其每帧图像分成图像层、宏块组 (GOB) 层、宏块 (MB) 层、块 (Block) 层来处理。

● H.263、H.263+和 H.263++

H.263 是在 H.261 基础上发展起来的, 它是 ITU-T 为低于 64kbps 的窄带通信信道制定的视频编码标准。与 H.261 相比, H.263 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。H.263 提供了比 H.261 更好的压缩性能以及灵活性。

H.263+和 H.263++分别是 H.263 标准的第 2 版和第 3 版, 它们分别为 H.263 提供了更多的新特性。H.263+提供了 12 个新的可协商模式和其他特征, 进一步提高了编码效率和某些

其他方面的性能。H.263++则又在 H.263+的基础上增加了选项 U、选项 V 和选项 W 这 3 个选项，分别是为了增强型参考帧选择，数据分片和在 H.263+的码流中增加补充信息。H.263++增强了码流在恶劣信道上的抗误码性能，同时提高了编码效率。

● H.264/AVC (MPEG-4-Part 10)

H.264 是由 ISO/IEC 与 ITU-T 组成的联合视频组 (JVT) 制定的新一代视频压缩编码标准，其目标是制定一个新的视频编码标准，以实现视频的高压缩比、高图像质量、和良好的网络适应性。在 ISO/IEC 中该标准命名为 AVC (Advanced Video Coding)，作为 MPEG-4 标准的第 10 个部分；在 ITU-T 中正式命名为 H.264 标准。H.264 具有如下特点：1) 比之前所有的标准的压缩效率都要高；2) 简单的语义规范；3) 与当前所有的协议都能实现无缝整合；4) 更好的容错机制；5) 更广的应用场景，例如广播视频、流媒体、视频会议、数码相机和高清电视等；6) 具有良好的网络兼容性；7) 能够在编码效率和实现复杂度之间得到一个均衡。此外，为了适应不同的场景和应用需要，引入了四个版本 (Profile)，分别为：基本版 (Baseline Profile)、主要版 (Main Profile)、扩展版 (Extended Profile) 和高保真版 (High Profile)。

2.1.2.3 AVS

AVS 是由数字音视频编解码技术标准工作组 (Audio Video Coding Standard Workgroup of China) 负责研究的我国具备自主知识产权的第二代信源编码标准。AVS 标准是《信息技术—先进音视频编码》系列标准的简称，包括系统、视频、音频、数字版权管理等四个主要技术标准和一致性测试等支撑标准。AVS 工作组的任务是：面向我国的信息产业需求，联合国内企业和科研机构，制 (修) 订数字音视频的压缩、解压缩、处理和表示等共性技术标准，为数字音视频设备与系统提供高效经济的编解码技术，服务于高分辨率数字广播、高密度激光数字存储媒体、无线宽带多媒体通讯、互联网宽带流媒体等重大信息产业应用。

2.2 压缩视频在网络上传输

随着全球信息化时代的到来，人们了解事物、交换信息的途径已经从纸、笔、书本、语音等发展到通过声光电信号等各种方式更准确、更快捷、更丰富地表达出来。在需求的推动下，多媒体计算机技术与通信技术相结合，逐渐发展成为一种新的边缘技术——多媒体通信技术。如果说 19 世纪是电报的时代，20 世纪是电话的时代，那么，即将到来的 21 世纪将是多媒体通信的时代。多媒体通信技术的发展状况在很大程度上昭示了当前信息技术发展的现状。多媒体通信业务主要包括以下几种类型：多媒体会话型业务，如会议电视、可视电话、远程教育、远程医疗等；多媒体检索型业务，如多媒体数据库查询、视频点播 (VOD)；多媒体分配型业务，如音频视频广播等；多媒体电子信函型业务等。其中，会议电视作为多媒

体会话型通信业务的典型,已在社会性的信息交流中发挥了巨大的沟通作用。它能为用户提供直接、全面的沟通交流,并能节约时间、降低成本、提高生产率。

2.2.1 目前常见的网络系统及其视频应用

由于多媒体通信的巨大市场需求,在研究各电视会议系统的基础上,ITU-T 制订了视听多媒体通信系统国际标准一系列建议,其发展主要经历了模拟会议电视、数字会议电视和国际统一标准的数字会议电视三个阶段。尤其是 90 年代中期以来,计算机互连网的飞速发展对电信业产生了巨大而深刻的影响,基于包交换网络(PBN, Packet Based Networks)的多媒体通信系统逐渐成为研究人员和电信、网络及计算机厂商关注的焦点之一。Internet 工程任务组(IETF)也开发出实时传输协议(RTP)、资源预留协议(RSVP)等与 ITU-T 的 H.32x 建议配合使用的底层多媒体传输控制协议,以保证在 IP 网络上传送实时音视频流。

ITU-T 制定了 H.32x 多媒体通信系列标准建议族,主要有 H.320[17]、H.321[18]、H.323[19] 和 H.324[20]等。目前常用的视频传输标准建议族可参见表 2.2。分别介绍如下:

表 2.2 视频传输标准建议族

应用和标准族	分发协议	使用的编码标准	视频比特率	包大小	差错特性
ISDN 视频电话 (H.320)	H.221	H.261 和 H.263	68-384kb/s	N/A	无差错
纯 ATM 视频会议 (H.310, H.321)	H.222	MPEG-2	1-12Mb/s	53 bytes	接近于无差错
有线/卫星电视	H.222	MPEG-2	6-12Mb/s	N/A	接近于无差错
包交换网络的视频 电话(H.323)	H.225/RTP/ UDP/IP	H.261, H.263, MPEG-2	10-1000kb/s	≤ 1500 bytes	BER=0,0-30% 丢包率
PSTN 视频电话 (H.324)	H.223	H.263	20kb/s	100 bytes	极少的 bit 错 误和包丢失
移动视频电话 (H.324M)	H.223 移动 扩展部分	H.263	10-300kb/s	100 bytes	BER= 10^{-3} - 10^{-5} , 有丢包

● 面向 ISDN 的 H.320

H.320 标准于 1990 年制订,是视频会议的早期建议之一,是关于在速率从 56Kbps 到 2Mbps 的 ISDN 和交换的 56Kbps 电路上进行视频会议的标准。由于窄带 ISDN 网是一种基于电路交换的网络,所以 H.320 标准主要满足和适应电路交换的特性。由于底层的分发协议采纳了 H.221[21],因此 H.320 系统提供了无差错的传输信道。H.320 标准获得了巨大的成功,其推荐的视频会议标准被广泛用于 VSAT、DDN、ISDN 等电路交换网络。

- 面向有线电视和卫星传输的 MPEG-2 传输流

这种传输环境是传统广播电视系统中最常用的传输环境。无论底层网络是有线的还是无线的, 这种传输系统都能提供一种近似无差错的环境。

- 面向 IP 网络 (或 Internet) 的基于 IP/UDP/RTP 协议的实时传输

在这种传输环境中, 主要包括两层的高级协议, 一为 H.323 协议族, 二为会话初始化协议 (SIP) [22] 和会话描述协议 (SDP) [23] 等。结合底层实时传输协议 (RTP) [24] 提供的反馈机制, 在编码端可以动态地调节码率和差错控制机制。而在解码端, 可以结合利用 RTP 协议提供的信息动态检测当前流传输状况, 以便检测错误并实施差错隐藏方法。

- Internet 流媒体规范

基于 Internet 流媒体规范的应用程序主要有 RealPlayer、QuickTime 等, 它们均允许几秒的延时。这种机制使得重传机制得以实施, 因此, 在这种环境下, 也几乎可以达到无差错的传输。

- 面向公用电话交换网 (PSTN) 的 H.324

PSTN 是目前最常用的基于点对点和电路交换的传输网络, 其带宽为 33Kbps, 并具有极小的延时。在 H.324 中, 为视频、音频和其他数据定义了一整套共享高速连接的通用规范。在 H.324 中使用的 H.223[25] 分发机制, 保证了以相对低的延时来获得重传, 因此, 可以提供一实用性很强的近似无差错的信道。

- 无线网

无线网传输可以分为无线广域网和无线局域网传输, 两者使用的技术和标准是有所差别的。

目前无线广域网主要有 GPRS、W-CDMA 等传统无线网和新兴网络 3GPP[26]-[30]。3GPP 改变了传统无线网带宽低、可靠性差的缺点, 使得在无线网上视频应用的商用成为可能。对于电路交换的无线网来说, 3GPP 定义了一套基于 H.324M (H.324 标准的 Mobile 扩展部分) 的多媒体电话服务标准; 而对于基于 IP 包交换的传输应用来说, 则使用 SIP 和 SDP 等会话控制协议以及 RTP 媒体传输协议。

无线局域网也是视频应用的一个重要研究热点, 主要的协议标准是美国电气及电子工程师学会 (IEEE) 的 802.11b。由于 802.11b 是非面向连接的, 因此在一定次数的重传失败后, 数据包就会做丢失处理。因此, 在无线局域网上传输视频数据, 要着重关注随机包的丢失, 以及由于重新路由而造成的连续包的丢失处理。

2.2.2 视频传输服务质量控制

上一节分析了目前常见的网络系统。当视频数据在低比特率、易出错的信道上传输时, 如果不实施一定的控制, 则极易造成视频质量的严重下降, 从而使得接收端视频无法让人眼

接受。为了保证在接收端获得可以接受的视频质量，必须在压缩效率和服务质量（QoS）之间做一个权衡，通过牺牲一定的压缩率，来换得服务质量的保证。

基于一般实时视频传输的分析，QoS 控制主要针对以下几个方面：传输码率、图像失真度和传输延时。基于 QoS 控制考虑，衍生了多种算法和控制方法，最常见的有码率控制技术、视频转码技术和差错控制技术。

2.2.2.1 码率控制

原始的视频数据一般是固定码率，然而在经过视频压缩之后，则会变成非固定码率，且不同的编码算法会对码率的变化产生很大的影响。而在视频传输过程中，由于通信信道带宽有限，因此需对视频编码码率进行控制，从而保证编码码流的顺利传输和信道带宽的充分利用。实时编码码率控制方法（Rate Control）主要有两种：用先前宏块编码产生的比特数来预测当前宏块编码产生比特数，或者通过视频编码率失真函数来预测当前宏块编码产生的比特数。

码率控制算法的目的，就是通过动态调整编码器参数，使得在给定码率下高效的传输数据，保证良好稳定的接收端质量。它为视频序列中的图像组 GOP、图像或者子图像分配一定的比特数。现有的码率控制算法主要是通过调整离散余弦变换的量化参数（QP）大小输出目标码率。实际上，量化参数反映了空间细节压缩情况，如 QP 选择得较小，则大部分的细节都会被保留；如 QP 选择得比较大，则一些细节会丢失，从而使得码率降低，但图像失真度也会增大，造成图像质量下降。也就是说，QP 和码率成是一种反比的关系，并且随着源视频复杂度的增加，这种反比关系会更明显。

目前常见的码率可分为两种，分别是可变比特码率（VBR）和固定比特码率（CBR），相应的码率控制方法分别称为可变比特码率控制和固定比特码率控制。VBR 模式是一种开环处理，根据输入的 QP 值，使视频源根据实际视频序列中图像的复杂度而变化。图像越细、运动越大，则比特率也相应调大；反之亦然。CBR 模式是一种闭环处理，它是根据目标比特和源视频的复杂度估计、解码缓冲的大小及网络带宽估计动态调整 QP，从而得到符合要求的码率。

常见的基于量化参数 QP 来实施码率控制的方法可参见图 2.8。关于码率控制的细节介绍，在[31]中有细述。

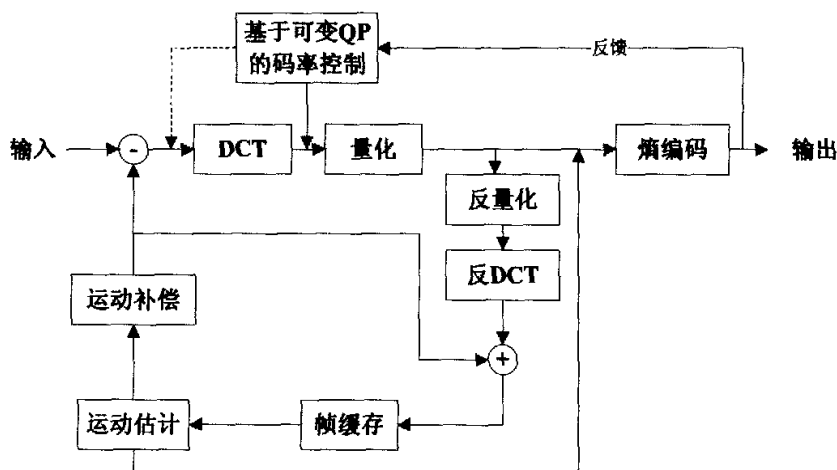


图 2.8 基于 QP 的码率控制

2.2.2.2 视频转码

由于多媒体应用和底层网络平台及其协议规范越来越多，不同标准之间的视频如何更好地传输成了一个研究的热点；此外，出于码率等方面的考虑，同种格式之间的视频往往也需要进行一定的转换。为了使得异构网络之间、不同种类的接入设备之间以及不同媒体数据格式之间能够无缝连接，常用的方法是通过一个中间器或者媒体网关（Media Gateway），通过转码（Video Transcoding）来使得不同标准得以兼容。

视频转码实际上是将压缩视频流转换成另一种视频流的过程，被转化的部分包括视频格式、视频大小、传输比特率或者简单语义，如图 2.9 所示。视频转码技术的目标是通过不需要完全的解码再编码过程，来完成视频流之间的转换。

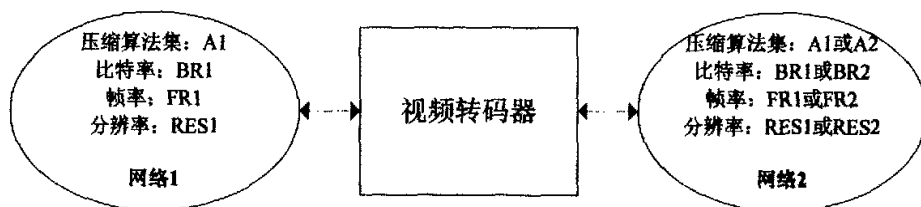


图 2.9 视频转码

同种格式之间的视频转码主要是通过调节 QP 值的方法获得的，类似于码率控制方法。实际上同种格式之间的转码往往也是出于降低码率的需要。常用的转码技术包括运动数据重用机制、运动数据重估计机制和运动优化机制。具体转码技术可参见文献[32]。

2.2.2.3 差错控制

当压缩视频在不可靠的网络（如 Internet 或无线网）上传输时，极易产生丢包或者比特

错误。在实时视频应用中，重传机制是不易实施的，从而极易造成视频质量的严重下降。误码对视频质量的下降不仅体现在当前帧，还会产生差错传播。如何更好地在视频压缩效率和鲁棒性之间做一个权衡，即是本文关注并需要解决的一个问题。在第三章中，将具体介绍视频传输差错控制机制（Error Control）。

2.3 本章小结

视频数据相对于其他数据类型来说，其信息量是巨大的。在有限的网络带宽下传输如此巨大的数据量，并且保证传输的服务质量，是本章的重点。本章主要分析了压缩视频在网络上传输的需求性及可行性，同时介绍了传输的发展状况，包括常见的网络系统及其视频的应用。最后，介绍了在压缩视频在网络上传输时要注意的几个方面，并分析了当前为保证服务质量所提出的几种常用方法，并由此引出了差错控制机制。

第三章 传统的视频压缩差错控制机制

第二章主要介绍视频编解码相关背景和技术，以及压缩视频的传输问题。从本章开始，着重介绍视频压缩差错控制机制。根据差错控制方法在编解码器中所处的位置，可以将差错控制机制分为三类，分别是：编码端的容错机制；解码端的差错隐藏方法；编解码、网络相互配合的差错控制技术。在本章中，将分别予以介绍。

3.1 差错控制机制简介

当数据包在服务质量（QoS）不能保证的网络上传输时，由于拥塞或者物理信道出错，往往会丢失或者发生错误。在这种情况下，如果仍然要保证数据的无差错传输，则只能使用重传机制，例如自动重复请求机制（ARQ）。然而，由 ARQ 引起的延时显然不能满足大部分实时应用的需求。因此，为视频编解码设计一种良好的容错机制，对于实时传输来说，是非常重要的。图 3.1 展示了当出现传输差错时的情景。

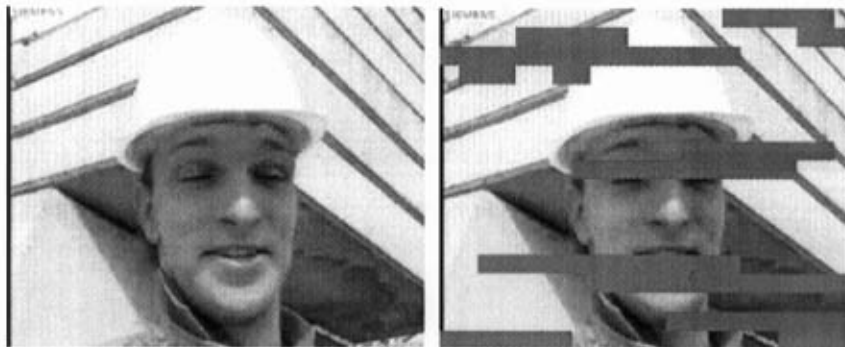


图 3.1 差错对压缩视频的影响

差错控制对于视频传输来说富于巨大的挑战性。首先，由于在视频压缩中往往使用了预测编码算法和可变长编码（Variable Length Coding, VLC）算法，使得压缩视频流对于传输错误来说非常敏感。基于时间和空间的预测机制，会使得一个样点的错误不仅影响到当前帧，也直接影响到后续帧；同样地，可变长编码使得一个比特的错误引起后续比特均无法解码。其次，由于视频源和网络的千变万化，很难设计出一种适合各样环境的最优方案。最后，考虑到实时应用的需求，在编码端和解码端采取的控制操作复杂度不能太高。

为了使压缩视频比特流具有容错特性，必须在视频流中加入一定的冗余信息，用来检测差错和纠错。这些冗余信息既可以加在源编码器（Source Coder）中，也可以加在信道编码器（Channel Coder）中。传统的仙农信息论指出，只要使得源编码器的码率低于信道编码器的码率，就可以对源编码器和信道编码器单独设计。基于该理论，可以设计一种理想的源编

码器和信道编码器的组合,即源编码器着重于最大限度地压缩视频数据,而信道编码器则通过前向纠错加入一定的冗余来保证压缩视频的传输和容错能力。然而,这种理想的方式在实践中是不可行的,它往往会造成无限的延时。因此,结合源编码和信道编码的混合编码器是一个更加明智的选择。目前所有的容错编码方法都是基于混合编码器的,通过有目的地牺牲一点编码效率来获得较强的抗误码能力。

尽管由于信道的出错,一个样点或者一个数据包可能丢失或者出错,但是,由于时间相关性和空间相关性,在解码端,仍然可以通过出错样点的相邻样点(包括空间相邻和时间相邻)来近似恢复。这种在解码端恢复差错的方法被称为差错隐藏(Error Concealment, EC)。这种方法之所以可行,是因为实际使用的编码器往往不能完全消除信号之间的冗余。例如,编码器往往会周期性地重启预测过程(加入1帧),这样使得即使在一段预测过程中产生了错误,也不会传播到下一段预测过程中去。常用的差错控制方法有时间插值法(Temporal Interpolation)和空间插值法(Spatial Interpolation)。在编码端通过有目的地加入少量冗余信息或改变编码方法,也会使得差错隐藏机制更加行之有效。例如,在编码端,对于相邻的样点或块来说,可以通过隔行的方式打包传输,这样,在解码端,即使由于丢包而产生差错,也极易通过相邻样点插值的方法来恢复和隐藏。纯差错隐藏的方法不会在编码端做任何的改变,只会在解码端增加一定的复杂度。如果接收端是移动设备,太复杂的差错隐藏方法也是不能用的。差错控制机制示意可参见图3.2。

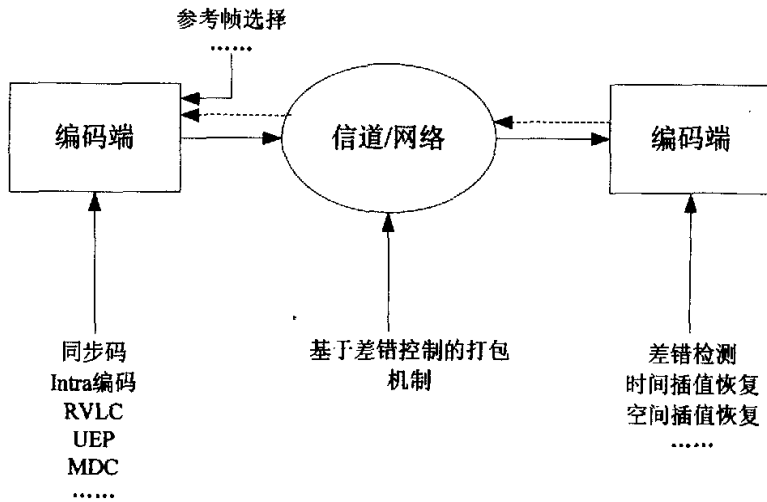


图 3.2 差错控制机制

3.2 差错检测

在解码端,在进行差错隐藏之前,必须先确定接收数据是否出错,如果出错,则在哪里出错了。这就是差错检测的方法。差错检测的方法也可以分为三类:一是在传输层通过编码

端和解码端相互配合的方法,二是通过在源编码器中加入冗余信息,三是只在解码端做的差错检测方法。下面分别予以介绍。

● 网络传输层差错检测

通过传输层来检测差错的方法是一种传统的方法。其中一种常用的差错检测的方法,是在编码端加入头信息。例如,在基于包的视频传输中,编码码流均被打包成一个个包传输,每个包都有其头信息和载荷类型信息。在头信息中,一般都有一个用来指示传输顺序的序列号,在解码端,可以通过该序列号检测出差错。该方法不仅适用于视频流的差错检测中,也可以用于其他类型数据流的差错检测。在 H.223 标准中,即使用该方法来检测差错。另一种常用的差错检测的方法,是使用前向纠错 (Forward Error Correction, FEC)。在这种方法中,纠错码被附加在编码后的码流中。在解码端,通过解码出的纠错码,就可以检测并纠正差错。在 H.261 中,为每个视频帧增加了一个 18 位的前向纠错码,该纠错码就可以用来进行差错检测和纠正[11]。

● 基于源编码器冗余信息的差错检测

在源编码器中加入一定的冗余信息,也可以用来进行差错检测。Lam 和 Reibman 提出了一种在频域检测差错的方法[33]。在这种方法中,每条块扫描线后都加了一同步码 (Synchronization Code)。当在解码端检测到同步码时,即将当前已解码出的块数目与一预值做比较。假如两者的值有出入,则可以断言发生了错误。出错块的位置可以通过计算带权均方差 (Weighted Mean Square Error, WMSE) 来判定 (通过计算当前扫描线与前一扫描线之间的 WMSE 值)。

数字水印技术是一种新兴的媒体研究,应用也非常广泛。数字水印除了可以用来进行信息隐藏之外,还可以用在差错检测上。在[34]中,设计了一种基于碎片水印 (Fragile Watermark) 的差错控制方法。在编码端,碎片水印被加在量化后的 DCT 系数上;当在解码端发现水印非完整无缺时,即能够确定产生了差错,并且能够在宏块级 (MB-level) 快速定位差错,这对于差错隐藏来说是非常有用的。

● 解码端差错检测

在解码端做的差错检测可以分为基于 VLC 码表的差错检测、基于语义的差错检测、基于视频信号特性的差错检测。

由于在大部分情况下,在一个编码器中使用的码字都不是可用码字的最大集合,而由于比特错误,往往会出现码表里面没有的码字。因此,在解码端,一旦发现与码表里面不匹配的码字,则可以马上断言差错。

基于语义的差错检测主要是根据解码端解码出的某些参数来进行的。例如,当出现量化阶为 0,或者解码出的 DCT 系数个数大于可能出现的最大个数时,都可认为是发生了差错。

此外,视频信号的某些自然特性的,例如空间相关性和时间相关性等,也可以用来检测差错。在脉冲调制编码 (Pulse Code Modulation, PCM) 和差分脉冲调制编码 (Differential Pulse

Code Modulation, DPCM) 中, 相邻行之间的像素差值可以用来检测差错[35][36]。一旦差值大于某个阈值, 则认为当前图像区域发生了错误。

3.3 差错控制方法

通过前面的分析可以得到, 传统的差错控制方法主要可以分为前向容错编码 (FEC)、差错隐藏 (EC) 和交互式抗误码技术 (Interactive Error Concealment)。许多方法既可以单独使用, 也可以结合使用, 从而达到更好的抗误码能力。传统的差错控制技术可以参见文献 [37]-[39]。由于不同的标准使用了不同的算法集, 因此, 基于不同算法集, 又可以为各个标准设计不同的差错控制方法。最新的标准在设计时都考虑了容错特性, 例如 H.263、MPEG-4 和 H.264, 基于标准的容错性研究可见[40]-[46]。本节中的各小节部分则分别介绍各种类型的差错控制方法, 与标准相关的各种差错控制方法也在其中予以介绍。

3.3.1 前向容错编码

容错编码的目的, 是为了使得传输错误不会造成接收端获得一个无法接受的视频。显然, 具有容错特性的编码器在编码效率上会有所降低, 但是, 当传输流出现错误时, 它的好处却时明显的, 即它保证了在接收端获得一个可接受的视频质量。在编码端加入的用以检测差错或者纠错的信息被称为冗余信息 (Redundancy)。带容错特性的编码器的设计目的, 就是为了以最少的冗余代价, 来尽可能大地获得容错功能。

3.3.1.1 视频传输中的前向纠错

视频传输中前向纠错的方法 (FEC in Video Communications) 主要是使用一定的冗余信息, 来降低出错时的影响。冗余信息的加入会造成比特率的增加, 因此, 在带宽受限的情况下, 往往是以降低编码的比特率来实现的。这样, 会造成在无差错情况下视频质量的下降。为了在无差错视频质量和抗误码性能之间做一个较好的权衡, 比较可行的办法是使得编码率自适应地随着网络状况而变化。RCPC (Rate-compatible Punctured Codes) 和 CRC (Cyclic Redundancy Check) 是常用的前向纠错方法[40]。RCPC 可以使得码率随着网络状况动态调节, 但由于其延时较大, 因此在实时传输中不易使用。CRC 即循环冗余校验, 通过加入一定的校验码, 来获得对差错的检测和修复。

3.3.1.2 数据分割和非平衡保护方法

由于压缩视频数据各部分对差错的敏感度不同, 因此可以将其分成几部分。通过对不同

部分实施不同程度的差错控制（对差错敏感的部分，实施较强的保护；反之，则实施较弱的保护），可以提高抗误码能力。传统的视频分割技术将压缩视频数据分为两部分：一部分是对差错较为敏感的，其中包含了每个视频分组（VOP）的轮廓信息和运动信息（MV 等）；另一部分是对差错敏感相对较低的纹理数据（Texture Data）。分割两个部分的同步码（Separation Marker），在帧间视频帧组中称为运动标志，在帧内视频帧组中称为 DC 标志。当差错不敏感的纹理数据部分发生差错时，视频解码器可存储视频分组中无差错的运动和轮廓数据。通过差错隐藏技术，可以成功隐藏纹理部分的差错，仅会有轻微的视觉失真。由于纹理数据占据了压缩视频数据的主要部分，而头信息和运动信息数据量相对较少，这使得非平衡保护方法的实施变得可行。数据分割的方法可见图 3.3。



图 3.3 数据分割方法

基于数据分割的差错保护是非常有效的，在 H.263++、MPEG-4 和 H.264 中都可以采纳该方法来增强容错性。在这些标准中，将序列头信息、片信息、宏块头信息（MB Header），MV 和 DCT 系数等分开传输[47][48]；在 H.264 中，甚至可以分成更多的部分，例如以往一个片，还可以更细分为区域 A、B 和 C，分别对应头信息、Intra 信息和 Inter 信息。在表 3.1 中，给出了 H.264 中的分块方法以及发生错误时推荐使用的隐藏方法。对重要的信息，如头信息和 MV，可以实施一些附加的保护手段，例如 RVLC 等。

表 3.1 H.264 中的分块方法

可用分割	隐藏方法
AB	使用 A 中的 MV 和 B 中的纹理信息；可以使用 Intra 恢复方法
AC	使用 A 中的 MV 和 C 中的 Inter 信息；可以使用 Inter 纹理恢复方法
A	只能使用 A 中 MV 来恢复
B、C 或 BC	丢弃 B 和 C 部分。使用空间插值方法得到 MV，然后进行差错恢复

在一幅图像中，有时人们只对其中的一块区域感兴趣，这块区域被称为感兴趣区域（Region of Interest, ROI）。由于人们往往只关心这些感兴趣的区域，而其他区域则作为衬托或背景，因此，这些区域数据的顺利传输尤为重要。ROI 的保护方法有很多，其中一种通过前处理和后处理（Pre- and Post- Processing）来保护 ROI 的非线性转换方法是非常有效的方法[49]。

3.3.1.3 基于非平衡保护的层次编码

在层次编码（Layered Coding）中，一般将视频编码成多个层，其中有一个基本层（Base Layer）和多个增强层（Enhancement Layer）。基本层提供了一个虽然较差但仍然可接受的视

频质量，各个增强层逐个提高视频质量，越多的增强层可用，则可获得越好的视频质量。基于容错考虑，LC 必须与 UEP 配合使用，在基本层中传输对差错较敏感的数据，而在增强层中，则分别传输对差错较为不敏感的数据。基于非平衡保护的层次编码示意图可见图 3.4。

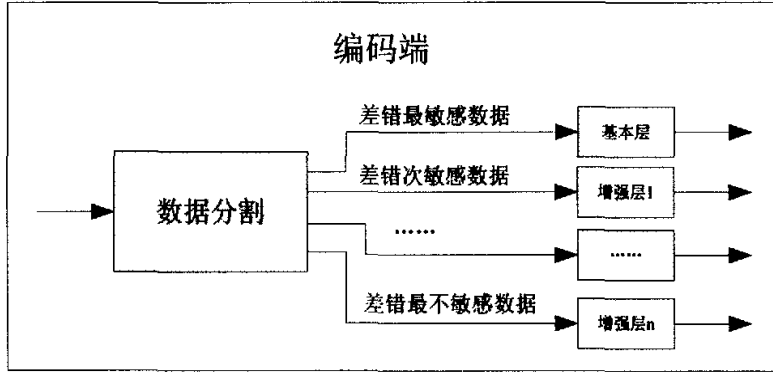


图 3.4 基于非平衡保护的层次编码

如何划分层次，对于层次编码来说也是非常重要的。常用的划分方法有时间亚采样、空间亚采样、频域亚采样、混合亚采样等。

3.3.1.4 多描述编码

与 LC 相似的是，在多描述编码（Multiple Descriptions Coding, MDC）中，也是将源视频流分为几个流来传输，分出来的每个流都是一个描述（Description）。但是与 LC 中分层次划分流的思想不同，在 MDC 中，每个描述都具有同样的重要性。同时，每个描述都能够单独解码，并且能够提供一个可以接收的视频质量。当接收到的描述越多时，视频质量就越高。由于要保证每个描述都能单独解码，因此那些基本的信息都必须包含在所有的描述中，例如序列信息，帧组信息等。从而可以知道，MDC 增加了比 LC 更多的编码冗余。但是，MDC 比 LC 好的一点是，它不需要底层网络提供一个可靠的子信道。这样，它比 LC 容易实现得多。典型的 MDC 编码方法可见图 3.5。在 MDC 中如何划分出各个描述的方法与 LC 类似，见文献[50]-[52]。

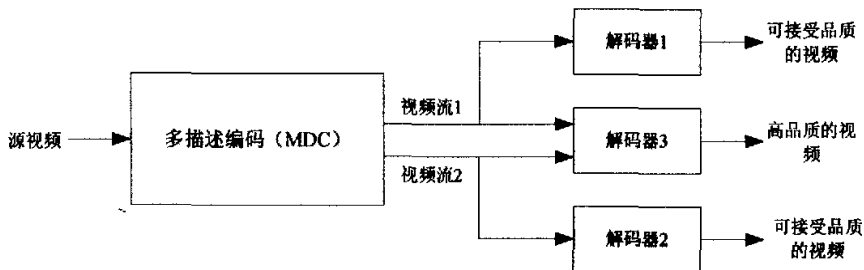


图 3.5 多描述编码

由于 MDC 具有良好的抗误码能力和易于实现的特性，已经越来越成为研究的热点。在

[53]中, 概述了近几年来 MDC 的研究状况, 并提出了许多新的 MDC 方法。

3.3.1.5 鲁棒熵编码

压缩视频流对差错敏感的原因之一就是由于使用了 VLC 来表示各种各样的符号。这使得任意一个比特的差错或丢失都会引起包括当前码字在内的后续码字无法解码, 即使后续码字没有错误。常用的鲁棒熵编码 (Robust Entropy Coding) 方法主要有三种:

1) 周期性插入重同步标志 (Inserting Resynchronization Markers), 见图 3.6。这种方法简单且有效。重同步标志必须能够非常容易地与其他码字相区别。那些重要的信息 (例如头信息等) 往往可以加在重同步字后面, 来得到保护。但是, 插入同步标志会降低编码效率。

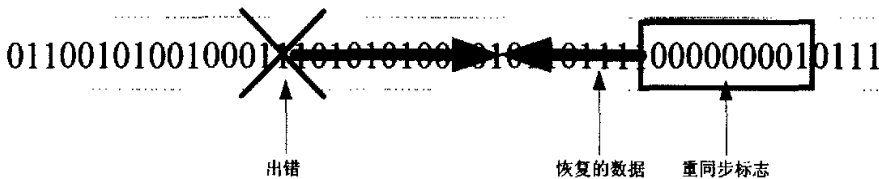


图 3.6 插入重同步标志

2) 可反向变长编码 (Reversible Variable Length Coding, RVLC)。以往, 一旦在比特流中检测到差错, 则直接将后面所有的比特都扔掉, 直到下一个同步字为止。这样, 即使后面的比特是正确接收到的, 也被丢掉了。一种用来克服这种问题的方法就是 RVLC, 它可以下一个同步字开始反向解码, 见图 3.7 所示。RVLC 还可以与前向 VLC 一起, 来增强熵编码的鲁棒性。

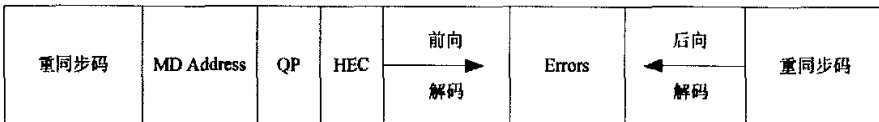


图 3.7 RVLC

3) 为基于语义的纠错做准备。由于在压缩视频码流中, 有许多受限制的语义。在解码端, 使纠错过的码流与正确的语义保持一致, 则使得差错能得到更好的恢复。

3.3.1.6 容错预测

压缩视频流对差错敏感的另一原因是由于使用了预测方法。一旦在解码端发生了错误, 则自出错帧开始, 在解码端使用的参考帧会变得与编码端的不同, 从而造成后续帧均存在差错。差错传播的影响可参见图 3.8。另一方面, 当使用空间预测时, 由于 DC 系数和 MV 的出错也会造成差错传播。

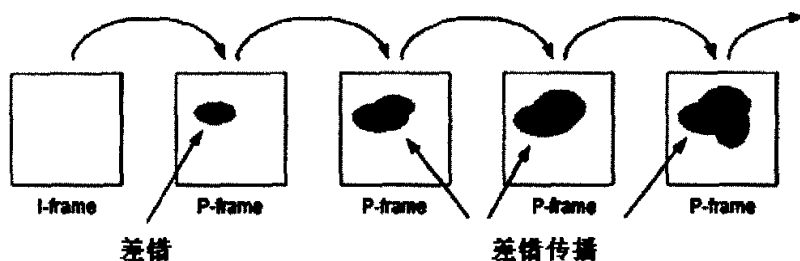


图 3.8 差錯傳播

为了抑制由时间预测产生的差錯傳播，往往周期性地在编码过程中插入帧内编码宏块（Intra Macroblock）或帧内编码帧（Intra Frame）。在实时应用程序中，由于延时限制，插入 I 帧的方法是不可行的。在这种情况下，往往使用插入 Intra 宏块的方法，见图 3.9。但这种方法有两个问题要考虑：一是可以 Intra 编码多少个宏块；二是对哪些宏块做 Intra 编码比较好。一种比较有效的方法，是使用自适应帧内刷新（Adaptive Intra Refresh, AIR）。这种方法最初是在 MPEG-4 中引入的，它周期性或随机地插入 Intra 帧，来抑制错误传播。在 H.264 中，对该方法做了改进，提出了一种基于码率失真优化的机制（Rate Distortion Optimization Scheme）[54]。它既可以根据当前网络状况选择合适的 Intra 编码宏块数，也可以选择那些最优的宏块进行编码，总之是使得如下的拉格朗日代价函数（公式 1）最小：

$$o^* = \arg \min_{o \in X} (D(o) + \lambda R(o)) \quad (1)$$

其中， o^* 是块编码模式（Inter 或 Intra 模式、块大小）， $D(o)$ 是由模式 o 引入的失真， $R(o)$ 是相应的编码率， λ 是拉格朗日系数[55]。

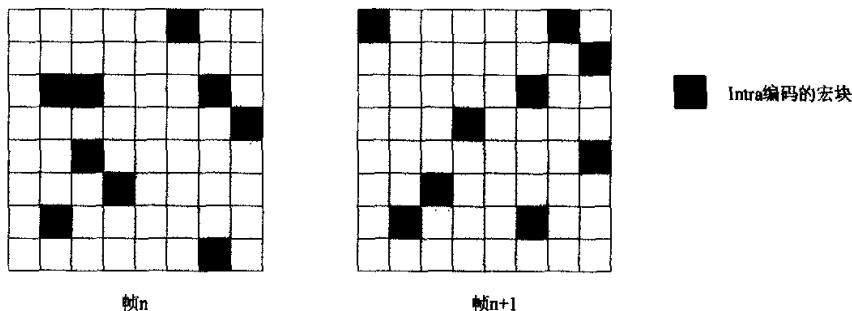


图 3.9 Intra 宏块刷新方法

另一种用来防止差錯傳播的方法是将数据域分为几个区域，每次进行时空预测时，都仅仅在相同的区域内进行。这样，可以使得一个区域内的差錯不会传播到另一个区域中去。简单地，可以将视频帧分为奇数序号组和偶数序号组，分别对应帧序号为奇数和偶数的帧。然后每个组内的帧只能使用同组中的其他帧进行预测。此外，还可以将一帧分成多个区域，例如块组（GOB）或片（Slice）就是这样的方法。在 H.264 中，还引入了一种可调宏块顺序法（Flexible Macroblock Ordering, FMO）。图 3.10 分别给出了两种简单的 FMO 方法。可以

看到，使用 FMO 方法，可以使得在解码端的差错恢复效果更好。

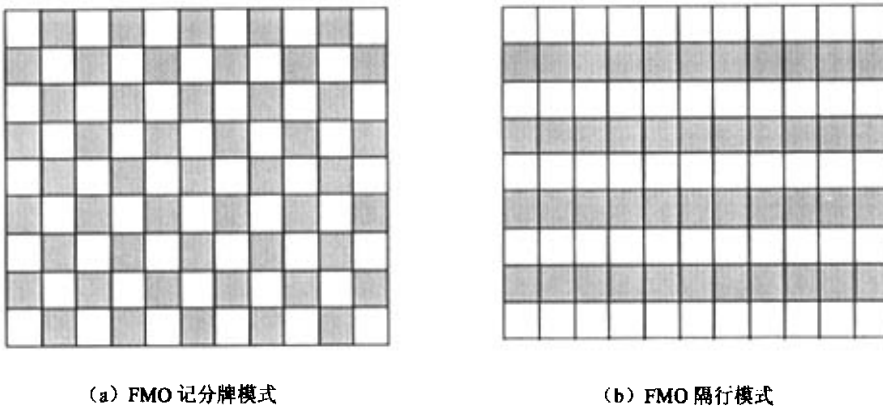


图 3.10 FMO

3.3.2 差错隐藏

差错隐藏是指在解码端通过估计的方法恢复由传输错误而丢失的信息。这些信息主要包括三个部分：纹理信息（Texture Information）、运动信息（Motion Information）和宏块编码模式（Coding Mode）。

鉴于自然图像信号具有邻接（时间和空间）像素值的平滑性，几乎所有的差错隐藏机制都是基于时间或空间插值技术来恢复纹理信息和运动信息的。对于编码模式来说，则主要是通过启发式的方法来恢复的。

3.3.2.1 纹理信息的恢复

纹理信息的恢复主要是通过时间预测和空间插值的方法来获得。

由于图像的运动性，前一帧图像中相同位置块并不一定是当前出错块的最佳匹配块，因此拷贝前一帧相同位置块的纹理信息并不是恢复的最佳选择。由于运动矢量（MV）指示了前一帧中的最佳匹配块，因此，使用 MV 所指向的块显然更加合适。这种方法被称为基于运动补偿的时间预测。图 3.11（a）和（b）分别给出相同空间位置块拷贝和基于运动补偿的块拷贝方法。此外，时间插值方法还往往与空间插值方法一起使用，来获得最佳差错隐藏效果。

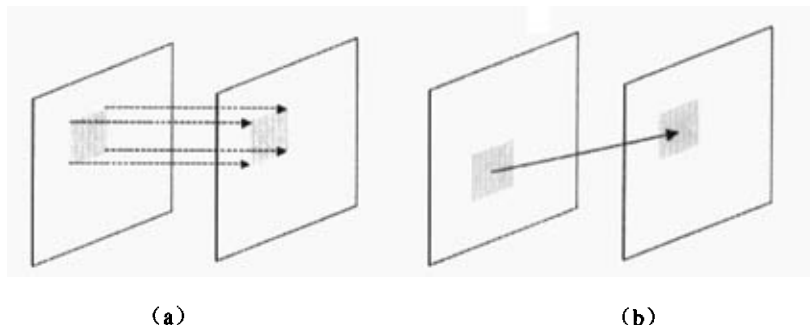


图 3.11 时间插值

空间插值的方法是使用出错块附近正确的块或像素来插值出错的像素点信息，见图 3.12。一般情况下，只有相邻像素点与当前块的相关性最大，因此使用相邻块边缘像素来插值也是一种比较合适的方法，它可以降低空间插值的复杂度。此外，还可以通过只预测 DC 系数来降低复杂度[56]。

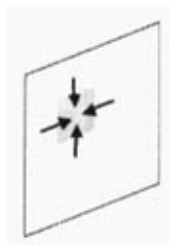


图 3.12 空间插值

常用的差错隐藏方法往往是结合了时间插值和空间插值的。例如，可以通过空间插值，估计出当前块的运动矢量，然后通过时间插值，来估计当前块纹理信息[57]。还可以结合场景切换的检测，来动态地选择时间插值方法或空间插值方法[58]。

凸集投影法（Projection onto Convex Sets, POCS）则是另一种通过空间插值的方法来达到差错隐藏的方法。POCS 利用了平滑块和带有明显边缘的块之间的区别，在变换域进行差错隐藏方法。首先由误码块和 8 个相邻块形成一个组合块，对组合块进行边缘检测，将其归为平滑块或者边缘块，边缘的方向被均匀量化为 $0^\circ \sim 180^\circ$ 中 8 个方向之一。然后定义两个算子，第一个算子是和边缘方向相关的带通滤波器，第二个算子是截断算子，使结果落在 $[0, 255]$ 中。将这两个投影算子反复作用到组合块上直到重建图像块稳定。

3.3.2.2 编码模式和运动矢量的恢复

前面对纹理信息的恢复中，许多方法都是假设编码模式和运动矢量是完好的。然而，这些信息也是可能丢失或出错的。对编码模式和运动矢量的恢复往往是通过周围块的相关信息来预估的。在[59]中，给出了面向 H.264 的模式估计方法。假如编码模式不容易估计，则可以使用 Intra 编码模式，然后使用空间插值的方法来隐藏错误。

对于运动矢量的估计来说,一般有以下几种方法[60]: 1) 假设丢失的 MV 为 0, 这对于运动很小的图像来说, 非常有效; 2) 使用前一帧中相应位置的 MV; 3) 使用空间相邻块的平均 MV; 4) 使用空间相邻块的中值 MV; 5) 重新估计 MV。

3.3.3 交互式抗误码技术

在 FEC 和 EC 中, 编码器和解码器都是单独工作的。在某些应用中, 还允许反馈信道。如果能够利用反馈信道传输一些反馈信息, 则可以在编码端动态地调节参数, 实施更有效地容错编码。自动重传 (ARQ) 是最简单的方法, 但是由于其可能造成无限延时, 而不适用于实时传输。但是, 如果在反馈信道中包含出错帧序号或出错的块信息, 则可以通过避免参考出错帧或使用 Intra 编码的方法, 来抑止差错的传播。使用 Intra 编码的方法显然代价太大, 因此, 当有可用的反馈信道时, 往往使用参考帧选择 (Reference Picture Selection, RPS) 的方法。传统的基于反馈信道的交互式抗误码技术在文献[61]中做了详尽的讨论。

3.3.3.1 参考帧选择

当编码端收到某个帧出错的反馈信息时, 在下一个 P 帧编码时, 可以不使用出错帧作为参考帧, 而使用更早的帧。虽然编码效率会有所降低, 但这能使得差错传播得到控制。当然, 这需要编码端和解码端都开辟一个较大的帧缓冲, 来存储多个参考帧。

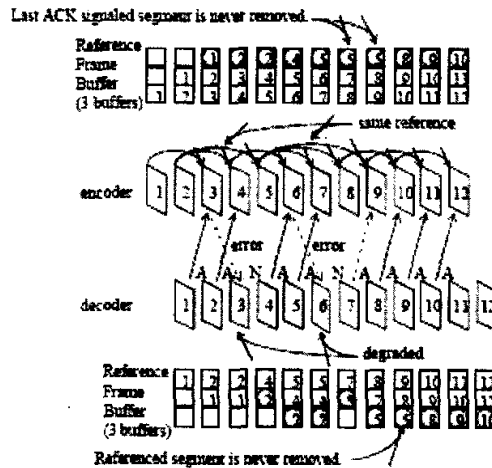


图 3.13 ACK

基于反馈信道的 RPS 方法可以分为两类, 一种是正确反馈响应 (ACK), 一种是出错反馈响应 (NACK) [62]。在 ACK 模式中, 编码端只使用正确接收到的帧作为参考帧, 见图 3.13; 在 NACK 模式中, 还是仍然使用前一帧作为参考帧, 但一旦接收到出错的反馈信息, 则选择一个正确接收到的帧作为参考帧, 见图 3.14。可以说, ACK 模式牺牲了编码效率,

但 NACK 在差错传播的控制方面则不如 ACK，因为当接收到 NACK 消息时，很有可能已经使用出错帧作为参考帧编码并传输了好几帧。

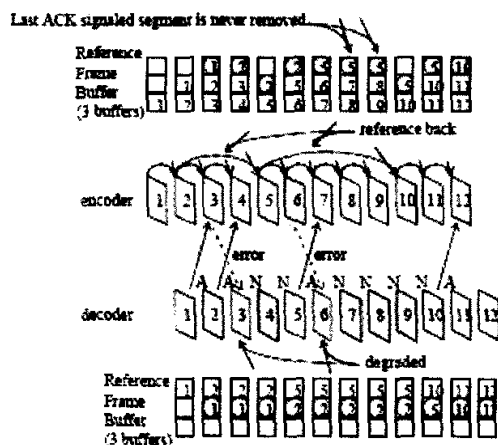


图 3.14 NACK

3.3.3.2 改进的 Intra 帧刷新

当编码端接收到出错反馈时，如果不考虑实时性，是完全可以通過 Intra 刷新的方法来抑制错误的。但是，由于 Intra 帧压缩代价较大，往往达不到实时应用的要求。因此，这种方法需要改进。在[63]中，介绍了一种重采样（Down-Sample）的方法。当接收到出错反馈时，编码端不会整帧地刷新接下来的帧，而是传输一个重采样过的帧。经重采样过帧的运动估计和运动补偿也都是基于之前重采样过的帧。

3.4 本章小结

为了满足压缩视频在网络上的实时传输，在视频相关技术中引入了差错控制机制。本章着重介绍了传统的视频压缩差错控制机制，分三大部分来介绍，分别是编码端容错编码（FEC）、解码端差错隐藏（EC）和编解码交互配合的抗误码技术。

第四章 基于容错特性的快速重要块决策方法

从主观感觉上,可以发现,在压缩视频中,那些无序运动的区域或者运动较大的区域总是对差错最敏感的。即当这些区域发生差错时,在解码端最容易造成视频质量的快速下降。如果能够通过某些参数很好地找出那些最敏感的块,然后施加一定的块保护方法,则必然能够提高容错的性能。

4.1 基于容错特性的重要信息决策和保护方法

在上一章中,曾经提到了基于感兴趣区域(ROI)的容错编码方法。基于感兴趣区域的容错编码方法可以看成是一种基于重要信息保护的方法,那些感兴趣的区域即是重要的信息。在[64]中,对[49]中原有的基于 ROI 的非线性变换方法做了改进。这种方法能够在不修改源解码器的基础上,只通过一个前处理模块和后处理模块,来保证 ROI 区域的高视频质量。然而,这种方法的一个重要缺陷是无法自适应地定义 ROI 区域,这些区域必须预先由用户定义。[65]中提出了一种自适应地确定 ROI 的方法。它首先计算当前帧和上一编码帧之间的均方差(Mean Absolute Difference, MAD),然后通过一个阈值,来确定视觉上的重要区域,即那些运动较大的区域。将那些运动较大的区域放在一起,即组成了 ROI。由于只使用了 MAD 作为确定 ROI 的惟一标准,因此,该方法只适用在视频监控系统中或播音员视频中。这类视频的一个重要共性就是大部分的背景区域都是静止的,只有少部分的区域是活动的。

前面提到的 ROI 区域往往都是用户关心的那些区域,以往基于 ROI 的容错编码也都是为了保护 ROI 区域的视频数据。然而,在一般视频中,往往是没有明显的 ROI 的。但是,尽管如此,在基于容错特性的考虑前提下,仍然存在着明显的重要区域,即那些一旦发生差错时对解码端视频质量下降特别严重的区域。由于往往在解码端都会使用差错隐藏(EC)的方法来恢复出错块或区域,因此,在编码端,如果预先考虑差错隐藏的效果,则找出来的块必然是差错隐藏机制最难恢复的那些块。在[66]中,提出了一种基于重要块(Significant Macroblock, SMB)决策的转码技术。在编码端,站在在解码端的角度来对每个宏块做分析,计算其差错隐藏后的失真,以此判断它们是否重要。差错隐藏失真模拟对一帧中的每一个宏块都要做。最后,根据可用带宽大小,选择适量的宏块,进行 Intra 编码,来达到最优的码率失真比。这种方法对于寻找所谓的重要块来说显然是最优的,但是,它的计算量非常大,因此不易在实时编码传输中,一般需要结合转码技术使用。

Intra 编码往往额外开销较大,因此,当网络带宽不足够满足 Intra 编码时,往往会采用其他的方法来代替,例如只选择性地保护那些重要的 MV 数据[67]。这种方法被称为重要运动矢量(Significant Motion Vector, SMV)决策和保护方法。它以峰值信噪比(PSNR)的下

降作为衡量标准, 计算了只使用运动矢量做运动补偿时 PSNR 的降低和使用差错隐藏时 PSNR 的降低的一个插值, 将那些差值较大的运动矢量看作是 SMV。这种方法还可以扩展, 不仅可以用来保护 SMV, 还可以用来保护那些重要的预测冗余。但这种方法与[66]中的方法类似, 同样存在着较大的计算复杂度。

如果能够在变换编码前确定出重要的信息, 那么许多针对重要信息保护的方法就变得可用了, 例如层次编码、多描述编码和复制等。通过观察和数据统计, 可以发现, 在运动估计 (ME) 后, 可以较准确地衡量出一个块的运动情况, 包括运动的规则性和运动幅度的大小等。而这些量恰好昭示了一个块在发生差错时对视频序列的影响。

为了克服 SMVP 等方法高复杂度的缺点, 又保留了在编码端基于解码端差错隐藏的容错编码机制, 本文设计了一个基于特定差错隐藏方法的快速重要块决策模型 (SMB Determination Model)。这个模型是一个基于大量序列的统计模型, 并通过参数训练 (Parameters Training) 得到几个目标参数。基于这些参数, 可以快速地找出那些视觉上重要的块 (SMB), 然后施加合理的保护。保护的方法被称之为重要块保护方法 (SMB Protection, SMBP)。通过此种方法决策出重要块和保护重要块, 可以在整体上提高编解码器的容错性。

值得注意的是: 1) SMBP 方法所保护的重要块都是 Inter 块。这是因为 Intra 块保护的额外开销太大, 而且, Intra 块不能通过运动估计 (ME) 来衡量其重要度。2) 模型参数训练阶段所采纳的差错隐藏方法与发生差错时解码端进行差错隐藏的方法必须相同, 因为, 决策出的块本身就是基于那种差错隐藏方法的重要块。而且, 对于不同的差错隐藏方法, 所决策出的重要块也不一定相同。

本文提出的一个重要思想即预先搭建一个模型, 通过大量序列的训练得到几个参数值, 用于在编码阶段决策重要块。在具体编解码阶段, 通过三个步骤来做容错编码:

1) 重要块决策阶段。该阶段利用训练得到的参数值, 计算当前帧所有宏块的重要度 (SD) 值, 并根据可保护块的数量, 选取 SD 值大的那些块作为重要块。

2) 重要块保护阶段。该阶段采用某些基于块的保护方法, 采用容错编码对那些重要块做重点保护。

3) 差错恢复阶段。当解码端检测到差错时, 就要进行差错隐藏。由于采用了重要块保护机制, 因此, 在差错恢复之前, 需首先判断出错块是否为重要块, 然后再做相应的差错恢复。

基于容错特性的重要块决策方法和决策模型的搭建将在本章详细介绍。第五章则着重介绍基于重要块的保护方法, 即具体的容错编码方法。

4.2 重要块决策模型

主观上来说, 对于运动较大或者无序的块, 当出错时一般比较难于在解码端做差错隐藏。

为了更好地保护这些块，首先得必须找出这些块。

对于一般的视频编码器来说，在经过了运动估计（ME）之后，都能直接获得分块模式（Block Division Mode）、运动矢量差（MVD）和绝对误差和（SAD）。经过统计分析，可以得出这三个量与一个块的 PSNR 降低（在检测到差错并隐藏之后）有着极大的关系。PSNR 值与视频质量存在着极大的相关性，在一般情况下都可以用来表示视频的质量好坏。也就是说，可以使用分块模式、MVD 和 SAD 作为计算一个宏块重要程度（Significant Degree, SD）的贡献因子（Contributing Factor, CF）。这些因子分别描述如下：

- 块模式：代表了一个宏块的复杂度；
- MVD：预测运动矢量（Predicted Motion Vector, PMV）和 ME 得到的 MV 之间的差值；
- SAD：绝对误差和，用以指示源编码块和其参考帧对应块之间的差错大小。

对于每个贡献因子（CF），如果能够基于统计的方法确定出该因子中每个值对宏块 SD 计算的影响力（Impact Factor, ID）和该 CF 的权重系数（Weight），则可以在具体的编码过程中，利用这些预估出来的参数（Parameter），快速地确定出块的重要性值（SD）。

图 4.1 给出了基于 SMB 决策模型的简单示意图。

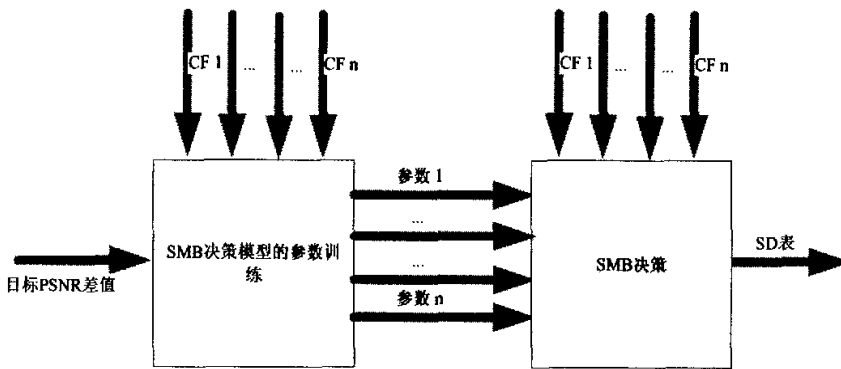


图 4.1 SMB 决策模型

4.2.1 贡献因子分析

对于 H.264 以前的标准，一般只有两种 Inter 块模式，即 16×16 和 8×8 ；但在 H.264 中，则有七种 Inter 块模式，分别是 16×16 、 16×8 、 8×16 、 8×8 、 8×4 、 4×8 和 4×4 。对于 MPEG-4 及其以前的标准来说，块模式无需再划分。但是在 H.264 中，由于 8×4 、 4×8 和 4×4 这三种块模式对应的是子块的模式，因此它们不能与前三种模式一起作为同一个贡献因子来衡量。由于在一个宏块中可以分出四个子块，且每个子块都有可能进行更细小的划分，因此，比较可行的办法是将子块模式单独作为一个贡献因子，这样就得到了四个贡

献因子，分别为：宏块模式（MB Mode）、子块模式（SB Mode）、MVD 和 SAD。但在计算子块每个模式的具体影响值（ID）的过程中，需首先去除 8×8 模式的影响（因为当出现子块划分的时候，必然首先会将 16×16 的宏块划分为 8×8 的四个子块），以防止其 ID 值的过高估计。

MVD 表示的是一个运动矢量，一般由一个水平分量和一个垂直分量组成。这样的一个矢量显然不适合作为模型的输入参数，而需经过一个合理的量化过程。简单而有效的方法，即是用 MVD 的水平分量和垂直分量的绝对值和作为 MVD 的衡量标准（之后 MVD 值即指的是其水平分量和垂直分量的绝对值）。经过统计分析，可以将 MVD 值划为三段：1）等于 0 的；2）小于 2.5 的；3）大于或等于 2.5 的。

SAD 值是千变万化的，一般为几十到几千不等。即使是在同一序列中，也有可能分布在极大的范围内。但是，当 SAD 值很大时，本身就说明了其对应宏块是非常重要的。因此，只需将 SAD 值较小的那一部分进行细分，划成一个个区间；剩余的那些特别大的 SAD 值，全部都归到一个区间中去。当输入一个具体的 SAD 值时，首先判断其落在哪个区间，然后以区间对应的 ID 值作为该 SAD 值的 ID 值即可。

4.2.2 模型参数训练

经过对贡献因子（CF）的分析可知，对于 MPEG-4 以前的标准，可以划分出三个贡献因子，对于 H.264，则可以划分出四个贡献因子。本文主要针对 H.264 标准做了训练和模拟，因此后面的参数训练和模拟实验以及第六章的内容都时基于 H.264 标准的。图 4.2 给出了基于 H.264 标准的简单参数训练示意图。

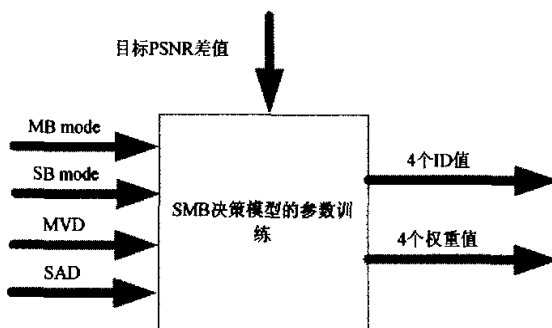


图 4.2 面向 H.264 标准的参数训练

4.2.2.1 训练环境

对于 H.264 标准，目前最流行的参考软件是 JM，其最新版本是 JM10.2[68]。在 SMB

训练模型中，用它作为实验平台，使用的轮廓（Profile）为基本轮廓（Baseline Profile），常用的编码参数设定为：QP=28，fps=30。由于在 SMB 决策训练中，是以每个宏块为最小单位考虑的，因此，解码端相应的差错隐藏机制也必须是宏块级（MB-level）的。在 JM10.2 中，默认的差错隐藏方法是帧拷贝和运动矢量拷贝（Frame Copy and MV Copy）。本文对此 EC 方法做了简单的修改，将其改为面向宏块级别的。

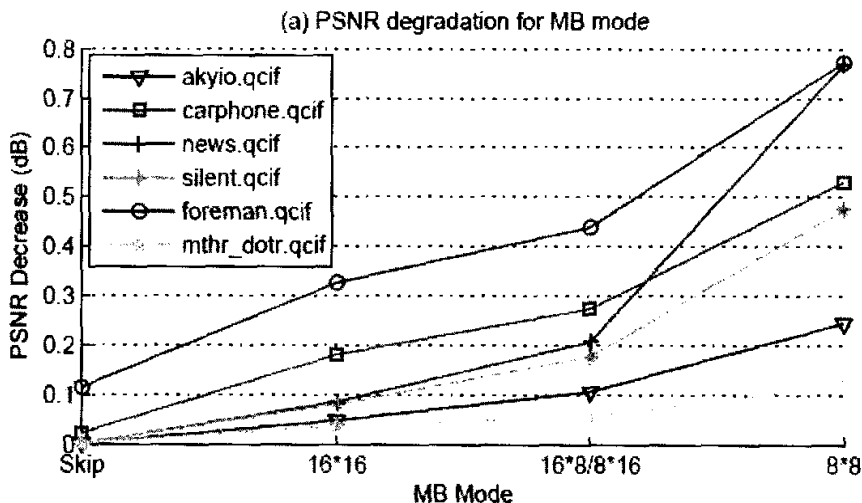
参与训练的序列是六个常用的视频测试序列，其格式都是 QCIF（大小 176×144 ）的，序列名和序列测试帧数如表 4.1 所示。

表 4.1 训练序列

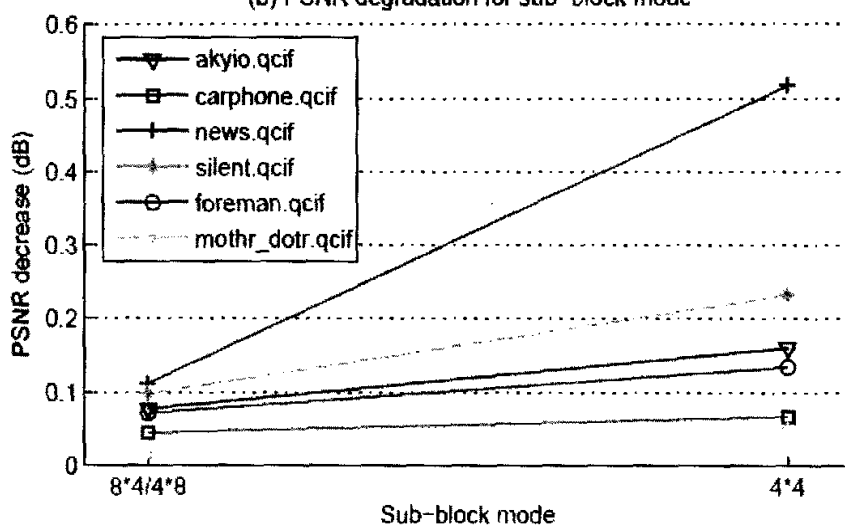
训练序列	foreman	akiyo	silent	news	carphone	mother&daughter
帧数	400	90	300	300	90	400

4.2.2.2 影响度值的确定

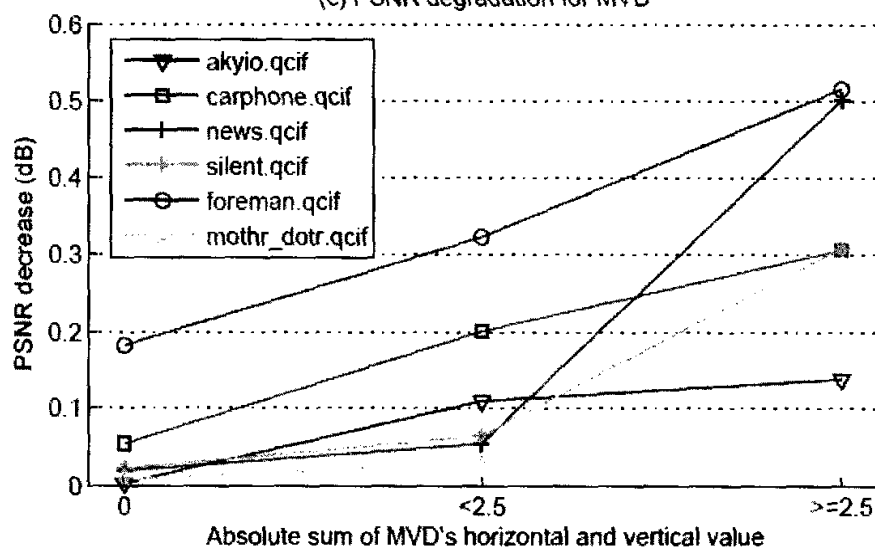
影响度值（ID 值）表示一个贡献因子的某个值对决策 SMB 的影响是多大。在确定 ID 值之前，先计算出各个序列中每个宏块丢失后的 PSNR 降低 $\Delta PSNR$ （已考虑解码端的差错隐藏），同时根据 4.2.1 中对 CF 的分析，确定对应宏块的每个 CF 的值。最后，针对每个序列，计算出每个 CF 值所对应的平均 PSNR 降低（ $\Delta PSNR$ ），见图 4.3 所示，其中（a）、（b）、（c）和（d）四幅图分别是宏块模式、子块模式、MVD 和 SAD 的 $\Delta PSNR$ 。其中子块模式的 $\Delta PSNR$ 是已经去除了宏块模式的影响之后的。



(b) PSNR degradation for sub-block mode



(c) PSNR degradation for MVD



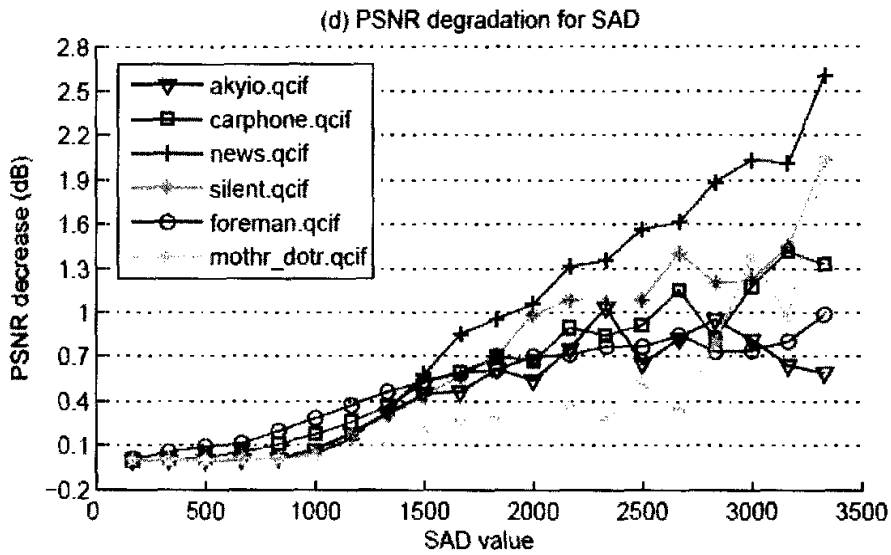


图 4.3 平均 PSNR 降低 $\Delta PSNR$

通过图 4.3，可以观察到，各影响因子与 PSNR 的降低有着一种近似线性的关系。虽然当 SAD 值较大时，序列的线性关系不是那么明显，但是，由于 SAD 值较大时，宏块的重要性已经很强了，因此，这并不影响 SMB 的决策。

此外，还可以看到，在不同序列中，各贡献因子对于 SMB 的决策影响也是不一样的。因此，在确定了平均的 $\Delta PSNR$ 之后，还需要对 $\Delta PSNR$ 做一个分析，将分析得到的结果作为一般化的影响度 (ID) 值。这个过程可以通过一个归一化 (Unify) 的过程来实现。它首先针对每个序列，对各贡献因子可能的值所对应的平均 $\Delta PSNR$ 做一个归一化处理，将所有序列放在一起平均化后，最后再做一次归一化处理。最终得到的归一化后的数据可参见表 4.2。这张表被定义为 ID_Table ，在后面的处理中会用到。

表 4.2 影响度表 ID_Table

CFs	可能取值 (已归类)	ID 值 (归一化后)
MB mode	Skip	0.000000
	16*16	0.234509
	16*8 / 8*16	0.416249
	8*8	1.000000
Sub-block mode	No sub-block division	0.000000
	8*4 / 4*8	0.383990
	4*4	1.000000
MVD	0	0.000000
	<2.5	0.402905
	≥ 2.5	1.000000
SAD	0~199	0.000000

200~399	0.008963
400~599	0.019046
600~799	0.029335
800~999	0.055977
1000~1199	0.095919
1200~1399	0.165315
1400~1599	0.249909
1600~1799	0.339742
1800~1999	0.400320
2000~2199	0.465979
2200~2399	0.499504
2400~2599	0.598390
2600~2799	0.645016
2800~2999	0.625197
3000~3199	0.714499
3200~3399	0.716955
3400~3599	0.804658
3600~3799	0.797420
>3800	1.000000

4.2.2.3 权重值的确定

在 4.2.2.2 中, 已经确定出了影响度值 (ID)。接下来, 就是确定各贡献因子的权重值 (weights)。权重值可以用 cf_weight 表示, 其中

$$cf_weight = [cf_w_1, cf_w_2, cf_w_3, cf_w_4]^T \quad (2)$$

在 (2) 中, cf_w_1 , cf_w_2 , cf_w_3 , cf_w_4 分别表示了宏块模式、子块模式、MVD 和 SAD 的权重系数。公式 (3) 给出了通过最小二乘法决策出权重系数的方法, 如下:

$$\begin{cases} \min \sum_j |\text{Unify}(\Delta PSNR_j) - \sum_{cf_i=1}^4 (\omega_{cf_i} * ID_{cf_i,j})|^2 \\ \sum_{cf_i=1}^4 \omega_{cf_i} = 1; \quad 0 \leq \omega_{cf_i} \leq 1; \end{cases} \quad (3)$$

其中, j 包含了训练序列中所有供考虑的宏块, $\text{Unify}()$ 是指归一化函数, $\Delta PSNR$ 代表了 PSNR 的降低 (无差错情况下和差错隐藏情况下的比较), $ID_{cf_i,j}$ 是宏块 j 中贡献因子 cf_i 的影响度值, 可以通过 ID_Table 查找得到, ω_{cf_i} 则是待决定的贡献因子 cf_i 的权重系数。通过对被考虑的六个序列的训练, 可以得到

$$cf_weight = [0.0250, 0.7968, 0.0106, 0.1676]^T \quad (4)$$

可以看到，子块模式是最重要的贡献因子，贡献比重非常大；SAD 次之；而宏块模式和 MVD 对重要块决策的影响则较小。

4.2.3 重要块决策

在参数训练都完成之后，可以得到两组值：ID 值和 cf_weight 值。在具体的编码过程中，只需通过这两组值，就可以快速简单地计算出一个 Inter 宏块的重要度（SD）。计算公式如下：

$$\begin{cases} SD(i) = \sum_{cf_i=1}^4 [cf_w_{cf_i} * ID_{cf_i}(i)]; \\ 0 \leq i < PicSizeInMBs; \end{cases} \quad (5)$$

其中，i 是当前宏块在帧中对应的宏块序号，ID 值可以简单地通过 ID_Table 查找获得， $PicSizeInMBs$ 是一帧中总共的宏块数目。

当一帧中每个宏块的 SD 值都计算完毕后，该帧的 SD_Table 就构建好了。每个 Inter 帧都会为之构建一个 SD_Table ，而 SD 值最大的那些宏块就是被确定为 SMB 的宏块。一帧中 SMB 的数量可以根据特定应用的需要，也可以实现为 SMB 保护预留一定量的带宽。第六章中的重要块保护方法（SMBP）就是基于本章决策出来的 SMB 的。

4.3 决策精度测试实验

对于决策出的重要块保护模型来说，决策出的重要块与理想的重要块之间的匹配程度，是重要块保护方法关心的重点。因此，本节着重通过实验，分析了本文中提出的重要块决策方法的决策精度。其中，理想的重要块即那些 PSNR 降低（ $\Delta PSNR$ ）最大的块。决策精度的衡量可以从两方面考虑：一是决策出的重要块与理想的重要块的匹配百分比，即匹配准确度；二是即使匹配有一定的误差，那么误差的那些块是否也是相对比较重要的。在 4.3.2 和 4.3.3 中，分别就这两方面做了实验，并给出了决策精度的衡量与分析。

4.3.1 测试环境

在 SMB 决策阶段，采纳了六个常用的序列。这里，为了证明决策的精度，除了使用原来的六个序列外，还引入了第七个 QCIF 序列，是 300 帧的“salesman”序列。使用的测试平台和其参数设定均如同训练时的环境配置。测试结果如下所述。

4.3.2 匹配准度衡量

对所选的七个序列，首先计算其决策的准度。在决策准度的评估前，需要首先构建两张 SD_Table ，分别是：1) 通过本文提出的 SMB 决策方法构建的 SD 表，是被评估的表；2) 通过 $\Delta PSNR$ 构建出的 SD 表，作为理想的 SD 表。然后对两张 SD 表做分析，最后得到其匹配准度。匹配准度示意图如下（图 4.4），它给出了 SMB 保护百分比与匹配准度的一个对应关系：

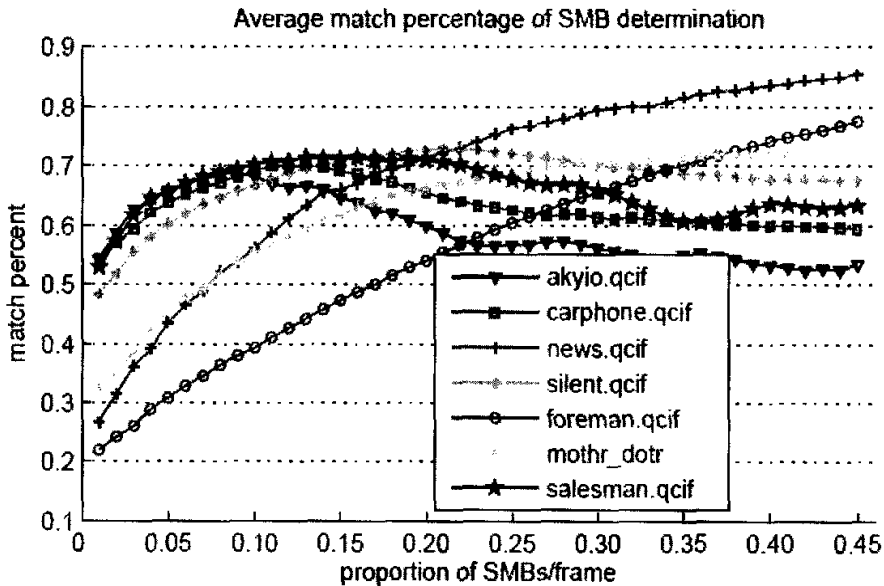


图 4.4 匹配准度示意图

通过该示意图，可以观察到，对于那些运动较少的视频，例如“akiyo”、“news”和“silent”，当受保护的 SMB 在 10%~20% 时，其匹配精度是最高的，可以达到 70% 左右；而对于那些运动较大的视频，其匹配精度随着受保护的 SMB 百分比增加而递增。从整体上说，不论是高运动和低运动的视频，其匹配精度都是可接受的。

而对于没有在参数训练中使用的序列“salesman”，其匹配精度也是相当高的，甚至比大部分的序列还要好。因此，可以说，本文提出的重要块决策方法的决策精度是相当高的。

4.3.3 SMB 与非 SMB 的 $\Delta PSNR$ 比较

如果在匹配失效的情况下，SMB 决策模型找出来的并不是最优的重要块。因此，有可能后续的保护措施会保护那些相对不重要的块。但是，如果即使失效匹配时，也保护了那些次要重要的块，而非不重要的块的话，则对决策出的 SMB 施加保护也是相对有效的。衡量决

策出的 SMB 保护的有效度, 即比较 SMB 与非 SMB 之间的 $\Delta PSNR$ 。

图 4.5 给出了 SMB 的平均 $\Delta PSNR$ 和非 SMB 的平均 $\Delta PSNR$ 之间的差值折线图。通过图中可以看到, 保护越少的重要块, 则保护效果越明显。由于“akiyo”、“mothr_dotr”和“salesman”本身的 $\Delta PSNR$ 就较低, 因此图中比较的差值也比其他序列相对低一些。

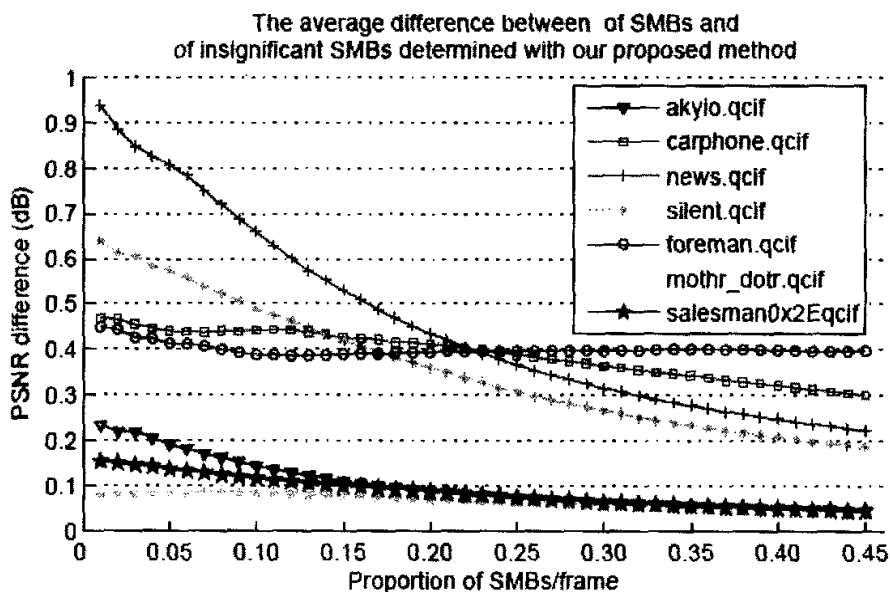


图 4.5 SMB 和非 SMB 的平均 $\Delta PSNR$ 比较

4.4 本章小结

本章着重介绍了基于容错特性的快速重要块的决策方法, 是本文的核心内容。重要块的决策是通过一个统计模型得到的, 并且统计模型的参数训练是事先做好的, 因此在编码过程中, 可以直接使用训练得到的参数, 从而无需太多额外的运算, 保证了其快速性。通过对实验结果的分析, 无论是对匹配准确度评估还是对决策出的重要块的重要性评估, 都可以得到, 本文提出的重要块决策精度是比较高的。此外, 在测试中, 还使用了一个参数训练过程中没有使用过的序列。通过对这个序列的测试和评估, 可以得到基于容错特性的重要块决策方法的决策精度是相当高的。因此只要对决策出的重要块施加一定的保护措施, 编码器的容错编码性能必然会有很大的提高。

第五章 基于重要块保护的容错编码

第四章介绍了基于容错特性的重要块的决策方法。决策出重要块的最终目的是通过保护它们，从而达到编码容错性的提高。Inter 块保护的目的是使得当信道发生差错时，差错传播能够得到很好的控制。本章着重介绍了重要块的保护方法，并通过比较和改进，设计了一种有效的重要块保护方法。

5.1 基于决策出的重要块保护方法的研究

基于块的保护方法从根本上说，也是一种非平衡差错保护（UEP）的方法。它的基本原理是：将较强的保护施加于重要块上，而将较弱的保护施加于非重要块上。基于块的保护方法有很多，包括拷贝复制、Intra 刷新、层次编码（LC）和多描述编码等（MDC）。在[49]和[64]中，所提出的基于 ROI 的保护方法从根本上说都是拷贝复制重要的区域（块）。当发生差错时，由于每个重要块都有一份拷贝，因此，其出错的概率相对非重要块要低一些。Intra 刷新的基本原理是通过对决策出的重要块做 Intra 编码，以达到抑制差错传播的目的[69]。在层次编码中，则是将重要信息（即重要块）保存在基本层中。由于基本层的鲁棒性比增强层要好，因此，重要块在基本层中就不容易丢失或出错。MDC 则是另一种目前流行的差错保护方法。MDC 对差错的保护主要是通过合理的数据分割方法，使得受保护区域的数据分散于各个逻辑信道或者物理信道中，并且每一个信道中的数据解码后都能够获得可接受的视频质量。MDC 分割后的每一块数据都被称为一个描述。MDC 方法已经被证明了是一种可靠有效的差错控制方法[53]。

5.1.1 块复制方法

块复制方法是最简单的保护方法，即将重要块的信息直接复制即可。然而，由于网络丢包往往具有连续性，如果将复制后的块信息直接加在当前帧后面，往往造成复制块与原始块一同丢失。这样，非但没有起到块保护的作用，反而造成了网络负荷的增加。在[67]中，重要的运动矢量（SMV）是附在下一帧中保护的。对于本文提出的方法，也可以采用类似的保护方法，即将重要的块信息与下一帧数据一起传输。

此外，也可以采纳[49]和[64]中的像条（slice）分割方法，即以宏块行（Macroblock Line）为单位分割像条，每行为一个像条。这样，重要块就可以通过复制到其他行中进行保护。在传输时，每一个像条打包成一个包。

[49]和[64]中的非线性变换方法在某些特定的应用中也是可采纳的，尤其是大部分时间

都静止的监控视频以及只有局部变化较大的播音员视频等。通过本文的决策方法，可以决策出一个重要区域，然后对此区域做非线性变换，从而达到对重要区域的保护。见图 5.1。



图 5.1 非线性变换的重要块保护方法

不过由于非线性变换方法是在做运动估计前做的，而我们的重要块决策是在运动估计后做的，因此，当采用非线性变换方法时，相当于做了两次运动估计。对于一个编码器来说，运动估计是最耗时的部分。因此，两次的运动估计显然大地增加了编码的时间，也就使得编码的实时性难以得到保障。不过，我们可以在重要块决策时，同时保存那些重要块的运动矢量（MV）和残差（Residual）信息。这样，非线性变换后的运动估计中，就无需再对那些重要块做运动估计。

5.1.2 层次编码方法

层次编码方法在 3.3.1.3 中已经介绍过。对于层次编码方法，一项重要的任务就是如何分割数据，即将数据按照重要性分成几层，并对各个层次施加不同程度的保护。在第四章中可以看到，在进行了重要块决策之后，我们已经可以得到一个重要度表（SD Table），这个表给出了每个块重要的程度。通过 SD 表，就能够将一帧中的块按照其 SD 值分层。

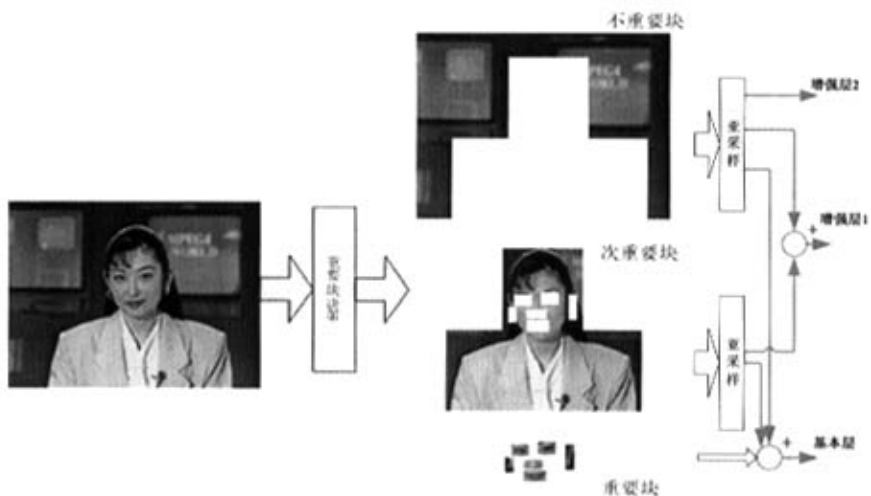


图 5.2 基于重要块保护的层次编码方法

基于层次编码的重要块保护方法可见图 5.2。值得注意的是,对于那些决策出的重要块,直接将它们放在基本层中保护;那些相对不重要的,则做亚采样,然后在基本层中只保护采样后的部分数据。其中越是不重要块的块,采样精度就越低。

层次编码虽然有效,但是要实现一个可靠性高的基本层传输信道却比较难。因此,往往很难在某些常用的网络上使用。

5.1.3 多描述容错编码方法

多描述容错编码方法(MDC)与层次编码方法类似,也是在决策出重要块后,对其施加重要保护。只不过与层次编码不同的是,这里采用的是 MDC 方法,即在解码端得到的每个描述都能单独解码。这样,就避免了在层次编码中,因基本层丢失而造成的不可解码问题。

在[70]中,提出了一种基于多相采样多描述(Polyphase Down-sampler Multiple Descriptions, PDMD)编码方法,如图 5.3 所示。它将源图像 S 经过水平采样(以参数 R)和垂直采样(以参数 C),最后得到 N 个独立且同等重要的描述。

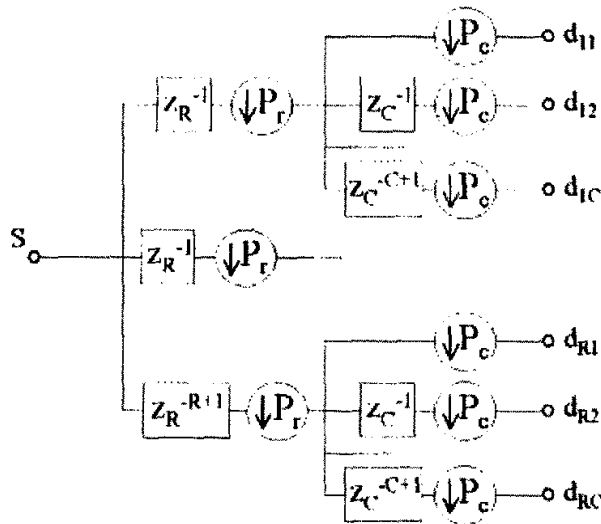


图 5.3 PDMD

PDMD 方法本来是针对整帧图像的。但是,我们可以观察到,对于 H.264 标准来说,运动估计和变换编码的最小单位都可以是 4×4 的子块。因此,可以设计一种面向 H.264 的 PDMD 方法,用来专门保护重要块数据。由于只需重点重要块区域的数据,因此,只对 SMB 部分施加 PDMD 保护方法。改进了的 PDMD 方法见图 5.4。可以看到每个重要块都被划分成了 16 个 4×4 的小块,通过水平采样和垂直采样,将其分成 4 个同等重要的描述。那些非重要的宏块,则可以任意保存在一个可用的描述中。

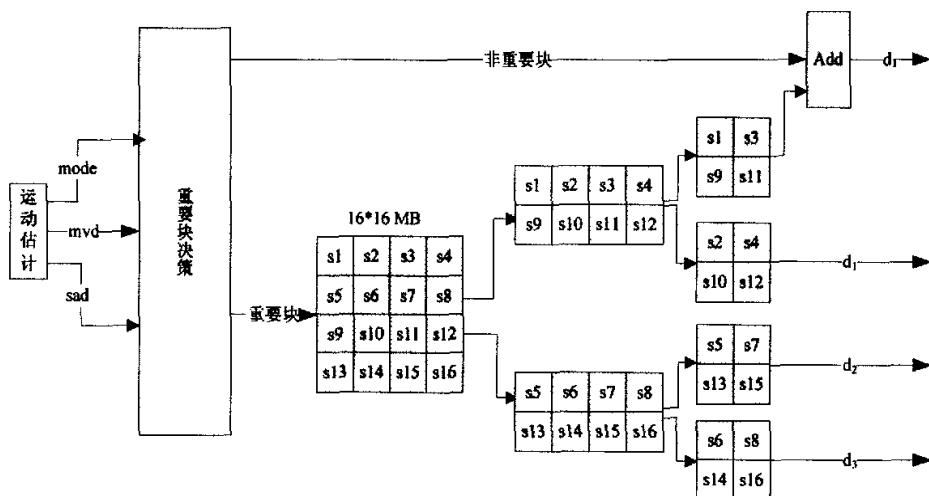


图 5.4 改进了的 PDMD 方法

5.1.4 Intra 块刷新方法

Intra 块刷新方法是当前最常用的块保护方法，它能非常有效地抑制差错传播。Intra 块刷新方法在 3.3.1.6 中有详细的介绍。对于本文决策出的重要块，我们可以简单地对那些 SD 值大的块做 Intra 编码。然而，由于 SD 值与差错传播影响并不一定呈正比关系，因此，这样做 Intra 编码并不一定十分准确。

不过，选择性的 Intra 块刷新方法仍然可以通过简单的处理 SD 值得到。由于 SD 值在一定程度上代表了一个宏块对差错的敏感度，因此，当一个宏块丢失时，后续参考它的宏块受其影响的程度可以近似地用其 SD 值来衡量。那么，将其与当前帧的 SD 值相加，就得到了一个用以衡量是否需要 Intra 编码的量，我们定义为 $SD_Intra(i)(j)$ ，其中 i 表示当前块在当前帧 j 中的索引 (index)。

$$SD_Intra(i)(j) = SD(i)(j) + SD_Intra(ref_index)(j-1) \quad (6)$$

其中 ref_index 表示当前块在参考帧 $j-1$ 中的参考块。如果采取的是非整像素运动估计，那么可以根据参考块覆盖到的宏块的比例，通过对覆盖到的宏块的 SD 值加权平均而得到 SD_Intra 值。

在得到 SD_Intra 值后，就可以通过将其与一个预先定义的阈值 (threshold) 比较，来决定是否对一个宏块做 Intra 编码。具体如下：

$$IF(SD_Intra(i)(j) > threshold)$$

$Intra_Refresh(MB(i)(j));$

$SD_Intra(i)(j)=0;$

END IF

(7)

Intra 块刷新方法在抑制差错传播上是非常有效的,不过,由于 Intra 编码需要相当于 Inter 编码 10 倍的比特数去编码同样的块,因此, Intra 编码最大的缺陷是容易造成编码码率的明显增加。这对于带宽本身就不足的网络来说,是不可行的。

5.1.5 改进的多帧参考方法和帧内预测方法

多帧参考方法和帧内预测方法是在 H.264 中使用的帧参考机制。通过多帧参考,可以为一个宏块找到更加相近的块,从而减少了编码冗余的字节数,降低了码率。而通过帧内预测,可以更好地降低帧内数据的冗余度。通过对多帧参考方法和帧内预测方法的探索与研究,可以发现,它们也可以应用到重要块的保护中。

不论是多帧参考方法还是帧内预测方法,在重要块的保护中,都是为了通过更多表示形式,来描述重要块。这样,当某一描述丢失时,仍然可以通过另一描述,完整地恢复某一块。与 MDC 不同的是,在这种方法中,只要正确接收到一种描述,即可完全恢复的重要块(假设参考块或者预测块是正确的)。

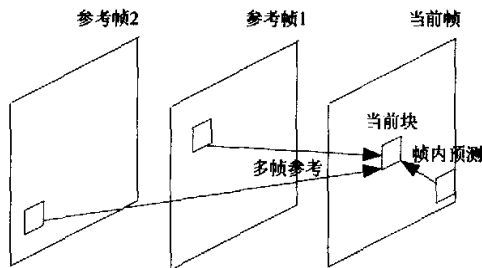


图 5.5 多帧参考和帧内预测

为了重点保护重要块的数据,应对决策出的重要块做多一些的帧参考或帧内预测,并将这些冗余信息以恰当的方式保存起来。同时,为了尽可能保证网络差错不至于造成所有的描述都丢失,需要将描述分散在不同的网络包中。常用的方法是把这些描述放在不同的像条(Slice)中保存,然后每个像条均各自打包发送。

不过,在运动估计匹配程度不高或帧内预测效果不好的情况下,要保存的冗余信息量将会是比较大的。在这样的情况下,在对重要块施加保护的同时,也应当考虑其所带来的额外编码码率的增加。

5.2 提出的基于容错特性的重要块保护方法

5.1 节对重要块的保护方法并不是孤立的,而是可以综合使用的。同时,各种方法针对不同的情况也会有不同的效果。通过研究和分析,我们提出了一种效果良好的基于容错特性的重要块保护方法。

非线性变换方法由于要做一次额外的运动估计,因此,显然不适用于实时应用环境。层次编码方法由于需要物理信道的支持,也不能适用于我们的容错编码方法。显然,MDC容错编码方法、Intra块刷新方法和改进的多帧参考/帧内预测方法是更加合适的办法。

5.2.1 基于容错特性的重要块保护方法

由于 Intra 块编码的高码率缺陷,在一帧中进行 Intra 编码的块数量往往是受到限制的,即不能超过可用的带宽大小。同时, Intra 编码的块数量越多,也就意味着越少的块可以接受 MDC 容错编码方法的保护。如何确定 Intra 编码的块数量和 MDC 容错编码的块数量呢?下面介绍本文提出的一种基于容错特性的重要块保护方法,称为 SMBP。

- 1) 对需要 Intra 编码的宏块做 Intra 刷新,但实际执行的 Intra 刷新的宏块数目不能超过一帧宏块数量的 3%。如果超过 3%,则对其中需要 Intra 刷新的一部分宏块做多帧参考/帧内预测方法。
- 2) 假如没有宏块需要 Intra 刷新,则全部做 MDC 编码。
- 3) 在已使用了某种保护方法之后,将不再对同一块使用其他的保护方法。

5.2.2 解码端差错恢复

在解码端,当检测到差错时,则需要使用差错隐藏方法(EC Tools)来恢复错误。在基于重要块的差错保护中,差错隐藏必须首先判断检测到的差错块是否为 SMB。如果是 SMB,则首先查看其是否有其他可用的参考帧信息或帧内预测信息;如果没有,则考虑改进了的 PDMD 的保护,来做差错恢复;否则,只使用改进了的帧拷贝和 MV 拷贝的差错恢复方法。具体过程如下:

- 如果出错的宏块不是 SMB,则使用改进了的宏块拷贝和 MV 拷贝的差错恢复方法(Modified EC)。
- 如果出错的宏块是 SMB,则首先判断是否采用了多帧参考/帧内预测方法。如果使用了该方法,则查找是否有可用的其他帧参考信息或帧内预测信息。如果没有,则使用改进了的宏块拷贝和 MV 拷贝的差错恢复方法。
- 如果未采用有多帧参考/帧内预测方法,则确定使用了 MDC 方法。查看有几个描述出

错了,然后通过空间插值的方法,来恢复那些出错的 4×4 小块(对应的是出错的描述)。如果所有的描述都丢失,则也使用改进了的宏块拷贝和 MV 拷贝的差错恢复方法。

5.3 块保护效果测试实验

衡量重要块决策和保护方法的最终标准还是要看块保护的效果,即通过与其他方法的比较,得到本文提出的方法在出错状况下抑制 PSNR 下降的效果要比其他的方法好。为了更好地与其他方法做比较,在模拟实验中,将重要块的保护份额预先定为 10%。参与比较的块保护方法有 10%随机块保护(Random MB Protection, RMBP)、10%随机 Intra 块刷新(Random Intra Refresh, RIR)和修改过的差错隐藏方法(Modified EC)。由于 Intra 块刷新方法在比特率上比本文提出的重要块保护方法要高许多,因此,如果 Intra 块刷新比率也是 10%的话,那么它的效果必然会比 SMBP 好。因此,在比较过程中,还增加了一种更为合理的比较方式,即 Intra 块刷新的比率为 5%。

5.3.1 测试环境

测试的环境与 4.3.1 中的配置环境是一致的。测试的序列是“foreman”和“silent”,分别代表了高运动序列和低运动序列。

信道包丢失率(Loss Rate)则模拟了两种情况,一种是高丢失率(0.4),一种是低丢失率(0.1)。

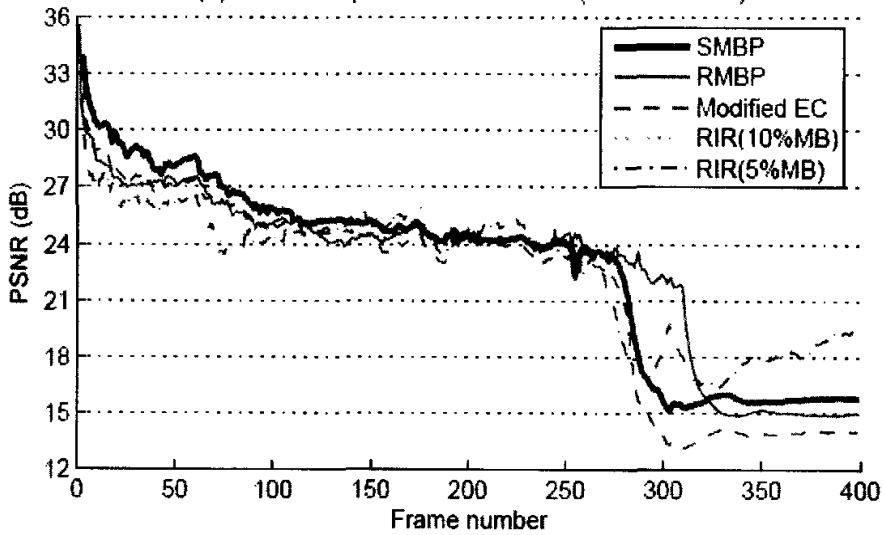
5.3.2 保护效果

图 5.6 直观地刻画了在不同的丢包率下,几种参考的块保护方法与 SMBP 的比较情况。表 5.1 则直接给出了 SMBP (10%)保护方法与参考方法的比较后得到的 PSNR 差值。通过比较,可以清楚地看到, SMBP 无论在什么情况下都比 RMBP 好。对于“foreman”序列,当出错率为 0.1 时, SMBP 的平均 PSNR 高了大约 0.07dB,而当出错率为 0.4 时,则为 0.46dB;对于“silent”序列,出错率为 0.1 和 0.4 时高出的 PSNR 值分别为 1.67dB 和 2.87dB。这可以看出,10%的 SMBP 在精度和保护效果上都是较好的,特别是对于那些低运动的视频序列。前面已经分析过,10%宏块 Intra 刷新方法必然比 SMBP 要好,但是在测试的所有情况中,它所得到的平均 PSNR 都只比 SMBP 高一点点,不会超过 0.9dB。而 5%宏块 Intra 刷新方法(比特率约为 SMBP 的 1.5 倍)在丢包率为 0.1 时的平均 PSNR 与 SMBP 相当,但在丢包率为 0.4 时,甚至比 SMBP 保护方法的效果还差。

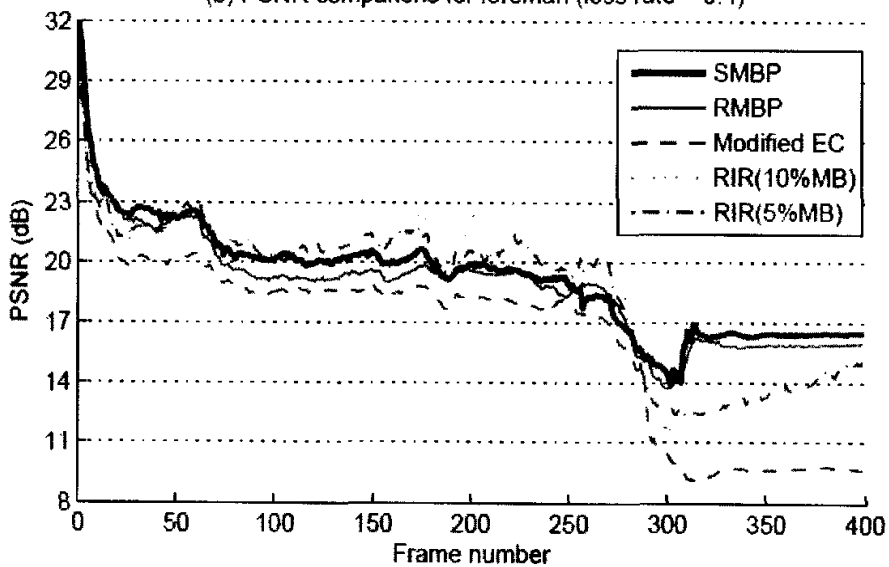
表 5.1 SMBP (10%) 保护方法与参考方法的比较

测试序列/丢包率	SMBP 单位: dB	RMBP 单位: dB	Modified EC 单位: dB	RIR(10%MB) 单位: dB	RIR(5%MB) 单位: dB
Foreman/0.1	22.90	0.07	1.13	-0.43	0.12
Foreman/0.4	19.20	0.46	2.94	-0.89	0.45
Silent/0.1	30.26	1.67	3.86	-0.50	-0.33
Silent/0.4	25.10	2.87	2.96	-0.84	0.26

(a) PSNR comparisons for foreman (loss rate = 0.1)



(b) PSNR comparisons for foreman (loss rate = 0.4)



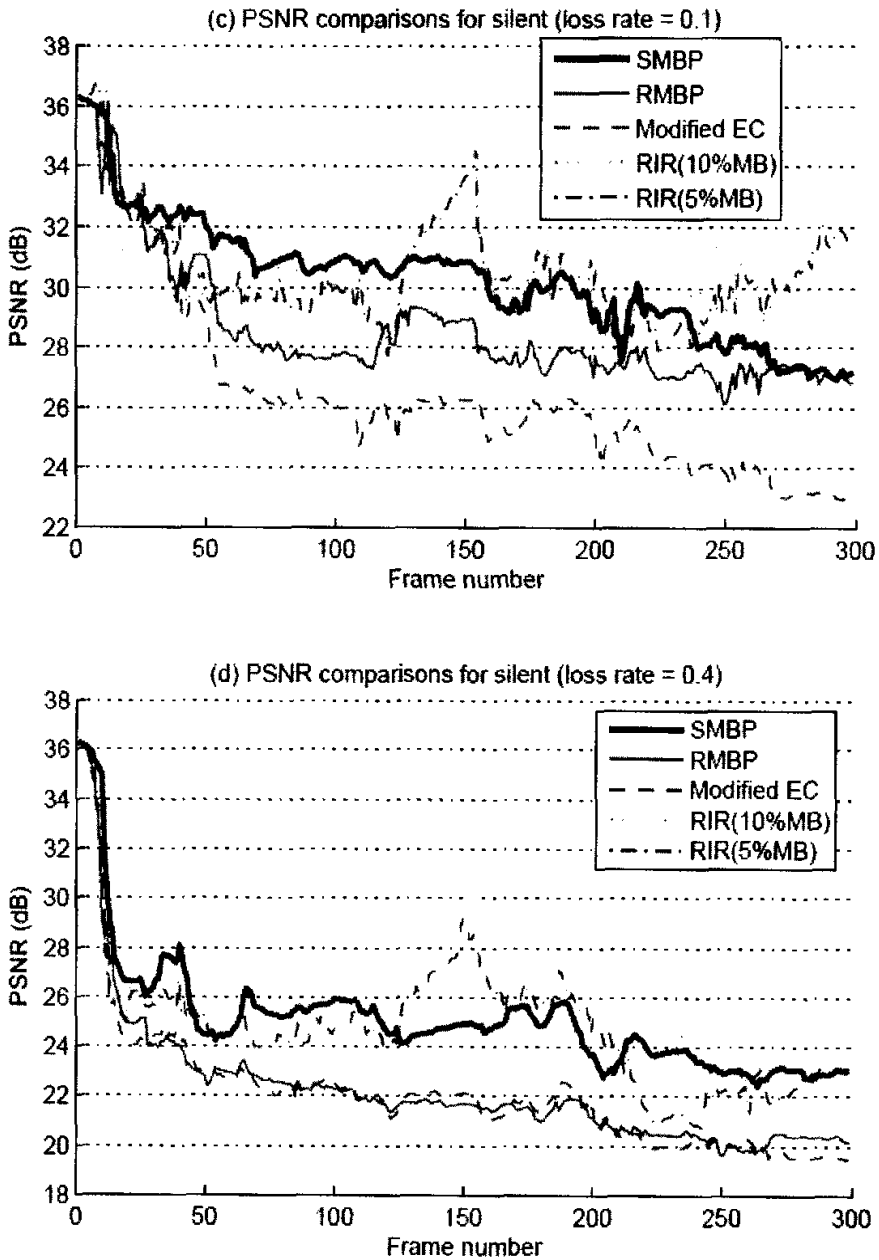


图 5.6 不同的丢包率下的块保护方法比较

5.4 本章小结

本章重点介绍了基于重要块保护的差错控制方法。通过对基于块保护的傳統方法的分析和改进，得到一种基于重要块保护的 **PDMD** 方法。使用该方法，对决策出的重要块做重点

保护。通过实验分析,可以看出,本文提出的基于重要块决策和保护的方法是行之有效的。不论是高运动的视频序列,还是低运动的视频序列,都能够起到良好的容错效果。

第六章 基于重要块保护的容错编码在移动视频中的应用

无线网的发展和 3G 网络的出现,使得移动视频得到更广泛的应用。在 3GPP/3GPP2 标准中,推荐在 3G 终端设备中使用 H.264/AVC。然而,由于无线环境的高误码率,差错控制机制对于移动视频来说就显得尤为重要。同时,由于无线网和移动终端的固有特性,许多在 PC 机和 Internet 中常用的差错控制方法在 3G 环境中却不能使用。在[71]中,着重研究了 3G 环境下的视频压缩差错控制方法。可以看到,对于移动视频来说,设计并采用一种计算复杂度较低但却有效的差错控制方法,是非常重要的。而通过第四章和第五章的实验结果可以看到,本文提出的基于重要块保护的容错编码方法显然是个不错的选择。

目前,常用的移动视频应用有移动视频监控系统、移动视频点播系统和移动视频会议系统等。本文的方法对于上述应用来说都是适用的。

6.1 移动视频系统简介

移动视频技术是集多媒体技术、计算机网络技术、工业控制和人工智能等技术于一体的综合运用。传统的非网络模拟图像系统具有所需存储量大、系统功能简单、可靠性差、搭建工程量大等许多不足之处。随着计算机处理能力和网络技术的发展,数字化、网络化和智能化已经成为视频系统的主要发展方向,它们弥补了传统系统的不足。同时,伴随着 3G 网络的出现,更进一步地拓宽了视频应用的空间限制。移动视频应用已经开始不断出现,并逐渐流行起来。它将越来越多地融入到人们的生活中。

目前,已有的或仍然在研究中的移动视频系统主要包括了移动视频监控系统,移动视频点播系统和移动视频会议系统。系统的简单示意图可见图 6.1。

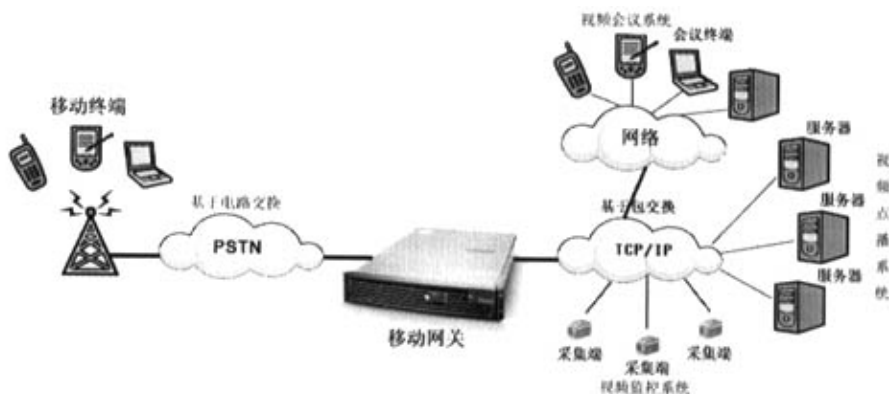


图 6.1 移动视频系统

由于移动视频主要是针对移动设备和无线环境的,因此在各系统中必然包括的两大部分是无线网关和移动终端。其中,无线网关作为无线网的核心部件,主要承担无线网和 IP 网络之间的包交换;而移动终端则主要作为客户端存在,接收各个系统传输过来的视频数据。在视频会议系统中,某些处理能力较强的移动终端也可作为采集端。

由于各个系统的应用场景不同,各系统又包含了不同的部分。在视频监控系统中,各采集端采集的数据通过基于 TCP/IP 的网络传输到无线网关。在视频点播系统中,数据源则直接从视频服务器上获取。而对于视频会议系统,则可以分为单向视频交互会话和双向视频交互会话两种。在单向视频会话中,移动终端只作为视频解码端,不参与视频数据采集和传送;在双向视频会话中,移动终端既要接收视频数据,也要参与视频数据的采集和传送。

6.2 基于重要块保护的容错编码在移动视频系统中的应用

基于重要块保护的容错编码方法具有快速和准确的决策能力以及较好的视频质量保证,因此可以应用到移动视频中。在本节中,我们首先构建了移动视频监控系统和移动视频点播系统和移动视频会议系统。在第四、五章的研究基础上,可以得出,当选取 10%左右的重要块作保护时,其决策精度是相对较高的。同时,根据无线网的带宽限制,10%的重要块保护增加的额外信息也较少。因此,在移动视频系统中,可以采纳 10%的重要块保护方法。

为了更加全面的衡量基于本文提出的容错编码的容错效果,我们仍然采纳了其他的几种方法以做比较,它们分别是:帧拷贝和运动矢量拷贝的差错隐藏方法、随机 Intra 编码方法(5%Intra 编码)、随机重要块保护方法(RMBP)。同时,为了给出不同网络环境下的结果,也分别模拟了高丢包率(40%)和低丢包率下(10%)的保护情况。最后,选取两个不同的序列,分别代表高运动和低运动的情况。

图 6.2 给出了当丢包率为 40%时,用 salesman 序列模拟的视频会议系统下的情况。可以看到,基于重要块的容错编码方法甚至比 5%的随机 Intra 编码方法还要好。



(a) 帧拷贝和运动矢量拷贝的差错隐藏方法



(b) 随机 Intra 编码方法(5%Intra 编码)



(c) 随机重要块保护方法 (RMBP)



(d) 重要块保护方法 (SMBP)



(e) 无差错情况

图 6.2 salesman 序列、40%丢包率、10%Inter 宏块保护的视频会议系统

图 6.3 给出了当丢包率为 10%时，用 akiyo 序列模拟的视频点播系统下的情况。可以看到，保护的效果也是相当好的。



(a) 帧拷贝和运动矢量拷贝的差错隐藏方法



(b) 随机 Intra 编码方法 (5%Intra 编码)



(c) 随机重要块保护方法 (RMBP)



(d) 重要块保护方法 (SMBP)



(e) 无差错情况

图 6.3 akiyo 序列、10%丢包率、10%Inter 宏块保护的視頻点播系統

从图 6.2 和图 6.3 可以看到，本文提出的差错控制方法对视频起到了很好的保护作用。对于视频监控系統来说，由于源视频类似于 akiyo 序列，并且有可能会有更多静止的块，因此，保护效果必然会更好。

6.3 本章小结

本章搭建了一个面向移动环境的视频应用系統，并在其上使用了基于重要块保护的容错编码，从而直观地衡量了基于重要块保护方法的效果。可以看到，本文提出的容错编码方法在不同的丢包率下对于不同的视频均有较好的保护作用。

第七章 总结与展望

7.1 总结

随着多媒体技术和应用的不断发展,结合网络传输而开发的视频应用越来越受到人们的欢迎。但是,在承载力有限的当前可用网络上传输巨大的压缩视频数据仍然不是一件简单的事情,特别是在 Internet 和无线网上。为了使压缩视频数据能够在网络上顺利传输,特别针对传统的面向视频存储的视频编解码系统做了一定的改进,加进了一些面向视频传输的特性。这些特性的根本目的,就是为了提高视频数据在网络上传输的鲁棒性,它是以牺牲一定的压缩效率为代价的。其中,差错控制就是其中一种重要的方法。

本文首先回顾了传统的视频编解码系统及相关标准,并分析了视频编解码系统的主要相关技术。然后分析了压缩视频流在网络上传输的特点,及当前针对压缩视频数据在网络上的传输,已有的标准族和协议等,并分析了为保证视频传输 QoS 的常用控制方法。本文在后几章的内容中着重介绍和分析了差错控制机制。在总结和分析传统的差错控制方法的基础上,提出了基于重要块保护的视频压缩容错编码方法。该方法可以分为三步:

1. 基于容错特性的快速重要块决策步骤
2. 基于重要块保护的容错编码步骤
3. 基于重要块检测的差错隐藏步骤

基于重要块保护的容错编码之所以快速,是因为其预先建立了一个概率统计模型,为参与决策的贡献因子统计出影响因素值,并通过训练得到各因子参与决策的权重值。在容错编码阶段,只需使用通过训练已得到的各参数值,近似模拟出各宏块丢失的影响值,并将该值作为该宏块的重要度值。最后,选取那些重要度最高的宏块作为重点保护的块。

本文提出的方法主要有以下几个特点:

1. 本文提出的基于重要块保护的容错编码适用于一般的视频序列。
2. 用于决策重要块的参数都是预先通过统计模型训练出来的,因此,在容错编码阶段,只需通过少量的计算就可以决策出重要块。这种特性使得本文提出的方法能够适用于实时应用的场景。
3. 在重要块保护阶段,本文的方法只保护那些重要的块区域,通过综合使用 Intra 刷新方法、MDC 保护方法和多帧参考/帧内预测方法来保护 Inter 数据。这样,使得仅仅通过少量的额外比特,就可以获得视频编码流鲁棒性的提升。
4. 在解码端,差错隐藏的方法也比较简单。因此,适用于移动设备。

经过实验验证,得到:

1. 基于容错特性的快速重要块决策方法能够快速并且准确地决策出重要块区域,即使有一定的匹配失误,那些失误块也一般是次重要的块。
2. 基于重要块保护的容错编码能够使得编码码流的鲁棒性显著提高,不论是高运动的视频还是低运动的视频。

7.2 展望

由于本文提出的基于统计模型的重要块决策和保护方法目前在其他论文中还没有出现过,因此,在这方面仍然有许多工作要做。此外,本文提出的重要块决策方法和重要块保护方法也有一些方面需要提高。这些工作主要分为以下几个部分:

1. 本文提出的重要块决策模型中,认为一个宏块的重要度和其贡献因子之间是线性的关系。基于这种关系的决策模型虽然已经保证了一定的决策精度,但是仍然有待提高。通过更好地挖掘各贡献因子之间的关系,而设计一种更准确的快速重要块决策模型,对容错编码来说,无疑是关键。
2. 对于视频点播等基于 C/S (客户端/服务器) 的视频服务中,还可以将重要块决策模型改进成一个学习型的概率统计模型,动态地调节各贡献因子的影响值和权重值,使其更逼近于实际的情况。
3. 如果能够通过上一编码帧的信息预测出当前帧的 Mode、MV 和 SAD,则完全可以在编码过程之前通过前处理决策出重要块,那么,非线性的保护方法就可以应用于重要块保护方法中。
4. 由于基于统计模型的重要块决策和保护方法预先考虑了解码端的差错隐藏方法,因而对于不同的差错隐藏方法,决策出的重要块是不同的。鉴于此,有必要对各种常用的差错隐藏方法都做一下比较,从而找出一种最佳的差错隐藏方法。

参考文献

- [1] 林福宗, *多媒体技术基础* (第二版), 北京, 清华大学出版社, 2002
- [2] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing (Second Edition)*, Pearson Education, Inc., publishing as Prentice Hall, U.S.A., 2002
- [3] Iain E. G. Richardson, *Video Codec Design – Developing Image and Video Compression Systems*, John Wiley & Sons, Ltd, Chichester, England, 2002
- [4] ISO/IEC 10918-1 /ITU-T Recommendation T.81, *Digital Compression and Coding of Continuous-Tone Still Images*, 1992
- [5] ISO/IEC FCD 15444-1, *JPEG2000 Final Committee Draft v1.0*, Mar. 2000
- [6] ISO/IEC 11172-2, *Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5 Mbit/s-Part 2: Video*, 1993
- [7] ISO/IEC 13818-2, *Information Technology: Generic Coding of Moving Pictures and Associated Audio Information: Video*, 1995
- [8] ISO/IEC 14996-2, *Information Technology-Coding of Audio-Visual Objects-Part 2: Visual*, 1998
- [9] ISO/IEC JTC1/SC29/WG11 N403 1, *Overview of the MPEG-7 Standard*, Singapore, Mar. 2001
- [10] ISO/IEC JTC1/SC29/WG11 N4318, *MPEG-21 Overview*, Sydney, July 2001
- [11] ITU-T Recommendation H.261, *Video Codec for Audiovisual Services at px64 kbit/s*, Mar.1993
- [12] ITU-T Recommendation H.263, *Video Coding for Low Rate*, May 1996
- [13] ITU-T SG16, G. Sullivan, etc., *Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision*, Jan. 1998
- [14] ITU-T Recommendation H.263v3, *Video Coding for Low Bitrate Communication*, Nov. 2000
- [15] Draft ITU-T Recommendation H.264 (a.k.a. "H.26L"), Joint Video Team (JVT) of ISO/IEC MPEG and ITU-TVCEG, *Document JVT C167*, May 2002
- [16] <http://www.avs.org.cn/>
- [17] ITU-T Recommendation H.320, *Narrow-Band Visual Telephone Systems and Terminal Equipment*, July 1997
- [18] ITU-T Recommendation H.321, *Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments*, 1996
- [19] ITU-T Recommendation H.323v2, *Packet Based Multimedia Communications Systems*, Mar. 1997
- [19] ITU-T Recommendation H.323v2, *Packet Based Multimedia Communications Systems*, Mar. 1997

- [20] ITU-T Recommendation H.324, *Terminal for Low Bit-Rate Multimedia Communication*, Feb. 1998
- [21] ITU-T Recommendation H.221, *Frame Structure for a 64 to 1920Kb/s Channel in Audiovisual Teleservices*, July 2000
- [22] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, *SIP*, RFC2543, Mar. 1999
- [23] M. Handley and V. Jacobson, *SDP: Session Description Protocol*, RFC2327, Apr. 1998
- [24] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-time Applications*, RFC1889, Jan. 1996
- [25] ITU-T Recommendation H.223, *Multiplexing Protocol for Low Bit Rate Multimedia Communication*, Mar. 1996
- [26] 3GPP TR 26.110, *Codec for Circuit Switched Multimedia Telephony Service: General Description*, June 2002
- [27] 3GPP TR 26.235, *Packet Switched Conversational Multimedia Applications: Default Codecs*, Sep. 2004
- [28] 3GPP TR 26.233, *Transparent End-to-end Packet-switched Streaming Service (PSS), General Description (Release 6)*, Dec. 2004
- [29] 3GPP TR 26.140, *Multimedia Messaging Service (MMS); Media Formats and Codecs*, Dec. 2002
- [30] 3GPP TR 29.846, *Multimedia Broadcast/Multicast Services*, Sep. 2004
- [31] Y. Yang, *Rate Control for Video Coding and Transmission*, Ph.D. Thesis, Cornell University, 2000
- [32] A. Vetro, C. Christopoulos, and H. Sun, *Video Transcoding Architectures and Techniques: an Overview*, IEEE Signal Processing Magazine, Mar. 2003
- [33] W.-M. Lam and A. Reibman, *An Error Concealment Algorithm for Images Subject to Channel Errors*, IEEE Trans. Image Processing, vol. 4, pp. 533-542, May 1995
- [34] M. Chen, Y. He, etc., *A Fragile Watermark Error Detection Scheme for Wireless Video Communications*, IEEE Trans. Multimedia, vol. 7, no. 2, pp. 201-211, Apr. 2005
- [35] K. N. Ngan and R. Stelle, *Enhancement of PCM and DPCM Images Corrupted by Transmission Errors*, IEEE Trans., Commun., vol. 30, pp. 257-265, Jan. 1982
- [36] K. M. Rose and A. Heiman, *Enhancement of One-dimensional Variable-length DPCM Images Corrupted by Transmission Errors*, IEEE Trans. Commun., vol. 37, pp. 373-379, Apr. 1989
- [37] Y. Wang and Q. Zhu, *Error Control and Concealment for Video Communication: a Review*, Proc. IEEE, vol. 86, pp. 974-997, May 1998

- [38] J. D. Villasenor, Y. Q. Zhang, and J. Wen, *Robust Video Coding Algorithms and Systems*, Proc. IEEE, vol. 87, pp. 1724-1733, Oct. 1999
- [39] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, *Error Resilient Video Coding Techniques: Real-Time Video Communications over Unreliable Networks*, IEEE Signal Processing Magazine, pp 61-82, July 2000
- [40] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, *Error Resilience Support in H.263+*, IEEE Trans. Circuit Syst. Video Technol., vol. 8, no. 6, pp. 867-877, Nov. 1998
- [41] R. Talluri, *Error-Resilient Video Coding in the ISO MPEG-4 Standard*, IEEE Communications Magazine, vol. 36, no. 6, pp. 112-119, June 1998
- [42] S. Gringeri, R. Egorov, K. Shuaib, A. Lewis, and B. Basch, *Robust Compression and Transmission of MPEG-4 Video*, Proc. of ACM Multimedia, pp. 113-120, 1999
- [43] S. Kumar, L. Xu, M. K. Mandal and S. Panchanathan, *Error Resiliency Schemes in H.264/AVC Standard*, Journal of Visual Communication and Image Representation, vol. 17, no. 2, pp. 425-450, Apr. 2006
- [44] T. Halbach and S. Olsen, *Error Robustness Evaluation of H.264/MPEG-4 AVC*, Visual Communications and Image Processing 2004, Proc. of SPIE-IS&T Electronic Imaging, vol. 5308, pp. 617-627, 2004
- [45] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, *H.264/AVC in Wireless Environments*, IEEE Trans. Circuit Syst. Video Technol., vol. 13, no. 7, pp. 657-673, July 2003
- [46] S. Wenger, *H.264/AVC over IP*, IEEE Trans. Circuit Syst. Video Technol., vol. 13, no. 7, pp. 645-656, July 2003
- [47] J. Wen, J. Villasenor, and H. D. Shin, *Proposal for Error Resilience in H.26L*, ITU-T documentation Q15-C-36, Eibsee, Germany, 1997
- [48] W. R. Heinzelman, M. Budagavi, R. Talluri, *Unequal Error Protection of MPEG-4 Compressed Video*, Proc. of Int. Conf. Image Proc. 99, vol. 2, pp. 530-534, Oct. 1999
- [49] A. Jerbi, J. Wang, S. Shirani, *Error Resilient ROI Coding Using Pre- and Post-Processing for Video Sequences*, IEEE International Symposium on Circuits and Systems, vol. 3, pp. 757-760, 2004
- [50] P. G. Sherwood and K. Zeger, *Progressive Image Coding on Noisy Channels*, Proc. of Data Compression Conf. '97, pp. 72-81, Mar. 1998
- [51] V. A. Vaishampayan, *Design of Multiple Description Scalar Quantizers*, IEEE. Trans. Inform. Theory, vol. 39, pp. 821-834, May 1993
- [52] X. Yang and K. Ramchandran, *Optimal Multiple Description Subband Coding*, Proc. of IEEE Int. Conf. Image Proc., vol. 1, pp 684-658, Oct. 1998

- [53] Y. Wang, A. R. Reibman, and S. Lin, *Multiple Description Coding for Video Delivery*, Proc. of the IEEE, vol. 93, no. 1, pp. 57-70, Jan. 2005.
- [54] G. Core, S. Shirani, and F. Kossentini, *Optimal Mode Selection and Synchronization for Robust Video Communications over Error Prone Networks*, IEEE J. Select. Areas Commun., vol. 18, pp. 952-965, June 2000
- [55] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, *Error-resilient Video Transmission Using Long-term Memory Motion-Compensated Prediction*, IEEE J. Select. Areas Commun., vol. 18, pp. 1050-1062, Dec. 2000
- [56] M. C. Hong, L. Kondi, H. Scwab, and A. K. Karsaggelos, *Video Error Concealment Techniques*, Signal Processing: Image Communications, vol. 14, nos. 6-8, pp. 437-492, 1999
- [57] Y. Xu and Y. Zhou, *H.264 Video Communication Based Refined Error Concealment Schemes*, IEEE Trans. Consumer Electronics, vol. 50, no. 4, pp. 1135-1141, Nov. 2004
- [58] A. Yoneyama, H. Yanagihara, Y. Takishima, and Y. Nakajima, *Video Error Concealment by the Combination of Spatio-temporal Recovery and Post-filter for H.264*, IEEE Trans. Consumer Electronics, no. 9.3-2, pp. 361-362, 2005
- [59] P. J. Lee, H. H. Chen, L. G. Chen, *A New Error Concealment Algorithm for H.264 Video Transmission*, Proc. of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 619-622, Oct. 2004
- [60] W. M. Lam, A. R. Reibman, and B. Liu, *Recovery of Lost or Erroneously Received Motion Vectors*, Proc. International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, pp. 417-420, 1993
- [61] B. Girod and N. Färber, *Feedback-Based Error Control for Mobile Video Transmission*, Proc. of IEEE, vol. 87, no. 10, pp. 1707-1723, Oct. 1999
- [62] G. S. Ma, M. S. Chen and J. C. Chen, *Error-Resilient Real-Time Video Communications over IP-Based LANs*, IEEE International Conference on Multimedia and Expo, pp. 80-83, Aug. 2001
- [63] H. Yu, C. Wang and S. Yu, *A Novel Error Recovery Scheme for H.264 Video and Its Application in Conversational Services*, IEEE Trans. Consumer Electronics, vol. 50, no. 1, pp. 329-334, Feb. 2004
- [64] A. Jerbi, J. Wang, and S. Shirani, *Error-Resilient Region-of-Interest Video Coding*, IEEE Trans. Circuits Syst. Video Technol., vol. 15, pp. 1175-1181, Sep. 2005
- [65] P. Baccichet, X. Zhu, and B. Girod, *Network-Aware H.264/AVC Region-of-Interest Coding for a Multi-Camera Wireless Surveillance Network*, Proc. Picture Coding Symposium (PCS-06), April 2006
- [66] H. Chiou, Y. Lee, C. Lin, *Error-resilient Transcoding Using Adaptive Intra Refresh for Video*

Streaming, Proc. of the 2004 International Symposium, Circuits Syst., vol. 3, pp. 777-780, May 2004

[67] J. Chen, C. S. Lu, and K. C. Fan, *A Significant Motion Vector Protection-based Error-Resilient Scheme in H.264*, IEEE 6th Workshop on Multimedia Signal Processing, pp. 287-290, 2004

[68] <http://iphome.hhi.de/suehring/ttml/download/jm10.2.zip>

[69] R. Schreier, A. Rothermel, *Motion Adaptive Intra Refresh for Low-Delay Video Coding*, International Conference on Consumer Electronics, pp. 453-454, Jan. 2006

[70] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, *Multiple Description Video Coding for Scalable and Robust Transmission Over IP*, IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 3, pp. 321-334, Mar. 2005

[71] L. Liu, S. Zhang, X. Ye, and Y. Zhang, *Error Resilience Schemes of H.264/AVC for 3G Conversational Video Services*, Proceedings of the Fifth International Conference on Computer and Information Technology, pp. 657-661, 2005

致 谢

把泪水抛于过去，把欢笑留在现在，把希望寄托未来！

不知不觉，两年的硕士生涯就要结束了，快得有些让人措手不及。想起就要离开待了六年的求是园，想起就要告别十多年的校园生活，不禁感慨万千。

首先，我要感谢陈纯教授，是您给我们创造了一个良好的学习和科研氛围，带动了实验室学术研究的不断进步。尽管与您接触的机会不多，但每一次的接触都是给我心灵的一次洗礼。您那亲切的微笑，您那缜密的思维，您那严谨的态度，您那饱满的激情，都给我留下了深刻的印象。您说的每一句警醒和忠告，我都深深地牢记在心里。有着它们，我才能更好地把握好自己人生要走的方向。

其次，我要深深感谢卜佳俊副教授一直以来对我的关怀和照顾。自从 2003 年暑期课程实践起，我就与您结下了不解之缘。您一丝不苟的精神给我留下了深刻的印象，让我在确信自己考上研究生后，毫不犹豫地选择您作为我研究生阶段的导师。在研究生阶段，我又经历了许多彷徨和困惑。也是您，在我最需要关心的时候给了我指点，让我重新找到了自己该走的方向。我会珍惜与您相处的这段时间，因为我知道，您带给我的，不仅仅是知识与智慧，还有那些值得用一生去品味的人生哲理。

在 MMG 的这段日子，是我一生中最重要的时光。我要特别感谢杨智博士、莫林剑博士和李娜博士，因为我知道，没有你们的勤劳和努力，就没有今天 MMG 的辉煌。我还要感谢 MMG 的所有成员，跟你们一起奋斗的日子令我终生难忘，从你们身上我学到了很多，我会永远记得你们的名字：谭永宝、段元亮、姜水勇、李旭、黄笑鹏、钱伟红、邓小宇、吕学良、虞俊杰、刘迪博、何斌、潘丹、洪朝群和陈蒲苇。

童年对于我来说是一段不堪回首的经历，饱受压抑和自卑。幸而这么多年来，一直都遇到了特别照顾我的师长和朋友，让我能够幸运地走到今天，感谢身边所有关心和照顾过我的人。这里我要特别感谢我的大哥，中企实业的副总李卫阳，你教给了我许多做人的道理，我会永远记住的。我还要感谢我的室友韩伟、施律和吕攀，六年相处的时光让我们结下了兄弟般的情谊。

最后，我要感谢我的妈妈、大姐和二姐，我知道，为了我，你们牺牲了很多很多。我会用一辈子来报答你们对我的爱 and 付出！

邵 根 富

二〇〇六年五月于求是园