

摘 要

MPEG-4 是运动图像专家组织制定的新一代多媒体压缩国际标准,它提供了一系列自然和合成的音视频编码工具。标准采用基于对象的编码技术、可扩展编码技术和容错机制等,实现了基于内容的交互性和更高的压缩效率。它将多媒体的音视频应用扩展到了前所未有的领域。

本文首先对 MPEG-4 标准和其校验模型 (VM) 的关键技术进行了阐述。然后,对 VM 中的形状编码过程进行了深入分析。利用宏块之间的相关性和形状信息的特点,提出了一种形状编码的快速运动估计算法;针对在不同误差阈值下,尺寸转化的宏块数目和比率不同的特点,对 VM 中形状编码的尺寸转化算法进行了改进。并且结合精细可扩展编码的思想,利用形状信息和四叉树的特点,提出了一种新的二值形状空域可扩展编码算法。

最后,结合自己的研究,对视频编码的进一步发展和形状编码进一步研究提出了自己的一些看法。

关键词: MPEG-4 形状编码 运动估计 尺寸转化 可扩展编码

ABSTRACT

MPEG-4 is a new generation ISO/IEC standard for multimedia communication developed by the Moving Pictures Expert Group (MPEG), and provides a set of tools for natural and synthetic video/audio coding. The MPEG-4 visual standard supports many techniques included object-based coding, scalable coding, and error resilient coding. The main functions provided by MPEG-4 are content-based interactivity, universal accessibility, and improved compression. MPEG-4 spreads the video/audio application of multimedia to everywhere.

The pivotal technology of MPEG-4 and its Verification Mode (VM) are described first in this paper. Then a deep analysis of shape coding in VM is performed. A fast and efficient motion estimation algorithm based on the features of video motion vector and shape information is proposed to improve search speed in motion estimation for mpeg-4 shape coding. A modify size conversion algorithm is proposed to reduce computational complexity for shape coding. A new shape spatial scalable coding algorithm for MPEG-4 is proposed based on the features of video's shape information and quad-tree.

Some ideas for the new development of video coding and research of shape coding are proposed based on the researches of this paper at last.

**Key word: MPEG-4 Shape Coding Motion Estimation Size Conversion
Scalable Encoding**

创新性声明

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他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，本人承担一切相关责任。

本人签名： 李高林 日期： 2004.1.8

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。本人保证毕业后离校后，发表论文或使用论文成果时署各单位仍然为西安电子科技大学。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内容，可以允许采用影印、缩印或其他复制手段保存论文。（保密的论文在解密后遵守此规定）

本人签名： 李高林 日期： 2004.1.8

导师签名： 姜星明 日期： 2004.1.8

第一章 绪论

1.1 多媒体编码技术的发展和概况

21 世纪的人类社会将是信息化和网络化的社会,网络信息具有数字化和海量性的特点。网络应用的最重要目标之一就是进行多媒体通信。多媒体信息主要包括图像、声音和文本三大类,其中视频、音频等信号的信息量非常大,它们的存储和传输给网络带来很大的困难,成为阻碍网络多媒体发展的瓶颈之一。同时,各种视频、音频信息的表达方式不同,在不同的应用环境下对内容的侧重点要求不同,不同的来源还可能有不同的输入、输出要求。因此,研究和开发新型有效的多媒体数据压缩编码方法并建立国际标准是最好的选择。

传统的压缩编码是建立在香农(Shannon)信息论的基础上,它以经典的集合论为基础,用统计概率模型来描述信源,但是,它未考虑信息接受者的主观特性及信息本身的具体含义、重要程度和引起的后果。压缩编码的发展历程实际是以香农信息论为出发点,不断完善和改进的过程,编码方法可以分成下述三类:

考虑信源的统计特性:预测编码方法,变换编码方法,矢量编码方法,矢量量化编码方法,子带一小波编码方法,神经网络编码方法等;

考虑人眼的视觉特性:基于方向滤波的图像编码方法,基于图像轮廓一纹理的编码方法;

考虑图像传递的景物特性:分形编码,基于内容的编码方法。

也有人将图像编码方法分为两代:第一代是基于数据统计,去掉的是数据冗余,称为低层压缩编码方法;第二代是基于内容,去掉的是内容冗余,其中基于对象(Object-Based)方法称为中层压缩编码方法,其中基于语义(Syntax-Based)方法称为高层压缩编码方法。基于内容压缩编码方法代表新一代的压缩方法,也是目前最活跃的领域,最早是由瑞典的 Forchheimer(1983 年)提出的,随后日本的 Harashima 等人也展示了不少研究成果^[1]。

为了实现多媒体传输的国际标准化,国际标准化组织 ISO(International Standard Organization)与 IEC(International Electronic Committee)于 1988 年联合成立了 MPEG(Moving Picture Expert Group)组织。MPEG 的任务是致力于运动图像及其伴音编码标准化工作。MPEG 先后推出和正在建立的标准有 MPEG-1^[1]、MPEG-2^[2]、MPEG-4^[7]、MPEG-7^[3]、MPEG-21^[4]等,不同标准具有不同的用途和解码质量。MPEG 标准的出现和发展对多媒体通信的发展起到了革命性的推动作用。

MPEG 标准一直是许多科研机构 and 大学的科研热点,也是工业界产品开发的热点。MPEG 标准阐明了声音和电视图像的编码和解码过程,严格定义了声音和图像数据编码后组成位数据流的句法,提供了编解码的测试方法等;但没有对所有内容都作严格规定,尤其是对压缩和解压缩的算法,这样既保证了解码器能对符合 MPEG 标准的声音数据和电视图像数据进行正确解码,又给 MPEG 标准的具体实现留有很大余地。人们可以不断改进编码和解码算法,提高声音和电视图像的质量以及编码效率。

MPEG-1 于 1993 年成为国际标准,它是对 1.5Mbit/s 以下数据传输率的数字存储媒体运动图像及其伴音的压缩编码标准,适用于 CD-ROM、VCD、CD-I (交互式 CD) 等。它可对 SIF (标准交换格式) 分辨率(NTSC 制式为 352×240 ; PAL 制式为 352×288) 的图像进行压缩,传输速率为 1.5Mbit/s,每秒播放 30 帧,具有 CD 音质,图像质量基本与 VHS 家用录像机相当。MPEG-1 也被用于数字通信网络上的视频传输,如基于 ADSL (非对称数字用户线路) 的视频点播(VOD)、远程教育等。

MPEG-2 于 1995 年成为国际标准,其目标是达到高级工业标准的图像质量以及更高的传输率。MPEG-2 所能提供的传输率在 3~10Mbit/s 之间,在 NTSC 制式下的分辨率可达 720×486 ,可提供广播级的图像质量和 CD 级的音质,适用于数字电视广播(DVB)、HDTV 和 DVD 的运动图像及其伴音的压缩编码。目前,MPEG-2 已得到广泛应用,如美国、欧洲、日本在 DVD 和数字电视广播方面都采用 MPEG-2 压缩技术。

MPEG-4 从 1994 年开始工作,于 1999 年初正式成为国际标准。MPEG-4 是一个适用于低传输速率应用的方案。与 MPEG-1 和 MPEG-2 相比,MPEG-4 的目标是要在异构网络环境下能够高度可靠地工作,并且更加注重多媒体系统的交互性和灵活性。

继 MPEG-4 之后,要解决的矛盾就是对日渐庞大的图像、声音信息的管理和迅速搜索。针对这一矛盾,1998 年 10 月,MPEG 启动了一个新的工作项目,即“多媒体内容描述接口(Multimedia Content Description Interface)”—MPEG-7。它的目标是扩展现有系统有限的查询能力,使其包括更多的信息形式。MPEG-7 将确立各种类型的多媒体信息标准的描述方法。MPEG 在 1999 年 10 月的墨尔本会议上又提出了“多媒体框架(Multimedia Framework)”—MPEG-21 的概念,并于 2000 年 7 月开始工作。MPEG-21 总体上来讲是一个支持通过异构网络和设备使用户透明而广泛地使用多媒体资源的标准,其目标是建立一个交互的多媒体框架。

在推出 MPEG-1 和 MPEG-2 的同时,另一个国际组织 CCITT (International Telegraph and Telephone Consultative Committee-国际电话电报咨询委员会),后改名叫 ITU-T(International Telecommunications Union-Telecommunication Sector)

也先后推出了 H.261 和 H.263^[5]两个用于传输数字视频信号的压缩标准。在 H.263 之后,ITU-T 视频编码专家组 (Video Coding Experts Group, VCEG) 的短期任务是对 H.263 标准的扩充制定 H.264^[13]标准,长期目标是低速率视频传输标准 H.26L。在 2001 年,MPEG 组织成立视频联合组 (Joint Video Team, JVT)。JVT 由 MPEG 和 VCEG 的一部分专家组成,负责将原来的 MPEG-4 的第 10 部分和 H.264 合并为 MPEG-4 的第 10 部分,这个标准又称为先进的视频编码^[13](Advanced Video Coding, AVC)。

我国在视音频编码的标准制定方面发展也很迅速。我国制定数字视音频编解码标准,目的不仅仅是为了把专利许可费控制在合理水平上,更重要的意义是开发出技术更先进的标准,从源头上提高我国数字音视频产业的核心竞争力,为我国数字电视音视频产业和相关芯片产业提供跨越发展的技术源头,为国际数字音视频市场提供更好的选择。国家信息产业部科学技术司于 2002 年 6 月批准成立数字音视频编解码技术标准工作组^[6] (Audio Video Coding Standard Workgroup of China,简称 AVS 工作组)。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。AVS 工作组制定的 AVS(《信息技术 先进音视频编码》)标准与国际标准 MPEG-2 兼容,并针对数字电视、光盘播放机、网络流媒体、多媒体通信等应用进行了具体的规定和定义,从框架上提供了对国内外各种主流视频、音频编码标准的支持。

按照编码方法发展的两代论,目前已经制定成熟的视频压缩编码标准 (MPEG-1、MPEG-2、H.261、H.263) 都属于第一代压缩编码方法——基于数据统计的压缩编码方法,而正在制定的 MPEG-4 和 MPEG-7 属于第二代压缩编码方法中的基于对象的压缩编码方法。

1.2 本文的研究内容

我国多媒体编解码标准 (AVS) 的制定和发展在很多内容上是借鉴和吸收国际上已经成功的标准的结果,所以研究国际先进标准对国内标准的发展非常重要。尽管 MPEG-4 标准版本已经制定出来,但是标准中所包括的编解码算法和技术要应用于实际还有许多问题需要解决,同时 MPEG-4 是一个开放的标准,它的具体各部分的算法实现是一个逐步改进和完善的过程。

编码速度、图像质量和压缩效率等是在标准实用化阶段需要考虑的问题,本文的研究基础是目前新制订的 MPEG-4 标准,研究内容是视频编码中的形状编码

编解码算法和可扩展编解码算法。在深入研究的基础上,本文对 MPEG-4 标准提供的校验模型中的形状编码的某些部分提出了改进算法,并用 MPEG-4 标准建议的测试序列进行了测试,取得了满意的效果。

主要研究内容如下:

1、对 MPEG-4 标准和其校验模型 (VM) 的关键技术进行了介绍。对 VM 中的形状编码算法进行了详细分析。

2、提出了一种快速的形状编码运动估计算法。此算法利用相邻块的相关性确定搜索的初始点,根据初始预测块和当前块的绝对误差和 (Sum of Absolute Difference, SAD) 选择要采用的搜索算法。实验表明此算法计算复杂度低,搜索范围大,预测精度高。

3、针对在不同误差阈值下,尺寸转化的宏块数目和比率不同的特点,对 MPEG-4VM 中形状编码的尺寸转化 (确定分辨率) 算法进行了改进。采用改进算法可以提高编码速度,减少运算复杂度,改善编码质量。

4、提出了一种新的空域可扩展形状编码算法。利用形状信息的特点和四叉树的原理来实现形状信息编码的渐进的可扩展分层编码。这种算法实现简单,可以很好的实现可扩展的编码思想。

1.3 章节的安排

本文的章节安排如下:

第一章 对多媒体编码的发展作简单介绍,并介绍了作者的主要工作。

第二章 对 MPEG-4 标准和其校验模型 (VM) 内容作简要介绍。

第三章 对 MPEG-4 形状编码做详细分析论述,并对本文提出的运动估计改进算法和尺寸转化改进算法作了详细的阐述。

第四章 对可扩展编码作简要介绍,对提出的一种新的空域形状可扩展编码算法做详细的阐述。

第五章 结束语,对 MPEG 标准的发展进行了展望,对本文的研究工作进行了总结。

附录和参考文献。

第二章 MPEG-4 视频编码

2.1 MPEG-4 技术及应用综述

2.1.1 MPEG-4 标准概要

MPEG-4 标准不仅针对一定比特率下的视频、音频编码，并且更加注重多媒体系统的交互性和灵活性。

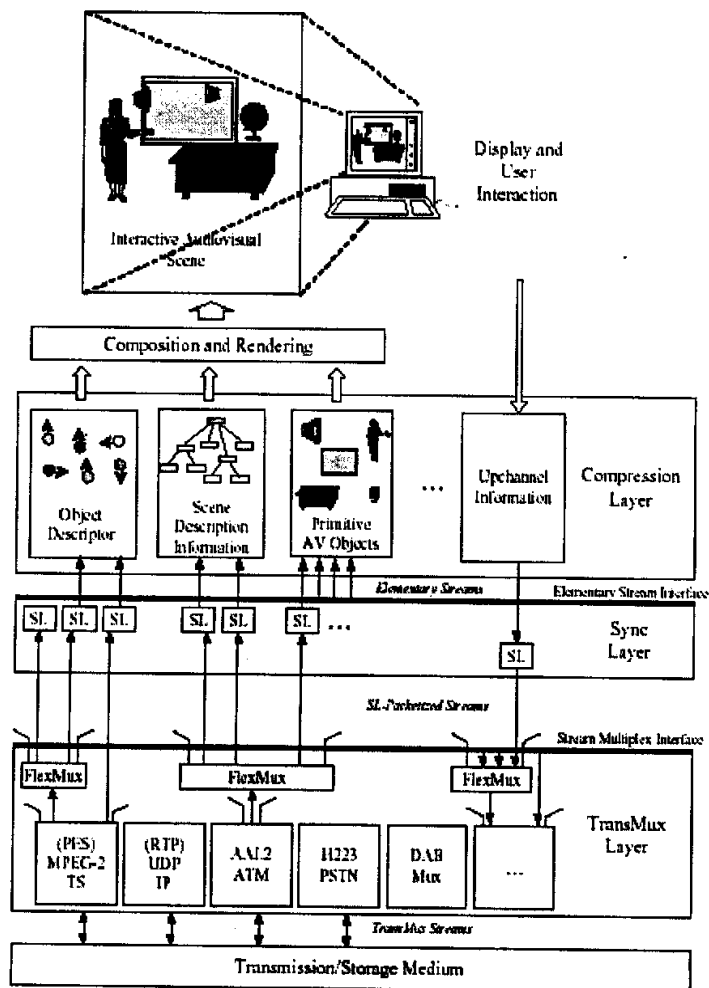


图 2.1 一个 MPEG-4 视听终端的处理过程

MPEG-4 标准不像以前的多媒体标准一样是基于帧的编码，而是基于对象的编码。对于观察者而言，一幅场景总可以划分成一些不同的内容及背景，每一部

分的内容对应一个独立的音频/视频 (Audio Video) 对象。这些 AV 对象都拥有自己单独的编解码器, 用户可以对场景中的内容进行交互操作。因此, MPEG-4 标准的基本内容就是高效率地编码、组织、存储、传输 AV 对象。

图 2.1 显示了一个 MPEG-4 视听终端的处理过程。一个场景 (包括一个站着的女士、一个桌子、一个地球仪、一个在其上有多媒体演示的“电子白板”等等) 通过对象分割、编码压缩、增加同步信息, 然后传输到用户端。用户端接收到这些信息后根据同步信息解压缩, 然后既可以根据传送过来的组合信息恢复原始场景, 也可以根据自己的需要选择要恢复的 AV 对象, 生成新的场景。

MPEG-4 提供了对音频对象、视频对象、情景描述以及与发送系统的接口进行编码的标准方法。由于在解码端要对 AV 对象进行重新组合, 所以 MPEG-4 既支持自然的 AV 对象编解码也支持合成的 AV 对象编解码。

2.1.2 MPEG-4 标准的主要特点和功能

MPEG-4 视频编码标准支持 MPEG-1、MPEG-2 中的大多数功能, 提供不同的视频标准源格式、码率、帧频下矩形图像的有效编码, 同时也支持基于内容的图像编码。

如图 2.2, 在 MPEG-4 功能集的底层是 VLBV (Very Low Bit Rate Video) 核心。它为码率在 5—64kbps 范围内的视频操作与应用提供算法与工具, 支持较低的空间分辨率 (低于 352×288 像素) 和较低的帧频 (低于 15Hz)。VLBV 核心支持的专用功能包括: 矩形图像序列的有效编码、多媒体数据库的搜索和随机存取。

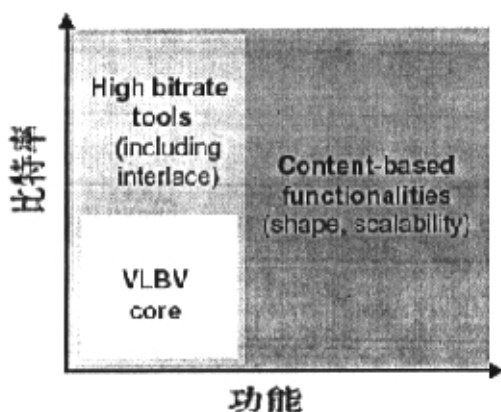


图 2.2 MPEG-4 视频编码工具结构

MPEG-4 的 HBV (High Bit Rate Video) 同样支持上述功能, 其码率范围在 64kbps—10Mbps 之间, 它与 VLBV 核心采用相同或相似的算法, 但它支持更高

的空间与时间分辨率，允许传输和存储适用于演播室的高质量视频信号，典型应用为数字电视广播与交互式检索。MPEG-4 最终支持的码率将高于 MPEG-2。

MPEG-4 包括如下几部分主要功能：

(1) 基于内容的编码

基于内容的交互性；基于内容的多媒体数据存取工具；基于内容的码流操纵和编辑；自然与合成数据的混合编码；增强的时间域随机存取。

(2) 编码效率的改进和并发数据流的编码

提高编码效率；对多个并发数据流的编码，支持对同一场景多视点的有效编码；对于立体视频应用，要求具有利用信息冗余的能力，并支持有无与正常视频兼容性要求条件下的联合编码方案。

(3) 容错机制

错误易发环境中的错误鲁棒性 (Robustness) 和错误修复 (resilience) 技术。

(4) 基于内容的可伸缩性 (scalable)。

包括基于内容的空域可扩展编码和时域可扩展编码。

2.1.3 MPEG-4 的体系结构

MPEG-4 标准主要有 6 个部分：系统标准、视频标准、声音标准、一致性测试、参考软件和多媒体集成框架。其核心部分有四个模块：系统层、视频、音频和 DMIF^[1] (Delivery Multimedia Integration Framework)。其中前三部分是独立于传输的，而 DMIF 则定义了面向传输的特性。另外还有两个支持模块：一致性测试 (Conformance Testing) 和参考软件 (Reference Software)。

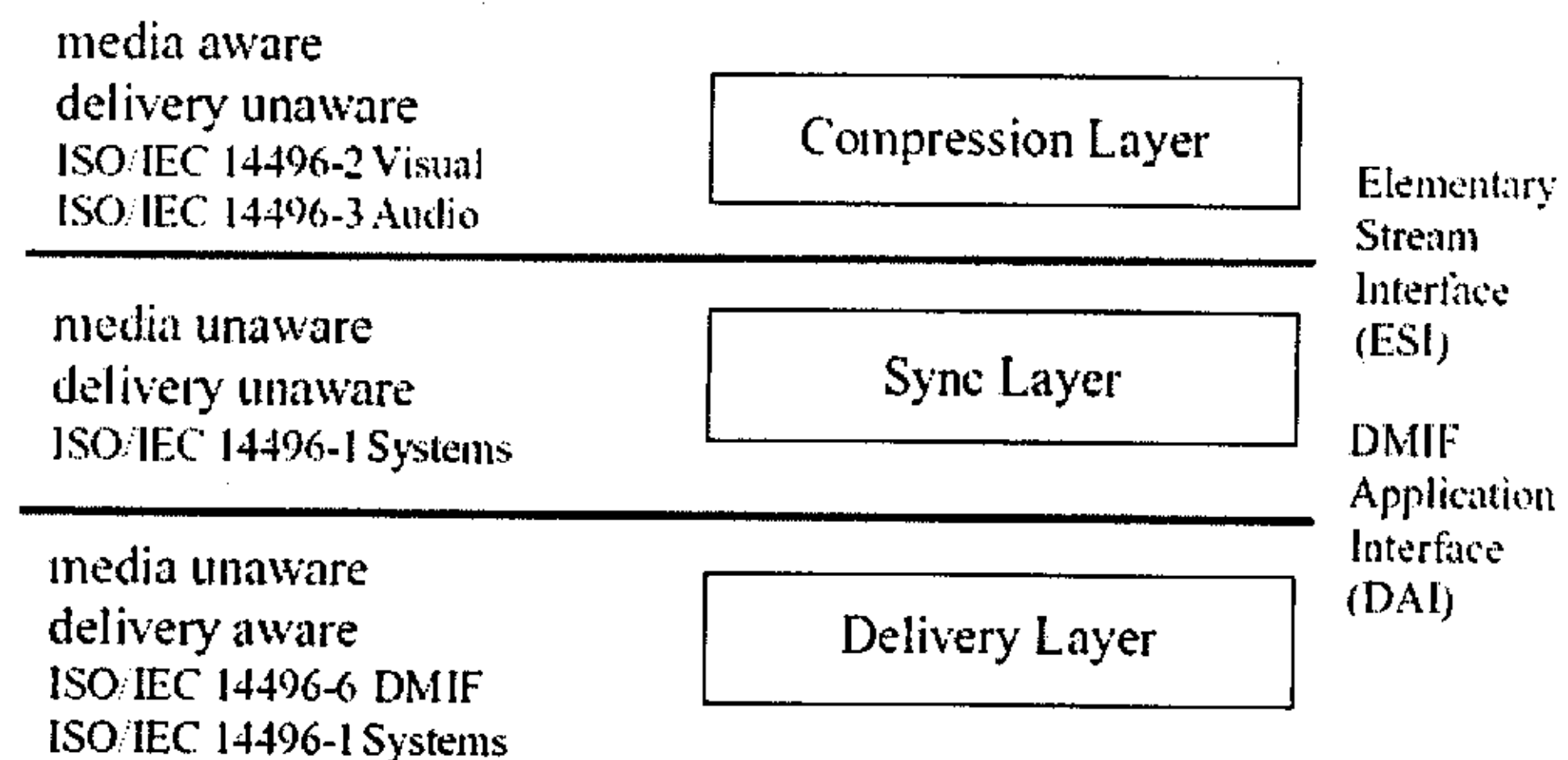


图 2.3 MPEG-4 标准层次结构

从层次上来说，四个核心模块可以划分成三个层次，如图 2.3 所示：传输层 (Delivery Layer) 定义：

传输层是对已经存在的各种传输协议的一般描述，这些协议能够用来传输和存储符合 MPEG-4 标准的视听内容。传输层的功能并不在系统流内部规定，只是和传输层有关的接口需要考虑，这个接口就是传输多媒体集成框架 DMIF 应用接口，由标准中的 DMIF 部分定义。

同步层（Sync Layer）定义：

各个数据源的经过压缩得到的数据主要是基本码流数据，它们包含内容信息，同步层将这些数据和同步信息封装成同步数据包流，再将它们传输到传输层（DMIF），这些数据包中不仅包含有定时和同步数据，而且还有随机访问信息。另一方面，同步层从传输层接受数据流，从流中提取同步数据，为以后同步解码和基本流解码的合成做准备。

压缩层（Compression Layer）定义：

压缩层在编码端将场景中的不同音视频对象进行压缩编码，产生对应的基本码流数据。在解码端压缩层接受从同步层传输来的压缩格式的数据，并完成解码操作，这些解码后的信息用于终端的视听对象的合成以及显示等。该层包括对象描述框架、场景描述流、视听码流以及上载码流等对象。

其中传输层和媒体无关和传输有关，同步层是媒体和传输都无关，压缩层是媒体有关而传输无关的。

2.1.4 MPEG-4 标准的技术细节

1、 传输多媒体集成框架 DMIF

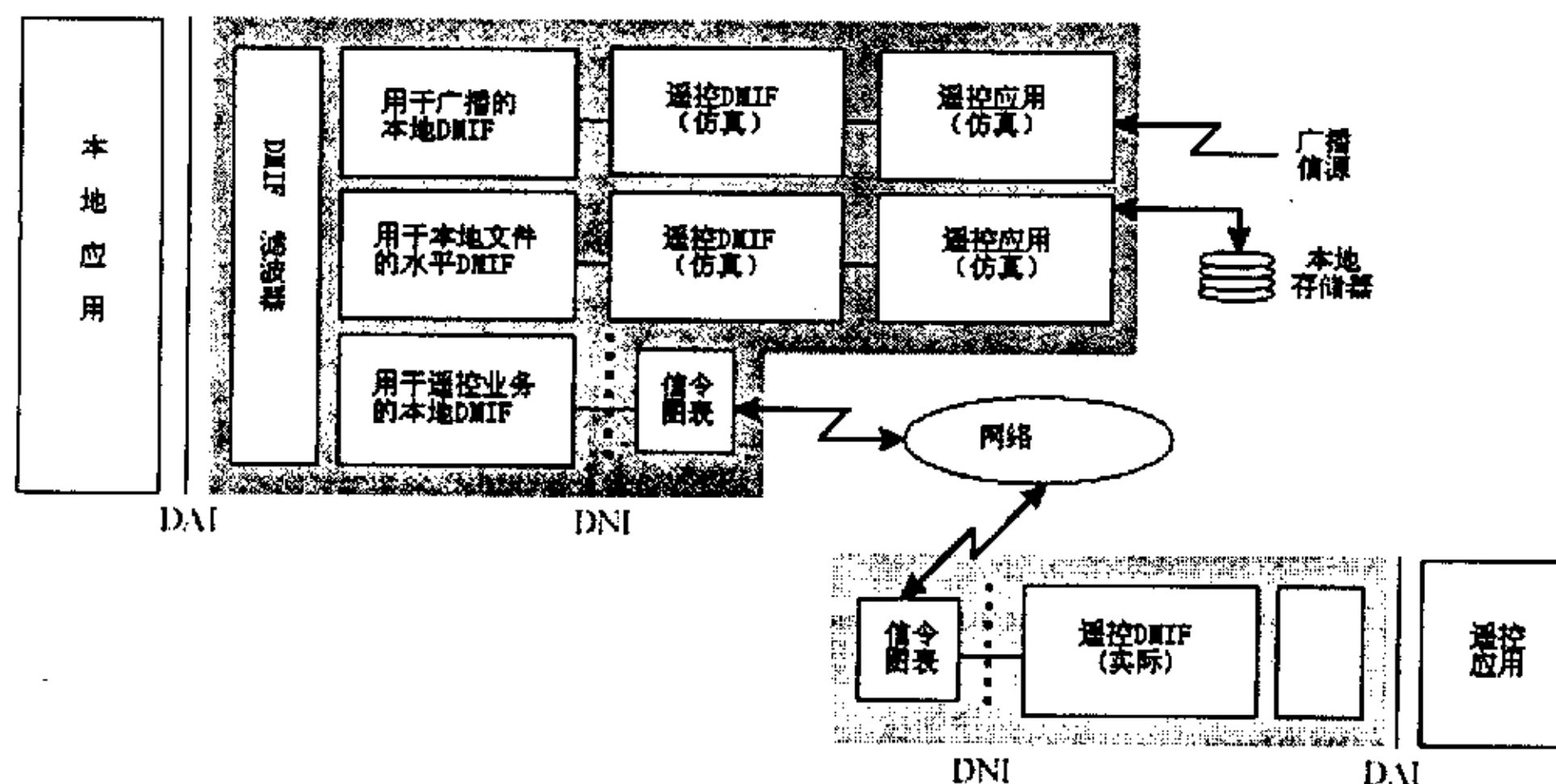


图 2.4 MPEG-4 的 DMIF 通讯结构

传输多媒体集成框架 DMIF（Delivery Multimedia Integration Framework）是在通用传输技术上的管理多媒体流的会话协议。原理上与 FTP 相似，唯一也是基本的差别是 FTP 返回数据，DMIF 返回获取（流）数据的指针。类似地，当 DMIF

运行时, 第一个动作是和远端建立会话。然后, 选择流并发要求 (request) 流注, DMIF 对应端将返回连接流注点的指针, 并建立连接。

MPEG-4 终端 (接收侧) 的主要部分与 FTP 相比, DMIF 既是框架又是协议。DMIF 提供的功能是由 DMIF 应用接口 (DAI) 来表达, 并翻译为协议消息。这些协议消息可能基于运行的网络而不同。服务质量同样为 DMIF 设计所考虑, DAI 允许 DMIF 用户为所需的流指定要求。这样就要求 DMIF 执行时保证要求得以实现。DMIF 规格提供了在几个新网络类型, 例如 Internet 上实现该任务的线索。

2、数据平面

MPEG-4 中的数据平面可以划分为两部分: 与传输有关和与媒体有关两部分。

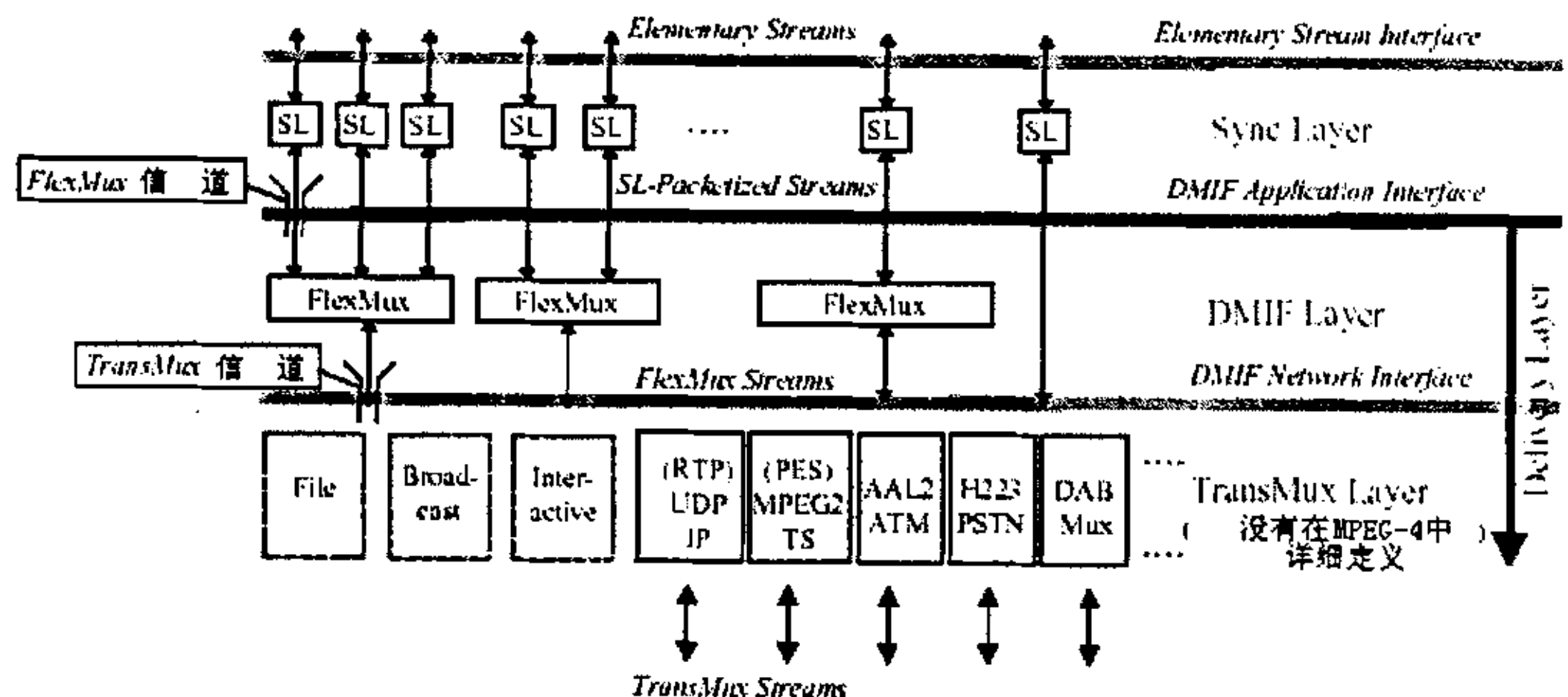


图 2.5 MPEG-4 中的数据平面

图 2.5 中的下部传送复用 TransMux (Transport Multiplexing) 从信息实际传送的方式 (并不由 MPEG-4 叙述) 这个意义来说, 已经超出 MPEG-4 标准的管辖范围。仅仅是这个分层的接口是由 MPEG-4 所规定。在适当的链接层中, 任何适当的现存的传送协议堆栈, 例如, RTP/UDP/IP, MPEG-2TS, AAL2/ATM, H.223 或 DABmux 等都可以采用。DMIF 用于协商所请求的网络服务质量 QoS (Quality of Service)。属于同一应用中不同 MPEG-4 对象, 可以通过不同的发送机制到达终端, 音频和视频可以采用具有各种实时特性的一种传送, 而静止图象采用另一种传送。这留给终端用户/业务提供者进行选择, 并允许 MPEG-4 在较宽的运行环境范围中使用。

灵活复用 FlexMux (Flex MultiPlexing) 提供一种复用过程的工具, 允许基本数据流 ES (Elementary Stream) 以较小的额外开销进行汇合。例如, 它可以用于以低比特率把具有相同的 QoS 的需求或相同的数据流进行汇合 (有可能不是很有效地利用传输资源)。是否使用 FlexMux 复用工具可以选择, 如果下部的 TransMux 提供相同的功能时, 这个分层可以旁路掉。

接入单元分层 AUL (Access Unit Layer) 总是出现的, 它可以识别基本数据流中的接入单元 (即视频和音频的数据帧、各种情景描述命令), 修复 AV 对象或情景描

述的时间基准,以实现它们之间的同步。接入单元的信头能够以极大量的方式来设置,以便在各种系统的广泛频谱中使用。

为了在一个情景中把基本数据流与音频/可视的对象(AVO)相联系,使用各种对象描述符和各种数据流映射表。对象描述符传递与各种特定 AVO 伴随的基本数据流的数目和特性的信息。数据流映射表把每个数据流链接到一个信道联系标记,用于管理携带此数据流的信道。把各种信道联系标记分解到实际的传送信道,以及各种对话和各种信道的管理,都是由 MPEG-4 标准的 DMIF 部分来提出的。

3、缓存器的管理和定时信息的识别

MPEG-4 定义一个系统解码器模型,描述一个理想解码设备的性能,还有数据流的语法和语义。这就给出了终端运行的准确定义,

不需要对设施的各种细节作不必要的各种假定,目的是使实施者可以自由地以各种方式设计各种 MPEG-4 终端和解码设备。缓存器和各种定时信息模型的技术规范对编码设备极为重要,因为事先可能不知道终端设备是什么,或者它们怎样来接收该编码的数据流。

当解码器对构成一次 MPEG-4 对话的各类基本数据流进行解码时,预报解码器将如何工作,系统解码器模型使得编码器可以设定和监视对一次对话所需的最小缓存器资源。所需的缓存器资源是当建立 MPEG-4 对话时,将在各种对象描述符内传递到解码器,因而解码器可以决定是否能够处理这次对话。

通过管理有限数量的缓存器的空间,该模型允许一个发送器事先转移非实时的数据,只要在接收机端具有足够的空间来存储它们。然后这些预先存储的数据在需要时就可以接入,使得实时信息可以使用更大量的信道容量(图 2.6)。

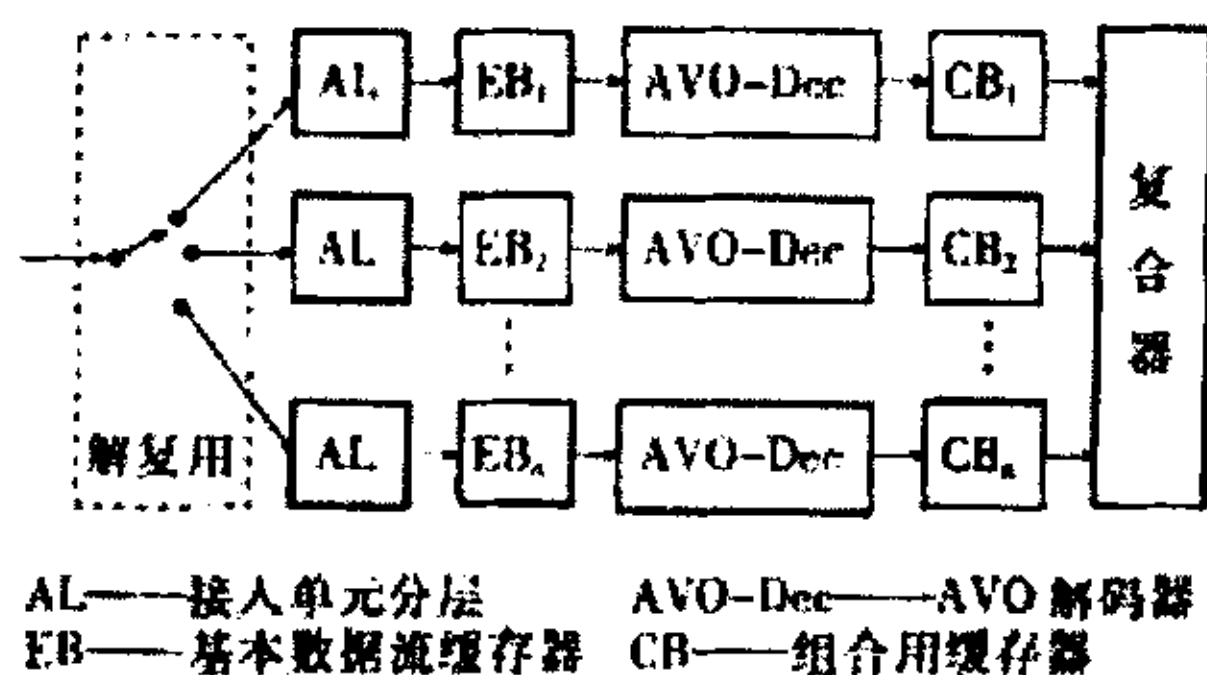


图 2.6 系统解码器的缓存器体系结构

为实时运行,设定了一个终端到终端的定时模型,其中从编码器输出的信号到解码器输入的信号之间的时延为常数。而且,传输的数据流必须包含隐式或显式的定时信息。共有两类定时信息:第一类用于把编码器时钟或时间基准的速率传递到解码器。而由依附于各部分的编码 AVO 数据之时间标记组成第二类,其中包含用于各个“接入单元”所需的解码时间,或者用于各个组合单元的组合时间和截止时

间。这些信息是在接入单元分层中生成的各种协议数据单元(AL-PDU)信头中传递。依靠这些定时信息,图象帧之间的间隔和音频取样率为了同步操作的目的,可以在解码器中进行调节,以匹配编码器的图象帧之间的间隔和音频取样率。

4、音频编码

MPEG-4 音频对象编码为表达自然声音(例如话音和音乐)和基于机构化描述的合成声音都提供工具。合成声音的表达可从文本数据或所谓的结构描述,以编码参数提供反射、空间定向等效果的方式实现。该方式提供压缩和其他例如可伸缩性和效果处理等其他功能。

5、视频信息的编码

MPEG-4 标准的视频信息部分^[8]提供一个包含各种工具和算法的工具箱,对于下述各项提供解决方案:

- (1) 各种图片和视频的高效率压缩;
- (2) 在 2D 和 3D 网格上进行纹理映射的各种纹理的高效率压缩;
- (3) 各种隐式 2D 网格的高效率压缩;
- (4) 各种网格动画的时变几何学的各种数据流的高效率压缩;
- (5) 到所有类型的可视对象之高效随机接入;
- (6) 扩充各种图片和视频序列的操纵功能;
- (7) 各种图片和视频的信息内容基编码;
- (8) 各种纹理、图片和视频的信息之内容基的可分级性(Scalability);
- (9) 空间域的、时间域的和质量的可分级性;
- (10) 在易于产生误码的各种环境中之误码鲁棒性和错误修复能力。

合成节目源的可视信息之编码包括:

- (1) 人脸及相应的各种动画数据流之参数描述;
- (2) 对纹理映射的静态的和动态的网格编码;
- (3) 依赖观看的纹理编码的各种应用。

自然图片和视频的编码是由很多编码工具来完成,由此给出任意形状的各种可视对象的高效率表达式。它也支持 MPEG-1 和 MPEG-2 提供的多数功能,其中包括按照不同等级的输入图象格式、帧频、像素灰度值和比特率,以及按照不同等级的空间域、时间域和质量的可分级性,高效压缩各种标准矩形尺寸的图象序列的有关规定。

MPEG-4 视频方面标准目前提供了应用于自然图片和视频各种比特率和功能的基本分类,其目的是把各种比特率的等级对比和各种功能的集合聚集到一起。

对各种可视的纹理和静止图片的高效编码,采用基于零树的小波算法,这种算法可以在非常宽的比特率范围内提高编码效率。在高效编码压缩的同时,还提供空

间域和质量的可分级性(高达 11 个等级的空间域可分级性及连续的质量可分级性),还有对任意形状的对象编码。

MPEG-4 支持各种图片和视频对象的编码,并具有空间域和时间域的可分级性。既可以对常规的矩形编码,也可以对任意的形状编码。可分级性是指对某个数据流中的一部分矩形解码,而重建各种图片或各种图象序列的能力,并且:

- (1) 降低解码器的复杂性,因而降低质量;
- (2) 降低空间域分辨率;
- (3) 降低时间域分辨率;
- (4) 具有相同的时间域和空间域分辨率,但降低质量。

MPEG-4 提供传输误码的鲁棒性和错误修复能力,以便在很宽范围存储和传输媒体内接入图片或视频信息。特别是由于各种移动通信的迅速增长,能够通过各种无线网络接入音频和视频信息变的极为重要。这就是在各种误码严重的环境中能够以低比特率(例如小于 64Kbps)有效运行各种音频和视频算法的需求。

6、情景描述

MPEG-4 提供把一组对象组合为一个情景的各种工具,而必须的组合信息构成情景描述,即二进制情景描述(BIFS)。BIFS 以二进制的形式表示,而且已被编码,并与各 AV 对象一起传输。

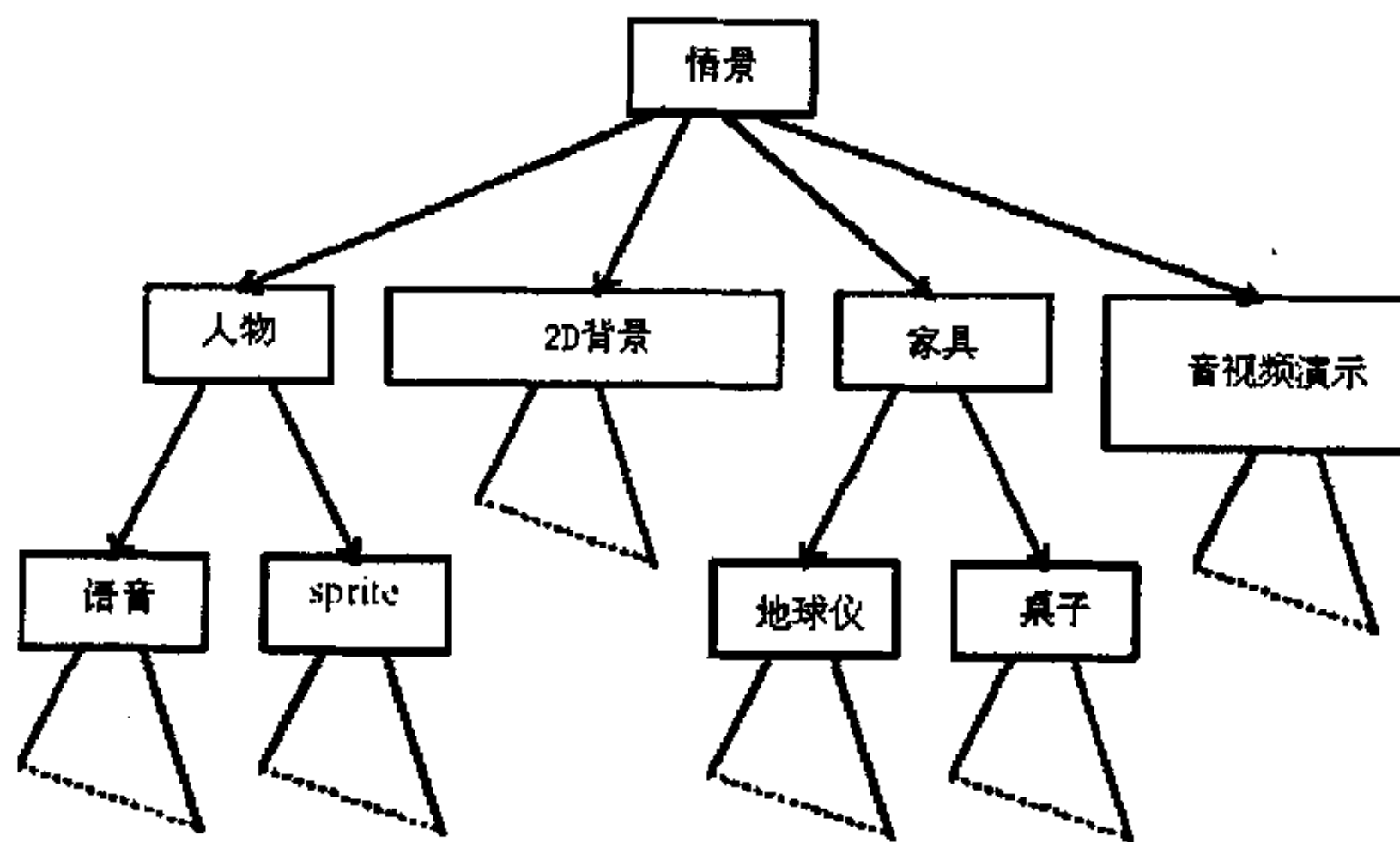


图 2.7 情景的逻辑结构

为了方便各种创作过程、控制和交互工具的开发,各种情景描述都是独立于有关原始 AV 对象而进行编码。对情景描述的各种参数的识别,采取了特殊措施。这是通过下述方式来完成:把改善一个对象的编码效率所采用的各种参数(例如各种视频编码算法中的运动矢量)与用作一个对象的各种修饰语的各种参数(例如该对象在情景中的位置)区别开来。(图 2.7 给出了在一个情景描述中描述的信息的一些例子)

2.1.5 MPEG-4 的应用领域

MPEG-4 在多媒体传输、多媒体存储等领域具有广泛应用:

(1) 低比特率下的多媒体通信, 如视频电话、视频电子邮件、移动多媒体通信、电子新闻等。这些应用对传输速率要求较低, 在 4.8~64kbit/s 之间, 分辨率为 176×144 。MPEG-4 利用很窄的带宽, 通过帧重建技术压缩和传输数据, 以最少的数据量获得最佳的图像质量。

(2) 互联网上的视频流与可视游戏, 如网上电影。

(3) 数字电视、动态图像、万维网 (WWW)。

(4) 实时多媒体监控。

(5) 基于内容存储和检索的多媒体系统, 如视频点播 (VOD) 等。

(6) 基于面部表情模拟的虚拟会议。

(7) DVD 上的交互多媒体应用。

2.2 MPEG-4 视频校验模型

MPEG-4 视频组开发了一个随着时间的推移不断改进的校验模型^[15] (Verification Model, VM)。校验模型是一个带有一个对编码和解码算法精确定义的通用平台, 这些算法能够作为表达实现特殊功能的工具。通过一些成功的核心试验, 新的算法和工具加入到 VM 中, 旧的算法和工具被覆盖。

2.2.1 视频对象平面

视频对象 (VO) 对应于位流中用户可以访问和操作 (剪贴、粘贴...) 的实体。给定时间的视频对象实例称为视频对象平面 (VOP)。编码器传送 VOP 和同步信息 (使用同步层语法), 指出每个 VOP 在什么地方和什么时间被显示。在解码端, 允许用户可以通过同步信息的交互来改变已显示的场景的同步。

VOP 可以是一个场景中的语义对象: 它由 Y、U、V 和形状信息组合而成。在 MPEG-4 视频测试序列中, 有些 VOP 是半自动分割定义的, 有些是已知序列构造的。在第一种情况下, 形状信息由 8 比特表示。第二种情况, 形状是一个二进制掩码。这两种情况现在编码过程都会考虑。

VOP 可以是任意形状的。当视频序列只有一个以固定间隔和固定尺寸显示的矩形 VOP 时, 它符合基于帧的编码技术。

2.2.2 编码器结构

1、综述

MPEG-4 视频编码器结构如图 2.8 所示，主要由两部分组成：传统的运动与纹理编码和形状编码（应用于相同的 VOP）。对每个 VO 的运动、纹理和形状信息分别独立编码，然后将各个 VOP 的码流复合（MUX）成一个位流。其中，在编码控制和复合阶段可以加入用户的交互控制或由智能化的算法进行控制。

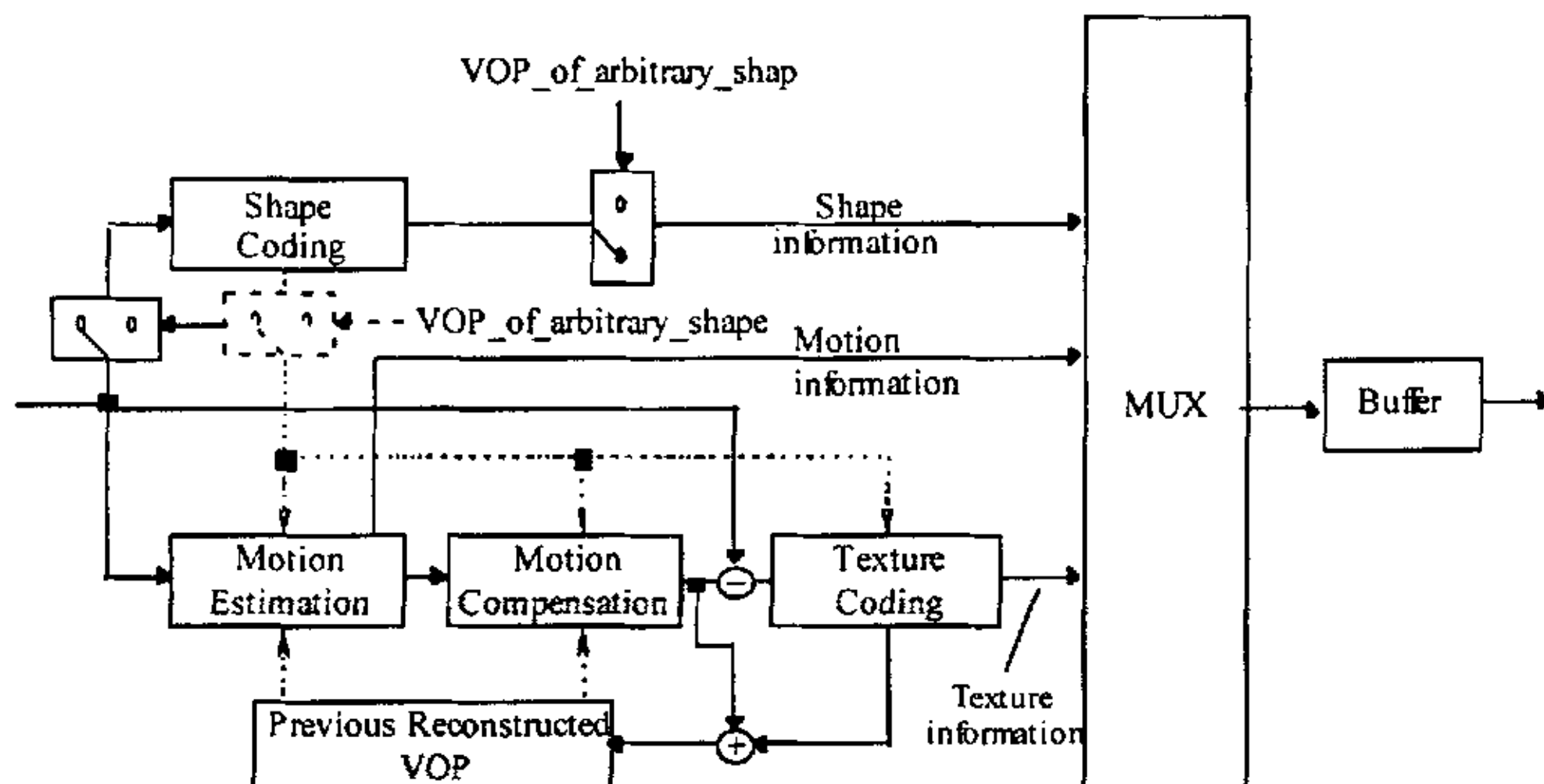


图 2.8 VOP 编码器结构

2、形状编码（Shape Coding）

形状编码分为二值和灰度形状信息的编码两种方式。形状信息后面指 alpha 平面。二值 alpha 平面采用 CAE 算法，灰度 alpha 平面编码采用类似于纹理编码的运动补偿的 DCT 算法。alpha 平面的边界为一个包围 VOP 形状的矩形。VOP 的边界矩形在右下角扩展为 16×16 的整数倍的块，扩展的 alpha 采样设置为 0。扩展的 alpha 平面被分割为 16×16 的采样块（今后指 alpha 块）并且编/解码处理在 alpha 块之前。

如果在一个宏块内的象素都为透明（全 0），这个块的运动和纹理编码跳不进行。由于这个透明信息可以从形状编码获得，所以不必要进行什么头信息来指出这种编码模式。这种跳过编码模式应用于所有 I、P 和 B-VOP。

3、运动估计（Motion estimation, ME）和运动补偿（Motion compensation, MC）

类似于以前的压缩标准（MPEG-1、H.263 等）的三种帧格式：I-帧、B-帧、P-帧，MPEG-4VM 中的 VOP 也有三种相应的帧格式：I-VOP、B-VOP、P-VOP，表示运动补偿类型的不同。

为了对 VOP 进行运动估计/补偿,需要在运动估计/补偿之前对参考 VOP 进行基于宏块的重复填充(Repetitive Padding)。对于 VOP 内不同的宏块采用不同的技术进行运动估计/补偿,对标准宏块采用块匹配技术进行运动估计;对边缘宏块采用多边形匹配技术进行运动估计;对外部宏块不进行运动估计。VOP 运动估计支持基本的运动估计/补偿模式(基于宏块)、无限制的运动向量模式和高级预测模式。无限制的运动向量模式允许运动向量指向 VOP 之外;高级预测模式允许每个宏块有 4 个运动向量。

4、纹理编码(Texture Coding)

帧内 VOP 和运动补偿的残余数据编码都使用相同的 8×8 DCT 块编码方案。每个亮度和色度平面各自进行 DCT 编码。对于任意形状的 VOP 编码,属于 VOP 的宏块编码描述如下。每个任意形状的 VOP 有两种宏块类型: VOP 内部块和 VOP 边界块。VOP 内部块编码技术和 H.263 相同。对属于 VOP 边界宏块的帧内 8×8 块先进行填充,对不属于 VOP 的块用 0 填充。对每个 8×8 的亮度和色度块的填充各自进行,填充使用这个 8×8 块内的原始的亮度和色度 alpha 值。由于透明块不编码,所以不填充。经填充后的块再作 DCT、量化、DC&AC 系数预测和 VLC 编码。

5、B-VOP 的预测和编码

B-VOP 的宏块既可以用 H.263 中的 B 块编码方式也可以用 MPEG-1 中的 B 帧编码方式,主要区别在于运动向量的数目和量化方法上。MPEG-4VM 中 B-VOP 支持像 H.263 中的 B 块宏块类型编码提到预测方向,前向预测、后向预测和 MPEG-1 中的 B 帧编码方式支持的双向预测。

6、容错机制

为了容许对一个很宽的范围内存储和传输的媒体进行图像和视频信息的访问, MPEG-4 提供了错误鲁棒性和错误修复技术。为 MPEG-4 开发的错误修复技术可以分为三个主要区域。这些区域或者种类包括重新同步、数据修复、错误消除和错误修复。这些种类并不是 MPEG-4 独有的,但许多研究人员在视频错误修复区域做了很多工作来替代以前的算法。同时,这些种类中包含的工具非常有用,它们对处理错误修复的问题有所贡献。

7、可扩展编码(Generalized Scalable Encoding)

可扩展包括空域分级扩展和时域分级扩展,这是 MPEG-4 的一个重要功能。

分级定义:当位流中至少存在一个子集能够产生一种有用的表示,则认为这个位流可分级。

视频分级编码就是指可以产生一种编码表示,它可以支持多于一种的分辨率/质量。

每种类型的分级一般至少都有两层：一个基本层和一个增强层。一般来讲，空域分级就是用增强层来增加基本层的空域分辨率；时域分级就是用增强层来增加基本层中感兴趣区域的时域分辨率—帧率。

8、Sprite 编码

Sprites 编码技术是针对背景视频对象的特点提出的一种有效的编码方法。在许多应用场合中，背景视频对象自身是没有任何局部运动的，其每帧图像所产生的变化是由于前景物体的运动，一部分背景被掩盖，而另一部分背景又显露出来；或者是由于摄像头的运动。

一个 sprite 就是一个图像，组成这个图像的所有象素属于在一个视频片断中一直可见的视频对象。例如，由一个面板序列生成的 Sprites 将包含整个序列的背景对象的所有可视象素。可能这个背景的某一部分由于前景对象的遮挡或者相机的运动而不可见。既然 sprite 包含所有最少有一次可见的背景，Sprites 就可用来进行背景 VOP 的直接重建或者背景 VOP 的预测编码。背景的 Sprites 用文学语言说就是指“背景镶嵌”。

2.2.3 解码器结构

综述

图 2.9 是一个 VOP 解码器的结构示意图。在解码一个给定片断的所有 VOP 时应用相同的解码方案。

解码器主要由两部分组成：形状解码和传统的运动、纹理解码。重建 VOP 包含形状、纹理和运动信息的组合。对于核心试验，要对解码输出进行一个特殊的块滤波^[15]。

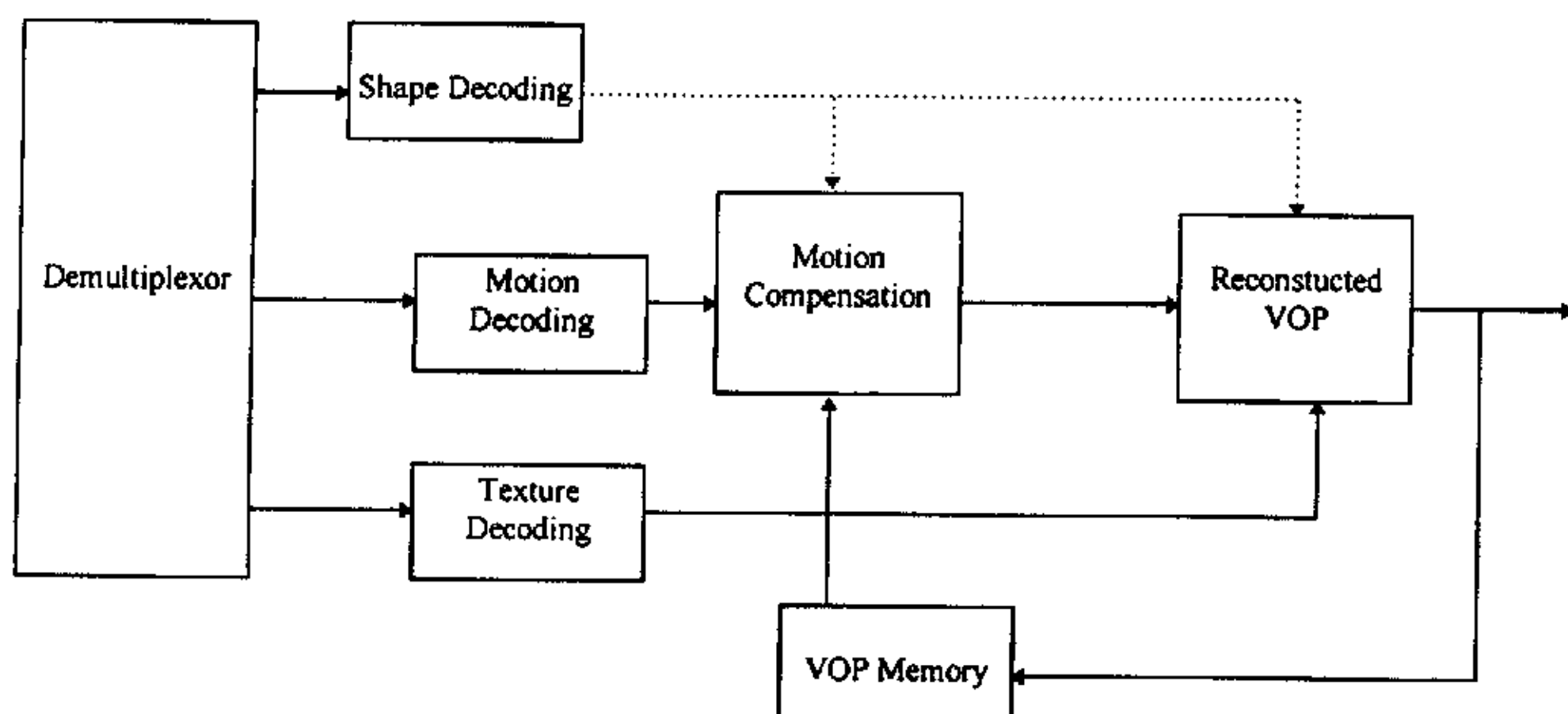


图 2.9 VOP 解码器结构

2、形状解码 (Shape decoding)

二值形状解码是基于块的表示的。主要的编码方式就是基于块的基于上下文的二值算术解码和基于块的运动补偿。主要用到的数据结构为二值 alpha 块

(BAB)表示。BAB 是一个二值表示的像素正方块，这些二元值表示了一个 16×16 像素的块形状空间区域的像素的不透明或者透明。实际上，每个 BAB 和每个纹理宏块处于相同的位置。

3、可扩展性解码 (GS Decoding)

考虑两层时的情况，一个基本层和一个增强层，每层可能有相同的分辨率或者不同的分辨率。当基本层和增强层具有不同的分辨率时，增强层为了使用基本层作为预测，需要对基本层数据进行向上或者向下的采样。如果低层和增强层有时间上的偏移，在不考虑空域分辨率时，可以在两层之间使用运动补偿预测。当两层在时间上一致但分辨率不同，可以不进行运动补偿。

第三章 形状编码的研究和算法改进

形状编码是 MPEG-4 标准的主要内容之一,也是其与以前视频编码标准的最大区别之一,也是本论文研究的核心内容。本章首先对 MPEG-4 形状编码和 VM 中的基于上下文的算术编码(CAE)进行了深入的分析 and 论述。然后详细阐述了本论文提出的一种形状编码快速运动估计算法和一种改进的形状编码尺寸转化算法。

3.1 MPEG-4 形状编码

为了提供基于内容的交互性,高效的压缩和通用访问性,MPEG-4 视频部分采用的最关键技术是:基于对象的视频数据表示方案。视频场景被看成一些视频对象的组合。视频对象具有形状,运动和纹理等属性。其中形状编码是 MPEG-4 与传统视频编码技术的最大区别。

MPEG-4 中的每个 VOP 都包含形状、纹理和运动信息。形状的表达主要有二值和灰度两种。二值图(每像素 8 比特)是指物体内部点值是 255,外部点值是 0;灰度图中的点值可以从 0 到 255 连续变化。MPEG-4VM 中有两种形状编码方法,过程如下图(图 3.1)。

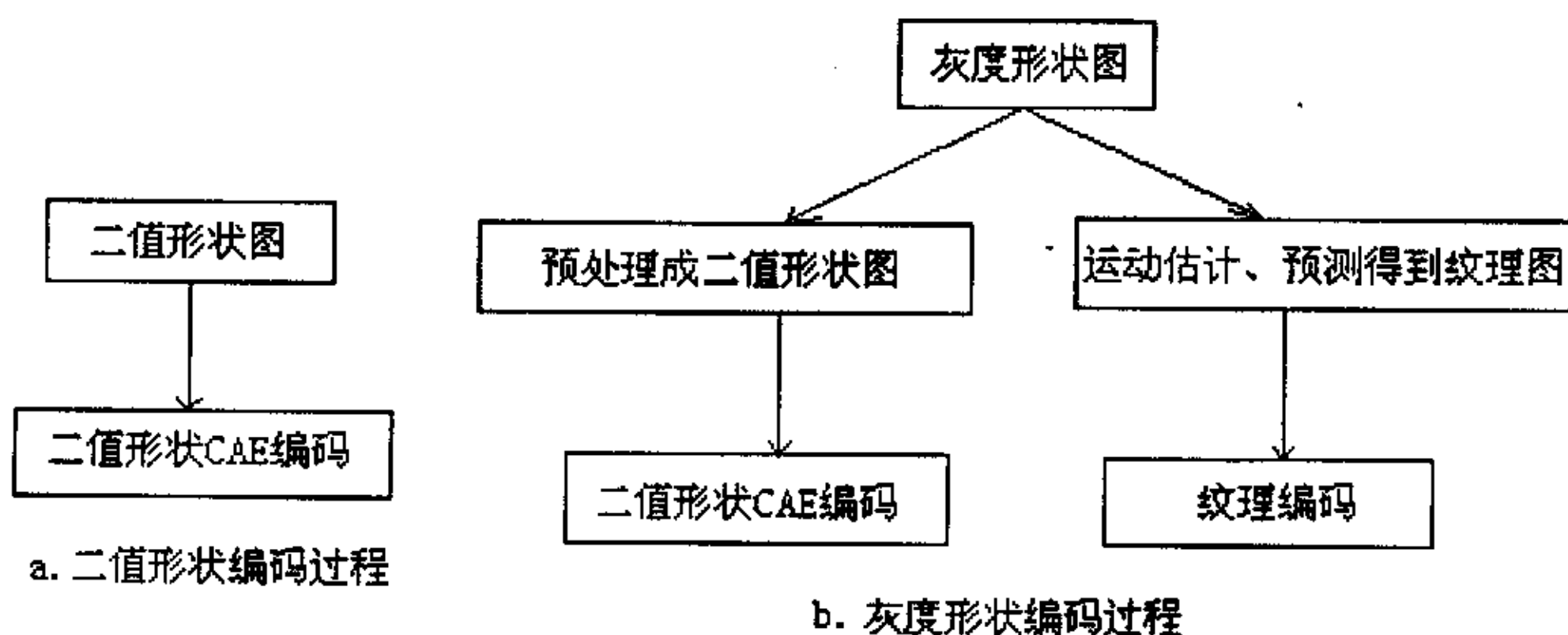


图 3.1 VM 中的形状编码方法

使用灰度形状图编码的码流,解码后的重构的 VOP 形状与背景合成后的图像清晰自然。而使用二值形状图编码的码流,解码后的直接合成图像物体边界过于突出,不够自然。因此,对二值形状图编码的码流,在解码端对于二值形状解码 CAD(Content-based Arithmetic Decoding)后还要进行羽化(Feathering)以产生与原来相似的灰度图。

不论二值形状图编码还是灰度形状图编码,都要进行二值形状编码的过程。而灰度形状图编码的其余部分和纹理编码很相似,所以研究形状编码主要就是研究二值形状编码。

3.1.1 二值形状编码

1. 目前存在的形状编码方法

二值形状编码器主要分为三类:基于位图的、基于轮廓的和隐含的色键值编码器。下面就目前的几类主要的形状编码算法进行介绍。

1) 基于位图的形状编码:

基于位图的形状编码器对宏块基础上的形状信息进行编码(判断一个像素不属于这个对象)。包括基于上下文的算术编码器(CAE),它编码形状信息和运动补偿信息(目前的MPEG-4标准校验模型^[15]中就采用的是这种编码);和基于改进的MR^[19](Modified Modified Reed)编码器(MR编码器可以替代算术编码器),算法的其它方面是相同的。基于位图的形状编码算法简单,压缩率较高。

2) 基于轮廓的形状编码:

这种编码方式是描述编码对象的轮廓。包括基于顶点的形状编码^[23](对形状的轮廓进行编码。对于有损形状编码,采用多边形近似的方法^[25]。节点是在形状近似误差范围内较容易控制的位置。对于无损形状编码,多边形近似就退化为链码^[24]。);B样条形状编码^[21](采用二次B样条曲线近似给定的边界)等等。基于轮廓的形状编码算法较复杂,压缩效果较好。

3) 色键编码^[26]:

这种编码技术将要编码的对象被放置在单色的背景中,背景的颜色必须超过对象纹理所占用的颜色空间,通常用高饱和度的颜色。然后采用全帧模式进行编码。对于解码器,色键值(背景色)被传送。解码器解码图像的过程中,与色键值相似的颜色被认为是背景色,其余的像素属于对象。这种方法的一个重要的优点就是较低的计算和算法复杂度。

2. VM中的二值形状编码

MPEG-4VM中采用基于上下文的算数编码(CAE)编码,一方面是为了与传统的基于宏块的编码(MPEG-1/MPEG-2中)相兼容,另一方面是因为它的算法实现简单,压缩率较高。

二值形状编码的步骤如下:

1) 对于给定的VOP的二值形状图重新确定形状边界。

二值形状编码的基本单位是 16×16 的形状块 BAB, 如果 VOP 的边界矩形不能划分成整数个形状块, 则扩展 alpha 平面的右下角坐标, 使其大小变成 16×16 的整数倍, 扩展出来的形状点取值为零。

2) 确定编码模式 (一)

形状编码的编码模式共有 7 种, 分别是:

(0) $MVDs == 0 \& \& NoUpdate$ 。其中 MVDs 表示形状信息运动, 矢量差值 NoUpdate 表示不进行帧内/帧间算术编码。

(1) $MVDs != 0 \& \& NoUpdate$ 。

(2) all_0。all_0 表示该块为完全透明块。

(3) all_255。all_255 表示该块为完全不透明块。

(4) intraCAE。intraCAE 表示帧内算术编码。

(5) $MVDs == 0 \& \& interCAE$ 。interCAE 表示帧间算术编码。

(6) $MVDs != 0 \& \& interCAE$ 。

对 I-VOP, 只采用 all_0、all_255 和 interCAE 模式。对于 B/P-VOP, 以上的七种编码方式都有可能。编码质量是否可以接受由 ACQ(Accepted Quality)函数定义。ACQ 函数比较近似 BAB 块与原始 BAB 块之间所有 16 个 4×4 子(Pixel Block, PB)块的 SAD 误差。

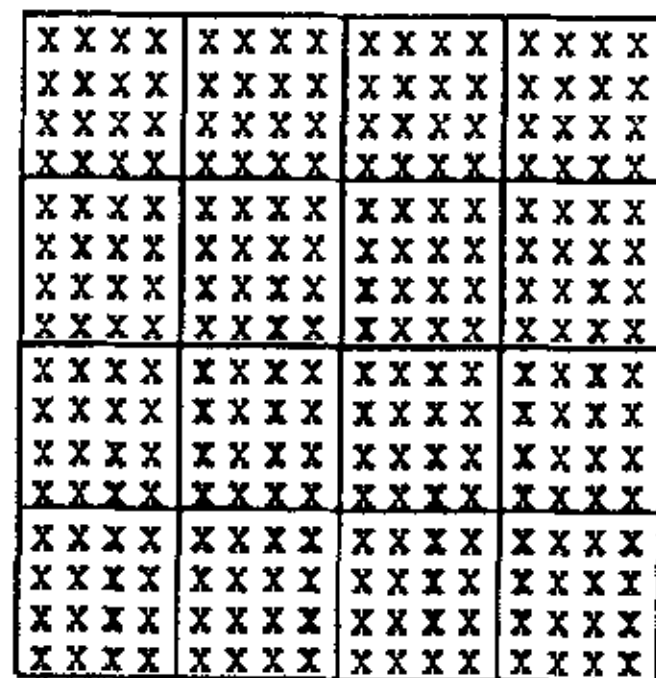


图 3.2 一个 BAB 块由 16 个 PB 块组成

$$ACQ(BAB) = \min(acq_1, acq_2, \dots, acq_{16})$$

如果 $SAD_PB_i > \alpha_{th}$, 那么 $acq_i = 0$, 否则 $acq_i = 1$ 。如果所有 PB 块误差都小于预定误差阈值 (α_{th}) 就认为可以使用该近似形状块代替原始形状块, 反之则不能。

编码模式的确定分两步进行, 这是第一步的算法:

```

if(ALL0(BAB)) ShapeMode=ALL_0;
else if(ALL255(BAB)) ShapeMode=ALL_255;
else if(ACQ(BAB255)&& ACQ(BAB0))
{

```

```

if(OPAQUE_PIXELS>=128) ShapeMode=ALL_255;
else ShapeMode=ALL_0;
}
else if(ACQ(BAB255)) ShapeMode=ALL_255;
else if(VopType==I-VOP)
{
ShapeMode=intra_CAE; // 如果是 B/P-VOP, ShapeMode=intra_CAE
DecideCR();//确定分辨率, 如果是 B/P-VOP, 这一步在后面进行
}

```

注意: 上面程序中 ALL0(BAB)和 ALL255(BAB)是指这个 BAB 块内像素全为 0 或者全为 255; ACQ(BAB0)和 ACQ(BAB255)是指这个 BAB 块如果被编码为全 0 或者全 255, 解码质量可以接受。

在这一步, 先大概确定该 BAB 块用 all_0、all_255 和还是 intra_CAE(interCAE) 模式, 后面再根据运动向量进行进一步的确定。

3) 运动估计和补偿

确定 BAB 块后, 如果该 VOP 是 B-VOP 或 P-VOP, 那么就对待编码的 BAB 进行运动估计, 得到运动矢量 MVs (MV for shape); 如果该 VOP 是 I-VOP, 则该步可以省去。(在论文 3.3 部分将对运动估计进行详细研究)

4) 确定编码模式 (二)

进行运动估计后, 如果该 VOP 是 B-VOP 或 P-VOP, 那么就对待编码的 BAB 进行进一步的编码模式确定; 如果是 I-VOP, 则不需要进行这一步。

```

if(ShapeMode!=ALL_0)
{
MotionEstimation();//运动估计
if(ALL0(BAB)|| !ACQ(MC_BAB)/*判断 16 个子块的 SAD 是否全小于误差阈值, MC_BAB 指运动补偿块*/)
{
//如果全透明或不接受 帧间 CAE
DecideCR();//确定分辨率
if(MVs.isZero())//运动向量为零
ShapeMode = INTER_CAE_MVZ;
else
ShapeMode = INTER_CAE_MVNZ;
}
else if(ShapeMode==ALL_255&& !MVs.isZero())

```

```
//如果全不透明 且 运动向量不为零 全不透明编码
```

```
ShapeMode == ALL_255;
```

```
else if(ALL255(BAB)&&!ALL255(MC_BAB))
```

```
ShapeMode = ALL_255;
```

```
else if(MVs.isZero())//运动向量为 0
```

```
ShapeMode = MVZ_NOUPDT;
```

```
else
```

```
ShapeMode = MVNZ_NOUPDT;
```

```
}
```

5) 尺寸转化 (确定待编码 BAB 块的分辨率)

由于有码率控制和空域可扩展性, 所以有时分辨率的改变是必需的。尺寸转化由两步组成。转化的比例由 VOP_CR 确定, VOP_CR 可以取 1/2 或者 1/4。当 VOP_CR 为 1/2 时, 整个 VOP 就通过下采样, 得到原来 1/4 大小的形状图。下采样可以通过平均值来代替多个采样点。上采样通过插值得到。

VOP 的形状编码是基于 BAB 块, 而 BAB 块的分辨率可根据不同块特点而改变。

VM 中确定 BAB 块的分辨率 CR 的算法如下:

```
CR=1/4;
```

```
DownSampleShape();
```

```
UpSampleShape();
```

```
if(ACQ(BAB)) return;
```

```
CR=1/2;
```

```
DownSampleShape();
```

```
UpSampleShape();
```

```
if(ACQ(BAB)) return;
```

```
CR=1;
```

先将当前 BAB 块按 CR 所代表比例值进行下采样, 再进行相应上采样得到与原来同样尺寸的重构 BAB 块, 再对这两个 BAB 块相应 PB (4×4) 块计算 SAD (绝对误差和) 值, 如果 SAD_Pbi>ALPHA_TH, 那么 ACQ(BAB) 返回为 false, 若所有 16 个 PB (4×4) 块 SAD_Pbi≤ALPHA_TH 则 ACQ(BAB) 返回为 true。

6) 基于上下文的算术编码

除过 ALL_0 和 ALL_255 两种编码模式, 其余编码模式的 BAB 块都要进行基于上下文的算术编码(CAE), 分帧内和帧间两种模式。对帧内编码模式, 分别尝试水平扫描和垂直扫描两种方式进行编码, 最终采用编码字节少的一种, 并输出到输出码流中。对帧间编码模式, 先尝试帧内模式下的水平和垂直扫描方式, 再尝

试帧间模式下的水平和垂直扫描方式编码，最终采用其中编码字节最少的一种，并输出到输出码流中。

CAE 编码过程将在下节详细论述。

3.1.2 灰度形状编码

1、轮廓编码和灰度 alpha 值编码

灰度的 alpha 平面编码由两部分组成：一个是它的形状轮廓编码，另一个是在轮廓中的 alpha 值的编码。轮廓编码采用二值形状编码，alpha 值编码采用任意形状的纹理编码。（见图 3.1b）

轮廓是通过在灰度级的 alpha 平面上通过设定阈值 0 得到的。除了 DCT 变换是基于帧的^[15]，像亮度值一样，灰度 Alpha 值被分割为 16×16 块进行编码。在码流中一个 alpha 宏块的编码将附加到它对应（纹理）宏块编码的后面。

2、羽化

许多视频序列使用灰度 alpha 掩码，它们的纹理相对简单一些。例如有些是由固定灰度值构成的灰度 alpha 掩码。还有一些灰度 alpha 掩码由一个在轮廓边缘处从 255 递减到 0 的二值 alpha 掩码构成，这样可以和背景形成光滑过渡。后面这种类型的掩码可以用一个二值的掩码和羽化描述（一个简单的改进的二进制 alpha 掩码）组成，羽化就是轮廓边缘光滑过渡到背景。这部分的详细内容见 MPEG-4 校验模型^[15]。

3.2 基于上下文的算术编解码

本节将详细讨论基于上下文的算术编码（CAE）和解码（CAD）过程。CAE 编码属于一种改进的，更高效的算术编码。

3.2.1 算术编码简介

算术编码^[30]在图像数据压缩标准中扮演了重要的角色。在算术编码中，信息用 0 到 1 之间的实数进行编码，算术编码用到两个基本的参数：符号的概率和它的编码间隔。信源符号的概率决定压缩编码的效率，也决定编码过程中信源符号的间隔，而这些间隔包含在 0 到 1 之间。编码过程中的间隔决定了符号压缩后的输出。

算术编码对整条信息（无论信息有多么长），其输出仅仅是一个数，而且是一个介于 0 和 1 之间的二进制小数，因此译码器在接受到表示这个小数的所有位之前不能进行译码。

算术编码和哈夫曼编码一样，也是对在信息中出现较多（概率大）的符号采用较少的编码位数，对在信息中出现较少（概率小）的符号采用相对较多的编码位数，从而达到压缩的目的。但由于算术压缩可以将一个符号出现的概率表示为小数个二进制位，而哈夫曼编码的符号编码位数都为整数，所以算术编码可以接近无损压缩的熵极限，比哈夫曼编码效率更高。

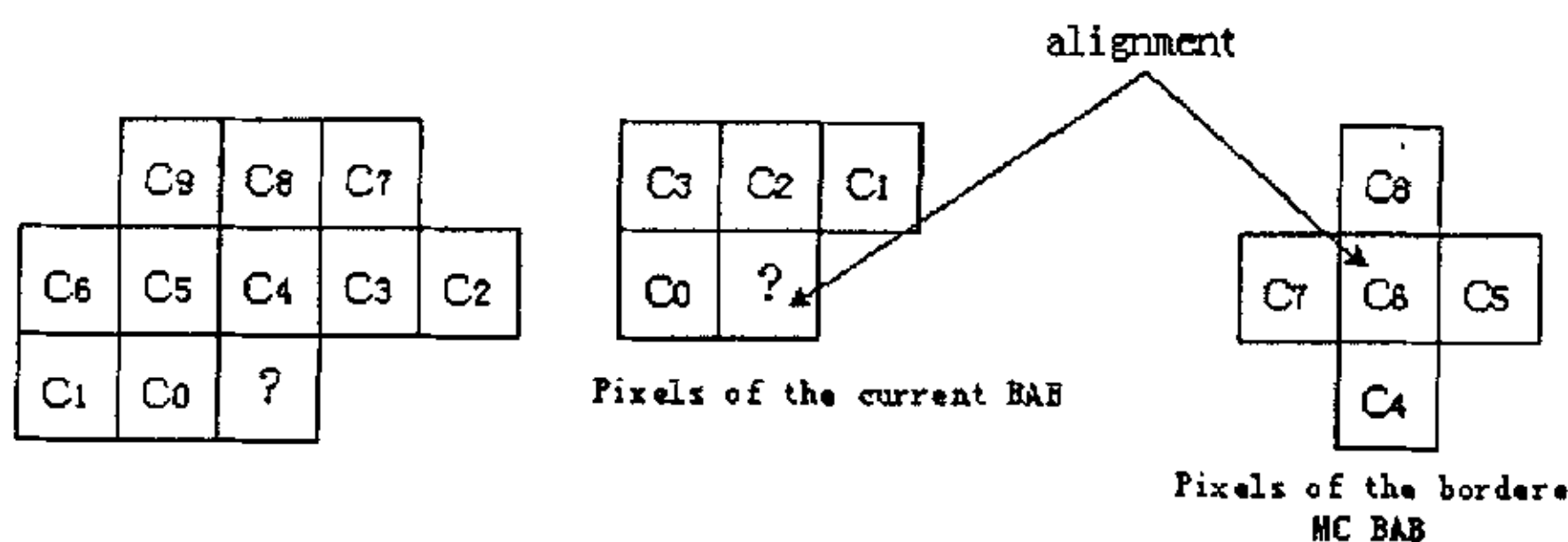
对于较长的信息，所有符号在大量信息中的概率差别很小，如果直接应用这些概率编码，压缩率不大。如果采用在一定限制条件（上下文环境）下的概率，即基于上下文的算术编码，其信息的熵建立在更高的概率层次上，这样得到的信息总熵值更小，压缩率更高。

3.2.2 CAE 编码

CAE 编码主要分为计算上下文值、查概率表和算术编码三部分。

1、计算上下文值（环境数）

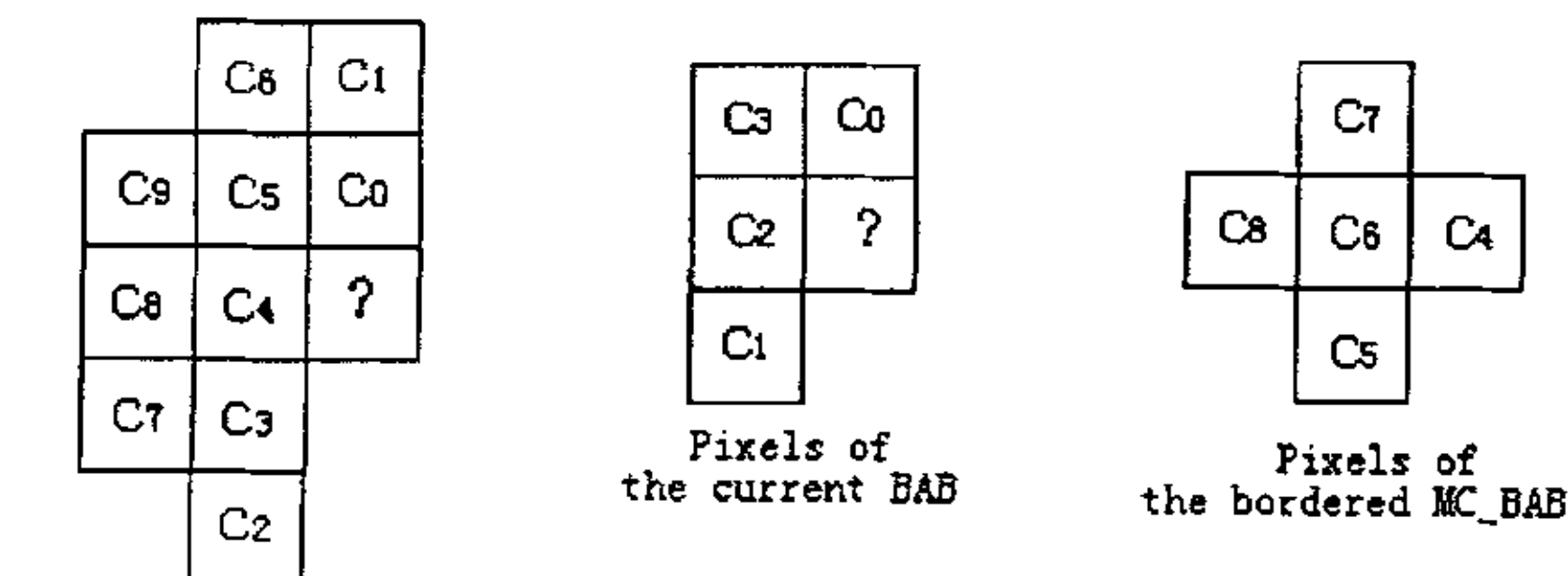
基于上下文的算术编码中符号的概率并不是该符号在整个信息中出现的概率，而是该符号在一定上下文环境下的概率，这时信息的总熵值更小。这里存在一个上下文模板的大小的问题，模板越大，信息的总熵值越小，但需要存储模板的概率表也越大。CAE 算法在两者均衡之下选择帧内模板大小为 10，帧间模板为 9。



(a) IntraCAE 时的相关点

(b) InterCAE 时的相关点

水平扫描方式时



(a) IntraCAE 时的相关点

(b) InterCAE 时的相关点

垂直扫描方式时

图 3.3 计算上下文时的相关点

上下文值就是根据当前的待编码点附近的多个点的像素值生成的一个数字。帧内与帧间形状块的取点位置如图 3.3。从图中可见，帧内块在当前点附近取 10 个点，帧间块在当前点附近取 4 个点，并在运动补偿块中的对应点周围取 5 个点。（注意：当扫描方向为垂直时，形状有所不同）帧内块与帧间块生成上下文值的公式都是：

$$C = \sum_k c_k \cdot 2^k$$

其中 c_k 取值为 0 或 1，表示对应环境点是否透明。 $K=0, 1, 2, \dots, 9$ ，如果第 k 点为 255，则 c_k 为 1，否则 c_k 为 0。

2、查概率表

形状块中每个点生成上下文值后，将以该上下文值为索引，从概率表中取得一个概率值。其中帧内和帧间形状块对应不同的概率表，帧内概率表的大小为 $2^{10}=1024$ ，帧间概率表的大小为 $2^9=512$ 。

VM 中 CAE 编码概率表只存储了模板中所有排列的上下文情况下输入符号为 0 的概率，概率为一个放大 2^{16} 的整数，输入符号为 1 的概率就是用 2^{16} 减去输入符号为 0 的概率的差（概率表见参考文献^[15]之附录 F）。

3、算术编码

算术编码主要涉及 5 个过程：初始化、符号编码、规格化、比特填充、终止处理。

初始化：算术编码在实际应用中并不是直接对小数进行变换，它将符号范围和符号概率都放大到一个很大的二进制整数，这样便于编解码操作。在开始编码前要初始化下次输入符号的编码输出范围。输出范围即编码输出的高低（H 和 L）边界，实际由两个参数确定，低边界 L 和范围 R（因为 $H=L+R$ ）。VM 中算术编码的初始化编码输出范围为 $0 \sim 2^{31}$ ，即 $L=0$ ， $R=2^{31}$ 。

符号编码：每次算术编码根据输入符号为高概率符号还是低概率符号和该符号的概率确定新的编码结果输出范围。CAE 编码的低概率符号（LPS）和高概率符号（MPS）并不是固定的 1 或者 0。在当前上下文下，如果输入符号 1 的概率大于符号 0 的概率，则 1 为高概率符号，0 为低概率符号；如果输入符号 0 的概率大于符号 1 的概率，则 0 为高概率符号，1 为低概率符号。如果输入为 LPS，将新的 L 变为上次编码的高边界 H，如果输入为 MPS，L 不变。新的 R 值由上次 R 值乘以符号概率得到。

规格化：每编码一个输入符号后，编码范围都会减小。为了防止溢出要对输出范围进行判断，如果太小则进行一定的放大，同时移位输出一些已经确定的输出位。在 VM 中，如果编码范围小于 2^{30} 则进行放大直到大于 2^{30} 。如果等待全部算术编码结束再输出编码结果，这个二进制串将非常长。计算机的内存有限，这

这样做显然不合理。当编码范围缩小时, 编码的 L 和 H 将会非常接近 (如 $L=0.6257, H=0.6259$)。这时不管后面编码范围如何缩小, 编码的最终输出结果前面的几位 (0.625) 都不会改变。所以在每进行一次符号编码后, 要判断 L 和 H 的最高有效位 (左边编码位), 如果相同就输出这些位。

比特填充: 在编码过程可能出现一些和系统控制符相同的二进制串, 必须在这些串中加入一些填充位以防系统判断错误。

终止处理: 编码结束时, 必须判断输出一个在编码输出范围内码长最短的二进制串, 同时添加编码结束符号。

3.2.3 CAD 解码

CAD 解码也分为计算上下文值、查概率表和算术解码三部分。算术解码过程和编码过程相似, 处理正好相反, 过程为初始化、符号解码、规格化、终止处理。

符号解码: 每次根据概率表, 判断输入编码结果落在高概率符号范围内还是低概率符号范围内。如果落在该概率符号范围内, 则输出高概率符号, 反之输出低概率符号。并更新符号范围。

规格化: 和编码类似, 对符号输入范围进行必要的放大, 同时根据需要读入新的比特位。在读入比特时, 要判断去除填充比特。

3.3 一种改进的 MPEG-4 形状编码的快速运动估计算法

在 MPEG 编码中, 一般采用三种手段进行图像压缩: 一是利用 DCT 变换来消除图像帧内冗余; 二是利用熵编码来消除符号编码冗余; 三是利用运动估计来消除图像的帧间冗余。对于运动图像而言, 帧间冗余远大于帧内冗余和符号冗余, 因此运动估计显得特别重要。然而同时, 运动估计算法的运算量也是非常大的, 其被认为是 MPEG 发展的障碍之一。一直以来, 多数研究学者都将注意力集中在纹理编码的快速运动估计算法上了。虽然形状编码的运动估计和纹理编码的运动估计相似, 但也存在不同的地方。

利用视频序列的相关属性和形状信息的边界特点, 本文提出了一种关于 MPEG-4 形状编码的快速运动估计 (ME) 算法。同时, 针对 BAB 块的不同运动量, 对 BAB 块进行分类, 不同类采用不同的搜索范围和搜索方法。这些大大的提高了运动估计的处理速度, 降低了搜索算法的搜索点数, 获得了比较满意的结果。

3.3.1 运动估计算法概述

运动估计是序列图像编码提高压缩比的一项关键技术，研究的重点多数是由物体的二维运动来估计其三维运动，运动目标在帧间有平移、旋转及其它变化，而背景的变化要小一些或者是静止的。如果要全面地估计运动物体的各种运动变化，需要含有许多参数的估计。由于实际编码和压缩效率的要求，目前视频国际标准大多仅考虑物体的平移运动，运动模型为：

$$\begin{cases} h = x + v_x \\ v = y + v_y \end{cases} \quad (1)$$

其中：(x,y)为运动物体的初始位置；(h,v)为运动物体的终止位置；(v_x,v_y)为运动矢量。

运动估计的方法有块匹配法和像素递归法。块匹配法是将当前图像帧分成大小相等的矩形子块，假定这些块只做平移运动，对于每一个子块通过搜索窗在前一帧的图像块中搜索具有最大相关性的匹配子块，从而确定其运动矢量。像素递归法是对每个像素的位移进行估计。块匹配法的精度较低，但它的位移跟踪能力强，易实现，得到广泛的应用。MPEG 标准亦推荐块匹配法。

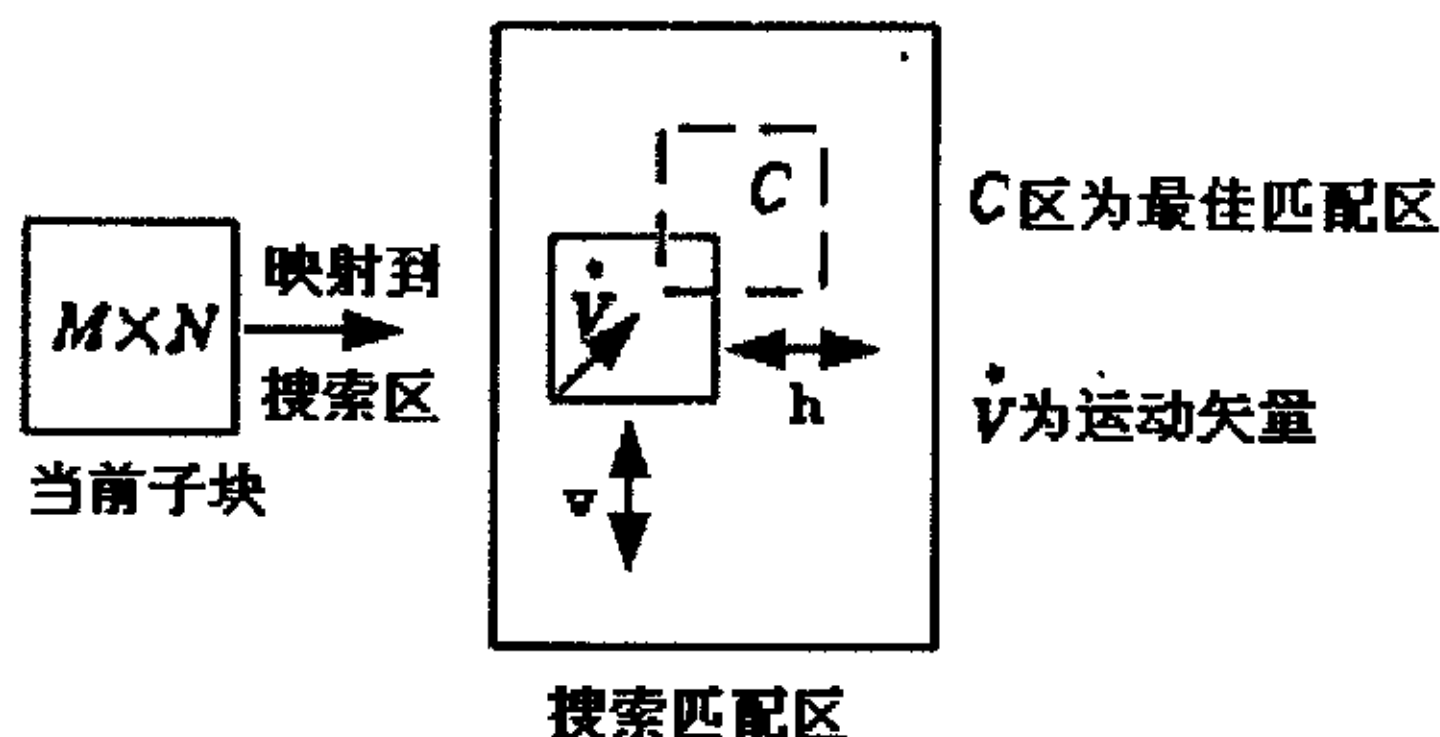


图 3.4 块匹配算法示意图

将当前帧（第 k 帧）均匀地划分为 M×N 图像宏块，并在上一帧（参考帧）内开辟大小为 (M+2h) × (N+2v) 的一块区域，用本帧宏块在参考帧的搜索区内寻找最优（匹配误差最小）的匹配块求得运动矢量（见图 3.4）。衡量匹配好坏的准则有最小均方误差（MSE）和最小绝对值误差（MAD），MAD 准则由于计算量小、硬件实现简单而得到广泛使用。MAD 准则定义为：

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

其中：i,j 分别为水平与垂直方向的偏移量；f_k 为第 k 帧当前块灰度值；f_{k-1} 为在参考帧第 k-1 帧搜索窗口内相对于当前块偏移为 (i,j) 的块。在 MPEG-4VM 中，

不是直接用 MAD 来作为误差准则,而是取其和一绝对误差和 SAD(Sum of Absolute Difference)来作为误差计算和判断的(如式(3)),其中的 f_k 和 f_{k-1} 分别为当前块与参考块的形状 alpha 掩码)。

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

在已知匹配准则的情况下,如何搜索匹配误差最小的子块就是搜索算法所要解决的问题。由于实际上一个块的运动分布一般集中在该块位置附近,因此,块匹配的搜索只需在一定的范围内进行。假设块的运动分布在水平和垂直方向上是各向均匀的(即不在其中某个方向上占优),那么搜索范围应该是以当前块所在位置为中心的正方形。显然,最佳的块匹配搜索算法是全搜索(Full Search, FS),即在搜索范围内逐个像素位置作块匹配运算,但此算法计算量很大。设搜索目的块的最大可能的位移为 w ,则 FS 的搜索次数为 $(2w+1)^2$ 。在实际的应用中 FS 由于巨大的计算量严重的影响了视频编码算法的整体效率,因而提出了很多改进的快速算法,有三步法^[46](3SS)、共轭方向搜索法^[46](CDS)、二维对数法^[46](LOGS)、新三步法^[36](N3SS)、交叉搜索法^[37](CS)、四步法^[38](4SS)、基于块的梯度下降法^[46](BBGDS)和钻石法^[39](DS)、两步搜索法^[40](TWSS)等。

3.3.2 对运动估计算法改进的可行性分析

上文提到的运动估计改进算法在搜索速度上都比 FS 有很大的提高,但在搜索结果上比 FS 都不同的有所降低。并且当搜索范围要求较大时,直接应用这些搜索算法会存在一些需要解决的问题。如 3SS、N3SS 等的搜索范围限制在 ± 8 个像素,LOGS 假设搜索目的块和预测块之间的位移差和两个块之间的匹配误差成正比(实际并非如此,这样可能回陷入局部误差最小的情况),DS 也有可能陷入局部最小的情况。所以搜索算法的改进还有很多工作要做,特别是形状编码的运动估计要求搜索范围为 ± 16 个像素,比纹理编码搜索范围要大的多。

在块匹配的搜索算法中,影响和决定搜索算法的计算复杂度和时间复杂度的主要因素有三点:1)搜索起始点的选择;2)搜索范围的确定;3)搜索方法的选择。

首先,搜索起始点的选择是第一步也是很重要的一步,如果起始点选的好,距离目标点很接近或就是目标点,那我们就只需要搜索很少的点或者不进行搜索直接找到匹配块。其次,虽然原则上是要搜索 $\pm 16 \times 16$ 的范围,但对有些运动比较小的块,我们一般在较小范围内就能找到一个满足误差要求的匹配块,那么就没必要全范围搜索了。还有,同一物体的运动矢量存在强的相关性,即物体的某个运动矢量与周围的运动矢量是有关联的,甚至可以说决定于其周围的运动矢

量。同时,当前帧运动矢量与前几帧也有很强相关性,只是在物体的边界存在突变,相关性才弱。一般情况下,在连续两帧的相同位置的运动矢量之间差异是很小的。

3.3.3 MPEG-4 的 VM 中形状编码的运动估计处理

MPEG-4 的 VM 中形状编码的运动估计在确定 BAB 块的编码模式后进行,如果该 VOP 是 B-VOP 或 P-VOP,那么就对待编码的 BAB 进行运动估计,得到运动矢量 MVs (MV for shape); 如果该 VOP 是 I-VOP,则该步可以省去。

在 MPEG-4VM 中将 MVs 分为两部分, $MVs = MVPs + MVDs$ 。

求 MVs 的步骤如下:

1) 先确定 MVs 的预测 MVPs。

MVPs 根据一些候选的形状信息运动向量 MVs 和对应于当前 BAB 的纹理宏块的运动向量 MV 得到。这些候选运动向量如图 3.5 所示,其中 MV1, MV2, MV3 取整数值。按照 MVs1, MVs2, MVs3, MV1, MV2, MV3 的顺序进行扫描(注意:在 MPEG-4 中纹理编码采用 4MV,即每个 BAB 内有 4 个运动向量^[8])。MVPs 取第一个有效的 MV 的值。如果没有一个有效,那么就取 $MVPs = 0$ 。

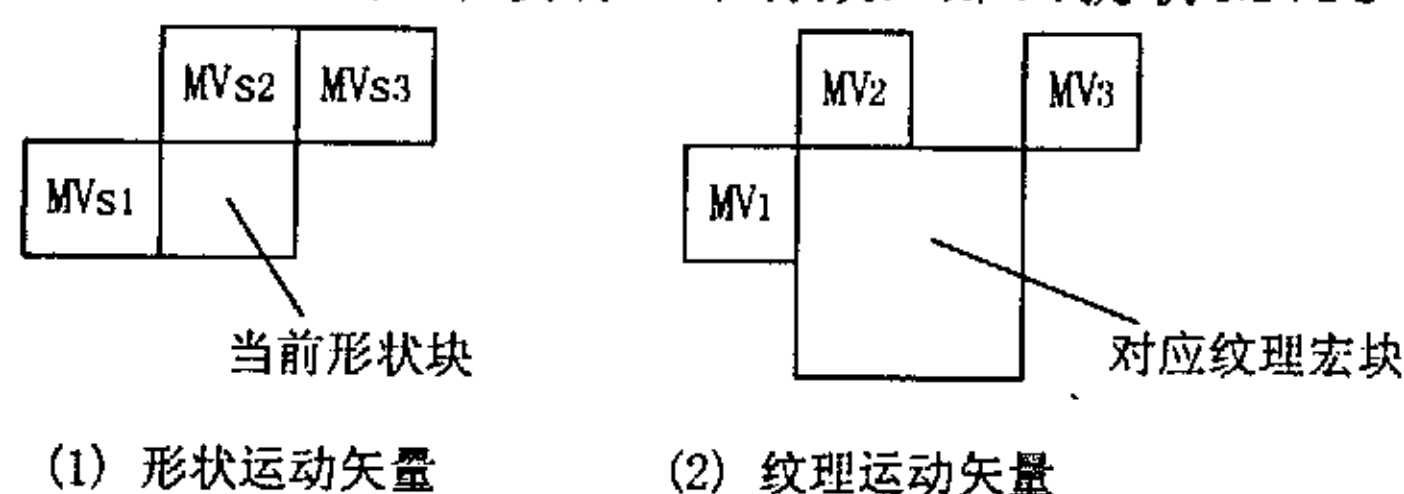


图 3.5 MVPs 的候选运动向量

2) 搜索运动矢量, 确定 MVDs。

接着根据确定的 MVPs,通过下面的过程计算 MVs。(1)计算当前 BAB 与 MVPs 指向的 BAB 的运动补偿差值,如果对于每一个 PB(4×4 的像素子块),差值都小于或等于 α_{th} ,则 $MVs = MVPs$,结束该过程。(2)如果上一步不满足,就在 MVPs 指向的 BAB 块周围 +/-16 像素的范围内进行搜索,并且计算绝对误差和(SAD)。使 SAD 最小的块就是运动补偿块,运动补偿块和当前块之间的位置偏移就是运动矢量 MVs,则形状信息运动向量差值 $MVDs = MVs - MVPs$ 。运动补偿只是根据 MVs 进行简单的运动更新来实现,即从参考 VOP 中将 MVs 指向的 BAB 以及它周围 1 个像素宽度的边框拷贝到当前 BAB 中,形成一个 18×18 的运动补偿的 BAB,用来进行帧间算术编码。

3.3.4 改进的算法

在本文中,提出了一种改进的形状编码的运动估计算法。

1、算法的整体思路

算法整体的思路如下:

- (1) 根据 BAB 块的属性(在 VOP 内,外,边界上)确定预测 MV 初始值;
- (2) 如果预测初始值满足误差要求,则转第五步;若不满足,则根据误差大小确定搜索范围和搜索方法;
- (3) 由选择的搜索方法在搜索范围内搜索。如果小的搜索范围不能搜索到比预测点初始误差小的点,则在大的搜索范围用相应的搜索方法搜索;
- (4) 如果搜索得到的目的 BAB 块满足误差要求,则转第五步,若不满足,则重新进行全搜索(FS);
- (5) 搜索结束,得到运动矢量和运动补偿块。

2、算法的具体实现

1) 预测初始块(值)的选取

通过分析可知,每个图像序列的连续帧之间运动向量的相关性是很强的。同时,每个 VOP 内部各块之间的运动相关性也是很强的。所以,只要我们的初始点能够选择和这块 BAB 运动相关性最强的块,就可以很快的找到该 BAB 的运动向量。从分析可以得出,每个 VOP 的边界部分运动向量变化较大,而 VOP 内部运动向量和其相邻块几乎是相同的。如,在测试序列 *brear* 中,不能由相邻块直接得到当前块运动向量的 BAB 中,VOP 内部块有 17 个,边界块有 503 个。这表明,绝大部分 VOP 内部 BAB 块,都可以通过将初始预测匹配块设为其相邻块,而大部分边界块却需要在初始运动向量的基础上进行搜索。

根据 VOP 运动向量的这些特征,我们将 BAB 块分成几类:VOP 内部 BAB, VOP 边界 BAB, VOP 外部 BAB。

对于 VOP 外部 BAB,不用处理,运动向量 $MV=0$ 。对于 VOP 内部 BAB,类似 VM 中的处理方法,按顺序搜索 MVs_1 、 MVs_2 、 MVs_3 (和 VM 中相似,如图 3.5),取第一个有效的值为初始预测运动向量。如果没有一个有效,初始预测运动向量为前一帧的运动向量平均值。如果这个初始预测运动向量已经满足误差要求,则这个向量即为所求运动向量,如果不满足,则以这个运动向量指向的 BAB 块为初始块进行搜索。对于 VOP 边界 BAB,我们又分为下面几种情况(如图 3.6)。

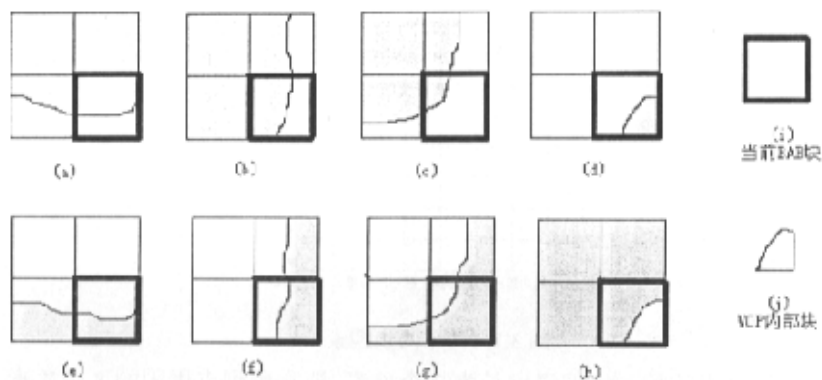


图 3.6 边界 BAB 块的几种主要情况

(1) 当前 BAB 块左边也为边界 BAB 块而上边无边界 BAB 块时 (如图 3.6 中 a, e), BAB 的运动向量和其左边相邻的 BAB 块相似, 所以预测运动向量取其左边相邻的 BAB 块的;

(2) 当前 BAB 块上边也为边界 BAB 块而左边无边界 BAB 块时 (如图 3.6 中 b, f), BAB 的运动向量和其上边相邻的 BAB 块相似, 所以预测运动向量取其上边相邻的 BAB 块的;

(3) 当前 BAB 块左边和上边都为边界 BAB 块时 (如图 3.6 中 c, g), BAB 的运动向量和其左边以及上边相邻的 BAB 块相似, 所以预测运动向量取其左边和上边相邻的 BAB 块的平均值;

(4) BAB 在其左和上方无相邻边界 BAB 块 (如图 3.6 中 d, h), 这时预测运动向量取前一帧的平均运动矢量作为其预测值。

所有边界 BAB 都可以归入以上四种情况之中。

2) 搜索范围的确定

根据视频特点和编码需要, 定义两个误差阈值 T_1 和 T_2 ($T_1 < T_2$)。由当前块和初始预测块的绝对平均误差 (SAD) 差值, 将运动块分为三类。

- 1) 当 $sad < T_1$ 时, 当前块为低速运动, 搜索范围为 ± 4 个像素;
- 2) 当 $T_1 \leq sad \leq T_2$ 时, 当前块为中速运动, 搜索范围为 ± 8 个像素;
- 3) 当 $sad > T_2$ 时, 当前块为高速运动, 搜索范围为 ± 16 个像素。

3) 搜索方法的选择

对于低速运动块, 采用简单的两步搜索 (图 3.7); 对于中速运动块, 采用三步法 (TSS, 见图 3.8); 对于高速运动块, 采用一种简单四步法。

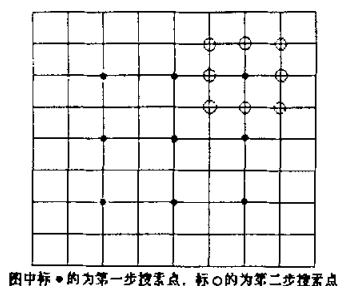


图 3.7 简单两步搜索

简单两步搜索的第一步搜索步长为 2 个像素，搜索预测点周围的 8 个像素点。第二步以第一步搜索的最小误差点为中心，搜索其周围相邻的 8 个像素点。简单两步法共搜索 17 个点。

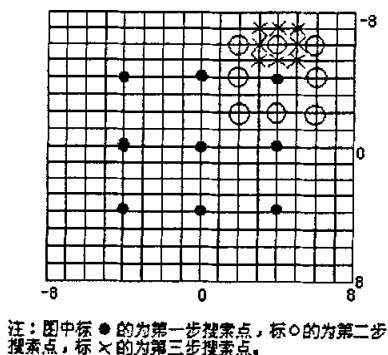


图 3.8 三步搜索法 (TSS) 的搜索步骤

三步法 (TSS) 是运动估计的一种很好的算法，快速而且高效，它基本上保持了 FS 的性能，但其计算量只有 FS 的 10% 左右。TSS 是通过三步搜索，逐步减小搜索步长。每次搜索都是以上一步的搜索结果为中心，进行周围一定步长的 3×3 像素的搜索，搜索精度为 1 个像素。TSS 共搜索了 25 个点。

对于高速运动块采用一种类似于三步法的简单四步法。开始以 8 个像素的步长，搜索窗口边缘的 8 个点。以后每步以前一步的搜索最小点为中心，步长为前一步的一半进行搜索，直到搜索步长减为 1 个像素。每次搜索点的数目为 8 个，加上初始点，总共搜索 33 个点。(图略)

如果，按照上面的方法，最终搜索到的块不能满足误差阈值限定，就需要对这一块进行 FS 搜索，当然这样的块是非常少的。

由于采用的是块匹配法，并不是全像素匹配，同时由于存在某些局部运动的变形，可能有些 BAB 块，并不能在其参考帧中找到完全匹配的块或者不能找到匹配误差比预测块小的块。所以，上文中的三种算法中都找的是所有比较块中匹配误差最小的块；但在 FS 搜索时，因为已经搜索了预测块周围比较大范围的很

多像素，所以在进一步搜索搜索范围内其它点时，只要找到的某个 BAB 与当前块 BAB 的匹配误差小于预测块的误差，就认为找到的这个 BAB 块已经满足需要，认为找到了当前块的运动向量。虽然这样存在一些误差，但考虑到经过三次搜索后需要进行全搜索的点已经很少，并且这些很少的点中还有一些可能在整个搜索范围内不存在比预测块匹配误差更小的匹配块。

4) 搜索过程的优化

在搜索过程中，不可避免后面的搜索算法可能存在重复搜索前面搜索算法中已经搜索过的点的问题。对于这种情况，一些文献建议通过建立设定一个需搜索点队列和一个待比较点队列，将要搜索的点插入搜索队列，将需比较的点插入比较队列。如果某个点已搜索过，则从搜索队列删除这个点。如果这个点的 MAD（均方误差）计算结果在下次比较中已经无用，则从比较队列删除该点。这样虽然在存储空间上有小小的开销（最多 16 个字节），但却去除了很多搜索上的冗余。由于本文的三种搜索算法在处理上非常相似，所以可以很容易的处理重复搜索点的问题，并且不需要另外多增加存储空间。本文的后面搜索算法可以看成是前面搜索算法在范围上的扩充，如果后面的搜索算法在外围点没有搜索到比预测点误差更小的匹配块，就直接进入后面的搜索，因为它下一步要搜索的点在前面搜索中已经搜索过了。对于那些需要进行 FS 搜索的点，只需要将前面已经搜索过的很规律的点跳过即可。

3.3.5 试验结果和结论

本文使用 MPEG-4 专用测试序列对改进算法进行了仿真测试，试验结果见下表（搜索范围为 $-16 \sim 16$ 个像素）。从表中可以看出，改进运动估计算法在对运动量大的块保证较大的搜索范围的前提下，不仅减少了需要搜索的宏块数目（由于采用了合理科学的初始值预测计算方法，可以使更多的宏块直接得到运动向量，跳过后面的搜索。），而且平均每个宏块的搜索点数也明显下降。改进算法在搜索的宏块数目上比 VM 中的算法减少了 6%，在平均搜索点数上比全搜索法降低 97%。由于搜索初始值的优化和搜索范围的保证，在估计质量上，改进算法也比传统搜索法有了一定的改善。同时这种算法相对比较简单，易于软件的实时处理。

表 3.1 平均搜索次数比较

搜索序列	全搜索算法 (VM)		本文算法	
	需要 搜索的宏块	平均 搜索点数	需要 搜索的宏块	平均 搜索点数
Bream	328	1078	311	33.4

3.4 MPEG-4 形状编码中尺寸转化的研究

MPEG-4 中的形状编码主要分为四步, 编码模式的确定、运动估计和运动补偿、尺寸转化(确定分辨率)和基于上下文的算术(CAE)编码。本文对 VM 中尺寸转化算法进行了详细研究, 并提出了改进算法。

3.4.1 VM 中的尺寸转化

尺寸转化在编码模式确定和运动估计之后, 在算术编码之前。它主要是在确定编码模式的前提下对需要算术编码的模块再进行一次分类, 看某些模块能不能在编解码误差容许的范围内进行更高效的压缩。

二值形状编码的 BAB(二值 alpha 块)编码模式共有 7 种(见论文 3.1 部分), 尺寸转化钟对的是 intra_CAE 、 $\text{MVDs}=0 \& \& \text{interCAE}$ 和 $\text{MVDs}!=0 \& \& \text{interCAE}$ 三种模式。尺寸转化的比率(CR)为 $1/4$, $1/2$ 或者 1(原始尺寸)。CR 为 $1/4$ 和 $1/2$ 的 BAB 块经过解码后存在一些信息丢失, 但这是在误差容许的范围内的, 即尺寸转化是受误差阈值限制的。

图 3.9 表示了尺寸转化的过程。每个 $M \times M$ 的 BAB 下采样为 $(M \times CR) \times (M \times CR)$, 然后再上采样为原始尺寸 $M \times M$ 。(注意: 形状编码中 0 表示透明(像素不在 VOP 内), 255 表示不透明(像素在 VOP 内)。)

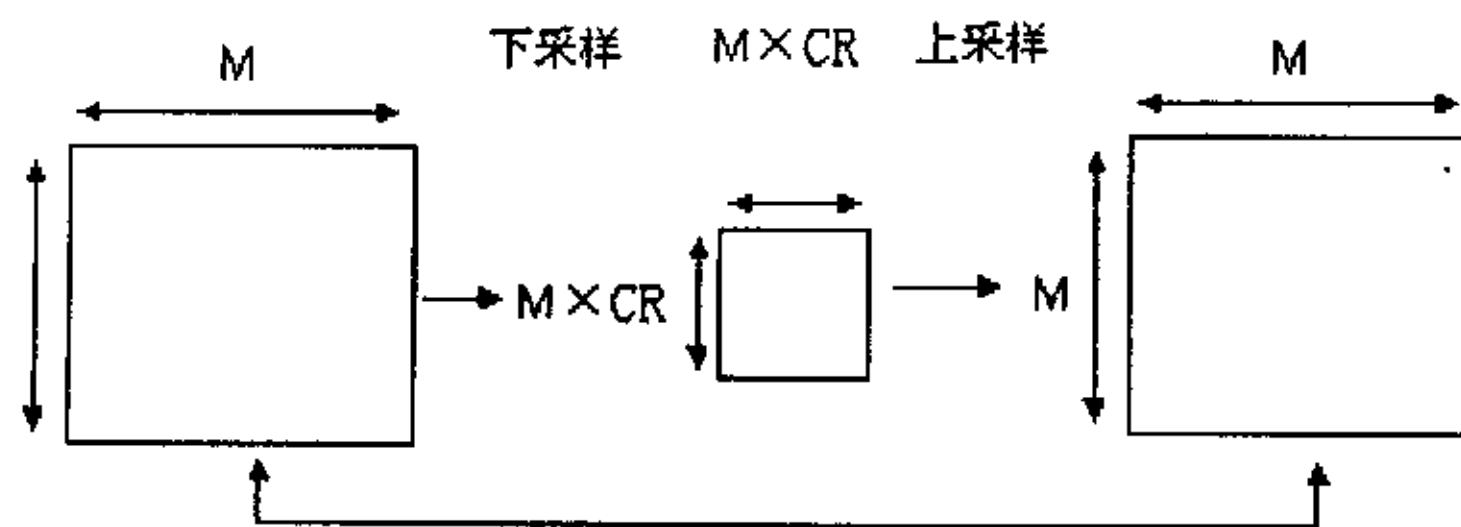


图 3.9 尺寸转化.

在 VM 中, 尺寸转化的具体步骤如下: 先设原 BAB 块的 $\text{CR}=1/4$ 进行下采样和上采样, 如果重构 BAB 块与原 BAB 块的 SAD 误差不大于误差阈值, 则该 BAB 块的 CR 为 $1/4$; 如果重构 BAB 块的 SAD 误差大于误差阈值, 则设原 BAB 块的 $\text{CR}=1/2$ 再进行下采样和上采样, 如果重构 BAB 块与原 BAB 块的 SAD 误差不大于误差阈值, 则该 BAB 块的 CR 为 $1/2$, 否则该 BAB 块的 CR 为 1。

1) 下采样

当 $\text{CR} = 1/2$ 时, 下采样的子块为 2×2 , 当 $\text{CR} = 1/4$ 时, 下采样的子块为 4×4 。如果原 BAB 块中 2×2 (或者 4×4) 的像素子块中的 4 (或者 16) 个像素的平均值大于或等于 128, 下采样的像素值设为 255 (不透明), 否则设为 0 (透明)。下采样的结果是一个新的 8×8 (或者 4×4) 的像素块。

2) 边界扩展

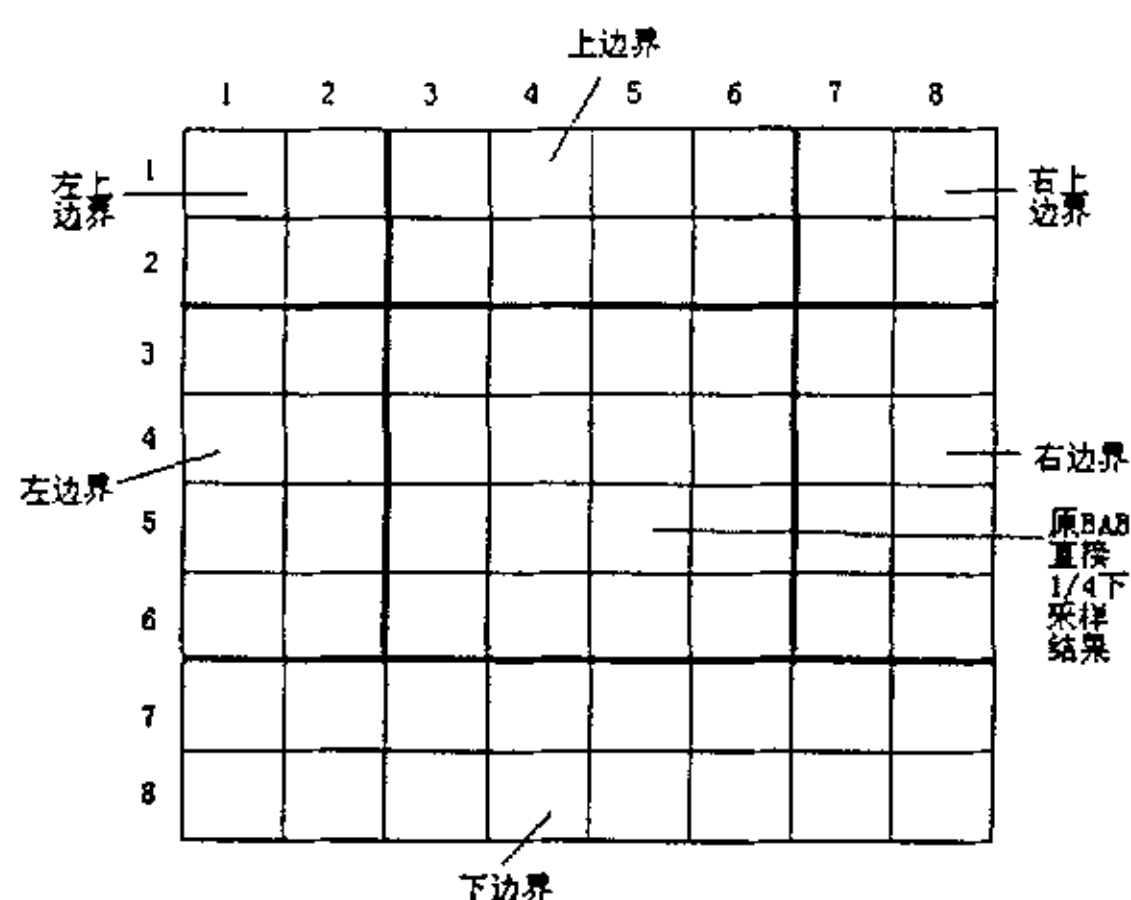


图 3.10 1/4 下采样块经过边界扩展后的结果

为了后面的上采样需要，必须对下采样后的采样块进行边界扩展。MPEG-4 中 BAB 块的尺寸为 16×16 ，1/4 采样后尺寸为 4×4 ，经过边界扩展变为 8×8 的尺寸（上下左右边界各扩展 2 各像素，见图 3.10），1/2 采样后尺寸为 8×8 ，经过边界扩展变为 12×12 的尺寸。

边界扩展包括左、上边界的扩展和右、下边界的扩展。上边界的扩展是将原始图像帧中该 BAB 所在位置的上边两行像素取相邻 4（1/2 采样时取 2 个）像素的平均值，当平均值大于或等于 128 时，扩展像素设为 255（不透明），否则设为 0（透明）。左边界的扩展和其类似，将左边两行像素取平均值即可。左上角（右上角）直接将扩展后的上边两行像素的左边（右边）两个像素复制即可。下边界和右边界的扩展是将内部靠近边界的两行像素直接复制而得。

3) 上采样

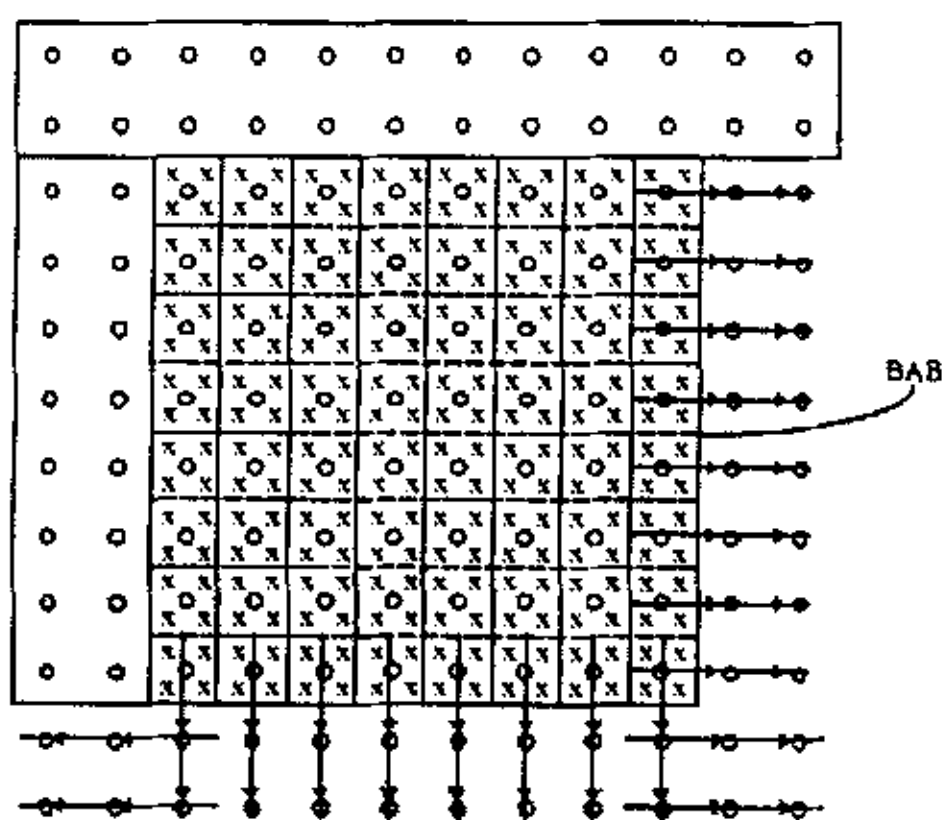


图 3.11 上采样

当 CR 不为 1 时，执行上采样。插值像素（图 3.11 中的“X”为插值像素“O”为编码像素）由上下文（相邻像素的）滤值器（filter context）计算出来。为了插值计算，“0”表示透明像素，“1”表示不透明像素。

插值像素的值由以下方程计算出来:

$$P1 = (4*A + 2*(B+C+D) + (E+F+G+H+I+J+K+L) > Th[C_f])?1:0$$

$$P2 = (4*B + 2*(A+C+D) + (E+F+G+H+I+J+K+L) > Th[C_f])?1:0$$

$$P3 = (4*C + 2*(B+A+D) + (E+F+G+H+I+J+K+L) > Th[C_f])?1:0$$

$$P4 = (4*D + 2*(B+C+A) + (E+F+G+H+I+J+K+L) > Th[C_f])?1:0$$

($P_i, i=1,2,3,4$, 如图 3.12)

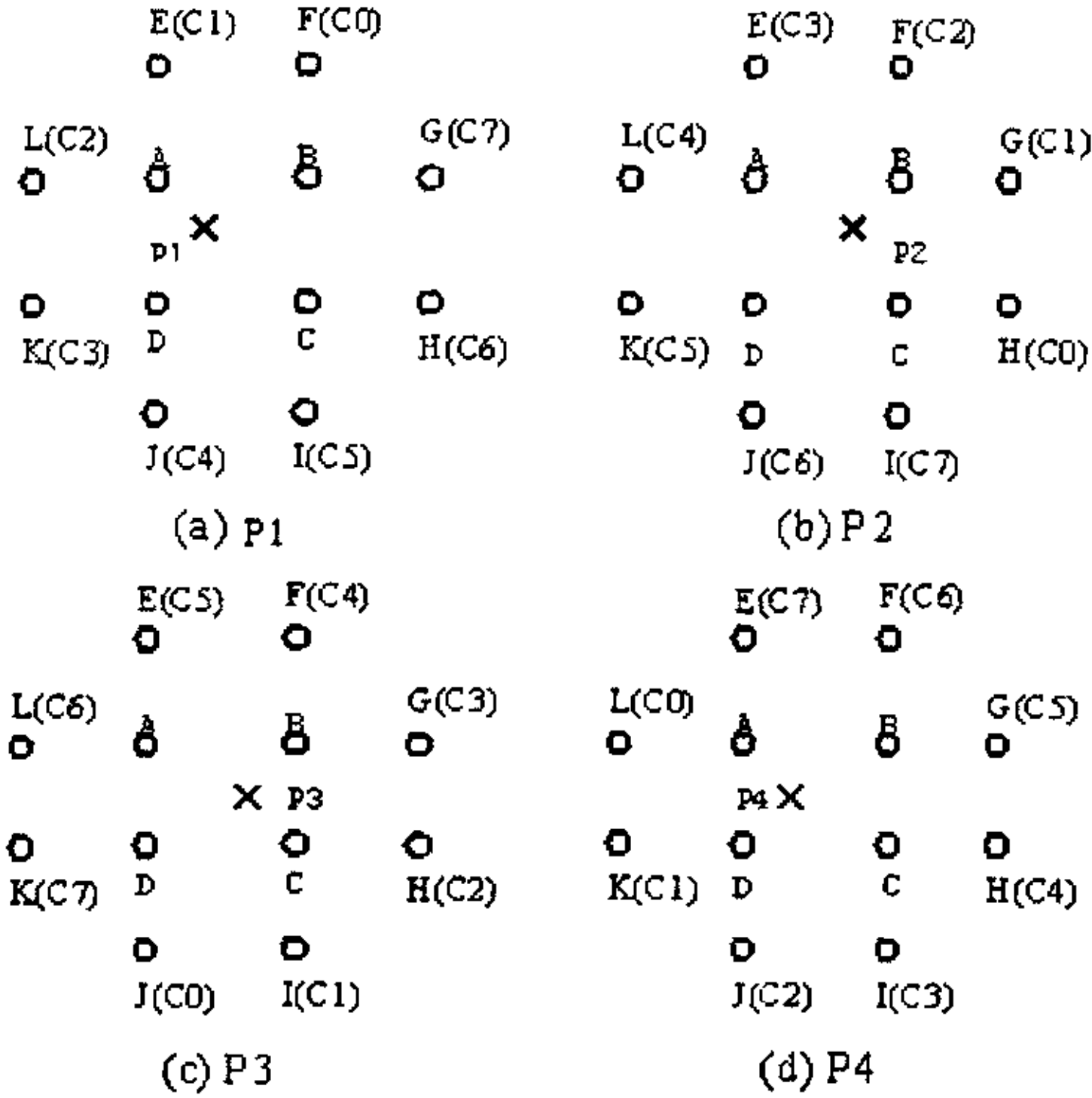


图 3.12 插值滤值和插值构造

8 位的上下文滤值器 C_f 由下式计算:

$$C_f = \sum_k c_k \cdot 2^k$$

在计算 C_f 的基础上, 阈值 ($Th[C_f]$) 能够通过查找表 (见附录) 获得, 边界扩展如上面所述。当 $CR=1/4$ 时, BAB 首先插值为 $CR=1/2$ 的尺寸, 然后对插值结果再进行插值为 $CR=1$ 的尺寸。

4) 判断重构块质量是否能够被接受

CR 的确定由原始 BAB 和经过一次下采样再经过一次上采样重建的 BAB 块的 SAD(Sum of Absolute Difference)误差决定。转化误差就是分别计算每个 4×4 PB 块(Pixel Block)的绝对值误差和。如果这 16 个 PB 块中有任何一个的 $SAD > Alpha_th$, 则这个重构 BAB 不能满足误差要求, 将不能被接受。如果重构块满足误差要求, 则将 CR 设为当前值。如果 $CR=1/4$ 不能被接受, 则将 CR 设为 $1/2$ 再进行尝试, 如果 $CR=1/2$ 不能被接受, 则将 CR 设为 1, 即保持原始尺寸。

如果下采样 BAB 结果是全透明或全不透明, 并且任何 4×4 子块的转化误差都等于或小于 Alpha_th , 这个 BAB 的编码模式为 ALL_0 或 ALL_255。除非 CAE 编码是在码率控制的尺寸确定算法之后执行的 BAB。

3.4.2 尺寸转化的特点和作用

尺寸转化一方面是为了提高压缩率, 另一方面是为了后面的码率控制和可扩展编码服务。通过试验发现(见表 1 和表 2), 如果增大形状编码的 Alpha_th , 总的需尺寸转化的块数目减少(这主要是因为当 Alpha_th 增大时, 可以使 ALL_0 和 ALL_255 编码的 BAB 数目增加, $\text{MVDs}==0 \& \&\text{interCAE}$ 和 $\text{MVDs}!=0 \& \&\text{interCAE}$ 编码的 BAB 数目减少。), 需尺寸转化的 BAB 中转化率为 $1/4$ 和 $1/2$ 的数目增加。当然, 尺寸转化在提高压缩率的同时, 会减少编码码率, 降低解码图像的质量。所以可以利用这些特点, 通过改变 Alpha_th 来实现码率控制和可扩展编码。

3.4.3 原算法中存在的问题分析

由 VM 中的尺寸转化算法可以看出, 每次尺寸转化都是先用 $\text{CR}=1/4$ 进行尝试, 如果结果不能接受再用 $\text{CR}=1/2$, 如果还不能接受则设定 $\text{CR}=1$ (保持原始尺寸不变)。如果 BAB 最终的 $\text{CR}=1/2$ 或 $\text{CR}=1$, 要进行两次尺寸转化。每次尺寸转化要进行一次下采样和一次上采样, 特别是上采样时要进行复杂的插值计算, 运算量是比较大的。

其实直接决定 $\text{CR}=1/4$ 、 $1/2$ 或 $\text{CR}=1$ 只需要一次尺寸转化, 所以如果能够尽量减少两次尺寸转化的数目, 就可以使运算量大大减少。本文就是从这一点出发来对算法进行改进的。

3.4.4 算法的改进

由表 3.2 和表 3.3 的试验结果可以看出, 当误差阈值 Alpha_th 增大时, 总的需尺寸转化的块数目在减少, 能够尺寸转化的块特别是 $\text{CR}=1/4$ 的 BAB 所占比例却大大增加。当 Alpha_th 小于一定的值时, $\text{CR}=1/2$ 的 BAB 和 $\text{CR}=1$ 的 BAB 所占比例大于 $\text{CR}=1/4$ 的 BAB。当 Alpha_th 大于这个值时, $\text{CR}=1/4$ 的 BAB 所占比例大于 $\text{CR}=1/2$ 的 BAB 和 $\text{CR}=1$ 的 BAB。这样, 我们就可以取一个中间值 mid_th , 当 $\text{Alpha_th} \leq \text{mid_th}$ 时, 先进行 $\text{CR}=1/2$ 的尺寸转化尝试, 满足误差要求, 再进行 $\text{CR}=1/4$ 的尺寸转化尝试。当 $\text{th} > \text{mid_th}$ 时, 先进行 $\text{CR}=1/4$ 的尺寸转化尝试, 如果不满足误差要求, 再进行 $\text{CR}=1/2$ 的尺寸转化尝试。显然, 这样将大大的减少两次尺寸转化的 BAB 数目, 减少运算量。

其次,下采样时,直接构造出两个采样数组(矩阵),一个为 4×4 的,一个为 8×8 的。这样虽然会增加一些存储开销(16 个字节),但却能减少很多重复计算(因为每次下采样都要计算一次)。

在采样过程中,VM 中将平均值为 128 的像素都设为不透明(255),这会使采样结果比原始图像边缘增加。本文算法中将平均值为 128 的像素每隔一个设为不透明(255),相邻的一个平均值为 128 的像素每隔一个设为透明(0)。这样将提高图像的重构 BAB 和原始 BAB 的近似性,有助于改进图像编码质量。

改进算法的伪码表示如下:

```

if(Alpha_th>mid_th)
{ CR=1/4;
  下采样; //构造  $4 \times 4$  和  $8 \times 8$  的两个数组
  扩展边界; //将  $4 \times 4$  数组扩展为  $8 \times 8$  的数组
  上采样; //由  $8 \times 8$  的数组重构 BAB
  if(重构 BAB 质量不能接受)
  { CR=1/2;
    扩展边界; //将  $8 \times 8$  数组扩展为  $12 \times 12$  的数组
    上采样; //由  $12 \times 12$  的数组重构 BAB
    if(重构 BAB 质量不能接受) CR=1;}
  }
else
{ CR=1;
  下采样; //构造  $4 \times 4$  和  $8 \times 8$  的两个数组
  扩展边界; //将  $8 \times 8$  数组扩展为  $12 \times 12$  的数组
  上采样; //由  $12 \times 12$  的数组重构 BAB
  if(重构 BAB 质量能接受)
  { CR=1/2;
    扩展边界; //将  $4 \times 4$  数组扩展为  $8 \times 8$  的数组
    上采样; //由  $8 \times 8$  的数组重构 BAB
    if(重构 BAB 质量能接受) CR=1/4;}
  }

```

3.4.5 试验结果和结论

本文使用的测试序列采用的是 bream_cif, 图像大小为 352×288 , 共 10 帧图像。试验中 mid_th=8, 结果见表 3.2 (InterCAE BAB) 和表 3.3 (IntraCAE BAB)。

当 $\text{Alpha_th} \leq \text{mid_th}$ 时, $\text{CR}=1/4$ 的 BAB 原来需要 1 次尺寸转化, $\text{CR}=1/2$ 和 $\text{CR}=1$ 的 BAB 原来需要 2 次, 现在 $\text{CR}=1$ 的 BAB 需要 1 次, $\text{CR}=1/2$ 和 $\text{CR}=1/4$ 的 BAB 需要 2 次; 当 $\text{Alpha_th} > \text{mid_th}$ 时, 保持原来的尺寸转化次数不变。(虽然 $\text{Alpha_th}=8$ 时, IntraCAE 的 BAB 尺寸转化增加了 10 次, 但结合 IntraCAE 的 BAB, 总的转化次数还是减少了 13 次。) 原来需要两次尺寸转化的 BAB 块(即 $\text{CR}=1/2$ 和 $\text{CR}=1$ 的 BAB) 都需要两次下采样, 现在任何 BAB 只需要一次下采样。

表 3.2 InterCAE BAB 的实验结果

Alpha_th	需要尺寸 转化的 BAB	CR=1/4 的 BAB	CR=1/2 的 BAB	CR=1 的 BAB	减少的尺 寸转化次 数	减少的下 采样次数
0	398	0	0	398	398	398
2	250	0	16	234	234	250
4	142	2	19	121	119	140
8	65	7	28	30	23	58
16	23	10	9	4	0	13
32	9	6	1	2	0	3
64	8	7	0	1	0	1

表 3.3 IntraCAE BAB 的实验结果

Alpha_th	需要尺寸 转化的 BAB	CR=1/4 的 BAB	CR=1/2 的 BAB	CR=1 的 BAB	减少的尺 寸转化次 数	减少的下 采样次数
0	59	0	0	59	59	59
2	54	1	9	44	43	53
4	53	6	13	34	28	47
8	49	20	19	10	-10	29
16	43	33	10	0	0	10
32	38	38	0	0	0	0
64	25	25	0	0	0	0

由表中可以看出, 采用改进算法, 很好的减少了尺寸转化的次数, 提高了编码速度, 减少了运算量。同时, 由于在下采样时对平均值为 128 的块部分设为透明(0), 部分设为(255), 改善了编码质量, 优化了算法。

第四章 可扩展编码的研究

可扩展编码是网络多媒体编码的一种重要技术。最近几年来,可扩展编码的研究和发展非常迅速。本章对可扩展编码的发展进行了简要介绍,然后结合 MPEG-4 中的精细可扩展编码思想,提出了一种新的空域可扩展形状编码算法。

4.1 可扩展编码简介

4.1.1 可扩展编码的背景和发展情况

20 世纪 90 年代中后期,Internet 迅猛发展,无线通信也迅速在全球普及,因此人们开始有了在网络上传输视频和图像的愿望,于是视频和图像编码的目标也就从传统的面向存储变为现在的面向传输。在网络上传输视频和图像等多媒体信息除了要解决误码问题之外,最大的挑战在于用户可以获得的带宽在不停地变化。这主要是因为网络的异构性,即各子网的网络资源,包括处理能力、带宽、存储和拥塞控制策略等分布得很不均匀,用户通过不同的通信子网传输数据会体验到非常不同的实际传输吞吐量、数据丢失率、传输延迟;另外由于现在的网络没有提供资源预留一类的协议保障,当网络拥塞发生时,多媒体数据流的有效传输带宽会突然降低,影响用户接收多媒体信息的质量和速度,甚至造成无法观看。

为了适应网络带宽的变化,可扩展性编码的思想被提了出来。可扩展性编码(也称分层可扩展性编码,Layered Scalable Coding)就是将多媒体数据压缩编码成多个流,其中一个可以独立解码,它称为基本层码流;其它的码流称为增强层,它们不可以单独解码,而只能与基本层和它以前的增强层联合在一起解码,用来提高观看效果。可扩展性编码主要分为时域可扩展性编码、空域可扩展性编码和质量可扩展性编码。在这些策略中,编好的码流可以按层为单位截断,具有一定的网络带宽适应能力。新一代的视听对象编码的国际标准 MPEG-4 中的精细可扩展性^[50](Fine Granular Scalable,FGS)视频编码更提供了完全可扩展的增强层码流,它可以在任何地点截断,具有极强的网络带宽适应能力。后来又提出了渐进的精细可扩展性^[53](Progressive Fine Granular Scalable,PFGS)的编码方案,它保留了 FGS 的所有优良性能,并且把编码效率提高了将近 1dB。最新的静止图像编码国际标准 JPEG2000,它利用小波变换、位平面编码和基于上下文的算术编码等一系列新技术,将图像编码的效率提高了一大步(同等质量下,JPEG2000 的码流尺寸只有 JPEG 的一半),并且提供可扩展性的码流。在当前的网络时代,视频和图像编码的目标从产生适合存储的固定尺寸的码流发展到产生适合一

定的传输码率范围的可扩展性码流, 因此今后的多媒体数据编码必然要支持可扩展性。当然, 可扩展性编码现在还是一个很不成熟的领域, 与不具有可扩展性的单层编码相比, 它的编码效率还是比较低的 (例如 FGS 的编码效率比 MPEG-4 非可扩展性编码低 2~3dB)。如何将可扩展性编码的效率尽可能地逼近非可扩展性编码, 是一个值得继续研究的问题。

4.1.2 分层可扩展性编码

分层可扩展性编码技术已经被许多现行的国际压缩标准接纳, 例如 H.263+ 和 MPEG-4。分层可扩展编码主要有三种: 时域可扩展性、空域可扩展性和质量可扩展性。其中质量可扩展性也称为 PSNR 可扩展性, 原因是视频质量通常用峰值信噪比 (PSNR) 来衡量。

1、时域可扩展性 (Temporal Scalability)

时域可扩展性编码通常是通过在码流中添加 B 帧来实现的。B 帧是使用与它在时间上最近邻的前后两个 I 帧或 P 帧来预测的, 而自己并不作为任何其它帧的参考图像, 因此在传输中丢弃 B 帧并不影响其它帧的质量, 而仅仅降低帧率。图 4.1 是 MPEG-4 的时域可扩展性编码的示意图。

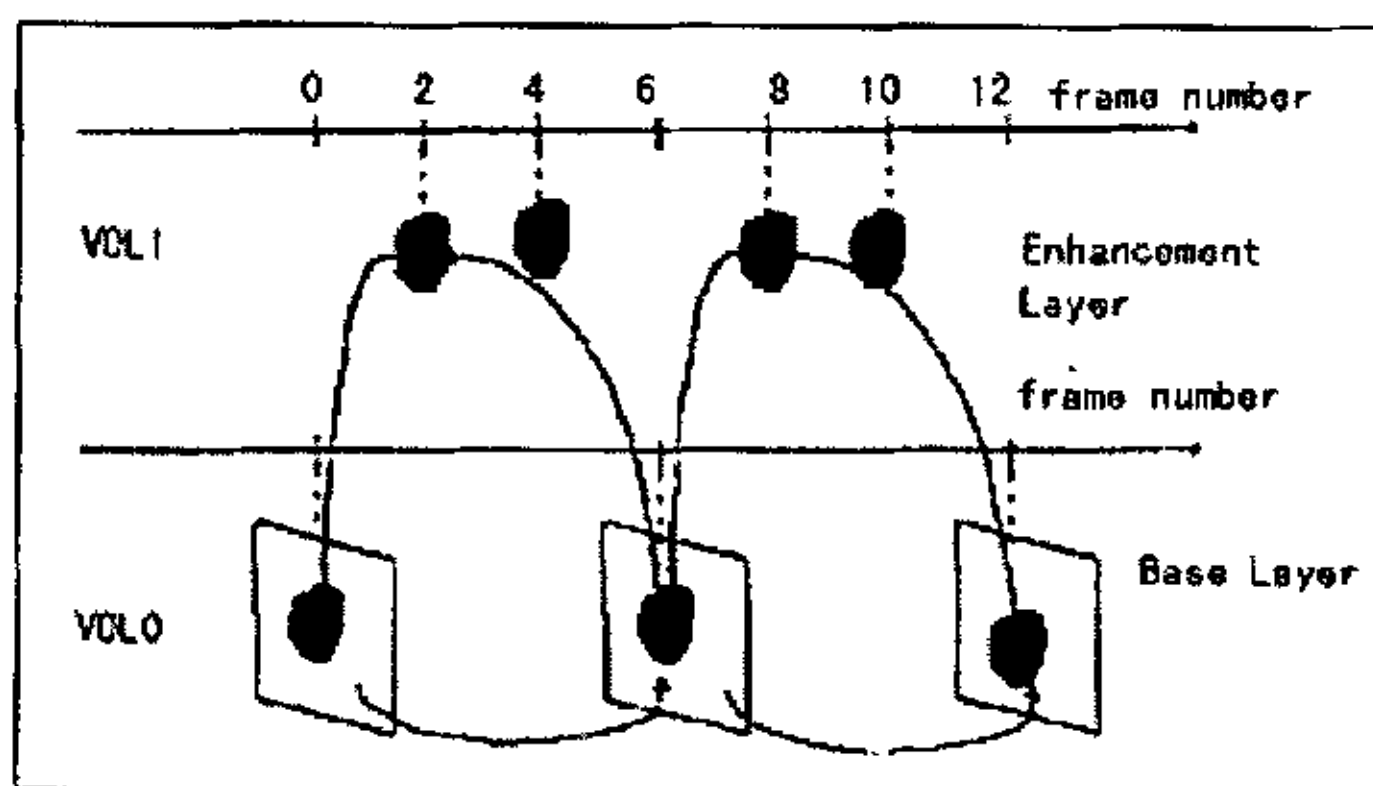


图 4.1 MPEG-4 中的时域可扩展性编码

2、空域可扩展性 (Spatial Scalability)

空域可扩展性编码是通过为视频中的每一帧都创建多分辨率的表示来实现的。当进行空域可扩展性编码时, 原始视频首先通过下采样得到低分辨率的视频, 编码得到基本层码流; 然后编码原始视频和基本层视频的差生成增强层码流。不过空域可扩展性编码在视频传输中应用较少, 因为任何一个用户都不能接收在前一个 GOP 中观看高分辨率视频, 而到下一个 GOP 只能获得低分辨率视频。因此即使增强层在传输中被丢弃, 客户端的解码器也要对低分辨率的图像进行插值, 这实际上是一种质量可扩展性的特殊情形。图 4.2 为空域可扩展性编码的示意图。

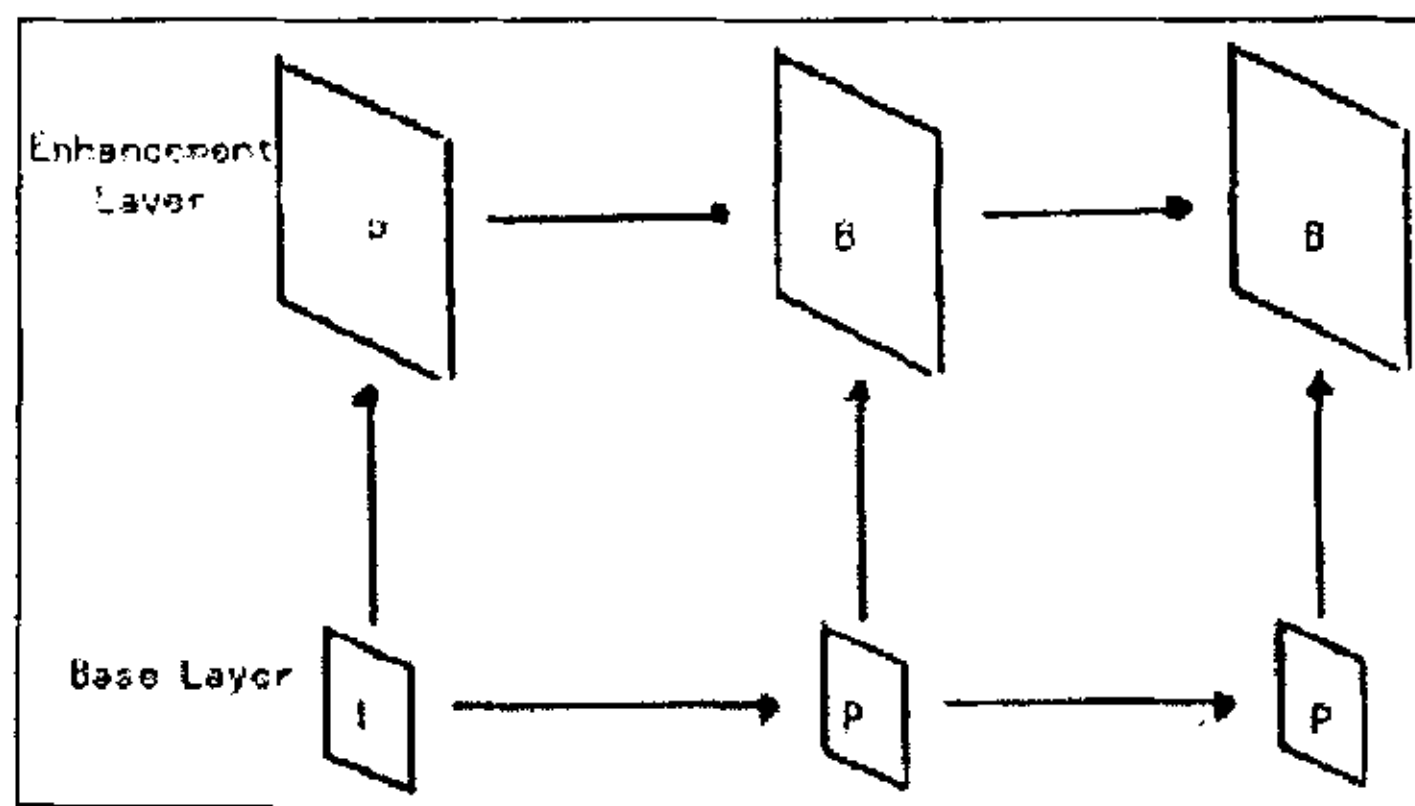


图 4.2 空域可扩展性编码

3、质量可扩展性(PSNR Scalability)

质量可扩展性编码的思想和空域可扩展性编码很类似，只不过这里不需对原始视频进行下采样，而是进行一次很粗的量化形成基本层码流。然后对原始视频和基本层视频的差再进行一次量化，生成增强层码流；如果有多个增强层码流则重复上面的过程。图 4.3 是质量可扩展性编码的示意图。

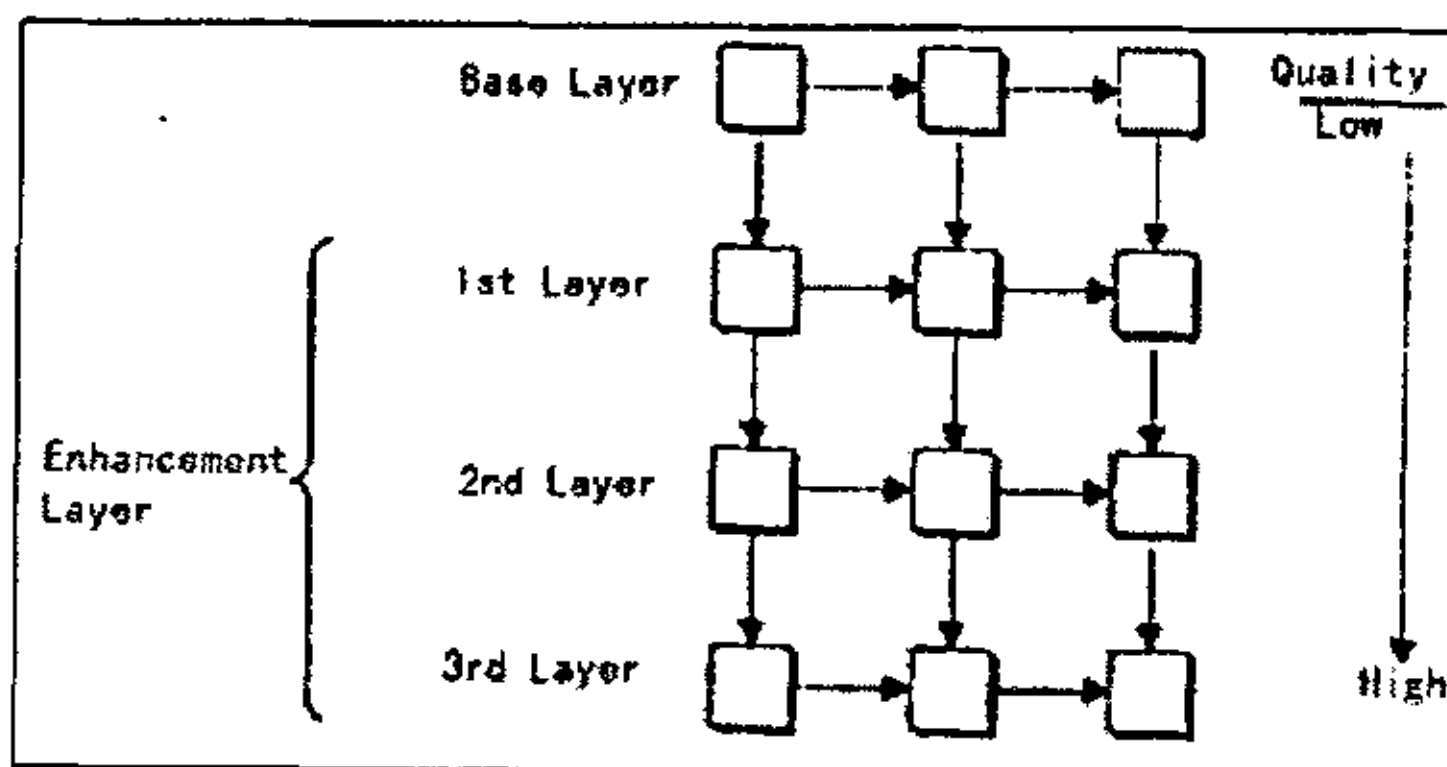


图 4.3 质量可扩展性编码

4.1.3 适合网络传输的编码方案

从以上可以看出，传统的视频编码方法不适合网络传输，其根本原因是它们的目标是将视频压缩成为适合一个或几个固定码率的码流，也就是说传统的视频编码是面向存储的。而由于网络的异构性和缺乏 QoS (Quality of Service) 保证，带宽在一个很大的范围内变化，因此面向网络传输的视频编码的目标是将视频压缩成为适合一个码率的范围。MPEG 组织看到了网络传输对视频编码的新要求，于是开始征集精细的可扩展性视频编码方案。它的基本思想是将视频编码成一个可以单独解码的基本层码流和一个可以在任何地点截断的增强层码流，其中基本层码流适应最低的网络带宽，而增强层码流用来覆盖网络带宽变化的动态范围。世界各地的学者提出了许多解决方案，主要分为三类，分别使用 DCT 变换技术、小波变换技术和 Matching Pursuit 技术。最终从编码效率和复杂性两方面的考虑，

精细可扩展性编码 (FGS) 及渐进的精细可扩展性编码 (PFGS) 被 MPEG 组织采纳。

1、精细的可扩展性编码

MPEG-4 的 FGS 主要有以下一些特性: 第一, 基本层使用基于分块运动补偿和 DCT 变换的编码方式达到网络传输的最低要求; 第二, 增强层使用位平面编码技术对 DCT 残差进行编码来覆盖网络带宽的变化范围; 第三, 每一帧的增强层码流可以在任何地点截断; 第四, 解码器重建的视频质量和收到并解码的比特数成正比。FGS 的基本层编码和普通的 MPEG-4 非可扩展性编码相同, 都是由运动估计、运动补偿、DCT 变换、标量量化和变长编码 (Variable Length Coding) 组成。而在增强层编码时, 从原始的 DCT 系数中减去基本层逆量化后重建的 DCT 系数值获得 DCT 残差, 然后对每一个 8×8 的块按从上到下从左到右的顺序使用位平面进行编码。使用位平面编码的好处就是使得每一个系数的重要部分 (较高的位) 优先编码, 这样在码流截断时就保证了解码器仍可以获得有关视频的重要信息, 从而提供了精细可扩展的特性。一般的 FGS 只考虑了要编码系数的值, 实际上 FGS 的应用是非常灵活的。例如我们可以对图像中比较感兴趣的前景部分优先编码, 具体作法就是将图像中的某些块上移若干个位平面, 就实现了对图像中感兴趣部分的优先传输。我们还可以对不同频率的 DCT 系数加以不同的权重, 也就是对不同的 DCT 系数上移不同的位平面, 从而满足人眼对不同频率成分的敏感程度。FGS 还可以和时域可扩展性编码相结合 (FGST), 即对 B 帧中的 DCT 系数都使用位平面技术编码, 这样 FGST 不仅保持了 FGS 的精细可扩展的特性, 而且支持帧率的变化。FGS 虽然具有优良的可扩展特性, 但它也有致命的弱点, 即效率太低。在同等码率下, FGS 的质量要比 MPEG-4 中的非可扩展性编码低 2~3dB (3 个 dB 意味着码率翻一番), 这是人们难以接受的。因此要想提高 FGS 的编码率, 必须改善它用作运动预测和补偿的参考图像的质量。

2、渐进的精细可扩展性编码

可以看出, 质量可扩展性编码 (我们对它的增强层使用位平面编码) 和 FGS 编码分别走了两个极端, 质量可扩展性编码保证了编码效率, 因为它用同层的解码图像作参考, 获得了较为准确的运动预测和补偿, 但它对错误极为敏感, 一旦某个增强层出现了错误, 它后面的增强层都将无法解码, 直到遇到下一个 I 帧为止; 而 FGS 保证了对错误的修复能力, 它可以从前一帧增强层的任何错误修复出来, 但是由于参考图像质量低, 因而效率不高。为了在编码效率和错误修复能力之间取得一个权衡, 微软亚洲研究院的吴枫、李世鹏、张亚勤等人提出了一种被称为渐进的精细可扩展性 (Progressive Fine Granular Scalable, PFGS) 的视频编码方案, 在保持了 FGS 所具有的网络带宽自适应和错误修复能力的同时, 提高了编

码效率将近 1dB。但是, PFGS 的编码效率和非可扩展性编码相比, 还有 1~2dB 的差距。

4.2 一种新的二值形状空域渐进可扩展编码算法

可扩展编码是 MPEG-4 标准的一个重要特点和内容。上一节介绍了可扩展编码的历史和发展情况, 从中可以看出, 虽然可扩展编码相比于开始提出时已经有很大发展, 但从编码质量和效率上还有很多可以继续发展和研究的地方。

本文在分析 VO (视频对象) 的形状信息特点基础上, 提出了一种新的基于四叉树的二值形状空域可扩展编码算法。

4.2.1 CAE 编码和形状信息的特点

CAE 编码的具体内容见论文 3.2 所述。它的优点是采用了上下文模板 (帧内模式时为 9 个像素, 帧间模式时为 10 个像素) 和算术编码, 大大的提高了编码效率, 在减少了码率的同时能够基本保证编码质量。但其存在的缺点是, 算术编码针对的是原始图像的 BAB (MPEG-4 中为了和 MPEG-1/2 相兼容采取的分块编码的编码单位) 块数据, 这些数据存在很大的信息冗余。其次虽然算术编码效率很高, 但算术编码是一种数据编码, 它只是对数据进行编码, 没有利用到二值形状信息的特点, 并且很难实现渐进的扩展性。

从 MPEG-4 的二值形状信息可以看出, 二值形状编码的主要目的就是表达出 VOP 的边界。因为在 VOP 之外的形状信息都为 0 (透明), 在 VOP 之内的形状信息都为 255 (不透明)。BAB 块的属性有三种: (1) 属于 VOP (全不透明); (2) 不属于 VOP (全透明); (3) 部分属于 VOP (边界块)。所以 BAB 块编码的过程实际主要就是对边界 BAB 块的编码。

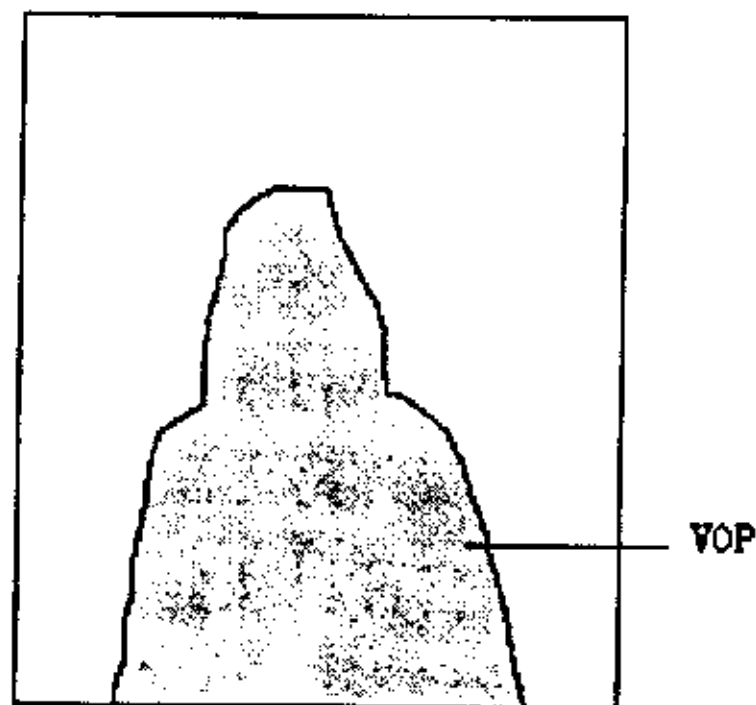


图 4.4 VOP 形状

4.2.2 可扩展编码的思想

通过上一节对可扩展编码的发展简介可以看出,可扩展编码的发展目的和方向是:

- (1) 尽量增强可扩展编码的解码质量。
- (2) 实现解码在质量上的可分层解码,并且向渐进的可分层发展。
- (3) 实现解码端对错误和丢失信息的良好修复能力。

实现目标(2)需要编码上对信息进行渐进的分级编码,同时尽量将比较重要的信息编码在比较低的层次;实现目标(3)一方面需要在增强层采用的参考帧不能对某一帧依赖行太强,另一方面也需要优化码率分配,给重要信息优先分配码流;目标(1)的实现则依赖于所有编解码算法的进一步改进和优化。

4.2.3 四叉树

树型结构是一种非常重要和有用的数据结构。一棵树由一个根结点和它的子树组成,根结点存储了树的简要信息和指向子树的链接信息。树型结构的一个显著特点就是它从根到叶子结点存储量逐渐增大,信息逐渐详尽,所以它可以很好的表达渐进分层的信息。一棵四叉树由一个根结点和四个子四叉树组成,四叉树是图像区域表达的一种重要方式。当图像是方形的,且像素点的个数是2的整数次幂时,用四叉树表达图像的内容非常方便。由于四叉树具有的这些特点,四叉树经常用在静态图像的多分辨率表达中。

4.2.4 基于四叉树的空域可扩展形状编码

算法思路

1) 可扩展编码的实现

MPEG-4 中 BAB 块的大小为 16×16 , 计算机中二值的存储单位一般为 Byte (字节) 或者 Word (字), 每个 Byte 占 8bit, 每个 Word 占 16bit。图像降低分辨率一般采用下采样, 将原图像中一个 2×2 或者 4×4 的块在低分辨率图像中作为一个像素处理。如果能够将 16×16 的 BAB 块的形状信息用 8bit 或者 16bit 的二进制由简到繁的分层表示, 这样不仅方便编码存储又可以很好的实现可扩展编码的思想。但是每个 BAB 块(或其子块)的属性有 3 种(见本节 4.2.1), 需要 1.5 个 bit 表示, 如果直接用 2 个 bit 表示, 在基本层会存在很大冗余, 如果用非整数位表示, 将给编解码和存储带来很多转换的麻烦。仔细分析发现, 虽然每个 BAB 块(或其子块)的属性有 3 种, 但可以分为 2 类, 一类是边界块, 一类是非边界块, 这样就可以将每个块编码 1 个 bit。将 BAB 块属性编码为 1 个 bit 后, 虽

然不能直接解码,但可以通过当前 BAB 块相邻已解码的 BAB 块属性判断出其内部边界的大概形状(如图 4.6)。

结合以上分析和形状信息的特点,对需要编码的高分辨率边界 BAB 块先进行一次 8×8 的下采样,得到 4 个 8×8 的子块。再对这些块中的边界子块进行 4×4 的下采样,得到 4 个 4×4 的 PB(子 BAB 块)块,然后再对这些 PB 块中的边界 PB 块(既不是全透明也不是全不透明的 PB 块)进行编码。将 BAB 块的粗略形状信息进行算术编码放在基本层,将边界 PB 块的详细形状信息进行算术编码放在增强层,这就是本文算法的思路。在编码过程中存在两种编码模式,一种是一定大小的块编码,一种是块内的像素编码。像素编码很简单,可以直接编码和解码。但块编码时,每个块的属性有三种(VOP 内部块,VOP 外部块和边界块),编码信息却只有 1bit,需要在解码端结合相邻块的属性信息和编码信息才可以确定块内子块的形状属性。

2) 渐进的可扩展编码实现

上面的算法思路实现了可扩展的分级编码思想,但分级层数只有两层。基本层和增强层信息量差别还是很大,如果要想实现渐进的多层可扩展编码,还需要采用其它的思想。

首先,增强层的 PB 块形状信息可以分成两层,将 4×4 的块进行 2×2 的下采样得到的粗略形状信息编码在增强层的第一层,再将 2×2 的子块信息编码在增强层的第二层。

其次,不论在那一层的形状信息编码中,都可以通过设置不同得尺寸转化误差阈值来对增强层信息进行进一步细化分层。例如,如果编码程序的整体误差阈值为 8(即对一个 4×4 的 PB 块重构后的像素误差不超过 $8 \times \text{OPAQUE}$, $\text{OPAQUE}=255$),则可以在增强层的第一层可以按误差阈值从大到小(128 到 8)再进行细化分层。同样对增强层的第二层也可以通过这种办法增加精细的渐进程度。

2、算法的具体实现

算法编码部分分为 6 步:(1)重新确定 BAB 边界,(2)确定编码模式,(3)运动估计,(4)确定分辨率,(5)四叉树编码,(6)算术编码。前面四步和 VM 中二值形状编码算法相似,只需对其中的一些地方进行微小的改动以适应后面第 5 步的边界 BAB 编码。这里主要描述后面两步的实现。

1) 四叉树编码

在四叉树编码中有两种编码方式。一种是块编码,如果块中所有像素全为 0 或者全为 255 就编码为 0,否则编码为 1。另一种是像素编码,对像素值为 0 的编码为 0,像素值为 255 的编码为 1。不论块编码还是像素编码,编码扫描方向都是按从左到右、从上到下的方式逐行扫描。

首先对待编码的 BAB 块先进行 8×8 的下采样, 采样结果进行块编码确定一个 4 位的二进制数, 再对 4 个 8×8 子块中编码为 1 的块进行 4×4 的下采样, 同样每个子块进行块编码得到一个 4 位的二进制数, 将这些信息编码在基本层。接着对基本层中 4×4 采样后编码为 1 的块进行 2×2 的下采样块编码作为增强层的第一层。然后再对其中的边界块 (既非全透明也非全不透明的 2×2 块) 进行像素编码, 编码结果作为增强层的第二层。在对增强层编码过程中, 根据需求和整个算法的增强层数目可以设定不同的误差阈值来增加增强层的数目。(图 4.5 为一个边界 BAB 块在增强层只有两层时的四叉树编码过程, 基本层编码结果为 1110 0010 0100 1000, 增强层编码结果为 0010 1000 0011 0111 0001 0011 0111。)注意, 对分辨率为 $1/4$ (满足 4×4 尺寸转化的块) BAB 块增强层没有编码信息, 分辨率为 $1/2$ (满足 2×2 尺寸转化的块) BAB 块增强层没有像素编码信息。

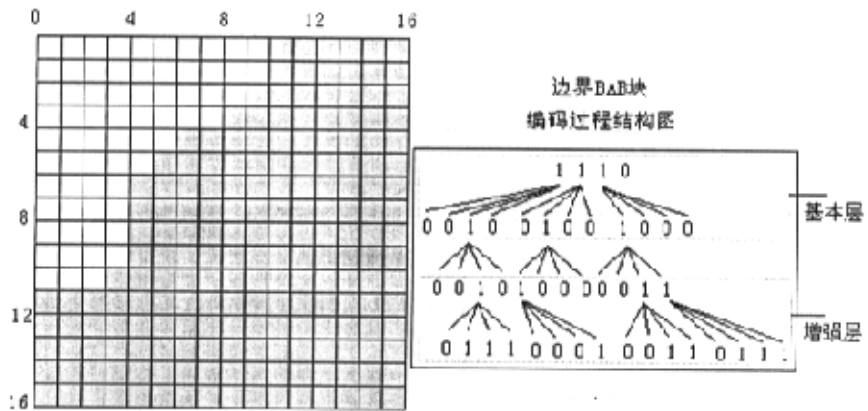


图 4.5 四叉树编码过程

2) 四叉树解码

四叉树解码分为块解码和像素解码两种。像素解码直接进行就行。块解码比较复杂, 因为透明块和不透明块都被编码为 0, 所以解码端要进行透明块和不透明块的判断。如图 4.6, 二值边界块的形状一般有 8 种情况。在程序中可以设定 3 个整形变量 $iLeft$ 、 $iTop$ 、 $iLeftTop$, 分别表示当前块的左边、上边、和左上角 BAB 块的属性。如果这个变量值为 0, 说明它是 VOP 外部块; 如果值为 1, 说明它是 VOP 内部块; 如果值为 2, 说明它是边界块。在解码时如果分辨率为 $1/4$, 则增强层直接对基本层进行上采样就行。如果分辨率为 $1/2$, 则对增强层解码得到 2×2 的块编码信息, 然后进行上采样得到像素的形状信息。

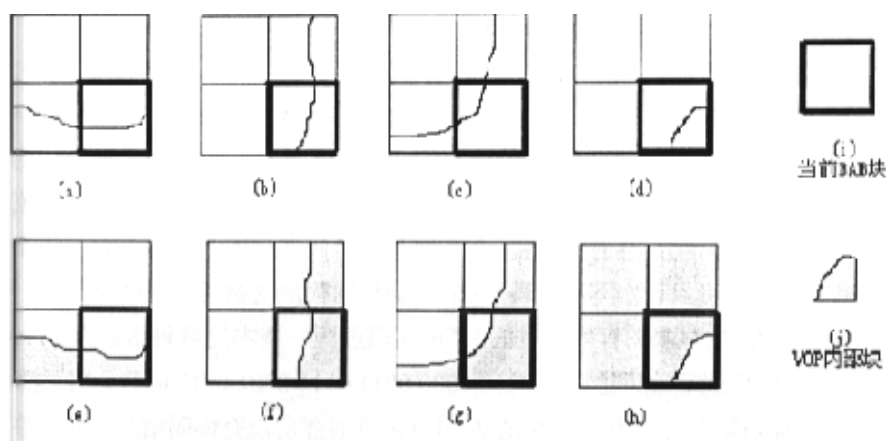


图 4.6 边界块的形状

解码端判断边界 BAB 中的子块是 VOP 内部块还是 VOP 外部块程序代码如下:

```

if(iLeft==2&&iTop==2&& iLeftTop==2)
{
//如果整个 VOP 就在这 4 个相邻块内
scan(RIGHTtoLEFT, VOPexterior); // (b)的特殊情况
//从右到左扫描, 在未遇到边界块之前的 PB 块都为外部块, 之后为内部块;
}
else if (iLeft==2&&iTop==2)
{
if(iLeftTop!=0)
scan(RIGHTtoLEFT,VOPexterior); //不是外部块 (c)
//从右到左扫描, 在未遇到边界块之前的 PB 块都为外部块, 之后为内部块;
else scan(RIGHTtoLEFT,VOPinterior); //(g)
//从右到左扫描, 在未遇到边界块之前的 PB 块都为内部块, 之后为外部块;
}
else if(iLeft!=2&& iTop!=2)
{
if(iLeftTop ==1)
scan(LEFTtoRIGHT,VOPinterior); //(h)
//从左到右扫描, 在未遇到边界块之前的 PB 块都为内部块, 之后为外部块;
else scan(LEFTtoRIGHT, VOPexterior); //(d)
//从左到右扫描, 在未遇到边界块之前的 PB 块都为外部块, 之后为内部块;
}

```

```

}
else if(iLeft==2&& iTop!=2)
{
if(iTop==1)
scan(TOPtoBOTTOM, VOPinterior); //(a)
//从上到下扫描, 在未遇到边界块之前的 PB 块都为内部块, 之后为外部块;
else scan( TOPtoBOTTOM, VOPexterior); //(e)
//从上到下扫描, 在未遇到边界块之前的 PB 块都为外部块, 之后为内部块;
}
else //if(iLeft!=2&& iTop==2)
{
if(iLeft ==1)//(b)
scan(LEFTtoRIGHT, VOPinterior);
//从左到右扫描, 在未遇到边界块之前的 PB 块都为内部块, 之后为外部块;
else scan(LEFTtoRIGHT, VOPexterior); //(f)
//从左到右扫描, 在未遇到边界块之前的 PB 块都为外部块, 之后为内部块;
}

```

其中 scan() 为扫描函数, 第一个参数表示扫描方向, 第二个参数表示在遇到边界块之前的块属性和边界块之后块的属性。

在解码端发生网络阻塞时, 首先可以先将增强层高层中的一部分详细信息扔掉, 如果仍不能满足网络负荷要求, 可以将增强层整个扔掉, 这样将大大减少网络负荷。从而满足接收端能够接收到流畅的视频。

3) 算术编码

对四叉树编码完毕的数据再进行算术编码去掉符号之间的冗余。算术编码需要统计信源概率, 传送概率表。每个 BAB 块传送一个概率表显然不实际, 本文是通过对信源概率进行简单分类来避免传送概率表的。

首先对四叉树编码得到的二进制串前面加“1”(表示这个 BAB 块为边界块), 然后对这个二进制串中的“0”、“1”进行统计, 确定低概率符号(LPS)和高概率符号(MPS)的比例范围($P_{LPS} \leq P_{MPS}$)。如果 $P_{MPS} \Rightarrow P_{LPS} > 2/3 P_{MPS}$, 则输出二进制 0, 然后直接输出四叉树编码结果, 跳过算术编码。如果 $2/3 P_{MPS} \Rightarrow P_{LPS} > 3/7 P_{MPS}$, 则输出二进制 10, 然后按 $P_{LPS}=0.4$ 、 $P_{MPS}=0.6$ 的概率进行算术编码输出。如果 $P_{LPS} \leq 3/7 P_{MPS}$, 则输出二进制 110, 然后按 $P_{LPS}=0.3$ 、 $P_{MPS}=0.7$ 的概率进行算术编码输出。

算术编码的具体步骤和 CAE 编码里的算术编码相似。

3) 算术解码

算术解码时首先根据读入的概率参数 PLPS（见表 4.1）确定解码符号概率，然后对后面二进制串进行算术解码。算术解码完后再进行四叉树解码。

表 4.1 不同的 PLPS 参数对应的解码符号概率

PLPS	符号“1”概率数	符号“0”概率数
0	不进行算术解码	不进行算术解码
10	0.4	0.6
110	0.3	0.7

如果算术解码结果的第一个二进制位为“0”（这一位实际应该为“1”，表示这个 BAB 块为边界块），则在进行四叉树解码前必须对算术解码的结果按位取反。

算术解码的具体步骤和 CAD 的算术解码相似。

4.2.5 实验结果

由于 CAE 算法和本文算法都是在原始图像数据进行一些有损的预处理之后再对这些数据进行无损压缩，所以，本文算法和 CAE 算法解码的图像质量是相同的，只是存在码率大小和可扩展性方面的区别。将本文算法应用于 MPEG-4 标准测试序列 bream.cif（图像大小 352×288 ，共 10 帧），下表统计出了在不同的误差阈值下的 CAE 编码和本文算法的形状编码 bit 数。可以看出，在阈值较小时，四叉树编码的 bit 数较大（特别是增强层），经过再进行算术编码后 bit 数小于 CAE 编码。随着阈值增大，本文算法 bit 数减小很快，这主要是由于增强层数据急剧减少造成四叉树编码的中间结果数据减少造成的。

表 4.2 在不同误差阈值下本文算法和 CAE 算法的编码位数比较

误差阈值 (Alpha_th)	CAE 编码 bit 数	本文算法 bit 数		
		四叉树编码（中间）结果		算术编码（最终） 结果
		基本层 bit 数	总 bit 数	
2	12841	8320	24020	12020
4	10898	7040	18420	9710
8	7915	5984	11412	6506
16	5699	4996	6380	4190
32	4432	4028	4208	3204

4.2.6 结论

由上面的试验结果可以看出, 采用这种分级可扩展编码总的的数据量比采用 CAE 编码有所减少, 特别是基本层数据量非常少。当网络带宽变小时, 可以根据网络容许流量丢弃部分或者全部增强层数据包, 在接收端对收到的基本层和部分增强层数据解码得到原始高分辨率图像的下采样图, 然后进行上采样就可以得到失真不大的 VOP 形状信息。

第五章 结束语

5.1 对 MPEG 标准的发展展望

MPEG 系列标准中的每个标准都是建立在前面标准的基础之上的。MPEG-1 和 MPEG-2 是目前成熟并且广泛应用的标准，他们的主要目的是提高压缩率。MPEG-4 是作为一个开放的多媒体标准出现在市场上而受到瞩目，并为多家厂商所采用。MPEG-4 的目的不仅仅是高效的压缩率，它更加注重媒体内容的交互性，并且包括网络传输的码率可变性和错误易发环境中的错误恢复以及版权问题等广泛的内容。后继标准 MPEG-7 及以后的 MPEG-21 标准重点已经不在压缩方面，MPEG-7 注重于多媒体数据基于内容的描述，用来帮助运营者管理日益丰富的多媒体信息，而 MPEG-21 则更多侧重于与多媒体信息内容的使用相关的诸多方面。可以看出，MPEG 标准的发展不再简单的只考虑音视频的压缩问题，它更加广泛的考虑一个多媒体系统的各个部分及其相关部分的组织、合成和管理等多方面的问题。MPEG 系列标准对多媒体以及相关产业产生了重大的影响，并将极大的推动多媒体通信领域的发展。我们有理由相信 MPEG 系列标准的不断发展必定会极大地促进今后信息产业的更加蓬勃的发展。

5.2 对本文工作的总结

本文研究就是以 MPEG-4 中的形状编码技术为基础，对其中某些部分的算法提出了一些改进算法，并结合可扩展编码技术提出了一种空域形状可扩展编码算法。论文对视频编码的发展作了介绍，对 MPEG-4 的视频编码系统和 MPEG-4 校验模型作了详细的介绍，对 MPEG-4 二值形状形状编码做了深入分析。然后在研究已有运动估计的搜索算法基础上，针对形状编码的运动估计部分，提出了一种快速的运动估计算法；对形状编码的尺寸转化部分提出了一种改进算法；在研究空域可扩展编码特别是 PFGS 编码的基础上，提出了一种新的空域形状可扩展编码算法。通过这段时间对视频编码课题的实践与研究，笔者对视频形状编码方面的问题已经有了比较深刻的理解，在此对将来的研究工作提出自己的一些看法和见解。

形状编码是 MPEG-4 编码实现上的一项关键技术，可扩展编码是 MPEG-4 编码功能上的一项重要技术。将来的研究工作主要集中在形状编码的可扩展性编码方面的研究。

1、尺寸转化和可扩展编码的结合

在尺寸转化中，误差阈值的改变会改变尺寸转化块的数目，这将直接改变后面 CAE 编码（或者采用别的 BAB 块形状编码算法）的待编码块的数据量，并进而改变编码码率和解码质量。所以可以在这一方面继续研究，在算法的细化和深入上更进一步。

2、对本文空域形状可扩展编码的改进

在对形状信息进行四叉树编码后，本文采用了简单的算术编码减少符号间的冗余，这可以提高编码的效率。其实后面的算术编码也可以采用哈夫曼编码来实现。由于论文时间有限，在这一块的研究不是很完善，以后可以进一步研究。不论哈夫曼编码或者算术编码的具体实现都要求知道编码信息的概率，这就需要预先对概率进行统计或者知道先验概率。如果采用算术编码，是采用自适应的算术编码还是基于上下文的算术编码等等问题都需要进一步的分析 and 研究。

在空域可扩展编码的解码端，当接收到的形状信息不能直接得到原始图像的分辨率时（可能由于网络资源紧张，丢弃掉了某些增强层信息），需要根据低分辨率形状信息采取上采样等手段来重构高分辨率的 VOP 形状。这一方面有待进一步深入研究。

附 录:

在尺寸转化过程中,要对下采样进行上采样重构 BAB。在上采样的计算插值结果时,要从阈值表查找阈值进行比较。这个阈值表如下:

Th[256]={5, 6, 6, 8, 6, 7, 7, 8, 6, 5, 7, 6, 8, 8, 8, 8, 6, 5, 5, 9, 8, 8, 8, 8, 7, 6, 8, 8, 8, 8, 8, 9, 6, 7, 5, 8, 7, 9, 8, 8, 7, 8, 8, 8, 8, 8, 8, 9, 7, 8, 8, 8, 8, 10, 8, 9, 8, 10, 8, 11, 8, 11, 9, 10, 6, 7, 7, 8, 7, 8, 8, 8, 7, 8, 8, 8, 9, 8, 10, 9, 7, 8, 8, 8, 8, 7, 8, 9, 8, 10, 8, 9, 8, 11, 9, 10, 7, 8, 8, 10, 8, 10, 10, 11, 8, 8, 10, 9, 10, 11, 11, 10, 8, 8, 8, 11, 8, 9, 11, 12, 8, 9, 11, 12, 11, 12, 10, 11, 6, 7, 7, 9, 7, 6, 8, 8, 5, 8, 8, 8, 6, 8, 10, 9, 8, 8, 8, 10, 8, 10, 8, 9, 8, 10, 10, 11, 10, 11, 9, 10, 8, 8, 6, 10, 8, 8, 10, 9, 8, 8, 10, 9, 8, 9, 9, 10, 9, 8, 10, 11, 8, 13, 9, 10, 10, 11, 11, 14, 11, 12, 10, 11, 7, 8, 8, 8, 8, 8, 8, 9, 8, 8, 8, 9, 8, 9, 11, 10, 9, 8, 8, 9, 8, 11, 9, 10, 10, 11, 9, 10, 11, 12, 10, 11, 8, 8, 8, 11, 10, 11, 11, 10, 8, 11, 9, 12, 11, 12, 14, 11, 10, 9, 11, 12, 11, 12, 12, 11, 9, 12, 12, 13, 12, 13, 11, 14};

致 谢

回顾过去两年半的硕士生活，我得到了许多老师、同学、朋友和亲人的指导、帮助、关心和鼓励，在此向他们表示衷心的感谢。

首先，感谢我的导师姜昱明副教授，他是一位治学严谨而又知识渊博的学者。在跟随姜老师学习的两年多的时间中，姜老师对我严格要求，教给我学习和从事科学研究的许多知识。尤其在论文工作期间，姜老师给我创造了一个良好的学习氛围，指导我阅读了大量的资料，经常与我讨论工作，及时指出我工作中存在的缺点和问题。在这期间，姜老师严谨求实的学者风范，身体力行的工作态度给我留下了永久的记忆。在此论文即将完成之际，谨向姜老师致以深深的敬意和衷心的感谢！

感谢王养利老师对我的指导和帮助；感谢我们教研室的所有同学：洪功义、何凯、王乃霞、张建荣、黄铁英和师姐李蕊等对我的帮助；感谢我们 502 宿舍所有舍友给予我的支持，与他们的朝夕相处和讨论使我增长了许多知识。

谨以此文献给我的家人，感谢他们多年来对我生活上的关心和精神上的支持，没有他们无私的爱，就没有我的大学和硕士生涯！

感谢我的女朋友在我完成论文过程中给我的鼓励和帮助。

感谢流媒体世界网站 (<http://liumeiti.cn/>) 论坛的网友们的帮助，与他们讨论和互相学习使我对 MPEG-4 标准的认识更加深入。

参考文献:

- [1] Coding of Moving Pictures and Audio: Short MPEG-1 description,ISO/IEC JTC1/SC29/WG11 N,MPEG 96/June 1996.
- [2] Coding of Moving Pictures and Audio: Short MPEG-2 description,ISO/IEC JTC1/SC29/WG11 N,MPEG 00/October 2000.
- [3] Coding of Moving Pictures and Audio: Overview of the MPEG-7 Standard (version 6.0),ISO/IEC JTC1/SC29/WG11 N4509,Pattaya, December 2001.
- [4] Coding of Moving Pictures and Audio: MPEG-21 Overview v.4,ISO/IEC JTC1/SC29/WG11/N4801,Fairfax,May 2002.
- [5] Coding of moving video:Video coding for low bit rate communication,ITU-T Recommendation H.263,02/98.
- [6] <http://www.avs.org.cn>, 数字音视频编解码技术标准工作组官方网站。
- [7] Coding of Moving Pictures and Audio: MPEG-4 Overview - (V.21 – Jeju Version), ISO/IEC JTC1/SC29/WG11 N4668,March 2002.
- [8] ISO/IEC 14496-2,Coding Of Audio-Visual Objects: Visual, ISO/IEC JTC1/SC29/WG11 N2502, November 1998.
- [9] ISO/IEC 13818-1,Generic Coding Of Moving Pictures and Associated Audio: Systems, 1994.
- [10] ISO/IEC 14496-5,Coding Of Audio-Visual Objects: Reference Software, Draft International Standard, ISO/IEC JTC1/SC29/WG11 N2505, November 1998.
- [11] ISO/IEC 14496-6,Coding Of Audio-Visual Objects: Delivery Multimedia Integration Framework, ISO/IEC JTC1/SC29/WG11 N2506, November 1998.
- [12] The Delivery Layer in MPEG-4,G. Franceschini - CSELT Centro Studi e Laboratori,Telecomunicazioni S.p.A.
- [13] ITU-T Rec. H.264 / ISO/IEC 14496-10, “Advanced Video Coding”, Final Committee Draft, Document JVT-E022, September 2002.
- [14] Coding of Moving Pictures and Audio: MPEG -4 FAQs,ISO/IEC JTC1/ SC29/ WG11 N,MPEG 97/,July 1997.
- [15] Coding Of Audio-Visual Objects: MPEG-4 Video Verification Model version 18.0,ISO/IEC JTC1/SC29/WG11 N3908,January 2001/Pisa.
- [16] Touradj Ebrahimi and Caspar Horne, MPEG-4 Natural Video Coding - An overview, <http://www.m4if.org/>.

- [17] 钟玉琢, 王琪, 贺玉文, 《基于对象的多媒体数据压缩编码国际标准—MPEG-4 及其校验模型》, 科学出版社 2000, 438—451 页。
- [18] Jagan Thinakaran, Duan-Juat Ho, and Nam Ling, An Architecture for MPEG-4 Binary Shape Decoder, ISCAS 2000-IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland, pp: II-457- II-460.
- [19] Noboru Yamaguchi, Takashi Ida and Toshiaki Watanabe, A Binary Shape Coding Method Using Modified MMR, IEEE 1997, PP: 504-507.
- [20] Shi Hwa Lee, Dae-Sung Cho, Yu-Shin Cho etc, Binary Shape Coding Using 1-D Distance Values from Baseline, IEEE 1997, pp: 508-511.
- [21] F. W. Meier. An efficient shape coding scheme using B-splines which is optimal in the rate-distortion sense, master's thesis, Northwestern University, Evanston, IL, June 1997.
- [22] Shi Hwa Lee, Dae-Sung Cho, Yu-Shin Cho etc, Binary Shape Coding Using 1-D Distance Values from Baseline, Special session on shape coding, ICIP97, pp: 508-511.
- [23] K.J.O'Connell, Object-adaptive vertex-based shape coding method, IEEE Trans. on Circuits and Systems for Video Technology, Vol.7, No.1, 1997, pp: 251-255.
- [24] Paulo Nunes, Fernando Pereira, Ferran Marques, Multi-Grid Chain Coding of Binary Shapes, IEEE 1997, pp: 114-117.
- [25] CLe Buhan Jordan and T Ebrahimi, Progressive Polygon Encoding of Shape Contours, IPA97, pp: 15-17.
- [26] Jorn Ostermann, Euee S.Jang, Jae-Seob Shin, Tsuhan Chen, Coding of arbitrarily shaped video objects in MPEG-4, IEEE 1997, pp: 496-499.
- [27] 徐凌云, MPEG-4 形状编码技术, 《通信技术》2001 年第 2 期, 37—41 页。
- [28] 王正勇, 李永合, MPEG-4 基于内容的视频编码研究, 《四川大学学报》(自然科学版) 2001 年 10 月, 675—678 页。
- [29] Eric Bodden, Lossless data compression using Arithmetic Coding, Abstract for the IAF congress 2002, Houston.
- [30] 精英科技 编著, 《视频压缩与音频编码技术》, 中国电力出版社, 2001 年 6 月, 85—91 页。
- [31] 林福宗, 《多媒体技术基础》, 清华大学出版社, 2000 年 8 月, 47—51 页。
- [32] Arturo San Emeterio Campos, Arithmetic coding, <http://www.arturocampos.com/>.
- [33] G.G.Langdon Jr. and J.Rissanen, Compression of black-white images with arithmetic coding, IEEE Transactions on Communications, COM-29(6), June 1981, pp: 858-867.

- [34] A.Murat Tekalp,《数字视频处理》,清华大学出版社,1998年5月,101—106页。
- [35] Donghoon Yu,Sung Kyu Jang,and Jong Beom Ra ,A fast motion estimation algorithm for MPEG-4 shape coding , IEEE Transaction on Circuits and Systems for Video Technology,vol 13,No.4,April 2003,pp:358—363.
- [36] R.Li, B.Zeng, and M.L.Liou,A new three-step search algorithm for block motion,IEEE Transactions on Circuits and Systems for Video Technology, vol.4,No.4,August 1994,pp:438-442.
- [37] M.Ghanbari, The cross-search algorithm for motion estimation, IEEE Transactions on Communications, Vol.38, No.7, July 1990,pp: 950-953.
- [38] Lai-Man Po, Wing-Chung Ma, A Novel Four-Step Search Algorithm for Fast Block Motion Estimation, IEEE Trans. Circuits Syst. Video Technol., Vol.6, No.3, Jun.1996, pp: 313-317.
- [39] Shan Zhu and Kai-Kuang Ma, A New Diamond Search Algorithm for Fast Block Matching Motion Estimation, International Conference on Information, Communications and Signal Processing, ICIDS'97 Singapore, 9-12 September 1997.
- [40] Fang-Hsuan Cheng and San-Nan Sun, New Fast and Efficient Two-Step Search Algorithm for Block Motion Estimation, IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No.7, October 1999,pp: 977-983.
- [41] 朱红彬, 薛方, 基于对象快速自适应的混合搜索运动估计方法, 电视技术, 2002年第2期, 12—14页。
- [42] 洪波, 庄健敏, 余松煜, 基于时空相关性的自适应运动估计方法, 计算机工程, 第26卷第8期, 2000年8月, 50—57页。
- [43] 肖德贵, 余胜生, 周敬礼, 快速而有效的块运动估计算法, Journal of Computer Research & Development , 第38卷第6期 ,679—683页。
- [44] 廖明, 冯穗力, 叶梧, 两种实时、低码率视频编码的运动估计搜索算法, 电视技术, 2003年第3期, 13—15页。
- [45] 初秀琴, 李玉山, 徐善锋, 一种快速分类搜索运动估计新方法, 中国图象图形学报, 第7卷(A版)第10期, 2002年10月, 1067—1070页。
- [46] 唐泽鹏, 秦雷, 朱秀昌, 刘峰, 运动估计算法分析 ,电视技术, 2001年第2期。
- [47] 刘李杰, 蔡德钧, 带宽限制条件下的二值形状速率控制, 通讯学报 Vol.22 No .6,2001年6月, 1-7页。

-
- [48] BRADY N. MPEG-4 standardized methods for the compression of arbitrarily shaped video object , IEEE Trans on Circuits Syst Video Technol, 1999.9(8), pp:1170-1189.
- [49] Mei-Juan Chen, Yuan-Pin Hsieh and Yu-Ping Wang, Multi-Resolution Shape Coding Algorithm for MPEG-4, IEEE 2000, pp:126-127.
- [50] Weiping Li, Overview of Fine Granularity Scalability in MPEG-4 Video Standard, IEEE Transactions on Circuits and Systems for Video Technology, VOL.11, NO.3, March 2001, pp:301-317.
- [51] 郑湘琴, 林正豹, MPEG-4 分层编码, 北京广播学院学报(自然科学版), 第 9 卷, 第 2 期, 2002 年 6 月, 62—67 页。
- [52] 欧阳书平, 浅谈 MPEG-4 标准中的 FGS 技术, 电视技术, 2002 年第 9 期, 27—28 页。
- [53] 吴枫, 李世鹏, 张亚勤, 渐进、精细的可伸缩性视频编码, 计算机学报, 2002 年第 12 期, 1276—1282 页。

作者在读研期间的研究成果

- [1] 李高林, 姜昱明, MPEG-4 形状编码中尺寸转化的研究, 被《计算机工程》录用, 拟于 2004 年 9 月刊出;
- [2] 李高林, 姜昱明, 在基于 JSP 的电子商务应用中利用 UML 建模与开发技术, 被《计算机应用研究》录用, 拟于 2004 年 4 月刊出;
- [3] 李高林, 姜昱明, MPEG-4 中形状编码的研究, 被《计算机应用研究》录用, 拟于 2004 年 7 月刊出;
- [4] 李高林, 姜昱明, 一种改进的快速 MPEG-4 形状编码的运动估计算法, 被西安电子科技大学第二届研究生年会录用, 收入大会论文集。