

## 摘要

近年来,随着计算机网络和通信等技术的发展,多媒体得到了越来越多的重视和应用。视频压缩技术是现代多媒体编码技术中最重要的研究领域。本文开展了这方面的初步研究,重点集中在视频压缩中相当关键的运动估计算法上。

数据压缩原理在于原始数据中存在着各种冗余。多媒体中的视频编码技术主要是通过消除帧内的空间冗余和帧间的时间冗余来实现数据的压缩。在消除时间冗余方面,帧间块匹配技术是比较完善的,其原理是用前一帧中的一个图像块来最大地近似当前编码块。如何尽快地找到这个最佳块便是快速搜索算法所要完成的工作。

针对视频编码系统运动估计的几种常用编码方法,包括三步法、正交共轭法、象限法等,我们分析了它们的优缺点。我们对正交共轭法进行了改进,提高了原有算法最佳匹配点的命中率。

上述快速算法因为搜索区域有空白区而可能在搜索时会遗漏全局极值点而陷入局部极值区域。本文又提出了预测起始搜索点的运动估计算法,对搜索的起始位置进行修订来尽可能地接近全局最优点,同时也在一定程度上将多极值问题变成单极值问题。另外该算法还对门限值的选取进行了改进,进一步加快了搜索匹配的速度。实验表明,该算法能在大幅度减少搜索步数的同时还能保持较高的信噪比。与传统的三步法和正交共轭法相比,改进算法由于搜索路径固定且前后点无关,因此适合用硬件实现多线程的搜索。

**关键词:** 视频编码, 运动估计, 快速搜索算法, 多极值, 起始搜索点

## Abstract

With the development of technology of computer network and communication, multimedia have been obtaining more and more emphasis and application. The technology of video encoding is the most important research domain in the multimedia. According to this situation, we did some research and focused on the video motion estimation in the video coding system

The theory of data compression is depended on the redundancies in the original data. The video coding technique of the multimedia emphasize particularly on the losing compression. Its compression is realized by eliminating the spatial and temporal redundancies. Block-based motion estimation technique is the fairly mature one. It uses an image block in the previous frame to replace the current block and tries to find the most similar one. How to find the best one as fast and good as possible is what the fast searching algorithms do.

After discussing the most used fast searching algorithms of the motion estimation which includes Three Step Search, Conjugate Direction Search, Quadrant Search, we then evaluated them. And we bring up an improved CDS algorithm, it is better than the old one in finding the best matching point.

These algorithms might loss the best matching point because their search areas have blank parts. We present an estimation algorithm that predicts the best beginning search point to approach the best matching point as close as possible. It can somewhat translate the multi- extremum problem into single extremum problem. And we also bring up a threshold value picking-up algorithm to accelerate the searching speed. The tests showed that these algorithms can reduce the searching steps while guaranteeing the fairly high PSNR value. Compared with the traditional algorithms, this algorithm's search path is fixed and the searching points are independent to each other, so these make it easier to be used in multi-thread searching.

**Keyword:** Video Coding, Motion Estimation, Fast Searching Algorithm, Multi-Extremum, Initial Searching Point

## 1 绪 论

### 1.1 引言

世界文明的发展始于人类相互之间能够进行复杂细致的交流之时，因为交流才能使个体凝聚成群体也才能够产生高于个体的智慧。纵观人类文明的发展，几个高峰期的开始总是伴随着更先进的通讯方式的诞生。语言，文字，印刷术，电视以及如今短短几年就征服了全球的国际互联网无不印证了这个规律。如今，人类已经踏入信息时代的快速增长期，代表这一时期的最先进的通讯方式是数字化通信、计算机网络以及多媒体技术。电子邮件、文件共享、远程办公等技术在大大加快了人们工作的效率的同时还降低了通讯交流的成本；BBS、即时通讯的QQ和各种网站论坛极大地丰富了人们的娱乐生活。但是集语音、数据、图像和文字为一体的交互式多媒体通信方式却始终未能浮出水面。现在，人们的需求已经不仅仅停留在每天回家后在电视机这种专一设备上欣赏由电视台统一传送的广播式节目、在互联网上同网友只能通过文字沟通以及在电话中仅能听到亲友的声音这么简单的通讯方式。通讯发展的趋势要求符合人类信息交换方式的多媒体通信技术能够尽快出台。

网络上的 flash 动画受到众多网民的喜爱并已成为一种标准。flash 的特色就是集矢量动画、音乐、语音和交互于一身，它的迅速发展体现了如今的通讯领域对交互媒体的渴望也反映了对如今互联网狭窄的带宽和多媒体庞大的数据量之间的矛盾的一种无奈。在实际效果上，flash 无法取代真实的视频。但要采用一般视频而视频数据不进行压缩的话，数据量将会变得很大，对于活动视频图像序列更是如此<sup>[1]</sup>，如表 1.1 所示。

表 1.1 常见的视频信号

类型	像素/帧	帧/秒	比特/像素	未压缩时比特率
NTSC	480×483	29.97	16	111.2Mb/s
PAL	576×576	25	16	132.7Mb/s
CIF	352×288	14.98	12	18.2Mb/s
QCIF	176×144	9.99	12	3.0Mb/s

可见, 视频的信息数据量非常巨大, 以 CIF (Common Intermediate Format) 格式为例, 假定在 56k Modem 这种常用的上网设备上以 56kbps 的信道传输这样的彩色视频信号, 则视频数据的压缩比需高达 1:325。即使采用 QCIF (Quarter Common Intermediate Format) 格式, 压缩比仍需有 1:81。达到这样高的压缩比且要保持较好的视频质量则必须要有一种高效的视频压缩技术。尽管通信网络和存储媒体也正在向高带宽、大容量方向发展, 如 ADSL 和光纤通讯等技术, 但是随着社会信息交互的日益广泛和新型的多媒体业务不断出现, 个人通信量会成指数增长, 信道的容量永远不会绰绰有余。因此对视频信息进行压缩处理, 使它能在较低速率的信道上实时传输是人们多年来孜孜以求的目标。

## 1.2 数据压缩原理

C. E. Shannon 在创立信息论时, 提出把数据看作是信息和冗余的组合。这些冗余数据对于信息的表达从信息的完备性角度来看是没有意义的, 只要能够把能表达全部意义的部分保存下来即可保留所有有效信息。所以, 去除数据冗余的部分就成为多媒体数据压缩的目标。

目前各种数据压缩技术都是与数据中数据冗余类型有关。压缩技术分为无损压缩和有损压缩两类。前者是严格遵循去数据相关性这条规则的, 换言之即在压缩的过程中将有效信息一视同仁。而有损压缩则遵循按用户要求在单位数据量一定的情况下保留尽可能多的有用信息。由此可见, 有损压缩要比无损压缩复杂灵活的多, 压缩比和单位有意义信息量也高得多。多媒体数据本身正好具有数据信息是非均匀的, 被人类感官察觉的灵敏度不同等因素, 因此特别适合于有损压缩。如今流行的多媒体压缩格式有视频方面的 MPEG 系列, 流媒体有 RM, 音频有 RA、MP3, 图像有 JPG 等。这些重要的压缩格式无一例外的都是有损压缩。

视频有损压缩有两方面问题要解决: 一是如何判定信息各部分的重要性, 二是如何在一定的信息量下保留最多意义的信息。

一般多媒体数据中存在着以下几种数据冗余<sup>[2-4]</sup>:

### (1) 空间冗余

用来表示图像的数据中存在着相当大的空间相关性。例如大面积的均匀背景中就有很大的空间冗余性, 因为所有这些点都可以用几种颜色来表示。原始图像的逐点描述明显不是最佳存储方案。

## (2) 时间冗余

这是序列图像和语音数据中因连续变化而包含的冗余。自然界的视频基本上平滑变化的，这也就意味着图像序列中前一幅图像和后一幅图像之间变化缓慢，变化差异只占两者的一小部分，故存在有较大的相关性。

## (3) 信息熵冗余(编码冗余)

信息熵冗余是一组数据所携带的信息量编码冗余。信息熵冗余是在信息在信源的符号表示过程中，由于未遵循信息论意义下最优编码而造成的冗余性，这种冗余性可以通过熵编码来进行压缩。

## (4) 知识冗余

许多图像的理解与某些基础知识有相当大的相关性，这类规律性的结构可由先验知识和背景知识得到，我们称之为知识冗余。

## (6) 视觉冗余。

人类的视觉系统对于图像场是非均匀和非线性的，特别是视觉系统并不是对图像场的任何变化都能察觉。图像在编码和解码处理时尽管由于压缩和量化截断引入了噪声而使图像发生了一定变化，但这些变化如果是不能被视觉所察觉得话，我们仍认为图像是完好的或是足够好的。像这样的冗余，我们称之为视觉冗余。

## 1.3 视频编码技术概述

对图像压缩编码的研究始于 50 年代，迄今已有五、六十年的历史。随着计算机、多媒体、网络、通信等技术的发展，图像编码也得到了相应的发展，并且它的地位越来越重要。

一般人们把视频编码技术分成两个阶段，把 1948 年至 1988 年间主要研究的图像编码方法称为“经典图像编码技术”，而后的称为“第二代编码技术”。

经典图像编码技术是只利用基于波形的图像编码方法进行图像压缩，它是以像素和像素块作为编码的基本对象进行操作，在设计编码器的时候没有或较少的考虑了人眼的视觉系统(HVS)。最初图像压缩编码是对帧内预测法和亚取样内插复原法进行研究。80 年代初开始对运动补偿(MC)所用的运动估计(ME)进行研究。变换编码是 1968 年 H. C. Andrews 等人提出的，采用的是二维离散傅里叶变换。此后相继出现了其他方式的变换编码，包括二维 DCT、小波等。

到了八十年代末，经典图像编码技术的发展开始达到饱和，在极低比特率视频编码(VLBV)应用中表现出了极大的局限性。图像的最终接受器是人眼的视



觉系统,因此图像质量的好坏要由人的主观感觉来判定。第二代编码技术是相对于经典编码技术而言更高效的技术,它充分考虑了人眼的视觉特性,并将人眼视觉系统作为编解码器中的基本组成部分。克服了第一代编码技术的不足,它最早是由 Kunt<sup>[5]</sup>在 1985 年提出的。如果把图像和视频编码分成两个步骤,第一步为把图像数据转化成某个消息序列,第二步为对这个消息序列分配码字,我们可以很容易地区分经典编码技术和第二代编码技术:经典编码技术着重考虑第二步,而第二代编码方法重点考虑的是第一步并把其结果用于第二步。第二代方法把图像看成是由不同对象组成的,这意味着图像序列必须先经过分解或分割来找到待处理的对象。第二代视频编码技术主要有基于分割的方法、模型基方法和基于分形的方法。

## 1.3.1 经典编码技术

经典编码技术主要有预测编码、脉冲编码调制、变换编码、矢量量化<sup>[6-7]</sup>、子带编码<sup>[8]</sup>和小波编码<sup>[9]</sup>等。

预测编码是利用信号间的相关性将每个像素用其前值(前像素、前行或前一帧)做预测,并传输实际信号与预测信号差值。实际信号与预测信号的差值越小,对其编码后的比特数可相应减少,这就是预测编码实现压缩的机制。

变换编码是 60 年代末提出并于 70 年代发展起来的,它将图像分割成若干子块,通过对其正交变换把时间域信号变换到频域上使变换后各系数的统计分布比较集中,能量集中在低频域,并使变换域中各系数间的相关性大大降低,然后对其分配比特。对于方差大的分配以较多的比特,反之,分配较少的比特数进而实现总比特率的压缩。

矢量量化是用矢量量化器代替标量量化器做量化,从而降低码率的方法。

子带编码先将原图用若干数字滤波器分解成不同频率成分的分量,再对这些分量进行亚抽样,形成子带图像,最后对不同的子带图像分别用与其相匹配的方法进行编码。在接受端,将解码后的子带图像补零、放大,并经合成滤波器的内插,将各子带信号进行相加,进行图像恢复。

经典图像和视频编码技术通过减少图像数据的相关性来压缩数据量。它将图像看成是需要进行解相关的像素组成的集合,采用了波形编码。利用率失真定理,在给定编码速率的条件下得到最小的波形失真,或者在达到最小编码速率的条件下得到可接受的波形失真。

## 1.3.2 第二代编码技术

第二代图像和视频编码技术在设计图像和视频编码器时更多地考察人眼的主观视觉特性。常见的第二代编码技术主要有：基于分割的方法、模型基方法和基于分形的方法等。

基于分割的图像和视频编码是把图像和视频分割成若干个对象，对每个对象进行编码。例如基于内容的视频编码<sup>[10]</sup>、轮廓/纹理(Texture)<sup>[11]</sup>编码、分裂(split)和合并(merge)编码<sup>[12]</sup>等。早期的轮廓/纹理编码是根据灰度等信息把图像分成同态(homogeneous)区域，每个区域作为对象进行编码，相比较基于内容的编码而言，这种划分是一种低层次的划分。近几年发展起来的基于内容的 MPEG-4 编码对编码对象的划分是一种有意义的划分（如，人物、地球仪、桌子等），而不是简单的图像区域划分。

模型基编码<sup>[13-16]</sup>(Model-based Coding)把图像看作 3D 真实物体的 2D 投影，充分利用了图像中目标和景物的结构化特征和知识。首先编码器利用编解码端共有的图像模型，通过图像分析提取景物参数，例如形状参数、运动参数等，然后解码器利用编码器传来的参数借助图像合成技术重建图像。此过程如图 1.1 所示。模型基编码在更深层次上研究了图像的冗余性，可以达到更高的编码效率。研究资料表明，动画效果的面部仅需要 500-1000bps，更逼真的面部表示可能需要 5-10kbps，CIF 和 QCIF 彩色头肩序列可达到 16-64kbps。模型基编码的三个关键技术是模型匹配（利用 2D 图像序列重构 3D 或 2D 模型）、图像分析（提取模型参数和运动参数）和图像合成。

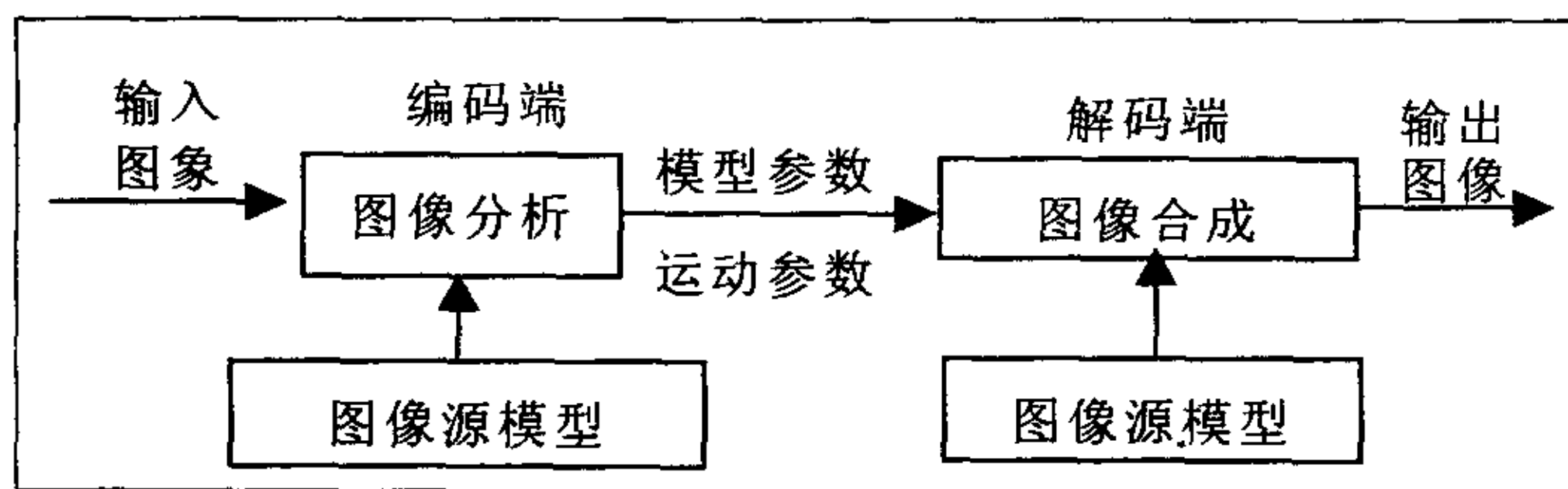


图 1.1 模型基编码系统原理框图

分形编码(fractal-based coding)<sup>[17-19]</sup>起步较晚，但其压缩比的潜力令人瞩目。分形图像编码主要利用自相似性的特点，通过迭代函数系统(IFS: iterated function

system) 来实现压缩。自相似性是指无论几何尺度怎么变化,景物任何一小部分的形状都与较大部分的形状极其相似。自相似有些是确定性的,但大量存在的是统计自相似及随机自相似性。分形编码正是利用图像了局部与局部、局部与整体间的自相似性冗余来进行压缩,不仅消除了图像中的近程相关性,而且有效的消除了自然图像中的远程相关性,因而能够获得相当高的压缩比,特别是在高压压缩比的情况下,分形编码的恢复图像质量要优于其他编码方法如 JPEG。目前常见的自动分形图像压缩编码方法都是以 Jacquin 的方法为基础,这类方法用仿射变换 (affine transformation) 来消除图像块中间的确定自相似性冗余。在编码时将图像分为互不重叠的子块,通常称这些子块为值域块,对每一值域块编码时都要用搜索的方法在图像中寻找与其相似的定域块,并按照某种误差准则确定一个相应的收缩仿射变换,使得编码误差满足给定要求。目前分形编码最主要的问题在于计算耗时太长,不适合视频编码的实时性要求,仍然有待进一步的深入研究。

## 1.4 论文的选题和主要工作

本文的选题主要是针对视频压缩中的运动估计算法的研究。视频压缩中决定其压缩后的数据码流大小的关键因素在于能否较好的去除其时间相关性。传统方法已经能很好的进行图像匹配,但是是以牺牲速度来获得最好的预测效果的。虽然现在的计算机运算能力已经很强大,但是相对于视频文件的数据量和一般个人使用的计算机系统而言仍远远不够。在较流行的视频压缩软件的使用中,经常会遇到压缩一段普通电影长度的视频需要一天甚至好几天的情况。这对于普通应用来说是不可忍受的。本文就是本人对视频编码技术中的运动估计的快速算法的研究和其算法实现的总结。

本论文共分五章,主要结构与内容如下:

第一章为综述,提出问题,介绍了视频压缩编码技术的发展与现状。

第二章对基于对象的 MPEG-4 视频编码系统进行描述,对 MPEG-4 中的形状编码、纹理编码、可伸缩性编码和码率控制技术进行的详细的探讨。

第三章着重讨论了 MPEG-4 视频编码系统运动估计的几种编码方法,并对常用的快速算法如三步法、共轭方向法、象限法等,详细分析了它们的优缺点,并对正交共轭法进行了改进,提供了相应的实验数据。

第四章根据上述算法编码因为搜索区域存在多极值而可能导致漏选最佳匹配点这一缺陷,提出了预测起始搜索点的预测算法。另外还对门限值的选取进行了



分析,进一步加快了搜索匹配的速度。实验表明,该算法与传统的运动估计算法相比在大幅度减少搜索步数的同时还能保持较高的信噪比,且算法简单能直接嵌入到大多数编码系统。

最后一章对全文工作进行了总结,并针对下一步的研究工作做出了初步设想。

## 2 基于对象的 MPEG-4 视频编码技术分析

本章首先对现有的视频编码标准进行简要介绍, 然后对 MPEG-4 视频编码系统的框架结构进行描述, 最后对 MPEG-4 中的形状编码、纹理编码、运动估计和补偿、可伸缩性编码和码率控制技术等进行详细的探讨。

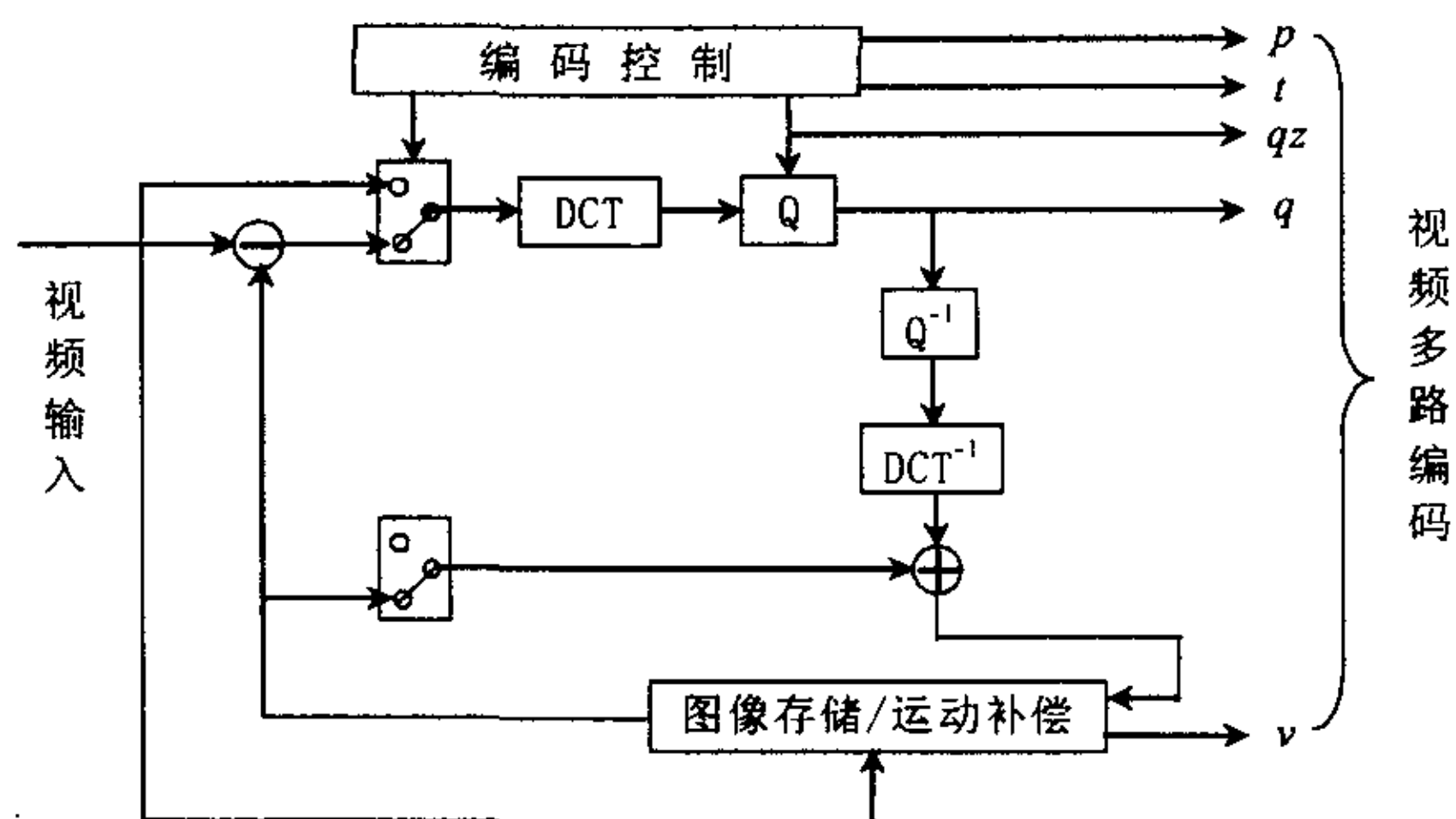
### 2.1 现有视频编码技术国际标准

视频编码的研究成果集中体现在相应的标准中, 目前已经制定出的比较重要的视频编码标准有: ITU-T/H.261/H.263/H.263+, ISO/IEC/MPEG-1/2/4 系列。下面将分别介绍。

#### 2.1.1 ITU-T H. 261、H. 263、H. 263+

于 1988 年形成草案、1990 年 12 月完成和通过的 CCITT 建议书 H.261, 是视频编码技术走向实用化的里程碑, 被誉为“图像编码 40 年经验之总结”。H.261 支持实时动态图像的压缩编解码。它采用的主要视频编码技术有 DCT 编码、帧间预测、运动补偿等混合编码技术。

在 1995 和 1996 年, ITU-T 制定了用于低比特流通信的 H.263<sup>[20-22]</sup> 视频编码标准。视频信源编码器原理框图如图 2.1 所示, 主要由运动补偿预测、离散余弦变换、量化、熵编码和编码控制几部分组成。



注: Q-量化器  $Q^{-1}$ -逆量化器 p-帧内 / 帧间标识

t-传输与否标识 qz-量化级 q-量化后的变换系数 v-运动矢量

图 2.1 H.263 视频信源编码器

最初的 H.263 草案主要是为了支持 64kbps 以下视频数据流编码,但不久这个限制又被取消了,使得 H.263 可以支持大范围变化的视频比特流编码,从而可以完全取代 H.261。H.263 使用的算法同 H.261 以及 MPEG-2 相似,都采用了 DCT 加运动补偿(MC-DCT)的编码方法,但 H.263 做出了以下改进:

- (1) 运动补偿采用了半像素精度并使用了环形滤波器;
- (2) 数据流的一些层次结构内容是可选的,使编码器的参数配置达到最优,以取得更低的码流,并能进行有效的差错掩盖和恢复;
- (3) 增加了四种可选择的编码模式以提高执行效率:自由运动向量模式,基于语法的算术编码模式,高级预测编码, PB-帧模式;
- (4) H.263 支持五种分辨率的图像格式: QCIF ( $176 \times 144$ ), CIF ( $352 \times 288$ ), sub-QCIF ( $128 \times 96$ ), 4CIF ( $704 \times 576$ ), 16CIF ( $1408 \times 1152$ )。

H.263+是 H.263 的扩展,它的整体结构和基本算法与 H.263 相同。同时在 H.263 的基础上做了一些新的改进。比如, H.263 只支持 5 种图像格式,而 H.263+支持更多种的自定义图像格式,并且图像尺寸、宽高比和时钟频率等都可以作为 H.263+数据流的一部分。与 H.263 相比,它增加了 12 个可选的编码模式:先进的帧内编码、除块滤波、片(slice)结构、附加增强信息、改进的 PB 帧、参考图像选择、时空及 SNR 可扩展性(scalability)、参考图像再抽样、减少分辨率更新、独立分段解码、可替换的帧内 VLC、修改的量化模式。H.263+中的无限制运动矢量模式与 H.263 中的无限制运动矢量模式也有所不同。

## 2.1.2 ISO/IEC/MPEG-1、MPEG-2、MPEG-4

成立于 1988 的活动图像专家组 MPEG (Moving Pictures Experts Group) [23-25]是国际标准化组织/国际电工委员会(ISO/IEC)下属的负责制定开发运动图像和音频的压缩、解压缩、处理和编码的国际标准的一个工作组。该专家组先后制定了 MPEG-1、MPEG-2 和 MPEG-4,并正在致力于制定 MPEG-7 和 MPEG-21。

MPEG-1 制定于 1993 年 8 月,是针对 1.5Mbit/s 数据传输率下的数字存储媒质运动图像及其伴音编码的国际标准。它的视频部分与 H.261 及 H.263 原理图大致相同。然而, MPEG-1 主要是针对存储媒体的视频编码标准,而 H.261 和 H.263 主要是针对传输的视频编码标准。故 MPEG-1 应当在顾及图像质量和压缩比的情

况下, 要考虑对序列图像进行随机访问和编辑的方便。MPEG-1 加入了双向预测帧—B 帧, 从而加大了压缩比。对随机访问及编辑的支持由每秒 2 个 I 帧来支持, 它不需依赖 P 帧和 B 帧而独立解码。因而利于实现 MPEG 比特流的编辑与随机访问。MPEG-1 对色差分量采用 4:1:1 的二次采样率。其视频压缩率为 26:1。

1995 年制定的 MPEG-2 主要是针对数字视频广播 (DVB)、高清晰度电视 (HDTV)、数字视盘 (DVD) 等制定的高于 1.5Mbit/s 运动图像及其伴音的编码标准<sup>[26]</sup>。MPEG-2 综合采用了利用运动补偿的帧间预测、空间域离散余弦变换、自适应量化和可变字长编码的混合算法, 该标准为了适应不同码率和不同场合, 仅对比特流语法和相应的解码方法做了规定, 而在编码器的设计上留有很大自由度, 如运动估计的搜索算法和匹配函数的选择, 利用人视觉特性的自适应量化等。

1994 年, MPEG 活动图像专家组开始制定 MPEG-4<sup>[27-33]</sup> 标准。它的目的是寻求支持数字音频/视频数据通信、存取和管理的新途径, 提供一个可以实现交互性和高压缩比, 以及具有很高灵活性和可延展性的新的视频音频编码标准。较之前几个 MPEG 标准, MPEG-4 为多媒体数据压缩提供了一个更为广阔的平台, 它更多定义的是一种格式和框架, 而不是具体的算法, 以支持各种各样新颖的和高效的功能。

MPEG-4 最显著的特点是基于内容的编码, 即场景由几个音/视频对象 (Audio Video Object) 组成, 如背景图片或音乐、处在焦点区的视音频内容、文本和图形等, 对场景的编解码建立在这些音/视频对象上, 这样用户可以对这些对象分别进行解码和重建, 使得使用者对原始场景能够以一种灵活的方式来操作。

基于对象的分级功能是 MPEG-4 提供的又一个新的功能, 同时兼容于 MPEG-2 标准中的图像分级功能, 分级工具主要用于互联网和无线网等窄带的视频通信、多质量视频服务和多媒体数据库预览等服务。MPEG-4 提供了两种基本的分级工具: 时域分级和空域分级, 时域分级是降低原视频序列的帧率, 空域分级是降低原视频序列的分辨率。在每类分级工具中, 视频序列都可以分为基层和增强层两层, 基层提供了视频序列的基本信息, 增强层提供了视频序列更高的分辨率和细节。基层可以单独传输和解码, 而增强层则必须与基层一起传输和解码。MPEG-4 也支持时域和空域的混合分级。

基于对象的编码除了能提高数据的压缩比, 还能实现许多基于内容的交互性功能, 如基于内容的多媒体数据存取、游戏或多媒体家庭编辑、网上购物和电子商店、远程监控、医疗和教学等。



## 2.2 MPEG-4 编码标准的基本概念

### 2.2.1 引言

基于对象的视频编码是以视频对象（VO）为基础的。所谓的对象是在一个场景中能够访问和操作的实体。对象的划分可以根据其独特的纹理、运动、形状、模型和高层语义为依据。每个视频对象含有内置的形状、运动、纹理等信息。

MPEG-4 视频编码技术采用了基于对象的编码方式<sup>[34-36]</sup>，这是 MPEG-4 能提供基于内容的交互性、改善压缩效率和通用的访问能力的基础。

MPEG-4 标准分为系统、视频和音频三个部分。MPEG-4 视频编码标准除了支持 MPEG-1 和 MPEG-2 提供的功能，此外还支持基于内容的功能，即能够按视频内容分别编解码和重建。以前的压缩算法只是去掉帧内和帧间的冗余，MPEG-4 则要求对图像和视频做更多的分析。如将图像分割为运动物体对象和静止不动的背景对象平面，并对这两个对象进行分别处理。背景对象采用压缩比较高、损失比较大的方法进行编码，运动物体对象采用压缩比低、损失较小的方法进行编码。这样就在压缩效率和解码图像质量间取得较好的平衡。

MPEG-4 校验模型是 MPEG 小组为 MPEG-4 标准的制定提供的完全定义的视频编码核心算法平台，为世界上各个实验室开展这方面的研究提供了基础和统一开发环境。图 2.2 是 MPEG-4 视频校验模型的编码方块图。

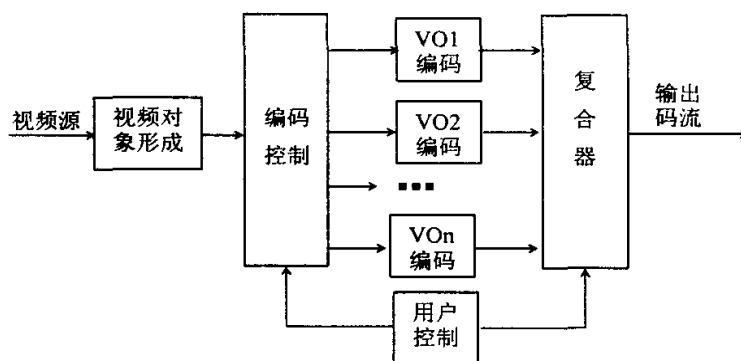


图 2.2 MPEG-4 视频校验模型的通用编码框架

在实际应用中，一个用户对不同的视频对象可以选择不同的参数、编码方式或者甚至不对某些对象进行编码来满足其需要。图 2.3 是 MPEG-4 视频校验模型的解码方块图，它的结构同编码器正好相反。合成器所采用的方法由用户和系统

及所采用的汇接方法来决定。不同视频对象之间的同步和视频对象于其他实体，如音频对象之间的同步由 MPEG-4 的系统层来处理。

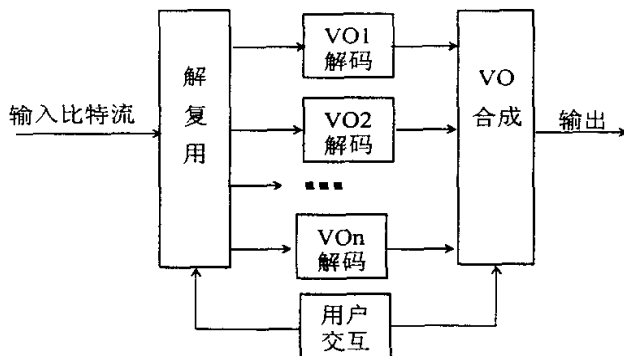


图 2.3 MPEG-4 视频校验模型的通用解码框架

MPEG-4 的视频校验模型提供的可选工具有：形状编码、纹理编码、运动编码（运动估计与补偿）、错误恢复、可分级性编码、Sprite 编码、纹理编码模式。在第三节我们将详细介绍以上概念。

MPEG-4 视频标准支持以下三类新功能：基于内容的交互性、高压缩率和通用访问<sup>[37-41]</sup>。

## 1. 基于内容的交互性

MPEG-4 允许在编码数据级上对压缩域中的视频物体进行存取和操纵，其目的是以高度灵活的方式使用和表示它们。尤其是将来的多媒体应用与计算机游戏以及相关的应用将得益于对视听内容日益增加的交互性。

### (1) 基于内容的多媒体数据访问工具

MPEG-4 通过使用各种数据访问工具，提供基于音/视频内容的数据访问，如索引、超链接、查询、浏览、上载、下载以及删除等工具。

### (2) 基于内容的操纵与数码流编辑

MPEG-4 允许对景物中不同的视频物体分别进行灵活的解码与表示，并通过对数码流的简单操作来控制原始景物的形状、透明性、空间坐标以及描述物体伸缩、旋转等相关参数。

### (3) 自然数据与合成数据混合编码

MPEG-4 支持解码器可控的将合成数据和原始音/视频组合在一起的方法。其应用范例有：在游戏中用自然的音频和视频结合人工产生的多媒体对象生成虚拟的场景，用户可以随意控制媒体对象，从不同的角度观察和描绘视频和声音。

## (4) 改进的时间随机存取

### 2. 高压缩率

这一功能实质上是针对需要对视听信息进行有效存储(或传输)的应用,这将增强如 Internet 上的信息浏览、虚拟现实等一些已有的应用。

#### (1) 改进的编码效率

提高编码效率,尤其是在甚低数码率(小于 64Kbit/s)的应用。MPEG-4 视频标准将提供更好的主观视觉质量。这一功能可望在迅速发展的移动通讯中得到应用,目前的 CDMA 标准为此提供了通讯协议平台,相信会成为无线通讯中最重要的业务领域。

#### (2) 对多路并存数据流的编码

MPEG-4 可以对景物进行有效的多视角编码,对于立体视频的应用, MPEG-4 能充分利用从多个视角对同一景物进行观察所造成的信息冗余度,提取该景物的 3D 模型。其应用范围包括:虚拟现实游戏、三维动画、工程建模等。

### 3. 通用访问

MPEG-4 标准涉及到在多媒体应用方面(尤其在无线通信和数据库存取)能预见的各种不同类型的网络环境,要求视频压缩算法能适应噪声环境、时变带宽以及不同的解码器可用资源。

#### (1) 容易出错环境中的健壮性

MPEG-4 将提供对差错的健壮性,以允许应用于各种无线、有线网络以及存储媒介。在恶劣的差错条件下,对于低数码率的应用,应提供足够的抗错能力。

#### (2) 基于内容的可伸缩性

MPEG-4 将提供在内容、质量(例如空间域分辨率和时间域分辨率)以及复杂度方面可伸缩的能力,允许对特定物体在时间域和空间域分辨率方面按不同的质量要求对编码数据进行存取,以便具有不同复杂度的解码器能对编码数据进行解码。用户能够选择或自动选择景物中各物体的解码质量,能按不同的内容等级、比例、分辨率和质量进行数据库的浏览。

通过以上对 MPEG-4 视频编码标准新功能的总结,我们可以看出 MPEG-4 提供了一个高度灵活的框架和一个开放的工具集,以方便地吸纳新的思想、新的软硬件技术,具有长久的生命力与广泛的适用性。

## 2.2.2 MPEG-4 的视频编码模式分类

MPEG-4<sup>[42-43]</sup>除完全支持 MPEG-1/2 已提供的全部视频功能, 包括在不同输入格式、帧率、比特率下对标准矩形区域图像序列进行有效编码外, 还增加了新的基于内容的视频编码功能。图 2.4 表示 MPEG-4 视频编码模型提供的模式和比特率的关系。

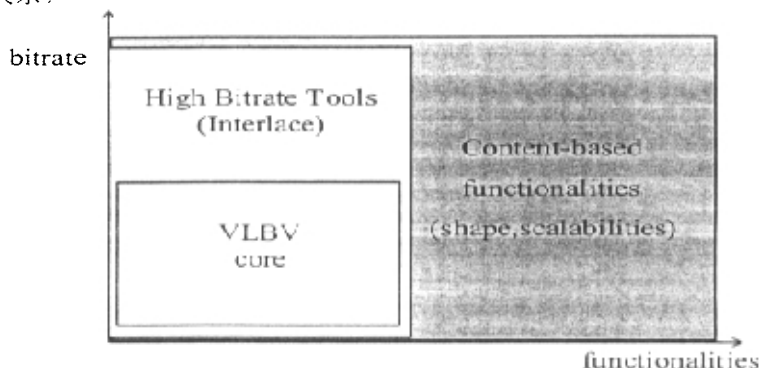


图 2.4 MPEG-4 图像和视频编码算法与工具分类

甚低比特率视频核(VLBV core)提供了从 5kbit/s 到 64kbit/s 的甚低比特率业务的算法和工具, 它支持甚低分辨率 (从每行每列几个像素到 CIF 级分辨率, 即每帧 288 行, 每行 352 个像素) 和甚低帧速率 (从 0Hz 到 15Hz) 的图像序列。

高比特率核(HBRV core)则提供了支持从 64kbit/s 到 10Mbit/s 的高比特率图像传输的工具和算法, 这些工具和算法与 VLBV core 所提供的相同。HBRV core 的主要应用对象是多媒体广播、数字电视图像级别的交互信号传递。

附加的基于内容的功能其核心在于支持视频按内容进行编解码, 扩展的 MPEG-4 基于内容的算法和工具可以看作是 VLBV 核和 HBV 核的超集。与标准的 MPEG-1 和 MPEG-2 对长方形图像序列进行编码不同, MPEG-4 的视频通常被认为是任意形状的, MPEG-4 基于内容的方法可以看作是 MPEG-4 VLBV 和 HBRV 核编码方法向任意形状的输入视频方向逻辑扩展的结果, 只有当原始图像序列没有被分割成任意形状的视频对象时, 才用传统的矩形图像序列进行编码。因此, MPEG-4 视频标准同时支持传统的以矩形帧为基础和现在的以具有任意形状的对象内容为基础的两种编码方式。

## 2.2.3 视频校验模型的数据结构

为了支持前面提到的各种功能: 高效压缩、基于内容交互(操作、编辑、访问



等)以及基于内容分级扩展(空域分级、时域分级),必然要求 MPEG-4 要以基于内容的方式表示视频数据。视频对象(VO)的构成依赖于具体应用和系统实际所处环境:在要求超低比特率的情况下,VO 可以是一个矩形帧(即传统 MPEG-1、H.263 中的矩形帧),从而与原来的标准兼容;对于对内容的细节表现要求较高的应用来说,VO 可能是场景中的某一物体或某一层面,如新闻节目中的解说员的头肩像;VO 也可能是计算机产生的二维、三维图形,等等。

MPEG-4 的视频校验模型的数据结构可以认为有四个层次的数据结构,它们都以类的形式定义,见图 2.5:

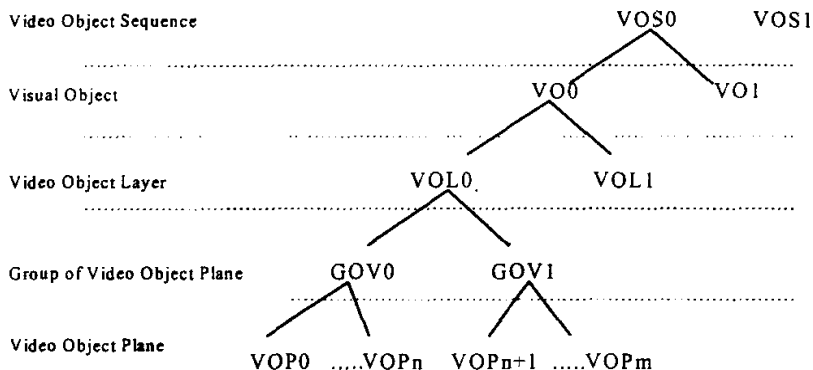


图 2.5 MPEG-4 视频校验模型的语法层次结构

视频对象序列 Video Object Sequence (VOS): 一个完整的视频序列可以由几个 VOS 组成。视频对象序列是指输入的一个实验序列,可以看成是一段画面。它包含一个或几个视频对象(VO)的集合。

视频对象 Video Object (VO): 如前所述,是场景中的某个物体,它是有生命期的,由时间上连续的许多帧构成。物体可伸缩性就是通过使用 VO 类实现。每个 VO 有三类信息来描述:形状信息、运动信息、纹理信息。一个视频对象可以包含一个或几个层,这些层可以是视频—视频对象层(VOL)或纹理内容—纹理对象层(SOL)。

视频对象层 Video Object Layer (VOL)或者是纹理对象层 Texture Object Layer (TOL): VO 的三种属性信息编码于这个类中,它是一个增强 VO 的时间域或空间域分辨率的类。这个类的引入主要用来扩展 VO 的时域或空域分辨率。每个视频对象层(VOL)包含一队由时间快照—视频对象平面(VOP)组成的序列。

视频对象平面 Video Object Plane (VOP): VOP 是一个 VO 在某一给定时刻

发生的一个事例，即某一帧 VO。两个不同的 VOP 可能属于两个不同时刻的同一个视频物体，而不一定属于两个不同的视频物体。VOP 相当于传统编码方式的一帧，它可以是任意形状的。

视频对象平面组 Group of Video Object Planes (GOV): 视频对象平面组 (GOV) 是一个可选层，它的主要功能是用来提供随机接入和错误恢复，因为 GOV 头包含着绝对时间信息。在一些应用轮廓，GOV 头后跟随的是 I-VOP。

视频校验模型的通用编码器流程：第一步是 VO 的形成，先要从原始视频流中分割出 VO，之后由编码控制机制为不同的 VO 以及各个 VO 的三类信息分配码率，之后各个 VO 分别独立编码，最后将各个 VO 的码流与场景描述信息和其它同步控制信息合成比特流，经过 MPEG 的多媒体继承传输框架 DMIF 包装后，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。DMIF 对不同的传输媒体提供了共同的接口并提供 QoS 服务。在解码端复合的比特流被分解，然后视频对象被解码并根据场景描述合成场景。视频校验模型的通用解码器流程基本上为编码器的反过程。

## 2.3 MPEG-4 视频编码关键技术分析

基于内容的视频编码系统主要由形状编码、运动编码和纹理编码等模块构成，具体的编码算法则作为实现系统中各个功能模块的工具，可以根据实际需要选取组合。尽管 MPEG-4 的视频校验模型主要针对任意形状的对象，但当前采用的工具大多是基于宏块的，这有利于同其他视频编码标准的兼容。当然，在不需要形状信息的应用中（如基于帧的视频编码），形状编码要被禁用。对于每一个视频对象平面（VOP）的编码流程可见图 2.6。

首先对输入的原图像序列进行场景分析和对象分割，以划分不同的 VOP，得到各个 VOP 的形状和位置信息。VOP 的形状和位置信息可以用一个二值的 alpha 平面来表示。

对运动和纹理信息编码仍然采用类似 MPEG 和 H.261/3 标准的预测/运动补偿法，该方法以被认为是四十年来经典图像编码方法研究成果的集中体现，是高效而且成熟的。输入的第 N 帧的 VOP 与帧存储器中的第 N-1 帧的 VOP 进行比较，找到运动矢量，然后对两帧 VOP 的差值进行量化、编码。编码后的纹理信息，与运动编码器和形状编码器输出的运动信息和形状信息，复接形成该 VOP 的比特流层。不同 VO 的 VOP 序列分别进行编码，形成各自的比特流层，经复接后在信道

上传送。传送的顺序一次为：形状信息、运动信息和纹理信息。接收端的解码过程是编码过程的逆操作。

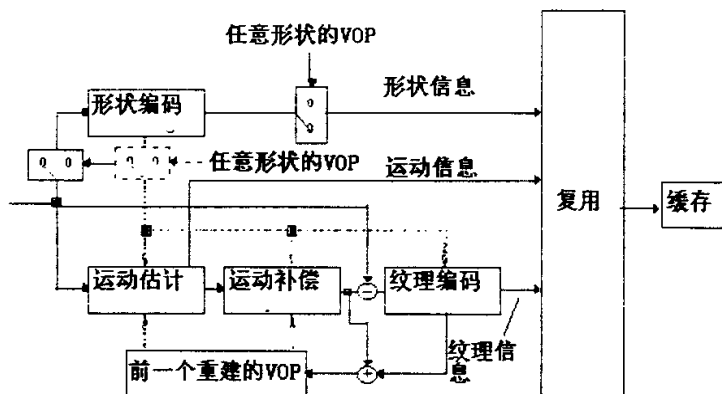


图 2.6 VOP 编码结构图

在某一时刻，VO 以 VOP 的形式出现，编码也主要针对这个时刻该 VO 的形状、纹理、运动这三类信息来进行。因此，MPEG-4 视频编码的主要工具就是对这三类信息进行编码的工具，编码系统结构见图 2.7。

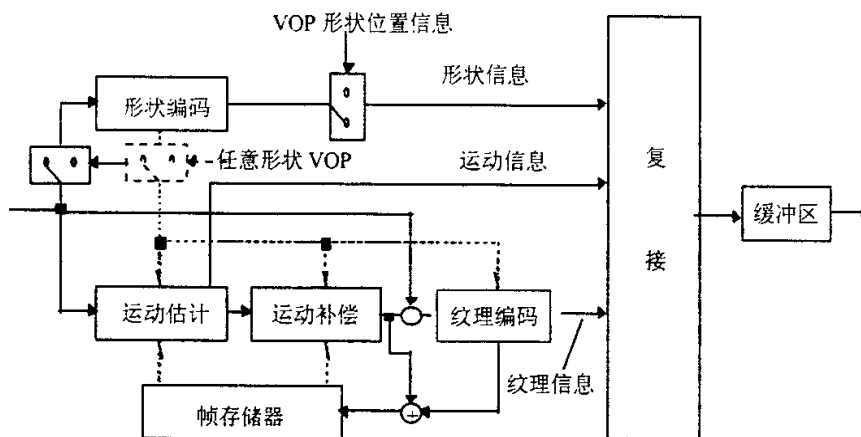


图 2.7 基于对象的视频编码系统

## 2.3.1 形状编码

视频对象的获取主要是通过分割视频码流并对其变化进行跟踪。图像和视频分割一直都是图像工程和机器视觉领域的重要研究课题，在 MPEG-4 标准中并没

有规定具体的视频对象分割算法，这一工作的终极目标是任意可操控的语义视频对象分割和提取，从而获得视频对象的形状和位置。

如图 2.8(c) 的 alpha 平面所示的，阴影部分用 0 表示，白色部分用 255 表示，编码端只需传送 alpha 平面，解码端就可以确定 VOP 的形状和位置。显然只要对 VOP 的轮廓进行压缩编码，解码端就可以恢复 alpha 平面。提取出的形状和位置信息又用来控制 VOP 的运动和纹理编码。如图 2.8 所示，(a) 是标准测试序列 news 的第一帧图像；(b) 为经过分割后的 VOP 图像；(c) 为 VOP 的形状和位置信息——alpha 平面。



图 2.8 标准测试序列 news 的 VOP 和 alpha 平面表示图

相对于以前的压缩标准来说，MPEG-4 视频编码校验模型第一次引入形状编码的算法。编码考虑两种形状信息：一种是二值形状信息；另一种是灰度级形状信息。二值形状信息通常用一个大小等同于某 VOP 的限制框的掩模矩阵来表示。矩阵元素的取值有两种可能，0 表示非 VOP 区域，255 表示 VOP 区域；灰度级形状信息与二值形状信息相比，有类似的对应结构，所不同的只是每个像素（矩阵元素）的取值范围是 0~255，此时取值代表了当前像素的透明度。灰度级形状对应于计算机图形学中的 alpha 平面的概念。灰度级形状信息的引入主要是为了使前景物体叠加到背景上时不至于边界太明显、太生硬，进行一下“模糊”处理。

由于基于位图的压缩技术相对简单且较成熟，迄今为止，视频 VM 一直采用该技术进行形状编码。实验结果表明基于位图的压缩技术具有高的压缩效率和相对



低的计算复杂度。

二值形状信息用基于块的运动补偿技术进行编码。该技术既可以进行无损编码,也可以进行有损编码。在当前 VM 中,每个 VOP 的形状和 VOP 的其它形状特征一起被编码。VOP 被一个“边界框”框住(如图 2.9),边框的大小是在水平和垂直方向各为 16 的整数倍的像素。矩形边界框位置的选定应使得边界框内所包含的  $16 \times 16$  非透明像素块的块数最少。边界框内及 VOP 外的样点值都置为 0(透明的)。将矩形边界框内所包含的像素划分成  $16 \times 16$  的二值 alpha 块,就相当于一个宏块的大小。形状编解码的过程是基于每个 alpha 块的。如果一个 alpha 块的所有值都是 0,那么对应的这个宏块在运动和纹理编码中被跳过不进行编码,从而不需有任何比特开销。对于二值 alpha 平面, MPEG-4 用改正的基于上下文的算术编码算法(Content-based Arithmetic Encoding, 简称 CAE)和基于块的运动估计和补偿算法对其进行编码。

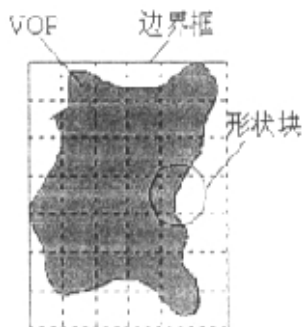


图 2.9 VM 中的 VOP 形状编码

灰值形状信息用基于方块的运动补偿 DCT 技术进行编码。类似于纹理编码技术,该技术只能进行有失真编码。

## 2.3.2 纹理编码

纹理信息有两种,可能是内部编码的 I-VOP,也可能是帧间编码的 P-VOP、B-VOP。对于 I-VOP,纹理对应于像素值;而对 P-VOP、B-VOP,纹理对应于运动估计后的补偿残差值。由于基于分块的纹理编码技术相对简单,且性能好,同时为了在 VM 中以直接的方式加入抗错等功能,视频校验模型选用基于方块的纹理编码。

MPEG-4 的校验模型的纹理编码方法主要有低通外推法 (LPE)、形状自适应的离散余弦变换 (SA-DCT) 和形状自适应的离散小波变换 (SA-DWT) 3 种方法,

低通外推法算法将分割出来的任意形状区域通过外推法填充成规则的矩形区域, 以便于基于分块的变换编码算法的应用, 采用低通外推算法进行填补既简单有效, 而且能在低频区域实现能量的高度集中。另外两种形状自适应的算法则需要利用到解码端的轮廓信息。

SA-DCT 最早是在 1995 年由 Sikora 等人提出的。它是在矩形图像块 DCT 编码方法的基础上发展起来的, 目的就是在实现复杂性、编码效率和对已有 DCT 技术的后向兼容性三者之间找到一个合理的折衷。SA-DCT 能够对任意形状的图像区域进行编码, 容易实现, 变换效率高, 而且同已有的 DCT 技术 (如游长编码、变长编码等) 完全兼容, 但是在应用中 SA-DCT 也暴露出了一些缺点, 即逆 SA-DCT 后图像量化噪声的统计参数 (如最小均方误差 MSE) 在相当大程度上取决于实际图像区域的形状。

S. Li 等人提出一种 SA-DWT 算法<sup>[4]</sup>。这种方法把任意形状区域内的像素变换成子带域上相同数目的小波因子, 同时保留了空间相关性, 局部性和自相似性。这种算法同 SA-DCT 相比效果更好, 但是它的计算复杂度较高。

要进行纹理编码首先将 I-VOP 边界框内的像素或经运动补偿后的残差值划分成若干  $16 \times 16$  的宏块。类似于运动估计和运动补偿的情况, 在一个 VOP 边界框内可能会出现三种类型的宏块, 即完全在 VOP 以内的宏块、完全在 VOP 之外但又在边界框以内的宏块、以及部分在 VOP 之外但又在边界框以内的宏块。完全在 VOP 之外的宏块不需要编码; 完全在 VOP 以内的宏块用常规的 DCT 方案进行编码; 对于部分位于 VOP 之内的宏块所属的  $8 \times 8$  块, 首先要进行填充处理。由于主要区别在于填充技术, 下面分别介绍的几种填充编码仅讨论第三种情况。

## 直接填充编码

与在进行运动估计时的情况一样, 不同的是对残差块处于块内同时又位于 VOP 之外的区域要用零填充。如果一个  $8 \times 8$  块内所有像素都是透明的, 则其值用 0 代替。经过填充后的块按常规的 DCT 方案进行编码。图 2.10 说明了用于任意形状 VOP 的基于宏块的纹理编码。

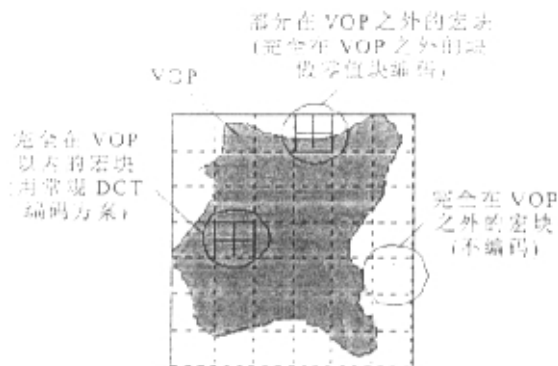


图 2.10 任意形状 VOP 的基于宏块的纹理编码

### 低通外推 (Low Pass Extrapolation, LPE) 填充技术

低通外推法对在 VOP 边界的块要先采用低通外推法进行填充。它可分 3 步：

1. 计算该块内属于对象内部区域 R 的像素算术平均值,

$$m = (1/N) \sum_{(i,j) \in R} f(i,j) \quad (2.1)$$

N 表示该块内位于 R 内的点数。m 取四舍五入后的整数；

2. 将 m 赋给块内每个不在 R 中的像素；
3. 对块内处于区域 R 外的点进行滤波操作，从块的左上角开始逐行操作一直到块的右下角结束，

$$f(i,j) = [f(i,j-1) + f(i-1,j) + f(i,j+1) + f(i+1,j)] / 4 \quad (2.2)$$

除的结果取整。如果滤波中使用的像素在块的外部，则这些不参加计算，相应的除数进行修改。例如当  $i=0$  且  $j=0$  时， $f(i,j) = [f(i,j+1) + f(i+1,j)] / 2$ 。

### 形状自适应的离散余弦变换 (SA-DCT)

与外推法不同，形状自适应变换只对对象区域内的像素进行处理，不对对象区域外的像素做任何操作。图 2.11 表示一个含不规则形状图像区域的边沿块进行 SA-DCT 的例子，图中 “·” 表示直流 DC。

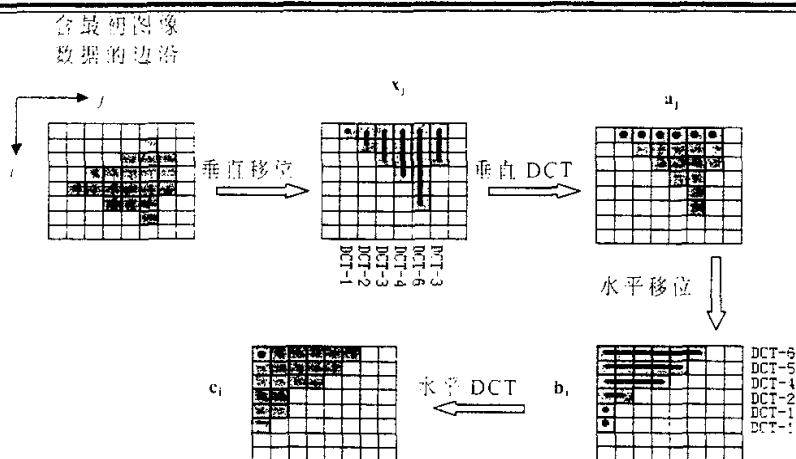


图 2.11 任意形状图像区域正向 SA-DCT 的过程

首先，每一特定列  $j$  的  $N_j$  个像素移到块的最顶部，每一列的像素会聚成列矢量  $x_j$ ，接着在垂直方向上对每一列矢量  $x_j$  进行标准的一维 DCT。变换后每一列的  $N_j$  个系数构成相应的系数矢量  $a_j$ 。接着，再把这些系数移到块的最左端，则每一行的  $M_i$  个元素构成一个行矢量  $b_i$ ，在水平方向上同样进行标准的一维 DCT，得到行系数矢量  $c_i$ ，这些系数也是最后 SA-DCT 的系数。从图 2.11 中可以看到，变换后的 DC 仍然位于块的最左上角的位置，系数的数目与像素的数目是相同的。

接着，对 DCT 系数进行量化、zig-zag 扫描以及采用游长 Huffman 方法的熵编码。量化步长有两种可能，两者基本上都是用系数去除量化步长。第一种方法是用两个可能的量化矩阵中的一个，根据系数的空间频率去除量化步长；第二种方法是对所有的系数使用相同的量化补偿，这个过程也允许 DC 值的非线性量化。

### 2.3.3 可伸缩性编码

MPEG-4 视频校验模型是使用 VOL 结构支持空间域和时间域的可伸缩这个重要功能的。从普通意义上讲，视频的可伸缩性就是能够同时得到多个分辨率或不同质量的视频。如果一个码流是可伸缩的，那么不同复杂度的解码器可以同时存在，低性能的解码器可以解码产生基本质量的一小部分码流，重建基本质量的图像；高性能解码器可以解码更多的码流并重建更好的质量的图像。

每种类型的可伸展性都涉及到一个以上的 VOL。在由低层和高层组成的两层中，低层称为基本层，高层称为增强层。视频校验模型的可伸缩性包括空间域可



伸缩性和时间域可伸缩性。对于时间域可伸缩性, MPEG-4 既支持帧(矩形 VOP), 也支持任意形状的 VOP。然而, 对于空间域可伸缩性, 目前只支持矩形 VOP。图 2.12 显示了两层可伸缩性的编解码结构。

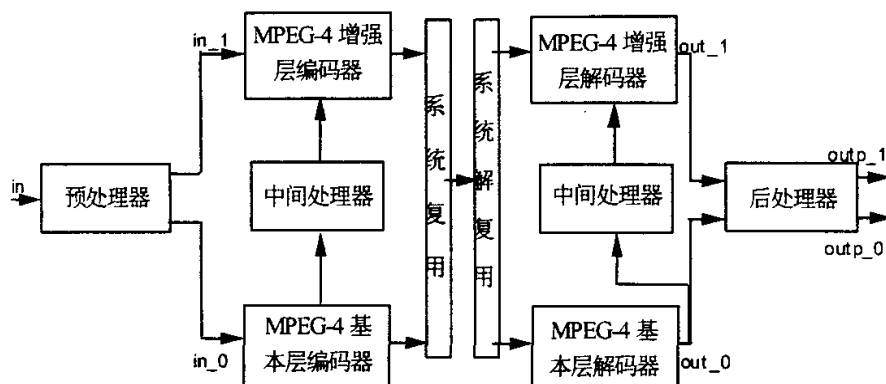


图 2.12 具有伸缩性的高级编解码器结构

从上图我们可以看到, 视频对象 VOP 输入到可伸缩预处理器中, 如果使用空域扩展编码, 那么基本层对应着低分辨率, 而增强层对应着较高分辨率图像。这个预处理器(preprocessor)执行输入 VOP 的空域下采样(down sampling), 形成基本层, 作为 PEG-4 基本层编码器的输入 in<sub>0</sub>, 经过编码、解码得到重建的 VOP; 基本层重构的 VOP 被输入到中间处理器中, 它将执行空域上采样(up sampling), 得到更高分辨率的 VOP, 并输入到增强层编码器中, 增强层编码器的另一个输入 in<sub>1</sub> 是预处理其产生的更高分辨率的 VOP, 它将参考重建 VOP 进行增强层的编码。基本层和增强层码流由系统复用(MSDL mux)后进行存储和传输, 基本层和增强层解码器可以通过系统解复用(MSDL demux)访问对应的基本层和增强层码流。解码器端的中间处理器和编码器端执行相同的操作, 可伸缩性编码后处理器(postprocessor)执行必要的采样转换操作。

当采用时域伸缩编码, 可伸缩性预处理器在时域上将一个 VO 分解成 VOP 两个子流, 其中一个被输入到 MPEG-4 基本层编码器中, 另一个输入到增强层中。在这种情况下, 中间处理器不执行任何空域分辨率的转换, 而是简单地将解码的基本层 VOP 输入到增强层编码器中, 增强层编码器将使用它们进行时域预测。系统复用和解复用的操作和空域伸缩类似。后处理器只是简单地输出基本层的 VOP, 并不进行任何转换, 但是在时域上混合基本层和增强层 VOP, 以产生更高

时域分辨率的增强输出。因为 VOP 有矩形或者不规则形状，所以传统的空域和时域伸缩和基于对象的伸缩编码都可以实现。

## 增强层的编码

在增强层中的 VOP 是以 P-VOP 或者 B-VOP 的方式编码。在基本层和增强层中的 VOP 对应关系如图 2.13 所示。在时间上和基本层 I-VOP 对应的增强层的 VOP 用 P-VOP 的方式编码；和基本层中 P-VOP 对应的增强层的 VOP 用 B-VOP 的方式编码。在空域伸缩编码时，在基本层中解码的 VOP 作为预测的参考 VOP，用作参考的基本层中的 VOP 编码必须在对应的增强层中的 VOP 编码之前先进行编码。

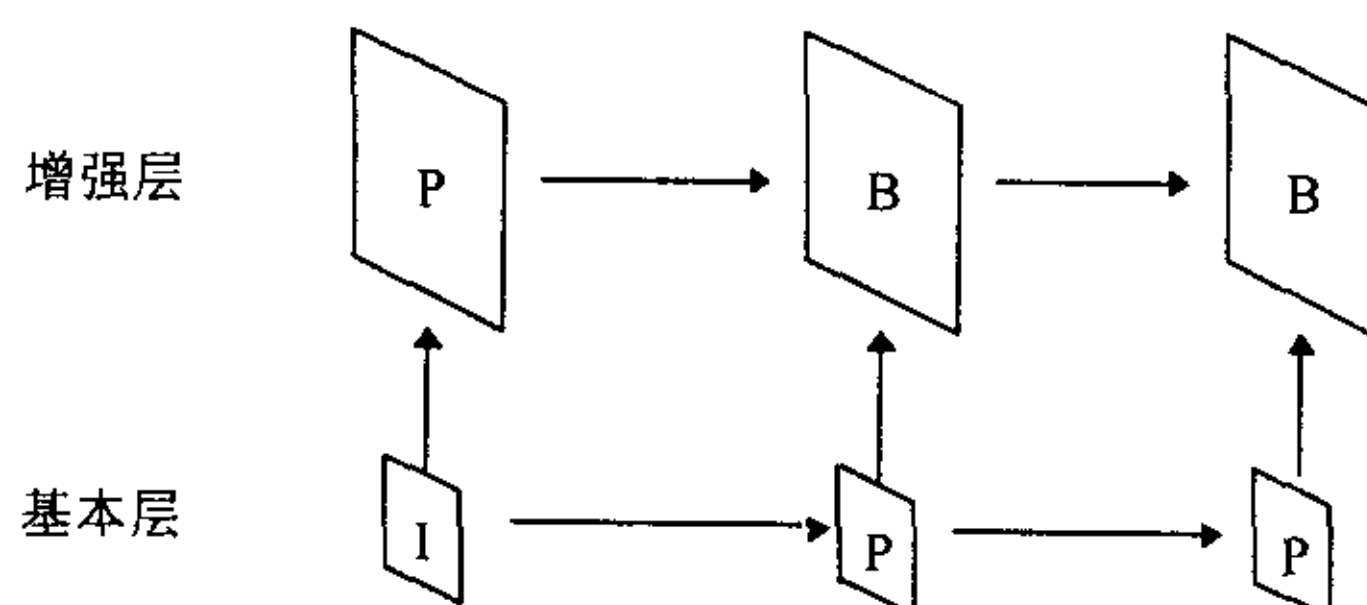


图 2.13 空域伸缩编码时基本层和增强层的对应关系

## 2.4 本章小结

作为新一代的视频编码标准，MPEG-4 编码的一个重要特点是具有“基于内容的功能”。本章首先对 MPEG-4 视频编码系统的框架结构进行描述，然后对 MPEG-4 中的形状编码、纹理编码、运动编码、可伸缩性编码等技术进行了详细的讨论与分析。

## 3 常用运动估计算法的分析

时间相关性是产生视频冗余的主要因素,本章研究的主要内容就是视频运动估计补偿技术及其相关的快速搜索算法,包括三步法、正交共轭法、象限法以及FOA统计法等。文章最后对正交共轭法和改进的正交共轭法进行了软件实现,并给出了相应的实验数据。

### 3.1 前言

MPEG-4 基于内容的特性要求视频编码器能对任意形状的图形对象进行运动估计补偿编码。校验模型在搜索匹配时采用了半像素搜索、无限制模式和高级模式等技术。经过前一章的论述,我们可以知道在运动估计中无论使用了哪一种模式,在对某一对象进行编码之前,由于要使用传统的基于宏块的预测方法,则必须对该任意形状的 VOP 进行填充以形成标准的  $16 \times 16$  宏块的整数倍大小的 VOP,具体填充方式在第二章已有介绍。鉴于上述原因且由 MPEG-4 的系统框架可知,基于矩形 VOP 研究的结果对于传统的和面向对象的视频编码系统均有实际意义,故这里我们依然以矩形 VOP 作为研究对象。

现在得到广泛应用的 MPEG-1 格式的视频采用的是 CIF( $352 \times 288$ )格式,74 分钟的文件大小约为 575MB(扣除音频部分,可根据 MP3 即 MPEG-1 layer3 标准推算得出)。视频压缩中仅利用空间相关性的 I-帧压缩类似于 JPEG 压缩标准,如果不采用去时间相关性的运动预测/补偿技术的话,则形成的是 JPEG 图像序列。一般的  $352 \times 288 \times 24$ (虽然视频原始文件采用的是 YUV 格式,其中 YUV 分量比为 4:1:1,但是其色深仍为 24b)的 JPEG 图像为 25K。由上述数字可知 74 分钟的 JPEG 序列大小为  $74 \times 60 \times 30 \times 0.025 = 3330\text{MB}$ ,考虑到 MPEG-1 本身为存储媒体要保留  $2 \times (\text{I-帧})/\text{秒}$ ,因此利用运动估计技术可以消除约 89%的文件大小。由此可见运动估计/补偿算法是视频压缩中起决定性作用的技术。

运动估计技术可分为像素递归算法(PRA)和块匹配算法(BMA),前者是以像素为基础来进行运动预测,而后者是以块为基础来进行运动估计。后者与前者比较,一般对不同的视频信号具有较好的鲁棒性,比较实用,所以多数采用块匹配运动估计,也是本文将要讨论的方法。

块匹配搜索算法最常用最小均方误差(MSE)或最小平均绝对误差(MAE)方

法, 在一定的搜索范围内从上一帧中找出与当前块最匹配的块。由于 MSE 和 MAE 对所需结果影响不大。为计算简便, 通常使用绝对误差总和(Sum of Absolute Difference, SAD), 其计算公式如下,

$$SAD_N(x, y) = \sum_{i=1, j=1}^{N, N} |original - previous| * (! (Alpha_{original} == 0)). \quad (3.1)$$

$x, y = \text{"up to } [-64, 63]\text{"}, N = 16 \text{ or } 8$

式中  $SAD_N(x, y)$  为在点  $(x, y)$  处进行块匹配时所得的绝对值误差总和;  $previous$  为前一帧里的像素点灰度值;  $original$  为后一帧里对应的像素点得灰度值;  $Alpha_{original}$  指示该点是否处于视频对象内部(仅用于 MPEG-4 里), 下面将先介绍 MPEG-4 校验模型中所采用的运动估计算法。

## 3.2 MPEG-4 编码标准的运动估计算法研究

### 3.2.1 基于对象的运动估计和补偿

为了利用大部分视频序列中的时间冗余度以提高压缩比, 在视频校验模型中引入了运动估计和补偿技术。

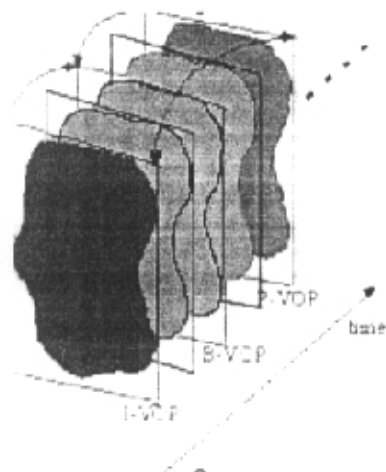


图 3.1 VOP 编码类型

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-4 视频 VM 的 VOP 也有三种编码方式(如图 3.1 所示):

(I) 帧内 VOP(I-VOP, intra): VOP 可以被独立编码, 于任何其它 VOP 无关。此时, 编码的 VOP 为一个帧内 VOP。

(II) 前向预测 VOP(P-VOP, predictive): VOP 可以基于前一个已解码的 VOP 做预测, 称为预测 VOP。

(III) 双向插值 VOP(B-VOP, bi-directional interpolation): VOP 根据前一解码重建的 VOP 和后继的重建 VOP 做出预测。这类 VOP 为双向插值 VOP。B-VOP 可以是基于 I-VOP 或 P-VOP 的插值帧。

MPEG-4 校验模型的运动编码(即运动估计与补偿)为了支持基于内容的视

频编码, 需要将运动估计和运动补偿技术加以扩展以便处理任意形状的 VOP 物体。这将通过将任意形状的 VOP 划分成  $16 \times 16$  的宏块并借助填充(padding)方法来实现。类似于其它标准如 H.263, MPEG-4 采用了半像素搜索、无限制模式和高级模式等技术。在这几种模式中, 运动矢量的搜索范围为  $[-2^{f\_code-3}, 2^{f\_code-3} - 0.5]$ , 其中  $f\_code$  在 0 到 7 之间选择。基本模式将运动矢量的范围限制在 VOP 的限制框内, 而无限制模式则突破这一限制。高级模式则允许一个宏块有多个运动矢量进行重叠运动补偿。这三种模式都需要对 VOP 进行基于宏块的填充。

在执行运动估计时, 首先将 VOP 边界框内的像素划分为  $16 \times 16$  像素的宏块。对于完全在 VOP 之外但又在其边界框以内的宏块, 不做运动估计(如图 3.1)。

对于完全在 VOP 以内的宏块, 采用于其它标准中类似的运动估计技术。对  $16 \times 16$  的宏块及其 4 个  $8 \times 8$  的子块分别使用块匹配技术, 用半像素精度为每个宏块和 4 子块的每一个各求得一个运动矢量。运动估计所使用的参考像素是过去或将来解码的 I-VOP 或 P-VOP 的像素。

对于部分在 VOP 以内的宏块, 采用改进块(多边形)匹配运动估计技术。在这种技术中, 由当前宏块中位于 VOP 内部的像素与参考 VOP 中的对应像素产生的绝对差值之和 SAD 作为匹配误差。由于匹配中用到的一些参考像素可能位于参考 VOP 之外(见图 3.2(a)), 而 VOP 区域以外的像素, 即透明像素是不存在的, 为了能从参考 VOP 内部的像素外推那些像素值, 需要执行基于块的扩充。同样, 为了进行无限制运动矢量搜索, 在边界框外扩展参考 VOP 时, 也用类似的填充方法。

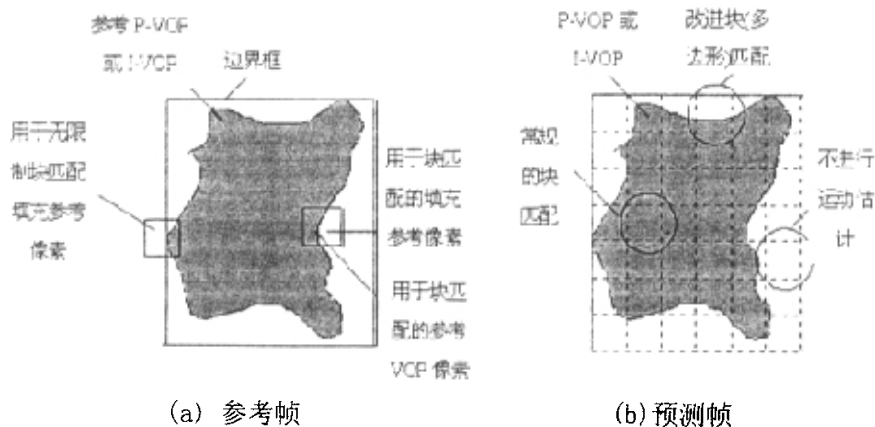


图 3.2 任意形状 VOP 运动估计

### 3.2.2 填充技术

为了对任意形状的 VOP 进行运动估计, 需要对于在 VOP 外的像素的亮度和色度值进行定义, 这就是填充过程。一个简化的填充流程图见图 3.3。

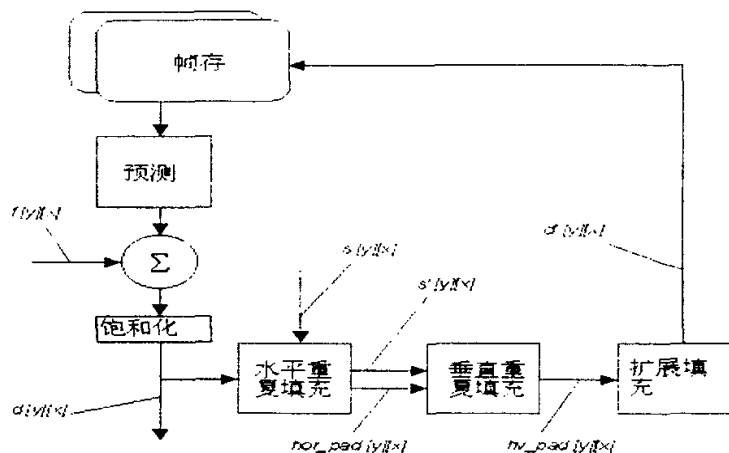


图 3.3 简化的填充流程图

填充法的具体过程是首先对于在 VOP 边界上的宏块（边界宏块），填充过程通过复制边界内的像素到边界外进行。这一过程先为水平重复填充后为垂直重复填充。然后对剩下的完全在 VOP 外部的宏块（外部宏块）则需取最近邻点的像素进行扩展填充。

对一个重建的宏块  $d[y][x]$  的填充需参考相应的重建的形状块  $s[y][x]$ 。水平填充(Horizontal repetitive padding)是取该点左端最近邻合法灰度值或者右端最近邻合法灰度值来填充，如果左、右两端都有合法像素值，那就采用两端的平均值。下面表示填充一行的过程，假设该行至少有一个像素值非零（如果全部为零，就等待下一步的垂直填充）。

```
for (x=0; x<N; x++)           //循环一行
{
    if (s[y][x] == 1)           //该点本身即为有效点, s[ ][ ] 为 VOP 的
                                //alpha 平面矩阵, 是水平填充的依据
    {
        hor_pad[y][x] = d[y][x]; //将原 VOP 值赋值给新生成的填充 VOP
        s'[y][x] = 1;             // s'[ ][ ] 为新生成的填充 VOP 的 alpha
                                //平面, 它会随着填充过程动态地变化,
    }
}
```

```

;
else                                为需要填充的点, 三种情况
;                                  //左右两边均有有效点
    if ( s[y][x'] == 1 && s[y][x''] == 1 )
    {   hor_pad[y][x] = (d[y][x'] + d[y][x''])/2;
        s'[y][x] = 1;
    }
    else if ( s[y][x'] == 1 )        //仅左边有有效点
        hor_pad[y][x] = d[y][x']; s'[y][x] = 1;
    else if ( s[y][x''] == 1 )      //仅右边有有效点
        hor_pad[y][x] = d[y][x'']; s'[y][x] = 1;
;
}

```

其中  $\text{hor\_pad}[y][x]$  表示水平填充后产生的值,  $s'[y][x]$  初始化为零。x' 表示和 x 左端最近邻合法像素的位置 (合法是指  $s[y][x'] = 1$ , 表示它是 VOP 内部的点), x'' 表示和 x 右端最近邻合法像素的位置。

垂直填充 (Vertical repetitive padding) 是根据该到位置上端最近邻合法灰度值或者下端最近邻合法灰度值来填充, 如果两端都有合法像素值, 就采用两端的平均值。下面公式表示一系列的填充过程。

```

for (y=0; y<M; y++)
{   if (s'[y][x] == 1)   hv_pad[y][x] = hor_pad[y][x];
    else
    {   if ( s'[y'][x] == 1 && s'[y''] [x] == 1 )
        hv_pad[y][x] = (hor_pad[y'][x] + hor_pad[y''] [x])/2;
        else if ( s'[y'][x] == 1 )   hv_pad[y][x] = hor_pad[y'][x];
        else if (s'[y''] [x] == 1 )   hv_pad[y][x] = hor_pad[y''] [x];
    }
}

```



其中  $hv\_pad[y][x]$  表示垂直填充后产生的值,  $M$  是一列的高度,  $y'$  表示和  $y$  上端最近邻合法像素的位置(合法是指  $s'[y'][x] == 1$ , 但不表示它是 VOP 内部的点因为垂直填充是基于已经进行过的水平填充的),  $y''$  表示和  $y$  下端最近邻合法像素的位置。

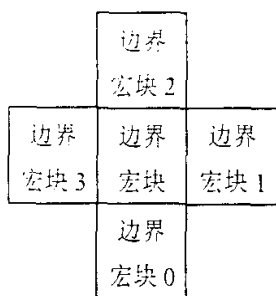


图 3.4 边界宏块的优先选择示意图

扩展填充是针对外部宏块的。对于同边缘宏块相邻的外部宏块, 可以用边缘宏块的边界值来对其进行填充。如果同时有几个边缘宏块与一个外部宏块相邻, 则采用优先数的方法, 见图 3.4, 环绕某外部宏块的边界宏块被赋予不同的优先数, 此外部宏块选择具有最大优先数的边界宏块的边界值来对其进行填充。对于不和任何边界宏块相邻的外部宏块, 用值 128 对其进行填充。

## 3.2.3 基本的运动估计和补偿模式

此模式包含多边形匹配、整像素的运动估计、帧内/帧间模式选择、半像素搜索、预测运动块的模式选择、交织的视频预测模式决策和对运动矢量的区分编码等技术。运动搜索、预测等其它技术与传统的基于帧的视频编码采用的技术相似。

多边形匹配用于边缘宏块, 它主要利用形状信息, 对当前宏块在 VOP 内的像素值与参考 VOP 中的相应像素值计算绝对误差和 SAD 值, 见图 3.5。

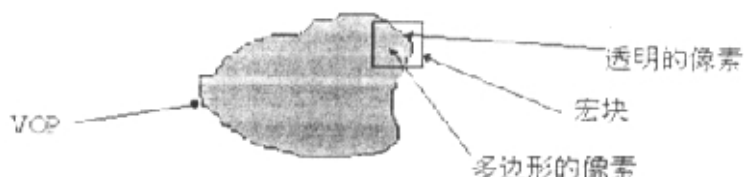


图 3.5 对任意形状 VOP 的多边形匹配

运动估计/补偿流程如图 3.6:

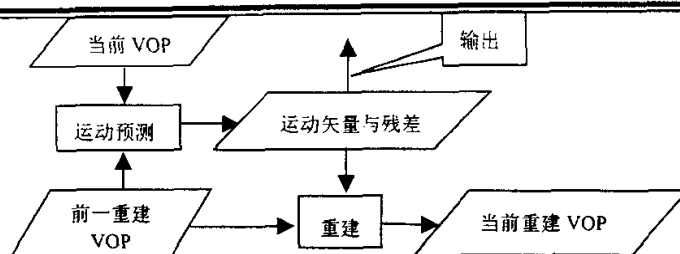


图 3.6 运动估计/补偿流程图

整像素运动估计是比较当前宏块和前一重建 VOP 中被替换的宏块的差值。见公式(3.1)。

在搜索范围之内具有最小的 SAD16 的  $(x, y)$  就作为  $16 \times 16$  宏块的整像素运动矢量  $V_0$ ，在这个矢量位置附近正负 2 个像素的窗口内，对于宏块内部的 4 个  $8 \times 8$  子块可以找到 4 个运动矢量  $V_1, V_2, V_3$  和  $V_4$ 。

在整像素的运动估计后，编码器根据所计算的 SAD 进行帧内/帧间模式选择。如果采用帧内方式，则停止运动估计；否则继续在  $V_0$  附近继续进行半像素的运动搜索。

半像素搜索需要对参考重建的 VOP 进行线性插值，它搜索的范围是  $V_0, V_1, V_2, V_3, V_4$  所指向的目标块的正负 1 个半像素的区域。

### 3.2.4 无限制的运动估计和补偿模式

在无限制的运动估计中，需把参考 VOP 进行边界扩展和填充，使得在矩形框边缘的宏块运动估计不必考虑边界限制。边界框的填充范围是向外的 16 个像素。计算运动矢量仍需用多边形的匹配来进行。

### 3.2.5 高级预测模式

在高级预测模式中，宏块的运动估计可以是进一步基于  $8 \times 8$  的像素子块进行的。高级预测模式的关键是在编码器将每一个宏块分为四个  $8 \times 8$  的像素子块，每一个子块独立进行运动估计，产生四个运动矢量，然后分别进行差分编码。每个矢量的差分编码值由其减去预测因子而获得，而预测因子分别取自三个候选预测因子的均值。这三个候选预测因子是三个已被解码的相邻像素子块  $(8 \times 8)$  的运动矢量。此措施利用了相邻宏块的运动相关性，好处是既保证精确恢复运动矢量，又减少运动矢量信息的编码比特数，能有效的消除方块效应，提高主观质量。

矢量差分编码的候选预测因子如图 3.7 确定。图中 MV 为分别对应宏块的像素子块( $8 \times 8$ )的运动矢量, MV1、MV2 和 MV3 是三个候选预测因子。

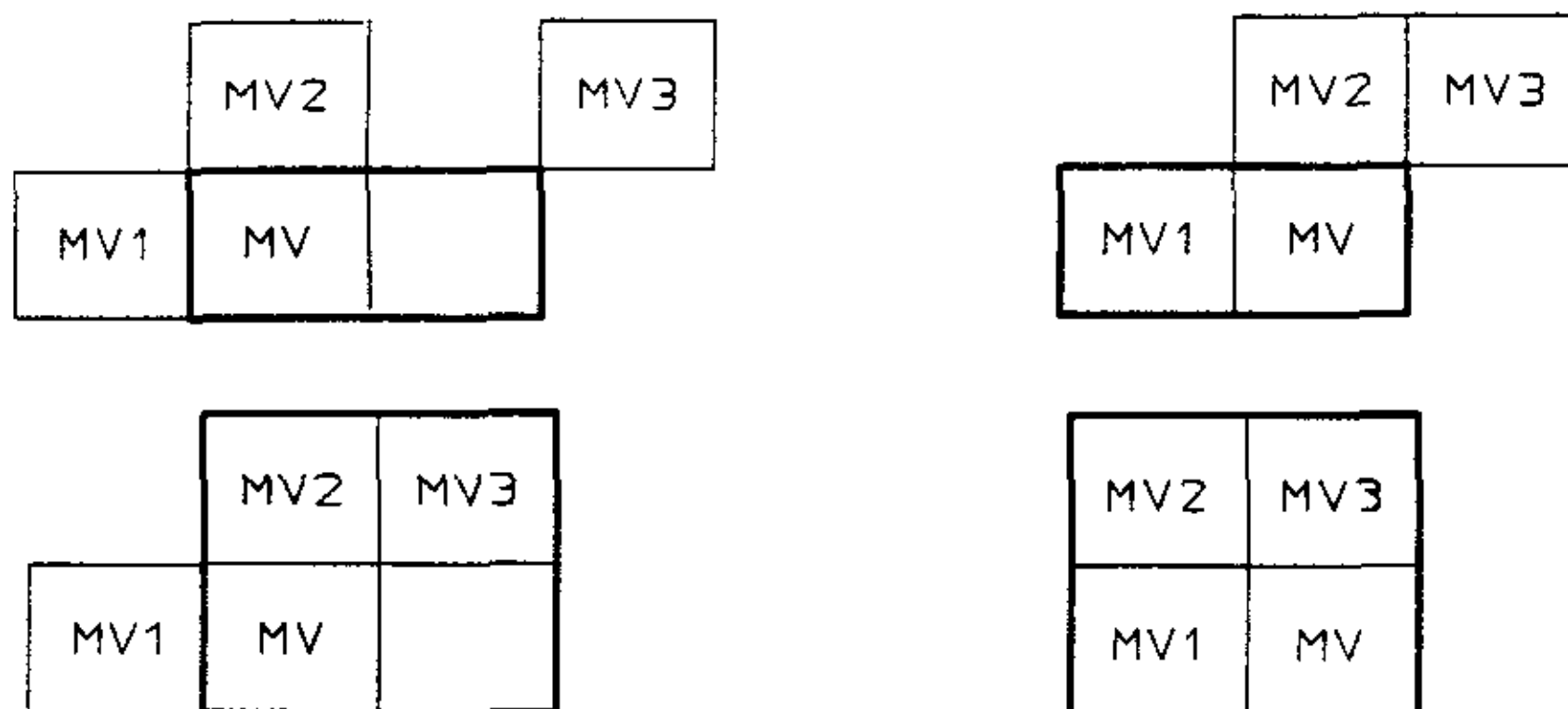


图 3.7 对每个宏块中的亮度块选择候选因子

块重叠的运动补偿技术是针对  $8 \times 8$  的灰度块的。此块中的像素值是通过三个预测值得加权和来得到的。这三个预测灰度值分别由三个运动矢量进行匹配获得, 这三个运动矢量中一个是本子块的运动矢量, 另外两个是在水平和垂直方向离此像素最近的邻近子块的矢量, 例如, 像素位于本块的左上角, 则这三个运动矢量分别用于本块, 位于本块上方的块和位于本块左方的块。

如果邻近子块没有被编码, 那么相应的运动矢量被认为为零; 如果邻近子块是以帧内模式编码, 那么此运动矢量被当前块的运动矢量替代; 如果当前块处在 VOP 的边缘, 那么不存在的邻近子块的运动矢量被当前块的运动矢量替代; 如果当前块在宏块的底部, 那么在此宏块之下的邻近子块的运动矢量也被当前块的运动矢量替代。

由上所述可知一般运动估计/补偿流程, 假设当前帧可分解成  $N \times N$  非重叠的块阵列, 最大的搜索步长为  $W$ 。块匹配搜索算法的目的就是在  $[x \leq W, y \leq W]$  的范围内找出运动矢量  $(x, y)$ , 使得 MAE 最小。在 H.263 中, 使用的块可以是宏块( $16 \times 16$ )或块( $8 \times 8$ )也可混和使用。在 H.263 基本结构中, 采用  $16 \times 16$  的宏块进行预测, 搜索范围  $W=15$ 。用全搜索方法(FS), 每个块需要进行  $(2W+1)^2 \times N^2$  次减法和绝对值运算才能在搜索窗内找到全局最优运动矢量, 每帧都需要巨大的运算量, 搜索效率不高。为了提高搜索速度, 必须采取快速算法来确定最优点。

### 3.3 常见快速搜索算法介绍

在各种算法当中基本思想大致有三种情形: (1) 减少参与搜索的位移矢量,

即根据某种策略,从搜索窗内选取某些位移矢量作为候选矢量比较,并不对搜索窗内所有的位移矢量进行计算或比较,这样,就减少了运算量,加快了匹配速度;(2)通过改进匹配失真函数的计算,来加快整个匹配速度,即对计算 SAD 值的公式采用投影的近似方法,达到减少运算量的最终目的;(3)对运动场进行 2:1 等多速率抽样,对抽样子块进行匹配。

这里专门讨论第一种情形。这类算法分多步进行时,每一步都要减少候选运动矢量的数目,如何选取并确定候选运动矢量是其关键。由于策略不同,分别产生了不同的快速算法,下面对其中常用快速块匹配搜索算法(如三步法、共轭法等)以及一些新的改进算法(如象限法,统计法等)进行简要介绍和分析<sup>[45-47]</sup>。

### 3.3.1 三步法(Three Step Search, TSS)

TSS 的算法如图 3.8 所示,这里假设搜索范围为  $W=8$ ,以当前块的左上角点  $(0,0)$  为初始点去与前帧的相应块比较,计算流程如下。

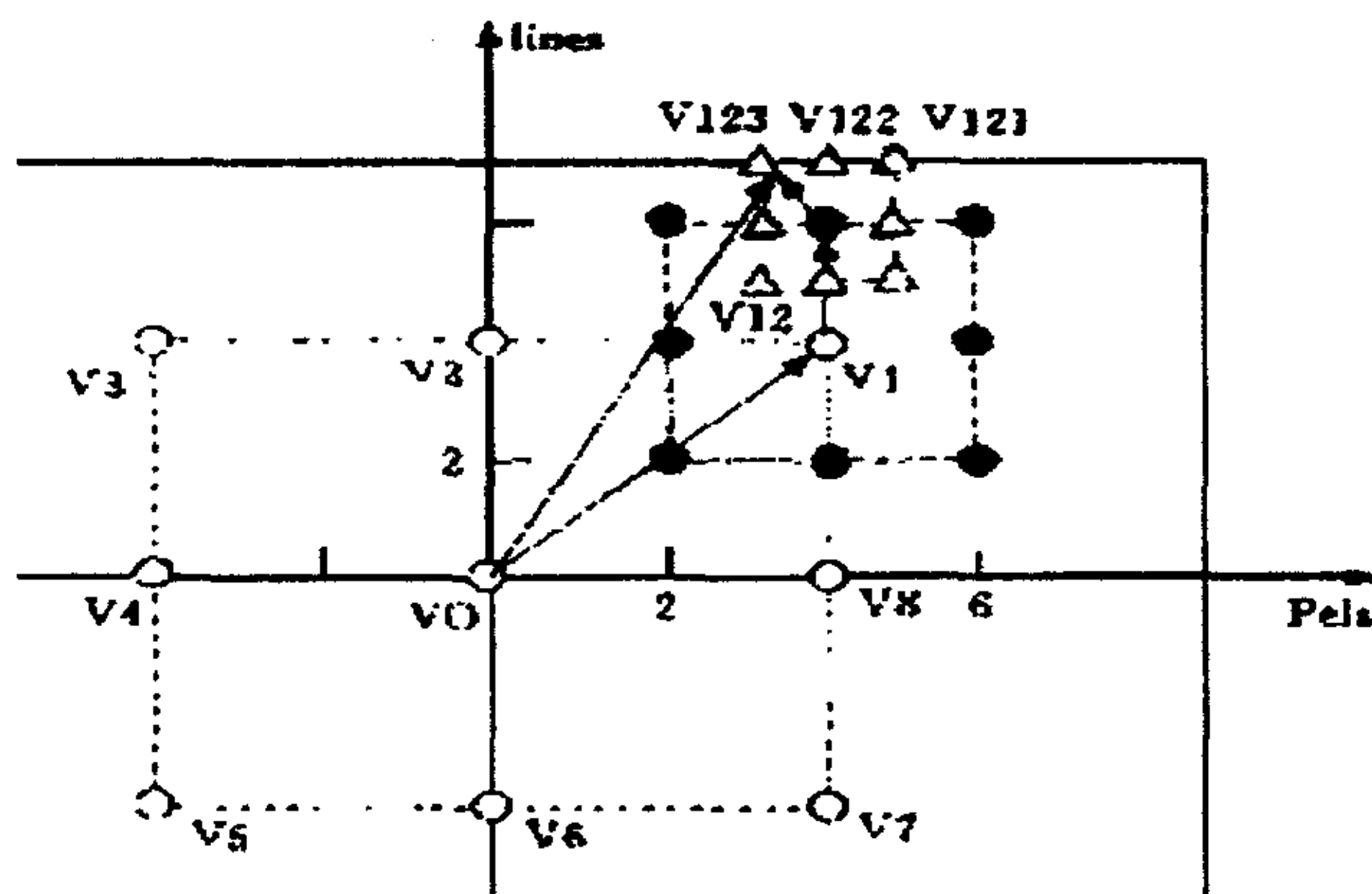


图 3.8 TSS 搜索过程图

- (1) 计算中心点的 SAD 值  $SAD(0,0)$ , 置搜索步长  $L=[W/2]$ , 置  $i, j$  为零;
- (2) 计算中心点周围八个点  $(i+L, j+L), (i, j+L), (i-L, j+L), (i-L, j), (i-L, j-L), (i, j-L), (i+L, j-L), (i+L, j)$  的 SAD 值并与中心点比较, 找出最小 SAD 值的点, 设为  $V_j$ ;
- (3) 搜索步长  $L$  减少一半;

- (4) 在可能搜索步长的条件下, 以步骤(2)中的  $V_j$  点为中心点, 重置  $i, j$  值为  $V_j$  点的坐标值, 重复步骤(2)。

我们可以看出 TSS 算法在搜索区域越大时搜索的速度越快, 其加速度也越大。同时, 它在有单一极值的搜索区域内可以以较少的步数得到较好的效果, 特别是图像的运动较剧烈、范围较大时效率很高。但是, 由 TSS 搜索流程图可知这个算法不能体现零向量点优先收敛的原则, 即对于所搜索到的最佳匹配块所得到的零向量和非零向量都需要一样多的计算量, 因此延迟较大。在 H.263 中,  $W=16$ , 则需进行 33 次块间运算, 即 SAD 值的计算。而且在存在多个极值的搜索区域进行搜索时由于其不可回归性而可能一开始就选择了错误的搜索方向而无法得到纠正, 导致最终搜索结果很不理想。具体实验结果见 4.3 节。

### 3.3.2 共轭方向法 (Conjugate Direction Search, CDS)

块匹配实质上是寻找 SAD 关于位移变量  $dx, dy$  的最小值的优化问题。共轭方向法就是根据拟 Newton 法的思路提出来的, 如图 3.9 所示。位移变量的初始值选定后, 固定一个坐标方向 (如垂直坐标), 而在另一个坐标 (水平坐标) 方向上按增量移位, 找到 SAD 在这个方向上的极小值点。再由该点和初始点确定搜索方向, 找到下一个极小点。在此基础上, 再固定水平坐标, 在垂直坐标方向上按增量移位, 搜索近似 SAD 全局最小值, 得到匹配位移矢量。

在上述算法的基础上, 每次只考虑一个方向而固定另一个坐标的方向, 也能得到性能较好的匹配矢量, 这种简化共轭法通常称为 OTS (One-at-a-Time Search) 算法。

以前一帧中的搜索区域的中心点  $(0,0)$  为原点建立  $x-y$  坐标, CDS 的算法描述如下:

- (1) 设  $i=0, j=0$ 。计算  $SAD(i, j), SAD(i-1, j), SAD(i+1, j)$ , 找出最小 SAD 值。如假设是  $SAD(i+1, j)$ , 则再计算  $SAD(i+2, j)$ , 从  $SAD(i, j), SAD(i+1, j), SAD(i+2, j)$  中找出最小值。重复该过程直到找到的最小值如  $SAD(l, 0)$  都小于两相邻点的  $SAD(l-1, 0), SAD(l+1, 0)$ 。则完成了  $x$ -方向的搜索;

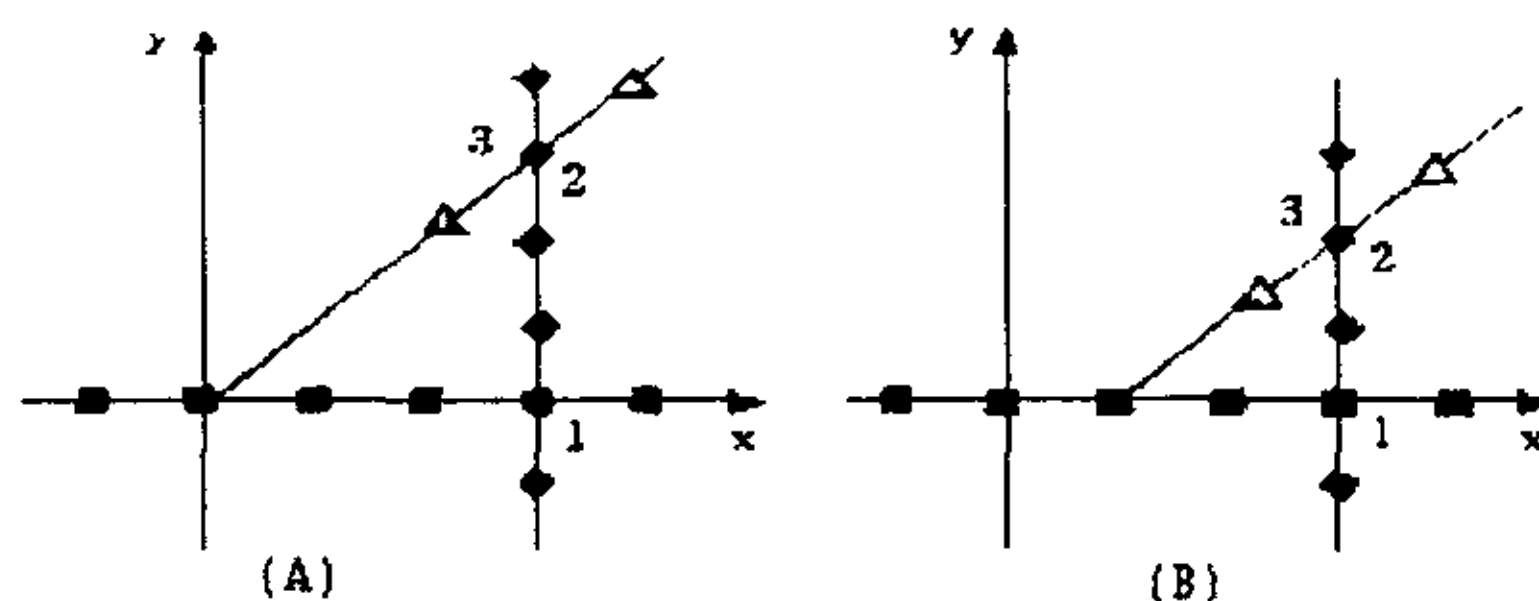


图 3.9 CDS 方法的搜索

□ 表示  $x$ -方向的搜索点, 其中 1 为  $x$ -方向的最优点  
 ◆ 表示  $y$ -方向的搜索点, 其中 2 为  $y$ -方向的最优点  
 ▲ 表示近似搜索方向轴上的搜索点, 其中 3 为该方向的最优点, 也是全局最优点 SAD 值

- (2) 以步骤(1)搜索到的(1,0)点为起点, 进行  $y$ -方向的搜索, 搜索的方法与步骤(1)相似。假设找到的目标点为(1,  $k$ );
- (3) 以步骤(2)搜索到的(1,  $k$ )点为起点, 在(0,0)和(1,  $k$ )两点建立搜索方向轴上进行搜索的方法与步骤(1)相似。需要说明的是在步骤(2)搜索的目标点(1,  $k$ ), 如果是图 3.9(A)这种情况即  $k$  等于 1, 则从(0,0)到(1,  $k$ )直接建立搜索方向; 如果是  $k$  不等于 1 这种情况, 从(0,0)到(1,  $k$ )必须近似地建立搜索方向, 如图 3.9(B)所示。

当 SAD 局部最小(Local Minimum, LM)和全局最小(Global Minimum, GM)比较接近时, 如图 3.10(A)所示的分布, 共轭方向发产生的误差较小, 搜索的性能也较好; 而当 SAD 局部最小和全局最小相差较大时, 如图 3.10(B)所示, 这时的误差就会比较大。

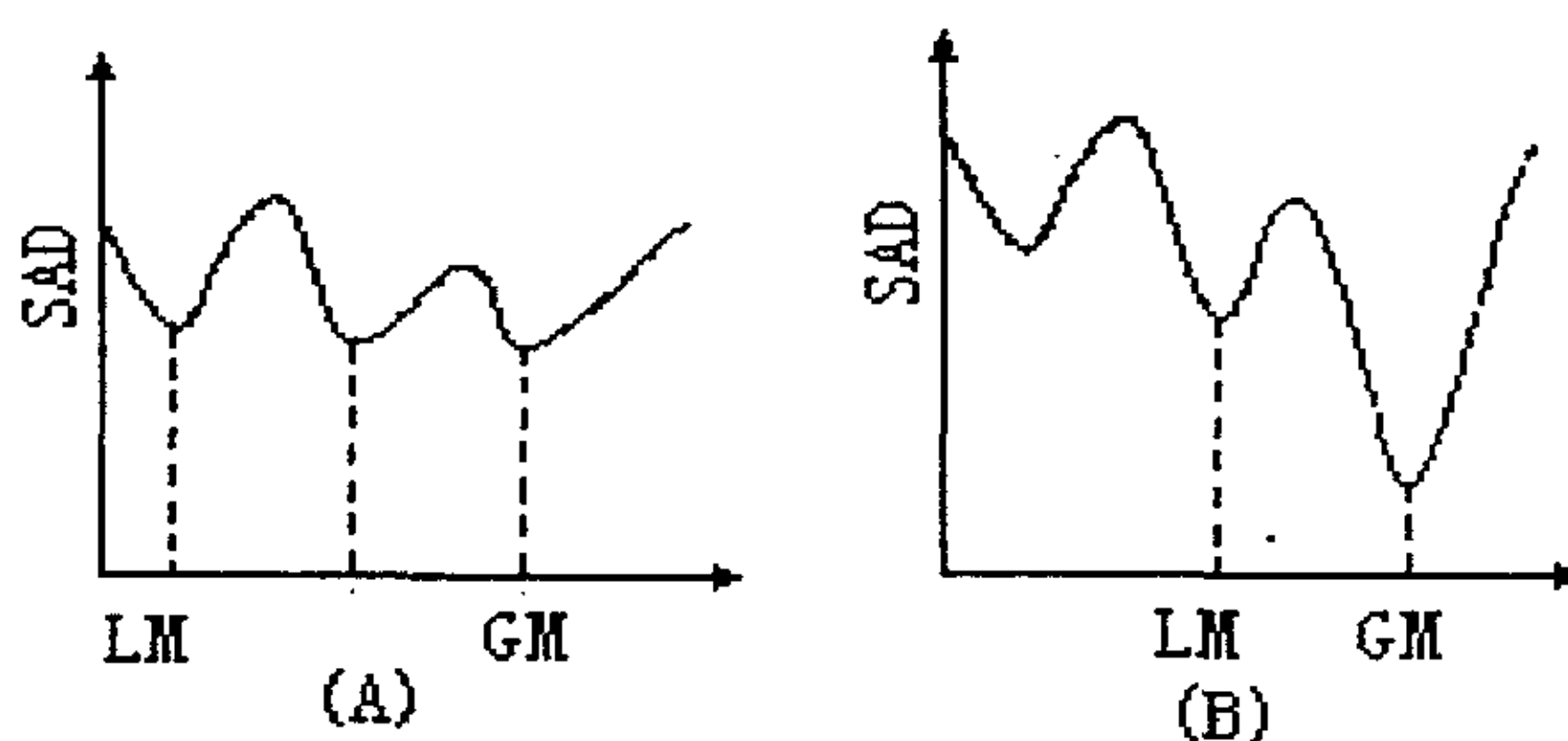


图 3.10 极值分布情况图

由 CDS 的算法流程可知, 其第三步存在着两个缺陷: (1) 当 1 不等于  $k$  时,

其与原点的连线不会经过所有的在其路径上的像素点,这就必然导致误差的产生,而且由于要寻找最近点会增加算法的复杂性;(2)在模拟的存在唯一最优点的单极值搜索区域中,这种方法往往并不能经过全局最优点。因此这里我们提出了一种改进算法,其算法在  $x$  和  $y$  轴进行搜索时的步骤与 CDS 法相同,区别在于第三步,描述如下:

(3)判断点  $(1, k)$  是否为点  $(i, j)$ , 如果是则结束搜索,最佳点即为  $(1, k)$ ; 否则使得  $i=1, j=k$ , 回到第一步重新进行搜索。

此搜索方法实现简单,仅在横竖坐标轴上进行搜索。其实际搜索效果与 CDS 基本相同(根据上文所述应该说是从理论分析的角度看上实际效果应该会更好一些),唯一略有不足的是可能因为反复地在  $x, y$  轴上进行搜索会稍微增加一些运算量。由于此法是从 CDS 改进过来的,因此我们决定使用该算法代替 CDS 算法对视频序列编码,在 3.4 节提供了该实验的结果。

### 3.3.3 象限法(QES)

象限法是在三步法的基础上引出的。如果搜索方向背离全局 SAD 最小点的距离越大,那么块匹配误差就会越大,这就是说最佳

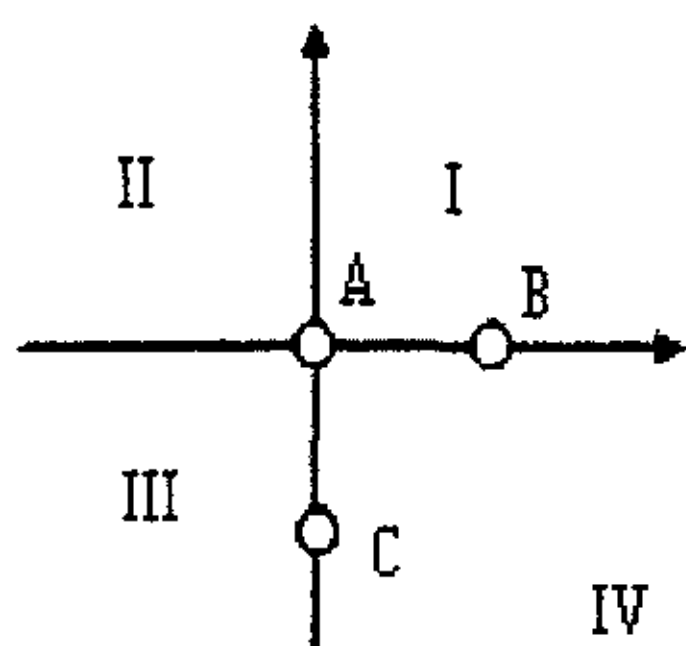


图 3.11 象限法示意图

点不可能同时出现在相互背离的两个方向。对于单极值 SAD,采用该方法是很有效的。在进行每一步块匹配时,先计算 3 个位置(如图 3.11)的 SAD,然后按下述规则确定搜索象限:

- (1) 如果  $SAD(A) \geq SAD(B)$ , 同时  $SAD(A) \geq SAD(C)$ , 则选象限 I;
- (2) 如果  $SAD(A) \geq SAD(B)$ , 同时  $SAD(A) < SAD(C)$ , 则选象限 II;
- (3) 如果  $SAD(A) < SAD(B)$ , 同时  $SAD(A) < SAD(C)$ , 则选象限 III;
- (4) 如果  $SAD(A) < SAD(B)$ , 同时  $SAD(A) \geq SAD(C)$ , 则选象限 IV。

根据每个选中的象限,如图 3.11 所示,其中 A, B, C, D 分别对应 I, II, III, IV 象限,进一步计算相应的位置(如黑点所示)的 SAD。

由象限法的计算流程可知其本质是与三步法类似,因此在继承了三步法的搜索速度快的优点的同时,也不可避免地会遇到不能体现零向量点优先收敛原则的问题;而且在多个极值的搜索区域进行搜索时最终搜索结果很不理想的问题。



### 3.3.4 统计法

如图 3.12,  $x=(i, j)$  为在搜索窗内的坐标,  $P$  为  $SAD(x)$  的全局最小值点, 对应位置为  $D$ , 另有两个局部极小值  $Q$  和  $R$ 。明显地, 如果候选位移矢量是  $A$ 、 $B$ 、 $F$ , 则搜索结果就可能是  $Q$  和  $R$ , 这样产生的误差就比较大, 而如果确定  $C$  和  $E$  为候选者, 那么就能够最终  $P$ 。所以, 根据与  $D$  距离  $d(x, D) = |x-D|$  来选定候选位置是可行的。于是, 将候选位置转化为下述优化问题

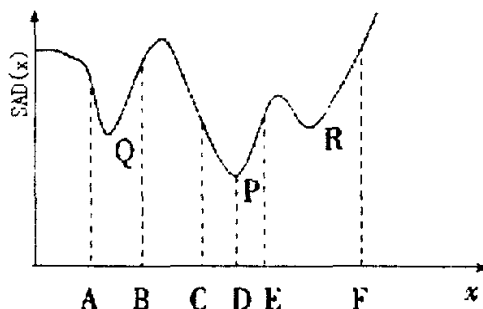


图 3.12 多极值分布示意图

$$\bar{d} = \sum_{x_i \in SR} |C_{x_i} - x_i| P(x_i) \rightarrow \min \quad (3.2)$$

其中:  $x_i$  为搜索窗  $SR$  内的点;  $C_{x_i}$  为最接近  $x_i$  的观察点;  $P(x_i)$  是  $x_i$  为最佳匹配矢量的概率, 这个参数通过对整个视频序列的观察统计获得, 即利用视频序列的时间及空间相关性来构造运动场, 得到运动矢量分布  $P(x_i)$ <sup>[48]</sup>。

统计法较好的利用了视频序列的空间-时间相关性, 消除了因多极值区域而带来的匹配误差。但是其  $P(x_i)$  最佳匹配矢量的概率的获得是通过对整个视频序列的观察统计获得, 不仅需要较大的运算量还需要较长的学习时间, 对于实时处理而言实际应用的可能性不大。

## 3.4 常见快速搜索算法试验评估

我们对比较常用的正交共轭法和改进的正交共轭法(CDS)进行了实验仿真。实验将上述两种快速搜索算法嵌入到 MPEG-4 验证模型中分别对 Susie 和 Foreman 两段 Qcif 视频进行运动估计算法的编码处理。Susie 是近距离的人脸图像序列, 有一定的不规则的头部运动; Foreman 是场景有大幅度但平滑变化的图像

序列。在这里需要介绍的是 Qcif 视频文件格式和编码配置文件中的细节参数，它们将会直接影响到实验的结果。

Qcif 是由一组灰度分辨率为  $176 \times 144$  的 YUV 图像序列组成的视频文件，其中 YUV 格式所采用的比例是 4: 1: 1。由于此实验序列的图像变化比较平稳，为了增加视频运动的幅度和造成搜索算法之间的差距，我们采用隔 3 帧编码一次的方法。因此图像序列因为每三中缺少两帧而导致变化较大较剧烈，使得编码环境比较接近实际的视频。我们仅对两个序列的前 150 帧进行编码。实验在 PC 机上进行，编码后生成压缩的视频流文件和解码还原后的 Qcif 视频文件，其结果如下表所示：

表 3.1 实验数据

指标\算法	正交共轭法(CDS)		改进的正交共轭法(CDS)	
	Foreman	Susie	Foreman	Susie
PSNR_Y	30.192186	34.839630	31.364222	35.429214
PSNR_U	36.380104	42.104721	37.011097	42.480751
PSNR_V	37.173290	41.350353	38.032509	41.744095
压缩率	45.28	50.94	47.9	52.7





对 Susie 和 Foreman 两段视频观察可知在 Susie 的 17, 19 帧和 Foreman 的 15, 27 帧有着较大的运动幅度，搜索算法的区别也最大故在此提供这几帧的抓图比较：

表 3.2 Foreman 实验抓图

算法\帧号	第 15 帧	第 27 帧
改进的正交共轭法(CDS)		



表 3.3 Susie 实验抓图

算法帧号	第 17 帧	第 19 帧
改进的正交共轭法(CDS)		
正交共轭法(CDS)		

仔细观察可以看出 PSNR\_Y 的值对图像质量有着一定的影响。PSNR\_Y 的值越大图像质量越好。由上述实验结果可以看出，改进的正交共轭法的信噪比相对于原有的正交共轭法有了相当大的提高，而且压缩比也提高了。

3.5 本章小结

本章对 MPEG-4 校验模型中所采用的运动估计算法进行了探讨，并对目前已经较成熟的快速运动估计算法进行了介绍和分析。对于正交共轭法提出了改进意见，最后对正交共轭法(CDS)和改进的正交共轭法进行了软件实验。

## 4 基于搜索起始点的运动估计改进算法

第三章的几种快速搜索算法对多极值的问题始终无法得到解决。而要避免漏选全局最佳点必须要进行全搜索。据此我们经过详细的理论分析后提出了一种新的快速搜索算法,即利用运动的空间-时间相关性来预测出较好的初始搜索点来跟踪运动的变化,尽可能在一开始就接近全局最优点来消除多极值的存在。前面所提出的快速算法无法进行多进程的搜索,而本章提出的算法前后搜索点无关可以并行处理,利于用硬件实现。我们对其进行了软件仿真试验并给出了试验结果。

### 4.1 改进算法分析

根据上述介绍的算法我们可以看出它们能实现快速搜索在于能够迅速地判别出极值点所在方向并逐步逼近,而缺点就是判别方向时可能因为多极值的存在而造成“全局误判”。如果能够近似地消除多极值的存在的话,上述算法的实用性将大大提高。在统计法里,提到了一种利用空间时间相关性的概念,但是由于其方法是基于大量统计的基础上的,因此不能满足实时处理的要求。下面本文将针对以上几点提出一种能较好地消除以上矛盾的快速预测搜索算法。

这里提出一种利用周围分块的空间相关性和前一帧的对应分块的时间相关性来重新确定起始搜索点,并以螺旋向外的方式搜寻最佳匹配点的算法。

#### (1) 视频对象内部变化较大时出现的问题

根据分块运动向量基于中心分布(center-biased)的特点,有人提出新三步法(N3SS)、四步搜索法(4SS)和梯度下降搜索法(BBGDS),它们有效减少了平均搜索点。但这些算法都是以分块所在位置为中心的一个窗口(大小为 $\pm 7$ 或 $\pm 15$ )内进行搜索,且都从该中心点开始搜索。所以如果视频对象内部的变化较快,此时全局最优点离中心点(即搜索起点)较远时,这些算法的平均搜索步数会较大,而且更有机会落入局部极值陷阱。三步法存在可能陷入局部极值的陷阱,原因在于它是从初始的搜索点进行搜索,而无法知道自己是否处在全局极值还是局部极值的单峰之内<sup>[49]</sup>。如果能够使其一开始就能以较大的概率处于全局极值的附近则既能够消除误判又能有效的缩减搜索的步数。此时需要先预测一个初始的运动矢量并以该矢量的顶点为初始搜索心,以快速接近全局最优点。

#### (2) 相邻分块的运动矢量的相关性

文献[50]的研究表明由于在现实世界的运动图像序列中, 一个运动物体会覆盖一帧内的多个分块, 所以空间相邻的分块因处在同一 VOP 内而使它们的运动较相似。又由于物体运动在时间上的连续性, 前后帧图像的时间相邻分块的运动也具有连续性。所以前面所述的初始搜索点此时则需利用时间和空间上的相关性来共同进行修正。

### (3) 运动的变化呈中心分布

实验表明, 真实世界里图像的分块运动是平滑的, 而且变化较慢。这使得运动矢量的变化呈现中心分布的规律。简单地在搜索区域内自上而下, 自左向右地搜索实际效率很低也不符合零搜索向量优先收敛的法则。因此搜索方式应自内向外呈向外散开的螺旋线, 这一点在三步法和象限法里已经得到了很好的应用。

## 4.2 改进算法的流程

根据上节分析, 本文提出的基于预测起始搜索点的运动图像分块匹配新算法步骤如下:

(1) 在实际的面向对象的视频编码中, 视频往往以对象的整体运动为主, 内部相对运动很小。所以当前帧的视频对象里的分块应将在前一帧的重建 VOP 里同一对象里的相应位置作为起始搜索点。而不能使用 MPEG-4 校验模型里采用的绝对坐标上的同一点为起始点。这里假设当前分块在上一重建 VOP 的对应位置的分块的绝对坐标的位置为  $(x_0, y_0)$ , 当前 VOP 与前一重建 VOP 在绝对坐标系里的距离为  $(l_x, l_y)$ 。

(2) 由于处于同一帧内, 因此我们可以认为当前帧内的相邻分块的运动特性应当被使用。一个分块周围可能被预测的分块数为 1 个-4 个。在时间分辨率较高时, 或者视频对象的运动较有规律时 (如简单的平移运动) 其空间相关性也能体现当前帧内部的分块的运动变化趋势, 所以前一重建帧内对应当前匹配块的分块以及其周围的存在的有效分块的运动矢量也具有参考价值。其可能的有效分块数为 1 个-9 个。但是在不同情况下, 两者之间的重要性是不断变化的: 在运动较没有规律的时候, 同帧的空间相关性更接近实际的运动趋势; 在同一 VOP 内部不同运动

区域的边界处的分块，因为其可用的已预测相邻分块与它的运动趋势并不一致而使其时间相关性更加重要。基于以上的原因，这里我们引入一个权系数  $A$ ，来调节两者之间的重要性的变化，削弱了其中之一的不规则变化带来的负面影响，计算公式如下：

$$(px_0, py_0) = \left[ A \cdot \sum_{i=1}^4 k_i (px_i, py_i) / \sum_{i=1}^4 k_i + (1-A) \sum_{j=1}^9 k_j (px_j, py_j) / \sum_{j=1}^9 k_j \right] \quad (4.1)$$

$$k_i = k_j = \begin{cases} 1, & \text{该块存在} \\ 0, & \text{该块不存在} \end{cases}, \quad 0 < A < 1$$

其中  $(px_0, py_0)$  是当前分块的初始搜索点的修正矢量， $(px_i, py_i)$  是当前 VOP 内的已被预测的相邻分块的绝对坐标下的运动矢量， $(px_j, py_j)$  是前一重建 VOP 内的对应的分块及其自身周围的有效相邻分块的绝对坐标下的运动矢量。其中  $k_i$ ， $k_j$  分别指示该块是否存在， $A$  为两部分的权重调节系数。

需要特别注意的是，那些在前一重建 VOP 里的对应分块及其周围的有效相邻分块的运动矢量  $(px_j, py_j)$  实际并不能代表当前分块的运动趋势，因此不可直接使用，而必须要经过修正才行。原理图见图 4.1：

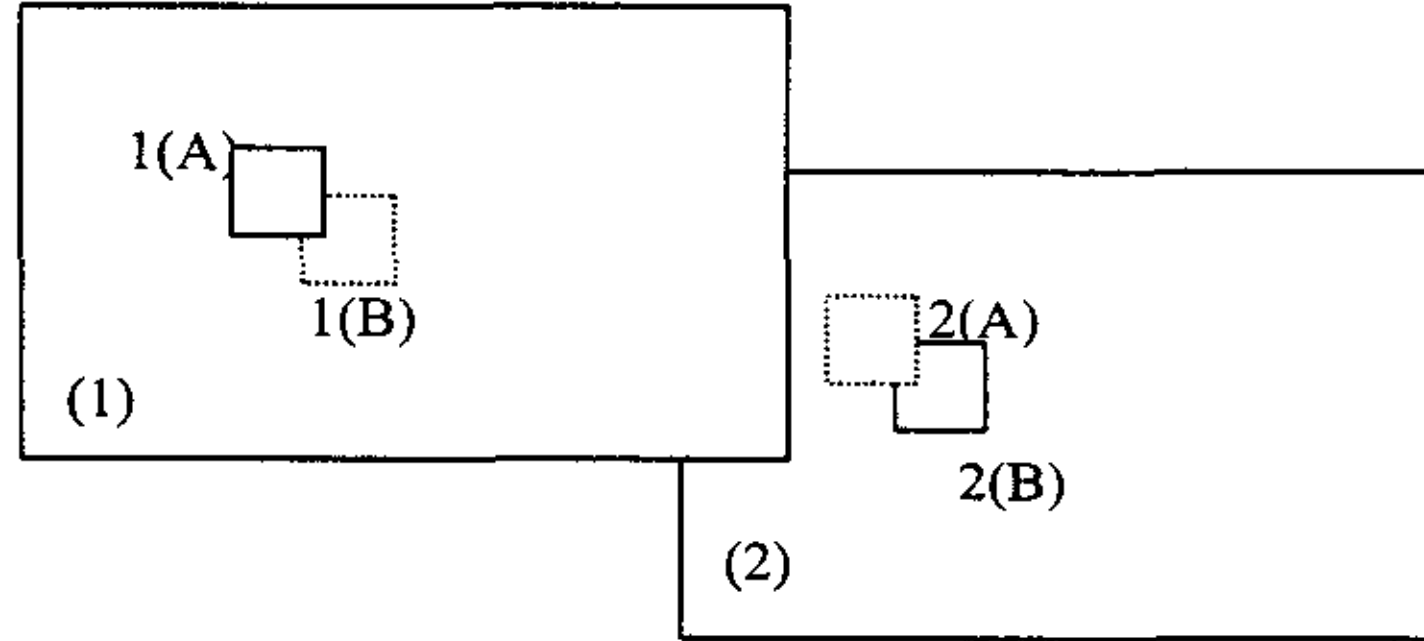


图 4.1 原理图

由上图可知一分块的运动轨迹呈现在前后连续的两帧内，如黑色线框所示，分别为(1)A，(2)B。它们在前后帧的对应位置分别用虚线框表示。显然要想体现(2)B的运动轨迹，需要参考的是(1)A，而非对应位置的(1)B。但是此时要求得的运动矢量正是(2)B到(1)A的位移矢量，是未知的，因此我们进行如下假设来解决这个

问题:

- (I) 一个分块的运动在短时间内是匀速直线运动;
- (II) 本帧内相邻分块的运动趋势在多数情况下可以近似认为与前一帧的上一个分块, 也就是(1)A, 的运动趋势一致。

基于以上两个假设, 我们认为可以用当前 VOP 内的已被预测的相邻分块的帧内相对(没有将 VOP 之间的位移考虑在内的)运动矢量  $(dpx_i, dpy_i)$  来反求出近似(1)A 的对应分块(1)A'。有如下公式:

$$(dpx_i, dpy_i) = (px_i, py_i) - (l_x, l_y) \quad (4.2)$$

$$(bx, by) = \left[ \sum_{i=1}^4 k_i (dpx_i, dpy_i) \right] / \sum_{i=1}^4 k_i \quad \text{其中 } k_i = \begin{cases} 1, & \text{该块存在} \\ 0, & \text{该块不存在} \end{cases} \quad (4.3)$$

为了反求(1)A'我们先要找到帧内相对反向运动矢量  $(bx, by)$ 。通过这个矢量我们可以在前一重建 VOP 内找到运动对应分块。但是考虑到这个反向运动矢量的长度并不为 16 或者 8 的整数倍, 因此还需要用就近原则来找到最适合的分块。设找到的适合分块的分块坐标(非像素坐标)为  $(i, j)$ , 块的大小为宏块长度  $16 \times 16$ , 公式如下:

$$i = \begin{cases} ((x_0 - bx) + 8) / 16, & \text{if } (x_0 - bx) \geq 0 \\ ((x_0 - bx) - 8) / 16, & \text{if } (x_0 - bx) < 0 \end{cases} \quad (4.4)$$

$$j = \begin{cases} ((y_0 - by) + 8) / 16, & \text{if } (y_0 - by) \geq 0 \\ ((y_0 - by) - 8) / 16, & \text{if } (y_0 - by) < 0 \end{cases} \quad (4.5)$$

此时再用前一重建 VOP 里的分块  $(i, j)$  和其周围的分块的运动矢量来做  $(px_0, py_0)$  计算式里的  $(px_j, py_j)$  即可。

由前所述, 在  $(px_0, py_0)$  计算式里因为空间相关性和时间相关性在不同的场合下的重要性是变化的, 因此 A 这个权值应该能自适应地调整, 从而使整个预测始终接近最佳。这里设实际搜索后的最佳运动矢量为  $(dx_0, dy_0)$ , 由下式求得下一分块将使用的新的 A 值。



$$A = \frac{\left[ (dx_0, dy_0) - \sum_{j=1}^9 K_j(px_j, py_j) / \sum_{j=1}^9 k_j \right]}{\left[ \sum_{i=1}^4 K_i(px_i, py_i) / \sum_{i=1}^4 k_i - \sum_{j=1}^9 K_j(px_j, py_j) / \sum_{j=1}^9 k_j \right]} \quad (4.6)$$

至此，可以用当前分块的初始搜索点的修正矢量  $(px_0, py_0)$  来修正 MPEG-4 校验模型里的绝对坐标下的初始搜索点，即与当前分块是同一绝对坐标的点。得到新的起始搜索点的绝对值坐标为：

$$(x_1, y_1) = (x_0, y_0) - (l_x, l_y) + (px_0, py_0) \quad (4.7)$$

为了避免出现误差累积，即由于一开始的搜索起点可能离当前编码分块较远，而导致最后所得的预测矢量也离该块较远甚至更远，最后极大的偏离原分块(在试验中证实了该现象几乎肯定会发生)，这里设置了一个搜索限制窗口限制搜索起点的范围以避免该情况发生。如图 4.2。

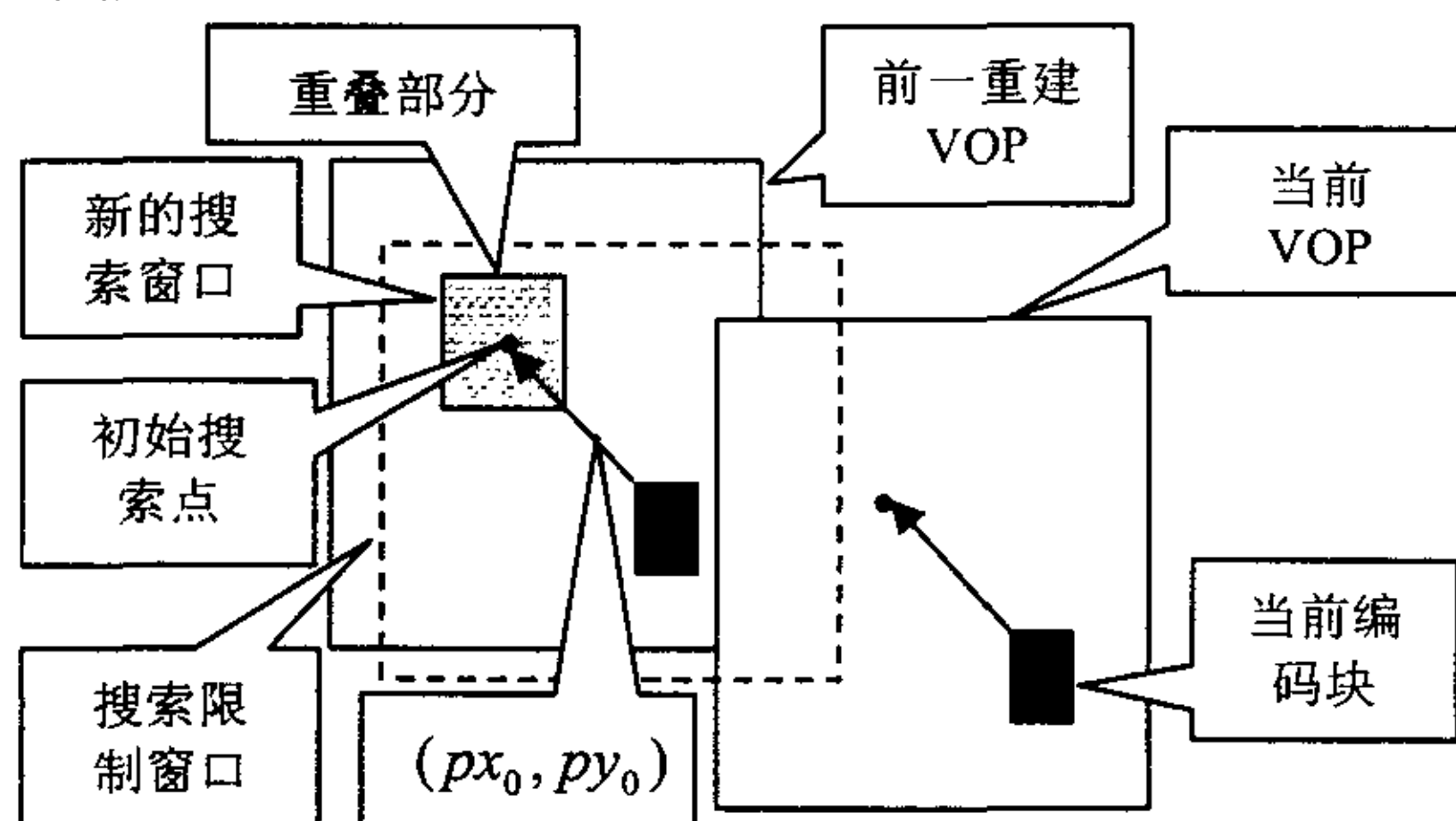


图 4.2 起始搜索点的示意图

(3)确定搜索起始点后，需要做的就是有一个较好的搜索方式。因为确定了起始搜索点，此时可以认为很接近最终的全局最优点，也就是处于全局极值点的单峰之内，无需再考虑多极值的问题。但是这里不采用三步法（适用于单极值得区域搜索），其原因是：

- (I)其优点在于在大的范围内搜索时可以快速收敛，而此时的搜索窗已经很小，无法体现其优点；
- (II)仍然不符合零向量优先原则；
- (III) 其算法较复杂不适合并行处理<sup>[51]</sup>。



这里我们采用一种螺旋向外的全搜索方式。全搜索虽然计算量大，但是具有步长固定，易于并行处理等优点。特别是易于并行处理这一点是全搜索算法最大的

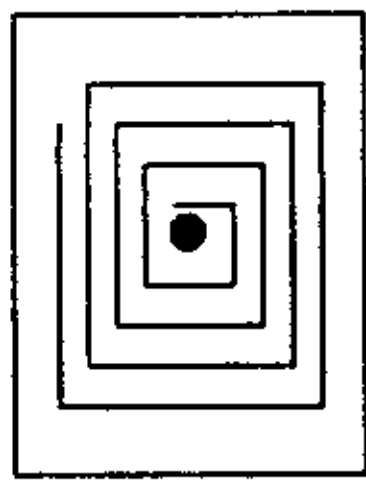


图 4.3 搜索示意图

的优点。快速算法虽然削减了计算量，但是却使程序复杂。正是这些原因限制了快速算法的硬件化。下面分别介绍这种搜索算法。

此搜索方式是以前面提到的修正后的新的起始搜索点为中心，螺旋散开的方式来进行搜索，过程见图 4.3。此时需要一个方向变量，其定义式为 `enum eDirect {0,1,2,3}`。此变量为枚举类型用于指示上下左右四个方向，当搜索完一圈时，自动向外扩展一圈。

为了更快的完成搜索且实现零向量优先收敛的原则，我们设置了绝对误差总和 SAD 的门限值。选择一个好的门限值可以始终保持较好的“质量/计算量”之比。通过对一般视频的研究，我们认为门限值取上一帧的分块绝对误差总和的平均值较为合理，但是这里就引出两个问题：

- (I) 当场景切换等导致分块绝对误差总和的平均值有较大变化的情况时如何使下一帧的门限值自动地进行调整；
- (II) 当分块绝对误差总和的平均值由小变到大时，会有多数分块未达到门限值就停止搜索，根据此信息可以根据其所占百分比进行调整。但对于分块绝对误差总和的平均值由大到小时，特别是当场景切换导致绝大部分甚至全部分块都是达到门限而停止搜索时，我们无法知道实际理想门限值降到何种地步，因此需要知道如何合理的降低门限以尽快的向理想门限值收敛。

基于以上两个问题的思考，这里提出一种门限值自适应算法，可以很好的解决以上问题，描述如下：

- (I) 首先假设门限值恒定不变来进行搜索，同时统计未触及门限值就停止搜索的分块的数量以及其各自的绝对误差总和；
- (II) 当前帧搜索完毕后，设当前帧的门限值为  $K_0$ ，上一帧的绝对误差总和的最大值为  $K_{\max}$ ，未触及门限值就停止搜索的分块的数量占总块数的百分比为  $x$ ，其绝对误差总和均值（注意这里的均值不是全局分块的绝对误差总和均值）为  $K_1$ ，则新的供下一帧使用的门限值为  $K_2$ ，公式如下：

$$K_2 = K_0 + (K_1 - K_0) \cdot \frac{(x - 50\%) \cdot \frac{x}{2}}{x} \quad \text{其中 } x > 20\% \quad (4.8)$$

$$K_2 = K_0 - (K_{\max} - K_0) \quad \text{其中 } x < 20\% \quad (4.9)$$

需要注意的是为了简化运算我们假设其分块的绝对误差总和的数值分布是均匀的。而对  $x$  的范围进行限制是考虑到因当前帧中分块的实际绝对误差总和均值太小而使未触及门限就停止搜索的分块数量很少，而且往往都是特殊分块，其绝对误差总和不具有代表性，另外这时作为分母的  $x$  因为很小而会导致本来就并不可信的修正值进一步变大，无法很好的向实际门限值快速收敛，因此设置了  $x$  的范围。

当  $x < 20\%$  时，即用上一帧中分块的绝对误差总和的“最小值”来近似实际门限值。这里最小值打上引号是因为实际最小值是无法得到的，因此假设绝对误差总和是对称且均匀分布，故用最大值来反求最小值。

在 MPEG-4 算法中当遇到最小绝对差值和相等的情况时，还需进行两点相对于初始点的距离大小判断以决定取舍。上述算法可以省去此判断。

## 4.3 改进算法实验结果及分析

实验采用 MPEG-4 验证模型的全搜索算法和改进算法分别对 Susie.qcif, 和 Foreman.qcif 进行运动估计算法的编码处理。Susie.qcif 是近距离的人脸图像序列；Foreman.qcif 是场景有大幅度但平滑变化的图像序列。

在 MPEG-4 算法里，运动矢量的搜索窗口为  $16 \times 16$ ，而改进算法的搜索窗口分别设为  $6 \times 6$  和  $4 \times 4$ ，其搜索限制窗口为  $16 \times 16$ ，其结果如下：

表 4.1 改进算法和全搜索的实验结果

指标\算法	改进算法 ( $6 \times 6$ )		改进算法 ( $4 \times 4$ )		全搜索	
	Foreman	Susie	Foreman	Susie	Foreman	Susie
PSNR_Y	31.698729	35.426498	31.286200	35.269375	31.741503	35.542427
PSNR_U	37.248287	42.514130	36.989876	42.518623	37.233387	42.652977
PSNR_V	38.322166	41.819191	38.018028	41.774193	38.256107	41.908924
压缩比	49.0	52.6	48.8	52.1	49.3	52.7

与第三章里的三步法和改进的正交共轭法的实验结果比较我们得到以下数据:

(1)Foreman 的 PSNR\_Y 曲线图

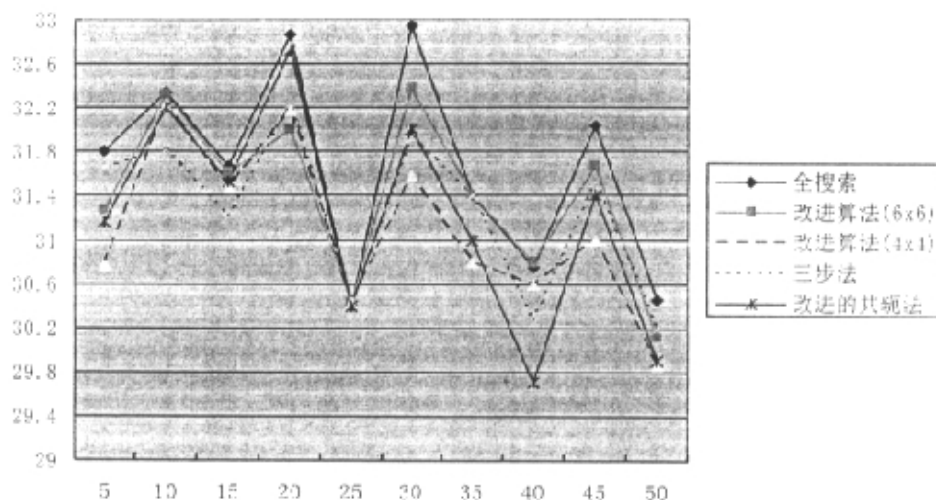


图 4.4 Foreman 的 PSNR\_Y 曲线图

(2)Susie 的 PSNR\_Y 曲线图

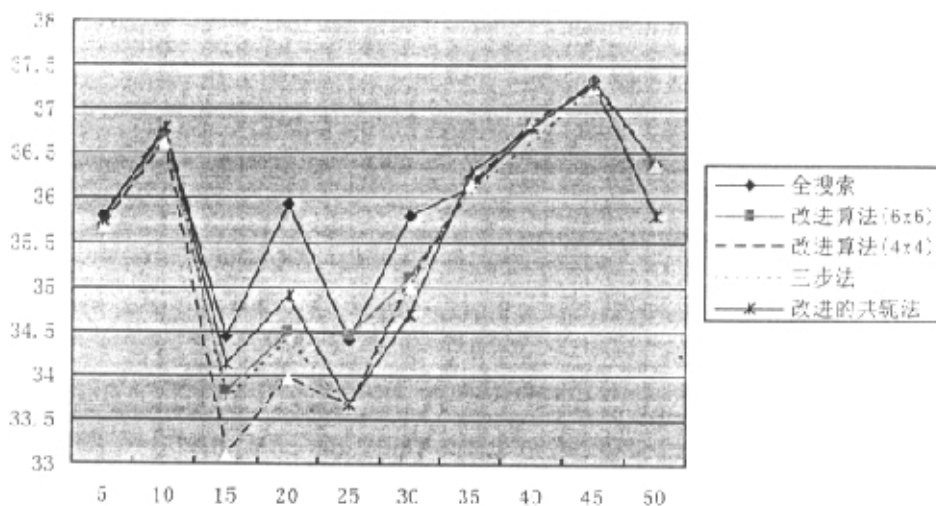


图 4.5 Susie 的 PSNR\_Y 曲线图

由上所提供的数据和图表我们可以看出改进的搜索算法的实际效果在搜索窗口为  $6 \times 6$  比三步法和正交共轭法更好；搜索窗口为  $4 \times 4$  时要差一些但是差距并不大，而此时的搜索步数较少提高了速度。

## 4.4 小结

本章提出一种利用空间—时间相关性的预测初始搜索点的改进算法并进行了软件试验。实验结果表明该算法能在较少的搜索步骤下获得较好的 PSNR 值，同时还一定程度上消除了多极值问题的存在，并且比较适合并行处理利于用硬件实现。

## 5 结束语

本文对视频编码中的运动估计的关键技术及其实现进行了深入的研究,提出了一种改进的正交共轭法。并针对原有算法的缺陷,提出了一种新的基于预测起始搜索点的快速搜索方法,经过试验仿真可以看出该算法较传统算法的效果要好。

### 5.1 本文的工作总结

(1) 详细分析形状编码、纹理编码、运动估计和补偿、可伸缩性编码等关键算法。着重探讨了几种运动估计常见的编码方法,包括三步法、共轭方向法、象限法以及统计法等。

(2) 根据视频运动在空间上相邻分块具有相关性和在时间上对应分块运动具有连续性的特点,提出了基于预测起始搜索点的改进算法。试验表明该算法在保持较高的信噪比的同时较大提高了搜索速度。

(3) 同时提出了一种门限设置方法,可以较好的适应图像变化所导致的门限值变化。其算法简单,且遵循零向量优先收敛的原则,相对于常数门限值有着较好的适应性。

### 5.2 有待继续的研究

时间冗余的消除不仅在时间域可以进行,在频率域上也可进行。本文仅对时间域做了研究,而对其他一些技术本文未及讨论和实现。就本算法而言对于门限值的选取还有其他可取方案,如分层判决的搜索方法等。

在分块匹配方法的基础上,尽管采用了不同的方法去掉了视频数据冗余,各个结果也略有不同,但都相差不大。因为把视频信号看成是一个或多个平稳随机的过程,无论采用什么方法,都要接受香农编码定理和编码逆定理的约束。本文提到的算法也都是以绝对误差总和的大小和信噪比来衡量视频算法的优劣。但是如果充分利用人的视觉特性,改变失真的度量方法,不再用上述的衡量标准,是有可能大幅度提高视频压缩比的。因此对视频运动估计应考虑从新的角度和计算域来进行。

## 致 谢

本文是在朱光喜教授的悉心指导下完成的。

朱光喜教授学识渊博、思想敏锐、平易近人，正是他的关心、教诲和帮助激励着我完成了此文的研究工作。作为作者的指导老师，本论文倾注了朱光喜教授的大量心血，没有他的悉心指导，本论文不可能顺利完成。朱光喜教授在学术上提倡开放创新，严格论证；在工作上一丝不苟，兢兢业业。这三年和朱光喜教授相处的时间，使我终身难忘，受益匪浅。在此表示最诚挚的感谢！

这里还要感谢本教研室的刘文予教授和张江山老师，在我三年的研究生学习过程中对我的学习和课题研究给予了不遗余力的支持和帮助，令我十分感动，在这里表示深深的谢意。

还有同教研室的朱焰赤，王飞，赵娟，陈汉强，邓勇强，刘玮，厉小龙，邱锦波，王德红等同学，我们一起度过了研究生三年的学习过程。与大家的相处使我受益匪浅，感谢他们的无私帮助。

还需要特别感谢的是提高班 96 级的全体同学，在这与众不同的六年大学生涯中，大家朝夕相处，结下了深厚的感情。无论酸甜苦辣，我都很高兴能和大家一起度过这段难忘的生活，这将永远是我一生中最珍贵的回忆。

论文评阅和评议专家在百忙之中抽出时间为我评审论文，在此特向他们表示衷心感谢。

衷心感谢我敬爱的父母、姐姐和我的亲人，他们对我的殷切期望、始终如一的无私关爱使我不断前行，也使我深刻地体会到亲情的可贵。你们的支持给予了我最大的动力和勇气！

## 参考文献

- [1] 高文. 多媒体数据压缩技术. 北京市:电子工业出版社, 1994. 11~12
- [2] T. Ebrahimi, M. Kunt. Visual Data Compression for Multimedia Application. In: Declare A. ed. *Proceedings of the IEEE*. CA Irven. 1998. CA: IEEE press, 1998. 1109~1121
- [3] T. Ebrahimi, E. Reusens, W. Li. New Trends in Very Low Bitrate Video Coding. In: Alan Patrick ed. *Proceedings of the IEEE*. CA Los Angels. 1995. CA: IEEE press, 1995. 877~891
- [4] 荆仁杰等. 计算机图像处理. 浙江: 浙江大学出版社, 1992. 43~46
- [5] Kunt M., Ikonomopoulos A., Kocher M.. Second generation image coding techniques. *IEEE Trans Communication*, 1985, 73(4): 549~575
- [6] Nasrabadi N. M., King R. A.. Image coding using vector quantization: review. *IEEE Trans Communication*, 1988, 36(8): 957~971
- [7] Westerink P. H., et al. Subband coding of images using vector quantization. *IEEE Trans Commun*, 1998, 36(6): 713~719
- [8] Woods J. W., O'Neil S. D.. Subband coding of image. *IEEE Trans ASSP*, 1986, 34(5): 1278~1288
- [9] De Vore R. A., Jawerth B., Lucier B. J. Image compression through wavelet transform coding. *IEEE Trans Information Theory*, 1992, 38(2): 719~745
- [10] Sokora T.. The MPEG-4 video standard verification model. *IEEE Trans CSVT*, 1997, 7(1): 19~31
- [11] Andre Kaup. Object-Based Texture Coding of Moving Video in MPEG-4. *IEEE Transactions on Circuits & Systems for Video Technology*, 1999, 9(1): 545~571
- [12] Willemin P., Reed T., Kunt M.. Image Sequence coding by split and merge. *IEEE Trans Communication*, 1991, 39(12): 1845~1855
- [13] 李海波. 模型基图像编码. *通信学报*, 1993, 14(2): 69~76
- [14] B. G. Haskell, P. G. Howard, et al. Image and Video Coding-Emerging Standards and Beyond. *IEEE Trans on Circuits and Systems for Video Technology*, 1998, 8(7): 814~835



- [15] K. Aizawa, T. S. Huang. Model-based Image Coding: Advanced Video Coding Techniques for Very Low Bit-rate Application. IEEE Trans on Circuits and Systems for Video Technology, 1998, 8(2): 259~269
- [16] W. J. Welsh. Model-based Coding of Moving Images at Very Low Bit Rates. In: John Kern ed. Proceeding of International Picture Coding Symposium. Stockholm, Sweden. 1987. Stockholm, Sweden: PCA press, 1987. 819~846
- [17] 吴敏金. 分形与图像压缩编码. 通信学报, 1993, 14(2): 78-83
- [18] 张明德, 王永东. 视频会议系统原理与应用. 北京: 北京希望电子出版社, 1999. 134
- [19] Mandelbrot B. B. The fractal geometry of nature. IEEE Trans Communication, 1982, 5(4): 125~141
- [20] Rijkse K. H.263: Video coding for very low bit rate communication. IEEE Communication Magazine, 1996, 34(12): 42~45
- [21] Rijkse K. ITU Standardlization of very low bitrate video coding algorithms. Singal Processing: Image Communication, 1995, 7(4~6): 553~565
- [22] 高剑林等. H.263+中新的帧类型研究. 电视技术, 1999, 28(5): 8~9
- [23] Chiariglione L.. MPEG past and future. 通信学报, 1995, 16(5): 46~59
- [24] Chiariglione L.. MPEG and multimedia communications. IEEE Trans CSVT, 1997, 7(1): 5~18
- [25] Chiariglione L.. Impact of MPEG standards on multimedia industry. IEEE Trans CSVT, 1998, 8(6): 1222~1227
- [26] 丁宗毫, 曾国良. 图像压缩标准 MPEG-2 的图像格式. 电视技术, 1995, 5: 5-11
- [27] Leonardo Chiariglione. The MPEG-4 Standard. Journal of Chi. Inst. Com., 1998, 19(9): 198~223
- [28] Epereira. MPEG-4: A new challenge for the representation of Audio Visual Information. Picture Coding Symposium, 1996, 76(2): 557~582
- [29] Sokora T.. The MPEG-4 video standard verification model. IEEE Trans CSVT, 1997, 7(1): 19~31
- [30] ISO/IEC JTC1/SC29/WG11 N2321. MPEG-4 application document. Dublin, 1998, 72~73



- [31] ISO/IEC JTC1/SC29/WG11 W2079. Overview of MPEG-4 functionalities supported in MPEG-4 version 2, 3. 1998, 114~117
- [32] ISO/IEC JTC1/SC29/WG11 N2995. MPEG-4 overview, Melbourne Version. Melbourne. 1999, 103~104
- [33] Rob Koenen. MPEG-4 overview ( Tokyo Version), ISO/IEC JTC1/SC29/WG11 N2196. 1998, 98~102
- [34] Rschaeffer, Tsikora: Digital video coding standards and their role in video communications. Proceeding of the IEEE, 1995, 83(7): 1022~1056
- [35] 朱秀昌, 宋建新. 多媒体网络通信技术的应用. 北京: 电子工业出版社, 1998. 56~58
- [36] Francois Fluckiger. 网络多媒体开发与应用. 冯博琴. 北京: 机械工业出版社, 1997. 312~317
- [37] Coding of audio-visual objects: Systems. ISO/IEC 14496-1:final committee draft. ISO/IEC JTC1/SC29/WG11 N2201. 1998, 209~211
- [38] Y. Q. Zhang, F. Pereira, T. Sikora. Special issue on MPEG-4. IEEE Trans Circuit System Video Technology, 1997, 7(2): 256~271
- [39] Avaro, P. Chou, A. Eleftheriadis, et al. The MPEG-4 System and Description Languages: A Way ahead in Audio Visual Information Representation, Signal Process. Image Commun., 1999, 9(4): 189~204
- [40] K. Sonokov. MPEG-4 simulation software of audio-visual objects: Delivery multimedia integration framework. IEEE Trans. ALBTY, 1998, 34(3): 510~534
- [41] J. Y. A. Wang, E. H. Adelson. Representing moving images with layers. IEEE Trans. Image Processing, 1996, 19(5), 137~152
- [42] Touradj Ebrahimi. MPEG-4 Video Verification Model: A video encoding/decoding algorithm based on content representation. Signal Processing Laboratory. Swiss Federal Institute of Technology. 1999, 4(7): 371~380
- [43] J. Ostermann. Methodologies Used for Evaluation of Video Tools and Algorithms in MPEG-4. Signal Process, 1997, 9(12): 47~61
- [44] S. Li, W. Li. Shape adaptive Discrete Wavelet Transform for Coding Arbitrarily Shaped Texture. IEEE Trans. Image Processing. 1997, 30(24): 12~14,
- [45] 汪朝阳, 周敬利, 余胜生. 基于可视电话系统的向量预测效率分析. 小型微

- 型计算机系统, 1999, 20(5): 383~387
- [46] 周俐利, 勒中鑫, 蔡国权. 块运动估计的快速搜索算法探讨. 数字视频, 1999, 206(12): 7~10
- [47] 张春田. 视频信号压缩编码第五讲---运动估计在序列图像编码中的应用. 广播与电视技术, 1995, 22(3): 58~63
- [48] Bing Zeng, Renxiang Li, Ming L. Liu. Optimization of fast block motion estimation algorithms. IEEE Trans. on CSVT, 1997, 7(6): 833~844
- [49] 黄晓宇. MPEG-4 极低位速率视频技术的研究. 吉林化工学院学报, 1999, 16(4): 147~158
- [50] Hsieh C., Et al. Motion Estimation Algorithm Using Inter-block Correlation. Electronics Letters, 1990, 26(5): 276~277
- [51] 周海城. 图像压缩编码中运动补偿快速算法的探讨. 电视技术, 1995, 6(3): 7~10

## 附录 1 攻读学位期间发表的论文

1. 袁炜, 朱光喜. 视频对象编码中初始搜索点的预测算法研究. 华中科技大学学报, 2001 年 11 月, 第 29 卷, 第 11 期: 58~60