

浙江大学
硕士学位论文
H. 264视频编码快速算法研究
姓名：杨黎波
申请学位级别：硕士
专业：信号与信息处理
指导教师：唐慧明
20050201

## 摘 要

H.264 是最新的视频压缩编码国际标准,与以往的标准如 MPEG-1/2/4 及 H.261/H.263 相比, H.264 的编码效率有了显著的提高,在可以预见的将来必将得到广泛的应用。但 H.264 的复杂度极高,不利于实时运用。本文就着眼于 H.264 的优化,提出了几种有效的快速算法。

本文首先介绍了视频压缩编码的理论及国际标准的的发展,着重介绍了 H.264 所采用的新的编码技术,并与之前标准的编码效率进行了比较。在分析了 H.264 编码器各部分的复杂度后,我们确定优化的模块为帧间预测(包括整像素运动预测和亚像素运动预测)及帧内预测。

对于整像素运动预测,本文提出了一种提前中止算法——多种块大小的最佳运动检测 VBBMD(Variable Block-size Best Motion Detection)。该算法通过计算一个搜索点来预测最佳运动矢量,因而能提前中止一部分运动搜索过程,跳过与之相对应的计算。与 H.264 标准所采纳的快速算法 UMHexagonS 相比,所提算法能进一步减少 76%~94% 的整像素搜索点,而平均 PSNR 下降只有 0.06dB。

对于亚像素运动预测,本文提出了快速算法——基于预测的方向亚像素搜索 PDFPS(Prediction-based Directional Fractional Pixel Search)。该算法采用两种预测方式来更准确地预测亚像素运动矢量,直接从预测位置或预测方向开始搜索,并采用 1/4 像素步长的菱形模式来进行细化搜索。与全搜索及 H.264 所采纳的快速亚像素搜索算法 CBFPS 相比,所提出的算法分别能减少最多 84% 和 74% 的亚像素搜索点,而 PSNR 平均只下降了 0.01dB。

对于帧内预测,本文提出了基于模式预测的快速帧内预测算法。该算法利用视频序列时域、空域及不同块大小之间的相关性来预测当前块的模式,并结合提前中止方法来减少帧内预测所需计算的模式数量。与全搜索相比,该快速算法能降低 50%~75% 的帧内预测计算量,同时保持编码视频的图像质量。

采用以上快速算法后,能极大地提高 H.264 编码器的速度,是对 H.264 的实时应用所进行的有益探索。

**关键词:** 视频编码, H.264/JVT, 快速算法, 整像素运动预测, 亚像素运动预测, 帧内预测

## Abstract

H.264 is the latest video coding international standard. It has achieved a significant improvement in coding efficiency compared to previous standards, such as MPEG-1/2/4 and H.261/H.263. Therefore H.264 will be widely used in the near future. However, its complexity is too high to be applied in real-time applications.

In this paper, we first introduced the video coding theory and international standards. We focused on analyzing the new coding features adopted by H.264 and compared it with the previous standards. After calculating the complexity of each module in H.264 encoder, we preferred to optimize the modules: inter prediction (integer pixel motion estimation, fractional pixel motion estimation) and intra prediction.

First, we proposed an early termination algorithm VBBMD (Variable Block-size Best Motion Detection) for integer pixel motion search. It predicts the best motion vector by examining only one search point. Therefore some of the motion searches can be stopped early, then a large number of search points can be skipped. Compared with the fast algorithm UMHexagon, the proposed method can further reduce 76% to 94% of the integer-pixel search points, while the PSNR degradation is only 0.06dB on average.

Secondly, we proposed a fast fractional pixel motion search algorithm PDFPS (Prediction-based Directional Fractional Pixel Search). It takes advantage of the more accurate motion vector prediction by applying two prediction methods and starts the search in the prediction location or prediction direction, followed by the refinement with 1/4-pel diamond search. Compared with full search and the fast algorithm CBFPS adopted by H.264, the proposed method reduces up to 84% and 74% of sub-pel search points, while the average quality degradation is only 0.01dB.

Finally, we proposed a fast intra prediction based on mode prediction. It takes advantage of the strong correlations in temporal domain, spatial domain and different block-sizes to predict the best mode of the current block. An early termination mechanism is further incorporated to reduce the candidate modes. Compared with the full search, our method can reduce the computation of intra prediction by 50%~75%, while the image quality keeps the same.

After adopting the above fast algorithms, we can speed up the H.264 encoder greatly. The work presented in this paper is the useful research for the real-time application of H.264.

**Key words:** Video coding, H.264/JVT, fast algorithm, integer pixel motion estimation, fractional pixel motion estimation, intra prediction

# 第一章 概述

随着数字技术的迅猛发展，人类社会已经步入了数字时代，各种技术层出不穷，极大地改变了人类的生活。多媒体技术就是其中一项重要技术，它利用数字编码来描述和表达图像、声音等各种媒体信息，带给人们极大的方便。

在各种媒体信息中，图像占据着尤其重要的位置。科学实验表明，人类从外界获取的知识之中有80%都是通过视觉感知获取的。然而，数字图像中包含的数据量十分庞大，如考虑中分辨率(640x480)下，全屏幕显示 (Full Screen)，真彩色 (True Color 24位)，全动作(Full Motion, 25-30帧/秒)的图像序列，播放1秒钟的视频画面数据量为 $640 \times 480 \times 300 = 27,648,000$ 字节，相当于存贮一千多万个汉字所占用的空间。如此庞大的数据量，给图像的传输、存贮以及读出造成了难以克服的困难。因此，如何有效地对图像进行压缩一直是研究的重点和热点，图像压缩编码日益成为一个越来越活跃的研究领域。图像编码技术的发展和广泛应用也促进了许多有关国际标准的制定。

在视频压缩标准中，最新制定的H.264能取得最高的编码效率，但也带来了极大的运算复杂度。为了进行实时的运用，就必须对现有的H.264编解码器进行优化。各种编码算法优化能在基本保持图像质量的基础上，显著地降低运算复杂度，因而是优化的一个主要手段。

本章首先概述了视频压缩编码的原理及采用的技术，然后介绍了视频编码国际标准的发展，最后提出了本文的主要工作。

## 1.1 视频压缩编码技术

多媒体视频信号可以压缩的主要根据为：一、视频信号上存在大量的冗余度并且这种冗余度在编解码后可以无失真地恢复；二、可以利用人的视觉特性，在图像变化不被觉察的条件下减少量化信号的灰度级，以一定的客观失真换取数据压缩(2)[1]。

从信息论观点来看，视频作为一个信源，描述信源的数据是信息量(信源熵)和信息冗余量之和。数据压缩实质上是减少冗余量，冗余量的减少可以减少数据量而不减少信源的信息量。多媒体视频信号的冗余度存在于结构和统计两方面。

统计冗余就是在视频信号中,各符号出现概率不等造成的冗余。结构上的冗余度表现为很强的空间(帧内)、和时间(帧间)相关性。

一般情况下画面的大部分区域信号变化缓慢,尤其是背景部分几乎不变。因此,视频信号在相邻像素间、相邻行间、相邻帧间存在强相关性,这种相关性就表现为空间冗余和时间冗余。

人眼对图像的细节分辨率、运动分辨率和对比度分辨率的感觉都有一定的界限。致使对图像处理时引入的失真不易察觉,仍会认为图像是完好的或足够好的。因此可以在满足对图像质量一定的要求的前提下,减少表示信号的精度,实现数据压缩。

视频压缩主要就是要消除数据中的冗余,因此基本的压缩技术按照作用主要分三类:去除空间冗余,通常用变换编码压缩;去除时间冗余,通常用运动补偿压缩;去除统计冗余,用信息熵编码来压缩。另外还有其他一些从提高图像视觉质量方面考虑的方法,比如量化、滤波等技术,前两者常结合成混合编码。除了以上几类基本技术外,近些年又出现了基于模型的编码技术[2][3],基于对象(Object Based)的编码技术[4],可分级的编码技术[8]等一系列技术。模型编码的关键是对特定的图像建立模型,并根据这个模型确定图像中景物的特征参数,如运动参数、形状参数等。基于模型的编码方法,适合于特定视频内容的编码,如视频压缩标准MPEG-4中引入的人脸编码部分[12],它对运动的人脸编码效果很好。基于对象(Object Based)的编码技术是MPEG-4的最大特色,它把对象作为视频压缩的基本单位,可以提供给用户更多的交互性功能,就使用的编码技术而言,主要是任意形状的图像块的压缩,比如利用空间分辨率较好的小波变换进行基于对象的视频压缩[5],或者使用形状自适应的DCT变换[6][7]等等。可分级的编码技术的基本思想是,我们知道,视频的编解码需要一定的资源的支持,比如计算资源、带宽资源、采集回放资源等,当用户拥有的资源达不到完全实现编码或解码的时候,该技术可以使用户同样实现编码或解码的目的,只是服务的质量差一些[9]。按照可分级的资源的不同,可以分为:采样率可分级、复杂度可分级、带宽可分级等等。以上三种方法中,模型编码由于不可能为自然界中所有物体建模,所以并不适用于通用的视频压缩。后两种技术都属于功能性的技术,与原有的编码技术相结合[10][11],可以用来提供给用户更多的功能,而它们的基本压缩技



术仍然是前面讲的三种压缩技术：变换、预测和熵编码。

### 1.1.1 冗余的消除

如前所述，视频中主要存在3种冗余：空间冗余、时间冗余和统计冗余，下面分别介绍这些冗余的消除办法。

#### 1.1.1.1 空间冗余的消除

空间冗余一般用变换编码消除。变换编码的基本依据是像素之间有很强的相关性，希望通过变换，减少相关性。Karhunen-Loeve(K-L)变换是均方误差(MSE)意义下的最佳变换，但计算复杂，无快速算法，难于实际应用。Ahmed, Natarajan和Rao提出的Cosine变换[13]，其性能接近理论上的最优变换(K-L变换)。它源自FT，但它的实对称性确保不会产生伪高频现象，且有快速算法，因而最终被广泛采用。DCT变换是通过将像素矩阵变换到系数空间(频域)上进行处理而实现的。最近几年来，小波变换用于图像压缩获得了较好的效果[14][15]，JPEG2000图像压缩标准[16][17]就采用了小波变换。小波变换在图像压缩中优于DCT，很大程度上是因为小波变换有更有效的快速算法，可以在整幅图像上作变换然后进行压缩[18]，而DCT变换则只能分块进行，也就是说，小波则可以在全局去除冗余，而DCT变换只能去除块内的冗余。

小波变换用于视频压缩时，由于预测图像有较强的块效应，所以一般要对预测图像进行处理(比如滤波去除块效应)，或者使用较为特殊的预测方法(比如块重叠的运动估计)[19]，但是这样既增加了运算复杂度，又可能会降低预测图像的信息量。所以DCT仍然广泛用于视频压缩系统。除了以上的变换技术外，空间冗余的消除还有其他一些方法。

多方向空间预测技术实际上是一种DPCM，它的预测有多种方向，按照图像本身的特点选择一个最佳的预测方向，由已经解码的相邻像素预测当前块，它可以有效地去除块间的冗余。多方向空间预测技术与DCT变换相结合，可以弥补DCT变换只能去除块内冗余的缺点，获得较高的编码性能，比如H.264中的I帧编码部分就采用了这样的配置[20][21]，其编码性能与采用小波变换的JPEG2000不

相上下[22]。

匹配逼近(matching-pursuit) [23]也常被用于视频压缩, 它用基图像的加权和逐渐逼近原始图像或残差图像, 以代替变换技术, 在低码率时可以获得高于DCT的编码性能[24], 但缺点是复杂度较高。

### 1.1.1.2 时间冗余的消除

时间冗余的消除主要依靠运动补偿预测编码技术。运动补偿预测编码的基本思想是把一幅动态图像看成是由静态部分和运动部分叠加而成。静态部分可以重复使用上一帧的数据, 而对运动部分则设法确定其位移量来帮助运动部分的预测, 即参考帧进行运动补偿之后再作为当前帧的预测。

对运动矢量MV的估值方法有多种, 例如: 块匹配法、梯度法、傅立叶变换法等。最简单的方法就是全搜索块匹配法, 即把所有可能位置都比较一遍, 但是这样做复杂度太高, 如何用较少的计算量来达到较好的运动估计效果, 正是快速运动估计所研究的内容, 目前, 快速运动估计也是视频压缩领域一大研究热点[25]。

由于受到硬件速度的限制, 早期的视频压缩标准, 在运动估计部分多采用了固定的块大小和单参考帧(B帧除外), 比如MPEG-1、MPEG-2等。而较新的视频压缩标准, 比如MPEG-4和H.264则采用了多参考帧和多种尺寸块的运动估计和补偿[26][27], 以获得更高的编码性能[28]。

亚像素运动估计[29]由于可以更准确地描述物体的运动, 所以被MPEG-2、H.263等多个视频压缩标准所采用, 目前最先进的H.264标准中, 使用了1/4像素的运动估计。

前面提到过的重叠块(Over-block)运动补偿的特点是, 相邻的运动补偿的块是重叠的, 最终的像素值由多个预测块内的像素通过加权平均或其他方法计算出来。由于产生的预测像素是加权平均出来的, 所以会丢失一些细节信息, 因此主要用于低码率的视频压缩中。

全局运动补偿(Global-Motion-Compensation)为整个帧给出一种运动模式, 可以非常有效地描述整个帧的运动情况, 适合镜头移动、缩放和旋转等情况, 通常与基于块的运动补偿相结合使用, H.263和MPEG-4第二版中都采用了这种技术

[30]。

3D小波压缩算法用三维小波变换来压缩视频信号(空间两维加时间), 熵编码部分使用扩展到三维的3DSPIHT算法[31], 由于没有充分利用运动信息, 编码性能一般低于运动补偿加DCT变换的混合编码框架。文献[32]中的算法对第三维(即时间)的变换进行了特殊处理, 用类似运动估计的方法把相邻的两个帧变换成和图像和差图像, 然后递归对得到的和图像做变换得到更低频的和图像, 其他两维则使用普通的小波变换, 获得了高于H.263+的编码性能。3D小波压缩算法的主要缺点是计算和存储复杂度较高。

Sprite技术[33]从视频的每一帧中提取出背景图像, 拼成一个较大的完整的场景, 然后分别编码场景和前景图像, 可以获得较高的压缩比, 但是由于必须先生成场景, 所以编码延迟较大。Sprite技术也是MPEG-4的一大特色[34]。

### 1.1.1.3 统计冗余的消除

统计冗余用信息熵编码来消除。信息熵编码的基本方法主要有变长编码(VLC)、行程编码(Run-Length)和算术编码。

DCT变换系数由于量化后高频分量较多, 所以最常用的压缩方法是Run-length加变长编码的方法, 由于复杂度低, 实现简单, 我们熟悉的JPEG、MPEG都使用了这样的方法。这种方法的缺点是, 假定了符号的概率是相对固定的, 但实际上不同的块之间差别很大, 同一块内不同位置的系数的分布规律差别也很大。在实际应用中, 不可能对全部大量的信息进行概率统计(时间较长), 并且很多符号的概率本身就是变化的, 所以各种低复杂度的自适应方法成为目前熵编码方面的研究重点。位平面编码方法把系数分层编码, 由于为每个块给出一个层数, 可以解决块间差别的问题, 使编码器在一定程度上自适应于每个块, 因此编码性能高于Run-length加变长编码的方法, 但是把系数分层编码复杂度较高, 同时也会损失一定的编码性能。算术编码[35]可以依据出现的符号而相应地改变各符号的概率模型, 因此可以很好地适应系数的压缩, 但是计算复杂度较高, 文献[36]中给出的自适应二进制算术编码器(CABAC), 用查表的方法得出下一个区域, 在一定程度上降低了计算复杂度。自适应变长编码(CAVLC) [37]依据上下文信息选择当前待编码符号应使用的码表, 也很好解决了符号概率变化的问题,



虽然编码性能低于自适应算术编码，但是具有复杂度低的优点。

### 1.1.2 基于块的运动补偿及变换的混合视频编码技术

基于块的运动补偿及变换的混合视频编码[38]是国际标准MPEG及H.26x系列所采用的主要编码技术，因此在本节中进行重点介绍。

在运动补偿和变换编码混合的视频编码器中，运动补偿预测首先降低了时间上的冗余度。接着对相应帧之间的差值进行变换编码，以降低空间冗余度。在去除时间、空间上的冗余度的同时，还要去除视觉心理引起的冗余。图1.1是典型的运动补偿预测和变换视频编码器的框图。

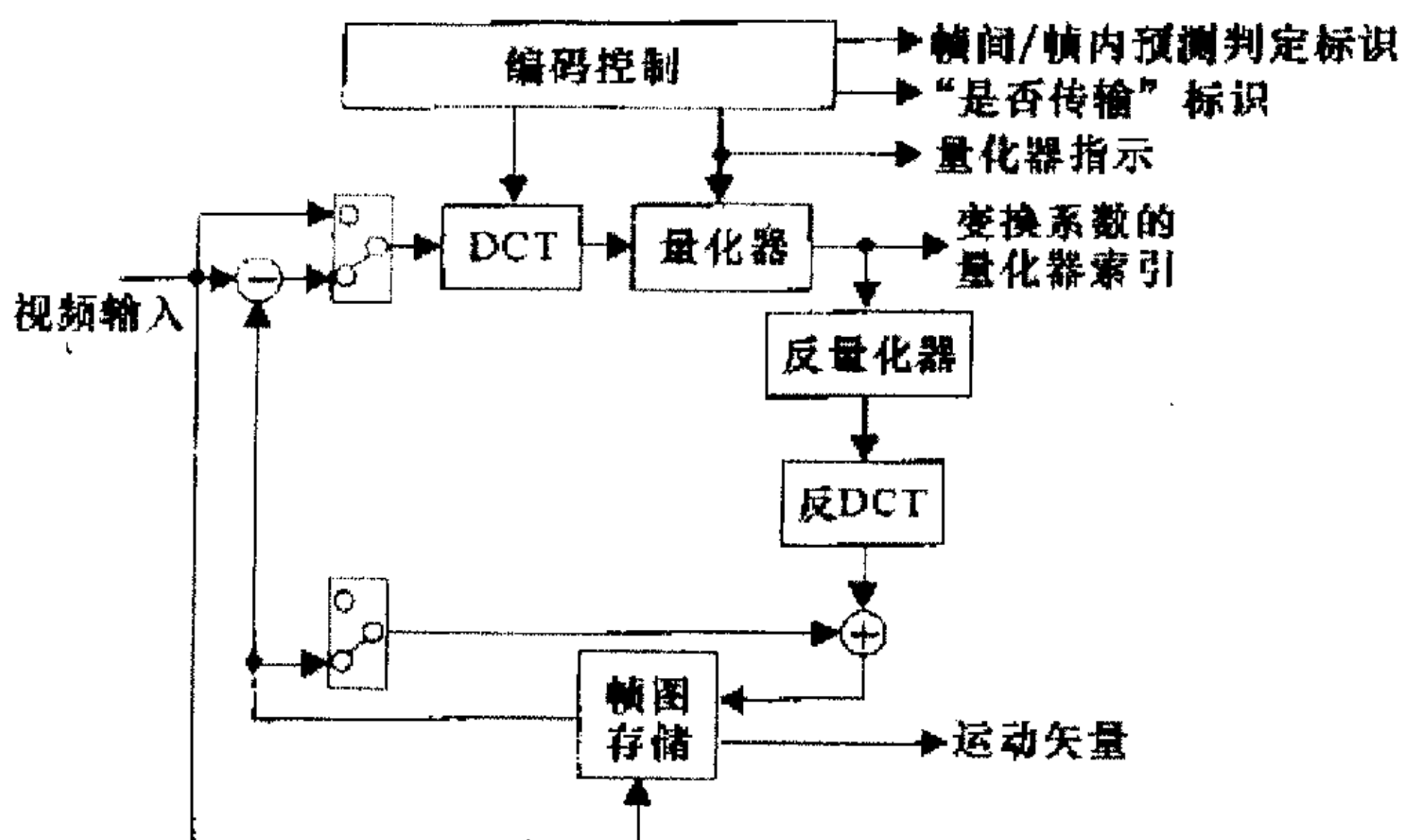


图1.1 基于块的运动补偿预测和变换视频编码器框图

#### 1.1.2.1 运动估算及补偿

首先，把每个视频帧划分为大小相等的宏块。运动补偿预测时，假定当前图像内的像素块是从前一图像内的某个块经平移而得。二维的位移矢量（或运动矢量）代表该块的运动信息。由于图像是基于块来表示，许多的运动估算方法采用块匹配技术。运动估算算法中最为广泛使用的方式是绝对差分SAD(sum of absolute difference)。为了找到匹配误差最小的块，需要计算在搜索窗内不同位置的SAD值。最简单但也是计算量最大的搜索方式是计算位于搜索窗内每个可能像

素位置上的SAD值（为了降低计算的复杂性，提出了把搜索范围限制在几个点的算法）。每一个块的运动矢量都考虑到了运动补偿。亚像素（sub pixel）运动估算算法可以提高重构图像的质量。目前，大多数的视频编码标准允许运动矢量的水平分量和垂直分量可以达到 1/4个像素的精度。运动矢量允许的取值范围限制了运动估算时搜索窗口的大小。在H.264以前的标准中，实施运动补偿的P和B-宏块的大小为16x16或8x8。P-宏块只能采用前向预测方式，B-宏块则从过去或/和将来帧预测得到。在给定的附加编码选项下，B-宏块的压缩性能要高于P-宏块。

### 1.1.2.2 变换

H.264以前的视频编码标准中都采用8x8 DCT，其目的就是去除8x8像素块内的空间相关性，并把能量压缩到尽可能少的系数中。除了相当高的去除相关性和能量压缩能力外，DCT的运算效率也很高，并且可以用软件和硬件来实现。

理论上，采用DCT反变换可以精确重构出原始信号，但采用精度有限的算法不能达到这一点。当前向的DCT误差在允许范围内时，DCT反变换的误差也必须满足最小精度等级，以避免在编码器和解码器上的重构帧之间出现不匹配。

8x8 DCT变换产生1个DC系数和63个AC系数。DC系数是被变换抽样的均值，代表图像块最粗糙的轮廓（最低的空间频率），AC系数表示图像的精细细节（较高的空间频率）。在编码器里，DCT变换过后接着就是量化，这个阶段，要在损失视频细节和提高视频压缩率之间做折衷处理。

### 1.1.2.3 量化

人对于空间频率的低频分量引起的重构误差比高频分量引起的重构误差要敏感得多。亮度和色度上缓慢的线性变化（低频信息）对于人眼十分重要，而急速的变化（高频信息）通常不能察觉到，因此可以采用非均匀的量化步长。DCT输出矩阵中的每个元素对应的量化值，由每个DCT系数除以一个量化参数获得。量化参数是控制视频压缩量和降低压缩后视频保真度的主要方法。实际结果是：同原始的DCT系数的方差相比，量化后系数的方差要变小，同时非零的系数也减少，这样也就提高了熵编码的效率。

#### 1.1.2.4 熵编码

在进行熵编码之前，要通过“之”字形扫描，把量化后的二维DCT系数变为一维的系数阵。这种重新编排方式使得DC系数位于系数阵的最前面，接下来的AC系数按低频到高频的顺序排列。再用三维变长VLC表对重新排列后的系数阵进行变长编码。

通过采用变长编码（VLC）的方式，熵编码减少了用于表示压缩后视频数据所需的平均比特数。通常使用Huffman编码来产生可变长度的码字：经常出现的符号用较短的码字表示。算术编码也可作为熵编码的一种方式。其他信息，如预测类型及量化器标记等，也通常采用VLC或算术编码方式进行熵编码。

#### 1.1.2.5 编码控制

图1.1中的两个开关表示帧内/帧间模式的选择，该选择通常在“宏块级”实施。根据相关的失真值来测定运动估算过程的性能，该性能作为选择编码模式的依据。运用了时间预测的编码模式叫帧间编码模式(INTER MODE)。如果运动补偿处理有效，则选择该模式，并只对预测误差即原始宏块和运动补偿预测后宏块的差值进行编码。如果没有采用时间预测，相应的编码模式叫帧内编码模式(INTRA MODE)。如果相对于参考帧，某个宏块没有显著的改变，则编码器可以选择不对它进行编码，解码器将简单地复制参考帧内位于相同空间位置的宏块。这种编码模式叫SKIP编码。

也可以采用基于率失真最优法（RD Optimization）的较为复杂的“编码模式选择”算法。例如，可以采用基于率失真理论的拉格朗日最小化技术，对多种可能代表不同“率失真”效率的选择方式做最优化处理。在给定编码输出速率下，率失真理论设定了信源编码级有效的输出失真范围，或者反过来，给定输出失真的大小，该理论也设定了有效输出速率范围。

在视频编码过程中，编码模式通常同信号的率失真特性相关联，同时，率失真的权衡也同编码参数的选择有内在联系。最优化的任务就是：对于每个图像块，在某给定率失真的前提下，选择效率最高的编码方式。可是，在不同的比特率及不同的场景内容下，不同的编码选项显示出不同的编码效率，这一事实使得最优

化任务变得很复杂。

## 1.2 视频压缩的国际标准

近年来,图像压缩编码研究成为一个越来越活跃的领域,图像编码技术的发展和广泛应用也促进了许多有关国际标准的制定。目前视频领域中最为重要的编解码标准有国际电联的H. 261, H. 263, H. 264和国际标准化组织运动图像专家组的MPEG系列标准MPEG1/2/4,而在网络的视频流传输中,则活跃着Real Video, WMV, QuickTime等,此外还有ON2的VP5, VP6以及我国自主研制的AVS等。

H. 26x和MPEG标准发展如图1.2所示[39]:

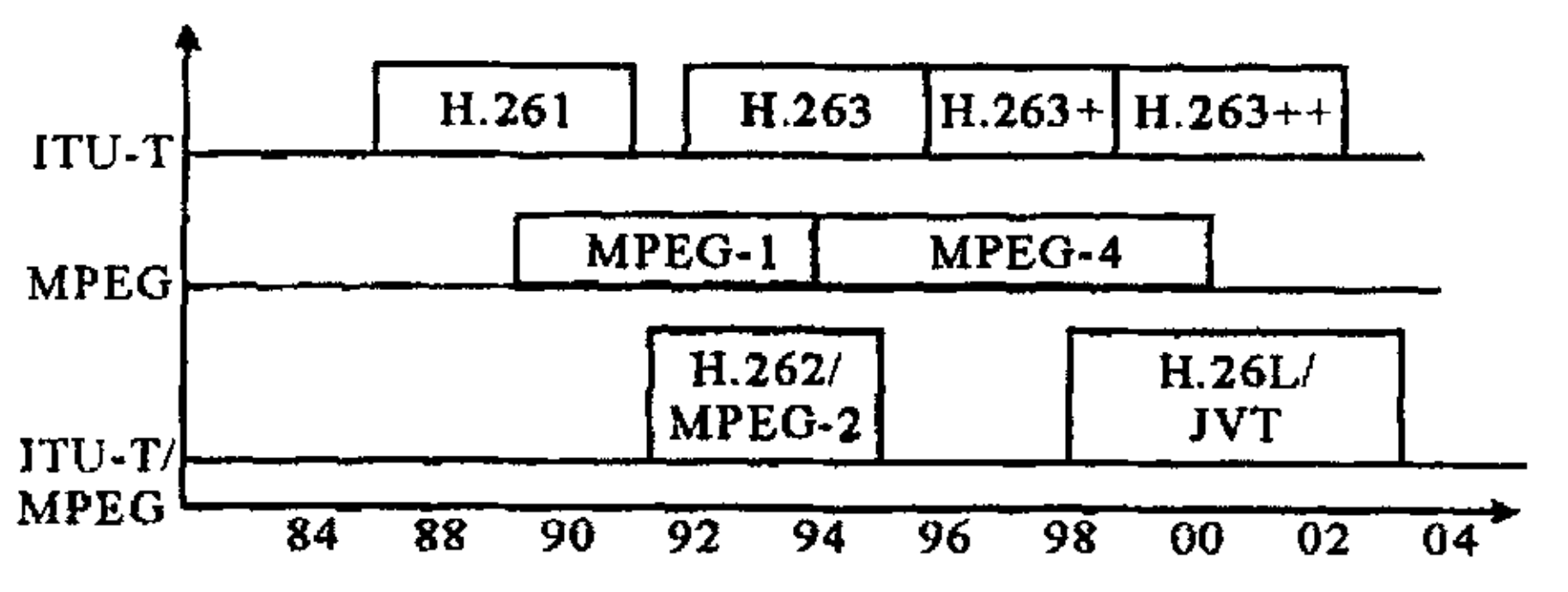


图1.2 标准发展图

可见, H.261是制定最早的视频压缩标准,也是后续标准的蓝本。H.264(早期称为H. 26L)则是两大组织的最新成果,由它们成立的JVT制定。在ITU—T称为H. 264,在MPEG称为MPEG-4 AVC (Part 10)。

### 1.2.1 H.26x 系列

#### (1) H.261

H.261是第一个广泛投入应用的数字视频编码标准,又称为px64,其中p为64 kb/s的取值范围,是1~30的可变参数, p取值较小时,只能传清晰度不太高的图像,适合于面对面的电视电话; p取值较大时(如 $p > 6$ ),则可以传输清晰度较好的会议电视图像,最初是针对在ISDN上实现电信会议应用特别是面对面的可视电话和视频会议而设计的, H. 261奠定了混合编码框架的雏形,其视频压缩算法的核心是运动估值预测和DCT编码,许多技术(包括视频数据格式、运动估算与补偿,DCT变换、量化和熵编码)都被后来的MPEG—1和MPEG—2所借鉴和采

用。它属于恒定码流可变质量编码而非恒定质量可变码流编码。

## (2) H.263

H.263是为低码流通信而设计的。但实际上这个标准可用于很宽的码流范围，而非只用于低码率范围，它在许多应用中可以被用来取代H.261。H.263的编码算法与H.261一样，但做了一些改善，以提高性能和纠错能力。H.263标准在低码率下能够提供比H.261更好的图像效果，例如在比特率低于30 kb/s的应用中，同样的视频质量前提下，H.263的输出码率仅为H.261的一半甚至更少。

两者的区别包括：

- ①H.263的运动补偿使用半像素精度，而H.261则用全像素精度和循环滤波；
- ②数据流层次结构的某些部分在H.263中是可选的，使得编解码可以配置成更低的数据率或更好的纠错能力；
- ③H.263包含4个可协商的选项以改善性能；
- ④H.263采用无限制的运动向量以及基于语法的算术编码；
- ⑤采用事先预测和与MPEG中的P-B帧一样的帧预测方法；
- ⑥H.263支持5种分辨率，即除了支持H.261中所支持的QCIF和CIF外，还支持SQCIF, 4CIF和16CIF。

## (3) H.263+/++

H.263+是H.263的改进版，它在保证原H.263标准的核心句法和语义不变的基础上，增加了若干选项以提高压缩效率或某方面的功能，它提供了12个新的可协商模式和其他特征，进一步提高了压缩编码性能。另一重要的改进是可扩展性，它允许多显示率、多速率及多分辨率，增强了视频信息在易误码、易丢包异构网络环境下的传输。新增选项可归纳为新的图像种类和新的编码模式两类。

H.263++是在H.263+的基础上又增加了3个选项，主要是为了增强码流在恶劣信道上的抗误码性能，同时也是为了增强编码效率，这3个选项分别为：

- ①增强型的参考帧选择；
- ②数据分片的模式；
- ③在H263+的码流中增加了补充信息，保证增强型的反向兼容。

## (4) H.264

H.264同样采用DPCM加变换编码的混合编码模式。但它采用“回归基本”的



简洁设计,不用众多的选项,获得比H.263 ++好得多的压缩性能;应用目标范围较宽,以满足不同速率、不同分辨率以及不同传输(存储)场合的需求。

技术特点包括:

- ①帧间编码,可变块大小的运动补偿预测技术;
- ②帧内编码,多方向的空间预测技术;
- ③环内滤波器,去除块效应;
- ④通4 x4的整数正交变换及相应的量化策略;
- ⑤改进的运动向量预测编码;
- ⑥更加高效的熵编码器;
- ⑦基于率失真的编码优化技术。

这些措施使得H. 264算法具有很高的编码效率,在相同的重建图像质量下,能够比H. 263节约50%左右的码率。H.264的码流结构网络适应性强,增加了差错恢复能力,能够很好地适应IP和无线网络的应用。但从总体上说,H. 264性能的改进是以增加复杂性为代价而获得的,其编码的计算复杂度大约相当于H.263的3倍,解码复杂度大约相当于H.263的2倍。

## 1.2.2 MPEG 系列

### (1) MPEG-1

MPEG-1主要是针对1.5 Mbps以下数据传输率的数字存储媒质运动图像及其伴音编码的国际标准。采用CIF视频格式,具有随机存取、快速正向/逆向搜索、逆向重播、视听同步、容错性、编码/解码延迟等功能。MPEG-1视频编码部分的基本算法与H. 261/H.263相似,但又引入了帧内帧(I)、预测帧(P)、双向预测帧(B)和直流帧(D)等概念,进一步提高了编码效率,旨在达到VHS质量,对于动作不激烈的视频信号能获得较好的图像质量,但如果图像对象动作激烈时,图像有可能产生马赛克现象。

### (2) MPEG-2

MPEG-2核心部分与MPEG-1基本相同,是在MPEG-1基础上的进一步扩展和改进,克服并解决了MPEG-1不能满足日益增长的多媒体技术、数字电视技术对分辨率和传输率等方面的技术要求的缺陷。最高图像分辨率:1920x1152;普通PAL

制式分辨率:720x576; 最高数据流量: 80 Mbps; 一般数据流量: 6500 kbps (720 x 576)。它是主要针对数字视频广播、高清晰度电视和数字视盘等制定的编码标准, 可以支持固定比特率传送、可变比特率传送、随机访问、信道跨越、分级编码、比特流编辑等功能。从本质上讲, MPEG-2可视为一组MPEG-1的最高级编码标准, 它保留了MPEG-1所提供的所有功能, 并设计成与MPEG-1兼容, 但又增加了基于帧/场的运动补偿、空间可伸缩编码、时间可伸缩编码、质量可伸缩编码以及容错编码等新的编码技术。MPEG-2有不可分级和可分级两种编码方式。它还定义了5个框架和4个级别, 框架是标准中定义的语法子集, 级别是一个特定框架中参数取值的集合。框架和级别限定以后, 解码器的设计和校验, 就可以针对限定的框架在限定的级别中进行, 同时也为不同的应用领域之间的数据交换提供了方便和可行性, 其中的主框架应用最广, 也是最为重要的一个。

### (3) MPEG-4

MPEG-4的第一版完成于1998年, 第二版完成于1999年。初始的名称是“甚低比特率音频视频编码”, 后定义为“音视频目标编码”。码率限制的消失意味着更宽广的适用码率范围, 而把“视音频编码”转变为对“视音频对象”的编码则是工作目标质的飞跃。但是MPEG-4的基本视频编码器还是属于和H. 263相似的一类混合编码器。它的最高图像分辨率:720x576; 普通PAL制式分辨率:720x576; 最高数据流量:5~10 Mbps; 一般数据流量: 880 kbps(720 x576)。

与MPEG-1和MPEG-2相比, MPEG-4的区别主要体现在以下几点:

- ①基于内容的编码, 即不是像MPEG-1, MPEG-2基于像素的编码, 而是基于对象(object)和实体(entity)进行编码;
- ②编码效率的改进和并发数据流的编码;
- ③错误处理的鲁棒性, 有助于低比特率视频信号在高误码率环境(如移动通信环境)下的存储和传输;
- ④基于内容的可伸缩性(scalable), 用户可以有选择的只对感兴趣的对象进行传输、解码和显示;
- ⑤ MPEG-2是基于“推”的概念, 就是说信号源提供什么, 显示设备就显示什么, 信号源在“推”显示设备, 这在网络时代表现了一定的局限性。而MPEG-4是基于“拉”的概念, 就是说显示设备需要显示什么, 就要求信号源提供什么。

如果连接速度比较慢,要么等待信号源传输高质量的图像和声音,要么要求信号源只传输实时的低质量图像和声音,就如同显示设备在“拉”信号源。

### 1.2.3 其他标准

H.26x和MPEG是两大系列标准,但在网络上,应用最广的是RV和WMV,它们也代表了流媒体领域的两大阵营, RV改变了人们观看电影等视频信息的习惯,使人们从必须全部下载或复制一个影音文件转变为可以一边下载一边观看,这不舍于一场革命,又由于它较低的压缩率,很快就风行于网络。WMV则是微软的产品,属于流媒体领域的后起之秀,由于微软的强大实力和垄断地位,很快就在流媒体领域占领了一席之地,目前它是RV格式的最大竞争对手。近期,微软刚刚推出了WMV9,并且希望把它的定位从互联网传送至PC机过渡到从任何网络传送到任何消费终端。

一个标准制定出来以后,必然涉及到是否收费的问题,前文所述的MPEG-1, H.263很少有收费案例出现,而MPEG-2由MPEG LA根据解码器收费, MPEG-4不仅对编解码器收费,还对运营商按时间收费,它的收费问题已经引起了轩然大波,目前仍在争议之中。新制定的H.264已确定收费方案,对国内厂商而言,以往多是直接采用国际标准MPEG,但昂贵的专利费是一个沉重的负担,所以,拥有自主知识产权至关重要。

正是为了完全摆脱专利费的困扰,我国正在研究开发自己的数字音视频编解码技术和标准——AVS标准,它在性能上与最新的国际标准相同或更高,兼容国际标准基本功能,所包含的技术专利不受控于境外企业与组织。

AVS视频当中具有特征性的核心技术包括:8x8整数变换、量化、帧内预测、1/4精度像素插值、特殊的帧间预测运动补偿、二维熵编码、去块效应环内滤波等[40],下面分别予以介绍:

#### (1) 变换量化

AVS的8x8变换与量化可以在16位处理器上无失配地实现,从而克服了MPEG-4 AVC/ H.264之前所有视频压缩编码国际标准中采用的8x8 DCT变换存在失配的固有问题。而MPEG-4 AVC/ H.264所采用的4x4整数变换在高分辨率的视频图像上的去相关性能不及8x8的变换有效。AVS采用了64级量化,可以完全适

应不同的应用和业务对码率和质量的要求。在解决了16位实现的问题后, AVS所采用的8x8变换与量化方案, 即适合于16位DSP或其他软件方式的快速实现, 也适合于ASIC的优化实现。

## (2) 帧内预测

AVS的帧内预测技术沿袭了MPEG-4 AVC/ H.264帧内预测的思路, 用相邻块的像素预测当前块, 采用代表空间域纹理方向的多种预测模式。但AVS亮度和色度帧内预测都是以8x8块为单位的。亮度块采用5种预测模式, 色度块采用4种预测模式, 而这4种模式中又有3种和亮度块的预测模式相同。在编码质量相当的前提下, AVS采用较少的预测模式, 使方案更加简洁、实现的复杂度大为降低。

## (3) 帧间预测

帧间运动补偿编码是混合编码技术框架中最重要的部分之一。AVS标准采用了16x16, 16x8, 8x16和8x8的块模式进行运动补偿, 而去除了MPEG-4 AVC/ H.264标准中的8x4, 4x8, 4x4的块模式, 目的是能更好地刻画物体运动, 提高运动搜索的准确性。实验表明, 对于高分辨率视频, AVS选用的块模式已经能足够精细地表达物体的运动。较少的块模式, 能降低运动矢量和块模式传输的开销, 从而提高压缩效率、降低编解码实现的复杂度。

AVS和MPEG-4 AVC/ H.264都采用了1/4像素精度的运动补偿技术。MPEG-4 AVC/ H.264采用6抽头滤波器进行半像素插值并采用双线性滤波器进行1/4像素插值。而AVS采用了不同的4抽头滤波器进行半像素插值和1/4像素插值, 在不降低性能的情况下减少插值所需要的参考像素点, 减小了数据存取带宽需求, 这在高分辨率视频压缩应用中是非常有意义的。

在传统的视频编码标准(MPEG-x系列与H.26x系列)中, 双向预测帧B帧都只有一个前向参考帧与一个后向参考帧, 而前向预测帧P帧则只有一个前向参考帧。而新近的MPEG-4 AVC/ H.264充分地利用图片之间的时域相关性, 允许P帧和B帧有多个参考帧, 最多可以有31个参考帧。多帧参考技术在提高压缩效率的同时也将极大地增加存储空间与数据存取的开销。AVS中P帧可以利用至多2帧的前向参考帧, 而B帧采用前后各一个参考帧, P帧与B帧(包括后向参考帧)的参考帧数相同, 其参考帧存储空间与数据存取的开销并不比传统视频编码的标准大, 而恰恰是充分利用了必须预留的资源。



AVS的B帧的双向预测使用了直接模式(direct mode)、对称模式(symmetric mode)和跳过模式(skip mode)。使用对称模式时,码流只需要传送前向运动矢量,后向运动矢量可由前向运动矢量导出,从而节省后向运动矢量的编码开销。对于直接模式,当前块的前、后向运动矢量都是由后向参考图像相应位置块的运动矢量导出,无需传输运动矢量,因此也可以节省运动矢量的编码开销。跳过模式的运动矢量的导出方法和直接模式的相同,跳过模式编码的块其运动补偿的残差也均为零,即该模式下宏块只需要传输模式信号,而不需要传输运动矢量、补偿残差等附加信息。

#### (4) 熵编码

AVS熵编码采用自适应变长编码技术。在AVS熵编码过程中,所有的语法元素和残差数据都是以指数哥伦布码的形式映射成二进制比特流。采用指数哥伦布码的优势在于:一方面,它的硬件复杂度比较低,可以根据闭合公式解析码字,无需查表;另一方面,它可以根据编码元素的概率分布灵活地确定以k阶指数哥伦布码编码,如果k选得恰当,则编码效率可以逼近信息熵。

对预测残差的块变换系数,经扫描形成(level、run)对串,level、run不是独立事件,而存在着很强的相关性,在AVS中level、run采用二维联合编码,并根据当前level、run的不同概率分布趋势,自适应改变指数哥伦布码的阶数。

AVS视频基本层分为4个级别(level),分别对应高清晰度与标准清晰度应用。与H. 264的基本层相比,AVS视频的压缩效率明显提高,而与H. 264的MP层(main profile)相比,又减少了CABAC等实现难度大的技术,从而增强了可实现性。

AVS是一套适应面十分广阔的技术标准。与MPEG-2等现有压缩标准相比,在相同的视觉质量下,码率至少可以降低50%,码率范围覆盖几十kbps的低带宽通信到数十Mbps的高清晰度电视广播,可以支持低延迟模式的视频会议应用,也支持高压压缩效率的视频存储应用等,明显超越了现有的所有视频编码标准。

### 1.2.4 结论与展望

总的来说,MPEG系列标准从针对存储媒体的应用发展到适应传输媒体的应用,其视频编码的基本框架是和H. 26x一致的,只不过H. 26x致力于高效率的视频编码技术,而MPEG更侧重系统和框架。具体来说,从H. 261到H. 262/3,



MPEG-1/2/4等都有一个共同的不断追求的目标,即在尽可能低的码率(或存储容量)下获得尽可能好的图像质量,但如何适应不同信道传输特性的问题也日益显现出来。可以说,H.261是视频编码的典范之作,H.263是其发展,但H.263选项众多,往往令使用者无所适从。MPEG-4的“基于对象的编码”部分由于尚有技术障碍,目前还难以普遍应用。在此基础上发展起来的H.264克服了上述弱点,在混合编码的框架下引入了新的编码方式,提高了编码效率,并大大加强了“网络友好性”。同时,它是两大国际标准化组织共同制定的,应用前景十分看好。

### 1.3 本文的工作

本文首先介绍了视频编码的理论及编码技术,着重介绍了最新的视频压缩编码标准 H.264,将其与之前的标准进行了比较,并分析了 H.264 编码器各模块的计算复杂度,选取复杂度最高的预测部分作为优化的主要对象,包括帧间预测(整像素运动搜索和亚像素运动搜索)和帧内预测。

在分析了以往针对这些模块所提出的优化算法的基础上,本文对这些算法进行了进一步的改进,分别提出了三种效率更高的算法,极大地降低了运算复杂度,简述如下:

- 1) 整像素运动搜索提前中止算法——不同块大小的最佳矢量检测 VBBMD(Variable Block-size Best Motion Detection)。该算法能通过计算一个搜索点来预测最佳运动矢量,因而能提前中止一部分运动搜索过程,跳过与之相对应的计算。与 H.264 标准所采纳的快速算法 UMHexagonS 相比,该算法能进一步减少 76%~94%的整像素搜索点,而平均 PSNR 下降只有 0.06dB。将 VBBMD 与 UMHexagonS 结合使用,其整像素运动搜索复杂度只有全搜索的 5%左右;
- 2) 快速亚像素运动搜索算法——基于预测的方向亚像素搜索 PDFPS(Prediction-based Directional Fractional Pixel Search)。该算法采用两种预测方式来更准确地预测亚像素运动矢量,并直接从预测位置或预测方向开始搜索,采用 1/4 像素步长的菱形模式来进行细化搜索。同时采用提前中止算法,进一步跳过更多的搜索点。与全搜索及 H.264 所采纳的快速亚像素搜索算法 CBFPS 相比,本文所提出的算法 PDFPS 能分别降

低最多 84% 和 74% 的亚像素搜索点，而图像质量的下降仅为 0.01dB，是一种非常有效的快速亚像素搜索算法。

- 3) 快速帧内预测算法——基于模式预测的快速帧内预测算法。该算法利用视频序列时域、空域及不同块大小之间的相关性，用前帧对应块、当前块周围块及 16x16 块的模式来预测当前块的模式，并结合提前中止方法，极大地减少了帧内预测所需计算的模式数量，降低了运算复杂度。与全搜索相比，该快速算法能降低 50%~75% 的帧内预测计算量，同时保持编码视频的图像质量。

通过所提出的快速算法对以上模块的优化，使其计算复杂度大为降低，并能将 H.264 参考软件 JM8.0 编码器的速度提高 3~4 倍，对 H.264 的实时应用进行了有益的探索。

## 第二章 H.264 视频编解码标准采用技术及复杂度分析

H.264 也被称为“MPEG-4 Visual Part 10”，也就是“MPEG-4 AVC (Advanced Video Coding)”，是由 ISO/IEC 下属的运动图像专家组 MPEG (Moving Pictures Experts Group) 和 ITU 下属的视频编码专家组 VCEG (Video Coding Experts Group) 共同成立的联合视频小组 JVT (Joint Video Team) 共同制定的，并于 2003 年 3 月被正式确定为国际标准 IS (International Standard)。与以往的标准相比，H.264 具有最高的编码效率，其主要优点有[41]：

- (1)在相同的重建图像质量下，H.264 比 H.263+和 MPEG-4(Part 2)节约 50% 码率；
- (2)H.264 可在很宽的码率范围内提供高质量的视频服务；
- (3)H.264 对信道时延的适应性较好，既可工作在低时延模式以满足实时业务，如会议电视等，又可工作于无时延限制的宽松场合，如视频存储等；
- (4)具备较强的抗误码能力，包括支持传输环境恶劣的无线网络；
- (5)采用分层模式，视频编码层(VCL)外特有的网络适配层(NAL)专为网络传输设计，因而进一步提高网络友好性。

以上特点使得 H.264 一推出就受到了极大的关注，在可以预见的将来必将得到广泛的应用。本章将介绍 H.264 所采用的技术，并对其复杂度进行分析。

### 2.1 H.264 编码框架

H.264 标准的主导思想是与现有的视频编解码标准一致的——基于块的混合编码方法，但是它同时运用了大量不同的技术，使得其视频编码性能远远优于任何其他标准。H.264 编码器基本框图 2.1 所示。

## 第二章 H.264 视频编解码标准采用技术及复杂度分析

H.264 也被称为“MPEG-4 Visual Part 10”，也就是“MPEG-4 AVC (Advanced Video Coding)”，是由 ISO/IEC 下属的运动图像专家组 MPEG (Moving Pictures Experts Group) 和 ITU 下属的视频编码专家组 VCEG (Video Coding Experts Group) 共同成立的联合视频小组 JVT (Joint Video Team) 共同制定的，并于 2003 年 3 月被正式确定为国际标准 IS (International Standard)。与以往的标准相比，H.264 具有最高的编码效率，其主要优点有[41]：

(1)在相同的重建图像质量下，H.264 比 H.263+和 MPEG-4(Part 2)节约 50% 码率；

(2)H.264 可在很宽的码率范围内提供高质量的视频服务；

(3)H.264 对信道时延的适应性较好，既可工作在低时延模式以满足实时业务，如会议电视等，又可工作于无时延限制的宽松场合，如视频存储等；

(4)具备较强的抗误码能力，包括支持传输环境恶劣的无线网络；

(5)采用分层模式，视频编码层(VCL)外特有的网络适配层(NAL)专为网络传输设计，因而进一步提高网络友好性。

以上特点使得 H.264 一推出就受到了极大的关注，在可以预见的将来必将得到广泛的应用。本章将介绍 H.264 所采用的技术，并对其复杂度进行分析。

### 2.1 H.264 编码框架

H.264 标准的主导思想是与现有的视频编解码标准一致的——基于块的混合编码方法，但是它同时运用了大量不同的技术，使得其视频编码性能远远优于任何其他标准。H.264 编码器基本框图 2.1 所示。

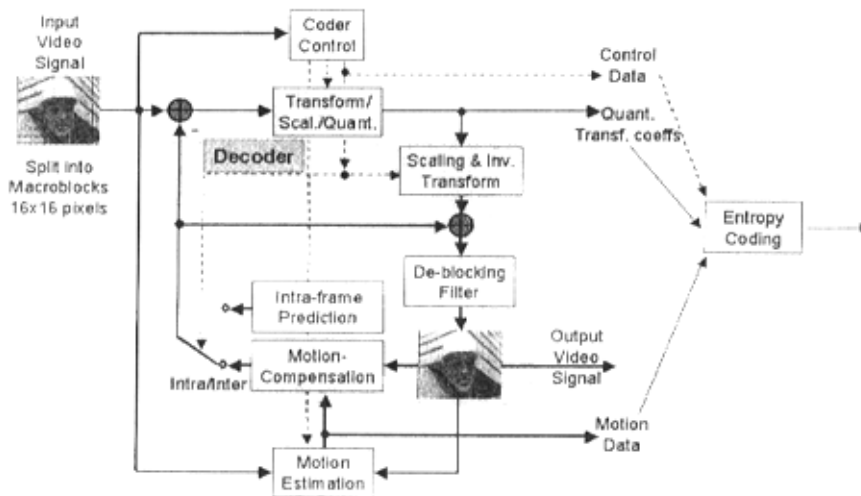


图 2.1 H.264 编码器框图

H.264 有 3 个 profile: Baseline Profile, Main Profile, X Profile 和 11 个 level 以满足不同应用的需要。表 2.1 列出了各个 profile 所针对的应用和采用的编码算法。

表 2.1 H.264 的 Profile

Profile	算法	典型应用
Baseline	<ul style="list-style-type: none"> <li>● I and P picture types (not B)</li> <li>● In-loop deblocking filter</li> <li>● 1/4-sample motion compensation</li> <li>● Tree-structured motion segmentation down to 4x4 block size</li> <li>● VLC-based entropy coding</li> <li>● Some enhanced error resilience features                             <ol style="list-style-type: none"> <li>1. Flexible macroblock ordering /arbitrary slice ordering</li> <li>2. Redundant slices</li> </ol> </li> </ul>	低延时应用 (可视电话、视频会议等)
Main	<ul style="list-style-type: none"> <li>● All Baseline features except enhanced error resilience features</li> <li>● B pictures</li> <li>● CABAC</li> <li>● MB-level frame/field switching</li> <li>● Adaptive weighting for B and P picture prediction</li> <li>● Interlace</li> </ul>	流媒体、移动应用等



Profile X	<ul style="list-style-type: none"> <li>● All Baseline features</li> <li>● B pictures</li> <li>● More error resilience: Data partitioning</li> <li>● SP/SI switching pictures</li> </ul>	电视广播应用等
-----------	---	---------

## 2.2 H.264 的编码技术

H.264 标准采用了与已经制定的视频编解码标准（如 H.263 和 MPEG-4）相类似的一些编解码方法。主要包括如下几项技术：

(1)将每个视频图像分成 16x16 的像素宏块。这种处理就使得视频图像能以像素宏块为单位进行处理。

(2)利用时域相关性。时域上的相关性存在于那些连续图像的块之间，这就使得在编码的时候只需要编码那些差值即可。一般我们是通过运动估值和运动补偿来利用时域相关性的。对于一个像素块来说，在已经编码好的前一幅或几幅图像中搜索其相关像素块，从而获得其运动矢量，而该运动矢量就在编码端和解码端被用来预测当前像素块。

(3)利用残差的空域冗余度。在运动估值后，编码端只需要编码残差即可，也就是编码当前块与其相应的预测块的差。编码过程还是类似的变换、量化、扫描和熵编码。

(4)其他技术。还包括传统的 4:2:0 的色度数据与亮度数据的采样关系；块运动置换；超越图像边界的运动矢量；变化块大小的运动；可分级的量化；I、P 和 B 图像类型。

与此同时，H.264 采用了一些与以往的编码方法不同的技术，极大地提高了编码效率，主要有：

### 2.2.1 改进的运动估值和运动补偿

首先，H.264 采用了不同大小和形状的宏块分割与亚分割的方法。一个宏块的 16x16 亮度值可以按照 16x16、16x8、8x16 或 8x8 进行分割，而如果选择了 8x8 分割，那么还可以按照 8x8、8x4、4x8 或 4x4 进行亚分割（如图 2.2 所示），这些宏块分割与亚分割的模式可以组合出许多种宏块的分割方法。这种把宏块划

Profile X	<ul style="list-style-type: none"> <li>● All Baseline features</li> <li>● B pictures</li> <li>● More error resilience: Data partitioning</li> <li>● SP/SI switching pictures</li> </ul>	电视广播应用等
-----------	---	---------

## 2.2 H.264 的编码技术

H.264 标准采用了与已经制定的视频编解码标准（如 H.263 和 MPEG-4）相类似的一些编解码方法。主要包括如下几项技术：

(1)将每个视频图像分成 16x16 的像素宏块。这种处理就使得视频图像能以像素宏块为单位进行处理。

(2)利用时域相关性。时域上的相关性存在于那些连续图像的块之间，这就使得在编码的时候只需要编码那些差值即可。一般我们是通过运动估值和运动补偿来利用时域相关性的。对于一个像素块来说，在已经编码好的前一幅或几幅图像中搜索其相关像素块，从而获得其运动矢量，而该运动矢量就在编码端和解码端被用来预测当前像素块。

(3)利用残差的空域冗余度。在运动估值后，编码端只需要编码残差即可，也就是编码当前块与其相应的预测块的差。编码过程还是类似的变换、量化、扫描和熵编码。

(4)其他技术。还包括传统的 4:2:0 的色度数据与亮度数据的采样关系；块运动置换；超越图像边界的运动矢量；变化块大小的运动；可分级的量化；I、P 和 B 图像类型。

与此同时，H.264 采用了一些与以往的编码方法不同的技术，极大地提高了编码效率，主要有：

### 2.2.1 改进的运动估值和运动补偿

首先，H.264 采用了不同大小和形状的宏块分割与亚分割的方法。一个宏块的 16x16 亮度值可以按照 16x16、16x8、8x16 或 8x8 进行分割，而如果选择了 8x8 分割，那么还可以按照 8x8、8x4、4x8 或 4x4 进行亚分割（如图 2.2 所示），这些宏块分割与亚分割的模式可以组合出许多种宏块的分割方法。这种把宏块划

分成不同大小的块和亚块进行运动补偿的方法被称为树结构的运动补偿。宏块分割与亚分割所产生的每一个亮度块都有自己的运动矢量。对于宏块中的色度值，分割方法是和亮度一样的，但是由于 4:2:0 采样的关系，色度分割块的大小是亮度分割块的一半，而且当色度块在利用运动矢量时，必须要将它各个分量除以二。

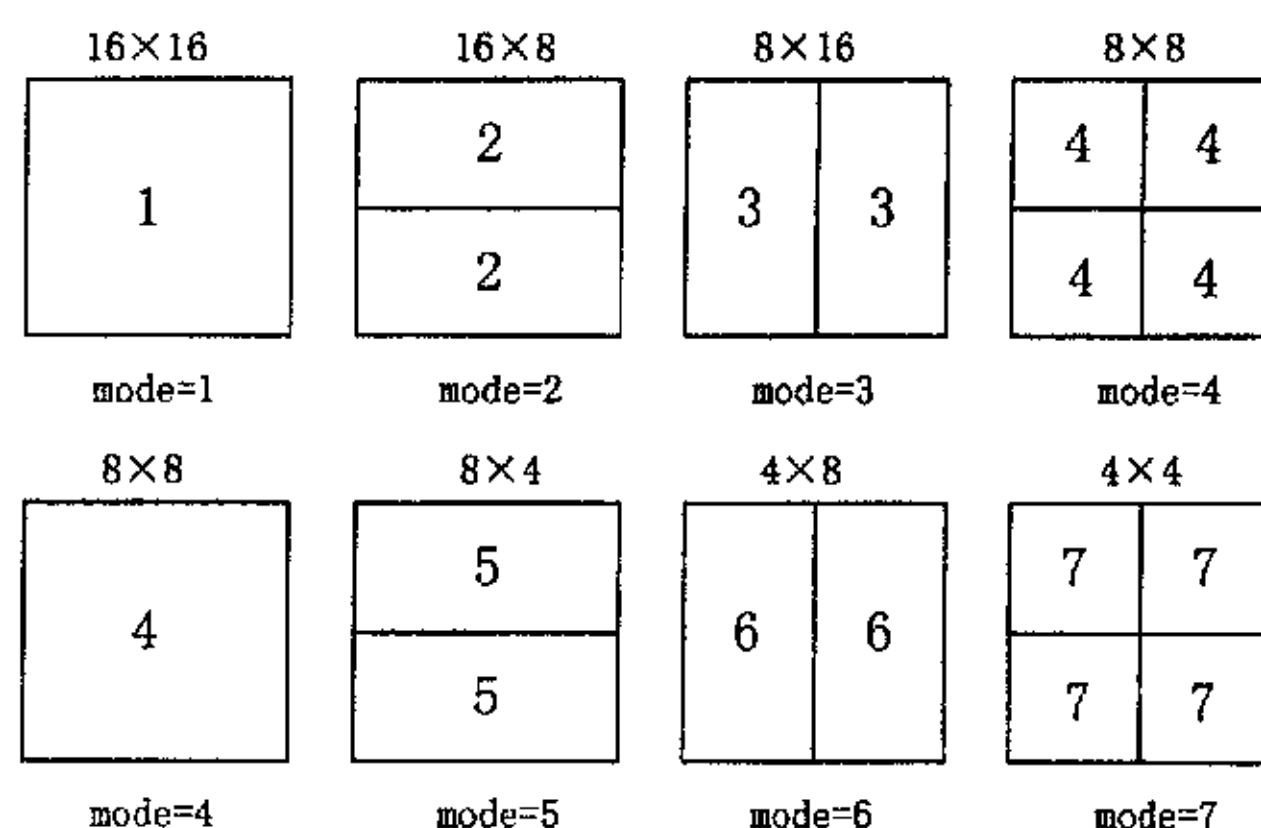


图 2.2 H.264 中的宏块划分

其次，H.264 可以达到四分之一像素的运动精度，这是通过利用整像素点的亮度值进行内插得到的。内插过程先是通过 6 抽头的滤波器来获得半像素精度，然后用线性滤波器来获得四分之一像素的精度。又由于 4:2:0 采样的关系，色度的运动精度就达到八分之一像素，这也是通过线性滤波器插值得到的。

由于运动矢量之间也有一定的相关性，所以 H.264 就利用已经编码块的运动矢量对当前未编码块的运动矢量进行预测，最后只需要编码和传输实际运动矢量与预测值的差值即可。

再次，H.264 还可以采用多参考图像（最多前向和后向各 5 帧）来进行运动预测，这样可以对周期性运动、平移封闭运动和不断在两个场景间切换的视频流有效果非常好的运动预测。使用了多参考图像，H.264 不仅能够提高编码效率，同时也能实现更好的码流误码恢复，但需要增加额外的时延和存储容量。

最后，H.264 也在 B 图像中利用后向运动预测，这和以前的标准是一致的，但不同的是 B 图像通过加权也能作为其他图像的参考图像。

## 2.2.2 新的帧内预测

在以往标准中，只采用 Intra 编码的图像称为 I 图像，在 I 图像中是直接像素块的数值进行变换，这样处理的结果就导致 I 图像包含了大量的数据，压缩效

分成不同大小的块和亚块进行运动补偿的方法被称为树结构的运动补偿。宏块分割与亚分割所产生的每一个亮度块都有自己的运动矢量。对于宏块中的色度值，分割方法是和亮度一样的，但是由于 4:2:0 采样的关系，色度分割块的大小是亮度分割块的一半，而且当色度块在利用运动矢量时，必须要将它各个分量除以二。

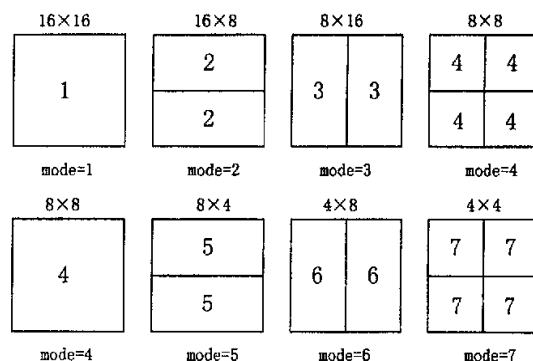


图 2.2 H.264 中的宏块划分

其次，H.264 可以达到四分之一像素的运动精度，这是通过利用整像素点的亮度值进行内插得到的。内插过程先是通过 6 抽头的滤波器来获得半像素精度，然后用线性滤波器来获得四分之一像素的精度。又由于 4:2:0 采样的关系，色度的运动精度就达到八分之一像素，这也是通过线性滤波器插值得到的。

由于运动矢量之间也有一定的相关性，所以 H.264 就利用已经编码块的运动矢量对当前未编码块的运动矢量进行预测，最后只需要编码和传输实际运动矢量与预测值的差值即可。

再次，H.264 还可以采用多参考图像（最多前向和后向各 5 帧）来进行运动预测，这样可以对周期性运动、平移封闭运动和不断在两个场景间切换的视频流有效果非常好的运动预测。使用了多参考图像，H.264 不仅能够提高编码效率，同时也能实现更好的码流误码恢复，但需要增加额外的时延和存储容量。

最后，H.264 也在 B 图像中利用后向运动预测，这和以前的标准是一致的，但不同的是 B 图像通过加权也能作为其他图像的参考图像。

## 2.2.2 新的帧内预测

在以往标准中，只采用 Intra 编码的图像称为 I 图像，在 I 图像中是把像素块的数值进行变换，这样处理的结果就导致 I 图像包含了大量的数据，压缩效

率不是很高。H.264 根据相邻像素可能有相同的性质，利用了相邻像素的相关性，采用了新的 Intra 预测这种模式。通过当前像素块的左边和上边的像素（已编码重建）进行预测，只对实际值和预测值的差值进行编码，这样就能用较少的比特数来表达 Intra 编码的像素块信息。在当前的标准中，亮度块可以有 9 种 4x4 块的 Intra 预测模式和 4 种 16x16 块的 Intra 预测模式，如图 2.3，图 2.4 所示。而色度 8x8 块的 4 种模式与亮度的 4 种 16x16 块的模式相同。

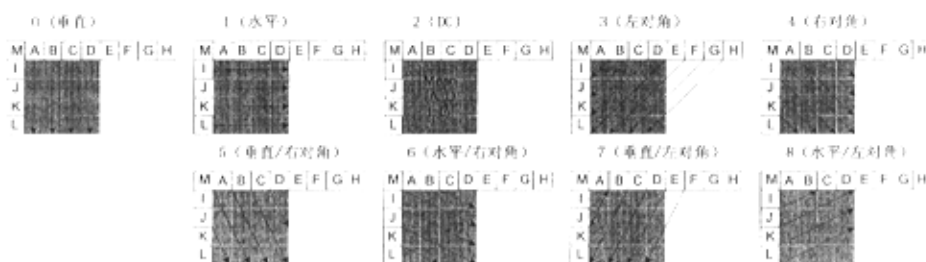


图 2.3 4x4 块的 9 种 Intra 预测模式

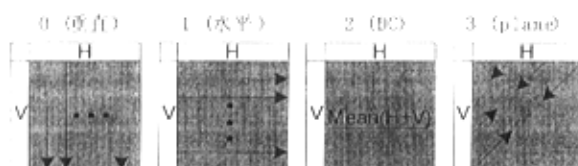


图 2.4 16x16 块的 4 种预测模式

### 2.2.3 系数变换及量化

H.264 把运动估值和 Intra 预测的残差结果从时域变换到频域，使用了类似于 4x4 离散余弦变换 DCT (Discrete Cosine Transform) 的整数变换，而不是象 MPEG-2 和 MPEG-4 那样采用 8x8 DCT 的浮点数变换。由于 H.264 使用的是以整数为基础的空间变换，具有效果好、计算快（只需加法与移位运算）、反变换过程中不会出现失配问题等优点，并且结合量化过程，保证了在 16 位计算系统中，计算结果有最大的精度且不会溢出。同时，由于变换块大小从 8x8 变为 4x4，能够减少块效应和振铃效应。根据残差系数的不同，H.264 采用了 3 种整数变换。

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$



图 2.5 H.264 的 3 种整数变换矩阵

H.264 采用了 52 个梯状量化系数，量化值的设计使得 QP (Quantization Parameter) 值每增加 1，量化步长大小就大约增加 12.5%。同时，H.264 将变换中的尺度变换计算并入量化中进行，通过巧妙的设计用移位代替了除法运算，极大地降低了运算复杂度。

## 2.2.4 扫描顺序

变换系数的读出方式也有两种：之字形 (Zigzag) 扫描和场 (Field) 扫描。如果块是按帧模式编码的，那么就使用之字形扫描；如果块是按场模式编码的，那么就使用场扫描。

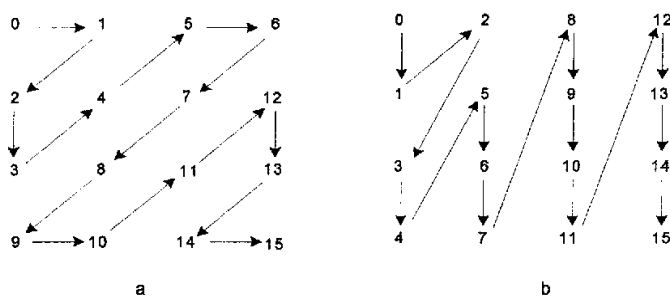


图 2.6 a) 之字形扫描 b) 场扫描

## 2.2.5 去块滤波器

为了消除块效应，提高解码图像的主观与客观质量，同时为了提供更好的参考图像，H.264 标准中引入了基于内容的去块滤波器。去块滤波是基于 4x4 块边界的，也就是说对于每一个宏块的 16x16 亮度分量，需要对其 4 条水平边界和 4 条垂直边界进行滤波，而 8x8 的色度分量则只需要对其 2 条水平边界和 2 条垂直边界进行滤波。在块边界，滤波的强度是和块的编码模式、运动矢量和残差数值相关的，而对于单个像素，基于量化系数的门限值可以取消对任意单个像素的滤波。

图 2.5 H.264 的 3 种整数变换矩阵

H.264 采用了 52 个梯状量化系数，量化值的设计使得 QP (Quantization Parameter) 值每增加 1，量化步长大小就大约增加 12.5%。同时，H.264 将变换中的尺度变换计算并入量化中进行，通过巧妙的设计用移位代替了除法运算，极大地降低了运算复杂度。

## 2.2.4 扫描顺序

变换系数的读出方式也有两种：之字形 (Zigzag) 扫描和场 (Field) 扫描。如果块是按帧模式编码的，那么就使用之字形扫描；如果块是按场模式编码的，那么就使用场扫描。

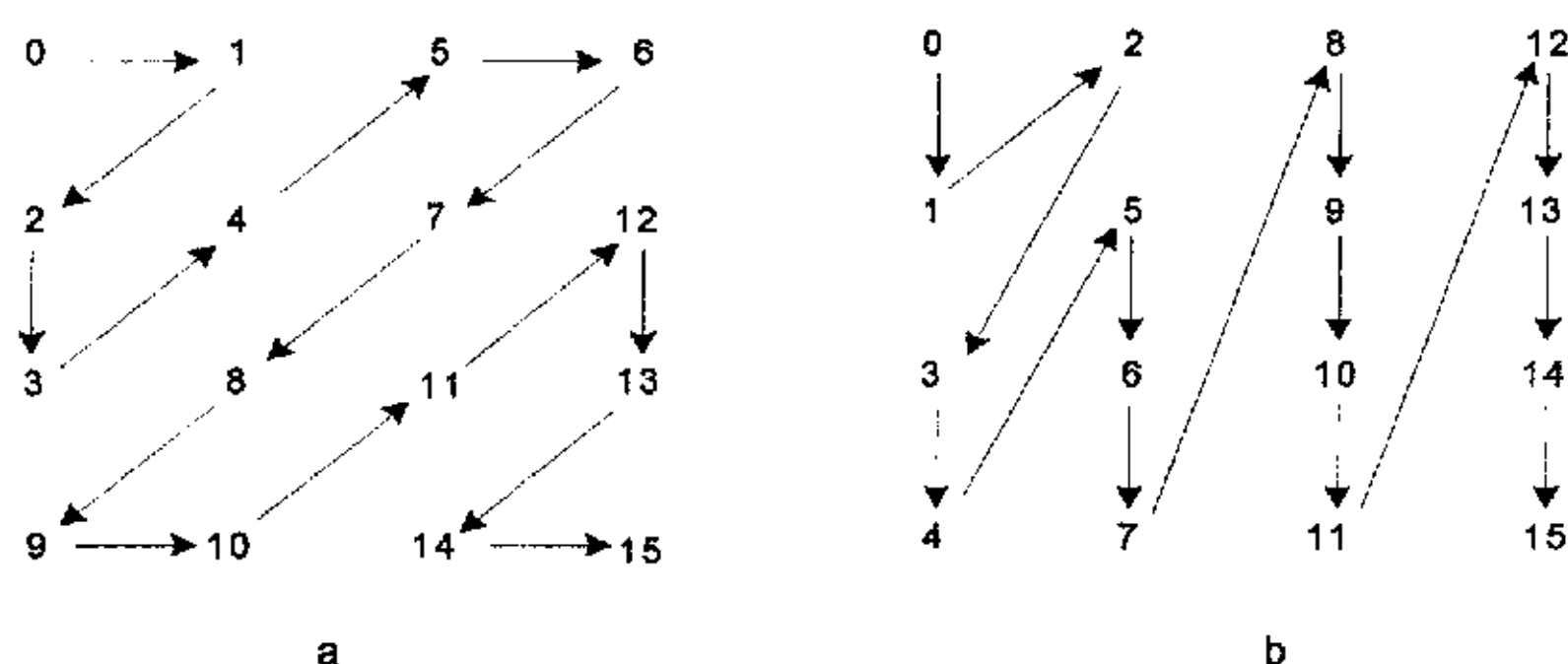


图 2.6 a) 之字形扫描 b) 场扫描

## 2.2.5 去块滤波器

为了消除块效应，提高解码图像的主观与客观质量，同时为了提供更好的参考图像，H.264 标准中引入了基于内容的去块滤波器。去块滤波是基于 4x4 块边界的，也就是说对于每一个宏块的 16x16 亮度分量，需要对其 4 条水平边界和 4 条垂直边界进行滤波，而 8x8 的色度分量则只需要对其 2 条水平边界和 2 条垂直边界进行滤波。在块边界，滤波的强度是和块的编码模式、运动矢量和残差数值相关的，而对于单个像素，基于量化系数的门限值可以取消对任意单个像素的滤波。

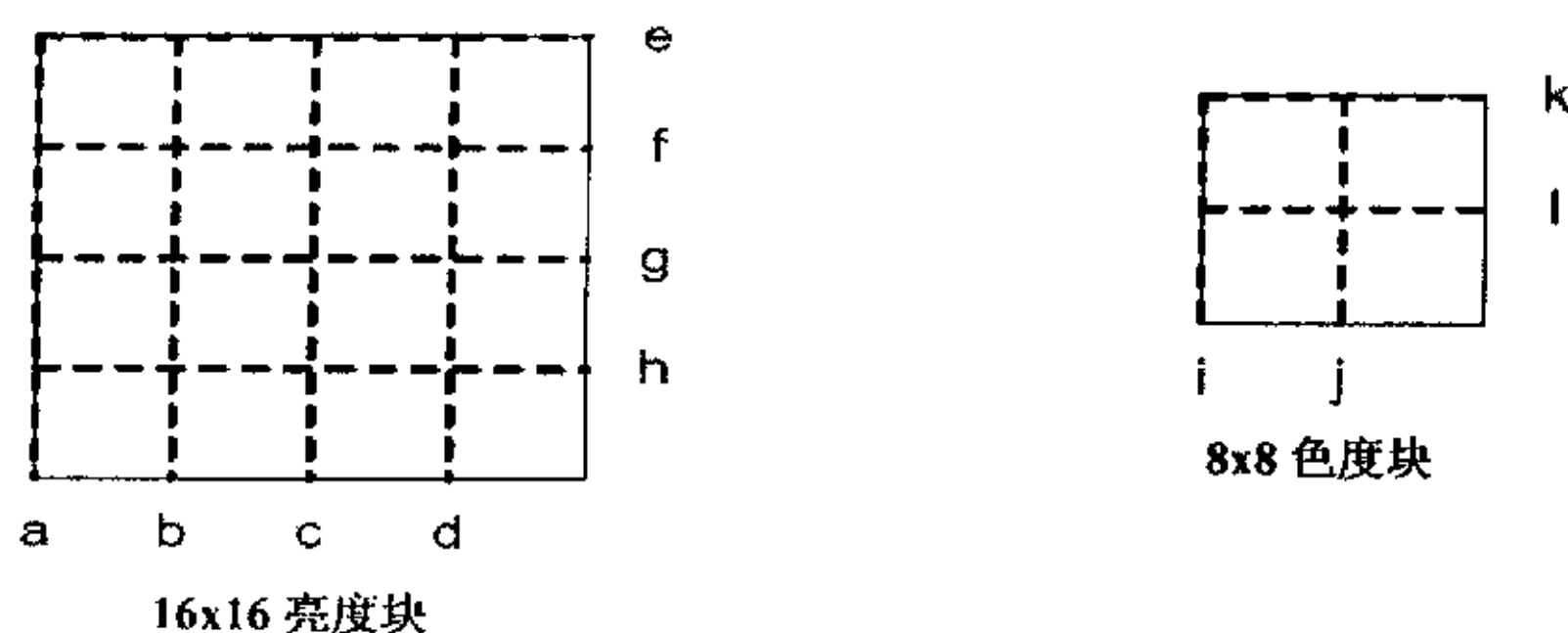


图 2.7 宏块滤波边界

## 2.2.6 熵编码

H.264 使用了两种熵编码方法，即基于上下文的自适应变长编码 CAVLC (Context-based Adaptive Variable Length Coding) 与一致变字长变码 UVLC (Universal Variable Length Coding) 相结合的编码和基于上下文的自适应二进制算术编码 CABAC (Context-based Adaptive Binary Arithmetic Coding)。

在以往的标准中，都是采用 UVLC 的，但是对于 UVLC 来说，所有的符号都采用一张仅从概率统计分布模型得出的码表。它虽然简单，但是可能会有以下的缺点：概率分布可能不是很符合实际情况；概率分布是静止的；忽略了符号相关性，没有利用条件概率；码字必须要有整数个比特。这些缺点使得 UVLC 在中高码率时压缩效果不是很好。

CAVLC 是用于对变换系数的编解码。变换和扫描后的系数有如下几个特点：经常等于 $\pm 1$ ，其他非 0 系数的变化有一定的规律性和相关性，0 系数的游程编码也具有一定特性。由于这些特点能与 CAVLC 相适应，所以 CAVLC 对变换后的系数可以进行较大的压缩。

CABAC 使编码和解码两边都能使用所有句法元素（变换系数、运动矢量）的概率模型。为了提高 CABAC 的效率，通过一种称之为内容建模的过程，使基本概率模型能适应随视频图像而改变的统计特性。内容建模提供了编码符号的条件概率估计，利用合适的内容模型，存在于符号间的相关性可以通过选择目前要编码符号邻近的已编码符号的相应概率模型来去除，不同的句法元素通常保持不同的模型。同时，由于利用了算术码字，对于每个符号来说，就用到了非整数个比特。如果给定的符号为非二进制，它将使用 UVLC 二进制树完成二进制化。

每一次二进制判决后,算术编码器使用新的概率估计进行编码,此概率估计是在前一内容建模阶段更新的。在每一次二进制化编码以后,向上调整用于刚刚编码的二进制符号的概率估计。这样,模型就保持了对实际统计特性的跟踪。

### 2.2.7 新的图像片类型

除了支持传统的图像片(Slice)类型外,H.264还支持新的码流间可转换图像片,称之为SP(Switching P)片和SI(Switching I)片。码流中包含了SP片和SI片之后,解码器能在有类似内容但码率不同的码流之间快速切换,并同时支持随机访问和快速回放模式。SP片用到了Inter预测的方法,并通过改变量化的大小以实现在不同码率的图像流之间转换。而SI片则是在由于传输错误而无法采用Inter预测的方法时,对SP片最好的近似的片。

### 2.2.8 场编码模式

在H.264中,对于一幅图像,可以按照帧的模式,也可以按照场的模式进行编码。把一幅图像拆成两场图像,其中一场采用Intra编码,而另一场则利用前一场的信息进行运动补偿编码,这样就能提高压缩效率,尤其是在存在剧烈水平方向运动的场景下。在某些情况下,图像的一些部分适合采用帧模式编码,而另一部分适合采用场模式编码,所以H.264支持在宏块级自适应的场帧编码模式转换。

### 2.2.9 网络适应层

在H.264标准中出现的网络适应层NAL类似于MPEG-2中的机制,目的是将图像片映射到传输层,利用RTP在IP网络中传输视频流,实现建立、成帧、封装、插入、逻辑通道、关断、时序和同步化等基本功能与反馈和优先级控制等特定网络问题。NAL单元不仅包含所有视频信息,它的头信息也提供传输层或存储媒体的信息,所以NAL单元的格式适合包传输网络和比特流系统。

## 2.3 H.264 的性能与复杂度分析

由于H.264采用了多种独特的运动补偿预测技术，使压缩效率大大提高[38]。例如，多种不同的大小和形状的帧间预测模式，能转换成相当于大于15%的码率节省；使用亚像素空间精度与使用整数像素空间精度相比，能产生大于20%的码率节省；使用多参考帧预测与只使用一个参考帧相比，能产生5%~10%的码率节省。此外，CABAC熵编码节省的码率通常在10%~15%之间。使用自适应环内去块状效应滤波器，在减少块状效应的同时，也显著提高了图像的主观质量。

有关研究表明，H.264 编码视频流与 MPEG-2 编码视频流相比要节省 64%的比特率，与 H.263v2 (H.263+) 或 MPEG-4 Simple Profile 编码视频流相比，平均可节省 39%的比特率，而最多可节省 50%的比特率（如图 2.8 所示）。

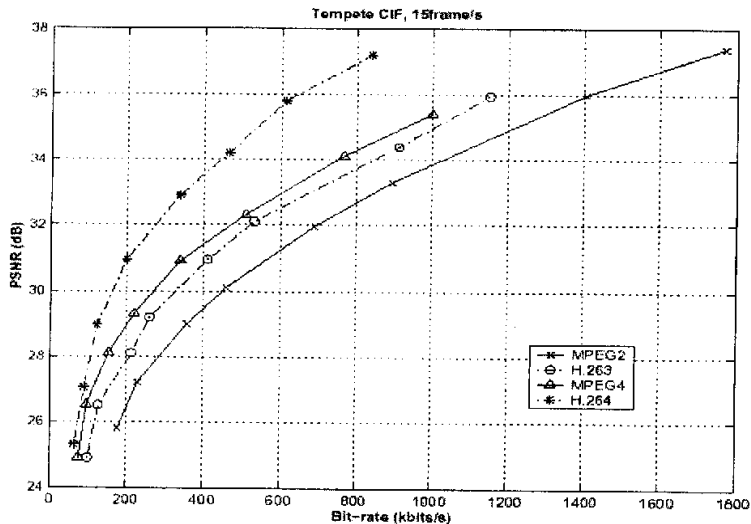


图 2.8 几种视频标准的性能比较

H.264 所独有的这些编解码技术使得该标准的视频压缩性能有飞跃性的提高，但与此同时，H.264 的复杂度也大幅度提高，与 H.263 相比，复杂度提高了 3~4 倍[42]。研究结果发现，在编码端采用多参考帧对编码效率只有有限的提高，但却是最重要的复杂度瓶颈；采用 RD-Opt 技术可以改善比特率和 PSNR 值，但是复杂度也会提高 100%至 200%；CABAC 能够节省一定的比特率，但是这就需要比基本 UVLC 更多的计算量和存储容量（最多 40%）；同样，采用双向预测技

术也有这样的效果与缺点。采用 Hadamard 变换可以显著提高 PSNR 值,但复杂度提高不大。在解码端,影响存取频率和解码速度的主要因素是双向预测技术,同时 CABAC 和 RD-Opt 也会增加少量复杂度。综合上述考虑,为了在编码效率和复杂度之间取得一个比较好的折衷,特别是着眼于实时应用的需要更多地降低编解码复杂度,我们在实际运用中不采用多帧参考、B 帧、RD-Opt、CABAC,保留 Hadamard 变换,并基于此展开算法优化的研究,在保证图像质量的前提下尽量降低算法复杂度。

根据上节的各编码算法的复杂度分析,我们用 Intel 的代码分析工具 VTune 对 H.264 参考软件 JM8.0[69]编码器的 Baseline Profile 的各模块复杂度进行了测试,测试条件如下:

参考帧为一帧,Search Range = 32, 使用 Hadamard 变换及 CAVLC, 不使用 B 帧及 RD-Opt, 测试序列第一帧为 I 帧,其余都为 P 帧, QCIF 格式。

测试结果如图 2.9 所示:

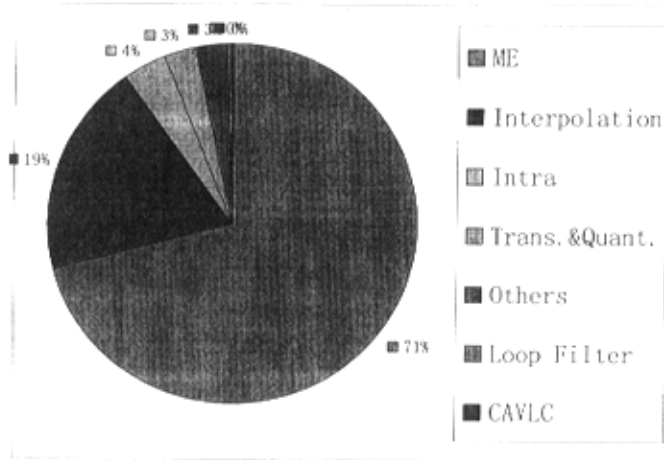


图 2.9 H.264 编码器各模块复杂度分析

从上图可看出, H.264 编码器复杂度最高的模块仍然是运动估计(ME, 包括整像素 ME 和亚像素 ME),占到整个编码器的 71%,其次是插值(Interpolation, 19%)及帧内预测(Intra, 4%)。由于差值滤波器的设计在标准中就有明确的规定,因而很难在算法上进行优化,而帧间运动预测与帧内预测则存在着很大的优化空间,因而本文以下章节就针对预测部分提出了相应的快速算法,对原有的运动估计及帧内预测算法进行优化,在保证图像质量的前提下显著地降低编码器复杂度。

## 第三章 快速整像素运动搜索

在 H.264 编码器中,运动估计(Motion Estimation)是复杂度最高的模块,占到整个编码器的 70%以上,所以必须对运动估计算法进行优化。运动估计由两部分组成:整像素运动估计和亚像素运动估计。其中,整像素运动估计占了 90%以上的计算量,因此我们首先对其进行算法优化。在本章中,我们首先介绍以往所提出的快速帧内预测算法及 H.264 所采纳的一种快速算法,然后着重介绍我们在此基础上提出的提前中止算法,能在以往快速算法的基础上进一步减少整像素运动估计的复杂度。

### 3.1 快速整像素运动搜索算法综述

通常,有两种方式可以减少运动搜索的运算量:第一是采用简化的搜索模式来降低算法复杂度,到目前为止,这方面已经提出了大量算法,包括三步法搜索(Three Step Search, STT)[43]、二维对数搜索(2-D Logarithmic Search, 2D LOGS)[44]、基于块的梯度下降搜索(Block-based Gradient Decent Search, BBGDS)[45]、六边形搜索(Hexagon-based Search, HBS)[46]、四步法搜索(Four Step Search, FSS)[47]、改进的预测带状搜索(Enhanced Predictive Zonal Search, EPZS)[48]、混合的非对称多六边形搜索(Hybrid Unsymmetrical-cross Multi-Hexagon-grid Search, UMHExagonS)[49]等。这些算法提出了各种各样的搜索模式来取代全搜索,取得了良好的效果。另一种方式是提前结束运动搜索的过程。通过提前预测出 DCT(Discrete Cosine Transform)系数将被量化为零的块,一些算法[50][51][52]被提出来提前跳过运动搜索、变换及量化,以减少运算量。同时,在[52]中提出的零矢量检测算法通过当前块的 SAD(Sum of Absolute Difference)与给定阈值比较而提前判定零运动矢量块,以跳过后续搜索点的计算。这些提前中止算法能与其他快速算法结合使用,进一步降低运动搜索的复杂度。但是这些算法都是针对以前的视频编码标准,如 H.263 提出的,并不适用于最新的视频编码标准 H.264,这是因为 H.263 等以前的标准只包含两种块大小 8x8 及





### 3.2.1 运动矢量预测

运动矢量预测是快速运动估计算法中的一项重要技术，通过对运动矢量的预测而决定搜索的起始位置，能迅速找到搜索窗内的最佳匹配块。如[53]所述，UMHexagonS 包括以下几种：

#### (1) 周围块均值预测

周围块预测用当前块的左方、上方、右上方(或左上方)块的运动矢量 MV(Motion Vector)来预测当前块的运动矢量，如图 3.2 所示。

$$MV = median(MV\_A, MV\_B, MV\_C) \quad (3.1)$$

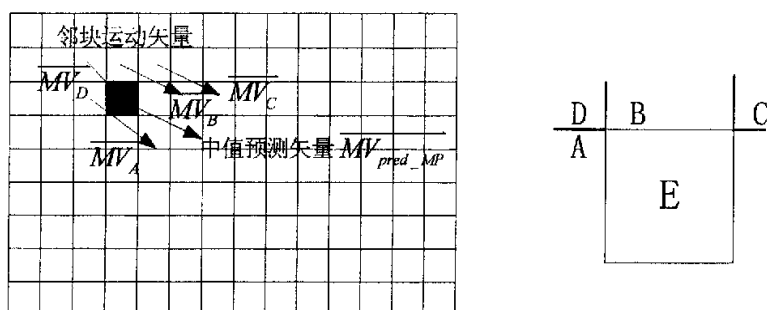


图 3.2 运动矢量的周围块预测

#### (2) UpLayer 预测

在 H.264 中，有 16x16、16x8、8x16、8x8、8x4、4x8、4x4 等 7 种不同的块大小(模式)，如图 3.3 所示。不同块大小之间的运动矢量存在很强的相关性。因此可以先搜索大的块(mode 1,2,3)，再搜索小的块(mode 4,5,6,7)，这样就可以用找出的大块的运动矢量来预测其中小块的运动矢量，如图 3.3、图 3.4 及式 3.2 所示。

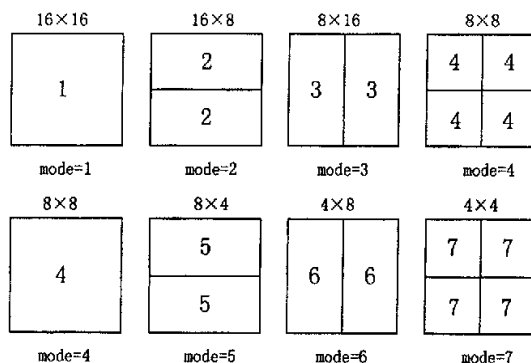


图 3.3 H.264 中的 7 种预测模式

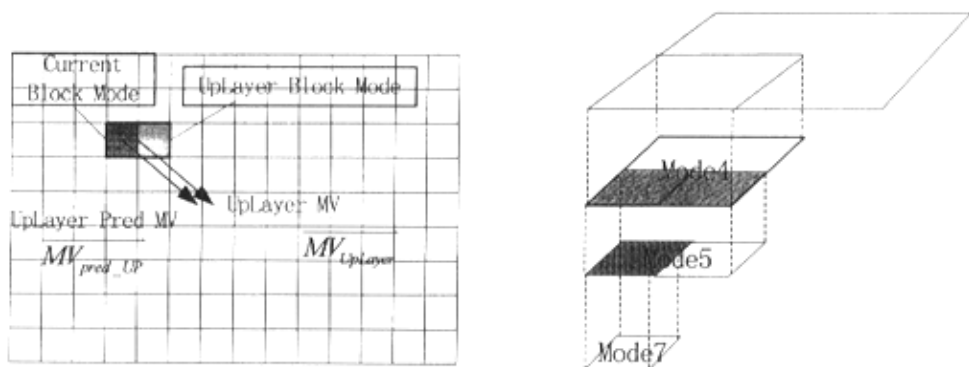


图 3.4 运动矢量的空间 Up-Layer 预测

$$MV_{pred\_UP} = MV_{UpLayer} \quad (3.2)$$

### (3) 前帧对应块预测

对通常的视频序列来说，物体的运动总是连续的，因此在时域中的运动矢量存在的很强的相关性，因此可以利用前一帧对应块的运动矢量来预测当前块，如图 3.5 所示。

$$MV_{pred\_CP} = MV_{CB} \quad (3.3)$$

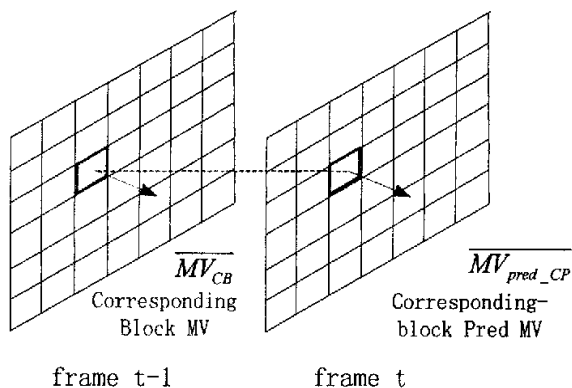


图 3.5 运动矢量的前帧对应块预测

在上述预测矢量中找出最小 cost 所对应的预测矢量，作为下一步搜索的起始位置。

## 3.2.2 非对称的十字形搜索

根据观察，对自然视频序列来说，水平方向的运动要大于垂直方向的运动，

因此能通过非对称的十字形搜索来近似找到最佳运动矢量。如图 3.1 中 step 2 所示，十字形搜索的横向长度为  $W$ ，纵向为  $W/2$ 。搜索点之间的步长为 2。找到的最佳匹配点将作为下一步搜索的起始位置。

### 3.2.3 多层六边形搜索

这部分由两个步骤组成：首先是以起始位置为中心的  $5 \times 5$  的小范围内的全搜索，如图 3.1 中 step3-1 所示；然后进行多级的六边形搜索，即 step3-2。这儿采用六边形的搜索模式，如图 3.6 所示，与四边形相比，六边形能覆盖更大的搜索区域。六边形搜索由内至外逐层展开，选出的最匹配块位置作为下一步搜索的起始位置。

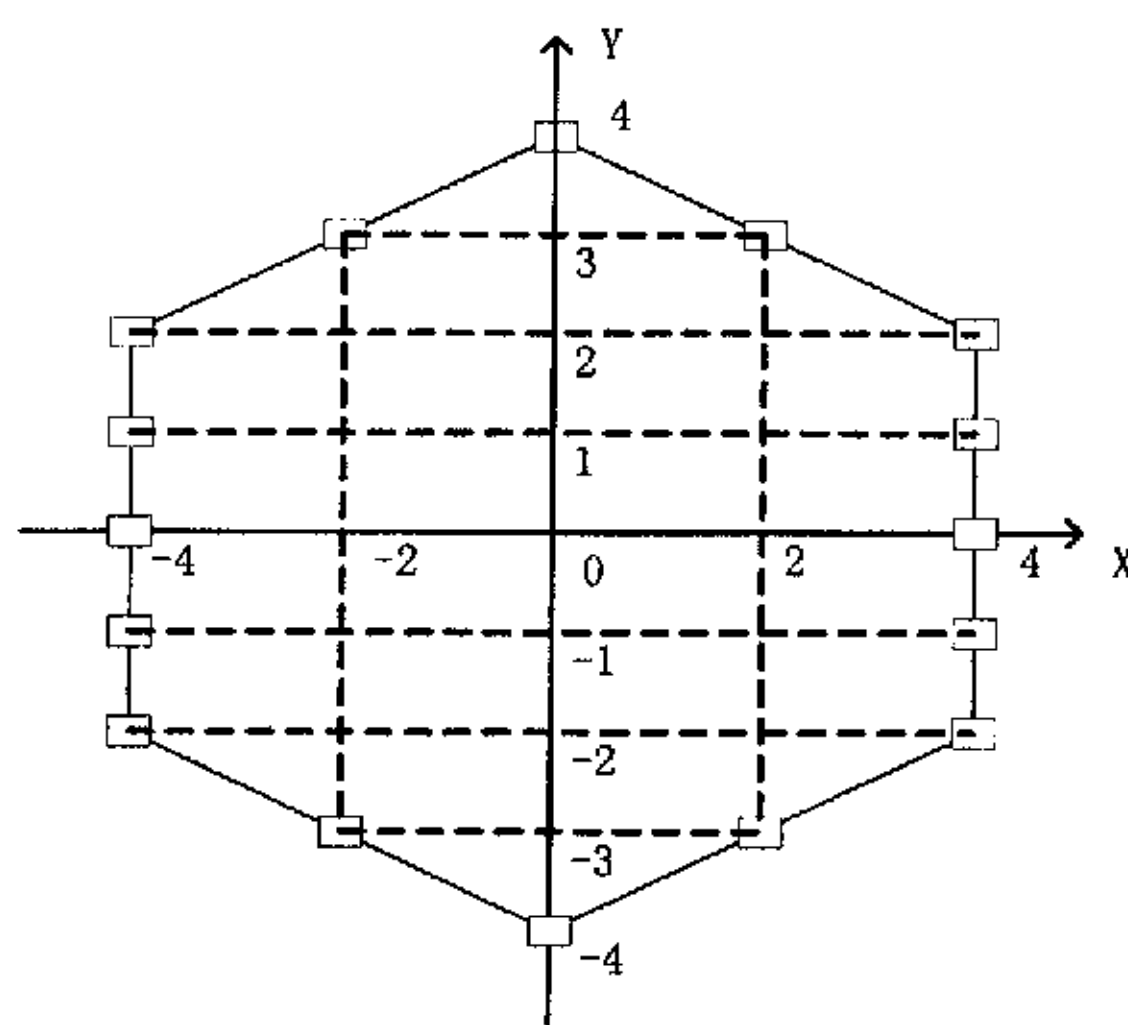


图 3.6 UMHexagonS 中采用的六边形搜索

### 3.2.4 扩展的细化搜索

距搜索起点的距离不同，多级搜索所得到的运动矢量的精度也不同。当运动矢量是在距起始位置较远的区域获得的时候，通常其精度较低，因而在此基础上进一步采用细化的六边形搜索，如图 3.1 Step4 所示。先用半径为 2 的六边形搜索，然后采用半径为 1 的小六边形，搜索到最小匹配误差 (MBD, minimum block distortion) 点位于新形成的六边形的中心为之。

同时，对于小的预测模式(如模式 7)，因为其预测矢量的精度较高，因此可以跳过十字形搜索和多级六边形搜索，直接进行细化的六边形搜索。

实验证明,与全搜索相比,UMHexagonS 能降低 90%的整像素运动搜索计算量,而 PSNR 的下降小于 0.1db,码率几乎无变化,是一种针对 H.264 的非常有效的整像素快速算法。

### 3.3 提前中止算法——VBBMD

在[52]中提出的零矢量检测算法(ZMD, Zero Motion Detection)是一种非常有效的提前中止算法,它通过当前块的 SAD(Sum of Absolute Difference)与给定阈值比较而提前判定零运动矢量块,以跳过后续搜索点的计算,能与其他快速算法结合使用,大大降低运动搜索的复杂度。但是这个算法是针对以前的视频编码标准如 H.263 提出的,并不适用于最新的视频编码标准 H.264,这是因为 H.263 等以前的标准只包含两种块大小 8x8 及 16x16,而在 H.264 中包含从 16x16 到 4x4 等 7 种不同的块大小。受 ZMD 思想的启发,我们提出了多种块大小的最佳运动检测(Variable Block-size Best Motion Detection, VBBMD)算法[54],与之前 ZMD 算法主要由以下三方面的区别:

- (1) VBBMD 中采用的阈值是根据不同块大小的检测准确率及运算量的下降来共同确定的,这使得阈值选择更加准确;
- (2) 在 VBBMD 中检验预测位置而不是零矢量位置,这样能跳过更多的搜索点;
- (3) 在 VBBMD 中对 16x16 的块采用双重阈值,低的阈值用来跳过其中小块的运动搜索。

所提算法能与任何快速整像素搜索算法结合使用,且不增加额外的计算开销。

下面就详细介绍我们所提出的 VBBMD 算法。所有的讨论都是基于 P 帧及整像素搜索展开。

#### 3.3.1 不同块大小的零运动矢量检测(VBZMD)

通常,大多数视频序列经运动预测后都存在着大量的块其运动矢量为零,如表 3.1 所示。在表中,Model~Mode7 表示在 H.264 中不同块大小(16x16~4x4)的 7 种预测模式。可以发现,在运动量从低到高的不同视频序列中,有 30.71%~98.03%的块具有零运动矢量。如果能预测出这些零矢量块 ZMB(Zero Motion

Block), 我们就能提前中止运动搜索过程, 从而减少运算量。基于此, 我们提出了不同块大小的零运动矢量检测(Variable Block-size ZMD, VBZMD)算法。

表 3.1 7 种预测模式的零运动矢量块比例(%) (QP = 32)

Sequence	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Akiyo	98.03	97.51	97.18	97.28	97.14	96.47	97.04
Salesman	96.80	95.92	95.69	95.15	94.78	94.13	94.33
News	91.90	91.72	91.98	92.35	91.93	91.31	91.78
Silent	85.01	84.65	84.15	85.01	84.75	84.23	85.06
Coastguard	32.44	33.51	32.80	30.91	31.40	30.71	31.06
Foreman	43.97	44.15	43.57	44.25	44.81	43.73	45.42

在之前的 ZMD 算法中, 当一个块的 SAD 值小于给定的阈值时, 它就能被认为是 ZMB。在 H.264 中, cost 函数  $J$  取代 SAD 成为衡量匹配块预测误差的标准。

$$J(m, \lambda) = SAD(s, c(m)) + \lambda R(m - p) \quad (3.4)$$

其中  $m = (m_x, m_y)^T$  是当前的运动矢量,  $p = (p_x, p_y)^T$  是预测矢量,  $\lambda$  拉格朗日常数,  $R(m-p)$  是编码 MV 所需的比特数。

如果一个块运动矢量为零, 那么在 MV(0,0) 处其 cost 较小。因此对 7 种不同的块大小, 我们分别定义阈值  $THZ_i (i=1, \dots, 7)$ 。在运动搜索中, 首先搜索 MV(0,0), 如果 MV(0,0) 的 cost 满足式 (3.5), 则当前块可被认为是 ZMB, 就可以跳过下面的运动搜索。

$$J_i < THZ_i, \quad \text{for } i = 1, \dots, 7 \quad (3.5)$$

如何决定不同块大小的阈值是 H.264 零运动矢量检测的关键。考虑到在实际运用中, 保证图像质量比复杂度更为重要, 因此, 我们把检测准确率作为阈值选取的标准。其中, 检测准确率表示当一个块的 cost 小于给定阈值时, 其 MV 为零的概率。很显然, 阈值选取得越大, 有更多的 ZMB 被检测出来, 因而能跳过更多的搜索点。然而, 与此同时也有更多的块被误判为 ZMB, 导致更大的图像质量的下降。我们的方法是通过统计准确率获得一系列参考阈值, 然后实际实验测得的图像质量和复杂度之间的折衷来选取一组最合适的阈值。很多序列的统计结果表明采用相同的阈值时, 运动小的序列的准确率和检出率都要高于运动大的序列。因为 Forman 能代表作为运动大的序列的代表, 因此我们能根据 Forman 的统计结果来选择阈值, 并应用到其他序列。表 3.2 给出了所得到的候选阈值。

表 3.2 Forman 不同准确率所对应的阈值(QP = 32)

Accuracy	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7
80%	600	550	570	400	315	300	250
75%	960	650	660	480	360	350	270
70%	1650	920	890	600	410	400	300
65%	2270	1200	1160	700	470	460	350
60%	2800	1500	1430	880	530	530	400

在我们的方法中, 如果一个块与参考帧零矢量位置的匹配块的差值  $\text{cost}$  小于相应的阈值, 那么我们就认为该块的最佳运动矢量为零, 跳过当前块的运动搜索而直接进入同一个宏块的下一个块的运动搜索。如果当前块是宏块的最后一个块, 则跳入下一个模式的运动搜索。当所有模式都计算过后, 对应于最小  $\text{cost}$  的那个模式就被选为宏块的最佳模式。特别的, 当为模式 1 时, 整个宏块只包含一个块,  $16 \times 16$ , 如果其零运动矢量块的  $\text{cost}$  足够小, 则可以认为不仅  $(0,0)$  是当前块的最佳矢量, 而且模式 1 是当前宏块的最佳模式。因此, 我们为模式 1 定义一个额外的小些的阈值  $\text{THS}$ 。如果模式 1 的零运动矢量  $\text{cost}$  小于  $\text{THS}$ , 整个当前宏块的运动搜索就可跳过。VBZMD 的算法流程如下所示:

```

Assume the current macroblock is MB(x,y)

For each mode  $i$  ( $i = 1, \dots, 7$ ) of the MB(x,y)
  For each block in the macroblock
    Calculate the cost at (0,0)
    If  $\text{cost} < \text{THZ}_i$ 
      Set  $\text{MV} = (0,0)$ 
      If  $\text{mode} = 1$  and  $\text{cost} < \text{THS}$ 
        Set best mode = 1
        Exit two loops
      End if
      Exit one loop
    Else
      Do the normal motion search process
    End if
  End for
End for

```

在实验中采用的 H.264 编码器是 JVT 参考软件 JM6.1e[69]。我们把提出的提前中止算法与 H.264 标准采纳的快速整像素预测算法 UMHexagonS 相结合。因为在实验中与模式 1 相对于的阈值 800(准确率为 78%)能取得很好的效果, 因此



我们取其作为 THS。如表 3.3 所示, Akiyo 代表运动较小的序列(QCIF), 最多能跳过有 93.47% 的搜索点(PPMB, search points per macroblock), 而平均 PSNR 的下降不超过 0.05db, 码率略有上升。即使对于包含较大的脸部运动和镜头移动的 Foreman 序列, 所提出算法仍能取得显著的计算量下降且 PSNR 损失很小。这里搜索点的计算是基于宏块来进行的, 比如, 一个模式 4(8x8 块)的搜索点在这种计算方法中记做 1/4。可以看到, 在这里准确率 65% 所对应的那组阈值能提供在复杂度和图像质量间提供一个很好的折衷。

表 3.3 VBZMD 取不同阈值的实验结果 (QP = 32)

Sequence	Method	Accuracy	PSNR (dB)	PPMB	Bit-rate (kbps)
Akiyo	Fast ME		35.22	63.86	14.52
	Fast ME +VBZMD	80%	35.18	29.02	14.65
		75%	35.20	21.41	14.67
		70%	35.19	7.96	14.70
		65%	35.18	5.17	14.59
		60%	35.17	4.17	14.67
Foreman	Fast ME		32.90	160.48	79.34
	Fast ME +VBZMD	80%	32.87	124.97	79.00
		75%	32.86	112.85	79.55
		70%	32.82	88.75	80.49
		65%	32.77	64.77	82.19
		60%	32.75	49.39	84.73

### 3.3.2 不同块大小的最佳运动矢量检测(VBBMD)

虽然 VBZMD 能减少相当多的运动估计运算量, 但我们还能进行改进以取得更好的效果。在 H.264 中, 当前块的左方、上方和右上方(左上方)块的运动矢量可用来预测当前块的运动矢量, 如图 3.7 所示。因为周围块的运动矢量在时间上有很强的相关性, 因此该预测的矢量很可能就是当前块运动搜索后的最佳矢量。为了叙述方便, 我们定义最佳矢量即为周围块的运动矢量预测的块为预测矢量块 PVB(predicted vector block)。如表 3.4 所示, 不同序列中 PVB 出现的概率要大大高于表 3.1 所示的 ZMB。特别的, 对运动较大的序列, 如 Costguard 及 Foreman, 与 ZMB 相比, PVB 的比例要高出 23%~122%。在另一方面, 另一个实验表明:

对绝大多数序列来说, 超过 97% 的 ZMB 同时也为 PVB。这些发现启发我们用预测矢量取代零矢量来预测当前块的最佳矢量。此外, 在 H.264 中, 是当前矢量与预测矢量的差值, 而不是预测矢量, 被编码和传输。因此, 采用预测矢量来预测最佳矢量也能节约编码的比特数。

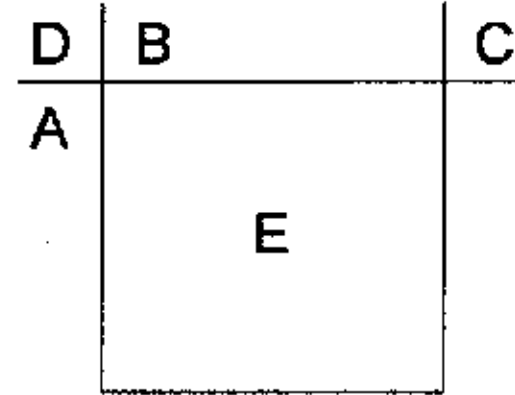


图 3.7 当前块的周围块

表 3.4 不同视频序列中预测矢量块(PVB)的比例(%)

Sequence	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Akiyo	98.05	97.51	97.23	97.37	97.27	96.61	97.24
Salesman	96.84	95.71	95.46	95.40	95.11	94.54	95.03
News	91.80	91.70	91.66	92.83	92.71	92.22	93.16
Silent	85.09	85.04	84.57	86.32	86.80	86.36	88.28
Coastguard	58.68	62.19	62.41	64.32	66.17	67.09	69.04
Foreman	54.19	56.26	55.46	57.79	60.00	59.15	63.15

基于前面所述的 VBZMD, 我们又提出了 VBBMD(Variable Block-size Best Motion Detection)算法。其核心的思想是将预测矢量的 cost 与给定阈值比较, 如果小于阈值, 预测矢量就被认为是当前块的最佳矢量, 因而可跳过剩下的运动搜索。

类似地, 如何根据不同的块大小来选择阈值就成为了问题的关键。采用与 VBZMD 相同的方法, 我们可根据不同的准确率获得几组候选阈值。VBBMD 的算法流程几乎与 VBZMD 一样, 唯一的区别是计算预测矢量而不是零矢量。我们以候选阈值来编码各种视频序列来选出最佳的一组。结果表明准确率为 66% 的一组阈值(2500, 1500, 1400, 920, 625, 570, 500)能提供最佳的 tradeoff。另外, THS 仍选为 800, 因为其效果良好。表 3.5 给出了采用最佳阈值时 VBBMD 算法的实验结果。与表 3.3 中的 VBZMD 相比, 我们能很明显的看到有更多的搜索点被跳过, 而 PSNR 保持几乎一致。这证明了 VBBMD 的算法效率较 VBZMD 更优。

表 3.5 取最佳阈值时 VBBMD 的实验结果(QP=32)

Sequence	Method	PSNR (dB)	PPMB	Bit-rate (kbps)
Akiyo	Fast ME	35.22	63.86	14.52
	Fast ME+VBZMD	35.18	5.17	14.59
	Fast ME+VBBMD	35.18	4.17	14.59
Foreman	Fast ME	32.90	160.48	79.34
	Fast ME+VBZMD	32.77	64.77	82.19
	Fast ME+VBBMD	32.77	30.12	82.34

以上的讨论都是基于 QP 值为 32 的, 然而, 实验证明不同的 QP 值会导致不同的阈值。这是因为匹配误差函数与 QP 值相关。在式(3.4)中,  $J$  与  $\lambda$  直接相关, 这里  $\lambda$  是根据 QP 值计算出来的一个常数。因此, 我们可以根据 QP 值来调整阈值。当一个块被认为是 PVB 时, 式(3.4)中当前运动矢量  $m$  和预测矢量  $p$  的最大差值为  $3/4$  像素, 需要 10bits 来编码。所以, 我们可近似地把  $R(m-p)$  设为 10。我们定义 QP=32 时的阈值为基准阈值, 另外 QP 值所对应的阈值可以以基准阈值加上  $(\lambda_{QP} - \lambda_{32}) \times 10$  来获得。

### 3.4 实验结果

我们将 VBBMD 与快速整像素搜索算法 UMHexagonS 结合使用, 应用于 H.264 参考软件 JM6.1e[69]。搜索范围设为 32, 参考帧为一帧。在实验中采用 CAVLC 熵编码, 未使用 RD-Opt。

为检验所提出算法在不同实验条件下的性能, 我们采用运动从小到大的六个不同序列, 分别是 Akiyo、Salesman、News、Silent、Coastguard 和 Foreman。所有序列都是 QCIF 格式, 30 帧/秒。只有第一帧为 I 帧, 其余均为 P 帧。另外, 为检验算法在不同码率下的效果, 在实验中采用了 3 个不同的 QP 值 28、32、36。如上节所述, VBBMD 中采用的阈值将根据 QP 值作调整。

为了叙述方便, 我们在表 3.6 的结果中给出了 PSNR 的增加  $\Delta\text{PSNR}$ , 平均 PPMB 的减少率  $\Delta\text{PPMB}$  及码率的上升  $\Delta\text{Bit-rate}$ 。分析结果我们可以发现, 所提出的算法在不同的实验条件下均有着很高的效率。与只采用快速算法 UMHexagonS 相比, 所提出的算法能进一步减少 76.10%到 94.12%的整像素搜索点, 而平均 PSNR 只下降了 0.06 dB, 码率略有上升。

表 3.6 VBBMD 在不同实验条件下的结果

QP	Performance	Akiyo	Salesman	News	Silent	Coastguard	Foreman
28	$\Delta$ PSNR (dB)	-0.05	-0.02	-0.06	-0.04	-0.05	-0.12
	$\Delta$ PPMB (%)	94.12	90.22	83.30	81.95	87.67	81.76
	$\Delta$ Bit-rate (%)	0.04	1.60	1.33	2.42	3.72	4.60
32	$\Delta$ PSNR (dB)	-0.03	-0.02	-0.06	-0.04	-0.06	-0.13
	$\Delta$ PPMB (%)	93.39	91.07	83.48	83.46	86.37	80.61
	$\Delta$ Bit-rate (%)	1.03	0.80	1.26	2.95	4.63	3.43
36	$\Delta$ PSNR (dB)	-0.05	-0.06	-0.08	-0.04	-0.06	-0.13
	$\Delta$ PPMB (%)	92.63	90.43	82.17	83.72	77.46	76.10
	$\Delta$ Bit-rate (%)	0.66	0.37	1.41	-0.07	3.00	3.45

### 3.5 本章小结

在本章中，首先分析了各种快速整像素运动搜索算法，然后根据 H.264 编码标准的特点，选择了标准所采纳的一种效率较高的简化搜索模式快速算法 UMHexagonS 作为我们的整像素优化算法。然后，在针对 H.263 的零矢量检测算法 ZMD 的基础上，根据 H.264 的特点提出了一种有效的提前中止算法——多种块大小的零矢量检测 VBZMD。该算法通过计算一个搜索点来预测最佳运动矢量，因而能提前中止一部分运动搜索过程，跳过与之相对应的计算，因而进一步减少整像素运动搜索的复杂度。最后在 VBZMD 基础上进一步优化，以检测预测矢量取代零矢量，提出了最佳运动检测算法 VBBMD。与单独采用 UMHexagonS 相比，VBBMD 能进一步减少 76%~94% 的整像素搜索点，而平均 PSNR 只下降了 0.06dB。将 VBBMD 与 UMHexagonS 相结合，与全搜索相比，能减少 96% 以上的整像素运动搜索，而 PSNR 只有略微的下降。

## 第四章 快速亚像素运动搜索

运动搜索分为两个步骤：整像素运动搜索和亚像素运动搜索。通常，整像素搜索在运动搜索中占据了绝大多数的计算量，因而是优化的重点，已有大量算法被提出以减少整像素搜索的复杂度。特别是如第三章所示，采用快速运动搜索算法 UMHexagonS 和提前中止算法 VBBMD 后，整像素搜索的计算量大大降低，从表 3.6 可以看出，以 Akiyo 和 Foreman 为例，平均每个宏块的 7 种模式的整像素搜索点已分别降为 4.17 和 30.12 个，平均每个模式的搜索点为 0.60 和 4.30 个，计算复杂度已大为降低。而与此同时，H.264 采用的 1/4 像素精度的亚像素运动搜索其全搜索需要计算 16 个亚像素点，其计算量已远远超过优化后的整像素运动估计，成为整个运动估计模块的计算瓶颈，因此必须对其进行优化以降低复杂度。本章就首先介绍了 H.264 中的亚像素全搜索算法，并对各种快速算法进行了介绍，最后在此基础上提出了一种最为有效的快速算法。

### 4.1 亚像素全搜索算法

在运动估计中，每一个块在做完整像素搜索后，以找出的最佳整像素位置为中心进行细化的全搜索亚像素运动估计，获得最终的运动矢量。H.264 采用分层的 1/4 像素精度的亚像素搜索算法，如图 4.1 所示。主要步骤如下所示：

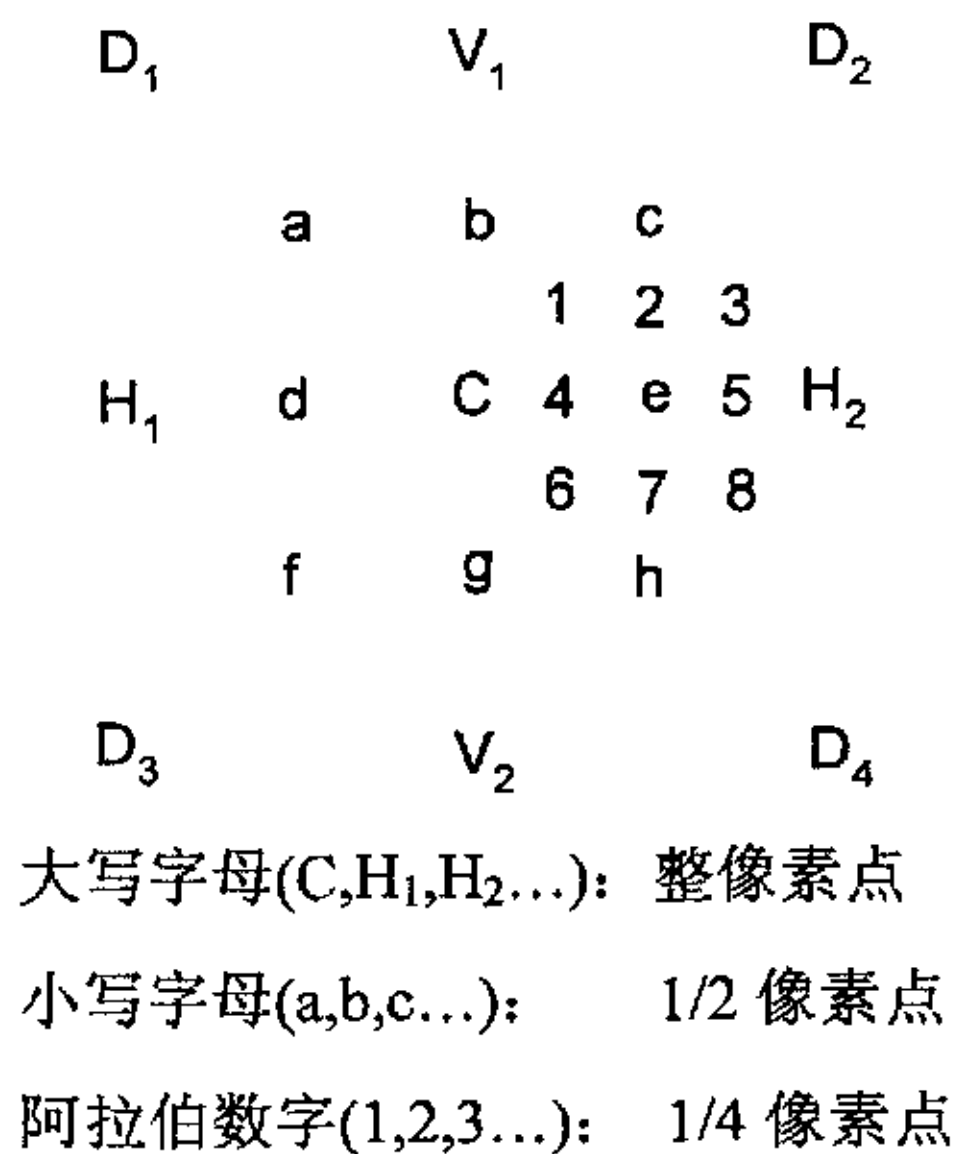


图 4.1 亚像素搜索的典型过程

- (1) 进行整像素运动估计以获得最佳整像素运动矢量, 假设位于  $C$ ;
- (2) 计算最佳整像素位置  $C$  周围的 8 个  $1/2$  像素位置  $a \sim h$ , 找到最佳的  $1/2$  像素运动矢量, 假设位于  $e$ ;
- (3) 计算最佳  $1/2$  像素位置  $e$  周围的 8 个  $1/4$  像素位置  $1 \sim 8$ , 找到最佳的  $1/4$  像素运动矢量;
- (4) 选择对应最小失真的运动矢量为最终的最佳运动矢量。

与整像素运动搜索时计算块匹配失真的方式不同, 亚像素搜索在计算  $\text{cost}$  值时又增加了 Hadamard 变换, 以获得更准确的匹配失真比较。因此最佳整像素点  $C$  的  $\text{cost}$  值在亚像素运动估值时必须重新计算, 再加上 8 个  $1/2$  像素及 8 个  $1/4$  像素, 全搜索亚像素运动估计需计算 17 个像素位置。

为获得这些亚像素值, 6-tap 滤波器(1,-5,20,20,-5,1)/32 被用来生成  $1/2$  像素插值,  $1/4$  像素插值由线性滤波器产生。

## 4.2 快速亚像素运动搜索算法综述

### 4.2.1 快速算法综述

为降低亚像素运动搜索的复杂度, 不少快速算法被提出, 它们共同的做法就是简化搜索模式。通过假设误差曲面是单调变化的, [55]中提出两种简化的搜索模式来细化最小误差点周围的运动矢量。类似地, 根据在预测区域  $\text{cost}$  函数是光滑的凸函数, [56]中提出一种基于抛物面预测的亚像素搜索(PPFPS, Paraboloid Prediction based Fractional Pixel Search), 只需计算最佳整像素周围的 3 个  $1/2$  像素点及最佳和次佳  $1/2$  像素位置的 3 个  $1/4$  像素点。但是, 所有这些算法都需要用到最佳整像素点周围整像素的  $\text{cost}$  值, 因而, 这些方法只能与满足这些限制条件的整像素运动搜索算法结合使用, 其应用有很大的局限性。基于中心的亚像素搜索(CBFPS, Prediction-based Directional Fractional Pixel Search)[49]通过直接计算预测位置并在周围进行细化搜索来解决了这个问题, 取得了很好的效果, 并



被 H.264 标准采纳，下节就详细加以介绍。此外，[57][58]提出一些提前中止算法，当  $\text{cost}$  小于给定阈值时就结束亚像素搜索过程。

#### 4.2.2 H.264 采纳的快速亚像素搜索算法——CBFPS

对亚像素运动搜索来说，可以假设在搜索窗内的块匹配误差曲面是单峰分布的，这样随着向全局最小点的逼近其匹配误差是单调下降的，[49]、[50]中的实验证明了这种假设。如图 4.2 所示，一个 1/8 像素精度搜索匹配误差的曲面是一个典型的光滑单峰曲面分布。

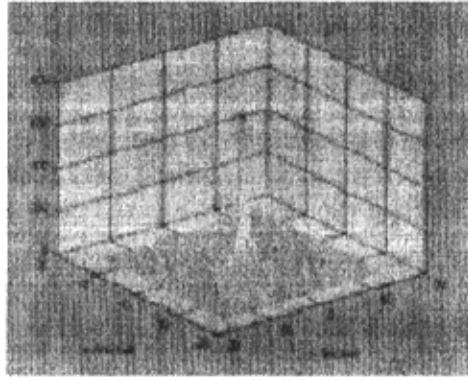
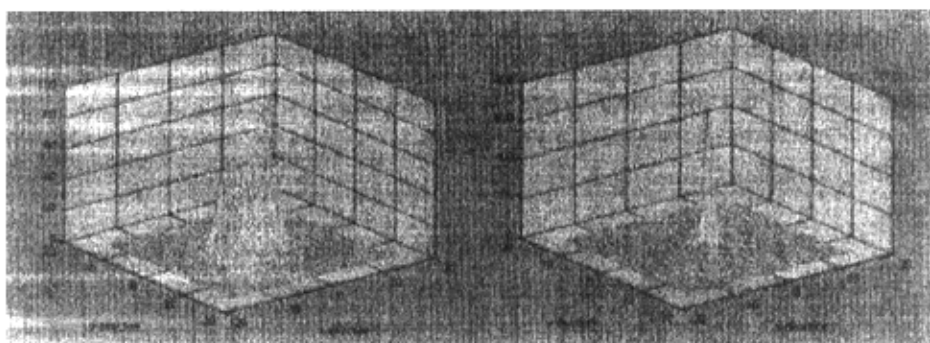


图 4.2 亚像素搜索的匹配误差曲面(1/8-pel)

基于以上结论，[49]提出了 Prediction-based Directional Fractional Pixel Search(CBFPS)。假设预测矢量为  $\text{pred\_mv}$ ，用亚像素单位来表示，其包含两部分信息：整像素预测矢量和亚像素预测矢量。可以用下式来提取出亚像素预测矢量。

$$\text{frac\_pred\_mv} = (\text{pred\_mv} - \text{mv}) \% \beta \quad (4.1)$$

其中  $\text{mv}$  是找到的整像素运动矢量，也为亚像素单位。 $\%$ 是取模操作，当为 1/4 像素精度时  $\beta=4$ ，1/8 像素精度时  $\beta=8$ 。图 4.3 表明了预测亚像素矢量和实际运动矢量之间的强相关性。图 4.3(a)给出了实际矢量对(0,0)的分布，图 4.3(b)给出了实际矢量对预测矢量的分布，很显然可以看出与(0,0)相比，预测矢量更集中于由全搜索所找到的实际矢量，因此在预测位置周围进行搜索能更容易地找到最佳矢量。



(a)

(b)

图 4.3 (a) 全搜索找到的实际运动矢量对(0,0)的分布; (b) 实际运动矢量对预测矢量的分布

菱形搜索在很多整像素快速算法中被证明是简单且有效的, 因此在 CBFPS 中采纳其作为搜索模式。

图 4.4 给出了 CBFPS 的图例, 其搜索过程为:

- (1) 以式(4.1)获得当前块的预测亚像素矢量(Pred\_x, Pred\_y);
- (2) 计算(0,0)及(Pred\_x, Pred\_y)位置的匹配误差, 匹配误差小的位置被选为下一步的起始位置;
- (3) 计算起始位置周围四个像素位置, 如果最小块匹配失真 MBD(Minimum Block Distortion)点位于菱形的中间, 跳到(4); 否则, 选择这次搜索中 MBD 点作为下次菱形搜索的中心位置, 再次进行菱形搜索;
- (4) 选择 MBD 点所对应的运动矢量为最佳亚像素运动矢量。

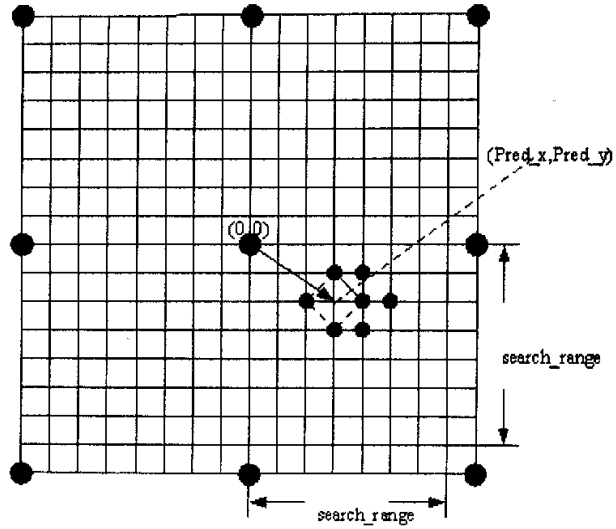


图 4.4 CBFPS 搜索算法的实现

与全搜索相比，CBFPS 算法能减少 33%左右的亚像素运动搜索复杂度，而保持图像质量几乎不变，因而是一种非常有效的快速亚像素搜索算法，并被 H.264 标准采纳。

但是，H.264 中采纳的快速亚像素搜索算法 CBFPS 在应用于一些大的块时预测不太准确，会导致较大的图像质量的损失。因此，在具体实现时只是对小块采用了这种算法，而对于较大的块(8x16,16x8,16x16)仍采用全搜索，以保证图像质量。这样就使得 CBFPS 不能更多地降低亚像素搜索的复杂度，降低了效率。

针对此，我们提出了一种基于预测的方向亚像素搜索 PDFPS(Prediction-based Directional Fractional Pixel Search)，与以往算法相比，所提算法具有以下三方面的优点：

- (1) 采用两种预测 Median Prediction 及 Uplayer Prediction，使得预测矢量更加准确；
- (2) 只计算预测像素及预测方向上的像素而不是所有 8 个像素，这样能极大地减少计算的像素点；
- (3) 采用提前中止算法，进一步减少了不必要进行的细化搜索。

与此同时，PDFPS 不需要周围整像素点的信息，因而可以与任何整像素搜索算法结合使用，是一种效率更高的算法。下节就详细介绍我们所提出的算法。

### 4.3 基于预测的方向亚像素搜索——PDFPS

很多快速运动搜索算法[60][61][62]采用预测矢量来作为其运动搜索的起始位置。一个预测运动矢量包含两部分预测信息：整像素预测矢量和亚像素预测矢量。可以设想，如果整像素预测矢量刚好与经过运动搜索找到的实际运动矢量相同，那么亚像素运动矢量也很可能就是最佳亚像素运动矢量。因此我们就可以从预测位置开始运动搜索。否则，如果预测整像素运动矢量与最佳运动矢量不相等，我们仍可以利用亚像素预测的方向信息，通过只计算预测方向上的像素而不是所有可能的像素位置来减少搜索点的数目。4.3.1 及 4.3.2 节分别介绍了这两种搜索策略，整个算法流程在 4.3.3 节给出。

#### 4.3.1 基于预测的搜索

##### 4.3.1.1 运动矢量预测

在 H.264 中，当前块的运动矢量是通过上方块、左方块、右上方(或左上方)块运动矢量的中值来预测的，如图 4.5 所示。通常，在运动搜索中预测矢量用来初始化搜索起点。

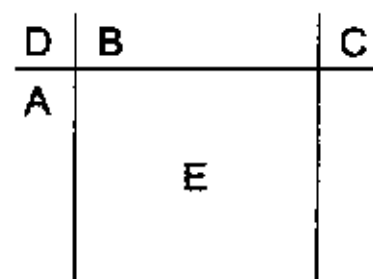


图 4.5 当前块和周围块

此外，H.264 编码标准采用 7 种不同大小的块，从 16x16 到 4x4，图 4.6 给出了一个宏块的两种块划分结构。很容易发现大块与其包含的小块很可能有相似的运动。与[60][61]类似地，为利用不同块大小间的相关性，我们采用大块的运动矢量来预测其中小块的运动矢量，并称之为 Up-layer 预测。比如，16x16 块就是其中包含的 8x16、16x8 块的 Up-layer，8x16 块就是其中包含的两个 8x8 块的 Up-layer，等等。

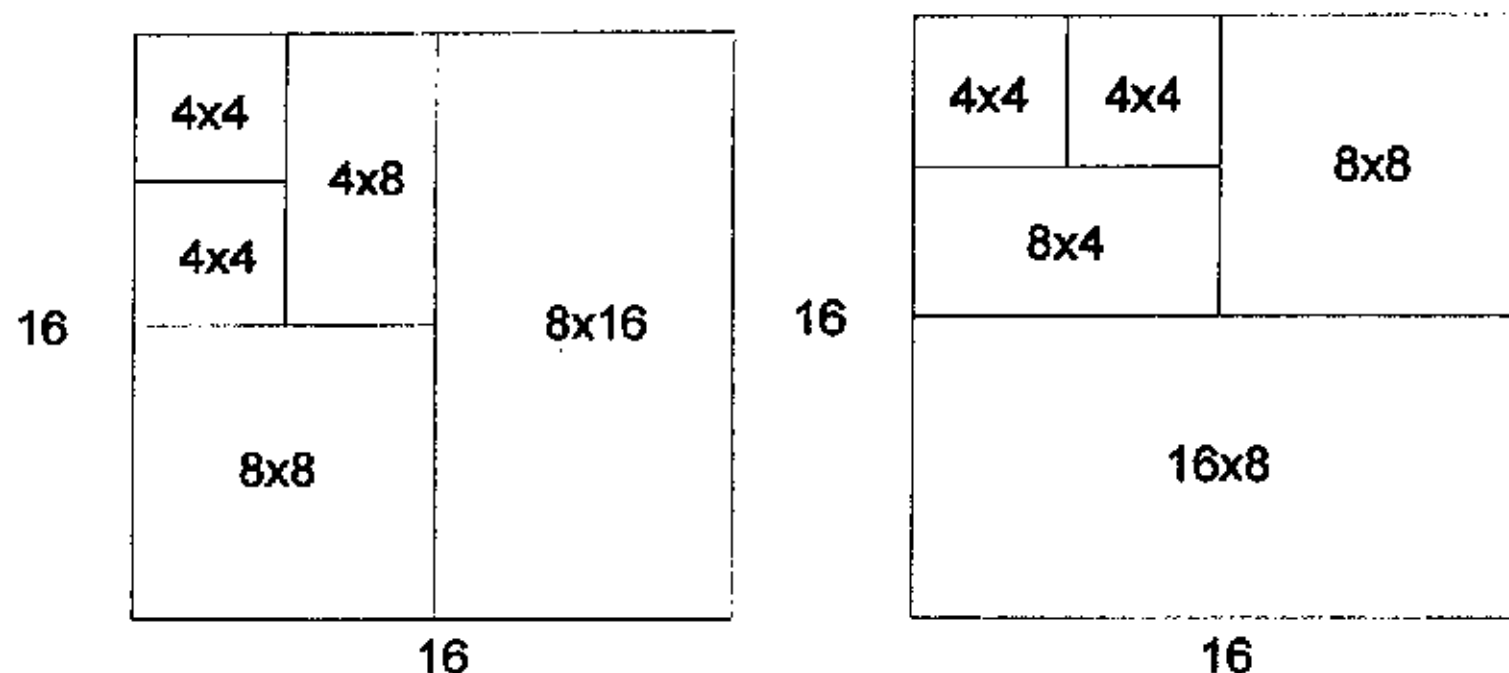


图 4.6 一个宏块中包含的各种不同大小块

可以发现 16x16 块由于是最大的块，因此没有 Up-layer 预测。因为自然视频序列中的物体运动通常是连续的，所以用前一帧对应 16x16 块的运动矢量来预测当前帧对应块，以使每个块都有两个预测矢量，提高预测的精度。

#### 4.3.1.2 亚像素预测矢量精度分析

以 1/4 像素为基准单位，假设预测矢量为  $pred\_mv$ ，最佳整像素运动矢量为  $mv$ ，类似的，可以用(4.1)式来获得亚像素预测矢量  $frac\_pred\_mv$ ，其中  $\beta=4$ 。

为了衡量亚像素预测矢量的精度，我们采用[60]中的方法来计算“预测矢量与全搜索得到的最佳矢量间差值小于或等于  $d$  个像素”的累计分布函数 CDF(Cumulative Distribution Function)，如式(4.2)所定义。

$$P(delta\_MV \leq d) = P\left(\bigcup_{i=1}^7 (|PMV_{i\_x} - MV_{i\_x}| + |PMV_{i\_y} - MV_{i\_y}|) \leq d\right) \quad (4.2)$$

这里  $PMV_{i\_x}$  和  $PMV_{i\_y}$  分别是块  $i$  预测运动矢量的水平和垂直分量， $MV_{i\_x}$  和  $MV_{i\_y}$  分别是是全搜索所得到最佳运动矢量  $MV$  的水平和垂直分量。

当整像素 Median 预测矢量或 Up-layer 预测矢量刚好等于运动搜索找到的最佳矢量时，我们计算 CDF，其结果如表 4.1 所示。这里我们用 Akiyo 及 Foreman 分别代表运动较小和较大的的视频序列。 $delta\_MV$  定义为 1/4 像素单位。

表 4.1 预测矢量与全搜索获得最佳矢量之差的 CDF 值

Sequence	Predicted MV	d = 0	d = 1	d = 2	d = 3
Akiyo	Median	80.65%	95.03%	98.28%	99.41%
	Uplayer	79.44%	94.94%	98.50%	99.51%
	Median + Uplayer	86.33%	97.40%	99.29%	99.77%
Foreman	Median	59.13%	77.99%	86.74%	93.14%
	Uplayer	50.46%	70.77%	83.69%	90.87%

	Median + Up-layer	72.07%	89.08%	95.38%	98.20%
--	-------------------	--------	--------	--------	--------

从表 4.1 可以看到, 对 Foreman 来说, Median 预测和 Up-layer 预测成为最佳矢量的概率分别是 59.13% 及 50.46%。如果把这两个预测结合起来, 则此概率上升为 72.07%, 大大高于单独一个预测矢量的精度。虽然这个精度还不足以选其作为最后的运动矢量, 但我们仍可用 Median 预测和 Up-layer 预测来作为搜索起始位置, 分别进行计算, 取 cost 较小的位置为下一步搜索的起始位置。随后, 如果再进行一次菱形搜索 SDSP(Small Diamond Search Pattern)[63], 就可以得到 89.08% 的最佳运动矢量。进行 3 步 SDSP 后, 准确率能提高到 98.20%, 这样就能基本上找到所有的最佳运动矢量。可以发现, 对于运动较小的 Akiyo 序列来说, 精度则更高。因此, 我们选择预测矢量作为搜索的起始位置, 并加以最多 3 步 SDSP 来获得最佳的运动矢量。

图 4.7 给出了一个典型的 SDSP 搜索过程。它在每一步都计算当前位置的周围四个像素。最小匹配失真所对应的那个位置就作为下一步搜索的中心。重复这样的搜索过程直至最小失真点位于 SDSP 中心或循环到最大的数目。

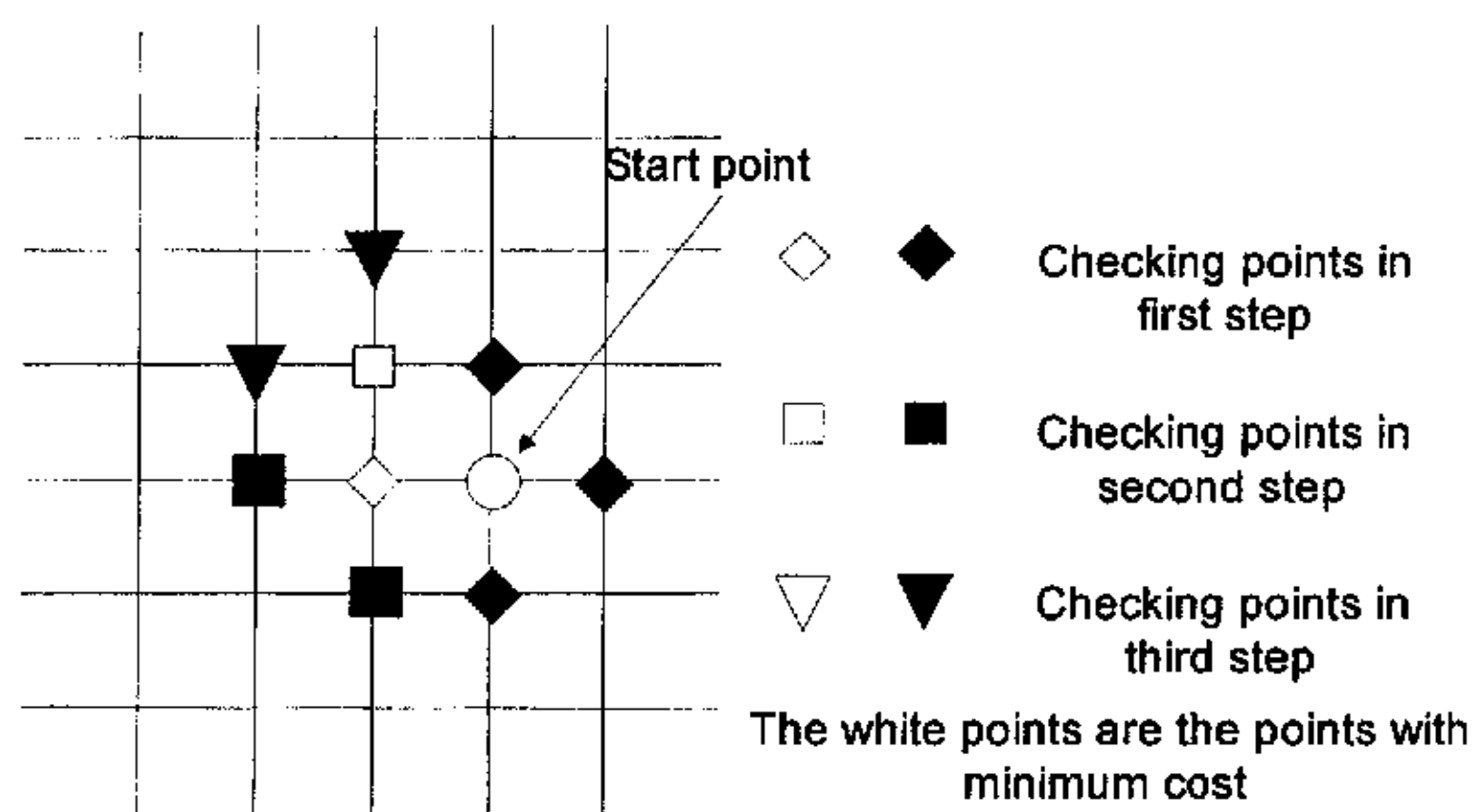


图 4.7 SDSP 搜索的典型过程

### 4.3.1.3 提前中止算法

从表 4.1 可以发现, 对于 Akiyo 和 Foreman 序列来说, 分别有 86.33% 及 72.07% 的块的最佳运动矢量就是其两个预测矢量中的一个。为了叙述方便, 我们定义最佳运动矢量恰好为预测矢量的块为预测矢量块 PVB(Predicted Vector Block)。如果能提前预测出 PVB, 那就能跳过细化搜索 SDSP, 省去部分运动搜索计算。



如果一个块为 PVB, 则必然在预测矢量位置处有较小的 cost 值。因此, 我们对不同的块大小分别定义  $TH_i (i = 1, 2 \dots 7)$ 。在亚像素运动搜索时, 先计算两个预测矢量位置, 如果任何一处的匹配误差满足式(4.3), 那么这个块就可认为是 PVB, 其剩下的亚像素运动搜索就可跳过。

$$Cost_i < TH_i, \quad \text{for } i = 1, \dots, 7 \quad (4.3)$$

### 4.3.2 方向搜索

如果预测的整像素运动矢量不等于最佳矢量, 那预测亚像素运动矢量即为最佳矢量的概率就比较小, 因而无法采用上述的方法直接从预测位置开始搜索。但是, 我们仍可以利用亚像素预测矢量的方向信息。如图 4.8 所示, 我们仅仅计算预测矢量方向上的 1/2 像素值, 而不是所有可能的 8 个 1/2 像素值。预测方向上需计算像素点的选取方法为:

- (1) 如果预测矢量为零, 只计算搜索的起始位置 C;
- (2) 如果预测方向为 1, 计算 1/2 像素位置 c 和 e;
- (3) 如果预测方向为 2 或 3, 则相应地计算 c 或者 e。

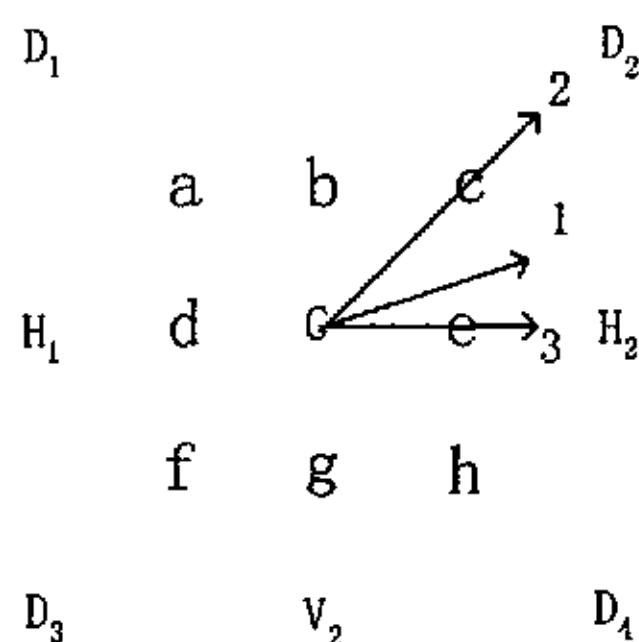
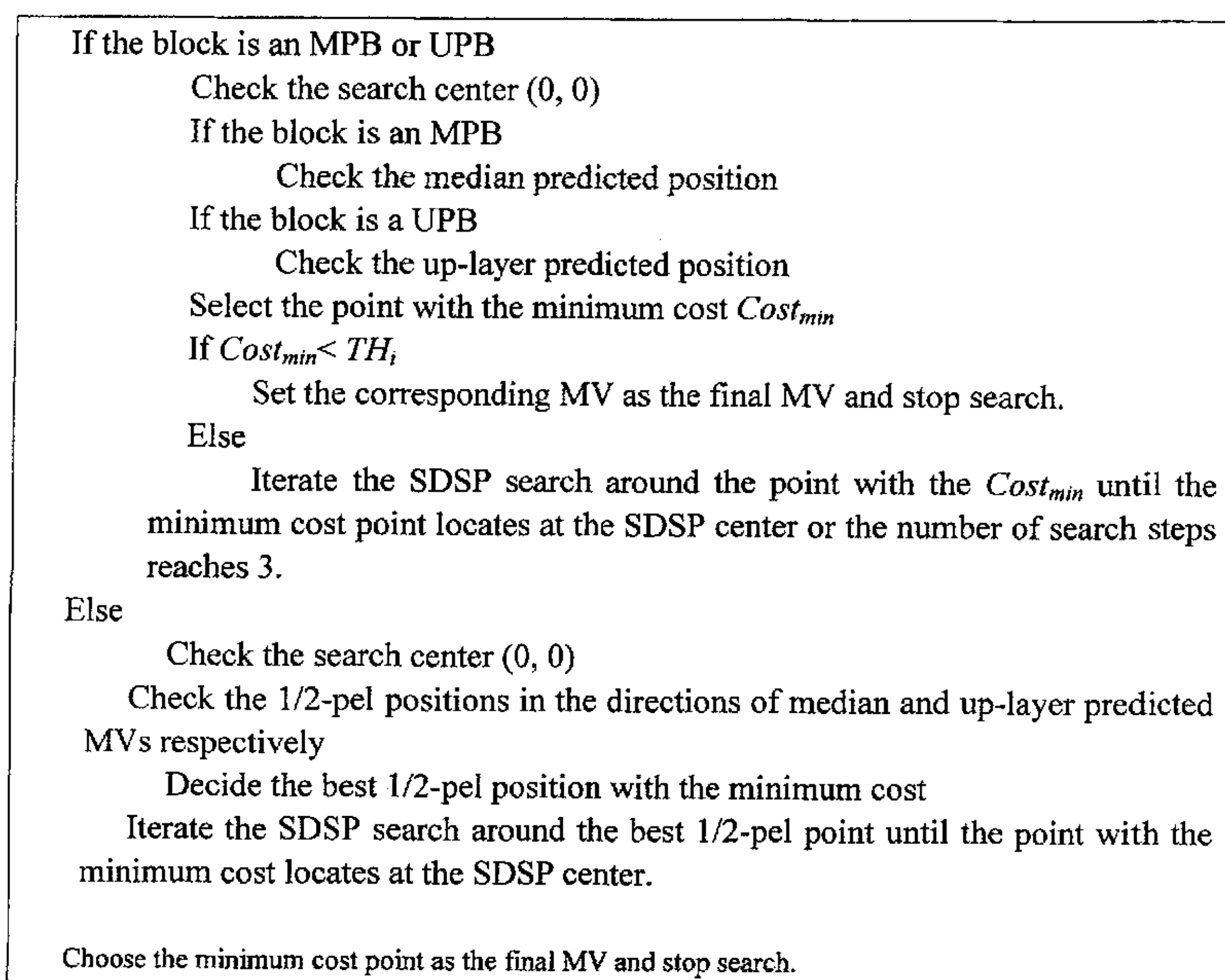


图 4.8 预测方向的 1/2 像素值

预测矢量位于其他方向时也进行类似的处理。上面选出的 1/2 像素点计算完后, 我们选取最小匹配失真所对应的那个像素位置为最佳 1/2 像素运动矢量, 然后在其周围用前一节所介绍的 SDSP 进行细化搜索, 找到最终的 1/4 像素运动矢量。

### 4.3.3 算法流程

基于以上的分析，我们提出了基于预测的方向亚像素搜索算法 PDFPS(Prediction-based Directional Fractional Pixel Search)。为了描述方便，我们定义最佳整像素运动矢量刚好等于 Median 预测矢量的整像素部分的块为 Median 预测块 MPB(Median Predicted Block)，等于 Up-layer 预测的块为 Up-layer 预测块 UPB(Up-layer Predicted Block)。Median 及 Up-layer 预测亚像素由式(4.1)所获得。 $TH_i$  是 4.3.1.3 节中定义的阈值。整个算法流程如下：



### 4.3.4 实验结果

我们将所提 PDFPS 算法应用于 H.264 参考软件 JM8.0[69]，设搜索范围为 16，参考帧为一帧，采用 Hadamard 变换及 CAVLC 编码，未使用 RD-Opt。

为了检验我们的算法在不同实验条件下的效率，我们测试了运动从小到大的

八个不同序列, 分别为 Akiyo, Salesman, News, Silent, Coastguard, Foreman, Container, Football。所有的序列都是 QCIF 格式, 编为 30 帧/秒。第一帧编为 I 帧, 其余帧都为 P 帧, QP(Quantization Parameter)设为 32。

不同视频序列其阈值大小应该不同, 因此七个块大小的阈值  $TH_i$  ( $i = 1, 2, \dots, 7$ ) 需要在编码过程中获得, 以进行自适应调整, 以适用于各种类型视频的运用。由于视频序列具有很强的时间相关性, 因此我们采用前帧对应块的亚像素搜索的 cost 值来作为阈值选择的参考值。大量实验证明, 当阈值  $TH_i$  选为参考值的 1.1 倍时, 能在计算复杂度和图像质量间取得一个很好的折衷。因此我们选其作为算法中的阈值。

为了检验所提出算法的效率, 我们将 PDFPS 与全搜索及 CBFPS 算法在相同条件下进行了对比测试, 结果如表 4.2 所示, 其中的 SP(Search Point)为每个块的模式平均搜索像素数。

表 4.2 不同亚像素搜索算法的比较结果

Sequence	Method	SP	PSNR (dB)	Bit-rate (kbps)
Akiyo	Full Search	17.00	35.29	15.29
	CBFPS	10.48	35.29	15.18
	Proposed Method	2.76	35.28	15.46
Salesman	Full Search	17.00	32.55	31.55
	CBFPS	10.69	32.53	31.50
	Proposed Method	3.36	32.55	31.75
News	Full Search	17.00	33.51	46.52
	CBFPS	10.58	33.54	46.69
	Proposed Method	3.20	33.55	47.23
Silent	Full Search	17.00	32.81	53.40
	CBFPS	10.80	32.79	53.56
	Proposed Method	3.86	32.81	53.91
Coastguard	Full Search	17.00	30.96	105.35
	CBFPS	10.86	30.97	105.15
	Proposed Method	4.54	30.98	105.81
Foreman	Full Search	17.00	33.28	91.31
	CBFPS	11.04	33.24	90.93
	Proposed Method	5.59	33.20	91.84
Container	Full Search	17.00	33.07	22.76
	CBFPS	10.36	33.08	22.60
	Proposed Method	2.80	33.10	22.84
Football	Full Search	17.00	32.15	297.12

	CBFPS	11.45	32.12	296.99
	Proposed Method	6.23	32.12	299.41

从以上的实验结果可看出，我们所提出的算法能取得相当好的结果。与全搜索相比，我们提出的算法能减少 63.38% 到 83.75% 的亚像素搜索点，而 PSNR 的下降平均只有 0.01db。此外，与 H.264 标准采纳的快速亚像素搜索算法 CBFPS 相比，我们的方法也能减少 45.64% 到 73.63% 的搜索点，而平均 PSNR 值保持一致。与此同时，提出的算法对码率的影响很小。

我们也观察到对一些序列，如 News, Coastguard 和 Container 来说，我们提出的算法和算法 CBFPS 的 PSNR 甚至比全搜索还要高一点。这主要是因为全搜索算法只是搜索 8 个  $1/2$  像素点及最佳  $1/2$  像素点周围的 8 个  $1/4$  像素点，而不是搜索区域内的所有  $1/2$  及  $1/4$  像素点，所以不一定总是能找到全局最小点。而我们的算法 PDFPS 及 CBFPS 从预测位置开始搜索，在某些情况下更能找到最佳运动矢量。

#### 4.4 本章小结

本章介绍了各种快速亚像素运动预测算法，着重介绍了 H.264 中采用的两种亚像素搜索算法——全搜索和快速算法 CBPFS，并在此基础上提出了一种更为有效的基于预测的方向搜索快速亚像素搜索算法——PDFPS。该算法通过两种预测方式 Median Prediction 和 Up-layer Prediction 来更准确地预测亚像素运动矢量，并直接从预测位置或预测方向开始搜索，最后采用  $1/4$  像素步长的菱形模式来进行细化搜索。同时采用提前中止算法，当预测矢量的 cost 值小于给定阈值时就跳过后面的运动搜索，以进一步降低运算量。本章所提出的 PDFPS 算法能极大地简化亚像素运动搜索的计算，实验结果表明，与全搜索及算法 CBFPS 相比，本章所提出的算法 PDFPS 分别能降低最多 84% 和 74% 的亚像素搜索点，而图像质量几乎没有下降，是一种非常有效的快速亚像素搜索算法。

## 第五章 快速帧内预测

前两章我们提出了有效的快速帧间运动预测算法，包括整像素运动预测与亚像素运动预测，能大大降低运动搜索的复杂度，使其下降到全搜索复杂度的 5% 以下，已不成为 H.264 编码器的计算瓶颈，必须对其他模块进行优化。

在 H.264 编码标准采用的新技术中，帧内预测是取得高编码效率的关键技术之一。H.264 采用了基于空间域进行预测的新的帧内预测算法，与 H.263+ 高级帧内编码模式中 DCT 域的帧内预测相比，其 PSNR 值平均提高了 4.37dB[64]。但是，由于采用了两种不同的分块大小 4x4, 16x16 及 13 种预测模式，大大增加了编码器的复杂度。因此，必须对帧内预测算法进行优化以降低其计算复杂度。本章就分别介绍 H.264 采用的帧内预测全搜索算法、以前所提出的快速帧内预测算法，并在此基础上提出了一种有效的基于模式预测的快速帧内预测算法。

### 5.1 H.264 中现有的帧内预测算法

H.264 中帧内预测算法支持两种块大小：4x4 及 16x16，根据已经编码及重建的块（当前块上方，左方及左上方的块）来预测当前块，充分利用子块间的相关信息来降低码率。

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

图 5.1 4x4 块及其周围像素

如图 5.1 所示，一个 4x4 块包括 16 个像素 a~p，周围块的像素 A~M 已经解码生成。4x4 预测模式表示如何用全部或部分周围块的像素 A~M 来预测生成当前块的 16 个像素值 a~p。在 4x4 帧内预测中共有 9 种不同方向的预测模式，如图 5.2 所示。其中，模式 2 为 DC 预测，所有像素值（a~p）的预测值为  $(A+B+C+D+I+J+K+L)/8$ 。

## 第五章 快速帧内预测

前两章我们提出了有效的快速帧间运动预测算法，包括整像素运动预测与亚像素运动预测，能大大降低运动搜索的复杂度，使其下降到全搜索复杂度的 5% 以下，已不成为 H.264 编码器的计算瓶颈，必须对其他模块进行优化。

在 H.264 编码标准采用的新技术中，帧内预测是取得高编码效率的关键技术之一。H.264 采用了基于空间域进行预测的新的帧内预测算法，与 H.263+高级帧内编码模式中 DCT 域的帧内预测相比，其 PSNR 值平均提高了 4.37dB[64]。但是，由于采用了两种不同的分块大小 4x4、16x16 及 13 种预测模式，大大增加了编码器的复杂度。因此，必须对帧内预测算法进行优化以降低其计算复杂度。本章就分别介绍 H.264 采用的帧内预测全搜索算法、以前所提出的快速帧内预测算法，并在此基础上提出了一种有效的基于模式预测的快速帧内预测算法。

### 5.1 H.264 中现有的帧内预测算法

H.264 中帧内预测算法支持两种块大小：4x4 及 16x16，根据已经编码及重建的块（当前块上方，左方及左上方的块）来预测当前块，充分利用子块间的相关信息来降低码率。

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

图 5.1 4x4 块及其周围像素

如图 5.1 所示，一个 4x4 块包括 16 个像素 a~p，周围块的像素 A~M 已经解码生成。4x4 预测模式表示如何用全部或部分周围块的像素 A~M 来预测生成当前块的 16 个像素值 a~p。在 4x4 帧内预测中共有 9 种不同方向的预测模式，如图 5.2 所示。其中，模式 2 为 DC 预测，所有像素值（a~p）的预测值为  $(A+B+C+D+I+J+K+L)/8$ 。



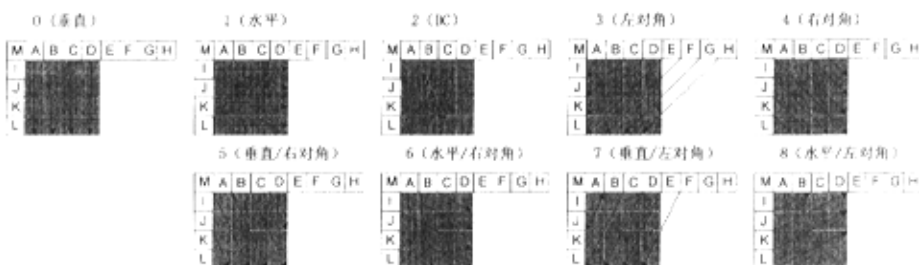


图 5.2 9 种 4x4 块的帧内预测模式

类似地，在 16x16 预测中，周围宏块的 33 个像素被用来预测当前块，其中 32 个像素来自上方和左方的宏块，一个像素来自左上方的宏块。16x16 有 4 种预测模式，如图 5.3 所示。

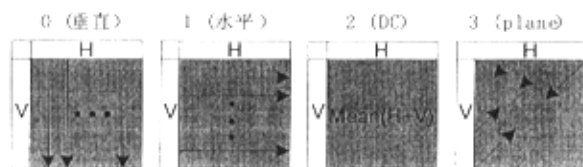


图 5.3 4 种 16x16 块的帧内预测模式

由于 4x4 预测有 9 种候选模式，需要 4 比特来编码，而且一个宏块包含 16 个 4x4 块，因此对每个宏块的 4x4 帧内模式进行编码需要大量的比特数。为了更有效地编码，根据相邻块的相关性，以上方和左方的 4x4 块的模式来预测当前块的模式，称之为最大概率模式(most probable mode)。先发送一比特表明当前块的模式是否为最大概率模式，如不是，才发送 3 比特来表征采用剩下的 8 种模式中的哪一个。这样，在 4x4 帧内预测时就能省下大量的比特开销。

H.264 的参考软件采用全搜索来进行帧内预测的模式选择，其主要步骤如下 [65] (1~5: 4x4 预测; 6~10: 16x16 预测):

1. 根据一个候选 4x4 模式来生成 4x4 的预测块;
2. 计算预测块和原始 4x4 块之间经 Hadamard 变换后的绝对差和(Sum of Absolute Transform Differences, SATD16);
3. 计算

$$Cost_{4 \times 4} = SATD_{16} + 4P\lambda_{MODE} \quad (5.1)$$

其中  $\lambda_{MODE}$  是由量化系数  $Qp$  决定的一个常量，可由查找表获得。当采用最大概率模式时  $P$  为 0，采用另外模式时取 1;

4. 对所有的 4x4 预测的 9 个模式重复步骤 1~3, 选取对应最小 Cost4x4 的那个模式为最佳 4x4 模式;
5. 对当前宏块中所有 16 个 4x4 块重复步骤 1~4, 把每个 Cost4x4 相加;
6. 根据一个 16x16 模式生成 16x16 预测宏块;
7. 将预测宏块与原始宏块相减, 获得残差宏块, 并对每个 4x4 块进行 Hadamard 变换;
8. 在所有 16 个经 Hadamard 变换后 4x4 块中提取 DC 系数组成另一个 4x4 块, 并对这个块进行 Hadamard 变换;
9. 将所有 Hadamard 变换系数(除了 7 中的 DC 系数)的绝对值相加, 取其值的一半作为该 16x16 模式的 cost 值;
10. 对所有的 4 个 16x16 模式重复 6~9, 选取最小的 cost 作为 Cost16x16;
11. 用(5.2)比较分别从步骤 5 和 10 中获得的 cost 值:

$$Cost_{16 \times 16} \geq \sum_{i=1}^{16} Cost_{4 \times 4, i} + 24 \lambda_{MODE} \quad (5.2)$$

如果(5.2)成立, 则当前宏块采用 4x4 预测, 否则, 采用 16x16 预测。

帧内预测的模式选定后, 当前宏块与其预测值的残差被变换、量化及编码。

虽然全搜索能获得最佳预测模式, 但其运算量很大, 不利于实时应用, 所以必须采用更为有效的优化算法来降低运算复杂度。

## 5.2 快速帧内预测算法综述

与大量的帧间预测快速算法相比, 快速帧内预测这方面的研究工作还不是很多。F.Pang 等人在[66]提出了基于边缘方向检测预处理的模式选择算法, 用图像的 edge map 来获得边缘方向的直方图, 因而可以选出较少的候选模式来进行 RDO(Rate-distortion Optimization)计算。C.S. Kim 等人提出一种利用绝对差和及方向信息的多级模式选择的快速帧内预测算法[67], 该算法计算 SATD(sum of absolute residual coefficients)、SAG(sum of absolute gradients)及可选择的 RDO 模式来决定最佳 4x4 块模式。具体就是首先计算所有 9 个模式的 SATD 值, 并滤除最不可能的模式, 然后计算剩余模式的方向信息并根据 posteriori 值来选择最佳模式。同时, 该算法还在不同的中间判断步骤中采用提前中止算法, 当有较好的

4. 对所有的 4x4 预测的 9 个模式重复步骤 1~3, 选取对应最小 Cost4x4 的那个模式为最佳 4x4 模式;
5. 对当前宏块中所有 16 个 4x4 块重复步骤 1~4, 把每个 Cost4x4 相加;
6. 根据一个 16x16 模式生成 16x16 预测宏块;
7. 将预测宏块与原始宏块相减, 获得残差宏块, 并对每个 4x4 块进行 Hadamard 变换;
8. 在所有 16 个经 Hadamard 变换后 4x4 块中提取 DC 系数组成另一个 4x4 块, 并对这个块进行 Hadamard 变换;
9. 将所有 Hadamard 变换系数(除了 7 中的 DC 系数)的绝对值相加, 取其值的一半作为该 16x16 模式的 cost 值;
10. 对所有的 4 个 16x16 模式重复 6~9, 选取最小的 cost 作为 Cost16x16;
11. 用(5.2)比较分别从步骤 5 和 10 中获得的 cost 值:

$$Cost_{16 \times 16} \geq \sum_{i=1}^{16} Cost_{4 \times 4, i} + 24 \lambda_{MODE} \quad (5.2)$$

如果(5.2)成立, 则当前宏块采用 4x4 预测, 否则, 采用 16x16 预测。

帧内预测的模式选定后, 当前宏块与其预测值的残差被变换、量化及编码。

虽然全搜索能获得最佳预测模式, 但其运算量很大, 不利于实时应用, 所以必须采用更为有效的优化算法来降低运算复杂度。

## 5.2 快速帧内预测算法综述

与大量的帧间预测快速算法相比, 快速帧内预测这方面的研究工作还不是很多。F.Pang 等人在[66]提出了基于边缘方向检测预处理的模式选择算法, 用图像的 edge map 来获得边缘方向的直方图, 因而可以选出较少的候选模式来进行 RDO(Rate-distortion Optimization)计算。C.S. Kim 等人提出一种利用绝对差和及方向信息的多级模式选择的快速帧内预测算法[67], 该算法计算 SATD(sum of absolute residual coefficients)、SAG(sum of absolute gradients)及可选择的 RDO 模式来决定最佳 4x4 块模式。具体就是首先计算所有 9 个模式的 SATD 值, 并滤除最不可能的模式, 然后计算剩余模式的方向信息并根据 posteriori 值来选择最佳模式。同时, 该算法还在不同的中间判断步骤中采用提前中止算法, 当有较好的

模式判断时就结束搜索,以避免不必要的计算。实验证明该算法能把帧内预测的复杂度降低 10~30 倍。但是以上两种算法都是基于 RDO 来进行模式选择, RDO 的运算本身就带来了巨大的运算复杂度,在针对实时运用时一般都不采用。

此外,还有 B.J. Meng 等人提出的基于预测块的部分计算和模式选择的快速帧内预测算法[65][68],该算法首先预测出一些可能为最佳模式的候选模式,然后通过计算 4x4 块中的部分像素而不是全部像素来选出最佳模式,以降低运算量,但其算法复杂度仍较高。

在下一节中,我们提出了一种基于模式预测的快速帧内预测算法[70]。该算法针对全 I 帧编码,充分利用了帧间对应块、帧内相邻块和不同块大小之间的相关性来进行模式预测,并结合多种提前中止算法来尽早选出最佳模式,减少不必要的模式选择计算。同时,由于 RDO 运算的复杂度不利于实时运用,算法不采用 RDO 来进行模式选择,以下的讨论均基于此。实验证明,该种算法能大大降低运算开销,同时保持图像质量。

## 5.3 基于模式预测的快速帧内预测算法

### 5.3.1 算法的提出

在帧内预测中,一个宏块如采用 4x4 预测,其 16 个 4x4 块的 9 种模式需花费大量比特进行编码,而如采用 16x16 预测,只需对一个 16x16 块的 4 种模式进行编码,比特数大大减少。同时,只有在图像的细节部分进行帧内预测时,需使用 4x4 预测来提高预测的精度,而对图像中大量的平坦区域可使用 16x16 预测。因此,在帧内预测时可优先考虑 16x16 预测,在保证预测精度的同时降低比特数及运算复杂度。

由于帧内预测有多达 13 种候选模式,对每种模式进行计算并选出最佳模式的模式选择过程是编码复杂度的瓶颈所在,所以帧内预测的优化关键在于减少所需计算的模式数,这可以通过已编码的块对当前块最佳模式的预测来实现。由于物体运动的连续性,视频序列在时域和空间域上均存在着很强的相关性,因此可以利用前一帧对应块或当前帧周围块的模式来预测当前块的最佳模式。与此同时,不同大小的块之间也存在着一定的相关性,可以用 16x16 块的模式来预测其

中  $4 \times 4$  块的模式。统计表明[65], 当  $16 \times 16$  块的最佳模式为 0, 1 或 2 时, 其中包含的  $4 \times 4$  块也大都为对应的模式;  $16 \times 16$  块为模式 3 (Plane Mode) 时, 其中  $4 \times 4$  块的模式可能为 3, 7, 8。

全搜索算法是对一个宏块先采用  $4 \times 4$  预测, 再采用  $16 \times 16$  预测, 但基于以上的讨论我们可以变换一下顺序, 即先进行  $16 \times 16$  预测, 然后再进行  $4 \times 4$  预测, 这是因为:

(1) 优先考虑了  $16 \times 16$  预测, 如其满足精度, 则省去  $4 \times 4$  预测, 以降低比特数及运算量;

(2) 用获得的  $16 \times 16$  块的模式来预测其中的  $4 \times 4$  块模式, 提高预测的准确性;

(3) 如多个  $4 \times 4$  块的 cost 之和大于  $\text{Cost}_{16 \times 16}$ , 则最佳预测模式必为  $16 \times 16$ , 这样可以跳过当前宏块剩下的  $4 \times 4$  块的帧内预测, 进一步降低计算量。

当通过相关性得到预测模式后, 就可采用提前中止的方法, 当预测模式的 cost 小于给定的阈值时, 可认为该种模式就是最佳模式, 进而跳过其余模式的计算。

基于上述的分析, 我们提出了基于模式预测的快速帧内预测算法, 主要步骤如下 (为了叙述方便, 记采用  $16 \times 16$  预测的块为  $116 \times 16$ , 采用  $4 \times 4$  预测的块为  $14 \times 4$ ):

1. 对当前宏块, 如前一帧的对应位置宏块为  $116 \times 16$ , 取其模式作为当前宏块的预测模式  $\text{Cand}_{16 \times 161}$ ; 如当前帧在其上方和左方的宏块都为  $116 \times 16$ , 取其中小的模式作为当前宏块的预测模式  $\text{Cand}_{16 \times 162}$ ; 如以上均不满足, 则取预测模式为 DC 预测;
2. 计算预测模式的 cost 值 (如有两个预测模式则取其 cost 较小值), 并与阈值  $\text{TH}_1$  比较。如小于  $\text{TH}_1$ , 跳至 11; 否则, 跳至 3;
3. 计算除预测模式外的其余  $16 \times 16$  模式的 cost 值, 并选取 4 个模式的最小 cost 作为  $\text{Cost}_{16 \times 16}$ , 其对应的模式为  $16 \times 16$  预测的最佳模式  $\text{Model}_{16 \times 16}$ ;
4. 如  $\text{Cost}_{16 \times 16}$  小于阈值  $\text{TH}_2$ , 跳至 11; 否则, 跳至 5;
5. 如果  $\text{Model}_{16 \times 16}$  为 0, 1 或 2, 选取其为  $16 \times 16$  块所包含的  $4 \times 4$  块的预测模式  $\text{Cand}_{4 \times 41}$ , 否则, 则取模式 3, 7, 8 为  $4 \times 4$  块的预测模式; 选取最大概率模式为  $4 \times 4$  块的另一个预测模式  $\text{Cand}_{4 \times 42}$ ;

6. 计算  $4 \times 4$  块的两组预测模式，并选取最小的 cost 为  $\text{Cost}_{4 \times 4 \min}$ ;
7. 如果  $\text{Cost}_{4 \times 4 \min}$  小于阈值  $\text{TH}_3$ ，取其作为  $\text{Cost}_{4 \times 4}$ ，跳至 9；否则跳至 8；
8. 计算除预测模式外的其余  $4 \times 4$  模式的 cost 值，并选取 9 个模式的最小值为  $\text{Cost}_{4 \times 4}$ ;
9. 对当前宏块中所有 16 个  $4 \times 4$  块重复步骤 5~8，把每个  $\text{Cost}_{4 \times 4}$  相加；如其和大于  $\text{Cost}_{16 \times 16}$ ，跳过所有剩下的  $4 \times 4$  块，并选择当前宏块的帧内预测为  $16 \times 16$  预测，跳至 11；
10. 根据(5.2)式选择当前宏块采用  $4 \times 4$  预测或是  $16 \times 16$  预测；
11. 选取对应的模式为当前宏块的最佳帧内预测模式，结束预测过程。

在以上的快速算法中，阈值的选取是个关键的因素，它直接影响到编码速度和图像质量：当阈值选得较大时，有大量块的预测模式被判为其最佳模式，因而跳过其余模式的计算，减少了运算量，但发生误判的概率较大，带来较大的图像质量的下降；反之，较小的阈值保证了图像质量，但运算量减少得较少。如何选择阈值以使在保证图像质量的前提下尽可能多地降低计算复杂度是阈值选择的关键。同时，不同视频序列的运动不同，导致一组固定的阈值无法适用于各种类型的视频。所以必须要求阈值能根据不同视频内容自适应调整。

由于视频序列的连续性，在帧与帧之间存在着很强的相关性，因此可以利用前一帧帧内预测的信息来决定提前中止的阈值。假设前帧对应宏块所选取的最佳帧内模式其对应的最小 cost 值为  $\text{Cost}_{\min}$ ，如果当前块的某一预测模式的 cost 值小于  $\text{Cost}_{\min}$ ，则其有很大的概率就为最佳模式，因此就可以跳过其他模式的计算。大量实验表明，当  $\text{TH}_1$  和  $\text{TH}_2$  都为前帧对应宏块的  $\text{Cost}_{\min} \times 1.1$  时，能在复杂度和图像质量间取得一个很好的折衷。 $\text{TH}_3$  是  $4 \times 4$  块的阈值，相应地，可以取为  $1.1 \times \text{Cost}_{\min} / 16$ 。同时，这样的阈值选取也满足根据不同内容自适应调整的要求。

帧内预测的计算复杂度可以近似地用所需要计算的像素点来表示。 $16 \times 16$  预测的一个模式需要计算 256 个像素点， $4 \times 4$  预测的一个模式需要计算 16 个像素点，由于一个宏块包含 16 个  $4 \times 4$  块，因此宏块的一个  $4 \times 4$  模式也需要计算 256 个像素点。为了方便说明，本文中的复杂度计算以一个宏块的 256 个像素为单位，



记为 CPMB(checking points per macroblock), 即一个 16x16 模式的计算量为 1CPMB, 4x4 块的一个 4x4 模式计算为 1/16CPMB, 宏块的一个 4x4 模式的计算为 1CPMB。对于全搜索来说, 一个宏块需要计算 4 个 16x16 模式和 9 个 4x4 模式, 因此计算量为  $4+9=13\text{CPMB}$ 。而在我们提出的快速算法中, 在最理想的情况下, 只需计算步骤 2 中的 1~2 个 16x16 预测模式, 然后跳过剩下的 16x16 预测与 4x4 预测, 因此计算量只为 1~2CPMB, 能减少 84%~92% 的帧内预测复杂度, 在图像的平坦区域如背景部分有极大的概率出现这样的情况; 在次佳的情况下, 如在步骤 4 中 16x16 预测的 cost 值满足阈值, 则将跳过整个 4x4 预测, 计算量为 4CPMB, 能降低 69% 的计算量; 如 16x16 预测不满足提前中止条件, 在随后进行的 4x4 预测中的两类提前中止判断也能降低相当的运算量。

### 5.3.2 实验结果与分析

我们将所提出的快速帧内预测算法应用于 H.264 的参考软件 JM8.0[69], 并采用 Hadamard 变换及 CAVLC 编码, 不使用 RD 优化及码率控制。

为了检验算法在不同实验条件下的效率, 我们采用了五个不同运动的视频序列, 分别为: Akiyo, News, Coastguard, Container, Football。所有序列均为 QCIF 格式, 帧率 30 帧/秒。每个序列都编 250 帧, 所有帧都编为 I 帧。同时, 为了检验算法在不同码率下的效率, 实验中采用了 3 个不同的 QP 值 24, 28, 32。阈值的选取如上一节所述。

我们将所提出的快速算法与全搜索帧内预测在相同的测试条件下进行了比较。表 5.1~表 5.3 列出了实验结果。其中, CPMB 在上一节中作了定义, 表示基于宏块的计算像素点数的单位, 用以表征计算复杂度, 全搜索 4x4 预测的 9 个模式和 16x16 预测的 4 个模式的计算量为 13CPMB。

表 5.1 QP=24 的实验结果

序列	算法	CPMB	PSNR (dB)	Bit-rate (kbps)
Akiyo	FS	13.00	41.70	796.69
	Proposed	6.48	41.65	817.75
News	FS	13.00	40.57	1111.90
	Proposed	6.21	40.52	1152.26
Coastguard	FS	13.00	38.48	1394.37

	Proposed	4.83	38.45	1449.39
Container	FS	13.00	39.86	1088.37
	Proposed	4.83	39.84	1117.09
Football	FS	13.00	39.17	1332.44
	Proposed	5.87	39.13	1373.99

表 5.2 QP=28 的实验结果

序列	算法	CPMB	PSNR (dB)	Bit-rate(kbps)
Akiyo	FS	13.00	38.87	575.11
	Proposed	5.73	38.87	595.63
News	FS	13.00	37.59	818.17
	Proposed	5.94	37.55	848.47
Coastguard	FS	13.00	35.35	976.29
	Proposed	3.90	35.31	1025.44
Container	FS	13.00	37.02	767.92
	Proposed	4.65	36.99	790.83
Football	FS	13.00	36.06	916.88
	Proposed	5.60	36.01	949.87

表 5.3 QP=32 的实验结果

序列	算法	CPMB	PSNR (dB)	Bit-rate(kbps)
Akiyo	FS	13.00	35.90	411.29
	Proposed	4.87	35.88	429.07
News	FS	13.00	34.38	590.60
	Proposed	5.32	34.37	618.07
Coastguard	FS	13.00	32.39	635.12
	Proposed	3.25	32.35	679.17
Container	FS	13.00	34.14	528.18
	Proposed	4.24	34.12	551.62
Football	FS	13.00	33.25	596.17
	Proposed	5.20	33.20	623.68

仿真实验结果表明，我们所提出的快速帧内算法取得了很好的效果。与全搜索相比，该快速算法能减少 50%~75%的帧内预测计算量，而 PSNR 平均只下降了 0.03dB，码率略有上升。

## 5.4 本章小结

本章首先介绍了 H.264 中所采纳的帧内预测算法，并对以往所提出的快速算法进行了综述。在此基础上，提出了一种基于模式预测的快速帧内预测算法，该算法利用视频序列时域、空域及不同块大小之间的相关性，用前帧对应块、当前块周围块及  $16 \times 16$  块的模式来预测当前块的模式，并结合提前中止方法，极大地减少了帧内预测所需计算的模式数量，降低了运算复杂度。实验结果表明：与全搜索相比，该快速算法能降低  $1/2 \sim 3/4$  的帧内预测计算量，同时保持编码视频的图像质量。

## 第六章 总结与展望

本文首先介绍了视频压缩编码的理论及编码技术,并对视频编码的国际标准进行了综述,着重介绍了最新的视频压缩国际标准 H.264。与以往的标准相比,H.264 有更高的压缩效率,但编码复杂度也最高,不利于实时应用,必须对算法进行优化。

在分析了 H.264 编码器各模块复杂度的基础上,我们确定优化的重点为预测部分,包括帧间预测(整像素运动预测和亚像素运动预测)及帧内预测。

通过调研,找出了以往文献中所提出了针对这三部分算法优化的快速算法,并在此基础上进行了改进,分别提出了搜索效率更高的快速算法:

- 1) 整像素运动搜索:首先采用了 H.264 标准所采纳的混合的非对称多六边形搜索算法 UMHexagonS,然后在应用于 H.263 的零矢量检测算法的基础上提出了一种提前中止算法 VBBMD(Variable Block-size Best Motion Detection)。该算法通过计算一个搜索点来预测最佳运动矢量,因而能提前中止一部分运动搜索过程,跳过与之相对应的计算。与单独采用 UMHexagonS 相比,所提算法能进一步减少 76%~94% 的整像素搜索点,而平均 PSNR 下降只有 0.06dB。所提算法与 UMHexagonS 结合使用,能减少 96% 以上整像素运动搜索计算量。与此同时,该算法能与任何快速整像素搜索算法结合使用,大大降低整像素运动搜索复杂度,且不增加额外的计算开销。
- 2) 亚像素运动搜索:在 H.264 标准所采纳的快速亚像素搜索算法 CBFPS 的基础上,提出了基于预测的方向亚像素搜索 PDFPS(Prediction-based Directional Fractional Pixel Search)。该算法采用两种预测方式来更准确地预测亚像素运动矢量,并直接从预测位置或预测方向开始搜索,最后采用 1/4 像素步长的菱形模式来进行细化搜索。同时采用提前中止算法,当预测矢量的 cost 值小于给定阈值时就跳过之后的运动搜索。本文提出的 PDFPS 算法能极大地简化亚像素运动搜索的计算,与全搜索及 CBFPS 相比,本文所提出的算法能分别降低最多 84% 和 74% 的亚像素搜索点,而图像质量保持基本不变,是一种非常有效的快速亚像素搜索算法。

- 3) 帧内预测：在分析了 H.264 采用的帧内预测及现有的快速算法的基础上，本文提出了一种基于模式预测的快速帧内预测算法。该算法利用视频序列时域、空域及不同块大小之间的相关性，用前帧对应块、当前块周围块及  $16 \times 16$  块的模式来预测当前块的模式，并结合提前中止方法，极大地减少了帧内预测所需计算的模式数量，降低了运算复杂度。实验结果表明：与全搜索相比，该快速算法能降低  $1/2 \sim 3/4$  的帧内预测计算量，同时保持编码视频的图像质量。

在进行了上述算法的优化后，能将 H.264/ JVT 参考软件 JM8.0 的编码器的速度提高 3~4 倍，是对 H.264 的实时应用进行的有益探索。

用 VTune 软件进行分析可发现，现在各模块复杂度所占的比例如图 6.1 所示。

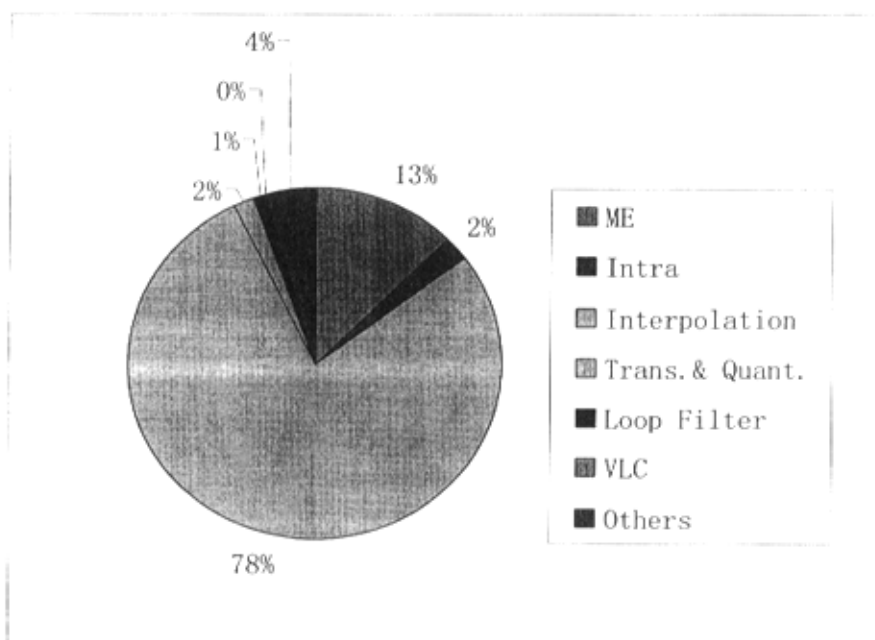


图 6.1 算法优化后 H.264 编码器各模块复杂度

上图与图 2.9 相对比可看出，运动预测(ME)及帧内预测(INTRA)所占比例大为降低，特别是 ME，比例从 71%下降为 13%，复杂度大为降低，已不再是编码器最高计算复杂度的模块。与此同时，亚像素差值(Interpolation)比例从 19%上升为 78%，取代 ME 成为编码器的计算瓶颈。

总结上述工作，可以发现以后的工作主要集中为：

- (1) 继续对差值模块进行优化。由于这一部分的算法很难再进一步简化, 因此可以考虑在代码实现层面进行优化, 包括用汇编改写等等。同时, 也可借鉴 AVS 的做法,  $1/2$  像素和  $1/4$  像素都用 4 抽头的滤波器产生, 在不降低性能的情况下减少插值所需要的参考像素点, 减小数据存取带宽需求, 降低复杂度。
- (2) 基于 DSP 的优化和实现。TI C64x DM642 是 TI 公司推出的新型高性能媒体处理器, 其高达 600MHz 的频率、每个指令周期可并行运行 8 条 32 位指令、二级 Cache、EMDA 通道结构及丰富的媒体指令集使其特别适合于视频处理, 因此可基于其进行各种优化, 包括存储器系统的优化及程序代码级的优化等, 进行 H.264 codec 的实时应用。



## 参考文献

- [1] 范晓鹏. 面向高清晰度视频的压缩技术研究, 哈尔滨工业大学硕士学位论文 2003
- [2] K.Aizawa, H. Harashima, T. Saito. Model-based Analysis Synthesis Image Coding (MBASIC) System for a Person's Face. *Signal Processing*, 1989, 1(2): 139-152
- [3] Essa, Pentland. Coding, Analysis, Interpretation, and Recognition of Facial Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 757-763
- [4] 吴枫. 基于对象的运动图像编码技术的研究. 哈尔滨工业大学博士学位论文, 1999: 5-33
- [5] Ji-Zheng Xu, Shipeng Li and Ya-Qin Zhang. Three-dimensional Shape adaptive Discrete Wavelet Transforms for Efficient Object-based Video coding, *IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000*, Perth, June 2000: 352-355
- [6] T. Sikora and B. Makai. Shape-adaptive DCT for Generic Coding of Video. *IEEE Trans. on Circuits and Systems for Video Technology*, 1995, 5: 5962
- [7] T. Sikora. Low Complexity Shape-Adaptive DCT for Coding of Arbitrarily Shaped Image Segments. *Signal Processing: Image Communication*, 1995, 7: 381-395
- [8] K. Shen, E.J. Delp. Wavelet Base Rate Scalable Video Compression. *IEEE trans. on CSVT*, 1999 9(1): 109-121
- [9] D. Wit, T. Hou, Y.-Q. Zhang. Scalable Video Coding and Transport over Broadband Wireless Networks. *Proceedings of the IEEE, Special Issue on Multi-Dimensional Broadband Wireless Technologies and Applications*, Jan, 2001, 89(1): 6-20
- [10] W. Li. Overview of Fine Granularity Scalability in MPEG-4 Video Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2001, 11(3): 301-317
- [11] Yuwen He, Feng Wu, Shipeng Li, Yuzhuo Zhong, Shiqiang Yang. H.26L based Fine Granularity Scalable Video Coding. *ISCAS 2002*, 2002, 4: 17548-551
- [12] MPEG-4 ISO/IEC Video Group. MPEG-4 Video Verification Model Version 16.0. JTC1/SC29/WG11. Noordwijkerhout, Netherlands, Mar. 2000: N3312
- [13] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete Cosine Transforms. *IEEE Trans. Comp.*, 1974, C-23: 90-93.
- [14] Shapiro. Embedded Image Coding Using Zerotree of Wavelet Coefficients. *IEEE Trans. Signal Processing*, 1993, 41(12): 3445-3463
- [15] A. Said and W. Pearlman. A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. *IEEE Transaction on CSVT*, 1996, 6(3): 243-250
- [16] M. Boliek, C. Christopoulos, E. Majani (editors). JPEG2000 Part I Final Draft International Standard. ISO/IEC JEC1/SC29/WG1, August 18, 2000: N1855
- [17] ISO/IEC JTC1/SC29/WG1. JPEG2000 Verification Model 7.0. ISO/IEC JTC1/SC29/WG1, April 2000: N1684(software)
- [18] P.J. Burt, E.H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. on Communications*, April 1983: 532540
- [19] Katto, J. Ohki, S. Nogaki, and M. Ohta. A Wavelet Codec with Overlapped Motion

- Compensation for Very Low Bit-rate Environment. IEEE Trans. Circuits Syst. Video Technol., June 1994, 4: 328-338
- [20] T. Wiegand and G. Sullivan. The Emerging H.264/H.26L Video Coding Standard. Tutorial at ICIP 2002, Rochester, NY, Sept. 2002
- [21] ITU-T/SG 16NCEG. H.26L Test Model Long Term Number 8 (TML-8). June 2001 (software)
- [22] T. Hallbach. Performance Comparison: H.26L Intra Coding vs. JPEG2000. Joint Video Team (JVT) of ISO/IEC MPEG&ITU-T 4th Meeting, July, 2002: D039
- [23] S. Mallat and Z. Zhang. Matching Pursuits with Time-Frequency Dictionaries. IEEE Transactions in Signal Processing, December 1993, 41 (12): 3397~3415
- [24] R. Neff and A. Zakhor. Very Low Bit Rate Video Coding Based on Matching Pursuits. IEEE Trans. Circuits and Systems for Video Technology, February, 1997, 7(1): 158~171
- [25] F. Dufaux, J. Konrad. Efficient, Robust, and Fast Global Motion Estimation for Video Coding. IEEE Trans. Image Processing, Mar. 2000, 9: 497-501
- [26] M. Flierl, T. Wiegand, and B. Girod. Multihypothesis Pictures for H.26L. IEEE International Conference on Image Processing, Thessaloniki, Greece, Oct. 2001: 341-350
- [27] M. Flierl and B. Girod. Multihypothesis Motion-compensated Prediction with Forward-adaptive Hypothesis Switching. Proceedings of the Picture Coding Symposium, Seoul, Korea, Apr. 2001: 195-198
- [28] B. Girod. Efficiency Analysis of Multihypothesis Motion-compensated Prediction for Video Coding. IEEE Trans. on Image Processing, Feb. 2000, 9: 173-183
- [29] B. Girod. Motion-compensating Prediction with Fractional-pel Accuracy. IEEE Transactions on Communications, Apr. 1993, 41(4): 604-612
- [30] ISO/IEC. Information Technology -- Coding of Audio-visual Objects: Visual. JTC1/SC29/WG11, Tokyo, March 1998: FCD14496-2
- [31] B.-J. Kim and W.A. Pearlman. An Embedded Wavelet Video Coder Using Three-dimensionnal Set Partitioning in Hierarchical Trees (SPIHT). Proceedings of Data Compression Conference, Snowbird, USA, Mar. 1997: 251-260
- [32] A. Wang, Z. Xiong, P. A. Chou and S. Mehrotra. Three-dimensional Wavelet Coding of Video with Global Motion Compensation. Proceedings of Data Compression Conference, Snowbird, USA, Mar. 1999: 404-413
- [33] N. Grammalidis, D. Beletsiotis, and M. G. Strintzis. Sprite Generation and Coding in Multiview Image Sequences. IEEE Trans. on CSVT, March 2000, 10: 302-311
- [34] Yan Lu, Wen Gao, Feng Wu. Fast and Robust Sprite Generation for MPEG-4 Video Coding. The Second IEEE Pacific-Rim Conference on Multimedia (PCM), Beijing, China, October, 2001: 1181-125
- [35] I.H. Witten. Arithmetic Coding for Data Compression. Comm. of the ACM, 1987, 30 (6): 520~541
- [36] Marpe, D., Blgttermann, G., Heising, G., Wiegand, T. Video Compression Using Context-Based Adaptive Arithmetic Coding. Proc. IEEE ICIP. 2001: 558-561
- [37] Gisle bjentegaard and Karl Lillevold. Context Adaptive VLC (CAVLC) Coding of Coefficients. Joint Video Team (JVT) of ISO/IEC MPEG&ITU-T VCEG 3<sup>rd</sup>

- Meeting, May, 2002: C028
- [38]胡伟军, 李克非. 视频压缩标准的最新进展. 中国有线电视, 2003.24: 57-61
  - [39]朱学涛, 向健勇, 张志伟. 数字压缩标准纵横谈. 中国有线电视, 2004, 18: 17-20
  - [40]虞露. AVS-视频技术概述. 中国多媒体视讯, 2004.1: 34-35
  - [41]李青龙. 视频压缩标准的新进展. 广播与电视技术, 2003, 9: 27-29
  - [42]A. Hallapuro, V. Lappalainen, and T.D. Hamalainen, "Performance analysis of low bit rate H.26L video encoder," Proc. ICASSP 2001, vol.2, pp.7-11, May 2001.
  - [43]T. Koga, K. Iinuma, A. Hirano, Y. Lijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proceedings Nat. Telecommunications Conf. 81, New Orleans, LA, pp.G5.3.1-G5.3.5, Nov. 1981.
  - [44]J.Jain, A.Jain, "Displacement measurement and its application in interframe image coding," IEEE Transactions on Communications, vol.COM-29, pp.1799-1806, Dec. 1981.
  - [45]L.Liu, E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Transactions on CSVT, vol.6, no.4, pp.419-422, June 1996
  - [46]Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Patten for Fast Block Motion Estimation", IEEE Trans. on CSVT, pp.349-355, Vol.12, No.5, May, 2002.
  - [47]L.M.Po, W.C.Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Transactions on CSVT, vol.6, no.3, pp.313-317, June 1996.
  - [48]A.M. Tourapis, Enhanced predictive zonal search for single and multiple frame motion estimation. In Proc .of VCIP 2002, pp.1069-79, Jan. 2002.
  - [49]Z.B. Chen, P. Zhou, and Y. He, Fast integer pel and fractional pel motion estimation for JVT.JVT-F017, 6th meeting: Awaji, Japan, 5-13 Dec. 2002.
  - [50]A. Yu, R. Lee, and M. Flynn, Performance enhancement of H.263 encoder based on zero coefficient prediction. In Proc. of ACM Multimedia 1997, pp. 21-29, Nov. 1997.
  - [51]X. Zhou, Z.H. Yu, and S.Y. Yu, Method for detecting all-zero DCT coefficients ahead of discrete cosine transform and quantization. Electronics Letters, vol.34, no.19, pp.1839-40, Sep.1998.
  - [52]J.F. Yang, S.H. Chang and C.Y. Chen, Computation reduction for motion search in low rate video coders. IEEE Trans. On CSVT, vol.12, no.10, pp.948-951, Oct. 2002.
  - [53]Z.B. Chen, P. Zhou, and Y. He, Fast motion estimation for JVT.JVT-G016, 7th meeting: Pattaya, Thailand, 7-14 Mar. 2003.
  - [54]L.B. Yang, K.M. Yu, J. Li, and S.P. Li, An effective variable block-size early termination algorithm for H.264 video coding, accepted by IEEE Trans. on Circuits and System for Video Technology.
  - [55]P. Yin, H.Y. C. Tourapis, A.M. Tourapis and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in Proc. of ICIP 2003, pp.853-856, Sep.2003
  - [56]Z.B. Chen, C. Du, J.H. Wang and Y. He, "PPFPS – A paraboloid prediction based fractional pixel search strategy for H.26L," In Proc. of ISCAS 2002, pp. 9-12,

May 2002

- [57]P. Yin, H.Y. C. Tourapis, A.M. Tourapis and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in Proc. of ICIP 2003, pp.853-856, Sep.2003
- [58]Z.B. Chen and Y. He, "Prediction based Directional Refinement (PDR) algorithm for fractional pixel motion search strategy," JVT-D069, 4th meeting: Klagenfurt, Austria, 22-26 July, 2002
- [59]Cheng Du, Yun He, Junli Zheng, "A paraboloid prediction based fast half pixel motion estimation" Journal of Tsinghua University- Science and Technology, Vol.41(1), pp.1-4, Jan. 2001
- [60]Z. Zhou, M.T. Sun and Y.F. Hsu, "Fast variable block-size motion estimation algorithms based on merge and split procedures for H.264/MPEG-4 AVC," In Proc. of ISCAS 2004, pp.725-728, May 2004
- [61]J.F. Xu, Z.B. Chen and Y. He, "Effective fast ME prediction and early-termination strategy based on H.264 statistical characters," In Proc. of PCM 2003, pp.218-222, Dec. 2003.
- [62]H.C. Tourapis, "Fast motion estimation with the JVT codec," JVT-E023, 5th meeting: Geneva, Switzerland, 09-17 October, 2002.
- [63]S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," In IEEE trans. Image Proceeding, vol.11, pp.1442-49, Dec, 2002
- [64]倪伟, 郭宝龙, 王勇. H.26L 中新的帧内预测技术研究[J]. 计算机工程与应用, 2004,8: 42-45.
- [65]MENG B, AU O C, WONG C W, LAM H K. Efficient intra-prediction algorithm in H.264 [A]. Proceedings of the 2003 IEEE International Conference on Image Processing (ICIP 2003) [C], Barcelona, Spain: IEEE, 2003, 9: 837-840.
- [66]PANG F, LIN X, SUSANTO R, et al. Fast mode decision for intra prediction [A]. JVT-G013, 7th meeting: Pattaya, Thailand, 7-14, March 2003.  
<ftp://ftp.imtc-files.org/jvt-experts/>
- [67]KIM C S, SHIH H H, KUO C J. Multistage mode decision for intra prediction in H.264 codec[A]. Proceedings of Visual Communications and Image Processing 2004 (VCIP 2004) [C], California, USA, 2004: 355-363.
- [68]MENG B, AU O C, WONG C W, LAM H K. Efficient intra-prediction mode selection for 4x4 blocks in H.264 [A]. Proceedings of the 2003 International Conference on Multimedia and Expo (ICME 2003) [C], Baltimore, Maryland: IEEE, 2003, 7: 521-524.
- [69]JVT reference software version 8.0,  
[http://bs.hhi.de/~suehring/tml/download/old\\_jm/](http://bs.hhi.de/~suehring/tml/download/old_jm/)
- [70]杨黎波, 唐慧明, 徐华根, H.264 快速帧内预测算法, 已被浙江大学学报(理学版)录用.

## 硕士在读期间发表的论文

1. 杨黎波, 唐慧明, 徐华根, H.264 快速帧内预测算法, 已被浙江大学学报(理学版)录用.
2. **Libo Yang**, Keman Yu, Jiang Li, and Shipeng Li, "An effective variable block-size early termination algorithm for H.264 video coding", accepted by *IEEE Trans. on Circuits and Systems for Video Technology*.
3. **Libo Yang**, Keman Yu, Jiang Li and Shipeng Li, "Prediction-based Directional Fractional Pixel Motion Estimation for H.264 Video Coding", accepted by *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP 2005)*.



## 致 谢

时光如白驹过隙，还未及细细品味，研究生生涯已接近尾声。在浙大七年的学习实在是带给我太多的东西，并给我烙上了深深的求是烙印，是我一生的宝贵财富。希望以这篇硕士论文作为一小结，为在浙大的求学生涯画上一个圆满的句号。同时，在完成论文的过程中及平时的学习生活中，我得到了很多老师和同学的指导和帮助，在此向他们表示衷心的感谢。

首先，我要感谢我的导师唐慧明老师。唐老师为我确定了研究方向并指导我开展科研工作，使我一步步地接触到奇妙的多媒体技术。唐老师深厚的学术造诣和对技术发展高屋建瓴的把握使我受益匪浅，同时，唐老师更是言传身教，以他那严谨求实的科研作风和谦虚平和的人格魅力深深地折服和感染着我，使我更深刻地明白了今后要以怎样的态度面对自己的工作和生活。

其次，我要感谢虞露老师对我科研工作的指导和帮助，在讨论班的学习为我之后的研究工作打下了坚实的基础，项目的指导使我对自己的方向有了更为深入的理解。与此同时，虞老师对待工作一丝不苟的态度和全身心的投入也使我深受感染。

特别地，我要感谢微软亚洲研究院的余可曼副研究员和李江主任研究员提供我实习的机会，并耐心地指导着我从事高水平的研究和完成高质量的论文，极大地提高了我的研究能力，在世界一流研究机构的熏陶也使我终身受益。

此外，我要感谢韩草铭老师的指导和我择业的帮助。感谢实验室同届的同学徐华根、吴参毅、万春新、雷东壁，在共同的研究生学习生涯中我们互相切磋、互相帮助，分享着彼此的欢乐和困惑，结成了兄弟般的情谊。同时，我要感谢其他同门师兄弟(妹) 杜庸、虞卫峰、崔文杰、曹治锦、蔡荣、林海涵、张芙蓉、刘先虎、王一刚、高国鲁..... 感谢你们的帮助，平时的讨论和交流带给我很多有益的启发，也给我带来了无尽的欢乐和乐趣，使我的研究生生活充满美好的回忆。

最后，我要感谢生我养我的父母，在漫长的求学之路上父母一直是最坚强的后盾，并尽全力提供给我最好的学习条件，悉心地照顾我的生活及各个方面。我唯有以不断的努力来报答父母的养育之恩及殷殷的期望！

杨黎波

2005.2.19 于求是园



作者：[杨黎波](#)  
学位授予单位：[浙江大学](#)  
被引用次数：[2次](#)

参考文献(71条)

- 参考文献
- 范晓鹏 面向高清晰度视频的压缩技术研究[学位论文]硕士 2003
- K Aizawa,H Harashima,T Saito Model-based Analysis Synthesis Image Coding(MBASIC) System for a Person's Face 1989(02)
- Essa Pentland Coding,Analysis,Interpretation,and Recognition of Facial Expressions
- 吴枫 基于对象的运动图象编码技术的研究[学位论文]博士 1999
- Ji-Zheng Xu,Shipeng Li,Ya-Qin Zhang Three-dimensional Shape adaptive Discrete Wavelet Transforms for Efficient Object-based Video coding 2000
- T Sikora,B Makai Shape-adaptive DCT for Generic Coding of Video[外文期刊] 1995
- T Sikora Low Complexity Shape-Adaptive DCT for Coding of Arbitrarily Shaped Image Segments 1995
- K Shen,E J Delp Wavelet Base Rate Scalable Video Compression 1999(01)
- D Wit,T Hou,Y -Q Zhang Scalable Video Coding and Transport over Broadband Wireless Networks 2001(zk)
- W Li Overview of Fine Granularity Scalability in MPEG-4 Video Standard 2001(03)
- Yuwen He,Feng Wu,Shipeng Li,Yuzhuo Zhong,Shiqiang Yang H.26Lbased Fine Granularity Scalable Video Coding 2002
- MPEG-4 ISO/IEC Video Group MPEG-4 Video Verification Model Version 16.0.JTCl/SC29/WG11 2000
- N Ahmed,T Natarajan,K R Rao Discrete Cosine Transforms 1974
- Shapiro Embedded Image Coding Using Zerotree of Wavelet Coefficients 1993(12)
- A Said,W Pearlman ANew,Fast,and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees 1996(03)
- M Boliek,C Christopoulos,E Majani ISO/IEC JEC1/SC29/WG1;JPEG2000 Part I Final Draft International Standard 2000
- ISO/IEC JTCl/SC29/WG1;ISO/IEC JTCl/SC29/WG1.JPEG2000 Verification Model 7.0 2000
- P J Burt,E H Adelson The Laplacian Pyramid as a Compact Image Code 1983(04)
- Katto J Ohki,S Nogaki,M Ohta A Wavelet Codec with Overlapped Motion Compensation for Very Low Bit-rate Environment[外文期刊] 1994
- T Wieg,G Sullivan The Emerging H.264/H.26L Video Coding Standard 2002
- ITU-T/SG 16NCEG,H.26L,Test Model Long Term Number 8 (TML-8) 2001
- T Hallbach Performance Comparison:H.26L Intra Coding vs.JPEG2000 2002
- S Mallat,Z Zhang Matching Pursuits with Time-Frequency Dictionaries 1993(12)
- R Neff,A Zakhor Very Low Bit Rate Video Coding Based on Matching Pursuits 1997(01)
- F Dufaux,J Konrad Efficient,Robust,and Fast Global Motion Estimation for Video Coding 2000
- M Flierl,T Wiegand,B Girod Multihypothesis Pictures for H.26L 2001
- M Flierl,B Girod Multihypothesis Motion-compensated Prediction with Forward-adaptive Hypothesis Switching 2001
- B Girod Efficiency Analysis of Multihypothesis Motion-compensated Prediction for Video Coding[外文期刊] 2000
- B Girod Motion-compensating Prediction with Fractional-pel Accuracy[外文期刊] 1993(04)
- ISO,IEC Information Technology --Coding of Audio-visual Objects:Visual.JTCl/SC29/WG11,FCD 14496-2 1998
- B -J Kim,W A Pearlman An Embedded Wavelet Video Coder Using Three-dimensionnal Set Partitionning in Hierarchical Trees (spiht) 1997
- A Wang,Z Xiong,P A Chou,S.Mehrotra Three-dimensional Wavelet Coding of Video with Global Motion Compensation 1999
- N Grammalidis,D Beletsiotis,M GStrintzis Sprite Generation and Coding in Multiview Image Sequences 2000
- Yan Lu,Wen Gao,Feng Wu Fast and Robust Sprite Generation for MPEG-4 Video Coding 2001
- I H Witten Arithmetic Coding for Data Compression 1987(06)
- Marpe D,Blgttermann G,Heising G,Wiegand,T Video Compression Using Context-Based Adaptive Arithmetic Coding 2001
- Gisle bjentegaard,Karl Lillevold Context Adaptive VLC (CAVLC) Coding of Coefficients 2002
- 胡伟军,李克非 视频压缩标准的最新进展[期刊论文]-中国有线电视 2003(24)
- 朱学涛,向健勇,张志伟 视频压缩标准纵横谈[期刊论文]-中国有线电视 2004(18)
- 虞露 AVS-视频技术概述 2004(01)
- 李青龙 视频压缩标准的新进展—H.264/MPEG-4 Part10[期刊论文]-广播与电视技术 2003(9)
- A Hallapuro,V Lappalainen,T D Hamalainen Performance analysis of low bit rate H.26L video encoder 2001
- T Koga,K Iinuma,A Hirano,Y.Lijima,T.Ishiguro Motion compensated interframe coding for video conferencing 1981
- J Jain,A Jain Displacement measurement and its application in interframe image coding 1981
- L Liu,E Feig A block-based gradient descent search algorithm for block motion estimation in video coding 1996(04)
- Ce Zhu,Xiao Lin,Lap-Pui Chau Hexagon-Based Search Patten for Fast Block Motion Estimation 2002(05)
- L M Po,W C Ma A novel four-step search algorithm for fast block motion estimation 1996(03)
- A M Tourapis Enhanced predictive zonal search for single and multiple frame motion estimation 2002
- Z B Chen,P Zhou,Y He Fast integer pel and fractional pel motion estimation for JVT 2002
- A Yu,R Lee,M Flynn Performance enhancement of H.263 encoder based on zero coefficient prediction 1997
- X Zhou,Z H Yu,S Y Yu Method for detecting all-zero DCT coefficients ahead of discrete cosine transform and quantization 1998(19)
- J F Yang,S H Chang,C Y Chen Computation reduction for motion search in low rate video coders 2002(10)
- Z B Chen,P Zhou,Y He Fast motion estimation for JVT 2003
- L B Yang,K M Yu,J Li,S.P.Li An effective variable block-size early termination algorithm for H.264 video coding
- P Yin,H Y C Tourapis,A M Tourapis,J.Boyce Fast mode decision and motion estimation for JVT/H.264 2003
- Z B Chen,C Du,J H Wang,Y.He PPPFS -A paraboloid prediction based fractional pixel search strategy for H.26L 2002
- P Yin,H Y C Tourapis,A M Tourapis,J.Boyce Fast mode decision and motion estimation for JVT/H.264 2003
- Z B Chen,Y He Prediction based Directional Refinement (PDR) algorithm for fractional pixel motion search strategy 2002
- 杜诚,何芸,郑君里 新的低码率视频编码快速半像素搜索算法[期刊论文]-清华大学学报(自然科学版) 2001(1)
- Z Zhou,M T Sun,Y F Hsu Fast variable block-size motion estimation algorithms based on merge and split procedures for H.264/MPEG-4 AVC 2004
- J F Xu,Z B Chen,Y He Effective fast ME prediction and early-termination strategy based on H.264 statistical characters 2003
- H C Tourapis Fast motion estimation with the JVT codec 2002
- S Zhu,K K Ma A new diamond search algorithm for fast block-matching motion estimation 2002
- 倪伟,郭宝龙,王勇 H.26L中新的帧内预测技术研究[期刊论文]-计算机工程与应用 2004(8)
- MENG B,AU O C,WONG C W,LAM H K Efficient intra-prediction algorithm in H.264 2003
- PANG F,LIN X,SUSANTO R Fast mode decision for intra prediction 2003
- KIM C S,SHIH H H,KUO C J Multistage mode decision for intra prediction in H.264 codec 2004
- MENG B,AU O C,WONG C W,LAM H K Efficient intra-prediction mode selection for 4x4 blocks in H.264 2003
- JVT reference software version 8.0
- 杨黎波,唐慧明,徐华根 H.264快速帧内预测算法

引证文献(2条)

- 王-刚 基于DM642的网络摄像机MPEG-4视频编码实现与优化[学位论文]硕士 2006
- 高国鲁 AVS-M快速运动搜索算法的研究[学位论文]硕士 2006

本文链接：[http://d.g.wanfangdata.com.cn/Thesis\\_Y689212.aspx](http://d.g.wanfangdata.com.cn/Thesis_Y689212.aspx)